
QGIS User Guide

Release 2.6

QGIS Project

22 May 2015

1	Introduzione	3
2	Convenzioni	5
2.1	Convenzioni per l'interfaccia grafica	5
2.2	Convenzioni per il Testo o la Tastiera	5
2.3	Istruzioni specifiche per un sistema operativo	6
3	Premessa	7
4	Caratteristiche	9
4.1	Visualizzazione dati	9
4.2	Esplorare dati e comporre mappe	9
4.3	Creazione, modifica, gestione ed esportazione dati	10
4.4	Analisi dei dati	10
4.5	Pubblicazione di mappe su internet	10
4.6	Estendi le funzionalità di QGIS attraverso i plugin	10
4.7	Console python	11
4.8	Problemi noti	12
5	Novità in QGIS 2.6	13
5.1	Applicazioni e Opzioni del progetto	13
5.2	Sorgenti dati	13
5.3	Compositore di stampe	13
5.4	QGIS Server	14
5.5	Simbologia	14
5.6	Interfaccia utente	14
6	Come Iniziare	15
6.1	Installazione	15
6.2	Dati campione	15
6.3	Sessione di esempio	16
6.4	Avvio e chiusura di QGIS	17
6.5	Opzioni linea di comando	17
6.6	Progetti	19
6.7	Output	20
7	QGIS GUI	21
7.1	Barra dei Menu	22
7.2	Barra degli Strumenti	28
7.3	Legenda	28
7.4	Mappa	31
7.5	Barra di Stato	31

8	Strumenti generali	33
8.1	Scorciatoie da tastiera	33
8.2	Guide contestuali	33
8.3	Visualizzazione	33
8.4	Misurazioni	35
8.5	Informazione elementi	37
8.6	Decorazioni	38
8.7	Note testuali	41
8.8	Segnalibri geospaziali	42
8.9	Progetti nidificati	43
9	Configurazione di QGIS	45
9.1	Pannelli e Barre degli strumenti	45
9.2	Proprietà progetto	46
9.3	Opzioni dell'interfaccia grafica (GUI)	46
9.4	Personalizzazione	55
10	Lavorare con le proiezioni	57
10.1	Panoramica sul supporto alle proiezioni	57
10.2	Specifiche globali delle proiezioni	57
10.3	Definire la riproiezione al volo (OTF)	59
10.4	Sistemi di riferimento personalizzati	60
10.5	Trasformazioni datum predefinite	61
11	QGIS Browser	63
12	Lavorare con i vettori	65
12.1	Formati supportati	65
12.2	The Symbol Library	77
12.3	Proprietà dei vettori	80
12.4	Expressions	108
12.5	Modifica	114
12.6	Costruttore di interrogazioni	131
12.7	Calcolatore di campi	132
13	Lavorare con i dati raster	135
13.1	Lavorare con i dati raster	135
13.2	Proprietà raster	136
13.3	Calcolatore raster	144
14	Lavorare con i dati OGC	147
14.1	QGIS come client dati OGC	147
14.2	QGIS come server dati OGC	156
15	Lavorare con i dati GPS	163
15.1	Plugin GPS	163
15.2	Tracciamento live GPS	167
16	Integrazione con GRASS GIS	173
16.1	Avviare il plugin GRASS	173
16.2	Caricare layer raster e vettoriali GRASS	174
16.3	LOCATION e MAPSET in GRASS	174
16.4	Importare dati nelle LOCATION GRASS	177
16.5	Il modello dati vettoriale di GRASS	177
16.6	Creare un nuovo layer vettoriale GRASS	178
16.7	Digitalizzare e modificare layer vettoriali GRASS	178
16.8	Lo strumento Regione di GRASS	182
16.9	The GRASS Toolbox	182

17 QGIS L'ambiente Processing	191
17.1 Introduzione	191
17.2 Strumenti	192
17.3 Modellatore grafico	201
17.4 L'interfaccia per i processi in serie	207
17.5 Usare gli algoritmi di Processing dalla console dei comandi	209
17.6 Il gestore della cronologia di Processing	214
17.7 Writing new Processing algorithms as python scripts	215
17.8 Handing data produced by the algorithm	217
17.9 Communicating with the user	217
17.10 Documenting your scripts	218
17.11 Example scripts	218
17.12 Best practices for writing script algorithms	218
17.13 Pre- and post-execution script hooks	218
17.14 Configurazione di applicazioni esterne	219
17.15 The QGIS Commander	225
18 Processing providers and algorithms	227
18.1 GDAL algorithm provider	227
18.2 LAStools	260
18.3 Modeler Tools	285
18.4 OrfeoToolbox algorithm provider	287
18.5 QGIS algorithm provider	362
18.6 R algorithm provider	416
18.7 SAGA algorithm provider	426
18.8 TauDEM algorithm provider	597
19 Compositore di stampe	629
19.1 Primi passi	630
19.2 Modalità Visualizzazione	634
19.3 Oggetti del compositore	635
19.4 Gestisci elementi	657
19.5 Strumenti Annulla e Ripristina	658
19.6 Generazione atlante	660
19.7 Creazione del file in output	662
19.8 Gestisci le composizioni di stampa	663
20 Plugin di QGIS	665
20.1 Plugin di QGIS	665
20.2 Uso dei plugin di base di QGIS	671
20.3 Plugin Cattura coordinate	672
20.4 Plugin DB Manager	672
20.5 Plugin Convertitore DXF2Shape	673
20.6 Plugin eVis	674
20.7 Plugin fTools	683
20.8 Plugin GDALTools	686
20.9 Plugin Georeferenziatore	689
20.10 Plugin Interpolazione	693
20.11 Plugin Offline Editing	694
20.12 Oracle Spatial GeoRaster Plugin	695
20.13 Plugin Analisi geomorfologica	697
20.14 Plugin Mappa di concentrazione	698
20.15 Client Catalogo MetaSearch	702
20.16 Plugin grafo strade	705
20.17 Plugin Spatial Query	707
20.18 Plugin SPIT	708
20.19 Plugin SQL Anywhere	708
20.20 Topology Checker Plugin	710
20.21 Plugin Statistica zonale	711

21	Aiuto e supporto	713
21.1	Le Mailing list	713
21.2	IRC	714
21.3	BugTracker	714
21.4	Blog	715
21.5	Plugins	715
21.6	Wiki	715
22	Appendix	717
22.1	GNU General Public License	717
22.2	GNU Free Documentation License	720
23	Letteratura e riferimenti web	727
	Indice	729

·
·

Introduzione

Questo documento è la guida utente originale del software QGIS. I componenti software e hardware descritti nel documento sono, nella maggior parte dei casi, marchi registrati e quindi soggetti a questioni legali. QGIS è soggetto alla licenza GNU General Public. Ulteriori informazioni sono disponibili all'indirizzo <http://www.qgis.org>.

I dettagli, i dati, i risultati ecc. presenti in questo documento sono stati scritti e verificati al meglio delle conoscenze e della responsabilità degli autori e degli editori. Ciononostante, possono essere presenti errori nei contenuti.

Pertanto, nessun dato è soggetto ad alcun obbligo o garanzia. Gli autori, i redattori e gli editori non si assumono alcun obbligo o responsabilità per malfunzionamenti e possibili conseguenze. Tutti gli utenti sono pertanto invitati a segnalare eventuali errori.

Questo documento è stato creato con il software reStructuredText. Il codice sorgente di questo documento è disponibile su [github](https://github.com) e online in formato HTML e PDF <http://www.qgis.org/en/docs/>. Le traduzioni di questo documento possono essere scaricate in diversi formati nella sezione dedicata di QGIS. Per ulteriori informazioni su come contribuire alla documentazione e alla traduzione visita il sito <http://www.qgis.org/wiki/>.

Collegamenti presenti in questo documento

Questo documento contiene collegamenti interni ed esterni. Cliccando su un collegamento interno puoi spostarti all'interno del manuale, mentre cliccando su un collegamento esterno si aprirà un indirizzo internet. In formato PDF i collegamenti interni ed esterni sono mostrati in colore blu e sono gestiti dal browser del sistema operativo. In formato HTML, il browser gestisce e mostra entrambi allo stesso modo.

Autori e redattori delle guide per l'utente, l'installazione e la programmazione:

Tara Athan	Radim Blazek	Godofredo Contreras	Otto Dassau	Martin Dobias
Peter Ersts	Anne Ghisla	Stephan Holl	N. Horning	Magnus Homann
Werner Macho	Carson J.Q. Farmer	Tyler Mitchell	K. Koy	Lars Luthman
Claudia A. Engel	Brendan Morely	David Willis	Jürgen E. Fischer	Marco Hugentobler
Larissa Junek	Diethard Jansen	Paolo Corti	Gavin Macaulay	Gary E. Sherman
Tim Sutton	Alex Bruy	Raymond Nijssen	Richard Duivenvoorde	Andreas Neumann
Astrid Emde	Yves Jacolin	Alexandre Neto	Andy Schmid	Hien Tran-Quang

Copyright (c) 2004 - 2014 QGIS Development Team

Internet: <http://www.qgis.org>

Licenza di questo documento

È garantito il permesso di copiare, distribuire e/o modificare questo documento in base ai termini della GNU Free Documentation License, Versione 1.3 o ogni versione successiva pubblicata dalla Free Software Foundation; senza alcuna sezione non modificabile, senza testo di copertina e retro-copertina. Una copia della licenza è inclusa nell'appendice

Convenzioni

Questa sezione descrive le convenzioni e gli stili che verranno usati in questo manuale.

2.1 Convenzioni per l'interfaccia grafica

Le convenzioni stilistiche per l'interfaccia grafica hanno lo scopo di imitarne l'effettivo aspetto. In generale, lo stile presentato nel manuale fa riferimento a ciò che compare nell'interfaccia grafica e non ai messaggi che compaiono se il cursore del mouse si ferma sopra un pulsante.

- Opzioni di menu: *Layer* → *Aggiungi raster* oppure *Impostazioni* → *Barre degli strumenti* → *Digitalizzazione*
- Strumenti:  *Aggiungi raster*
- Pulsante: **[Salva come predefinito]**
- Titolo finestra di dialogo: *Proprieta layer*
- Scheda (tab): *Generale*
- Casella di controllo: *Visualizzatore*
- Pulsante di scelta: *Postgis SRID* *EPSG ID*
- Seleziona un numero: 
- Seleziona un testo: 
- Cerca un file:
- Seleziona un colore: *Border*
- Cursore:
- Inserimento testo: *Display name*

L'ombreggiatura caratterizza un componente dell'interfaccia grafica che è cliccabile.

2.2 Convenzioni per il Testo o la Tastiera

Questo manuale include anche convenzioni stilistiche relative al testo, a comandi da tastiera e a parti di codice che identificano costrutti diversi come classi o metodi. Questi stili non corrispondono all'attuale aspetto di nessun testo o codice presente in QGIS.

- Collegamenti web: <http://qgis.org>
- Combinazioni di tasti: `Ctrl+B` significa premere il tasto B mentre si tiene premuto il tasto Ctrl.

- Nome di un file: `lakes.shp`
- Nome di una classe: **NewLayer**
- Metodo: `classFactory`
- Server: `myhost.de`
- Inserimento di testo nel terminale: `qgis --help`

I frammenti di codice sono identificati con un carattere a spaziatura fissa:

```
PROJCS["NAD_1927_Albers",  
  GEOGCS["GCS_North_American_1927",
```

2.3 Istruzioni specifiche per un sistema operativo

Sequenze dell'interfaccia grafica e piccole porzioni di testo possono essere formattate linearmente, ad es.: Clicca   **File X QGIS** → *Esci per chiudere QGIS*. Questo significa che su piattaforme Linux, Unix e Windows devi prima cliccare il menu File e poi Esci. Su piattaforme Macintosh OS X devi cliccare prima il menu di QGIS e poi Esci.

I testi di grandi dimensioni possono venire formattati come elenco:

-  fai questo
-  fai quello
- **X** fai qualcos'altro

o come paragrafi:

 **X** Fai questo e questo e questo. Quindi fai questo e questo, e questo.

 Fai quello. Poi fai quello e quello, e quello.

Le schermate riportate nella guida sono state create su diversi sistemi operativi, indicati da apposite icone alla fine della didascalia.

.

Premessa

Benvenuti nel meraviglioso mondo dei Sistemi Informativi Geografici (GIS)!

QGIS è un sistema informativo geografico open source. Il progetto è nato a maggio del 2002 ed è stato confermato come progetto su SourceForge a giugno dello stesso anno. Abbiamo lavorato sodo per creare un software GIS (che normalmente è software proprietario e molto costoso) facilmente installabile e utilizzabile da tutti. QGIS attualmente funziona sulla maggior parte delle piattaforme Unix, Windows e OS X. QGIS viene sviluppato usando il programma Qt (<http://qt.digia.com>) ed è scritto in C++. Questo significa che QGIS ha un'interfaccia utente (GUI) snella, piacevole e facile da usare.

QGIS punta a essere un GIS facilmente utilizzabile da chiunque, fornendo tutte le funzioni e caratteristiche principali. L'obiettivo iniziale del progetto era quello di fornire un visualizzatore di dati GIS. Evolvendosi, QGIS supporta ora moltissimi formati raster e vettoriali, supporto che si amplia con l'uso dei plugin esterni.

QGIS è rilasciato sotto la GNU General Public License (GPL). Sviluppare QGIS con questa licenza significa che puoi ispezionare e modificare il codice sorgente e garantisce che tu, nostro utente, avrai sempre accesso a un programma GIS libero che potrai liberamente modificare. Insieme alla copia di QGIS dovresti aver ricevuto anche una copia completa del testo della licenza che puoi trovare anche nell'Appendice di questo manuale *GNU General Public License*.

Suggerimento: Documentazione aggiornata

Puoi sempre trovare la versione più recente di questo documento sul sito di QGIS all'indirizzo <http://www.qgis.org/en/docs/>.

Caratteristiche

QGIS offre molte delle più comuni funzionalità GIS grazie alle sue caratteristiche di base e ai plugin. Di seguito viene elencato un piccolo riassunto delle sei categorie principali e dei plugin, seguito da una panoramica della console python integrata.

4.1 Visualizzazione dati

Puoi visualizzare e sovrapporre vettori e raster di diversi formati e con diverse proiezioni, senza che sia necessaria alcuna conversione di formato. I formati supportati includono:

- Tabelle e viste spaziali PostGIS, SpatiaLite e MS SQL Spatial, Oracle Spatial e vettori supportati dalla libreria OGR come ESRI shapefile, MapInfo, SDTS, GML e molti altri, vedi la sezione *Lavorare con i vettori*.
- Raster e immagini supportati dalla libreria GDAL (Geospatial Data Abstraction Library), come GeoTIFF, ERDAS IMG, ArcInfo ASCII GRID, JPEG, PNG e molti altri ancora, vedi la sezione *Lavorare con i dati raster*.
- Raster e vettori GRASS dai relativi database (location/mapset), vedi la sezione *Integrazione con GRASS GIS*.
- Dati spaziali accessibili da Web Services OGC, come (WMS, WMTS, WCS, WFS, WFS-T, ...), vedi sezione *Lavorare con i dati OGC*.

4.2 Esplorare dati e comporre mappe

Puoi creare delle mappe ed esplorare i dati spaziali con un'interfaccia grafica molto facile da usare. L'interfaccia grafica ti mette a disposizione molti strumenti, fra cui:

- QGIS browser
- Riproiezione al volo
- DB Manager
- Compositore di stampe
- Pannello vista generale
- Segnalibri spaziali
- Note testuali
- Funzioni di identificazione/selezione
- Modifica/visualizzazione/ricerca degli attributi
- Etichettatura con dati definiti dall'utente

- Simbologia definita dall'utente per vettori e raster
- Creazione atlante
- Freccia nord, barra di scale ed etichetta copyright per le mappe
- Supporto per il salvataggio e il ripristino di progetti

4.3 Creazione, modifica, gestione ed esportazione dati

Puoi creare, modificare, gestire ed esportare i vettori e i raster in molti formati. Ecco un elenco di alcune caratteristiche principali di QGIS:

- Strumenti per la digitalizzazione per i formati OGR e per i vettori GRASS
- Possibilità di creare e modificare shapefile e vettori GRASS
- Plugin georeferenziatore per geocodificare le immagini
- Strumenti GPS per importare ed esportare formati GPX, convertire altri formati GPS in GPX o scaricarli/caricarli direttamente su di una unità GPS (nella versione Linux, usb: è stata aggiunta alla lista degli strumenti GPS)
- Supporto per la visualizzazione e la modifica di dati OpenStreetMap
- Creazione tabelle di database spaziali da shapefile con il plugin DB Manager
- Gestione delle tabelle di database spaziali migliorata
- Strumenti per gestire le tabelle degli attributi di un vettore
- Salvataggio di schermate come immagini georiferite

4.4 Analisi dei dati

Puoi effettuare analisi spaziali su database spaziali e altri formati supportati da OGR. Attualmente QGIS offre strumenti di analisi vettoriale, geoprocessing, geometria e gestione database. Puoi anche utilizzare gli strumenti integrati di GRASS, ovvero avrai completo accesso agli oltre 400 moduli di GRASS (vedi sezione *Integrazione con GRASS GIS*). Puoi anche utilizzare il plugin Processing che permette di effettuare potentissime analisi geospaziali grazie agli algoritmi provenienti da altri programmi come GDAL, SAGA, GRASS, fTools e molti altri ancora (vedi sezione *Introduzione*).

4.5 Pubblicazione di mappe su internet

QGIS può essere utilizzato come client WMS, WMTS, WMS-C e WFS-T e come server WMS, WFS o WCS (vedi sezione *Lavorare con i dati OGC*). Inoltre puoi esportare e pubblicare i tuoi dati in internet usando un webservice con installato UMN MapServer o GeoServer.

4.6 Estendi le funzionalità di QGIS attraverso i plugin

Puoi adattare QGIS ai tuoi scopi grazie all'architettura estensibile dei plugin. QGIS fornisce librerie che possono essere usate per la creazione di plugin. Ma puoi anche creare le tue nuove applicazioni con C++ o python!

4.6.1 Plugin nativi

I plugin nativi includono:

1. Cattura coordinate (cattura le coordinate, tramite il mouse, nei diversi Sistemi di Riferimento)
2. DB Manager (scambia, modifica e visualizza layer e tabelle; esegue interrogazioni in SQL)
3. Diagrammi (aggiunge diagrammi su un vettore)
4. Convertitore Dxf2Shp (converte file DXF in shapefile)
5. eVIS (visualizza eventi)
6. fTools (analisi e gestione di vettori)
7. Strumenti GDAL (Strumenti GDAL integrati in QGIS)
8. Georeferenziatore raster (aggiunge ai raster informazioni sulla proiezione utilizzando GDAL)
9. Strumenti GPS (carica e importa dati GPS)
10. GRASS (integrazione con GRASS)
11. Mappe di concentrazione (Genera delle mappe di concentrazione raster partendo da dati puntuali)
12. Plugin interpolazione (interpolazione basata sui vertici in un vettore)
13. Offline Editing (consente le modifiche offline e la sincronizzazione con un database)
14. Oracle Spatial Georaster
15. Processing (SEXTANTE nelle versioni precedenti)
16. Plugin per l'analisi geomorfologica (analisi del terreno basata su raster)
17. Grafo strade (analisi del percorso più breve)
18. Plugin Interrogazione spaziale
19. SPIT (importa uno shapefile in PostgreSQL/PostGIS)
20. Plugin SQL Anywhere (immagazzina vettori in un database SQL Anywhere)
21. Validatore topologico (trova errori topologici in un vettore)
22. Plugin statistiche zonali (calcola il conteggio, la somma, la media di un raster per ogni poligono di un vettore)

4.6.2 Plugin esterni in python

QGIS offre un crescente numero di plugin python esterni creati dalla comunità. Questi plugin sono presenti all'interno del repository ufficiale dei plugin e possono essere facilmente installati usando l'installatore dei plugin python (vedi sezione *La finestra di dialogo Plugins*).

4.7 Console python

Se usi script, puoi sfruttare la console python integrata, accessibile dal menu *Plugin* → *console python*. La console si apre come una finestra di dialogo non modale. Per interagire con l'ambiente di QGIS esiste una variabile `qgis.utils iface` che è un'istanza di `QgsInterface`. Questa interfaccia ti permette di accedere alla mappa, ai menu, alle barre degli strumenti e ad altre applicazioni di QGIS.

Per maggiori informazioni su come utilizzare la console python e programmare plugin per QGIS, fai riferimento a questa risorsa http://www.qgis.org/html/en/docs/pyqgis_developer_cookbook/index.html.

4.8 Problemi noti

4.8.1 Limitazione numero di file aperti

Se stai aprendo un grande progetto di QGIS e sei sicuro che tutti i layer sono validi, ma qualche layer viene segnalato come corrotto, probabilmente ti stai scontrando con questo problema. Linux (e probabilmente anche altri sistemi operativi) hanno un limite di file aperti per ogni processo. I limiti delle risorse e per ogni processo vengono automaticamente ereditati. Il comando `ulimit`, preinstallato nella console dei comandi, cambia i limiti solo per il processo attuale; il nuovo limite viene ereditato da ogni altro processo.

Puoi vedere tutti gli `ulimit` attuali digitando

```
user@host:~$ ulimit -aS
```

Poi vedere l'attuale numero permesso di file aperti per ogni processo con questo comando da console

```
user@host:~$ ulimit -Sn
```

Per cambiare i limiti di una **sessione esistente**, potresti usare qualcosa del genere

```
user@host:~$ ulimit -Sn #number_of_allowed_open_files
user@host:~$ ulimit -Sn
user@host:~$ qgis
```

Risolverlo per sempre

Sulla maggior parte dei sistemi Linux, i limiti alle risorse sono impostati al momento del login tramite il modulo `pam_limits` in funzione delle impostazioni contenute in `/etc/security/limits.conf` o `/etc/security/limits.d/*.conf`. Dovresti modificare questi file se hai i permessi di amministratore (anche tramite `sudo`), ma dovrai effettuare di nuovo il login prima che i cambiamenti siano effettivi.

Maggiori informazioni:

<http://www.cyberciti.biz/faq/linux-increase-the-maximum-number-of-open-files/> <http://linuxaria.com/article/open-files-in-linux?lang=en>

.

Novità in QGIS 2.6

This release contains new features and extends the programmatic interface over previous versions. We recommend that you use this version over previous releases.

This release includes hundreds of bug fixes and many new features and enhancements that will be described in this manual. You may also review the visual changelog at <http://changelog.linfiniti.com/qgis/version/2.6.0/>.

5.1 Applicazioni e Opzioni del progetto

- **Project filename in properties:** You can now see the full path for the QGIS project file in the project properties dialog.

5.2 Sorgenti dati

- **DXF Export tool improvements:**
 - Tree view and attribute selection for layer assignment in dialog
 - support fill polygons/HATCH
 - represent texts as MTEXT instead of TEXT (including font, slant and weight)
 - support for RGB colors when there's no exact color match
 - use AutoCAD 2000 DXF (R15) instead of R12

5.3 Compositore di stampe

- **Update map canvas extent from map composer extent:** **On the Item** properties of a Map element there are now two extra buttons which allow you to (1) set the Map canvas extent according with the extent of your Map element and (2) view in Map canvas the extent currently set on your Map element.
- **Multiple grid support:** It is now possible to have more than one grid in your Map element. Each grid is fully customizable and can be assigned to a different CRS. This means, for example, you can now have a map layout with both geographic and projected grids.
- **Selective export:** To every item of your map composer layout, under Rendering options, you may exclude that object from map exports.

5.4 QGIS Server

5.5 Simbologia

5.6 Interfaccia utente

Come Iniziare

Questo capitolo fornisce una rapida panoramica sull'installazione di QGIS, su alcuni dati campione scaricabili dal sito di QGIS e su come avviare una prima semplice sessione in cui visualizzare raster e vettori.

6.1 Installazione

L'installazione di QGIS è molto semplice. Pacchetti standard per l'installazione sono disponibili per MS Windows e Mac OS X. Per le distribuzioni GNU/Linux sono disponibili pacchetti binari (rpm e deb) o repository da aggiungere al gestore di installazione. Maggiori informazioni sul sito <http://download.qgis.org>.

6.1.1 Installazione da codice sorgente

Se devi compilare QGIS da codice sorgente, fai riferimento alle istruzioni di installazione. Queste sono fornite insieme al codice sorgente di QGIS in un file chiamato 'INSTALL'. Puoi trovare questo file all'indirizzo <http://htmlpreview.github.io/?https://raw.githubusercontent.com/qgis/QGIS/master/doc/INSTALL.html>

6.1.2 Installazione su supporti esterni

Puoi avviare QGIS con l'opzione `--configpath`; questa andrà a sostituire il percorso predefinito delle cartelle create da qgis (per esempio, `~/qgis2` in Linux) e obbligherà **QSettings** ad utilizzare questa cartella. Questa opzione permette di creare un'installazione personalizzata di QGIS, inclusi plugin e impostazioni, e di copiarla su un supporto esterno (penna USB). Vedi la sezione *Menu Sistema* per ulteriori informazioni.

6.2 Dati campione

La guida utente presenta alcuni esempi basati sull'insieme di dati campione di QGIS.

 Durante l'installazione di QGIS in Windows hai la possibilità di scaricare un insieme di dati campione. Se hai selezionato questa opzione, i dati verranno scaricati nella cartella *Documenti* e verranno copiati in una cartella chiamata *GIS Database*. Puoi usare Windows Explorer per spostare questa cartella in qualunque altra posizione. Se non hai selezionato l'opzione per scaricare l'insieme di dati campione durante l'installazione iniziale di QGIS puoi scegliere fra:

- usare dati GIS che hai già;
- scaricare i dati campione da http://download.osgeo.org/qgis/data/qgis_sample_data.zip
- disinstallare QGIS e reinstallarlo selezionando l'opzione per lo scaricamento dei dati (opzione consigliata solo se le soluzioni precedenti non sono riuscite)

 **X** Per GNU/Linux e Mac OSX non ci sono ancora pacchetti in formato rpm, deb o dmg. Per utilizzare l'insieme di dati campione scaricare il file `:file: qgis_sample_data` come archivio ZIP da http://download.osgeo.org/qgis/data/qgis_sample_data.zip e decomprimere l'archivio sul tuo computer.

L'insieme di dati Alaska comprende tutti i dati GIS usati come esempi e schermate nel manuale d'uso, e include anche un piccolo database GRASS. La proiezione per l'insieme di dati campione di QGIS è Alaska Albers Equal Area con unità in piedi. Il codice EPSG è 2964.

```
PROJCS["Albers Equal Area",
GEOGCS["NAD27",
DATUM["North_American_Datum_1927",
SPHEROID["Clarke 1866",6378206.4,294.978698213898,
AUTHORITY["EPSG","7008"]],
TOWGS84[-3,142,183,0,0,0,0],
AUTHORITY["EPSG","6267"]],
PRIMEM["Greenwich",0,
AUTHORITY["EPSG","8901"]],
UNIT["degree",0.0174532925199433,
AUTHORITY["EPSG","9108"]],
AUTHORITY["EPSG","4267"]],
PROJECTION["Albers_Conic_Equal_Area"],
PARAMETER["standard_parallel_1",55],
PARAMETER["standard_parallel_2",65],
PARAMETER["latitude_of_center",50],
PARAMETER["longitude_of_center",-154],
PARAMETER["false_easting",0],
PARAMETER["false_northing",0],
UNIT["us_survey_feet",0.3048006096012192]]
```

Se vuoi usare QGIS come interfaccia grafica per GRASS, scarica una location campione di GRASS (per esempio Spearfish o South Dakota) direttamente dal sito ufficiale <http://grass.osgeo.org/download/sample-data/>.

6.3 Sessione di esempio

Ora che hai installato QGIS e hai a disposizione un insieme di dati campione, ti faremo vedere una breve e semplice dimostrazione. Visualizzeremo un raster e un vettore. Useremo il raster `landcover`, `qgis_sample_data/raster/landcover.img`, e il vettore `lakes`, `qgis_sample_data/gml/lakes.gml`.

6.3.1 Avvio di QGIS

-  Avvia QGIS digitando “QGIS” nel terminale, oppure, se hai installato QGIS con pacchetti binari precompilati, trovalo nel menu delle applicazioni.
-  Avvia QGIS usando il menu Start o l'icona sul desktop, oppure fai doppio click su un file di progetto QGIS.
- **X** Doppio click sull'icona nella cartella Applicazioni.

6.3.2 Caricare raster e vettori dall'insieme di dati campione

1. Clicca sull'icona  `Aggiungi Raster`.
2. Trova la cartella `qgis_sample_data/raster/`, seleziona il file ERDAS IMG `landcover.img` e clicca **[Apri]**.
3. Se il file non è in elenco, controlla nella casella in basso della finestra *Files of type*  e inserisci “Erdas Image Images (*.img, *.IMG)”.

4. Ora clicca sull'icona  **Aggiungi Vettore**.
5.  **File** deve essere selezionato come *Tipo di sorgente* nella finestra di dialogo *Aggiungi vettore*. Ora clicca su **[Sfogli]** per selezionare il vettore.
6. Scorri la cartella `qgis_sample_data/gml/`, seleziona 'Geography Markup Language [GML] [OGR] (.gml,.GML)' dalla casella combinata *Files of type* , poi seleziona il file `GML lakes.gml` e click **[Apri]**. Nella finestra di dialogo *Aggiungi vettore*, click **[OK]**. La finestra di dialogo *Selettore del sistema di riferimento* si apre con *NAD27 / Alaska Albers* selezionato, click **[OK]**.
7. Ingrandisci la mappa su un'area con alcuni laghi.
8. Fai doppio click sul vettore `lakes` nella legenda per aprire la finestra di dialogo *Proprietà*.
9. Clicca sulla scheda *Stile* e seleziona blu come colore di riempimento.
10. Clicca sulla scheda *Etichette* e spunta la casella di controllo  *Etichetta questo valore con* per abilitare l'etichettatura. Scegli il campo "NAMES" come campo per l'etichetta.
11. Per migliorare la leggibilità delle etichette, puoi aggiungere un contorno bianco cliccando su "Contorno" nell'elenco a sinistra. Spunta la casella  *Disegna contorno del testo* e scegli 3 come dimensione.
12. Clicca **[Applica]**, controlla se il risultato è buono e infine clicca **[OK]**.

Hai visto come è facile visualizzare raster e vettori in QGIS!? Prosegui con la sezione successiva per imparare ulteriori funzionalità, caratteristiche ed impostazioni.

6.4 Avvio e chiusura di QGIS

Nella sezione *Sessione di esempio* abbiamo già visto come avviare QGIS. Ora ripetiamo questa operazione per dimostrare come QGIS fornisca ulteriori opzioni all'avvio da riga di comando.

-  Assumendo che QGIS sia installato nel tuo PATH, puoi avviarlo semplicemente digitando `qgis` nel terminale oppure facendo doppio click sull'icona di QGIS presente sul desktop o nel menu delle applicazioni.
-  Avvia QGIS usando il menu Start o l'icona sul desktop, oppure fai doppio click su un file di progetto QGIS.
- **X** Fai doppio click sull'icona di QGIS nella cartella Applicazioni (Applications). Se vuoi avviare QGIS da una terminale, digita `/percorso-installazione-eseguibile/Contents/MacOS/Qgis`.

Per uscire da QGIS, clicca il menu opzioni   *Progetto* **X** *QGIS* → *Esci*, oppure usa la scorciatoia `Ctrl+Q`.

6.5 Opzioni linea di comando

 QGIS supporta un certo numero di opzioni se avviato da riga di comando. Per avere una lista delle opzioni possibili, digita "`qgis -help`" nel terminale. La corretta sintassi di QGIS è:

```
qgis --help
QGIS - 2.6.0-Brighton 'Brighton' (exported)
QGIS is a user friendly Open Source Geographic Information System.
Usage: /usr/bin/qgis.bin [OPTION] [FILE]
OPTION:
  [--snapshot filename]  emit snapshot of loaded datasets to given file
  [--width width]        width of snapshot to emit
  [--height height]      height of snapshot to emit
  [--lang language]      use language for interface text
  [--project projectfile] load the given QGIS project
  [--extent xmin,ymin,xmax,ymax] set initial map extent
```

```
[--nologo]      hide splash screen
[--noplugins]   don't restore plugins on startup
[--nocustomization] don't apply GUI customization
[--customizationfile] use the given ini file as GUI customization
[--optionspath path] use the given QSettings path
[--configpath path] use the given path for all user configuration
[--code path]   run the given python file on load
[--defaultui]   start by resetting user ui settings to default
[--help]       this text
```

FILE:

Files specified on the command line can include rasters, vectors, and QGIS project files (.qgs):

1. Rasters - supported formats include GeoTiff, DEM and others supported by GDAL
2. Vectors - supported formats include ESRI Shapefiles and others supported by OGR and PostgreSQL layers using the PostGIS extension

Suggerimento: Esempio di utilizzo delle opzioni da riga di comando

Puoi avviare QGIS con dei file grazie all'opzione fornita da riga di comando. Per esempio, ipotizzando di trovarsi nella cartella `qgis_sample_data`, puoi avviare QGIS con un vettore e un raster inserendo il comando: `qgis ./raster/landcover.img ./gml/lakes.gml`

Opzioni linea di comando `--snapshot`

L'opzione consente di catturare una schermata in formato PNG della mappa. Utile quando hai molti progetti e vuoi generare schermate dai propri dati.

Il file PNG generato ha una risoluzione di 800x600 pixels. Puoi adattare la risoluzione grazie agli argomenti `--width` e `--height` da riga di comando. Puoi anche aggiungere il nome del file dopo l'argomento `--snapshot`.

Opzioni linea di comando `--lang`

QGIS seleziona automaticamente la lingua in funzione delle impostazioni di localizzazione del sistema. Puoi scegliere di avviare QGIS con un'altra lingua. Ad esempio, aggiungendo l'opzione `--lang=en`, QGIS sarà localizzato in inglese. L'elenco delle lingue supportate è disponibile all'indirizzo http://hub.qgis.org/wiki/quantum-gis/GUI_Translation_Progress.

Opzioni linea di comando `--project`

Puoi avviare QGIS anche con un file di progetto. Basta semplicemente aggiungere l'opzione `--project` alla riga di comando seguita dal percorso e dal nome del progetto. In questo modo QGIS si aprirà caricando tutti i layer indicati nel file specificato.

Opzioni linea di comando `--extent`

Per avviare QGIS con un specifica estensione devi aggiungere i confini della bounding box in questo ordine e separati da una virgola:

```
--extent xmin,ymin,xmax,ymax
```

Opzioni linea di comando `--nologo`

Questa opzione nasconde lo splash screen all'avvio di QGIS.

Opzioni linea di comando `--noplugins`

Se all'avvio di QGIS si verificano problemi con i plugin puoi evitare di caricarli con questa opzione. I plugin rimarranno comunque disponibili nel Gestore plugin.

Opzioni linea di comando `--customizationfile`

Usando questa opzione puoi specificare un file di personalizzazione dell'interfaccia grafica che verrà caricato all'avvio.

Opzioni linea di comando `--nocustomization`

Usando questa opzione le personalizzazioni dell'interfaccia non verranno applicate all'avvio.

Opzioni linea di comando `--optionspath`

Puoi avere più configurazioni e decidere quale usare all'avvio QGIS con questa opzione. Vedi *Opzioni dell'interfaccia grafica (GUI)* per confermare dove il sistema operativo salva i file di impostazioni. Attualmente, non c'è modo di specificare un file da scrivere impostazioni; Pertanto, puoi creare una copia del file impostazioni originali e rinominarlo. L'opzione specifica il percorso alla cartella con le impostazioni. Ad esempio, per utilizzare il file di impostazioni `/path/to/config/QGIS/QGIS2.ini`, utilizza:

```
--optionspath /path/to/config/
```

Opzioni linea di comando `--configpath`

Questa opzione è simile alla precedente, ma in più sovrascrive il percorso predefinito delle cartelle create da QGIS (`~/qgis2`) e obbligherà **QSettings** ad utilizzare questa cartella. Questa opzione permette di creare un'installazione personalizzata di QGIS, inclusi plugin e impostazioni, e di copiarla su un supporto esterno (penna USB).

Opzioni linea di comando `--code`

Puoi utilizzare questa opzione per eseguire un file python subito dopo l'avvio di QGIS.

Per esempio, se hai un file python `load_alaska.py` con il seguente contenuto:

```
from qgis.utils import iface
raster_file = "/home/gisadmin/Documents/qgis_sample_data/raster/landcover.img"
layer_name = "Alaska"
iface.addRasterLayer(raster_file, layer_name)
```

Dando per scontato che sei nella cartella dove si trova il `load_alaska.py`, puoi avviare QGIS, caricare il file raster `landcover.img` e dare il layer il nome di 'Alaska' con il seguente comando: `qgis --code load_alaska.py`

6.6 Progetti

Lo stato della sessione QGIS è considerato un Progetto. QGIS lavora su un progetto alla volta. Le impostazioni sono considerate proprie di un progetto oppure predefinite per nuovi progetti (vedi la sezione *Opzioni dell'interfaccia grafica (GUI)*). QGIS può salvare lo stato del tuo spazio di lavoro in un file di progetto usando il menu opzioni *Progetto* →  *Salva 'o :menuselection:'Progetto* →  *Salva Come...*

Carica i progetti memorizzati in una sessione di QGIS usando *Progetto* →  *Apri ...*, *Progetto* → *Nuovo da modello* oppure *Progetto* → *Apri recenti* →.

Se vuoi iniziare una nuova sessione, scegli *Progetto* →  *Nuovo*. Se hai apportato qualche modifica al progetto, ti verrà chiesto se lo vuoi salvare.

Le informazioni salvate nel file di progetto includono:

- Layer aggiunti
- Proprietà dei layer, inclusa la simbologia
- Proiezione usata per la mappa
- Ultima estensione della mappa

Il file di progetto è un file XML; in questo modo, se conosci la sintassi XML, puoi modificarlo con un qualunque editor di testo. Il formato del file di progetto è stato modificato parecchie volte rispetto a quello delle precedenti versioni di QGIS, quindi i progetti salvati con versioni precedenti di QGIS potrebbero non funzionare più correttamente. Caricando un progetto di un'altra versione di QGIS sarai avvertito con un messaggio. Puoi scegliere di disattivare questo avviso dalla scheda *Generale* del menu *Impostazioni* → *Opzioni*:

-  Chiedi di salvare il progetto e cambia sorgenti dati quando richiesto
-  Avvisa quando viene aperto un file di progetto salvato con una vecchia versione di QGIS

Ogni volta che salvi un progetti in QGIS 2.2 verrà creata automaticamente una copia di backup.

6.7 Output

Ci sono diversi modi per generare file di output da una sessione QGIS. Uno è già stato discusso nella sezione *Progetti*, ovvero il salvataggio di un progetto. Qui un elenco di altri file di output che puoi generare:

- L'opzione *Progetto* →  Salva come immagine apre una finestra di dialogo dove puoi scegliere il nome, il percorso e il tipo di immagine (formato PNG e JPG). L'immagine è georeferenziata grazie al file world (PNGW o JPGW).
- L'opzione *Progetto* → *Esportazione DXF ...* apre una finestra di dialogo in cui puoi definire la 'Modalità simbologia', la 'Scala simbologia' e i vettori che vuoi esportare come DXF.
- L'opzione: menuSelection:*Progetto* →  :menuSelection: *Nuova composizione di stampa* apre una finestra di dialogo in cui puoi creare una composizione di stampa della mappa (vedi sezione *Compositore di stampe*).

QGIS GUI

All'avvio di QGIS l'interfaccia grafica si presenta come mostrato in figura (i numeri da 1 a 5 cerchiati di giallo fanno riferimento alle 5 aree principali dell'interfaccia grafica come verrà in seguito discusso):

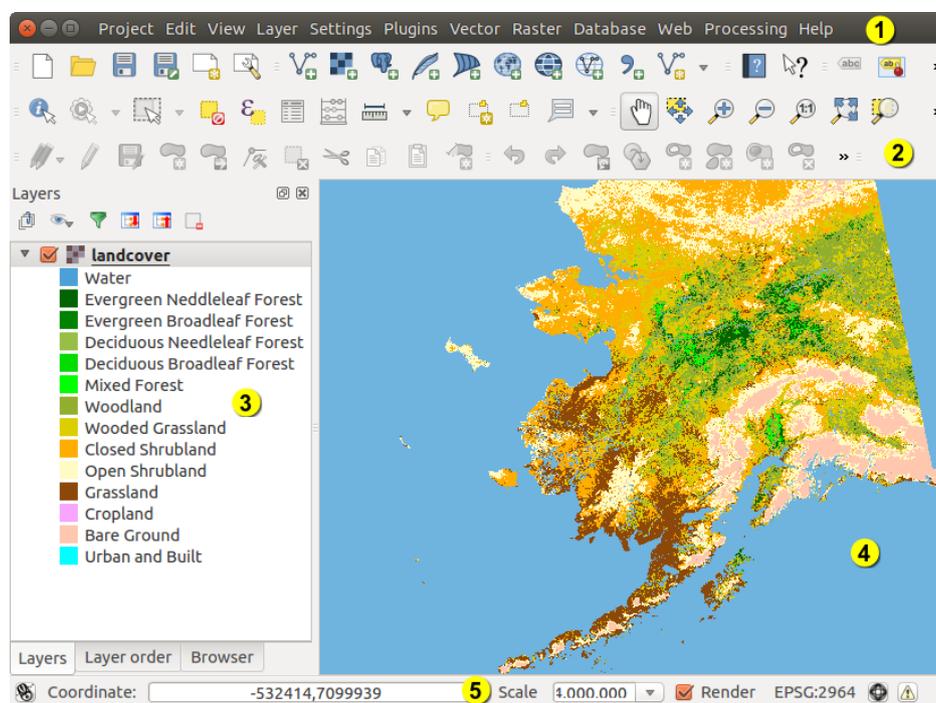


Figura 7.1: QGIS GUI con i dati campione Alaska 🐧

Nota: L'aspetto delle finestre (barra del titolo, ecc.) potrà apparire diverso a seconda del sistema operativo e dell'ambiente desktop.

L'interfaccia di QGIS è divisa in cinque aree:

1. Barra dei Menu
2. Barra degli Strumenti
3. Legenda
4. Mappa
5. Barra di Stato

I cinque componenti dell'interfaccia di QGIS sono descritti con maggior dettaglio nelle sezioni seguenti. Due ulteriori sezioni illustreranno le scorciatoie da tastiera e la guida contestuale.

7.1 Barra dei Menu

La barra dei menu fornisce accesso alle varie caratteristiche di QGIS utilizzando un menu gerarchico standard. I menu al livello superiore e una sintesi di alcune opzioni del menu sono elencate di seguito, insieme alle icone corrispondenti così come appaiono nella barra degli strumenti ed alle scorciatoie da tastiera. Le scorciatoie da tastiera possono essere configurate manualmente (quelle presentate in questa sezione sono quelle predefinite), usando l'opzione *Configura le scorciatoie...* del menu *Impostazioni*.

Anche se la maggior parte dei menu ha uno strumento corrispondente (e viceversa), i menu non sono organizzati come le barre degli strumenti. Gli strumenti contenuti in queste ultime infatti sono identificate con una casella di controllo nel menu corrispondente. Alcuni strumenti sono visibili solamente se il plugin corrispondente è attivo. Per maggiori informazioni sugli strumenti e sulle barre degli strumenti, vedi la sezione *Barra degli Strumenti*.

7.1.1 Progetto

Voce di Menu	Scorciatoia	Riferimento	Barra degli Strumenti
 <i>Nuovo</i>	Ctrl+N	vedi <i>Progetti</i>	<i>Progetto</i>
 <i>Apri</i>	Ctrl+O	vedi <i>Progetti</i>	<i>Progetto</i>
<i>Nuovo da modello →</i>		vedi <i>Progetti</i>	<i>Progetto</i>
<i>Apri recenti →</i>		vedi <i>Progetti</i>	
 <i>Salva</i>	Ctrl+S	vedi <i>Progetti</i>	<i>Progetto</i>
 <i>Salva Come...</i>	Ctrl+Shift+S	vedi <i>Progetti</i>	<i>Progetto</i>
 <i>Salva come immagine...</i>		vedi <i>Output</i>	
<i>Esportazione DXF...</i>		vedi <i>Output</i>	
 <i>Nuova composizione di stampa</i>	Ctrl+P	vedi <i>Compositore di stampe</i>	<i>Progetto</i>
 <i>Gestore di stampe ...</i>		vedi <i>Compositore di stampe</i>	<i>Progetto</i>
<i>Stampe →</i>		vedi <i>Compositore di stampe</i>	
 <i>Esci da QGIS</i>	Ctrl+Q		

7.1.2 Modifica

Voce di Menu	Scorciatoia	Riferimento	Barra degli Strumenti
 <i>Annulla</i>	Ctrl+Z	vedi <i>Digitalizzazione avanzata</i>	<i>Digitalizzazione avanzata</i>
 <i>Ripristina</i>	Ctrl+Shift+Z	vedi <i>Digitalizzazione avanzata</i>	<i>Digitalizzazione avanzata</i>
 <i>Taglia geometrie</i>	Ctrl+X	vedi <i>Modifica di un layer esistente</i>	<i>Digitalizzazione esistente</i>
 <i>Copia elementi</i>	Ctrl+C	vedi <i>Modifica di un layer esistente</i>	<i>Digitalizzazione esistente</i>
 <i>Incolla elementi</i>	Ctrl+V	vedi <i>Modifica di un layer esistente</i>	<i>Digitalizzazione esistente</i>
<i>Incolla elementi come →</i>		vedi <i>Lavorare con la tabella degli attributi</i>	
 <i>Inserisci punto</i>	Ctrl+.	vedi <i>Modifica di un layer esistente</i>	<i>Digitalizzazione esistente</i>
 <i>Muovi elemento/i</i>		vedi <i>Modifica di un layer esistente</i>	<i>Digitalizzazione esistente</i>
 <i>Elimina il selezionato</i>		vedi <i>Modifica di un layer esistente</i>	<i>Digitalizzazione esistente</i>
 <i>Ruota i simboli per i punti</i>		vedi <i>Digitalizzazione avanzata</i>	<i>Digitalizzazione avanzata</i>
 <i>Semplifica geometria</i>		vedi <i>Digitalizzazione avanzata</i>	<i>Digitalizzazione avanzata</i>
 <i>Aggiungi un buco</i>		vedi <i>Digitalizzazione avanzata</i>	<i>Digitalizzazione avanzata</i>
 <i>Aggiungi una parte</i>		vedi <i>Digitalizzazione avanzata</i>	<i>Digitalizzazione avanzata</i>
 <i>Riempi buco</i>		vedi <i>Digitalizzazione avanzata</i>	<i>Digitalizzazione avanzata</i>
 <i>Elimina buco</i>		vedi <i>Digitalizzazione avanzata</i>	<i>Digitalizzazione avanzata</i>
 <i>Elimina parte</i>		vedi <i>Digitalizzazione avanzata</i>	<i>Digitalizzazione avanzata</i>
 <i>Modifica la forma</i>		vedi <i>Digitalizzazione avanzata</i>	<i>Digitalizzazione avanzata</i>
 <i>Taglia geometrie</i>		vedi <i>Digitalizzazione avanzata</i>	<i>Digitalizzazione avanzata</i>
 <i>Spezza elemento</i>		vedi <i>Digitalizzazione avanzata</i>	<i>Digitalizzazione avanzata</i>
 <i>Dividi parti</i>		vedi <i>Digitalizzazione avanzata</i>	<i>Digitalizzazione avanzata</i>
 <i>Unisci gli elementi selezionati</i>		vedi <i>Digitalizzazione avanzata</i>	<i>Digitalizzazione avanzata</i>
 <i>Unisci gli attributi degli elementi selezionati</i>		vedi <i>Digitalizzazione avanzata</i>	<i>Digitalizzazione avanzata</i>
 <i>Strumento vertici</i>		vedi <i>Modifica di un layer esistente</i>	<i>Digitalizzazione esistente</i>

Quando si attiva la modalità  *Modifica* per un vettore, si attiveranno altri pulsanti nel menu *Modifica* in funzione del tipo di geometria (punto, linea o poligono).

7.1.3 Modifica (extra)

Voce di Menu	Scorciatoia	Riferimento	Barra degli Strumenti
 <i>Inserisci punto</i>		vedi <i>Modifica di un layer esistente</i>	<i>Digitalizzazione</i>
 <i>Inserisci linea</i>		vedi <i>Modifica di un layer esistente</i>	<i>Digitalizzazione</i>
 <i>Inserisci poligono</i>		vedi <i>Modifica di un layer esistente</i>	<i>Digitalizzazione</i>

7.1.4 Mappa

Voce di Menu	Scorciatoia	Riferimento	Barra degli Strumenti
 <i>Sposta mappa</i>			<i>Navigazione mappa</i>
 <i>Zoom sulla selezione</i>			<i>Navigazione mappa</i>
 <i>Ingrandisci</i>	Ctrl++		<i>Navigazione mappa</i>
 <i>Rimpicciolisci</i>	Ctrl+-		<i>Navigazione mappa</i>
<i>Seleziona →</i>		vedi <i>Selezionare e deselezionare elementi</i>	<i>Attributi</i>
 <i>Informazioni elementi</i>	Ctrl+Shift+I		<i>Attributi</i>
<i>Misura →</i>		vedi <i>Misurazioni</i>	<i>Attributi</i>
 <i>Zoom completo</i>	Ctrl+Shift+F		<i>Navigazione mappa</i>
 <i>Zoom sul layer</i>			<i>Navigazione mappa</i>
 <i>Zoom sulla selezione</i>	Ctrl+J		<i>Navigazione mappa</i>
 <i>Ultimo Zoom</i>			<i>Navigazione mappa</i>
 <i>Zoom successivo</i>			<i>Navigazione mappa</i>
 <i>Rimpicciolisci</i>			<i>Navigazione mappa</i>
<i>Proprietà →</i>		vedi <i>Decorazioni</i>	
 <i>Suggerimenti mappa</i>			<i>Attributi</i>
 <i>Nuovo segnalibro</i>	Ctrl+B	vedi <i>Segnalibri geospaziali</i>	<i>Attributi</i>
 <i>Mostra segnalibri</i>	Ctrl+Shift+B	vedi <i>Segnalibri geospaziali</i>	<i>Attributi</i>
 <i>Aggiorna</i>	Ctrl+R		<i>Navigazione mappa</i>

7.1.5 Layer

Voce di Menu	Scorciatoia	Riferimento	Barra degli Strumenti
<i>Nuovo →</i>		vedi <i>Creating new Vector layers</i>	<i>Gestione layer</i>
<i>Includi layer e gruppi</i>		vedi <i>Progetti nidificati</i>	
 <i>Aggiungi vettore</i>	Ctrl+Shift+V	vedi <i>Lavorare con i vettori</i>	<i>Gestione layer</i>
 <i>Aggiungi raster</i>	Ctrl+Shift+R	vedi <i>Caricare dati raster in QGIS</i>	<i>Gestione layer</i>
 <i>Aggiungi vettore PostGIS</i>	Ctrl+Shift+D	vedi <i>Vettori PostGIS</i>	<i>Gestione layer</i>
 <i>Aggiungi vettore Spatialite</i>	Ctrl+Shift+L	vedi <i>Vettori Spatialite</i>	<i>Gestione layer</i>

Continua alla pagina successiva

Tabella 7.1 – continua dalla pagina precedente

Voce di Menu	Scorciatoia	Riferimento	Barra degli Strumenti
 <i>Aggiungi vettore MSSQL</i>	Ctrl+Shift+M	vedi <i>Vettori Spatial MSSQL</i>	<i>Gestione layer</i>
 <i>Aggiungi layer Oracle GeoRaster</i>		vedi <i>Oracle Spatial GeoRaster Plugin</i>	<i>Gestione layer</i>
 <i>Aggiungi layer SQL Anywhere</i>		vedi <i>Plugin SQL Anywhere</i>	<i>Gestione layer</i>
 <i>Aggiungi layer WMS/WMTS</i>	Ctrl+Shift+W	vedi <i>Client WMS/WMTS</i>	<i>Gestione layer</i>
 <i>Aggiungi layer WCS</i>		vedi <i>Client WCS</i>	<i>Gestione layer</i>
 <i>Aggiungi vettore WFS</i>		vedi <i>Client WFS e WFS-T</i>	<i>Gestione layer</i>
 <i>Aggiungi layer di testo delimitato</i>		vedere <i>File di testo delimitato</i>	<i>Gestione layer</i>
 <i>Copia elementi</i>		vedi <i>Menu Stile</i>	
 <i>Incolla elementi</i>		vedi <i>Menu Stile</i>	
 <i>Apri tabella attributi</i>		vedi <i>Lavorare con la tabella degli attributi</i>	<i>Attributi</i>
 <i>Modifica</i>		vedi <i>Modifica di un layer esistente</i>	<i>Digitalizzazione</i>
 <i>Salva modifiche vettore</i>		vedi <i>Modifica di un layer esistente</i>	<i>Digitalizzazione</i>
 <i>Modifiche in uso →</i>		vedi <i>Modifica di un layer esistente</i>	<i>Digitalizzazione</i>
<i>Salva come...</i>			
<i>Salva selezione come vettore...</i>		See <i>Lavorare con la tabella degli attributi</i>	
 <i>Elimina layer</i>	Ctrl+D		
 <i>Duplica vettore/i</i>			
<i>Imposta SR del/i layer</i>	Ctrl+Shift+C		
<i>Imposta SR progetto dal layer</i>			
<i>Proprietà</i>			
<i>Interrogazione</i>			
 <i>Etichettatura</i>			
 <i>Aggiungi alla panoramica</i>	Ctrl+Shift+O		<i>Gestione layer</i>
 <i>Aggiungi tutto alla panoramica</i>			
 <i>Rimuovi tutto alla panoramica</i>			
 <i>Mostra tutti i layer</i>	Ctrl+Shift+U		<i>Gestione layer</i>
 <i>Nascondi tutti i layer</i>	Ctrl+Shift+H		<i>Gestione layer</i>

7.1.6 Impostazioni

Voce di Menu	Scorciatoia	Riferimento	Barra degli Strumenti
<i>Pannelli →</i>		vedi <i>Pannelli e Barre degli strumenti</i>	
<i>Barre degli strumenti →</i>		vedi <i>Pannelli e Barre degli strumenti</i>	
<i>Schermo intero</i>	F11		
 <i>Proprietà progetto</i>	Ctrl+Shift+P	vedi <i>Progetti</i>	
 <i>SR personalizzato</i>		vedi <i>Sistemi di riferimento personalizzati</i>	
<i>Gestore di stili</i>		vedi <i>Presentation</i>	
 <i>Configura scorciatoie</i>			
 <i>Personalizzazione</i>		vedi <i>Personalizzazione</i>	
 <i>Opzioni</i>		vedi <i>Opzioni dell'interfaccia grafica (GUI)</i>	
<i>Opzioni di snap</i>			

7.1.7 Plugins

Voce di Menu	Scorciatoia	Riferimento	Barra degli Strumenti
 <i>Gestisci e installa i plugin Console python</i>		vedi <i>La finestra di dialogo Plugins</i>	

Quando si avvia QGIS per la prima volta non tutti i plugin di base verranno caricati.

7.1.8 Vector

Voce di Menu	Scorciatoia	Riferimento	Barra degli Strumenti
<i>Open Street Map</i> →		vedi <i>Caricare vettori OpenStreetMap</i>	
 <i>Strumenti di analisi</i> →		vedi <i>Plugin fTools</i>	
 <i>Strumenti di ricerca</i> →		vedi <i>Plugin fTools</i>	
 <i>Strumenti di Geoprocessing</i> →		vedi <i>Plugin fTools</i>	
<i>Strumenti di Geometria</i> →		vedi <i>Plugin fTools</i>	
 <i>Strumenti di Gestione Dati</i> →		vedi <i>Plugin fTools</i>	

Quando si avvia QGIS per la prima volta non tutti i plugin di base verranno caricati.

7.1.9 Raster

Voce di Menu	Scorciatoia	Riferimento	Barra degli Strumenti
<i>Calcolatore raster ...</i>		vedi <i>Calcolatore raster</i>	

Quando si avvia QGIS per la prima volta non tutti i plugin di base verranno caricati.

7.1.10 Processing

Voce di Menu	Scorciatoia	Riferimento	Barra degli Strumenti
 <i>Strumenti</i>		vedi <i>Strumenti</i>	
 <i>Modellatore grafico</i>		vedi <i>Modellatore grafico</i>	
 <i>Storico e log</i>		vedi <i>Il gestore della cronologia di Processing</i>	
 <i>Opzioni e configurazione</i>		vedi <i>Configurazione dell'ambiente di elaborazione</i>	
 <i>Visualizzatore risultati</i>		vedi <i>Configurazione di applicazioni esterne</i>	
 <i>Linea di comando</i>	Ctrl+Alt+M	vedi <i>The QGIS Commander</i>	

Quando si avvia QGIS per la prima volta non tutti i plugin di base verranno caricati.

7.1.11 Guida

Voce di Menu	Scorciatoia	Riferimento	Barra degli Strumenti
 <i>Contenuti della guida</i>	F1		<i>Guida</i>
 <i>What's This?</i> <i>Documentazione sulle API</i> <i>Serve supporto commerciale?</i>	Shift+F1		<i>Guida</i>
 <i>Home Page di QGIS</i>	Ctrl+H		
 <i>Controlla versione di QGIS</i>			
 <i>Informazioni</i>			
 <i>Sponsor QGIS</i>			

Per Linux  gli elementi della barra dei menu sopra elencati fanno riferimento all'ambiente desktop KDE. Per chi utilizza GNOME invece, il menu *Impostazioni* è leggermente diverso; gli elementi corrispondenti possono essere trovati qui:

 <i>Proprietà progetto</i>	<i>Progetto</i>
 <i>Opzioni</i>	<i>Modifica</i>
 <i>Configura scorciatoie</i>	<i>Modifica</i>
<i>Gestore di stili</i>	<i>Modifica</i>
 <i>SR personalizzato</i>	<i>Modifica</i>
<i>Pannelli →</i>	<i>Visualizza</i>
<i>Barre degli strumenti →</i>	<i>Visualizza</i>
<i>Schermo intero</i>	<i>Visualizza</i>
<i>Scala delle mattonelle</i>	<i>Visualizza</i>
<i>Tracciamento GPS in tempo reale</i>	<i>Visualizza</i>

7.2 Barra degli Strumenti

Le barre degli strumenti forniscono accesso alla maggior parte delle funzioni presenti nei menu, oltre a funzioni aggiuntive volte ad interagire con la mappa. Ogni oggetto della barra degli strumenti ha un aiuto a comparsa (pop-up). Lasciando il cursore del mouse sopra l'icona, verrà visualizzata una breve descrizione della funzione di quello strumento.

Puoi spostare a piacimento ogni barra. Inoltre puoi disattivare ogni barra cliccando con il tasto destro sulla barra degli strumenti e disattivando la voce relativa al menu (vedi *Pannelli e Barre degli strumenti*).

Suggerimento: Ripristinare le barre degli strumenti

Se hai disattivato accidentalmente tutte le barre strumenti, le puoi ripristinare dalla voce di menu *Visualizza → Barre degli strumenti →*. Se una barra degli strumenti scompare in Windows, il che sembra succedere di tanto in tanto, è necessario rimuovere `\HKEY_CURRENT_USER\Software\QGIS\qgis\UI\state` dal registro. Al riavvio di QGIS viene ripristinato lo stato predefinito e le barre degli strumenti saranno nuovamente visibili.

7.3 Legenda

L'area della legenda elenca tutti i layer del progetto. La casella di controllo in ogni layer può essere utilizzata per mostrarlo o nascondere. La barra degli strumenti della legenda ti consente **Aggiungi gruppo**, **Gestisci la visibilità del layer** di tutti i layer o di una combinazione preselezionata, **Filtra leggenda in base al contenuto della mappa**, **Espandi tutti** o **Comprimi tutti** e **Rimuovi layer/gruppo**. Il pulsante  ti consente di aggiungere

vista **Predefinita** nella leggenda. Ciò significa che puoi scegliere di visualizzare alcuni layer con categorizzazione specifica e aggiungere questa vista alla lista **Predefinita** lista. Per aggiungere una vista preimpostata basta che fai click su **ImActionShowPresets**, scegli *** Aggiungi vista ...*** dal menu a tendina e dai un nome alla vista predefinita. Dopo di che vedrai un elenco con tutte le viste che si possono richiamare premendo sul pulsante **ImActionShowPresets**.

Tutte le viste predefinite aggiunte sono presenti anche nel compositore di stampe in modo di permetterti di creare stampe impostate sulle suddette viste (vedi: ref: *composer_main_properties*).

Puoi selezionare un layer e trascinarlo in modo da modificarne la visibilità (Z-ordering). Z-ordering significa che i layer in cima alla leggenda coprono nella mappa quelli sottostanti.

Nota: Puoi disabilitare questo comportamento dal pannello 'Ordine dei layer'.

Puoi organizzare in gruppi i layer presenti nella leggenda. Ci sono due modi per farlo:

1. Premi l'icona  per aggiungere un nuovo gruppo. Digita un nome per il gruppo e premi **Enter**. Ora clicca su un layer esistente e trascinalo nel gruppo.
2. Seleziona alcuni layer, clicca con il tasto destro e scegli *Gruppo selezionato*. I layer selezionati saranno automaticamente spostati nel nuovo gruppo.

Per rimuovere un layer da un gruppo puoi selezionare e trascinare il layer al di fuori dello stesso o in alternativa puoi cliccare con il tasto destro del mouse sul layer e selezionare *Muovi fuori dal gruppo* →. I gruppi possono essere nidificati all'interno di altri gruppi.

Puoi usare la casella di controllo di un gruppo per mostrare/nascondere tutti i layer del gruppo con un singolo click.

Le voci del menu contestuale (click con il tasto destro sul layer) sono diverse a seconda se il file è un raster un vettore. Per i vettori di GRASS la voce di menu  *Modifica* non è abilitata. Vedi la sezione [Digitalizzare e modificare layer vettoriali GRASS](#) per informazioni su come modificare i vettori di GRASS.

Menu contestuale per i raster

- *Zoom all'estensione del layer* →
- *Aggiungi alla panoramica* →
- *Zoom alla scala migliore (100%)*
- *Stira usando l'estensione attuale*
- *Rimuovi* →
- *Duplica*
- *Imposta la scala di visibilità del layer*
- *Imposta il SR del layer* →
- *Imposta SR progetto dal layer*
- *Salva come ...*
- *Salva come file di definizione del layer*
- *Proprietà*
- *Rinomina* →
- *Copia stile*

Inoltre, a seconda della posizione e della selezione del layer

- *Crea un elemento al livello più alto*
- *Gruppo selezionato*

Menu contestuale per i vettori

- *Zoom all'estensione del layer*
- *Aggiungi alla panoramica*
- *Rimuovi* →
- *Duplica*
- *Imposta la scala di visibilità del layer*
- *Imposta il SR del layer* →
- *Imposta SR progetto dal layer*
- *Apri tabella attributi*
- *Modifica* (non disponibile per i layer GRASS)
- *Salva come ...*
- *Salva come file di definizione del layer*
- *Filtro*
- *Mostra totale elementi*
- *Proprietà*
- *Rinomina* →
- *Copia stile*

Inoltre, a seconda della posizione e della selezione del layer

- *Crea un elemento al livello più alto*
- *Gruppo selezionato*

Menu contestuale per i gruppi di layer

- *Zoom al gruppo*
- *Rimuovi* →
- *Imposta il SR del gruppo*
- *Rinomina* →
- *Aggiungi gruppo*

Puoi selezionare più di un layer o di un gruppo allo stesso tempo tenendo premuto il tasto `Ctrl` e cliccando il tasto sinistro del mouse sui vari layer. Potrai così spostare contemporaneamente tutti i layer selezionati in un nuovo gruppo.

Puoi anche rimuovere più layer o gruppi contemporaneamente selezionando i diversi layer tenendo premuto il tasto `Ctrl` e premendo poi `Ctrl+D`. In questo modo tutto ciò che hai selezionato verrà rimosso dalla lista dei layer.

7.3.1 Lavorare con la legenda indipendentemente dall'ordine dei layer

Hai anche la possibilità di usare un pannello per impostare una legenda indipendentemente dall'ordine di visualizzazione. Puoi attivare questa opzione dal menu *Visualizza* → *Pannelli* → *Ordine dei layer* e determinare così l'ordine di visualizzazione dei layer nella mappa. Così facendo puoi ordinare i layer in base ad un determinato criterio, pur mantenendo l'ordine di visualizzazione iniziale corretto (vedi [figure_layer_order](#)). Spuntando la casella di controllo *Controllo ordine di visualizzazione* collocata sotto l'elenco dei layer, puoi ritornare al comportamento iniziale predefinito.

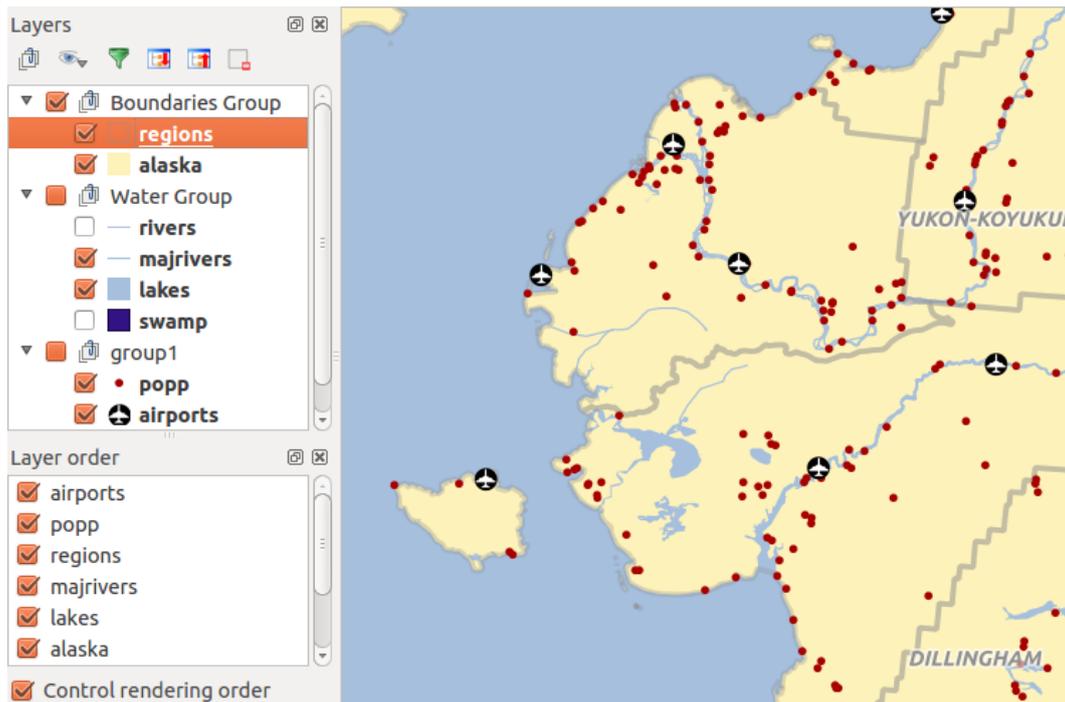


Figura 7.2: Lavora con una legenda indipendente dall'ordine dei layer 🐧

7.4 Mappa

Questa è l'area in cui le mappe vengono visualizzate. La mappa visualizzata in questa finestra è il risultato dei vettori e/o raster che hai scelto di caricare (vedi le sezioni che seguono per ulteriori informazioni su come caricare i layer). Puoi modificare la zona di visualizzazione della mappa (spostando la messa a fuoco dell'esposizione della mappa ad un'altra regione) puoi effettuare operazioni di zoom in ed out. Varie altre operazioni sono descritte nella sezione relativa alla barra dei menu. La vista nell'area di mappa e la legenda sono strettamente legate l'una all'altra: le mappe che vengono visualizzate riflettono i cambiamenti fatti nella area della legenda.

Suggerimento: ZOOM IN E ZOOM OUT CON LA ROTELLA DEL MOUSE

Per le operazioni di zoom puoi anche utilizzare la rotella del mouse. Posizionando il puntatore del mouse nell'area di visualizzazione delle mappe aumenterai lo zoom girando la rotella verso lo schermo, lo ridurrai girandola nel verso contrario. La posizione del puntatore costituisce il centro per l'ingrandimento. Puoi regolare il comportamento della funzione di zoom con la rotella del mouse nella scheda *Strumenti mappa* del menu *Impostazioni* → *Opzioni*.

Suggerimento: MUOVERE LA MAPPA CON I TASTI FRECCIA E LA BARRA SPAZIATRICE

Puoi spostare la mappa anche con le frecce della tastiera. Posiziona il mouse sulla mappa e clicca la freccia destra per spostarti verso est, la freccia sinistra per spostarti verso ovest, la freccia in su per spostarti verso nord e la freccia in giù per spostarti verso sud. Puoi anche spostare la mappa con la barra spaziatrice oppure premendo la rotellina del mouse: nel primo caso tieni premuta la barra spaziatrice e muovi il mouse, mentre nel secondo caso tieni premuto il tasto della rotellina mentre muovi il mouse.

7.5 Barra di Stato

La barra di stato mostra le coordinate di mappa (es. metri o gradi decimali) della posizione del mouse. Nella barra di stato a sinistra delle coordinate è presente un piccolo pulsante che ti consente di passare dalla visualizzazione delle coordinate del movimento del mouse alla visualizzazione delle coordinate dell'estensione della mappa.

Accanto alla finestra delle coordinate c'è la finestra della scala. Questa mostra la scala della mappa attuale. Sia ingrandendo che rimpicciolendo la scala, QGIS mostrerà la scala corrente. Un menu a tendina ti permette di scegliere fra alcune scale predefinite da 1:500 a 1:1000000.

La barra di avanzamento nella barra di stato mostra il progredire della visualizzazione di ogni layer nella mappa. In alcuni casi, come quando vengono raccolte informazioni statistiche su raster, questa barra è usata per mostrare lo stato di avanzamento di questi processi, in genere molto lunghi.

Se è disponibile un nuovo plugin o un aggiornamento di un plugin installato, apparirà un avviso nella barra di stato. Nella parte destra della barra di stato è presente una casella di controllo che, se attivata, può essere usata per impedire temporaneamente la visualizzazione dei layer. L'icona  disattiva temporaneamente la visualizzazione dei layer nella mappa (vedi sezione [Visualizzazione](#)).

All'estrema destra della barra di stato è visibile il codice EPSG del SR del progetto corrente; cliccando sull'icona a forma di proiettore subito a destra del codice EPSG è possibile accedere alle proprietà del SR.

Suggerimento: Calcolare la scala corretta della mappa

I gradi sono l'unità di misura predefinita di QGIS, quindi le coordinate di ciascun layer verranno visualizzate in gradi decimali. Per ottenere valori di scala corretti, puoi cambiarli manualmente in unità metriche nella scheda *Generale* del menu *Progetto* → *Proprietà progetto* oppure puoi scegliere un sistema di riferimento (SR) cliccando sull'icona  Stato SR nell'angolo in basso a destra della barra di stato. In quest'ultimo caso le unità di misura sono impostate secondo le caratteristiche delle proprietà del progetto, ad esempio '+units=m'.

Strumenti generali

8.1 Scorciatoie da tastiera

Ci sono molte scorciatoie da tastiera per molte funzionalità di QGIS. Le scorciatoie sono elencate nella sezione *Barra dei Menu*. Inoltre, l'opzione di menu *Impostazioni* → *Configura scorciatoie* → ti permette di modificare le scorciatoie da tastiera esistenti e di aggiungerne di nuove.

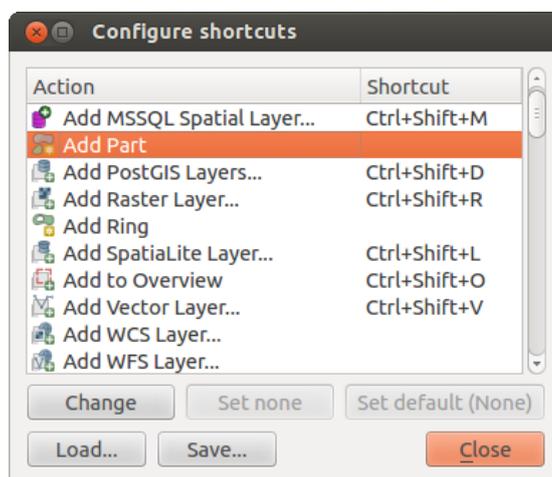


Figura 8.1: Finestra scorciatoie 🐧 (Gnome)

Configurare le scorciatoie è molto semplice; basta selezionare una funzionalità di interesse e cliccare [**Cambia**], [**Elimina**] o [**Selezione predefinita**]. Le configurazioni possono essere salvate in un file XML e caricate su un'altra installazione di QGIS.

8.2 Guide contestuali

Se vuoi ottenere maggiori informazioni su una funzionalità specifica puoi usare il pulsante [**Aiuto**] disponibile in molte finestre di dialogo. Nota che nel caso di plugin di terze parti la guida contestuale potrebbe rimandare ad una pagina web dedicata.

8.3 Visualizzazione

In modo predefinito, QGIS visualizza tutti i layer visibili ogni volta che la mappa viene aggiornata. La mappa viene aggiornata ogni volta che:

- Aggiungi un layer
- Sposti, ingrandisci o riduci la mappa
- Ridimensioni la finestra di QGIS
- Cambi la visibilità di uno o più layer

QGIS consente di controllare il processo di visualizzazione in diverse maniere.

8.3.1 Visualizzazione in funzione della scala

La visualizzazione in funzione della scala permette di specificare la scala minima e massima alla quale il vettore verrà visualizzato. Per impostare questa funzionalità, apri la finestra *Proprietà* facendo doppio click sul vettore.

Nella scheda *Generale*, clicca sulla casella di controllo *Visualizzazione dipendente dalla scala*: potrai così inserire i valori minimi e massimi di visualizzazione della scala.

I valori di scala possono essere determinati usando prima lo zoom sul layer per il quale si vuole impostare l'opzione e prendendo successivamente nota del valore di scala visualizzato nella barra di stato di QGIS.

8.3.2 Controllare la visualizzazione della mappa

Puoi controllare la visualizzazione della mappa in molti modi diversi, come descritto di seguito.

Sospensione della visualizzazione

Per interrompere la visualizzazione, clicca sulla casella di controllo *Visualizza* in basso a destra della barra di stato. Quando la casella *Visualizza* non è spuntata, QGIS non aggiorna la vista quando si verifica uno degli eventi precedentemente descritti nella sezione *Visualizzazione*. Alcuni casi in cui potresti voler sospendere la visualizzazione sono:

- Aggiunta di molti layer con simbologia predefinita prima della visualizzazione
- Aggiunta di uno o più layer di grosse dimensioni e impostazione di una scala prima della visualizzazione
- Aggiunta di uno o più layer di grossa dimensione e zoom ad un'area specifica prima della visualizzazione
- Combinazioni delle precedenti

Se la casella di controllo *Aggiorna* è spuntata, la visualizzazione e l'aggiornamento della mappa saranno immediati.

Controllare la visibilità dei layer quando sono caricati

Puoi scegliere l'opzione di caricare i nuovi layer senza che questi vengano immediatamente visualizzati sulla mappa. Ciò significa che quando aggiungerai un layer al progetto, la casella di controllo per la visibilità nella legenda risulterà disabilitata. Per impostare questa opzione, apri il menu *Impostazioni* → *Opzioni* → e clicca sulla scheda *Visualizzazione*. Deseleziona la casella di controllo *Per impostazione predefinita i nuovi layer aggiunti alla mappa vengono visualizzati subito*. Ogni layer aggiunto alla mappa risulterà essere quindi spento (invisibile).

Fermare la visualizzazione

Per fermare la visualizzazione della mappa premi il tasto **ESC**. In questo modo l'aggiornamento della mappa verrà bloccato e la mappa rimarrà parzialmente disegnata. Dopo aver premuto il tasto **ESC** potrebbe passare un po' di tempo finché l'interruzione della visualizzazione della mappa sia effettiva.

Nota: Attualmente non si può interrompere la visualizzazione in corso: questa opzione è stata disabilitata nella porta Qt4 a causa di diversi problemi dell'interfaccia utente (UI).

Aggiornamento della mappa durante la visualizzazione

Puoi impostare un'opzione per aggiornare la mappa man mano che gli elementi del layer vengono letti. In modo predefinito, QGIS non visualizza alcun elemento a video finché l'intero layer non è stato caricato. Per aggiornare la visualizzazione man mano che gli elementi vengono caricati, seleziona il menu *Impostazioni* → *Opzioni* → e clicca sulla scheda *Visualizzazione*. Imposta il numero di elementi che vuoi che vengano caricati prima che la mappa venga aggiornata. Il valore pari a 0 disabilita l'aggiornamento durante la visualizzazione degli oggetti (impostazione predefinita). Un valore troppo basso diminuisce le prestazioni in quanto la mappa viene continuamente aggiornata man mano che gli elementi del layer vengono caricati. 500 è il valore suggerito.

Modificare la qualità della visualizzazione

Ci sono due opzioni per modificare la qualità della visualizzazione. Dal menu *Impostazioni* → *Opzioni* → clicca sulla scheda *Visualizzazione* e seleziona o deseleziona le seguenti caselle di controllo.

- *Rendi le linee meno irregolari a spese delle prestazioni*
- *Risolvi problemi con i poligoni riempiti non correttamente*

Velocizzare la visualizzazione

Ci sono due opzioni per modificare la qualità della visualizzazione. Dal menu *Impostazione* → *Opzioni* → clicca sulla scheda *Visualizzazione* e seleziona o deseleziona le seguenti caselle di controllo.

- *Abilita buffer geometrie*. Questa opzione fornisce prestazioni grafiche migliori a scapito della possibilità di annullare un ridisegno sullo schermo o di visualizzare i vettori in maniera incrementale. Se disabilitata, puoi impostare il *Numero di geometrie da disegnare prima di aggiornare lo schermo*, altrimenti questa opzione risulterà essere inattiva.
- *Usa il caching del disegno quando possibile per velocizzare la visualizzazione*

8.4 Misurazioni

Puoi effettuare misurazioni metriche solo con i sistemi di coordinate piane (es. UTM). Se la mappa caricata è definita in un sistema di coordinate geografiche (latitudine/longitudine), il risultato della misurazione di linee o aree sarà errato. Per effettuare misurazioni devi quindi impostare correttamente il sistema di coordinate della mappa (vedi sezione *Lavorare con le proiezioni*). Tutti i moduli di misura usano le opzioni di snapping del modulo di digitalizzazione. Questo è utile se vuoi misurare lungo linee o aree di vettori.

Per selezionare uno strumento di misura clicca su  e scegli lo strumento che vuoi usare.

8.4.1 Misurare lunghezze, aree ed angoli

 **Misura linea**: QGIS è in grado di fornire la misura della distanza reale tra due punti in funzione di uno specifico ellissoide. Puoi configurare queste opzioni dal menu *Impostazioni* → *Opzioni* → cliccando sulla scheda *Strumenti mappa* e scegliere l'ellissoide appropriato. Qui puoi scegliere il colore dell'elastico, l'unità di misura preferita della distanza (metri o piedi) e degli angoli (gradi, radianti o gradi decimali). Con questo strumento potrai cliccare diversi punti sulla mappa: la misura di ogni segmento verrà mostrata nella finestra dello strumento insieme alla misura totale. Per terminare la misura clicca con il tasto destro del mouse.

 **Misura area** Questo strumento consente di misurare le aree; la finestra mostrerà unicamente l'area totale misurata. In più, se hai impostato una tolleranza di snap per il vettore (sezione *Settare la tolleranza dello snapping e il raggio*

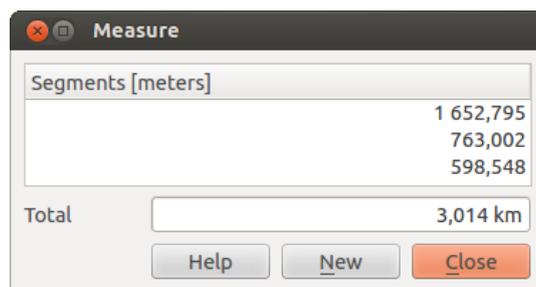


Figura 8.2: Misura distanze 🐧 (Gnome)

di ricerca degli elementi), lo strumento farà lo snap sul vettore selezionato. Quindi, se vuoi misurare esattamente lungo una linea o un poligono devi prima definire la tolleranza di snap e poi selezionare il vettore. In questo modo, quando vengono usati gli strumenti di misura, ogni click del mouse (all'interno della tolleranza definita) si aggancerà a quel preciso vettore.

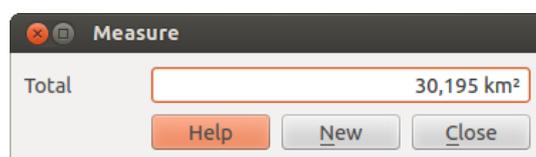


Figura 8.3: Misura area 🐧 (Gnome)

 **Misura angoli**: grazie a questo strumento puoi misurare gli angoli. Clicca e disegna il primo segmento dell'angolo, poi sposta il mouse per disegnare l'angolo stesso; la misura apparirà in una finestra pop-up.

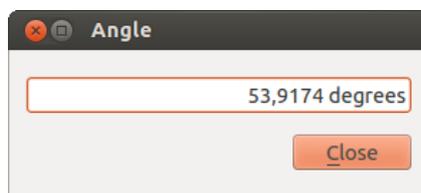


Figura 8.4: Misura angolo 🐧 (Gnome)

8.4.2 Selezionare e deselegionare elementi

QGIS fornisce diversi strumenti per la selezione di elementi nella mappa. Per selezionare uno o più elementi clicca semplicemente su  e scegli lo strumento che preferisci:

-  Seleziona il singolo elemento
-  Seleziona elementi con un rettangolo
-  Seleziona elementi con un poligono
-  Seleziona elementi a mano libera
-  Seleziona elementi con un cerchio

Per deselegionare tutti gli elementi selezionati clicca su  Deselegiona gli elementi da tutti i vettori.

 **Seleziona la geometria con una espressione** permette di selezionare le geometrie tramite la finestra di dialogo delle espressioni. Vedere il capitolo *Expressions* per alcuni esempi..

Gli utenti possono salvare le geometrie selezionate in un **Nuovo vettore in memoria** oppure in un **Nuovo vettore** dal menu *Modifica* → *Incolla geometrie come...* e selezionare la modalità desiderata.

8.5 Informazione elementi

Con lo strumento Informazioni elementi puoi interagire con gli elementi nella mappa ed ottenere gli attributi delle geometrie. Attiva lo strumento cliccando su *Visualizza* → *Informazione elementi*, oppure usa la scorciatoia da tastiera `Ctrl + Shift + I` oppure ancora clicca sull'icona  *Informazione elementi* nella barra degli strumenti.

Se clicchi su più elementi, la finestra *Informazione elementi* elencherà tutti gli attributi di tutte le geometrie. Il primo elemento nella lista farà riferimento all'ordine dell'elemento cliccato, seguito dal nome del vettore. Gli altri elementi della lista faranno riferimento ai vari campi con i rispettivi valori. Infine, vengono fornite anche le informazioni sulla geometria.

Puoi personalizzare questa finestra in modo da visualizzare determinati campi, ma in modo predefinito vengono mostrati tre tipi di informazione:

- **Azioni:** puoi aggiungere le azioni nella lista degli elementi. Devi solamente cliccare sull'icona azione per eseguire l'azione. In modo predefinito è presente solo l'azione che apre il modulo degli attributi per la successiva modifica.
- **Derivato:** queste informazioni vengono calcolate o derivano da altre informazioni. Verranno visualizzate le coordinate X e Y cliccate, l'area e il perimetro in unità di mappa per i poligoni, la lunghezza per le linee e gli ID delle geometrie.
- **Attributi dei dati:** questa è l'elenco dei campi e relativi attributi della geometria cliccata.

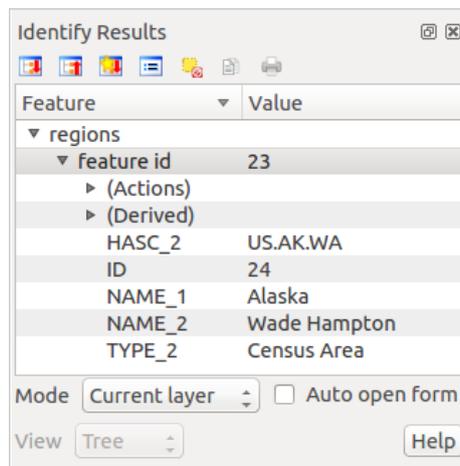


Figura 8.5: Finestra di dialogo Informazioni elementi  (Gnome)

In fondo alla finestra ci sono cinque icone:

-  Espandi albero
-  Racchiudi albero
-  Impostazione predefinita
-  Copia attributi
-  Stampa il responso HTML selezionato

Puoi trovare altre opzioni nel menu contestuale dell'elemento identificato. Per esempio, dal menu contestuale puoi:

- Visualizzare modulo geometria

- Zoomare alla geometria
- Copiare elementi: copiare tutti gli elementi e gli attributi della geometria
- Imposta/ rimuovi la selezione della geometria: aggiungi alla selezione la geometria identificata
- Copiare un valore di un attributo: copiare solo il valore dell'attributo identificato
- Copiare attributi geometria: copiare solamente gli attributi;
- Cancellare risultati: verranno cancellati i risultati nella finestra
- Cancellare evidenziati: verranno cancellate le geometrie evidenziate sulla mappa
- Evidenziare tutto
- Evidenziare vettore
- Attivare un vettore: scegliere un vettore che deve essere attivato
- Proprietà del vettore: aprire la finestra delle proprietà del vettore
- Espandi tutto
- Racchiudi tutto

8.6 Decorazioni

Le decorazioni di QGIS includono: il reticolo, l'etichetta copyright, la freccia nord e la barra di scala. Sono usate per 'decorare' la mappa aggiungendo elementi cartografici.

8.6.1 Reticolo



Reticolo ti permette di aggiungere un reticolo e le coordinate alla mappa.

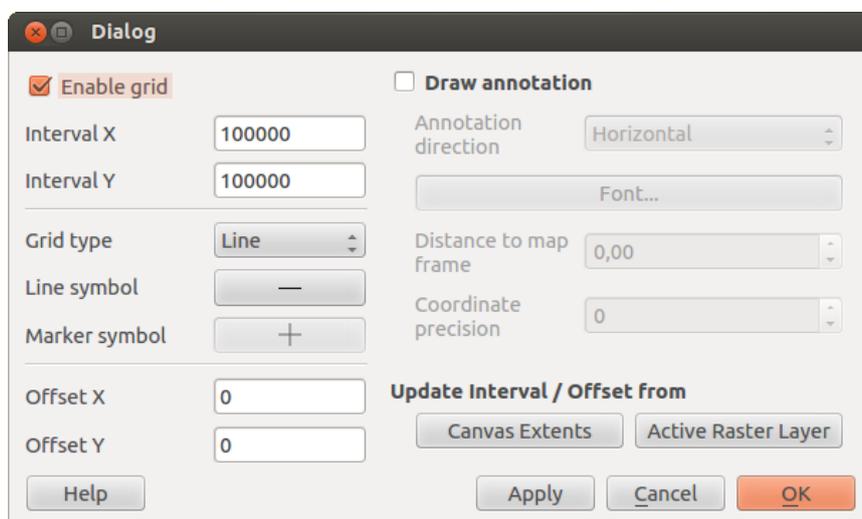


Figura 8.6: Finestra di dialogo delle proprietà del reticolo 

1. Seleziona dal menu *Visualizza* → *Decorazioni* → *Reticolo*. Si aprirà un'altra finestra (vedi [figure_decorations_1](#)).
2. Attiva la casella di controllo  *Abilita reticolo* e imposta i valori che preferisci in funzione dei layer caricati sulla mappa.

3. Attiva la casella di controllo  *Scrivi coordinate* e imposta le proprietà migliori in base agli elementi che hai caricato sulla mappa.
4. Clicca su [Applica] per verificare che il risultato sia corretto.
5. Clicca [OK] per chiudere la finestra di dialogo.

8.6.2 Etichetta Copyright

 *Etichetta copyright* aggiunge un'Etichetta Copyright personalizzata in base al testo che preferisci far apparire sulla mappa.

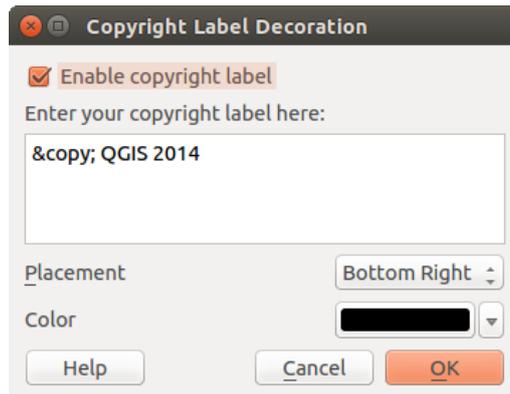


Figura 8.7: La finestra di dialogo Copyright 

1. Seleziona dal menu *Visualizza* → *Decorazioni* → *Etichetta copyright*. Si aprirà un'altra finestra (vedi [figure_decorations_2](#)).
2. Digita il testo che vuoi aggiungere alla mappa. Puoi anche usare il linguaggio HTML come mostrato nell'esempio.
3. Scegli la posizione dell'etichetta dal menu a tendina *Posizione* .
4. Assicurati che la casella di controllo  *Abilita etichetta di copyright* sia spuntata.
5. Clicca [OK].

Nell'esempio di sopra, QGIS inserisce il simbolo di copyright, seguito dalla data, nell'angolo in basso a destra della mappa.

8.6.3 Freccia Nord

 *Freccia nord* aggiunge alla mappa una semplice freccia indicante il nord. Attualmente c'è un solo stile disponibile. Puoi modificare manualmente l'angolo della freccia o lasciare che QGIS imposti automaticamente la direzione. Per il posizionamento della freccia hai quattro possibilità, corrispondenti ai quattro angoli della mappa.

8.6.4 Barra di Scala

 *Barra di scala* aggiunge una semplice barra di scala alla mappa. Puoi controllare il posizionamento, lo stile, il colore e le dimensioni della barra.

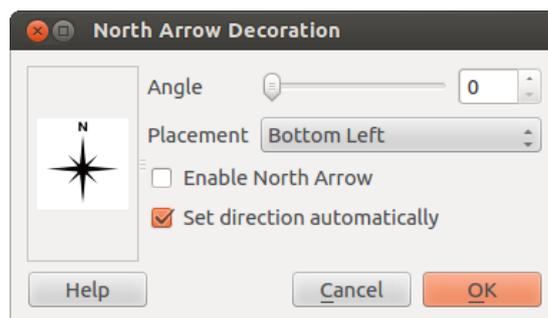


Figura 8.8: La finestra Freccia Nord 🐧

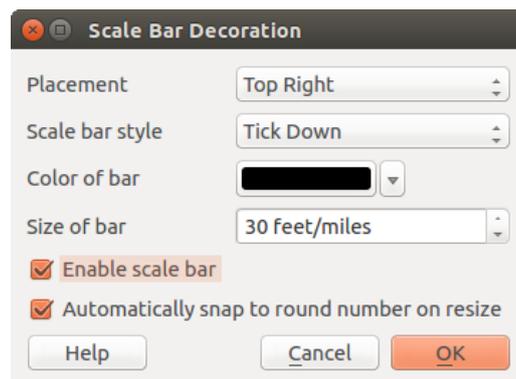


Figura 8.9: La finestra Barra di Scala 🐧

QGIS supporta solamente la visualizzazione della scala nella stessa unità di misura della mappa. Se l'unità di misura dei layer è il metro, non potrai quindi creare una barra di scala in piedi. Allo stesso modo, se usi i gradi decimali, non potrai creare una barra di scala che mostri le distanze in metri.

Per aggiungere una barra di scala:

1. Seleziona dal menu *Visualizzazione* → *Decorazioni* → *Barra di scala*. Si aprirà così una finestra di dialogo (see [figure_decorations_4](#))
2. Scegli la posizione dell'etichetta dal menu a tendina *Posizione* .
3. Scegli lo stile dal menu a tendina *Stile della Barra di Scala* .
4. Scegli il colore della barra di scala dal menu *Colore della barra*  o usa il colore nero predefinito.
5. Imposta la dimensione della barra e la sua etichetta *Dimensione della barra* .
6. Assicurati che la casella di controllo  *Abilitare barra di scala* sia spuntata.
7. Se vuoi, spunta anche la casella di controllo  *Arrotonda automaticamente il numero durante il ridimensionamento*.
8. Clicca [OK].

Suggerimento: Impostazioni delle decorazioni

Quando salvi un progetto .qgs, ogni impostazione relativa alle decorazioni viene salvata nel file e ripristinata alla successiva apertura del progetto.

8.7 Note testuali

Lo strumento  *Nota testuale* nella barra degli attributi permette di posizionare del testo formattato sulla mappa. Per creare una nota, seleziona lo strumento *Nota testuale* e clicca sulla mappa.

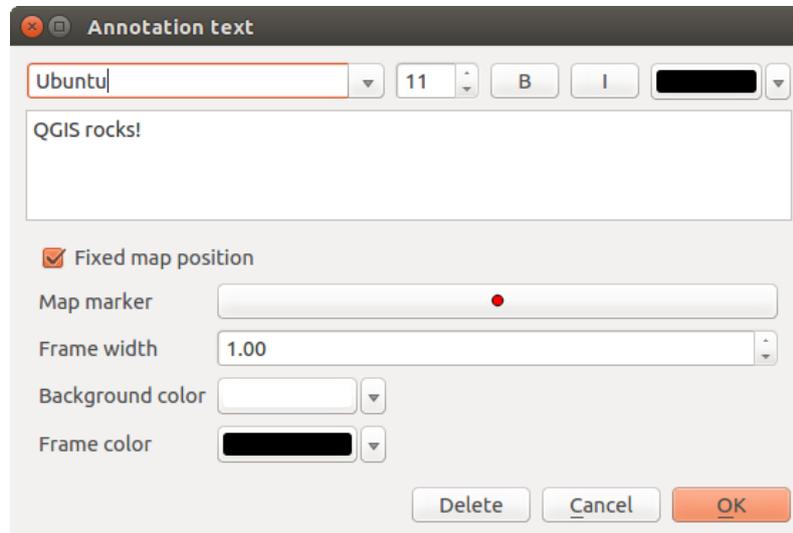


Figura 8.10: La finestra di annotazione testuale 

Se fai doppio click sull'elemento aggiunto alla mappa si aprirà una finestra di dialogo con diverse opzioni. Avrai accesso a un editor per aggiungere il testo della nota. Inoltre hai anche la possibilità di scegliere se la nota dovrà essere posizionata su un punto preciso della mappa (visualizzata come indicatore) oppure se la posizione della nota dovrà essere relativa a una posizione dello schermo (quindi indipendente dalla mappa). Puoi muovere sia tutta la nota (trascinando l'indicatore) sia solamente il testo (trascinando il riquadro del testo).

Con lo strumento  *Muovi nota* puoi spostare la nota dove preferisci.

8.7.1 Note Html

Lo strumento  *Nota Html* presente nella barra degli attributi, ti permette di creare una nota con del testo formattato in linguaggio html. Seleziona lo strumento *Nota Html*, clicca sulla mappa e inserisci il testo html nel riquadro della nota.

8.7.2 Note SVG

Lo strumento  *Nota SVG* ti permette di aggiungere un simbolo SVG nel riquadro della nota e piazzarlo sulla mappa. Per creare una nota SVG, seleziona lo strumento *Nota SVG*, clicca sulla mappa e aggiungi il percorso al file SVG nella finestra di dialogo.

8.7.3 Modulo annotazioni

Puoi anche creare moduli di note personalizzati. Usa lo strumento  *Nota con modulo* per visualizzare gli attributi di un vettore con un modulo personalizzato creato con Qt Designer (vedi [figure_custom_annotation](#)). La nota con modulo è simile al modulo dello strumento *Informazioni elementi*, con la differenza che le informazioni sono visualizzate attraverso una nota. Vedi il video <https://www.youtube.com/watch?v=0pDBuSbQ02o> di Tim Sutton per ulteriori informazioni.

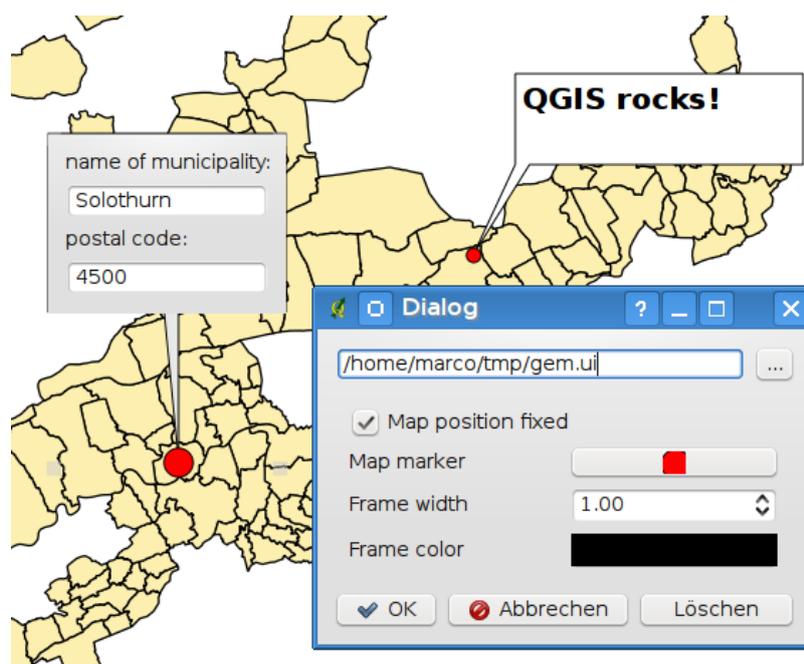


Figura 8.11: Modulo personalizzato qt designer di annotazioni 🐧

Nota: Nota: Premendo `Ctrl+T` con uno strumento nota attivo (Nota testuale, Nota con modulo, Muovi nota) lo stato di visualizzazione delle note si inverte: se sono visibili diventano invisibili e viceversa.

8.8 Segnalibri geospaziali

I segnalibri geospaziali consentono di memorizzare una posizione geografica alla quale potrai ritornare in un secondo momento.

8.8.1 Creazione di un segnalibro

Per creare un segnalibro:

1. Usa lo zoom o muovi la mappa all'estensione d'interesse.
2. Seleziona l'opzione *Visualizza* → *Nuovo segnalibro* → oppure premi `Ctrl-B`.
3. Inserisci un nome descrittivo per il segnalibro (fino a 255 caratteri).
4. Premi `Invio` per aggiungere il segnalibro o `[Close]` per uscire senza aggiungerlo.

Nota che puoi avere più di un segnalibro con lo stesso nome.

8.8.2 Uso e gestione dei segnalibri

Per usare o gestire i segnalibri, seleziona l'opzione *Visualizza* → *Mostra segnalibri* →. La finestra di dialogo *Segnalibri geospaziali* ti consente zoomare a un segnalibro o di eliminarlo. Non puoi modificare né il nome né le coordinate di un segnalibro.

8.8.3 Zoom a un segnalibro

Dalla finestra *Segnalibri geospaziali*, seleziona il segnalibro desiderato cliccandoci sopra, quindi clicca su **[Zoom A]**. Puoi usare lo zoom su un segnalibro anche facendoci doppio click.

8.8.4 Cancellare un segnalibro

Per cancellare un segnalibro, clicca sulla finestra di dialogo *Segnalibri geospaziali* **[Elimina]**. Conferma la scelta cliccando su **[OK]** o annulla cliccando su **[Close]**.

8.9 Progetti nidificati

Se vuoi nidificare dei layer di altri progetti nel tuo progetto attuale, seleziona *Layer → Includi layer e gruppi...*

8.9.1 Layers inclusi

La finestra di dialogo ti permette di scegliere quali layer di altri progetti puoi includere. Di seguito un piccolo esempio:

1. Scegliere  per cercare un altro progetto del dataset Alaska.
2. Seleziona il progetto `grassland`. Puoi vedere il contenuto del progetto (vedi [figure_embed_dialog](#)).
3. Premi `Ctrl`, clicca sui layer `grassland` e `regions` e poi premi su **[OK]**. I layer selezionati verranno aggiunti sia nella legenda che sulla mappa.

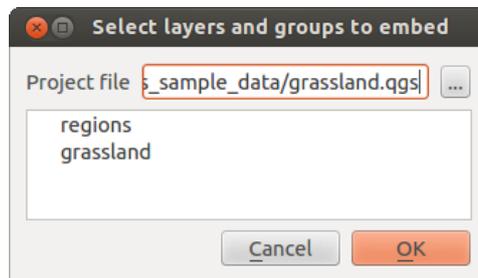


Figura 8.12: Selezione dei layer e dei gruppi da inserire 

Anche se i layer nidificati sono modificabili, non puoi modificarne le proprietà, come stile ed etichette.

8.9.2 Rimuovi i layer nidificati

Cliccare con il tasto destro sul layer aggiunto e scegliere  **Rimuovi**.

Configurazione di QGIS

QGIS è altamente configurabile grazie al menu *Impostazioni* →. Altre opzioni di configurazione sono possibili attraverso i menu Pannelli e Barre degli strumenti presenti nel menu *Visualizza* →.

Nota: QGIS segue le linee guida del desktop per la posizione di opzioni e proprietà del progetto. Quindi in relazione al sistema operativo che si sta utilizzando, la posizione di alcuni degli elementi di cui sopra può essere situata nel **:MenùSelection:‘Visualizza‘** (pannelli e barre degli strumenti) o in: MenuSelection: *Progetto* per Opzioni.

9.1 Pannelli e Barre degli strumenti

Nel menu *Pannelli* → è possibile attivare e disattivare i widgets di QGIS. Il menu *Barre degli strumenti* → offre la possibilità di attivare e disattivare gruppi di icone nella barra degli strumenti di QGIS (vedi figure_panels_toolbars).

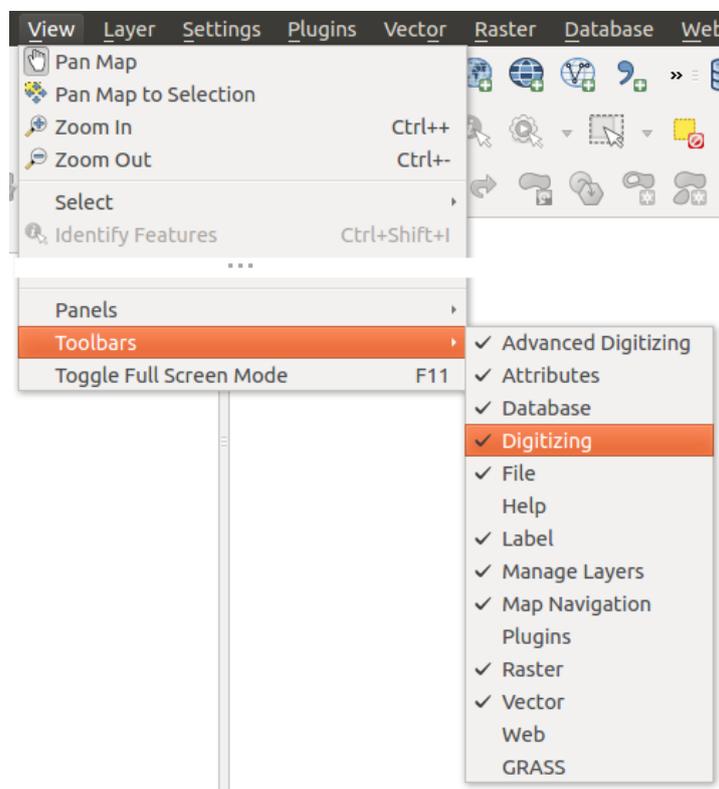


Figura 9.1: Pannelli e Barre degli strumenti 

Suggerimento: Panoramica

Attraverso il pannello panoramica hai la possibilità di visualizzare tutti i layer aggiunti. Puoi attivare il pannello dal menu *Visualizza* → *Pannelli*. All'interno della panoramica è presente un rettangolo che mostra l'estensione attuale della mappa. Questo ti permette di determinare rapidamente quale area della mappa stai attualmente osservando. Le etichette dei layer non sono visualizzate nella panoramica anche se la funzione di visualizzazione delle etichette è attiva. Cliccando e trascinando il rettangolo rosso che mostra l'estensione corrente della tua vista, la mappa si sposterà di conseguenza.

Suggerimento: Messaggi di log

Puoi facilmente tenere traccia dei messaggi di QGIS. Attiva la casella di controllo  *Messaggi di log* nel menu  *Visualizza* → *Pannelli* e leggi i messaggi relativi alle varie operazioni nelle diverse schede.

9.2 Proprietà progetto

Nella finestra proprietà del progetto in :MenuSelection: *Impostazioni* -> *Proprietà progetto...* (kde) o  :MenuSelection: 'Progetto -> Proprietà progetto...' (Gnome), puoi impostare specifiche opzioni del progetto. Queste includono:

- Nella scheda *Generale* puoi impostare: il titolo del progetto, il colore della selezione e dello sfondo, l'unità di misura della mappa, la precisione e l'opzione di salvare il percorso ai layer come relativo o assoluto. Se hai attivato la trasformazione del SR puoi specificare un ellissoide per il calcolo delle distanze. Puoi scegliere le unità della mappa (solo se la trasformazione del SR è disattivata) e la precisione del numero (cifre decimali). Infine puoi anche inserire una lista di scale che sovrascriveranno le scale preimpostate.
- La scheda *SR* permette di scegliere il sistema di riferimento e di abilitare la riproiezione al volo di raster e vettori quando questi hanno un SR diverso.
- La terza scheda *Informazioni vettore* permette di attivare/disattivare i vettori che possono essere interrogati (vedi anche il paragrafo relativo agli strumenti mappa nella sezione *Opzioni dell'interfaccia grafica (GUI)* per maggiori informazioni sull'interrogabilità di vettori multipli).
- La scheda *Stili predefiniti* ti consente di controllare gli stili dei nuovi layer se non hanno già uno stile definito .qml. Puoi impostare il livello di trasparenza predefinito per nuovi layer e se assegnare colori casuali ai simboli. C'è anche una sezione aggiuntiva in cui puoi definire colori specifici per il progetto in corso. Potrai trovare i colori aggiunti nella lista a cascata della finestra di dialogo dei colori presente in ogni visualizzazione.
- La scheda *Server OWS* permette di impostare le informazioni dei server WMS e WFS, l'estensione e le restrizioni dei SR.
- La scheda *Macro* serve per modificare le macro di Python per il progetto. Attualmente sono disponibili tre macro: `openProject()`, `saveProject()` e `closeProject()`.
- La scheda *Relazioni* permette di impostare relazioni 1:n. Le relazioni sono definite nella finestra di dialogo delle proprietà del progetto. Una volta impostate le relazioni per un vettore, la vista modulo (cioè quando si interroga il vettore con lo strumento informazioni elemento) elencherà le relazioni in una finestra dedicata. Le relazioni 1:n sono uno strumento molto prezioso che può essere utilizzato, per esempio, per tenere traccia delle ispezioni storiche di un tratto stradale o di manutenzione delle tubature. Per maggiori informazioni sulle relazioni 1:n dai un'occhiata alla sezione *Creating one to many relations*.

9.3 Opzioni dell'interfaccia grafica (GUI)

 Alcune opzioni di base per QGIS possono essere impostate nella finestra *Opzioni*. Seleziona la voce *Impostazioni* →  *Opzioni*. Le schede dedicate alla personalizzazione sono descritte di seguito.

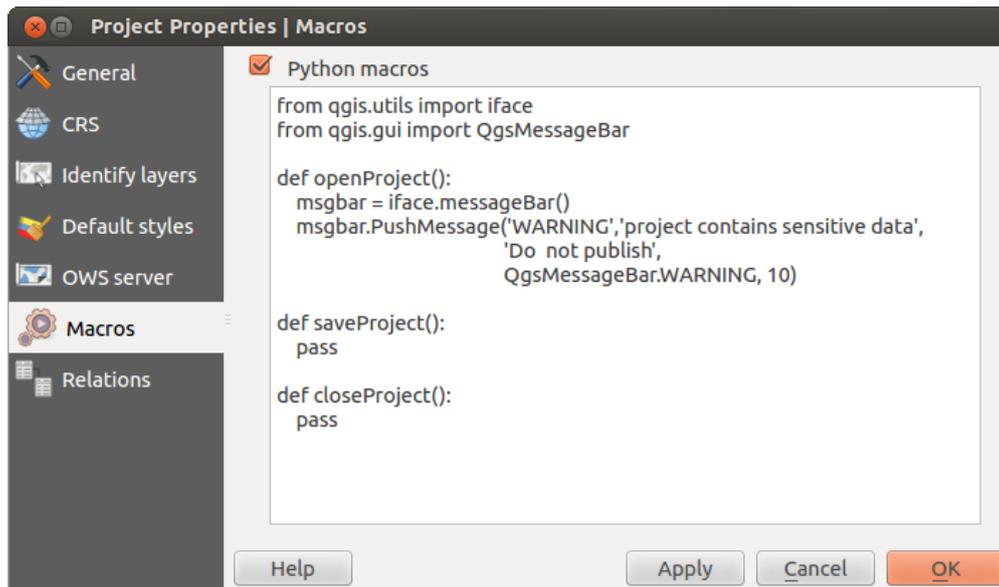


Figura 9.2: Impostazioni macro in QGIS

9.3.1 Menu Generale

Applicazione

- Seleziona *Stile* (Necessario il riavvio di QGIS) e scegli fra 'Windows', 'Motif', 'CDE', 'Plastique' e 'Cleanlooks' .
- Definisci il *Tema delle icone* . Attualmente è disponibile solo 'default'.
- Definisci la *Dimensione delle icone* .
- Definisci il *Carattere*. Scegli fra *Qt predefinito* o un carattere di tua scelta.
- Cambia il *Timeout per messaggi o dialoghi a tempo* .
- *Nascondi lo splash screen all'avvio*
- *Mostra suggerimenti all'avvio*
- *Titoli del gruppo box in grassetto*
- *Box gruppi secondo stile QGIS*
- *Usa pannelli scelta colore per aggiornare*

File di progetto

- *Apri un progetto all'apertura di QGIS* (scegli fra 'Nuovo', 'Più recente' e 'Specifico'). Se hai scelto 'Specifico' usa per scegliere il progetto da aprire.
- *Crea un nuovo progetto dal progetto predefinito*. Puoi scegliere *Definisce il progetto attuale come predefinito* oppure *Ripristina il predefinito*. Sfoglia fra i tuoi file e specifica la cartella in cui sono presenti i progetti da usare come modello. Se hai spuntato la casella di controllo *Crea un nuovo progetto dal progetto predefinito* e hai salvato un progetto nella cartella dei modelli, comparirà la nuova voce *Progetto* → *Nuovo da modello*.
- *Chiedi di salvare il progetto e cambia sorgente dati quando richiesto*
- *Avvisa quando viene aperto un file di progetto salvato con una vecchia versione di QGIS*

- **Abilita macro** . Questa opzione serve per gestire le macro create per eseguire azioni sul progetto. Puoi scegliere fra ‘Mai’, ‘Chiedi’, ‘Solo per questa sessione’ e ‘Sempre (sconsigliato)’.

9.3.2 Menu Sistema

Ambiente

Il gruppo **Ambiente** mostra le variabili di ambiente e permette anche di configurarle (vedi [figure_environment_variables](#)). Questa opzione è particolarmente utile per piattaforme stile Mac, dove le applicazioni GUI non ereditano necessariamente l’ambiente shell dell’utente. È utile anche per impostare/visualizzare le variabili di ambiente usate da strumenti esterni, gestiti da Processing come, SAGA e GRASS. Infine è utile anche per abilitare gli output del debug per specifiche sezione del codice sorgente.

-  **Usa variabili utente (necessario il riavvio - includere i separatori)**. Gestisci le variabili con i pulsanti **[Aggiungi]** e **[Rimuovi]**. Il **Variabili di ambiente attuali** → mostra le variabili di ambiente già definite e puoi scegliere di filtrarle spuntando la casella di controllo  **Mostrare le sole variabili specifiche di QGIS**.

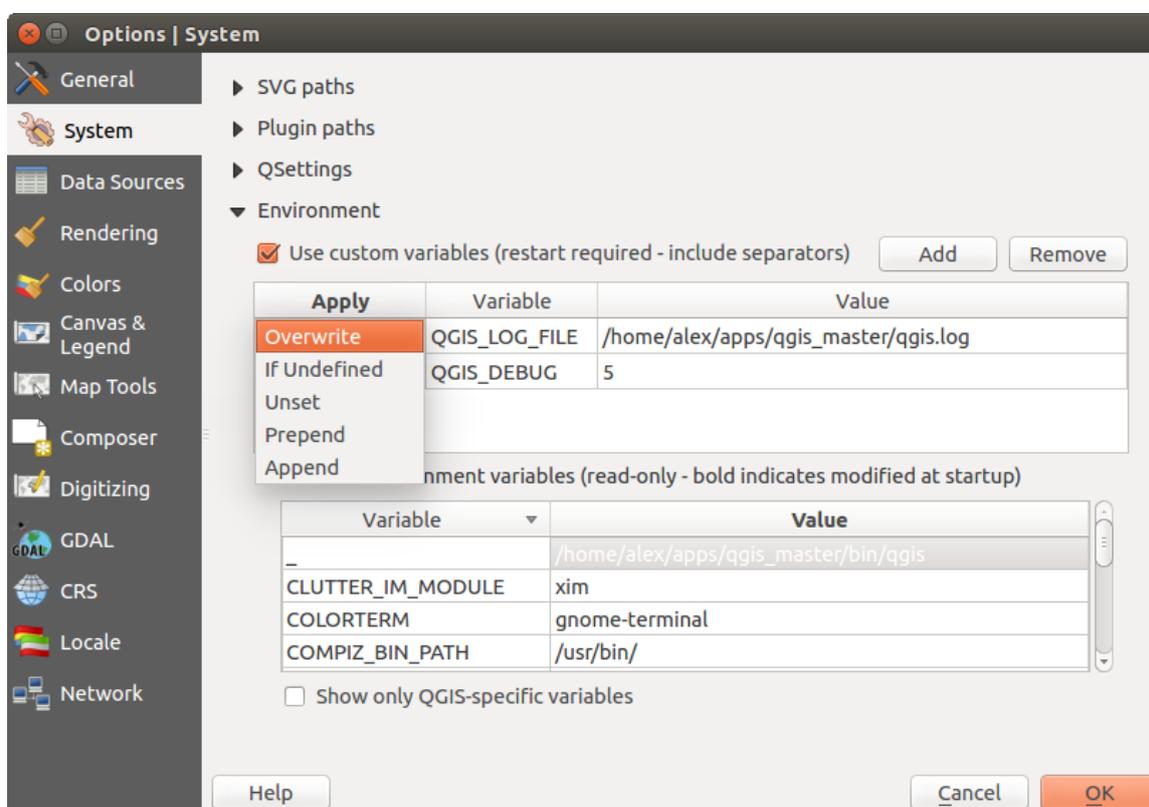


Figura 9.3: Variabili di ambiente in QGIS

Percorsi verso i plugin

[Aggiungi] o **[Rimuovi]** Percorsi per cercare ulteriori librerie plugin C++

9.3.3 Menu Sorgente dati

Attributi delle geometrie e tabelle

-  **Apri la tabella degli attributi in una finestra agganciata (richiede il riavvio di QGIS)**

-  *Copia la geometria in formato WTK dalla tabella degli attributi.* Cliccando su  *Copia le righe selezionate nel blocco appunti* dal menu della *Tabella degli attributi* verranno copiate, oltre agli attributi, anche le coordinate dei vertici della geometria.
- *Comportamento della tabella di attributi* . Sono presenti tre possibilità: ‘Mostra tutti gli elementi’, ‘Mostra gli elementi selezionati’ e ‘Mostra gli elementi visibili nella mappa’.
- *Cache riga attributi tabella* . La cache permette di salvare le ultime n righe degli attributi caricate: in questo modo il lavoro con la tabella degli attributi risulterà essere molto più veloce. La cache verrà cancellata alla chiusura della tabella degli attributi.
- *Mostra i valori NULL come.* Puoi definire un attributo con cui verranno visualizzati i valori NULL (nessun valore).

Trattamento delle sorgenti dati

- *Ricerca elementi validi nella finestra del browser* . Puoi scegliere fra ‘Controlla estensione’ e ‘Controlla il contenuto del file’.
- *Ricerca contenuto dei file compressi (.zip) nella finestra del browser* . Puoi scegliere fra ‘No’, ‘Scansione base’ e ‘Scansione completa’.
- *Richiedi i sublayer raster al caricamento.* Alcuni raster supportano i sublayer, chiamati subdataset in GDAL. Un esempio sono i file netCDF: se sono presenti diverse variabili netCDF, GDAL riconosce ogni variabile come un subdataset. L’opzione permette di gestire i sublayer quando uno di questi viene aperto. Puoi scegliere fra:
 - ‘Sempre’: chiede sempre (se sono presenti sublayer)
 - ‘Se necessario’: chiede se il layer non ha bande, ma ha sublayer
 - ‘Mai’: non chiede mai e non carica niente
 - ‘Carica tutto’: non chiede, ma carica tutti i sublayer
-  *Ignora la dichiarazione di codifica per lo shapefile.* Se lo shapefile ha informazioni sulla sua codifica, queste verranno ignorate.
-  *Aggiungi un layer PostGIS con un doppio click e seleziona la modalità estesa*
-  *Aggiungi layer Oracle con un doppio click e seleziona modalità estesa*

9.3.4 Menu Visualizzazione

Impostazioni di visualizzazione

-  *Per impostazione predefinita i nuovi layer aggiunti alla mappa vengono visualizzati subito*
-  *Usa il caching del disegno quando possibile per velocizzare la visualizzazione*
-  *Visualizza i layer in parallelo usando più processori della CPU*
-  *Numero massimo di processori*
- *Intervallo di aggiornamento della mappa (predefinito a 250 ms)*
-  *Abilita semplificazione delle geometrie per i nuovi vettori aggiunti*
- *Semplifica dal lato provider se possibile*
-  *Semplifica dal lato provider se possibile*
- *Scala massima alla quale il layer dovrebbe essere semplificato*

Impostazioni di visualizzazione

- *Rendi le linee meno irregolari a spese delle prestazioni*

Raster

- Con *Selezione banda RGB* puoi scegliere il numero di bande rosse, verdi e blu.

Miglioramento contrasto

- *Banda singola grigia* . Una singola banda grigia può avere: ‘Nessuno stiramento’, ‘Stira a MinMax’, ‘Stira e taglia a MinMax’ e anche ‘Taglia a MinMax’.
- *Colore multibanda (byte/band)* . Le opzioni sono ‘Nessuno stiramento’, ‘Stira a MinMax’, ‘Stira e taglia a MinMax’ e ‘Taglia a MinMax’.
- *Colore multibanda (>byte/band)* . Le opzioni sono ‘Nessuno stiramento’, ‘Stira a MinMax’, ‘Stira e taglia a MinMax’ e ‘Taglia a MinMax’.
- *Limiti (minimo/massimo)* . Le opzioni sono ‘Taglio del conteggio cumulativo pixel’, ‘Minimo/Massimo’, ‘Media +/- deviazione standard’
- *Limiti di taglio del conteggio cumulativo pixel*
- *Moltiplicatore deviazione standard*

Debugging

- *Aggiornamento della visualizzazione della mappa*

9.3.5 Menu Colori

Questo menu ti consente di aggiungere colori personalizzati che potrai trovare in ogni finestra di dialogo colore delle visualizzazioni. Vedrai una serie di colori predefiniti nella scheda, che potrai modificare o eliminare. Inoltre puoi aggiungere i colori desiderati ed eseguire operazione di copia e incolla. Infine è possibile esportare il set di colori come `gpl` o importarne.

9.3.6 Menu Mappa & Legenda

Aspetto della mappa (modificato dalle proprietà del progetto)

- Scegli un *Colore della selezione* e un *Colore di sfondo*

Legenda dei layer

- *Azione eseguita in legenda sul comando doppio click* . Puoi scegliere fra ‘Apri proprietà layer’ oppure ‘Apri tabella degli attributi’ facendo doppio click sul layer.
- Puoi scegliere diverse opzioni per gli *Stili elementi legenda*:
 - *Nomi dei layer in maiuscolo*
 - *Nomi dei layer in grassetto*
 - *Nomi gruppo in grassetto*
 - *Visualizza nomi di classificazione degli attributi*
 - *Crea le icone raster (potrebbe essere lento)*
 - *Aggiungi nuovi layer al gruppo selezionato*

9.3.7 Menu Strumenti mappa

Questo menu ti permette di scegliere alcune opzioni sul comportamento di *Informazioni*.

- *Raggio di ricerca per identificare e visualizzare le relative informazioni sulla mappa* è un fattore di tolleranza espressa come percentuale della larghezza mappa. Ciò significa che lo strumento di informazione mostrerà i risultati se si fa clic all'interno di questa tolleranza.
- *Colore di evidenziazione* di scegliere con quale colore saranno evidenziati gli elementi identificati.
- *Buffer* espressa come percentuale della larghezza mappa, determina una distanza buffer dal contorno dell'elemento da evidenziare.
- *Larghezza minima* espressa come percentuale della larghezza mappa, determina lo spessore che dovrebbe il contorno di un oggetto evidenziato.

Strumenti di misura

- Colore elastico
- Posizioni decimali
- *Mantieni le unità di base*
- *Unità di misura preferita*  ('Metri', 'Piedi', 'Miglia nautiche' o 'Gradi')
- *Unità preferite per gli angoli*  ('Gradi', 'Radianti' o 'Gradi decimali')

Spostamento e zoom

- *Define Mouse wheel action*  ('Zoom', 'Zoom and recenter', 'Zoom to mouse cursor', 'Nothing')
- Fattore di zoom

Scale preimpostate

Qui trovi un elenco di scale preimpostate. Con i pulsanti [+] e [-] puoi aggiungere ed eliminare altre scale.

9.3.8 Menu Compositore

Opzioni predefinite del compositore

Qui puoi impostare il *Carattere predefinito*.

Reticolo

- Specifica lo *Stile reticolo*  ('Pieno', 'Punti', 'Croci')
- Scegli il *Colore...*

Opzioni predefinite del reticolo

- Specifica la *Spaziatura*
- Scegli l' *Offset reticolo* per x e y
- Definisci la *Tolleranza di snap*

Opzioni predefinite delle guide

- Definisci la *Tolleranza di snap*

9.3.9 Menu Digitalizzazione

Creazione di geometrie

- *Non aprire la finestra degli attributi dopo la creazione di ogni geometria*

- *Ripeti i valori degli attributi usati per ultimi*
- *Verifica le geometrie.* Modificare linee/poligoni con molti nodi può portare a una visualizzazione molto lenta: questo succede perché le procedure di verifica delle geometrie in QGIS richiede molto tempo. Per velocizzare la visualizzazione è possibile selezionare GEOS (GEOS 3.3) dal menu a tendina oppure disattivare del tutto la verifica. La verifica effettuata da GEOS è molto più rapida, ma ha lo svantaggio di comunicare solamente il primo problema geometrico riscontrato.

Elastico

- Definisci le proprietà dell'elastico, *Spessore della linea* e *Colore della linea*

Snapping

- *Apri le opzioni di snap in una finestra agganciata (richiede il riavvio di QGIS)*
- Imposta la *Modalità di snap predefinita*  ('Al vertice', 'Al segmento', 'Al vertice e al segmento', 'Spento')
- Imposta la *Tolleranza di snapping predefinita* in unità di mappa o pixel
- Imposta il *Raggio di ricerca per le modifiche dei vertici* in unità di mappa o in pixel

Indicatori di vertice

- *Utilizza indicatori solo per le geometrie selezionate*
- Imposta lo *Stile indicatore*  ('Croce' (predefinito), 'Cerchio semi trasparente' o 'Nessuno')
- Specifica le *Dimensioni indicatore*

Strumento per la curva di offset

Le 3 opzioni successive si riferiscono allo strumento  *Curva di offset* in *Digitalizzazione avanzata*. Attraverso i vari settaggi è possibile influenzare la forma della linea di offset. Queste opzioni sono possibili da GEOS 3.3.

- *Stile unione*
- *Segmenti di quadrante*
- *Limite di smusso*

9.3.10 Menu GDAL

GDAL è una libreria di lettura e scrittura per file raster. In questa scheda puoi *Modificare le opzioni di creazione* e *Modificare le opzioni per le piramidi*. Scegli quali driver GDAL devono essere utilizzati per un formato raster specifico poiché spesso sono disponibili più driver GDAL per lo stesso formato.

9.3.11 Menu SR

SR predefinito per nuovi progetti

- *Non abilitare la riproiezione 'al volo'*
- *Abilita automaticamente la 'riproiezione al volo' se i layer hanno SR differente*
- *Effettua sempre la riproiezione al volo*
- *Seleziona un SR e Inizia un nuovo progetto sempre con questo SR*

SR per i nuovi layer

Questa sezione permette di specificare il comportamento di QGIS quando viene creato un nuovo layer oppure quando viene caricato un layer privo di SR.

- *Richiedi SR*

- Usa il SR del progetto
- Utilizza come predefinito SR visualizzato sotto

Trasformazioni datum predefinite

- Chiedi la trasformazione del datum quando non è definito un valore di riferimento
- Se hai lavorato con la trasformazione del SR ‘al volo’ puoi vedere i risultati della trasformazione nella finestra di sotto. Questa finestra mostra infatti le informazioni del ‘SR sorgente’, ‘SR destinazione’, ‘Trasformazione datum sorgente’ e ‘Trasformazione datum destinazione’.

9.3.12 Menu Lingua

- Sovrascrivi lingua in uso
- Informazioni sulla lingua correntemente impostata nel sistema

9.3.13 Menu Rete

Generale

- Indirizzo di ricerca WMS (Quello predefinito è “<http://geopole.org/wms/search?search=%1&type=rss>”)
- Imposta il *Timeout per le richieste di rete (ms)* - il valore predefinito è 60000
- Imposta il *Periodo di scadenza predefinito per piastrelle WMSC/WMTS (ore)* - valore standard è 24
- Specifica il *Numero massimo di tentativi in caso di errore nella richiesta della mattonella*
- Definisci l’*Utente-Agente*

Impostazioni della cache

Specifica la *Cartella* e la *Dimensione* per la cache.

- Utilizza un proxy per l’accesso web, definizione di host, porta, utente e password.
- Definizione del *Tipo proxy*  .
 - *Default Proxy*: Il proxy è determinato sulla base delle impostazioni in uso del proxy dell’applicazione
 - *Socks5Proxy*: Proxy generico per ogni tipo di connessione. Supporta TCP, UDP, associazione a una porta (connessione in entrata) e autenticazione.
 - *HttpProxy*: Realizzato usando il comando “CONNECT”, supporta solamente connessioni TCP in uscita; supporta l’autenticazione.
 - *HttpCachingProxy*: Realizzato usando normali comandi HTTP, è utile solamente nel contesto di richieste HTTP.
 - *FtpCachingProxy*: Realizzato usando un proxy FTP, è utile solamente nel contesto di richieste FTP.

È possibile escludere alcuni URL aggiungendo il testo nella sezione dedicata (vedi [Figure_Network_Tab](#)).

Per informazioni più dettagliate sulle diverse impostazioni del proxy, fai riferimento al manuale della documentazione delle librerie QT su <http://doc.trolltech.com/4.5/qnetworkproxy.html#ProxyType-enum>.

Suggerimento: UTILIZZO DEI PROXY

L’utilizzo dei proxy può risultare complicato. È utile testare i tipi di proxy elencati sopra e controllare il loro funzionamento nel tuo caso specifico.

Queste opzioni possono essere modificate in funzione delle proprie esigenze. Alcuni cambiamenti potrebbero richiedere il riavvio di QGIS prima di essere attivi.

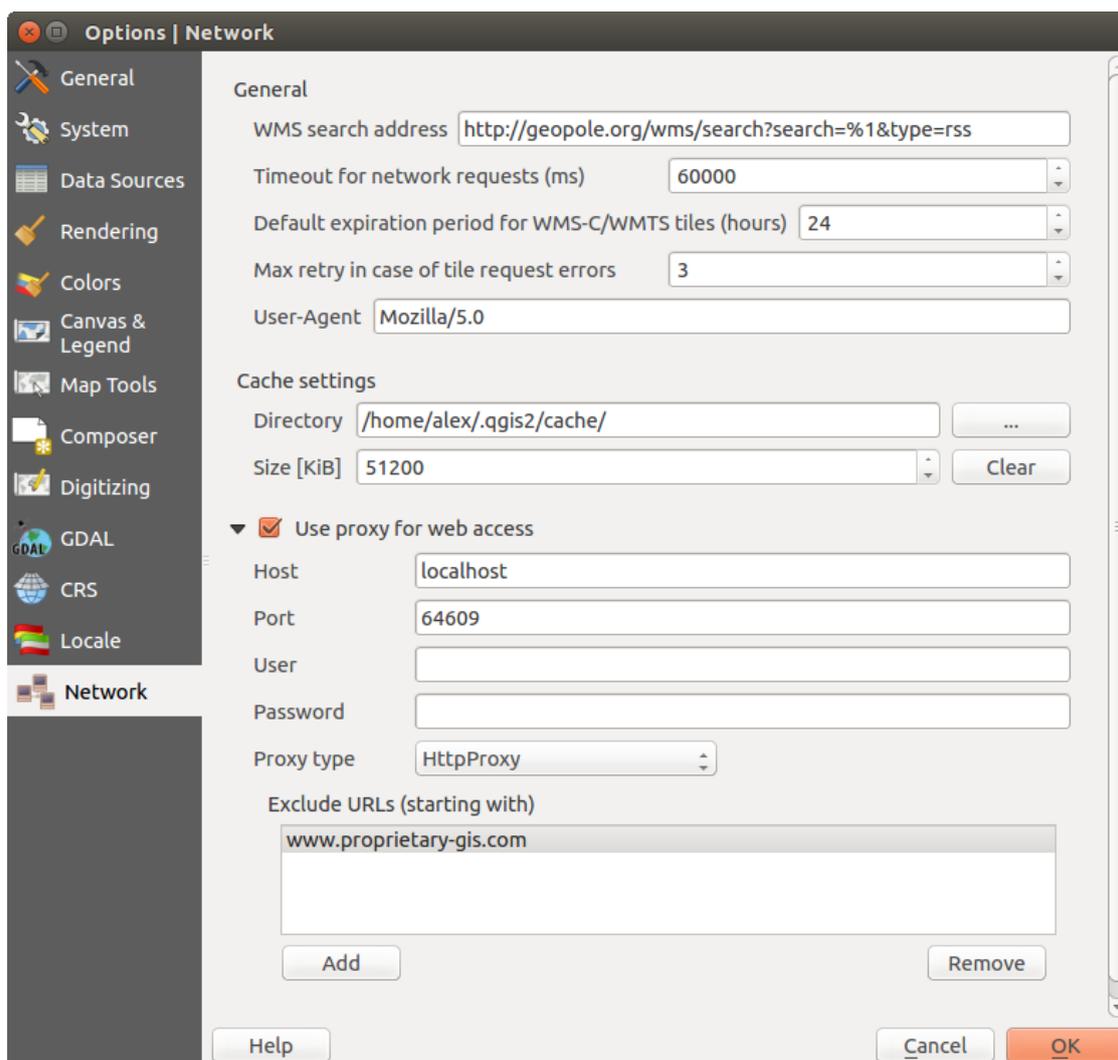


Figura 9.4: Impostazioni dei proxy in QGIS

-  Impostazioni sono salvate in file testo \$HOME/.config/QGIS/QGIS2.conf
- **X** puoi trovare le impostazioni in: \$HOME/Library/Preferences/org.qgis.qgis.plist
-  le impostazioni sono salvate nel registro: HKEY\CURRENT_USER\Software\QGIS\qgis

9.4 Personalizzazione

Lo strumento di personalizzazione permette di attivare/disattivare praticamente qualunque elemento dell'interfaccia di QGIS. È molto utile se sono stati installati molti plugin, usati raramente e che riempiono lo schermo e la barra degli strumenti.

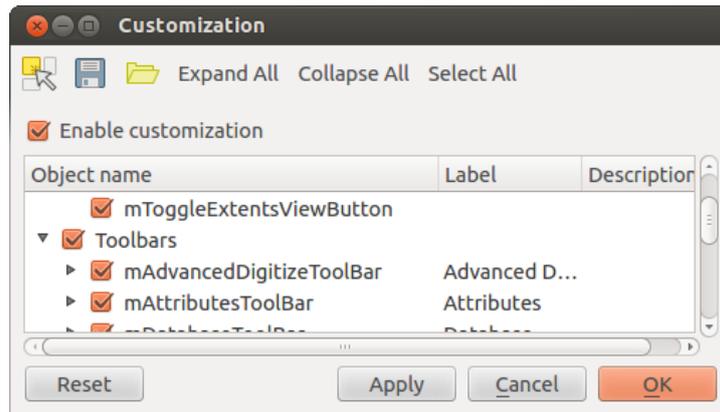


Figura 9.5: Finestra di dialogo Personalizzazione 

Il menu Personalizzazione di QGIS è diviso in cinque gruppi. In  *Menu* è possibile nascondere funzioni all'interno della Barra dei Menu. In  *Panelli* puoi trovare tutti i pannelli presenti in QGIS. I pannelli sono applicazioni che possono essere avviate e usate come finestre mobili, fisse oppure possono essere incorporate nella finestra principale di QGIS come se fossero dei widget (see also *Pannelli e Barre degli strumenti*). In  *Barra di stato* è possibile disattivare caratteristiche come le informazioni sulle coordinate. In  *Barra degli strumenti* è possibile attivare/disattivare le icone della barra degli strumenti di QGIS e in  *Widgets* si possono attivare/disattivare i diversi pannelli presenti in QGIS.

Con  *Passa ai widget per la cattura nell'applicazione principale* puoi cliccare sugli elementi di QGIS che vuoi nascondere dall'interfaccia e ritrovarli così nel menu Personalizzazione (vedi [figure_customization](#)). Puoi anche salvare le diverse impostazioni a seconda dei diversi utilizzi. Prima che i cambiamenti diventino effettivi devi riavviare QGIS.

Lavorare con le proiezioni

QGIS ti consente di definire un sistema di riferimento - SR - (Coordinate Reference System, ovvero sistema di riferimento delle coordinate) globale o a livello di singolo progetto per i layer privi di un SR predefinito. Ti consente inoltre di definire sistemi di coordinate personalizzati e supporta anche la riproiezione al volo (on-the-fly, OTF) di vettori e raster. Tutte queste funzionalità ti permettono di visualizzare contemporaneamente layer con SR diversi.

10.1 Panoramica sul supporto alle proiezioni

QGIS supporta all'incirca 2.700 SR. Le definizioni di ognuno di questi SR sono memorizzate in un database SQLite che viene installato insieme a QGIS. Normalmente non è necessario manipolare il database direttamente, infatti potresti causare il malfunzionamento del supporto alla proiezione. I SR personalizzati invece, sono salvati in un database utente. Vedi la sezione *Sistemi di riferimento personalizzati* per informazioni sulla gestione dei SR personalizzati.

I SR disponibili in QGIS sono basati su quelli definiti dall'European Petroleum Survey Group - EPSG - e dall'Institut Geographique National francese (IGN) e sono ricavati essenzialmente dalle tabelle di riferimento spaziale usate da GDAL. I codici EPSG sono presenti nel database e li puoi usare per identificare e specificare i SR in QGIS.

Per usare la riproiezione al volo (OTF), i dati devono contenere informazioni sul proprio sistema di riferimento, altrimenti devi definire un SR per il layer, a livello di progetto o a livello globale. Per i layer PostGIS, QGIS usa l'identificatore del riferimento spaziale specificato al momento della creazione del layer. Per i dati supportati da OGR, QGIS fa affidamento sulla presenza di un "mezzo" specifico per ciascun formato, che definisce il SR. Nel caso degli shapefile, ad esempio, si tratta di un file contenente l'indicazione del SR in formato Well Known Text (WKT). Il file della proiezione ha lo stesso nome dello shapefile, ma ha estensione `.prj`. Per esempio lo shapefile chiamato `alaska.shp`, avrà un corrispondente file di proiezione chiamato `alaska.prj`.

Ogni volta che selezioni un nuovo SR le unità dei layer vengono automaticamente cambiate nella scheda *Generale* della finestra di dialogo  *Proprietà progetto* del menu *Progetto* (Gnome, OS X) o *Impostazioni* (KDE, Windows).

10.2 Specifiche globali delle proiezioni

QGIS imposta il SR di ogni nuovo progetto su quello definito a livello globale: il SR globale predefinito è EPSG:4326 - WGS 84 (`proj=longlat +ellps=WGS84 +datum=WGS84 +no_defs`). Puoi modificare il SR predefinito tramite il pulsante **[Scegli...]** presente nella parte superiore della scheda SR, come mostrato in figura [figure_projection_1](#). Questa impostazione verrà salvata e sarà valida per tutte le successive sessioni di QGIS.

Quando usi layer che non hanno un SR, devi specificare come QGIS deve interpretare questi layer. La scelta può essere globale oppure solamente per il progetto corrente: le impostazioni sono disponibili nella scheda *SR* del menu *Impostazioni* →  *Opzioni*.

Le opzioni mostrate in figura [figure_projection_1](#) sono:

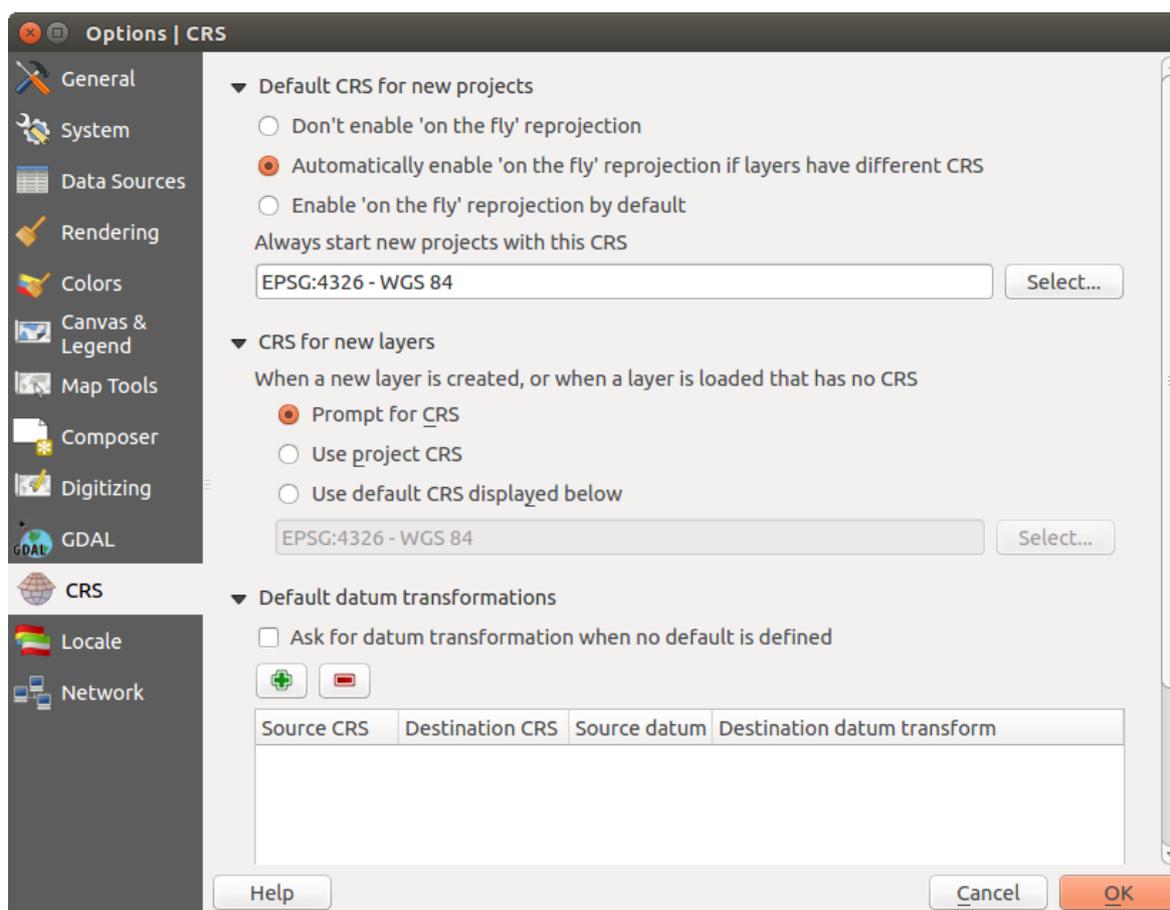


Figura 10.1: Scheda SR nella finestra di dialogo Opzioni 🐧

- *Richiedi SR*
- *Usa il SR del progetto*
- *Utilizza come predefinito SR visualizzato sotto*

Se vuoi specificare il SR di un determinato layer che non ha queste informazioni, puoi farlo anche attraverso la scheda *Generale* delle proprietà dei raster (vedi *Menu Generale*) e dei vettori (vedi *Menu Generale*). Se il layer ha già le informazioni sul SR, saranno visualizzate come in figura *Finestra di dialogo delle proprietà del vettore*.

Suggerimento: SR NELLA LEGENDA

Facendo click con il tasto destro su di un layer in legenda (sezione *Legenda*) hai a disposizione due scorciatoie per l'impostazione del SR. *Imposta il SR del layer* apre direttamente la finestra per la scelta del sistema di riferimento (figura [figure_projection_2](#)), mentre *Imposta il SR del progetto dal layer* imposterà il SR del progetto sulla base di quello del layer

10.3 Definire la riproiezione al volo (OTF)

QGIS supporta la riproiezione al volo (OTF) sia per i raster che per i vettori, ma l'opzione non è abilitata in modo predefinito. Per usare la riproiezione OTF, devi spuntare la casella di controllo *Abilita la riproiezione al volo* nella scheda *SR* della finestra di dialogo  *Proprietà progetto*.

Hai tre modi diversi per farlo:

1. Seleziona  *Proprietà progetto* dal menu *Progetto* (Gnome, OSX) oppure *Impostazioni* (KDE, Windows).
2. Cliccare sull'icona  Stato SR nell'angolo in basso a destra della barra di stato.
3. Abilita la riproiezione al volo in modalità predefinita spuntando la casella di controllo *Effettua sempre la riproiezione al volo* nella scheda *SR* della finestra di dialogo *Opzioni* oppure spunta *Abilita automaticamente la riproiezione al volo se i layer hanno SR differente*.

Se hai già caricato un layer e vuoi abilitare la riproiezione al volo, la scelta migliore è: aprire la scheda *Sistema di riferimento (SR)* della finestra di dialogo *Proprietà progetto*, selezionare nell'elenco il SR attualmente impostato, quindi attivare la casella di controllo *Abilita la riproiezione al volo*. Ogni layer caricato successivamente sarà riproiettato al volo nel SR mostrato vicino all'icona  Stato SR e quest'icona diventerà attiva a tutti gli effetti.

La scheda *SR* della finestra di dialogo *Proprietà progetto* contiene cinque importanti componenti, come puoi vedere nella figura [Figure_projection_2](#):

1. **Abilita la riproiezione al volo** — puoi utilizzare questa casella di controllo per abilitare o disabilitare la riproiezione al volo. Quando è disabilitata, ogni layer verrà visualizzato in funzione del proprio sistema di riferimento e le componenti descritte sotto non saranno attive. Quando è abilitata, ogni layer verrà visualizzato nel sistema di riferimento specificato.
2. **Filtro** — se conosci il codice EPSG, l'identificatore o il nome del SR che vuoi impostare, puoi utilizzare questa area di ricerca per trovarlo nell'elenco. Inserisci il codice EPSG, l'identificatore o il nome.
3. **Sistemi di riferimento usati di recente** — se ci sono dei SR che usi frequentemente, questi verranno visualizzati in questa sezione della finestra di dialogo. Clicca su una voce per impostare il SR associato.
4. **Sistemi di riferimento mondiali** — questa è una lista di tutti i SR supportati da QGIS, compresi quelli geografici, proiettati e personalizzati. Per specificare un SR, selezionalo dalla lista: il SR attivo verrà evidenziato.
5. **Testo PROJ.4** - è la stringa SR usata dal motore di proiezione Proj4. È un testo di sola lettura, a solo scopo informativo.

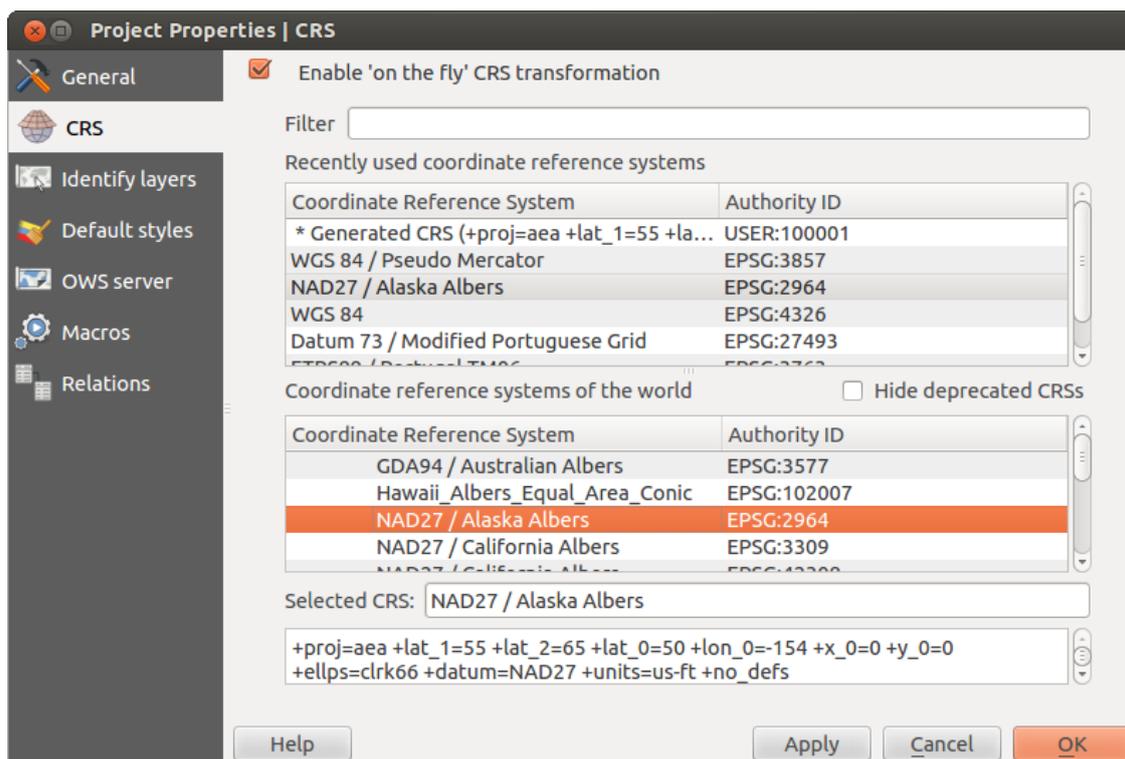


Figura 10.2: Finestra di dialogo delle proprietà delle proiezioni 🐧

Suggerimento: Finestra di dialogo Proprietà del progetto

Se apri la finestra di dialogo *Proprietà progetto* dal menu *Progetto*, per poter visualizzare le impostazioni del SR devi cliccare sulla scheda *SR*.

Se clicchi sull'icona  Stato SR si aprirà automaticamente la scheda *SR*.

10.4 Sistemi di riferimento personalizzati

Se QGIS non ha le informazioni sul sistema di riferimento di cui hai bisogno, puoi crearne uno personalizzato. Per farlo, seleziona  *SR personalizzato...* dal menu *Impostazioni*. I SR personalizzati vengono salvati nel database utente di QGIS. Oltre ai SR personalizzati, questo database contiene anche i segnalibri geospaziali e altri dati utente.

Definire un SR personalizzato in QGIS richiede una buona comprensione delle librerie PROJ.4. Per iniziare, fai riferimento al documento *Cartographic Projection Procedures for the UNIX Environment - A User's Manual* di Gerald I. Evenden, U.S. Geological Survey Open-File Report 90-284, 1990 (disponibile all'indirizzo <ftp://ftp.remotesensing.org/proj/OF90-284.pdf>).

Questo manuale descrive l'uso di `proj.4` e delle relative utilità da riga di comando. I parametri cartografici usati da `proj.4` sono descritti nel manuale e sono identici a quelli usati da QGIS.

La finestra di dialogo *Definizione Sistema Riferimento Spaziale Personalizzato* richiede solamente due parametri per definire un SR personalizzato:

1. Il nome
2. I parametri cartografici in formato PROJ.4.

Per creare un nuovo SR, clicca sul pulsante  *Aggiungi nuovo SR* e inserisci il nome e i parametri.

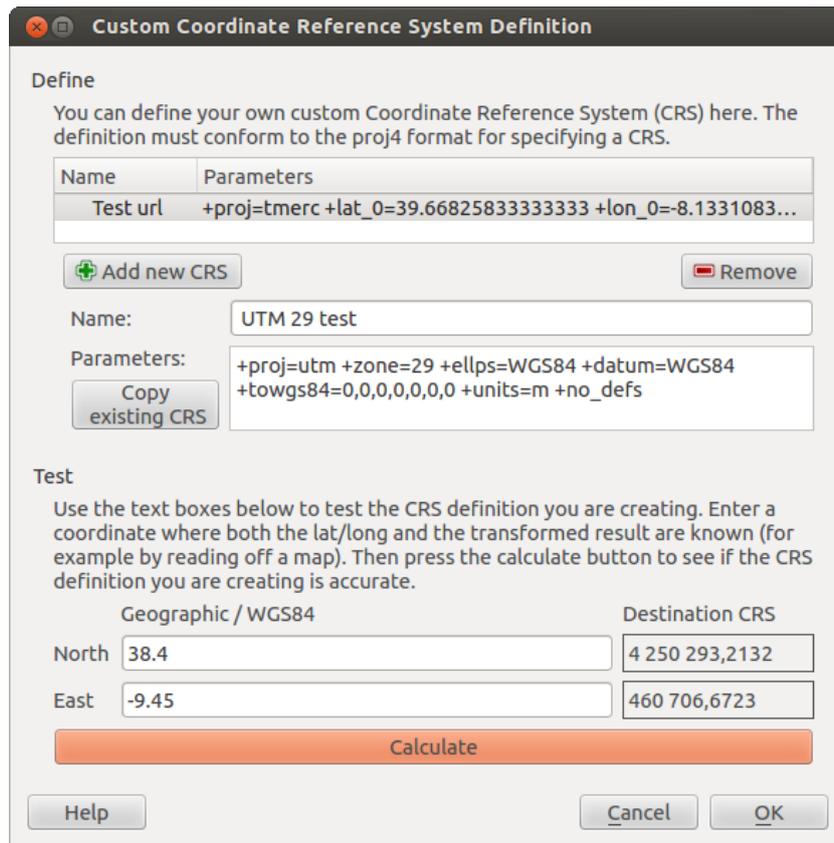


Figura 10.3: Finestra di dialogo SR 

La voce *Parametri* deve iniziare con un blocco `+proj=`, per rappresentare il nuovo SR.

Puoi testare i parametri del tuo SR per vedere se danno risultati validi. Per farlo, inserisci due valori noti di latitudine e longitudine nel sistema WGS 84 rispettivamente in *Nord* ed *Est*. Clicca su **[Calcola]** e fai un paragone dei risultati con i valori noti del tuo SR personalizzato.

10.5 Trasformazioni datum predefinite

La riproiezione al volo dipende dalla capacità di trasformare i dati in un 'SR predefinito' che in QGIS è WGS84. Per alcuni SR sono disponibili molti tipi di trasformazione. QGIS ti permette di definire la trasformazione da usare, altrimenti verrà usata la trasformazione predefinita.

Dalla scheda *SR* del menu *Impostazioni* →  *Opzioni* puoi:

- impostare se QGIS deve chiedere quando deve definire una trasformazione tramite il pulsante *Chiedi la trasformazione del datum quando non è definito un valore di riferimento*.
- modificare la lista di trasformazioni specificate dall'utente.

In una finestra di dialogo, QGIS chiede quale trasformazione deve usare visualizzando il informazioni di PROJ.4 che descrivono la trasformazione di partenza e quella di destinazione. Puoi ottenere altre informazioni fermando il cursore del mouse sopra una trasformazione. Puoi salvare le impostazioni selezionando il pulsante  *Ricorda la selezione*.

QGIS Browser

QGIS Browser è un pannello in QGIS che ti consente di navigare facilmente nel file system e gestire dati geografici. Puoi avere accesso a vettori (ad esempio, gli shapefile ESRI o file MapInfo), database (ad esempio, PostGIS, Oracle, SpatiaLite o MS SQL spaziale) e connessioni WMS / WFS. Puoi inoltre visualizzare i dati di GRASS (per ottenere i dati in QGIS, vedi *Integrazione con GRASS GIS*).

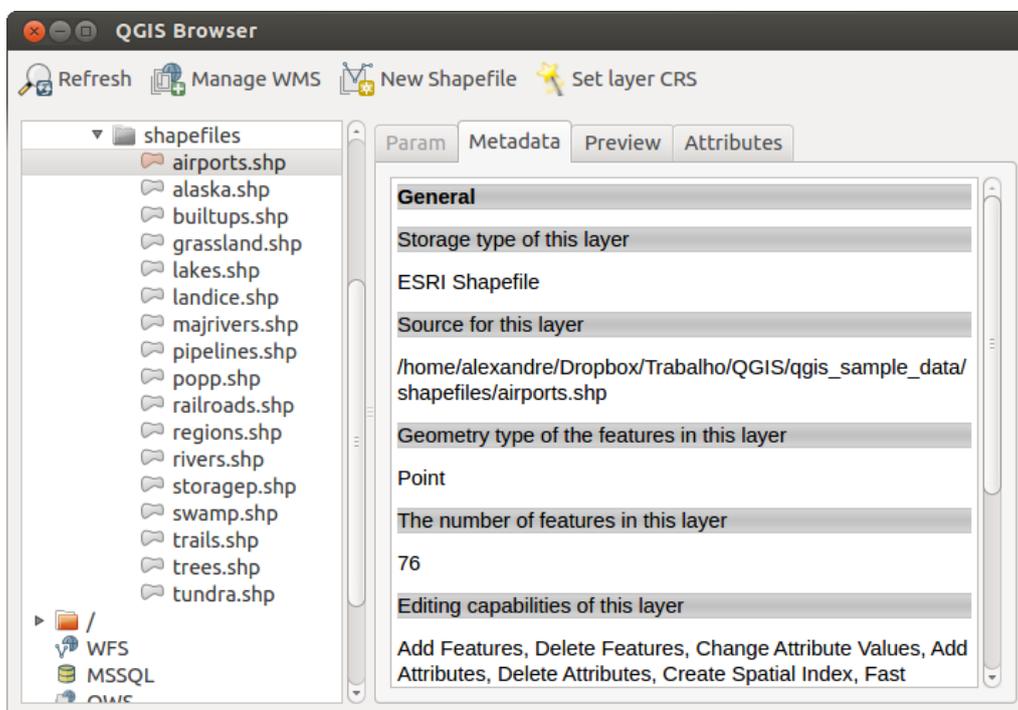


Figura 11.1: QGIS browser come applicazione portabile 🐧

Utilizza QGIS Browser per visualizzare in anteprima i tuoi dati. La funzione di drag-and-drop rende facile ottenere i dati nella mappa e la relativa legenda.

1. Attiva QGIS Browser: click destro sugli strumenti scegli  *Browser* o selezionalo da *Visualizza* → *Pannelli*.
2. Trascina il pannello nella finestra della legenda e rilascialo
3. Click *Browser*.
4. Sfoglia il tuo filesystem e scegli la cartella *shapefile* nella cartella *qgis_sample_data*.
5. Premi Shift e seleziona i file *airports.shp* e *alaska.shp*.
6. Premi il tasto sinistro del mouse, poi trascina i file nella mappa.

7. Cliccare con il tasto destro su un layer e scegliere *Imposta il SR del progetto dal layer*. Per ulteriori informazione vedi *Lavorare con le proiezioni*.
8. Click on  Zoom Full to make the layers visible.

C'è un secondo browser che puoi utilizzare in *Visualizza* → *Pannelli*. Questo è utile quando è necessario spostare file o layer tra posizioni.

1. Attiva un secondo QGIS Browser: click destro sulla barra degli strumenti e scegli  *Browser (2)* o selezionalo da *Visualizza* → *Pannelli*.
2. Trascinare il pannello nella finestra della Legenda.
3. Scegli *Browser (2)* e individua uno shapefile nel tuo file system
4. Select a file with the left mouse button. Now you can use the  Add Selected Layers icon to add it into the current project.

QGIS automatically looks for the coordinate reference system (CRS) and zooms to the layer extent if you work in a blank QGIS project. If there are already files in your project, the file will just be added, and in the case that it has the same extent and CRS, it will be visualized. If the file has another CRS and layer extent, you must first right-click on the layer and choose *Set Project CRS from Layer*. Then choose *Zoom to Layer Extent*.

The  Filter files function works on a directory level. Browse to the folder where you want to filter files and enter a search word or wildcard. The Browser will show only matching filenames – other data won't be displayed.

It's also possible to run the QGIS Browser as a stand-alone application.

Start the QGIS browser

-  Digitare “qbrowser” nella finestra del terminale.
-  Start the QGIS Browser using the Start menu or desktop shortcut.
-  The QGIS Browser is available from your Applications folder.

In [figure_browser_standalone_metadata](#), you can see the enhanced functionality of the stand-alone QGIS Browser. The *Param* tab provides the details of your connection-based datasets, like PostGIS or MSSQL Spatial. The *Metadata* tab contains general information about the file (see *Menu Metadati*). With the *Preview* tab, you can have a look at your files without importing them into your QGIS project. It's also possible to preview the attributes of your files in the *Attributes* tab.

Lavorare con i vettori

12.1 Formati supportati

QGIS usa la libreria OGR per leggere e creare vettori, inclusi i formati ESRI shapefile, MapInfo e MicroStation, AutoCAD DXF, PostGIS, Spatialite, database Oracle Spatial e MSSQL, e molti altri ancora. Il supporto ai vettori GRASS e a PostgreSQL è fornito da plugin nativi. I vettori possono anche essere caricati in modalità lettura da archivi zip e gzip. Attualmente la libreria OGR supporta 69 formati di vettori (vedi OGR-SOFTWARE-SUITE in *Letteratura e riferimenti web*). La lista completa è disponibile all'indirizzo http://www.gdal.org/ogr/ogr_formats.html.

Nota: Alcuni dei formati elencati all'indirizzo citato potrebbero non essere supportati da QGIS per diverse ragioni: ad esempio, alcuni richiedono librerie commerciali esterne oppure GDAL/OGR è stata installata senza scegliere il supporto per uno specifico formato. Solo i formati adeguatamente testati appariranno nella lista di tipi di file al momento del caricamento di un vettore all'interno di QGIS. Altri formati, non testati, possono essere caricati selezionando *.*.*.

La sezione *Integrazione con GRASS GIS* ti mostra come lavorare con i dati di GRASS.

Questa sezione descrive come lavorare con diversi formati comuni: ESRI shapefile, vettori PostGIS, vettori Spatialite, vettori OpenStreetMap e file CSV. Molti degli strumenti disponibili in QGIS funzionano allo stesso modo a prescindere dal tipo di formato (ad es. l'identificazione, la selezione, le etichette e le funzioni degli attributi).

12.1.1 Shapefile ESRI

Il formato di file predefinito in QGIS è ESRI shapefile. Il supporto al formato è fornito dalla libreria OGR Simple Feature Library (<http://www.gdal.org/ogr/>).

Uno shapefile è costituito da di un minimo di tre file:

1. `.shp` contenente le geometrie
2. `.dbf` contenente gli attributi in formato dBase
3. `.shx` contenente l'indice

Uno shapefile può anche includere un file con suffisso `.prj` che contiene le informazioni sulla proiezione. Anche se non è obbligatorio, è molto utile avere informazioni sulla proiezione del file. Un insieme di dati shapefile può contenere anche altri tipi di file. Per ulteriori informazioni, vedi le specifiche tecniche di ESRI all'indirizzo <http://www.esri.com/library/whitepapers/pdfs/shapefile.pdf>.

Caricare uno shapefile

Per caricare uno shapefile, avvia QGIS e clicca sul pulsante  **Aggiungi vettore** o semplicemente premi **Ctrl+Shift+V**. Si aprirà così una nuova finestra (vedi [figure_vector_1](#)).

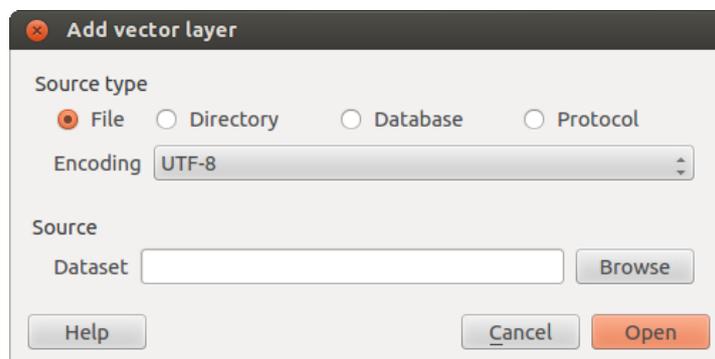


Figura 12.1: Finestra di dialogo aggiungi vettore 

Nella finestra di dialogo seleziona **File** e clicca su **[Sfogli]**. Si aprirà così una finestra di dialogo standard (vedi [figure_vector_2](#)) che ti consentirà di cercare nel computer lo shapefile o qualunque altro dato vettoriale che vuoi caricare. La casella di controllo *Tipo file*  consente di selezionare in anticipo specifici formati supportati da OGR.

Se vuoi, puoi selezionare il tipo di codifica per lo shapefile.

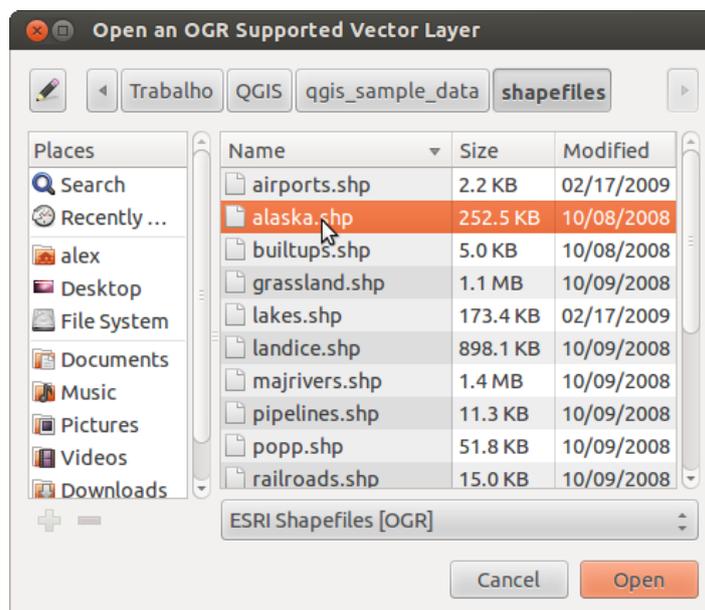


Figura 12.2: Finestra di dialogo apri vettore OGR 

Per caricare un vettore in QGIS, seleziona uno shapefile dalla lista e clicca su **[Apri]**. La figura [Figure_vector_3](#) mostra come appare l'interfaccia di QGIS dopo aver caricato il file `alaska.shp`.

Suggerimento: Colori dei vettori

Quando aggiungi un vettore alla mappa, gli viene assegnato un colore casuale. Se aggiungi più vettori in una sola volta, ciascuno avrà un colore diverso.

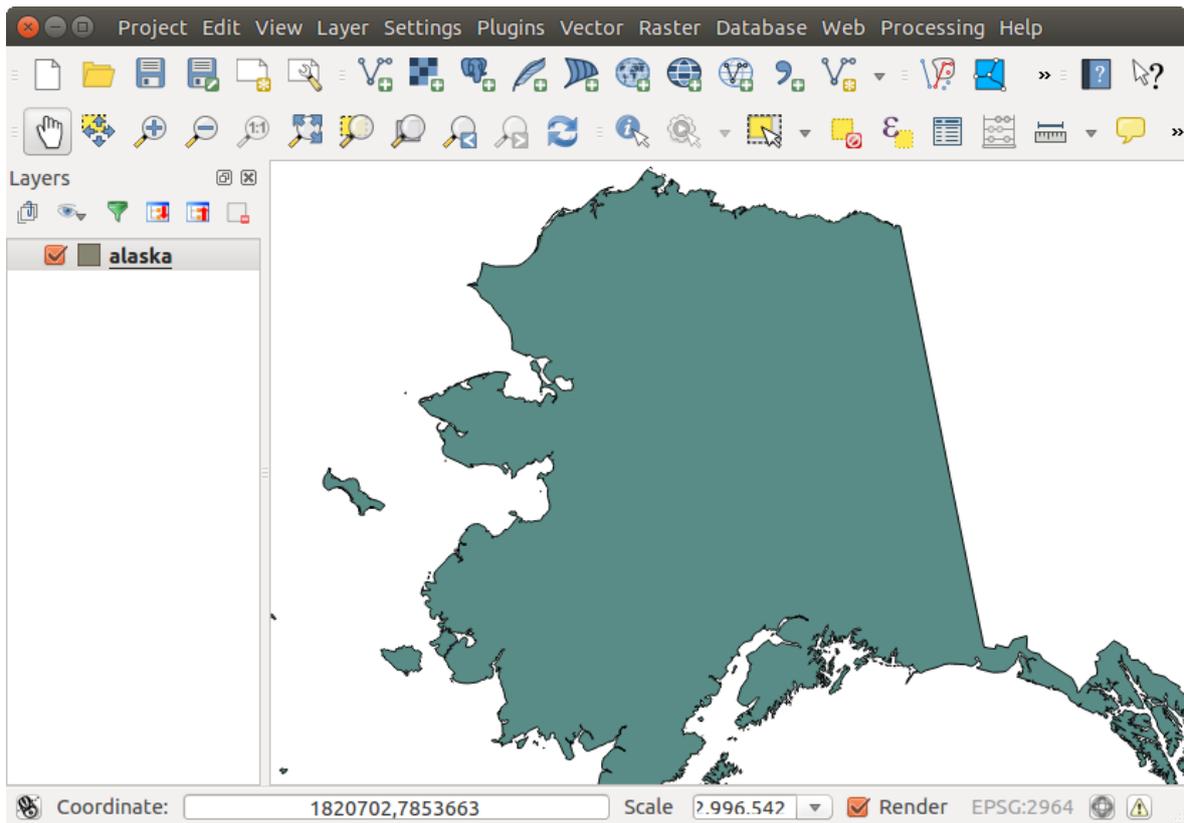


Figura 12.3: QGIS con i dati campione Alaska caricati 🐧

Una volta caricato lo shapefile, puoi interagire con la mappa usando gli strumenti di navigazione. Per cambiare lo stile di un vettore, apri la finestra di dialogo *Proprietà layer* facendo doppio click sul nome del vettore oppure cliccando con il tasto destro sul nome del vettore e scegliendo *Proprietà*. Vedi la sezione *Menu Stile* per ulteriori informazioni su come impostare la simbologia dei vettori.

Suggerimento: Caricare vettori e progetti da periferiche esterne in OS X

In OS X le periferiche esterne montate accanto al disco fisso principale non sono visibili nel menu *File* → *Apri progetto*. Stiamo lavorando per cercare di risolvere questo problema con OS X; come soluzione temporanea digita */Volume* nella casella *Nome file* e clicca *invio*. In questo modo potrai utilizzare le periferiche esterne.

Ottimizzare le prestazioni per gli shapefile

Per migliorare le prestazioni di visualizzazione di uno shapefile, puoi creare un indice spaziale. L'indice spaziale migliora la velocità di visualizzazione quando usi le funzioni di zoom e di spostamento. Gli indici spaziali usati da QGIS hanno estensione `.qix`.

Segui questi passi per creare un indice spaziale:

- Carica uno shapefile cliccando sul pulsante  *Aggiungi vettore* oppure premi `Ctrl+Shift+V`.
- Apri la finestra di dialogo *Proprietà layer* facendo doppio click sul nome dello shapefile nella legenda o cliccandoci con il tasto destro e scegliendo *Proprietà* dal menu contestuale.
- Nella scheda *Generale* clicca sul pulsante **[Crea indice spaziale]**.

Problemi nel caricare un file .prj

Se carichi uno shapefile con file `.prj` associato e QGIS non riesce a leggere le informazioni della proiezione, devi inserire manualmente queste informazioni nella scheda *Generale* della finestra di dialogo *Proprietà layer* cliccando su **[Specifica...]**. Questo è dovuto al fatto che spesso i file `.prj` non forniscono i parametri di proiezione completi che richiede QGIS e che sono elencati nella finestra di dialogo *SR*.

Per la stessa ragione, quando crei un nuovo shapefile in QGIS, vengono creati due differenti file di proiezione. Un file `.prj` che ha un insieme limitato di parametri compatibili con il software ESRI, e un file `.qpj` che memorizza l'insieme completo dei parametri del SR utilizzato. Quando QGIS trova un file `.qpj` utilizza quest'ultimo invece del file `.prj`.

12.1.2 Caricare un vettore MapInfo

 Per caricare un file Mapinfo, clicca sul pulsante  **Aggiungi vettore** o premi `Ctrl+Shift+V`, cambia il filtro nel menu a tendina a *Tipo file* : in 'file MapInfo [OGR] (*.mif *.tab *.MIF *.TAB)' e seleziona il file che vuoi caricare.

12.1.3 Caricare una coverage binaria ArcInfo

Per caricare una ArcInfo Binary Coverage clicca sul pulsante  **Aggiungi vettore** o premi `Ctrl+Shift+V` per aprire la finestra di dialogo *Aggiungi vettore*. Seleziona  *Cartella* come *Tipo sorgente* e cambia il filtro del *Tipo*  con *Coverage binaria Arc/Info*.

In modo simile puoi caricare vettori nel formato di trasferimento UK National Transfer Format oppure anche nel formato TIGER del US Census Bureau.

12.1.4 File di testo delimitato

I dati tabellari sono un formato molto comune ed utilizzato proprio grazie alla loro semplicità e leggibilità – infatti i dati possono essere visualizzati e modificati con un semplice editor di testo. Un file di testo delimitato è una tabella di attributi in cui ogni colonna è separata da un preciso carattere mentre le righe sono separate da un nuovo capoverso. Di solito la prima riga contiene i nomi delle colonne. Un formato molto comune è il CSV (Comma Separated Values), dove ogni colonna è separata da una virgola.

Questi dati possono anche contenere informazioni sulla posizione in due forme principali:

- Come coordinate puntuali in colonne separate
- Come rappresentazione geometrica in Well-Known-Text (WKT)

QGIS ti permette di caricare un file in formato di testo delimitato come layer o come tabella. Prima però assicurati che soddisfi questi requisiti:

1. Il file deve avere una riga di intestazione per il nome dei campi. Questa deve essere la prima riga del testo.
2. La riga di intestazione deve contenere campi relativi alla definizione geometrica. Questi campi possono avere un nome qualunque.
3. Le coordinate X e Y (se la geometria è identificata da coordinate) devono essere specificate come numeri. Il sistema di coordinate non è importante.

Come esempio di un file di testo valido, importa il file di punti quotati `elevp.csv` presente nell'insieme di dati campione di QGIS (vedi sezione *Dati campione*):

```
X;Y;ELEV
-300120;7689960;13
-654360;7562040;52
```

1640; 7512840; 3
[...]

Alcune cose da tenere in considerazione in merito al file di testo:

1. Il file di testo usato come esempio usa ; (punto e virgola) come delimitatore. Ma qualsiasi carattere può essere usato per delimitare i campi.
2. La prima riga è la riga di intestazione. Questa contiene i campi X, Y e ELEV.
3. Nessun tipo di virgoletta (") dev'essere usata per delimitare i campi di testo.
4. Le coordinate X sono contenute nel campo X.
5. Le coordinate Y sono contenute nel campo Y.

Caricare un file di testo delimitato

Clicca sull'icona  Aggiungi layer testo delimitato nella barra degli strumenti *Gestione layer* per aprire la finestra *Crea un vettore da un file di testo delimitato* come mostrato in figura [figure_delimited_text_1](#).

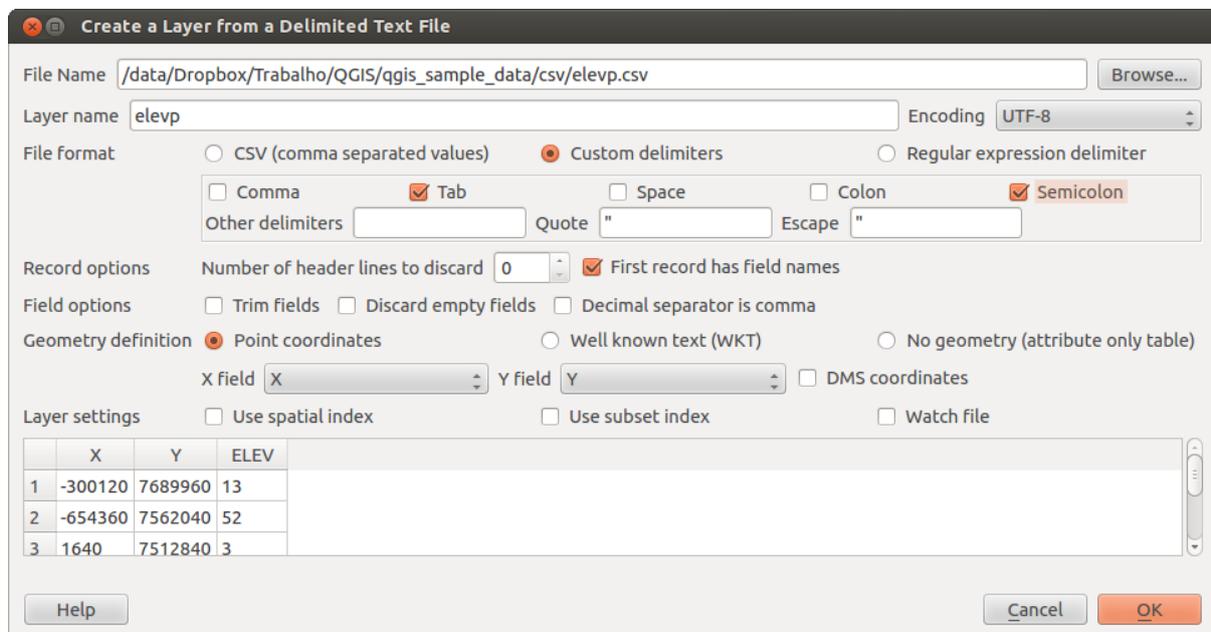


Figura 12.4: Finestra di dialogo Testo Delimitato 

Seleziona il file da importare (es. `qgis_sample_data/csv/elevp.csv`) cliccando sul pulsante **[Sfoggia]**. Una volta che hai scelto il file, QGIS tenta di interpretare il file usando i delimitatori più recenti. Per far sì che QGIS interpreti il file nel modo corretto devi scegliere il delimitatore corretto. Lo puoi specificare con il pulsante *Delimitatori personalizzati* oppure attivando *Delimitatore espressione regolare* e inserendo un testo nel campo *Espressione*. Per esempio, se vuoi usare TAB come delimitatore usa `\t` (questa è l'espressione regolare usata per TAB).

Interpretato il file, imposta *Definizione geometria* in  *Coordinate del punto* e scegli i campi X e Y dal menu a tendina. Se le coordinate sono definite in gradi/minuti/secondi, attiva la casella di controllo  *Coordinate GMS*.

Infine, inserisci il nome del vettore (es. `elevp`), come mostrato in figura [figure_delimited_text_1](#). Per aggiungere il vettore alla mappa, clicca su **[OK]**. Il file di testo delimitato si comporterà così come ogni altro vettore.

Puoi anche tagliare gli spazi iniziali e finali dai campi spuntando la casella di controllo  *Rifinisci i campi*. Inoltre puoi anche spuntare la casella di controllo  *Scarta i campi vuoti*. Se è necessario puoi forzare la lettura della virgola come separatore decimale spuntando la casella di controllo  *La virgola è il separatore decimale*.

Se le informazioni spaziali sono rappresentate da WKT, attiva l'opzione  *Well Known Text* e seleziona il campo del WKT per i punti, linee o poligoni. Se il file contiene dati non-spaziali, attiva  *Nessuna geometria (solo tabella degli attributi)*; in questo modo verrà caricato come una tabella normale.

Puoi anche attivare:

-  *Usa indice spaziale* per migliorare le prestazioni di visualizzazione di selezione delle geometrie.
-  *Usa indice di raggruppamento*
-  *Controlla file* per controllare le modifiche del file effettuate da altri programmi mentre QGIS è avviato.

12.1.5 Dati OpenStreetMap

Il progetto OpenStreetMap (OSM) si sta ampiamente diffondendo, soprattutto in quei paesi dove non si hanno a disposizione dati geografici liberi. L'obiettivo di OSM è la creazione di una mappa mondiale aggiornabile e libera partendo da dati GPS, foto aeree e conoscenza locale. QGIS appoggia questo progetto fornendo supporto ai dati OSM.

Caricare vettori OpenStreetMap

QGIS integra l'importazione dei dati OpenStreetMap come funzionalità di base.

- Per connetterti al server OSM e scaricare i dati, apri il menu *Vettore* → *OpenStreetMap* → *Download dati*. Puoi saltare questo passaggio se hai già a disposizione un file XML `.osm` ottenuto con JOSM, Overpass API o altre fonti.
- Il menu *Vettore* → *OpenStreetMap* → *Importa topologia da XML* convertirà il tuo file `.osm` in un database SpatiaLite e creerà una connessione a questo database.
- Il menu *Vettore* → *OpenStreetMap* → *Esporta topologia in SpatiaLite* ti permette quindi di connetterti al database, selezionare le etichette e i tipi di dati (punti, linee o poligoni) che vuoi importare.

Questo creerà una geometria SpatiaLite che potrai aggiungere al tuo progetto cliccando sul pulsante .

Aggiungi vettore SpatiaLite... oppure selezionando l'opzione  *Aggiungi vettore SpatiaLite...* dal menu *Layer* (vedi sezione *Vettori SpatiaLite*).

12.1.6 Vettori PostGIS

I vettori PostGIS sono memorizzati in un database PostgreSQL. PostGIS ha diversi vantaggi: creazione dell'indice spaziale, capacità di filtraggio e di interrogazione molto potenti. Usando PostGIS, le funzioni vettoriali come la selezione e l'identificazione sono più precise che con i vettori OGR in QGIS.

Creare una connessione

La prima volta che usi PostGIS come sorgente dati, devi creare una connessione al database PostgreSQL che contiene i dati. Clicca sul pulsante  *Aggiungi vettore PostGIS...* oppure seleziona l'opzione  *Aggiungi vettore PostGIS...* dal menu *Layer* oppure ancora premi `Ctrl+Shift+D`. Puoi anche usare la finestra di dialogo *Aggiungi vettore* e selezionare il pulsante  *Database*. Si aprirà così la finestra di dialogo *Aggiungi tabella(e) PostGIS*. Clicca su **[Nuovo]** per far aprire la finestra di dialogo *Crea una nuova connessione PostGIS* e accedi al gestore delle connessioni. I parametri richiesti per la connessione sono:

- **Nome:** Nome della connessione. Può essere uguale a quello del *Database*.

- **Servizio:** Parametri del servizio da usare alternativamente a host/porta (e potenzialmente database). Lo puoi definire in `pg_service.conf`.
- **Host:** Nome del server che ospita il database. Deve essere un host con indirizzo raggiungibile, lo stesso che potrebbe essere usato per aprire una connessione telnet o per fare il ping all'host. Se il database è sullo stesso computer sul quale è installato QGIS, inserisci semplicemente `'localhost'`.
- **Porta:** Numero della porta del server database PostgreSQL. La porta predefinita è 5432.
- **Database:** Nome del database.
- **Modalità SSL:** come la connessione SSL verrà negoziata con il server. Puoi velocizzare la visualizzazione dei vettori PostGIS disabilitando la connessione SSL. Sono disponibili le seguenti opzioni:
 - disabilitato: prova solo una connessione SSL non criptata
 - permesso: tenta una connessione non-SSL, se questa fallisce ne tenta una SSL
 - preferito (predefinito): tenta una connessione SSL, se questa fallisce ne prova una non-SSL
 - richiesto: tenta solo una connessione SSL
- **Nome utente:** Nome dell'utente usato per accedere al database.
- **Password:** Password usata dallo *Username* per collegarsi al database.

Se vuoi, puoi attivare le seguenti caselle di controllo:

- *Salva nome utente*
- *Salva Password*
- *Cercare solamente nella tabella `geometry_columns`*
- *Non risolvere tipo di geometria senza restrizioni (`GEOMETRY`)*
- *Cerca solamente nello schema*
- *Mostra anche tabelle senza geometria*
- *Usa i metadati stimati della tabella*

Quando hai impostato tutti i parametri, puoi testare la connessione cliccando sul pulsante **[Test Connessione]**.

Caricare un vettore PostGIS

Una volta definita la connessione, puoi caricare i vettori dal database PostgreSQL. Ovviamente devi avere i dati in PostgreSQL. Vedi la sezione [Importare dati in PostgreSQL](#) per informazioni su come importare dati nel database.

Per caricare vettori PostGIS, segui i seguenti passaggi:

- Se non hai ancora aperto la finestra di dialogo *Aggiungi vettore PostGIS*, seleziona l'opzione  *Aggiungi vettore PostGIS...* dal menu *Layer* oppure premi `Ctrl+Shift+D`.
- Scegli la connessione dal menu a tendina e clicca su **[Connetti]**.
- Seleziona/deseleziona *Mostra anche le tabelle senza geometria*.
- Spunta la casella di controllo *Opzioni di ricerca* per specificare quali elementi caricare dal vettore oppure usa **[Imposta filtro]** per avviare la finestra di dialogo *Costruttore interrogazioni*.
- Scegli il vettore che vuoi caricare dalla lista di quelli disponibili.
- Seleziona il vettore cliccando sul nome. Puoi selezionare più vettori tenendo premuto il tasto `Shift` mentre stai selezionando. Vedi la sezione [Costruttore di interrogazioni](#) per informazioni su come usare il Costruttore di interrogazioni PostgreSQL.
- Clicca su **[Aggiungi]** per aggiungere il vettore alla mappa.

Suggerimento: Vettori PostGIS

Solitamente un vettore PostGIS è definito da una voce nella tabella `geometry_columns`. Dalla versione 0.9.0 in poi, QGIS può caricare vettori che non hanno questa voce nella tabella `geometry_columns`. Questo vale sia per le tabelle che per le viste. Definire una vista spaziale fornisce un mezzo molto potente per visualizzare i tuoi dati. Fai riferimento al manuale di PostgreSQL per informazioni su come creare le viste.

Alcuni dettagli sui vettori PostgreSQL

Questa sezione contiene alcuni dettagli su come QGIS accede ai vettori PostgreSQL. La maggior parte delle volte QGIS dovrebbe semplicemente fornirti un elenco di tabelle del database che possono essere caricate e dovrebbe poi caricarle in base alla tua richiesta. Se hai difficoltà a caricare una tabella di PostgreSQL in QGIS, le informazioni seguenti ti possono aiutare a capire tutti i messaggi di QGIS e ti possono dare un'indicazione di come cambiare la tabella o le definizioni delle viste di PostgreSQL in modo che QGIS le possa caricare.

QGIS richiede che i vettori PostgreSQL contengano una colonna che possa essere usata come chiave univoca per il vettore. Le tabelle devono contenere una chiave primaria o una colonna con un vincolo univoco. Questa colonna deve essere di tipo `int4` (un numero intero di 4 byte). Alternativamente, la colonna `ctid` può essere usata come chiave primaria. Se a una tabella mancano queste informazioni, verrà usata la colonna `oid`. Le prestazioni saranno migliori se la colonna è indicizzata (le chiavi primarie sono indicizzate automaticamente in PostgreSQL).

Le viste di PostgreSQL hanno gli stessi requisiti di un vettore, ma non necessitano di chiavi primarie o colonne con vincoli univoci. Devi definire un campo della chiave primaria (deve essere un intero) nella finestra di dialogo di QGIS prima di caricare la vista. Se non c'è una colonna adatta nella vista, QGIS non caricherà il vettore. Se succede la soluzione è di modificare la vista in modo che contenga una colonna adatta (un intero e una chiave primaria con un vincolo univoco, preferibilmente indicizzato).

In QGIS è disponibile una casella di controllo **Scegli all'id**, attivata in modo predefinito. Questa opzione permette di ottenere gli id senza gli attributi, velocizzando il processo. Deseleziona questa opzione se usi viste molto grosse.

12.1.7 Importare dati in PostgreSQL

I dati possono essere importati in PostgreSQL/PostGIS usando diversi strumenti, come il plugin DB Manager e gli strumenti da riga di comando `shp2pgsql` e `ogr2ogr`.

DB Manager

QGIS ha un plugin nativo chiamato  **DB Manager**. Lo puoi usare per caricare shapefile e altri formati, e include il supporto per gli schemi. Vedi la sezione *Plugin DB Manager* per ulteriori informazioni.

shp2pgsql

Puoi usare lo strumento **shp2pgsql** di PostGIS per importare shapefile in un database PostGIS. Per esempio, per importare uno shapefile chiamato `lakes.shp` in un database PostgreSQL chiamato `gis_data`, usa il seguente comando:

```
shp2pgsql -s 2964 lakes.shp lakes_new | psql gis_data
```

Questo comando crea un nuovo vettore, chiamato `lakes_new`, nel database `gis_data`. Il nuovo vettore avrà un identificatore del sistema di riferimento (Spatial Reference Identifier - SRID) corrispondente a 2964. Vedi la sezione *Lavorare con le proiezioni* per ulteriori informazioni sui sistemi di riferimento spaziali e sulle proiezioni.

Suggerimento: Esportare dati da PostGIS

Come lo strumento di importazione **shp2pgsql**, esiste anche il comando che permette di esportare set di dati da PostGIS come shapefile: **pgsql2shp**. Lo strumento è incluso con la tua versione di PostGIS.

ogr2ogr

Oltre a **shp2pgsql** e **DB Manager** c'è un altro strumento per caricare dati in PostGIS: **ogr2ogr**. Questo comando fa parte dell'installazione di GDAL.

Per importare uno shapefile in PostGIS con **ogr2ogr** digita il seguente comando:

```
ogr2ogr -f "PostgreSQL" PG:"dbname=postgis host=myhost.de user=postgres
password=topsecret" alaska.shp
```

Questo comando importerà lo shapefile `alaska.shp` nel database PostGIS `postgis` usando come utente `postgres` e come password `topsecret` sull'host `myhost.de`.

Nota che OGR deve essere compilato con il supporto PostgreSQL per poter effettuare questa operazione. Puoi verificarlo digitando

```
ogrinfo --formats | grep -i post
```

Se volessi usare il comando interno di PostgreSQL **COPY** al posto del metodo predefinito **INSERT INTO**, devi impostare le variabili d'ambiente come segue (su piattaforme  e ):

```
export PG_USE_COPY=YES
```

ogr2ogr non crea indici spaziali come **shp2pgsql**. Devi crearli manualmente, usando il comando SQL **CREATE INDEX** dopo l'importazione, come passo aggiuntivo (sezione *Migliorare le prestazioni*).

Migliorare le prestazioni

Richiamare dati geografici da un database PostgreSQL può richiedere molto tempo, specialmente se il server è collegato alla rete. Puoi migliorare le prestazioni visualizzazione di vettore PostgreSQL creando un indice spaziale PostGIS `spatial index` per ogni vettore del database. PostGIS supporta la creazione di un indice GiST (indice dell'albero generalizzato di ricerca, Generalized Search Tree) per velocizzare le ricerche spaziali di dati (le informazioni su GiST fanno riferimento alla documentazione di PostGIS disponibile all'indirizzo <http://postgis.refrains.net>).

La sintassi per la creazione di un indice GiST è:

```
CREATE INDEX [indexname] ON [tablename]
  USING GIST ( [geometryfield] GIST_GEOMETRY_OPS );
```

Nota che per tabelle molto grandi, la creazione dell'indice può richiedere parecchio tempo. Non appena l'indice è stato creato, dovresti effettuare un `VACUUM ANALYZE`. Vedi la documentazione di PostGIS (POSTGIS-PROJECT *Letteratura e riferimenti web*) per ulteriori informazioni.

Segue un esempio di come creare un indice GiST:

```
gsherman@madison:~/current$ psql gis_data
Welcome to psql 8.3.0, the PostgreSQL interactive terminal.
```

```
Type: \copyright for distribution terms
      \h for help with SQL commands
      \? for help with psql commands
      \g or terminate with semicolon to execute query
      \q to quit
```

```
gis_data=# CREATE INDEX sidx_alaska_lakes ON alaska_lakes
gis_data=# USING GIST (the_geom GIST_GEOMETRY_OPS);
CREATE INDEX
gis_data=# VACUUM ANALYZE alaska_lakes;
VACUUM
```

```
gis_data=# \q
gsherman@madison:~/current$
```

12.1.8 Vettori a cavallo dei 180° di longitudine

Molti software GIS non gestiscono al meglio le mappe vettoriali, con sistema di riferimento geografico (lat/lon), a cavallo della linea di longitudine 180 degrees (http://postgis.refrations.net/documentation/manual-2.0/ST_Shift_Longitude.html). Se carichi una di queste mappe in QGIS vedrai aree geografiche distanti fra di loro anche se in realtà sono vicine. Nella figura [Figure_vector_4](#) il piccolo punto all'estrema sinistra della vista mappa (Chatham Islands) dovrebbe essere all'interno della griglia subito alla destra dell'isola principale della Nuova Zelanda.



Figura 12.5: Mappa in lat/lon a cavallo dei 180° di longitudine 🐧

Una soluzione consiste nel trasformare i valori di longitudine utilizzando PostGIS e la funzione **ST_Shift_Longitude**. Questa funzione legge i punti/vertici di ogni elemento di una geometria e se la coordinata di longitudine è $< 0^\circ$, aggiunge 360° . Il risultato sarà una versione $0^\circ - 360^\circ$ dei dati, che verranno poi visualizzati su una mappa centrata a 180° .

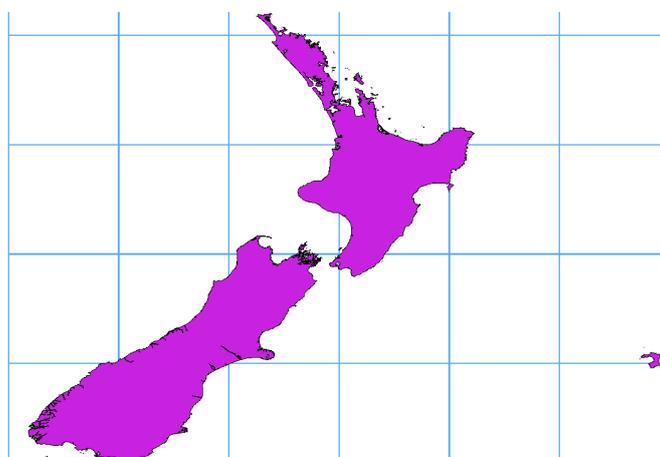


Figura 12.6: Vettori a cavallo di 180° di longitudine usando la funzione **ST_Shift_Longitude**

Guida all'uso

- Importare i dati in PostGIS (*Importare dati in PostgreSQL*) usando, per esempio, il plugin DB Manager.
- Usa l'interfaccia da linea di comando di PostGIS per dare il seguente comando (nell'esempio -"TABLE" è il nome della tua tabella PostGIS):
- Se tutto è andato a buon fine, riceverai la conferma sul numero di elementi che sono stati aggiornati. Potrai così caricare la mappa e vedere le differenze (figura [Figure_vector_5](#)).

12.1.9 Vettori SpatiaLite

Per caricare dei dati da un database SpatiaLite clicca sullo strumento  *Aggiungi un layer SpatiaLite* o seleziona l'opzione  *Aggiungi un layer SpatiaLite...* dal menu *Layer* oppure premi `Ctrl+Shift+L`. Si aprirà una finestra di dialogo che ti permetterà di accedere ai dati di un database SpatiaLite già connesso a QGIS oppure di definire la connessione ad un nuovo database. Per connettersi ad un nuovo database clicca su **[Nuovo]** e seleziona il database SpatiaLite, ovvero un file con l'estensione `.sqlite`.

Per salvare un vettore in formato SpatiaLite, clicca con il tasto destro del mouse sul vettore nella legenda e seleziona l'opzione *Salva con nome...*, scegli il nome del file in output, seleziona SQLite come formato e il SR che preferisci. Aggiungi anche `SPATIALITE=YES` nel riquadro Sorgente dati delle opzioni di creazione OGR. Vedi http://www.gdal.org/ogr/drv_sqlite.html per ulteriori informazioni.

QGIS supporta anche viste modificabili in SpatiaLite.

Creare un nuovo vettore SpatiaLite

Per creare un nuovo vettore SpatiaLite, fai riferimento alla sezione *Creare un nuovo layer SpatiaLite*.

Suggerimento: SpatiaLite data management plugin

Per la gestione dei dati SpatiaLite puoi usare diversi plugin: QSpatialite, SpatiaLite Manager o DB Manager (plugin nativo, raccomandato). Li puoi scaricare e installare con il gestore dei plugin.

12.1.10 Vettori Spatial MSSQL

QGIS fornisce anche supporto nativo a MS SQL 2008. La prima volta che carichi dai dati MSSQL, inizia cliccando il pulsante  *Aggiungi vettore MSSQL* oppure seleziona l'opzione  *Aggiungi vettore MSSQL...* dal menu *Layer* oppure ancora premi `Ctrl+Shift+M`.

12.1.11 Vettori Oracle Spatial

Le geometrie spaziali in Oracle Spatial ti aiutano a gestire e localizzare i dati geografici nativi all'interno di un database Oracle. QGIS supporta questi vettori.

Creare una connessione

La prima volta che usi Oracle Spatial come sorgente dati, devi creare una connessione al database che contiene i dati. Clicca sul pulsante  *Aggiungi vettore Oracle*, seleziona l'opzione  *Aggiungi vettore Oracle...* dal menu *Layer* oppure premi `Ctrl+Shift+O`. Per accedere al gestore delle connessioni clicca sul pulsante **[Nuovo]** per aprire la finestra di dialogo *Crea una nuova connessione Oracle*. I parametri richiesti per la connessione sono:

- **Nome:** Nome della connessione. Può essere uguale a quello del *Database*.
- **Database:** SID o SERVICE_NAME dell'istanza Oracle.
- **Host:** Nome del server che ospita il database. Deve essere un host con indirizzo raggiungibile, lo stesso che potrebbe essere usato per aprire una connessione telnet o per fare il ping all'host. Se il database è sullo stesso computer sul quale è installato QGIS, inserisci semplicemente *'localhost'*.
- **Port:** numero della porta del server PostgreSQL. La porta predefinita è 1521.
- **Nome utente:** Nome utente che accede al database.
- **Password:** Password usata dallo *Username* per collegarsi al database.

Come opzione, puoi spuntare le seguenti caselle di controllo:

- *Salva nome utente* specifica se vuoi salvare il nome utente del database nelle configurazioni di connessione.
- *Salva password* specifica se vuoi salvare la password del database nelle configurazioni di connessione.
- *Cerca solo nelle tabelle dei metadati* restringe le tabelle visualizzate a quello che sono presenti nella vista `all_sdo_geom_metadata`. Questo procedimento velocizza la visualizzazione iniziale delle tabelle spaziali.
- *Cerca solo tabelle dell'utente*: nella ricerca di tabelle spaziali, si limita a quelle di proprietà dell'utente.
- *Mostra anche tabelle senza geometria* specifica che anche le tabelle senza geometria devono essere elencate.
- *Usa i metadati stimati del vettore*: quando il vettore è stato impostato, la tabella Oracle richiede diversi metadati. Sono necessarie informazioni come il conteggio delle righe della tabella, il tipo di geometria e l'estensione spaziale nella colonna geometria. Se la tabella contiene un grande numero di righe che descrivono i metadati, stimare questi metadati porterà via molto tempo. Attivando questa opzione verranno eseguite le seguenti rapide operazioni sulla tabella dei metadati: Il conteggio delle righe è determinato da `all_tables.num_rows`. Le estensioni della tabella saranno sempre determinate con la funzione `SDO_TUNE.EXTENTS_OF` anche se è stato applicato un filtro al vettore.
- *Solo tipi di geometrie esistenti* elenca solo i tipi di geometria esistenti e non permettere di aggiungerne altre.

Quando hai impostato tutti i parametri, puoi testare la connessione cliccando sul pulsante **[Test Connessione]**.

Suggerimento: Impostazioni utente e sicurezza

A seconda delle impostazioni del tuo computer, la memorizzazione delle password nelle impostazioni di QGIS può essere un rischio per la sicurezza. Le password vengono salvate in testo leggibile nelle configurazioni di sistema e nel file di progetto! Le impostazioni personalizzate di QGIS sono salvate in modo diverso in base al sistema operativo:

-  le impostazioni sono salvate nella tua cartella home nel file `~/ .qgis2`.
-  le impostazioni sono salvate nel registro di sistema.

Caricare un vettore Oracle Spatial

Una volta che hai stabilito una o più connessioni, puoi caricare i vettori dal database Oracle. Ovviamente i dati devono essere memorizzati in Oracle.

Per caricare un vettore da Oracle Spatial, segui i seguenti passaggi:

- Se non hai ancora aperto la finestra *Aggiungi vettore Oracle Spatial*, clicca il pulsante 
 Aggiungi vettore Oracle Spatial
- Scegli la connessione dal menu a tendina e clicca su **[Connetti]**.
- Seleziona/deseleziona *Mostra anche le tabelle senza geometria*.
- Spunta la casella di controllo *Opzioni di ricerca* per specificare quali elementi caricare dal vettore oppure usa **[Imposta filtro]** per avviare la finestra di dialogo *Costruttore interrogazioni*.
- Scegli il vettore che vuoi caricare dalla lista di quelli disponibili.
- Seleziona il vettore cliccando sul nome. Puoi selezionare più vettori tenendo premuto il tasto `Shift` mentre stai selezionando. Vedi la sezione *Costruttore di interrogazioni* per informazioni su come usare il Costruttore di interrogazioni Oracle.
- Clicca su **[Aggiungi]** per aggiungere il vettore alla mappa.

Suggerimento: Vettori Oracle Spatial

Normalmente un vettore Oracle Spatial è definito con una voce nella tabella `USER_SDO_METADATA`.

12.2 The Symbol Library

12.2.1 Presentation

The Symbol Library is the place where users can create generic symbols to be used in several QGIS projects. It allows users to export and import symbols, groups symbols and add, edit and remove symbols. You can open it with the *Settings* → *Style Library* or from the **Style** tab in the vector layer's *Properties*.

Share and import symbols

Users can export and import symbols in two main formats: qml (QGIS format) and SLD (OGC standard). Note that SLD format is not fully supported by QGIS.



share item displays a drop down list to let the user import or export symbols.

Groups and smart groups

Groups are categories of Symbols and smart groups are dynamic groups.

To create a group, right-click on an existing group or on the main **Groups** directory in the left of the library. You can also select a group and click on the  **add item** button.

To add a symbol into a group, you can either right click on a symbol then choose *Apply group* and then the group name added before. There is a second way to add several symbols into group: just select a group and click  and choose **Group Symbols**. All symbols display a checkbox that allow you to add the symbol into the selected groups. When finished, you can click on the same button, and choose **Finish Grouping**.

Create **Smart Symbols** is similar to creating group, but instead select **Smart Groups**. The dialog box allow user to choose the expression to select symbols in order to appear in the smart group (contains some tags, member of a group, have a string in its name, etc.)

Add, edit, remove symbol

With the *Style manager* from the **[Symbol]**  menu you can manage your symbols. You can  **add item**,  **edit item**,  **remove item** and  **share item**. 'Marker' symbols, 'Line' symbols, 'Fill' patterns and 'colour ramps' can be used to create the symbols. The symbols are then assigned to 'All Symbols', 'Groups' or 'Smart groups'.

For each kind of symbols, you will find always the same dialog structure:

- at the top left side a symbol representation
- under the symbol representation the symbol tree show the symbol layers
- at the right you can setup some parameter (unit,transparency, color, size and rotation)
- under these parameteres you find some symbol from the symbol library

The symbol tree allow adding, removing or protect new simple symbol. You can move up or down the symbol layer.

More detailed settings can be made when clicking on the second level in the *Symbol layers* dialog. You can define *Symbol layers* that are combined afterwards. A symbol can consist of several *Symbol layers*. Settings will be shown later in this chapter.

Suggerimento: Note that once you have set the size in the lower levels of the *Symbol layers* dialog, the size of the whole symbol can be changed with the *Size* menu in the first level again. The size of the lower levels changes accordingly, while the size ratio is maintained.

12.2.2 Marker Symbols

Marker symbols have several symbol layer types:

- Ellipse marker
- Font marker
- Simple marker (default)
- SVG marker
- Vector Field marker

The following settings are possible:

- *Symbol layer type*: You have the option to use Ellipse markers, Font markers, Simple markers, SVG markers and Vector Field markers.
- *colors*
- *Size*
- *Outline style*
- *Outline width*
- *Angle*
- *Offset X,Y*: You can shift the symbol in the x- or y-direction.
- *Anchor point*
- *Data defined properties ...*

12.2.3 Line Symbols

Line marker symbols have only two symbol layer types:

- Marker line
- Simple line (default)

The default symbol layer type draws a simple line whereas the other display a marker point regularly on the line. You can choose different location vertex, interval or central point. Marker line can have offset along the line or offset line. Finally, *rotation* allows you to change the orientation of the symbol.

The following settings are possible:

- *colour*
- *Pen width*
- *Offset*
- *Pen style*

- *Join style*
- *Cap style*
- *Use custom dash pattern*
- *Dash pattern unit*
- *Data defined properties ...*

12.2.4 Polygon Symbols

Polygon marker symbols have also several symbol layer types:

- Centroid fill
- Gradient fill
- Line pattern fill
- Point pattern fill
- SVG fill
- Shapeburst fill
- Simple fill (default)
- Outline: Marker line (same as line marker)
- Outline: simple line (same as line marker)

The following settings are possible:

- *Colors* for the border and the fill.
- *Fill style*
- *Border style*
- *Border width*
- *Offset X,Y*
- *Data defined properties ...*

Using the color combo box, you can drag and drop color for one color button to another button, copy-paste color, pick color from somewhere, choose a color from the palette or from recent or standard color. The combo box allow you to fill in the feature with transparency. You can also just clic on the button to open the palette dialog. Note that you can import color from some external software like GIMP.

‘Gradient Fill’ *Symbol layer type* allows you to select between a *Two color* and *Color ramp* setting. You can use the *Feature centroid as Referencepoint*. All fills ‘Gradient Fill’ *Symbol layer type* is also available through the *Symbol* menu of the Categorized and Graduated Renderer and through the *Rule properties* menu of the Rule-based renderer. Other possibility is to choose a ‘shapeburst fill’ which is a buffered gradient fill, where a gradient is drawn from the boundary of a polygon towards the polygon’s centre. Configurable parameters include distance from the boundary to shade, use of color ramps or simple two color gradients, optional blurring of the fill and offsets.

It is possible to only draw polygon borders inside the polygon. Using ‘Outline: Simple line’ select *Draw line only inside polygon*.

12.2.5 Color ramp

You can create a custom color ramp choosing *New color ramp...* from the *color ramp* drop-down menu. A dialog will prompt for the ramp type: Gradient, Random, colorBrewer, or cpt-city. The first three have options for number

of steps and/or multiple stops in the color ramp. You can use the  *Invert* option while classifying the data with a color ramp. See [figure_symbology_3](#) for an example of custom color ramp and [figure_symbology_3a](#) for the cpt-city dialog.

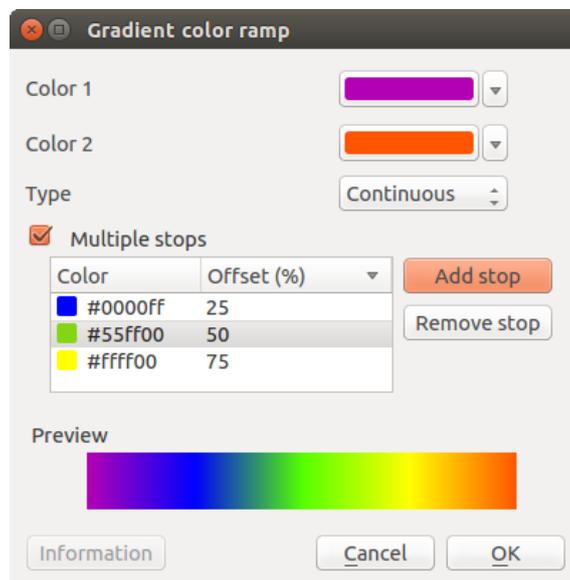


Figura 12.7: Example of custom gradient color ramp with multiple stops 

The cpt-city option opens a new dialog with hundreds of themes included 'out of the box'.

12.3 Proprietà dei vettori

La finestra di dialogo *Proprietà layer* fornisce informazioni sul vettore, sulla simbologia e sulle opzioni di visualizzazione delle etichette. Inoltre, se hai caricato il vettore da un database PostgreSQL/PostGIS, puoi modificare l'espressione SQL che hai usato per caricarlo, grazie al *Costruttore di interrogazioni* nel menu *Generale*. Per accedere alla finestra di dialogo *Proprietà layer*, fai doppio click sul vettore nella legenda o clicca con il tasto destro sul vettore e seleziona *Proprietà* dal menu contestuale.

12.3.1 Menu Stile

Con il menu *Stile* puoi modificare lo stile e la visualizzazione dei vettori. Puoi usare sia lo strumento *Visualizzazione del layer* →, comune a tutti i tipi di vettori, sia simbologie specifiche in funzione del tipo di vettore.

Tipi di visualizzazioni

Il visualizzatore è responsabile del disegno di una geometria con il simbolo corretto. Ci sono quattro tipi di visualizzazione: simbolo singolo, categorizzato, graduato e tramite regole. Non c'è visualizzatore di colore continuo, perché è solo un caso particolare dello stile graduato. I visualizzatori categorizzato e graduato possono essere creati specificando un simbolo e una scala di colori - i colori per i simboli saranno realizzati in modo appropriato. Per i vettori puntuali, c'è disponibile un visualizzatore spostamento punto. Per ogni tipo di vettori (punti, linee e poligoni), sono disponibili tipi di simboli. A seconda del visualizzatore selezionato, la *Style* offre diverse sezioni aggiuntive. In basso a destra della finestra di simbologia, c'è un pulsante **** [Apri libreria] ****, che dà accesso al Gestore di stili (vedi: ref: *vector_style_manager*). Il gestore di stili consente di modificare e rimuovere i simboli esistenti e aggiungerne di nuovi.

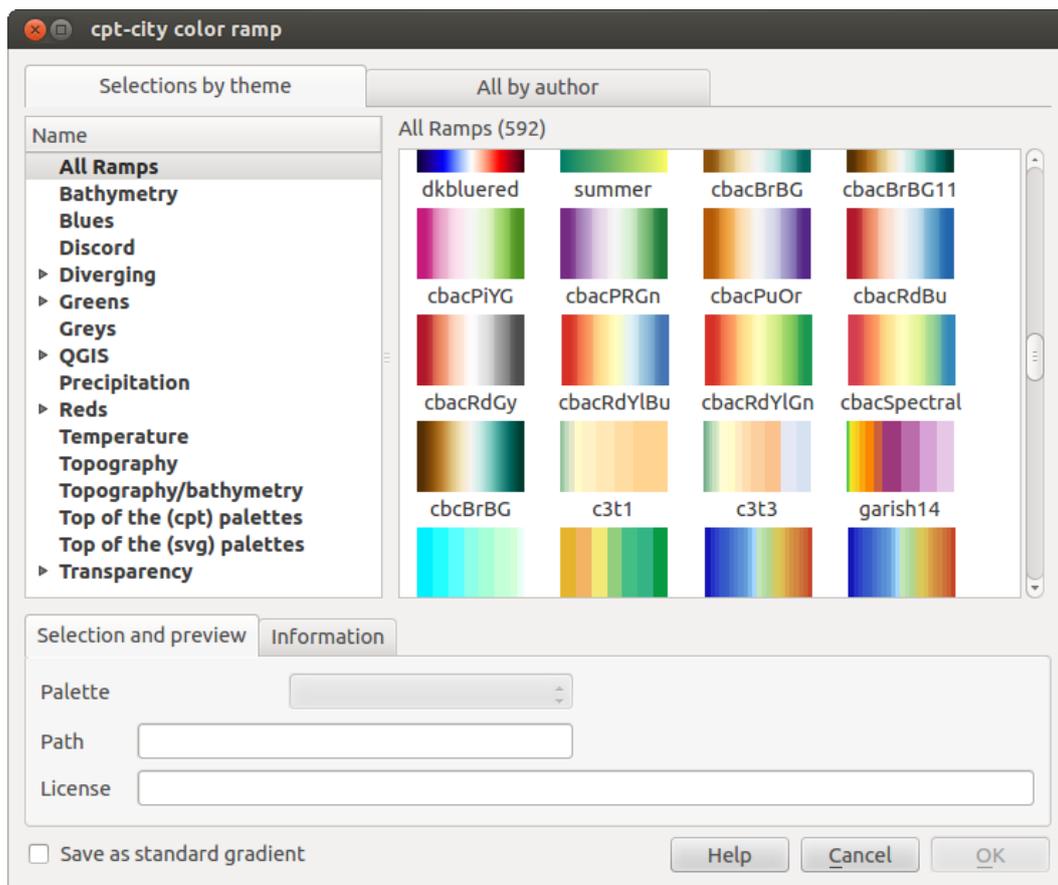


Figura 12.8: cpt-city dialog with hundreds of color ramps 🐧

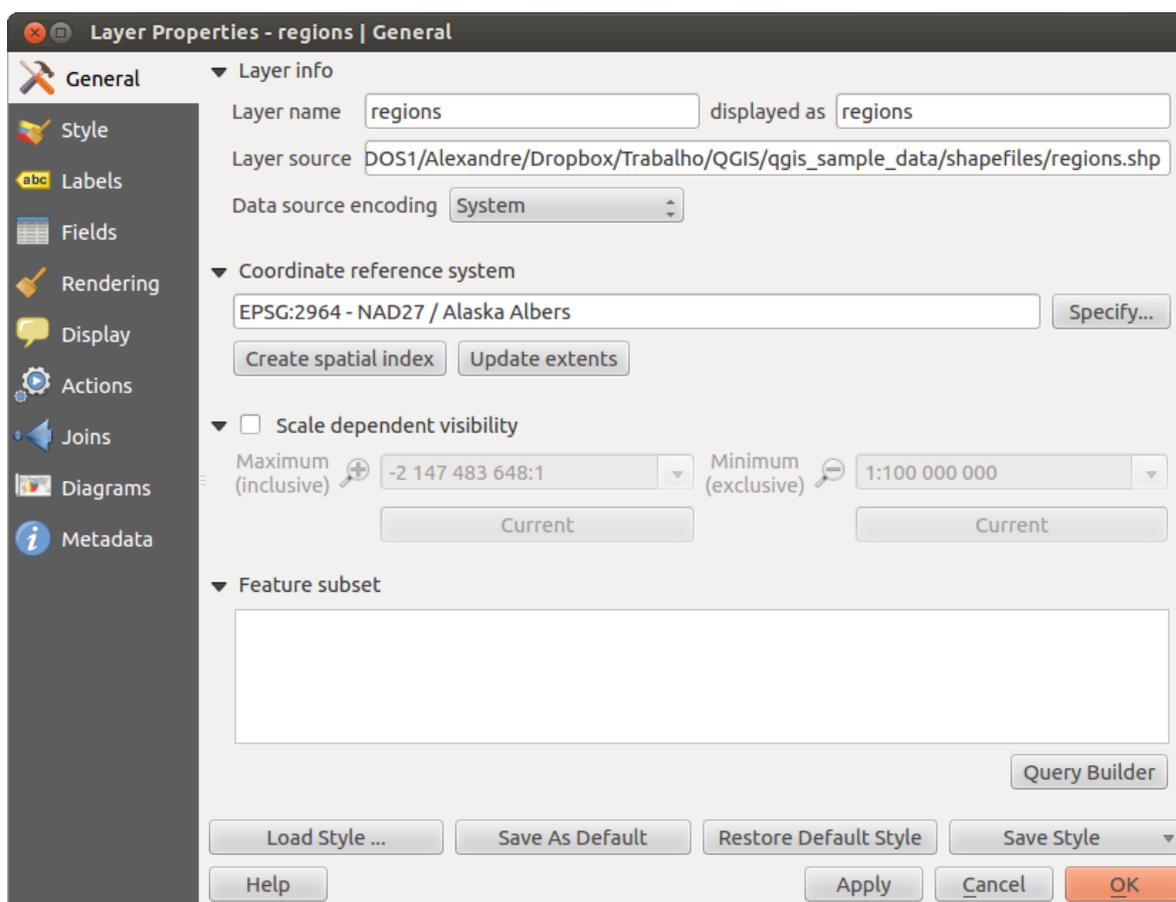


Figura 12.9: Finestra di dialogo delle proprietà del vettore 🐧

Dopo aver fatto le necessarie modifiche, il simbolo nella finestra di dialogo Selettore simbolo può essere aggiunto alla lista dei simboli (utilizzando ** [Symbol] **  : guilabel: *Salva*), e quindi può essere facilmente utilizzato in futuro. Inoltre, puoi utilizzare il pulsante [Salva stile]  per salvare il simbolo come file QGIS di stile (.qml) o un file SLD (.sld). SLD può essere esportata da qualsiasi tipo di visualizzatore - simbolo singolo, categorizzato o graduato o tramite regole si è laureato o basata su regole - ma se imèporti un file SLD, viene creato uno stile di simbolo singolo o di tramite regole. Ciò significa che gli stili categorizzati o graduati vengono convertiti in tramite regole. Se si desidera mantenere gli stili, è necessario attenersi al formato QML. D'altra parte, può essere molto utile a volte avere questo modo semplice di convertire stili tramite regole.

Se cambi il tipo di visualizzatore mentre imposti lo stile di un vettore, le impostazioni effettuate per il simbolo saranno mantenute. Questo funziona solo per un cambiamento. Se si ripete la modifica del tipo di visualizzatore le impostazioni per il simbolo saranno perse.

Se il layer è in un database (PostGIS o Spatialite per esempio), è possibile salvare lo stile di livello all'interno di una tabella del database. Clic sulla casella combinata :guilabel: *Salva stile* e scegli la voce **Salva nel database**, quindi compila la finestra di dialogo per definire un nome di stile, aggiungi una descrizione, un file ui e l'opzione se lo stile è uno stile predefinito. Quando carichi un layer dal database, se uno stile esiste già QGIS lo caricherà con il suo stile. Puoi aggiungere più stile nel database. Solo uno sarà comunque lo stile predefinito.

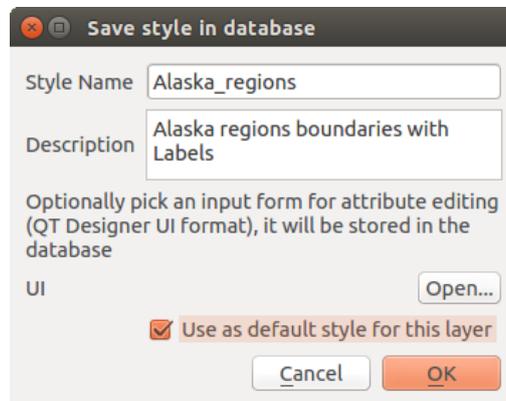


Figura 12.10: Finestra di dialogo Salva nel database 

Suggerimento: Selezionare e cambiare simboli multipli

La Simbologia ti permette di scegliere simboli multipli e di cliccare con il tasto destro per cambiare il colore, la trasparenza, la dimensione e lo spessore di quello che hai selezionato.

Visualizzatore Simbolo singolo

Usa il visualizzatore Simbolo singolo per visualizzare tutti gli elementi di un vettore usando un simbolo singolo predefinito. Le proprietà, che puoi modificare nel menu *Stile*, dipendono parzialmente dal tipo di vettore, ma tutti i vettori condividono la seguente struttura. Nella parte superiore sinistra del menu puoi vedere l'anteprima del simbolo attuale che hai scelto. Nella parte destra del menu vedrai un elenco di simboli già definiti che puoi selezionare semplicemente cliccandoci sopra. Puoi modificare il simbolo che hai scelto usando il menu nella parte destra. Se clicchi sul primo livello nella parte sinistra puoi definire i parametri come *Unità*, *Trasparenza*, *Colore* and *Rotazione*.

Visualizzatore Categorizzato

Il visualizzatore Categorizzato viene usato per visualizzare tutti gli elementi di un vettore, dove il colore di un simbolo singolo riflette i valori di un determinato attributo. Il menu *Stile* ti permette di scegliere:

- L'attributo (usa la lista Colonna o l'espressione ... *Imposta espressione*, vedi *Expressions*)
- Il simbolo (Simbolo)
- I colori (usando la lista gradiente di colore)

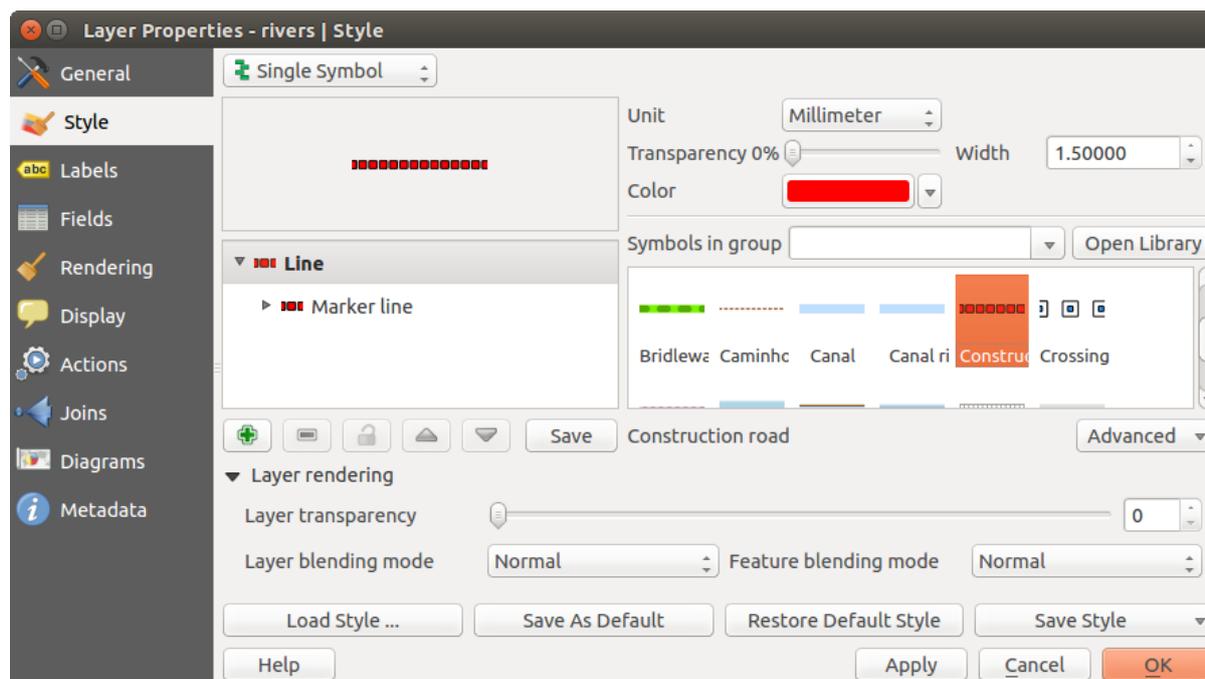


Figura 12.11: Finestra di proprietà del simbolo singolo 🐧

Poi click sul pulsante **Classifica** per creare la classi dei distinti valori degli attributi colonna. Ogni classe può essere disabilitata spuntando la casella a sinistra del nome della classe.

Puoi cambiare il simbolo, il valore e/o l'etichetta, basta fare doppio click sulla voce che si desidera modificare.

IL tasto destro mostra il menu relativo con **Copia/Incolla**, **Cambia Colore**, **Cambiamento trasparenza**, **Cambia unità di output**.

Il pulsante [**Avanzato**] nell'angolo inferiore destro nella finestra di dialogo ti permette di impostare i campi che contengono i valori di dimensione e rotazione. Per comodità, la parte centrale del menu elenca tutti i valori attualmente selezionati, inclusi i simboli che verranno visualizzati.

L'esempio in figura [figure_symbology_2](#) mostra la finestra di dialogo della visualizzazione categorizzata del vettore rivers dell'insieme di dati campione di QGIS.

Visualizzatore Graduato

Il visualizzatore Graduato visualizza tutti gli elementi del vettore tramite un unico simbolo. I colori riflettono la classificazione in classi dell'attributo scelto.

Come il Visualizzatore Categorizzato, quello Graduato ti permette di impostare la rotazione e la dimensione della scala in base a valori presenti in colonne specifiche.

Ugualmente al visualizzatore Categorizzato, la scheda *Stile* ti permette di selezionare:

- L'attributo (usando il menu a tendina per scegliere la colonna oppure la funzione **E... Imposta espressione per la colonna**, vedi il capitolo [ref:vector_expressions](#))
- Il simbolo (Simbolo)
- I colori (usando la lista gradiente di colore)

Inoltre, puoi specificare il numero di classi e il modo di classificare gli elementi. Hai a disposizione i seguenti tipi di classificazione:

- Intervallo uguale: ogni classe ha la stessa dimensione (per esempio valori da 0 a 16 e quattro classi, ogni classe ha dimensione 4)
- Quantile: ogni classe avrà lo stesso numero di elementi (l'idea di un diagramma a scatola e baffi)
- Natural Breaks (Jenks): la varianza all'interno di ogni classe è minimo, mentre quella tra le classi è massima;

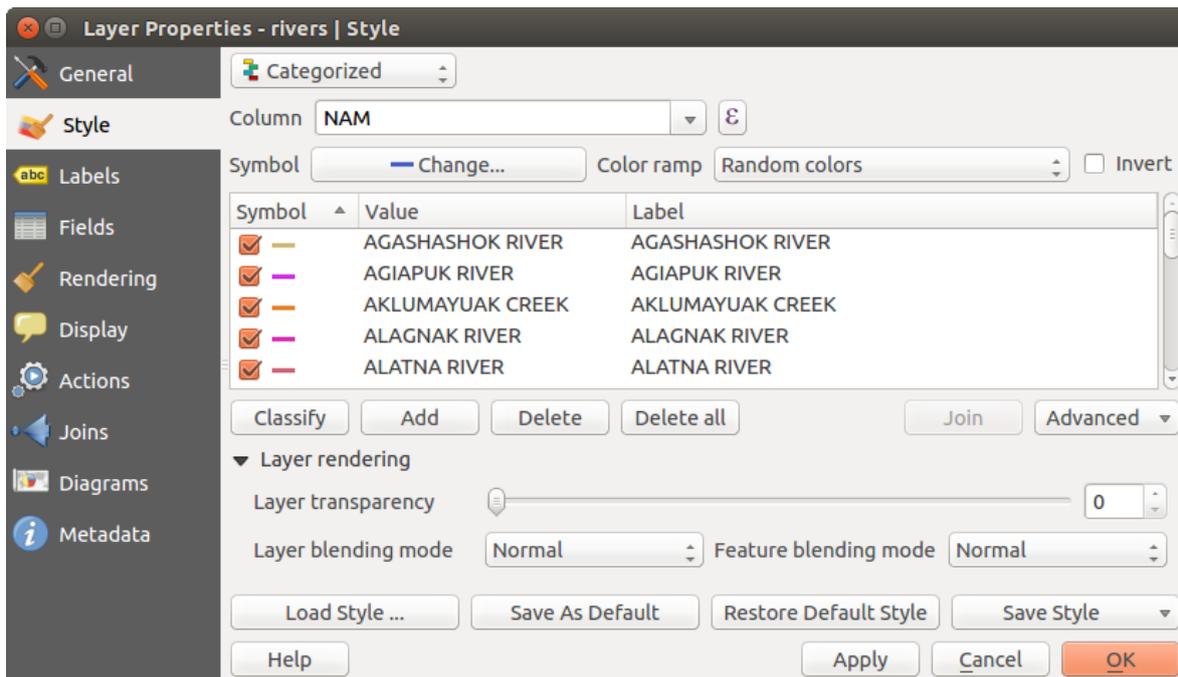


Figura 12.12: Finestra di dialogo simbologia categorizzata 🐧

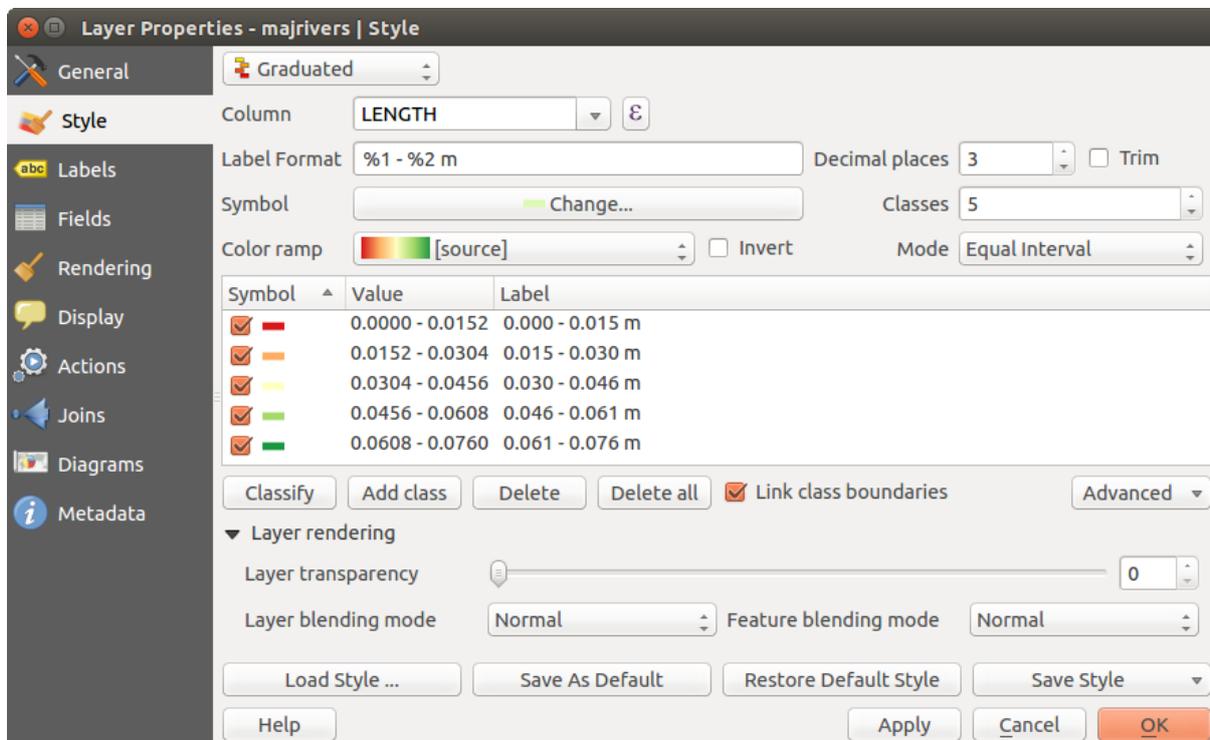


Figura 12.13: Finestra di dialogo simbologia graduata 🐧

- Deviazione standard: le classi sono costruite in funzione della deviazione standard dei valori;
- Pretty Breaks: lo stesso del natural breaks ma i numeri estremi di ogni classe sono interi

Nella parte centrale della finestra *Stile*, puoi vedere tutte i valori e le classi di suddivisione insieme ai loro intervalli, etichette e simboli.

Poi click sul pulsante **Classifica** per creare la classi usando il metodo scelto. Ogni classe può essere disabilitata spuntando la casella a sinistra del nome della classe.

Puoi cambiare il simbolo, il valore e/o l'etichetta, basta fare doppio click sulla voce che si desidera modificare.

IL tasto destro mostra il menu relativo con **Copia/Incolla**, **Cambia Colore**, **Cambiamento trasparenza**, **Cambia unità di output**.

L'esempio in figura [figure_symbology_4](#) mostra la finestra di dialogo della visualizzazione categorizzata del vettore rivers dell'insieme di dati campione di QGIS.

Suggerimento: Mappe tematiche usando un'espressione

Puoi creare mappe tematiche categorizzate o graduate usando il risultato di un'espressione. Nella finestra di dialogo delle proprietà del vettore puoi selezionare su  *Imposta espressione della colonna*. Non hai quindi più bisogno di creare una nuova colonna nella tabella degli attributi di un vettore e sceglierla in un secondo momento per la categorizzazione.

Visualizzatore Tramite regole

Il visualizzatore Tramite regole visualizza tutti gli elementi di un vettore tramite simboli basati su regole, con i colori che riflettono la classificazione di un attributo di interesse. Le regole si basano su istruzioni SQL, che puoi creare con il Costruttore stringhe. Puoi creare dei raggruppamenti attraverso filtri o scale e puoi decidere se vuoi abilitare i livelli del simbolo oppure se far corrispondere la prima regola.

L'esempio in figura [figure_symbology_5](#) mostra la finestra di dialogo della visualizzazione tramite regole del vettore rivers dell'insieme di dati campione di QGIS.

Per creare una regola, attivare una riga esistente facendo doppio clic su di esso, o fare clic su '+' e fare clic sulla nuova regola. Nella finestra di dialogo :guilabel: *Proprietà delle regole*, è possibile definire un'etichetta per la regola. Premere il pulsante  per aprire il Costruttore stringhe espressione. Nella **lista delle funzioni**, clicca su: :guilabel: *Campi e valori* per visualizzare tutti gli attributi della tabella da cercare. Per aggiungere un attributo al campo **Espressione**, fare doppio clic sul suo nome nella lista :guilabel: *Campi e valori*. In generale, puoi utilizzare i campi, i valori e le funzioni per costruire l'espressione di calcolo, o si può semplicemente digitare nella casella (vedi: ref: *vector_expressions*). Puoi creare una nuova regola copiando e incollando una regola esistente con il tasto destro del mouse. Puoi inoltre utilizzare la regola 'ELSE' se nessuna delle altre viene utilizzata. Da QGIS 2.6 l'etichetta per le regole appare nella legenda. Basta fare doppio click le regole nella legenda e il menu *Stile* delle proprietà appare mostrando la regola che fa da sfondo il simbolo nella legenda.

Visualizzatore Spostamento punto

Il visualizzatore Spostamento punto ti permette di visualizzare gli elementi di un vettore di punti anche se questi hanno la stessa posizione. I simboli vengono posizionati lungo un cerchio di spostamento intorno al centro del simbolo.

Suggerimento: Esporta simbologia vettore

Hai la possibilità di esportare la simbologia del vettore da QGIS in file Google * kml, * dxf e MapInfo * .tab. Apri con il tasto destro del mouse sul layer e click su: MenuSelection: *Salva come* -> per specificare il nome del file di output e il suo formato. Nella finestra di dialogo, utilizzare il menù :guilabel: *Esporta Simbologia* per salvare la simbologia sia come :MenuSelection: *Simbologia elementi* -> o come :MenuSelection: *Simbologia simbolo del vettore*->. Se hai usato i livelli di simboli, si consiglia di utilizzare la seconda impostazione.

Poligoni invertiti

Il visualizzatore Poligoni invertiti ti consente di definire un simbolo per riempire fuori del poligoni del vettore. Puoi selezionare un sotto visualizzatore. Questi sotto visualizzatori sono gli stessi di quello principale.

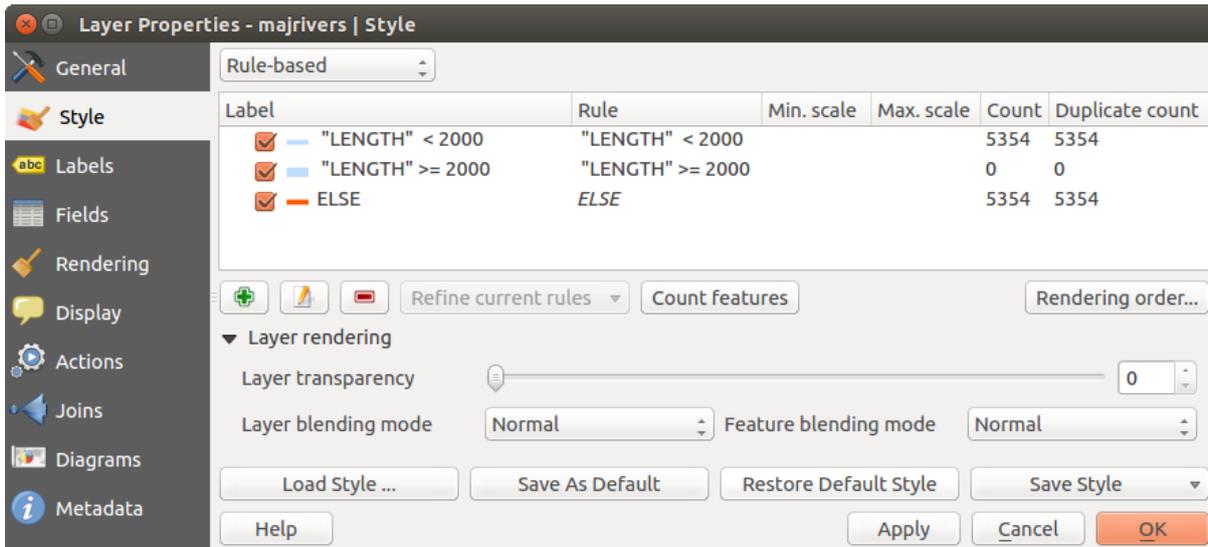


Figura 12.14: Finestra di dialogo simbologia basata su regole 🐧

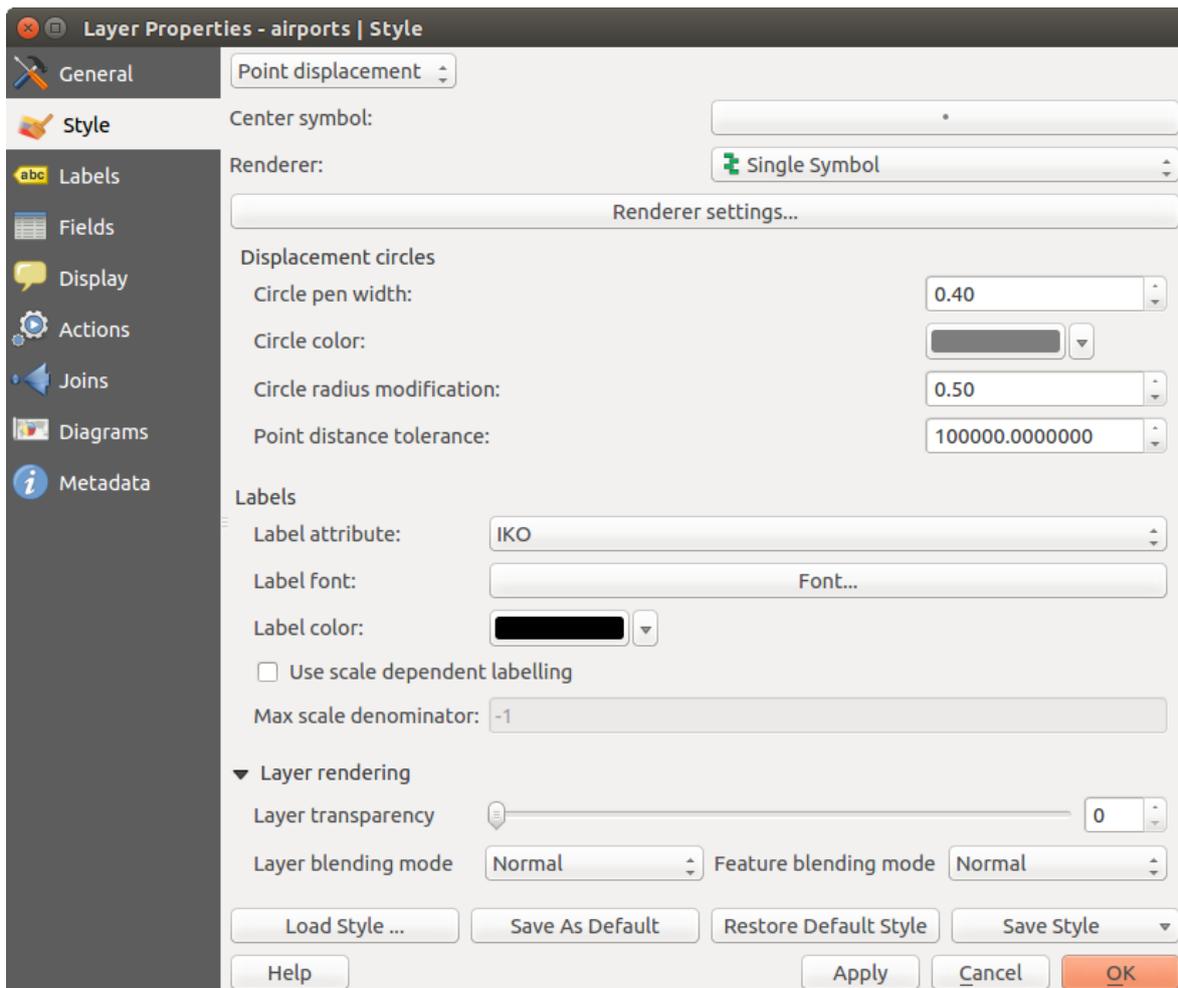


Figura 12.15: Finestra di dialogo simbologia spostamento punto 🐧

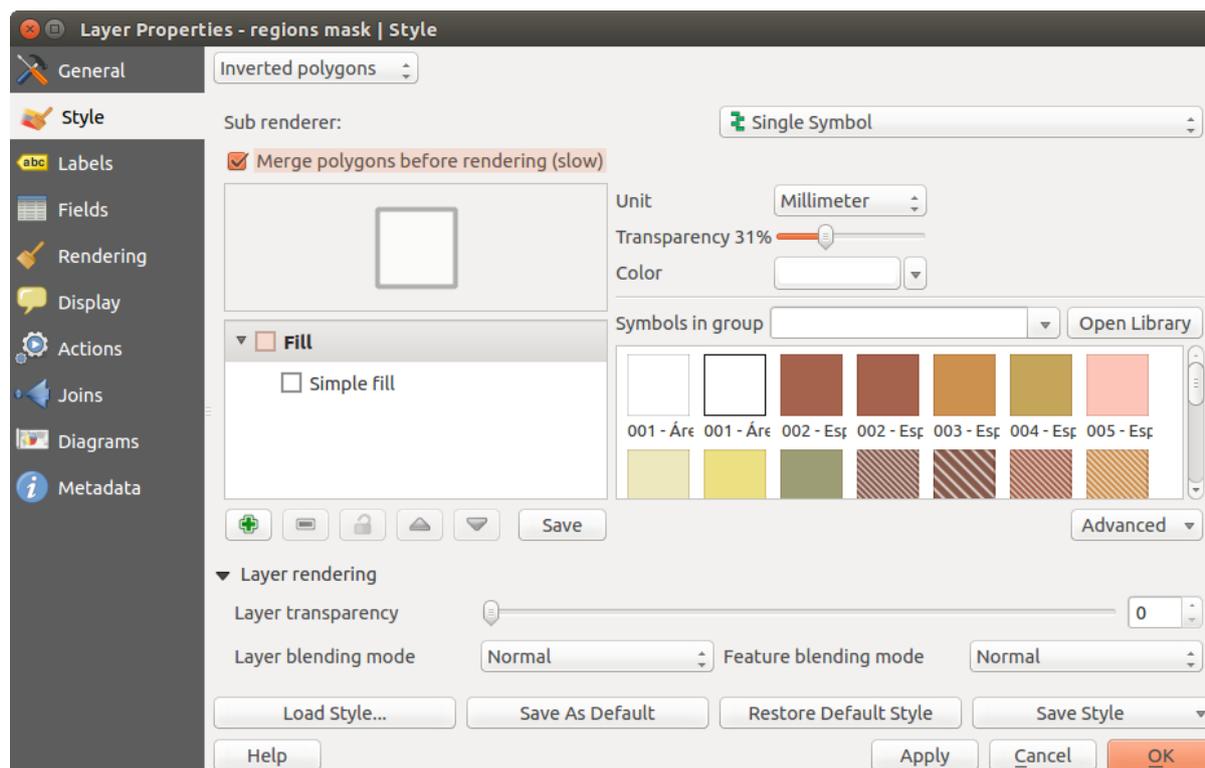


Figura 12.16: Finestra di dialogo Poligoni invertiti 

Selettore di colore

Indipendentemente dal tipo di stile da utilizzare, la finestra di dialogo `:guilabel: Scegli colore` si mostrerà quando fai clic per scegliere un colore - o bordo o colore di riempimento. Questa finestra ha quattro diverse schede che consentono di selezionare i colori:  scala di colori,  ruota colori,  campioni di colori or  `:sup:selettore di colori`.

Qualunque sia il metodo utilizzato, il colore selezionato viene sempre descritto nei cursori dei colori per i valori HSV (Hue, Saturation, Value) e RGB (rosso, verde, blu). C'è anche un cursore `:guilabel: opacità` per impostare il livello di trasparenza. Nella parte in basso a sinistra della finestra di dialogo si può vedere un confronto tra il colore `:guilabel: Vecchio` e quello `:guilabel: Attuale` e nella parte in basso a destra si ha la possibilità di aggiungere il colore che appena ottimizzato in un pulsante di archivio colore.

Con  `:sup: scala di colori` o con  `:sup: ruota di colori`, è possibile individuare tutte le possibili combinazioni di colori. Ci sono altre possibilità però. Usando *campioni di colore*  si può scegliere da un elenco preselezionato. Questo elenco preselezionato viene popolato con uno dei tre metodi: `:guilabel: colori recenti`, `:guilabel: colori standard` o `:guilabel: colori di progetto`.

Un'altra opzione è quella di utilizzare il  `: sup: selettore di colore` che permette di campionare un colore da sotto il puntatore del mouse in qualsiasi parte di QGIS o anche da un'altra applicazione premendo la barra spaziatrice. Da notare che il selettore colore dipende dal sistema operativo e non è attualmente supportato da OSX.

Suggerimento: selettore rapido di colore + copia/incolla colori

Puoi rapidamente scegliere da *colori recenti*, da *colori attuali* or semplicemente *copia* o *incolla* un colore facendo click sulla freccia a discesa di un pulsante colore.

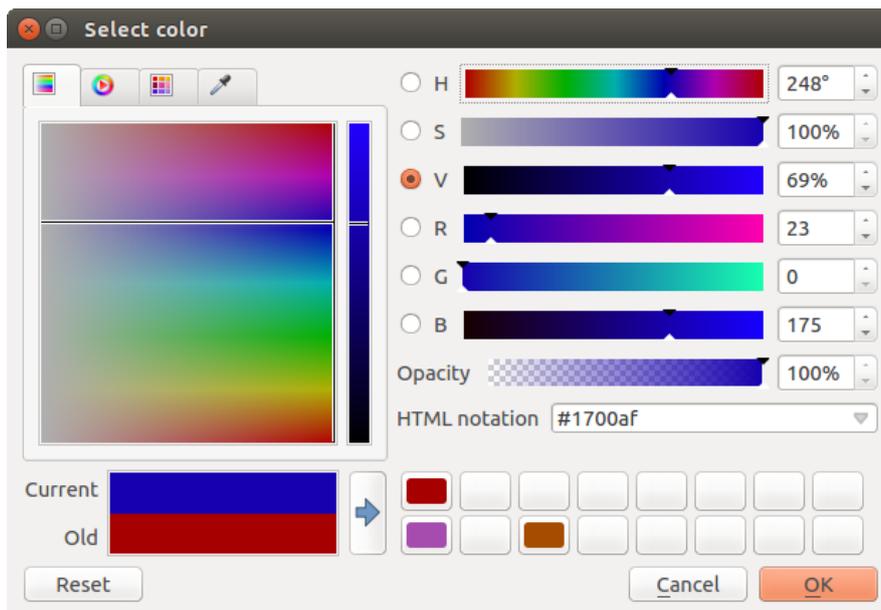


Figura 12.17: Scheda selettore da scala di colore 🐧

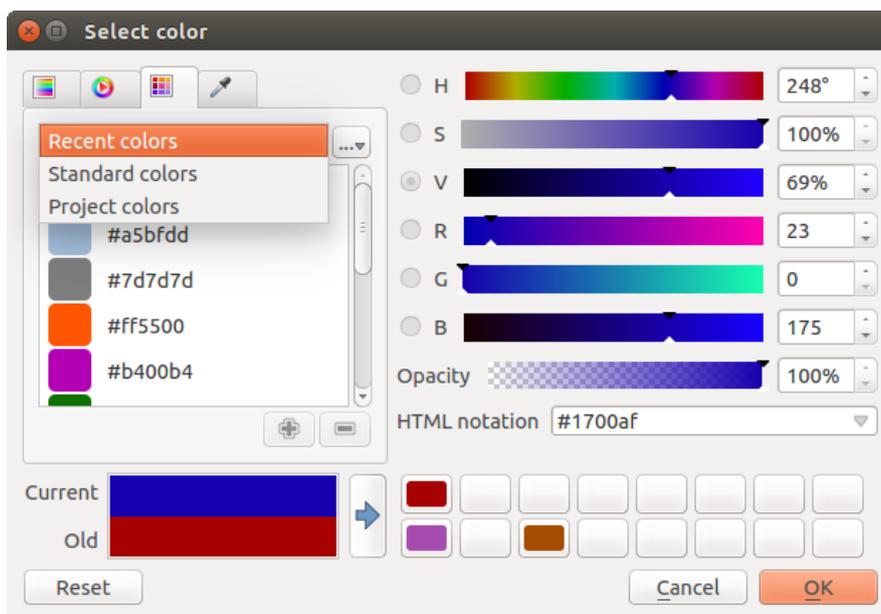


Figura 12.18: Scheda selettore da campioni di colore 🐧

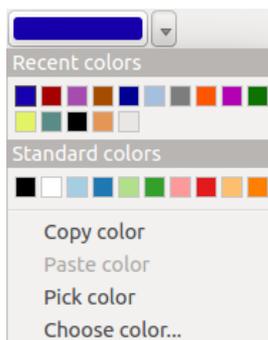


Figura 12.19: Menu del selettore rapido di colore 🐧

Visualizzazione del layer

- *Trasparenza del layer* : con questo strumento puoi decidere il grado di visibilità del vettore. Usa questo cursore per adattare la trasparenza del vettore. Puoi anche impostare un valore preciso di trasparenza nella casella presente a destra del cursore.
- *Modalità fusione layer e Modalità fusione elementi*: grazie a questi strumenti usati soprattutto in ambito grafico, potrai creare effettivi visivi speciali. I pixel del vettore più in alto vengono mischiati con i pixel di tutti i layer sottostanti. Hai molte scelte diverse.
 - Normale: è la modalità fusione predefinita che usa il canale alpha del pixel più in alto fondendolo con quello sotto. I colori non sono quindi mescolati.
 - Schiarisci: seleziona il valore massimo di ogni componente dal basso verso l'alto. Il risultato può apparire frastagliato e duro.
 - Scolorisci: i pixel chiari provenienti dal vettore sorgente vengono dipinti sopra la destinazione, mentre i pixel più scuri no. Questa modalità è molto utile per mescolare le trame di un vettore con un altro (per esempio un raster di ombreggiatura con un altro layer).
 - Scherma: questa modalità schiarirà e saturerà i pixel sottostanti in base a quanto sono chiari i pixel di sopra. In questo modo, i pixel più chiari in cima aumenteranno la saturazione e schiariranno i pixel sottostanti. Otterrai il miglior risultato se i pixel in cima non sono troppo chiari, altrimenti l'effetto sarà troppo estremo.
 - Addizione: questa modalità addiziona semplicemente il valore dei pixel di un vettore con i valori dei layer sottostanti. Se i valori sono maggiori di uno (ovvero quando si lavora con bande RGB) verrà mostrato il bianco. Questa modalità è ottima per evidenziare dei particolari.
 - Scurisci: il pixel finale conserva il valore minore dei pixel del layer in cima e in fondo. Come la modalità schiarisci, il risultato tende a essere frastagliato e duro.
 - Moltiplica: qui, il valore di ogni pixel del layer in cima viene moltiplicato per il valore dei pixel corrispondenti di tutti i layer sottostanti. Il risultato tende quindi a essere piuttosto scuro.
 - Brucia: i colori più scuri del layer in cima scuriranno i layer sottostanti. Questa modalità è utile per aggiustare e colorare i layer sottostanti.
 - Sovrapponi: è una combinazione delle modalità moltiplica e scolorisci. Le parti chiare risulteranno ancora più chiare e quelle scure ancora più scure.
 - Luce diffusa: molto simile alla modalità sovrapponi, ma invece di combinare le modalità moltiplica/scolorisci, combina brucia/scherma. Il risultato è una luce chiara e luccicante su tutta l'immagine.
 - Luce intensa: anche questa modalità è simile alla modalità sovrapponi. Proietta una luce molto intensa su tutta l'immagine.
 - Differenza: vengono sottratti i pixel in cima da quelli sul fondo, oppure al contrario, ma in modo da ottenere sempre valori positivi. Questa modalità non ha effetti con il nero, perché la differenza di questo colore con tutti gli altri è sempre zero.
 - Sottrai: questa modalità sottrae semplicemente i valori di un pixel dagli altri pixel. Se il valore dovesse essere negativo verrà visualizzato il nero.

12.3.2 Menu Etichette

Lo strumento  Etichette ti dà la possibilità di etichettare in modo facile e intelligente punti, linee e poligoni specificando solamente pochi parametri. È supportata anche la riproiezione al volo. Abbiamo ridisegnato l'intera applicazione per le etichette, aggiungendo nuove caratteristiche e altre funzionalità. Hai a disposizione questi menu:

- Testo

- Formattazione
- Contorno
- Sfondo
- Ombra
- Posizionamento
- Visualizzazione

Ecco come puoi usare il nuovo menu delle etichette con varie tipologie di vettori. **Etichettare vettore di punti**

Avvia QGIS e carica un vettore di punti. Attiva il vettore nella legenda e clicca sull'icona  Opzioni per le etichette dei layer nella barra degli strumenti.

Il primo passo è quello di spuntare la casella di controllo  *Etichetta questo vettore con* e selezionare un attributo fra quelli disponibili. Clicca su  se vuoi definire un'espressione per le etichette - vedi [labeling_with_expressions](#).

I seguenti passi descrivono un'etichettatura semplice, senza usare la funzione di *Sovrascrittura definita dai dati*, situata vicino ai menu a tendina.

Puoi specificare lo stile del testo dal menu *Testo* (vedi [Figure_labels_1](#)). Usa l'opzione *Tipo maiuscolo* per cambiare lo stile del testo. Puoi scegliere fra: 'Tutto maiuscolo', 'Tutto minuscolo' e 'Prima lettera maiuscola'. Puoi anche usare le modalità di fusione per ricreare alcuni effetti di programmi di grafica (vedi [blend_modes](#)).

Nel menu *Formattazione* puoi scegliere un carattere per andare a capo grazie alla funzione 'A capo con il carattere'. Usa l'opzione  *Numeri formattati* per formattare i numeri della tabella degli attributi. Qui puoi inserire la parte decimale. In modo predefinito, se attivi questa opzione, vengono impostati tre decimali.

Per creare un contorno, attiva la casella di controllo  *Disegna contorno del testo* nel menu *Contorno*. Puoi scegliere sia il colore del contorno che la modalità di fusione (vedi [blend_modes](#)).

Se hai attivato la casella  :guilabel: 'colore riempimento del contorno, allora interagirà con il testo, se parzialmente trasparente, e darà risultati di trasparenza del colore misti. Se spegni il riempimento del contorno, ridurrai i colori misti (salvo dove il contorno si interseca con il riempimento del testo), e permetti anche di rendere il testo delineato.

Nel menu *Sfondo*, tramite *Dimensione X* e *Dimensione Y* puoi ridimensionare lo sfondo. Usa *Tipo dimensione* per inserire un 'Contorno' aggiuntivo. In questo caso la dimensione dello sfondo è predefinita. Lo sfondo sarà costituito dal contorno più le dimensioni impostate in *Dimensione X* e *Dimensione Y*. Puoi impostare diversi tipi di *Rotazione*: 'Sincronizza con l'etichetta', 'Offset dell'etichetta' e 'Fisso'. Queste due ultime opzioni ti permettono di ruotare lo sfondo. Se imposti un *Offset X,Y* allora lo sfondo verrà spostato. Ancora, se attivi *Raggio X,Y* allora lo sfondo avrà gli angoli arrotondati. Infine, puoi scegliere diverse *Modalità fusione* (vedi [blend_modes](#)).

Usa il menu *Ombreggiatura* per aggiungere un'ombra alle etichette. Il risultato finale varierà molto in base alla scelta che farai. Hai a disposizione diverse possibilità: 'Componente etichetta inferiore', 'Testo', 'Contorno' e 'Sfondo'. L'angolazione dell' *Offset* dipende dall'orientamento dell'etichetta. Se spunti la casella di controllo  *Usa ombreggiatura globale* allora il punto zero dell'angolazione è sempre orientato verso nord e non dipende dall'orientamento dell'etichetta. Puoi influenzare l'apparenza dell'ombra tramite *Raggio di sfumatura*: più è alto il valore e più leggera sarà l'ombra. Puoi usare anche diverse modalità di fusione (vedi [blend_modes](#)).

Il menu *Posizionamento* ti permette di impostare la posizione e la priorità delle etichette. Usa il pulsante  *Offset dal centroide* in modo da attivare il *Quadrante* e posizionare le etichette. Puoi usare le impostazioni di *Rotazione* per cambiare l'angolazione delle etichette. Puoi combinare le opzioni di posizionamento tramite il quadrante e quelle di rotazione.

Nel menu *Visualizzazione* puoi impostare opzioni relative alle etichette e agli elementi delle etichette. Dal sottomenu *Opzioni etichetta* puoi impostare la visibilità delle etichette in funzione della scala. Puoi dire a QGIS di visualizzare solamente le etichette selezionate spuntando la casella di controllo  *Mostra tutte le etichette di*

questo vettore (incluse le etichette che collidono). Il sottomenu *Opzioni elementi* ti permette di scegliere quali elementi di una geometria di parti multiple devono essere etichettati. Puoi decidere di limitare il numero di elementi da etichettare e puoi spuntare la casella di controllo *Evita che le etichette si sovrappongano alle geometrie*.

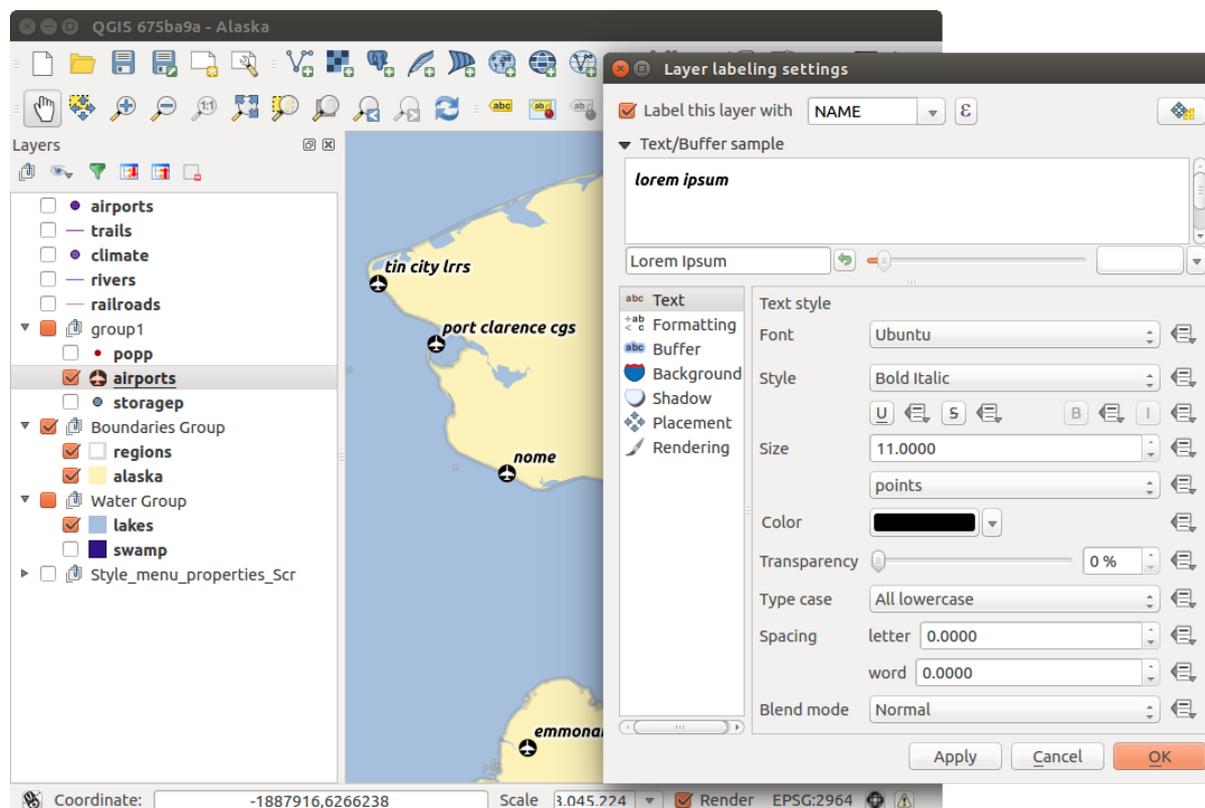


Figura 12.20: Etichettatura intelligente per vettori puntuali 

Etichettare vettore di linee

Per prima cosa, dalla scheda *Impostazioni delle etichette* spunta la casella di controllo *Etichetta questo vettore con* e seleziona un attributo fra quelli disponibili. Clicca su  se vuoi usare un'espressione per le etichette - vedi [labeling_with_expressions](#).

Imposta uno stile del testo dal menu *Testo*. Ci saranno le stesse impostazioni del vettore di punti.

Anche le impostazioni del menu *Formattazione* sono le stesse sia per i vettori di punti che per quelli di linee.

Il menu *Contorno* ha le stesse funzionalità di quelle descritte nella sezione [labeling_point_layers](#).

Il menu *Sfondo* ha le stesse voci di quelle descritte nella sezione [labeling_point_layers](#).

Anche il menu *Ombra* ha le stesse funzionalità di quelle descritte nella sezione [labeling_point_layers](#).

Nel menu *Posizionamento* troverai impostazioni particolari per i vettori di linee. Puoi scegliere fra *Parallelo*, *Curvato* oppure *Orizzontale*. Con le opzioni *Parallelo* e *Curvato* puoi impostare la posizione *Sopra la linea*, *Sulla linea* e *Sotto la linea*. Puoi scegliere più impostazioni alla volta. In questo caso, QGIS cercherà automaticamente il posizionamento migliore per le etichette. Se hai scelto l'opzione *Curvato* puoi impostare anche un *Angolo massimo tra caratteri curvi* (vedi [Figure_labels_2](#)).

Puoi impostare una distanza minima per cui non si ripetano le etichette. La distanza può essere in mm o in unità di mappa.

Alcune impostazioni di posizionamento mostrano più opzioni, ad esempio, *Curved* e *Parallel* ti consente di impostare la posizione dell'etichetta (sopra, sotto o sulla linea), : `guiabel: Distanza dalla linea` e `guiabel: Curvato`, puoi anche impostare interno esterno angolo massimo fra i caratteri delle etichette curvate.

Il menu *Visualizzazione* ha praticamente le stesse voci dei vettori di punti. Dal sottomenu *Opzioni elementi* puoi scegliere anche l'opzione *Elimina l'etichettatura degli elementi più piccoli di*.

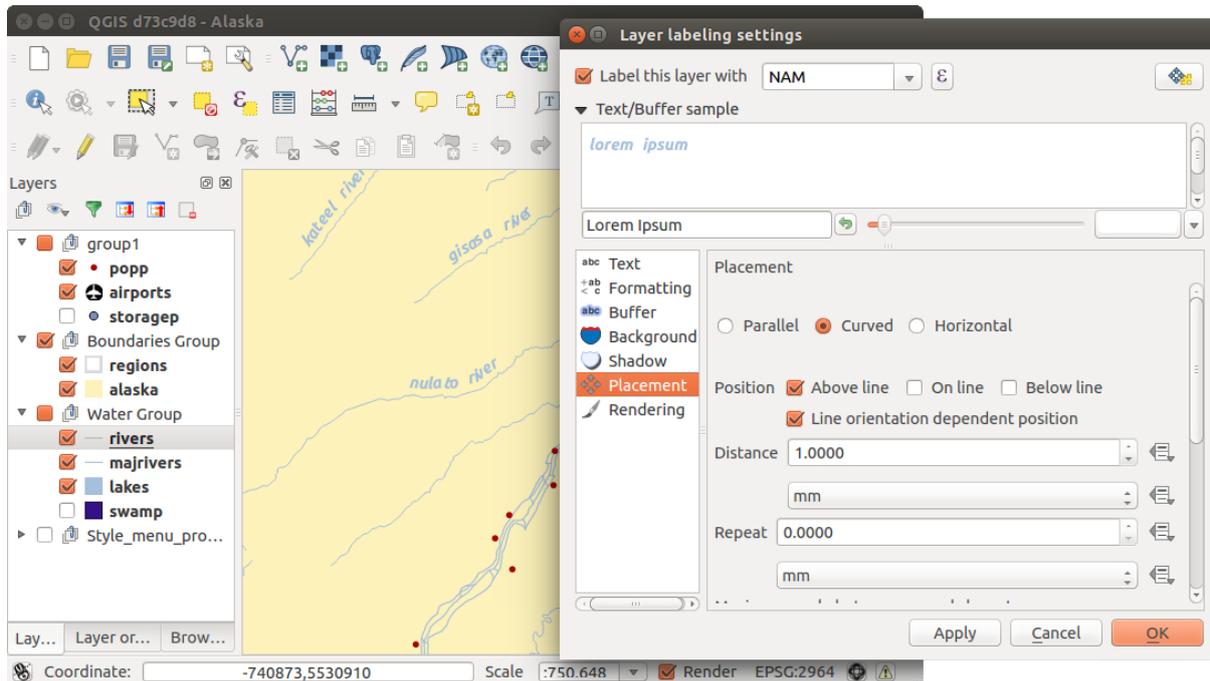


Figura 12.21: Etichettatura intelligente per vettori lineari 

Etichettare vettore di poligoni

Per prima cosa spunta la casella di controllo *Etichetta questo vettore con* e seleziona un attributo fra quelli disponibili. Clicca su ϵ se vuoi usare un'espressione per definire le etichette - vedi

Il menu *Testo* ti permette di definire lo stile per il testo. Le voci sono le stesse sia per i vettori di punti che per quelli di linee.

Il menu *Formattazione* ti permette di scegliere il formato migliore per linee multiple, le opzioni disponibili sono simili per i vettori di punti e quelli di linee.

Come per i vettori di punti e linee, puoi creare un contorno al testo fal menu *Contorno*.

Usa il menu *Sfondo* per creare uno sfondo complesso per i vettori di poligoni. Puoi usare il menu anche per i vettori di punti e linee.

Le voci nel menu *Ombre* sono le stesse sia per i punti che per le linee.

Nel menu *Posizionamento* troverai impostazioni particolari per i vettori di poligoni (vedi [Figure_labels_3](#)). Scegli fra: *Offset dal centroide*, *Orizzontale (lento)*, *Attorno al centroide*, *Libero (lento)* e *Utilizzando il perimetro*.

L'opzione *Offset dal centroide* ti permette di scegliere fra *poligono visibile* o *intero poligono*. Nel primo caso il centroide viene calcolato per il poligono visibile sulla mappa, nel secondo caso invece viene calcolato sull'intero poligono, anche se non tutte le geometrie sono visibili sulla mappa. Puoi usare il quadrante per determinare la posizione delle etichette, oltre che il loro offset e rotazione. L'opzione *Attorno al centroide* ti permette di posizionare le etichette ad una certa distanza attorno al centroide. Anche in questo caso puoi scegliere fra *poligono visibile* o *intero poligono*. L'opzione *Usando il perimetro* ti permette di scegliere distanza e posizione precise per le etichette. Per la posizione puoi scegliere di spuntare diverse caselle di controllo: *Sopra la linea*, *Sulla linea*, *Sotto la linea* e *Posizione dipendente dall'orientazione della linea*.

In relazione alle scelte di posizionamento, appariranno diverse opzioni. Come nel posizionamento dei vettori punto, puoi scegliere la distanza dal contorno del poligono, ripeti l'etichetta intorno al perimetro del poligono.

Le voci nel menu *Visualizzazione* sono le stesse dei vettori di linee. Puoi anche usare l'opzione *Elimina l'etichettatura degli elementi più piccoli* di dal menu *Opzioni elementi*. **Definisci le etichette usando**

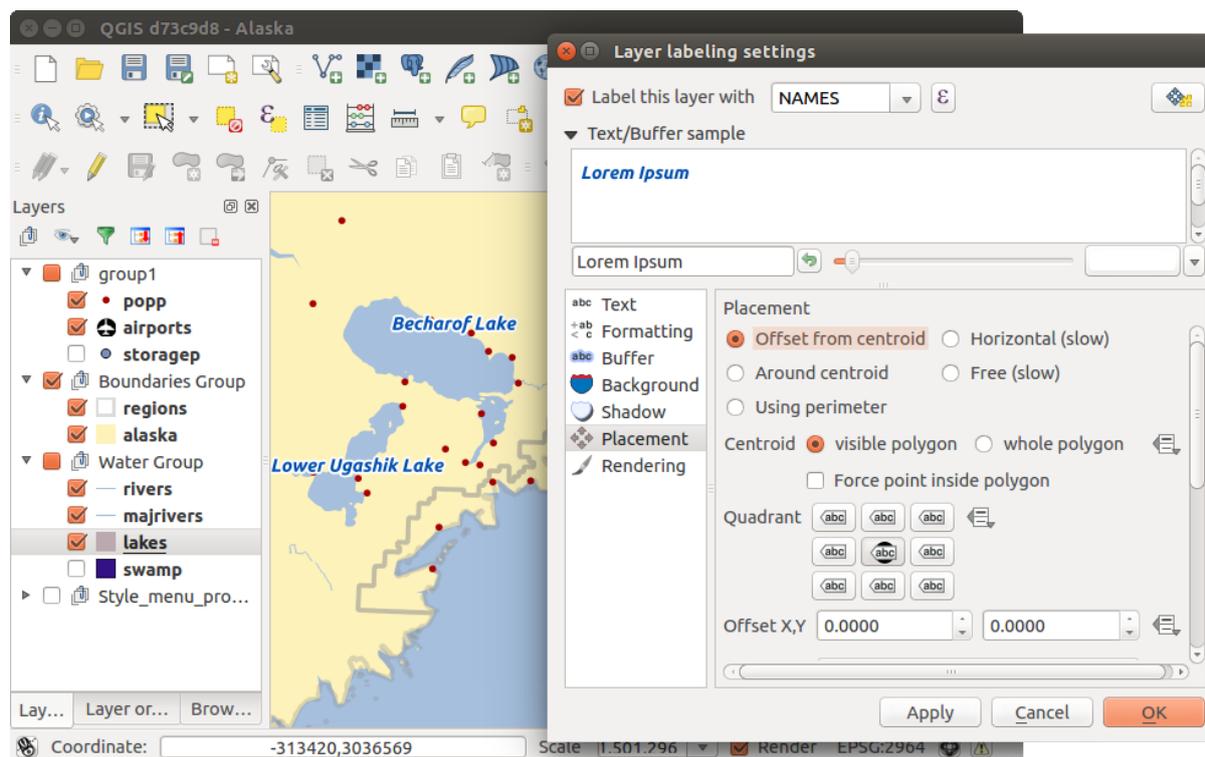


Figura 12.22: Etichettatura intelligente per vettori poligonali 

un'espressione

QGIS ti permette di usare delle espressioni per etichettare i vettori. Clicca sull'icona  nel menu  Etichette. La figura [figure_labels_4](#) mostra un esempio di espressione per etichettare la regione dell'Alaska con il nome e la dimensione dell'area. Sono stati scelti il campo 'NAME_2', un breve testo descrittivo e la funzione '\$area()' combinata con 'format_number()' per una visualizzazione ottimale.

È facile utilizzare le espressioni per etichettare i tuoi dati. Ricordati solo che devi combinare tutti gli elementi (stringhe, campi e funzioni) con il simbolo di concatenazione '||', che i campi devono essere inclusi fra le "doppie virgolette" e le stringhe fra 'virgolette singole'. Ecco qualche esempio:

```
# label based on two fields 'name' and 'place' with a comma as separator
"name" || ', ' || "place"
```

-> John Smith, Paris

```
# label based on two fields 'name' and 'place' separated by comma
'My name is ' || "name" || 'and I live in ' || "place"
```

-> My name is John Smith and I live in Paris

```
# label based on two fields 'name' and 'place' with a descriptive text
# and a line break (\n)
'My name is ' || "name" || '\nI live in ' || "place"
```

-> My name is John Smith
I live in Paris

```
# create a multi-line label based on a field and the $area function
# to show the place name and its area size based on unit meter.
'The area of ' || "place" || 'has a size of ' || $area || 'm²'
```

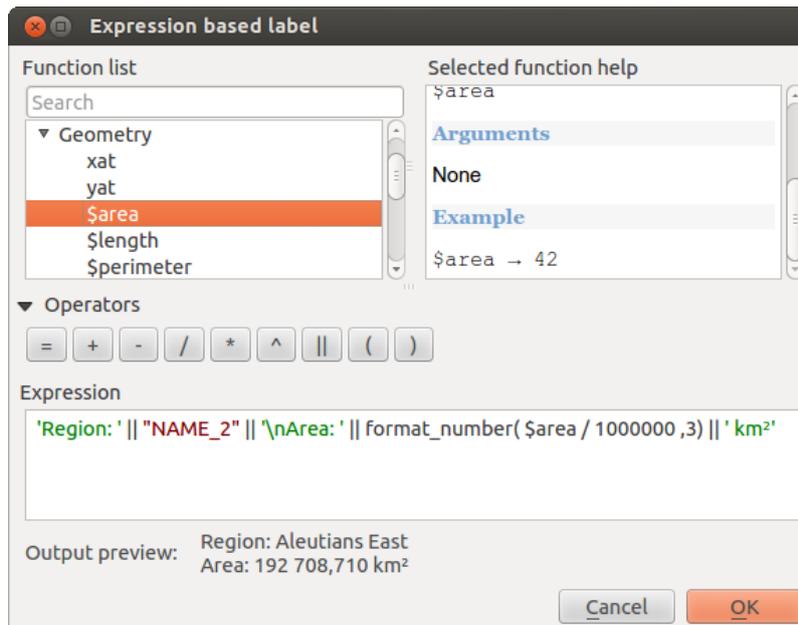


Figura 12.23: Usare le espressioni per le etichette 

-> The area of Paris has a size of 105000000 m²

```
# create a CASE ELSE condition. If the population value in field
# population is <= 50000 it is a town, otherwise a city.
'<div data-bbox="116 504 330 518" data-label="Text">


-> This place is a town


```

Come puoi vedere nel costruttore di espressioni, disponi di molte funzioni utili per produrre semplici o complesse espressioni per etichettare i dati in **lngl**. Vedi nel capitolo :ref:vector_expressions per maggiori informazioni e esempi sulle espressioni.

Usare le opzioni definite dai dati sovrascrive le impostazioni dell'etichettatura

La funzione **definito dai dati** sovrascrive tutte le impostazioni delle etichette e usa le voci presenti nella tabella degli attributi. Usa il tasto destro del mouse per attivare/disattivare questa funzione. Trascinando il mouse sopra il simbolo vedrai quali sono le informazioni usate per etichettare i dati. Ti mostriamo ora un esempio per la funzione

 Muovi etichetta (vedi figure_labels_5).

1. Importa il file lakes.shp dai dati campione.
2. Fai doppio click per aprire la finestra di dialogo Proprietà del layer. Clicca su *Etichette* e poi scegli *Posizionamento*. Seleziona *Offset dal centroide*.
3. Cerca la voce *Dati definiti*. Clicca sull'icona  per scegliere il campo per la *Coordinata*. Scegli 'xlabel' per X e 'ylabel' per Y. Vedrai le icone evidenziate in giallo.
4. Fai zoom su un lago
5. Vai sulla barra delle Etichette e clicca sull'icona . Ora potrai spostare manualmente le etichette in un'altra posizione (vedi figure_labels_6). Le coordinate della nuova posizione verranno salvate nelle rispettive colonne della tabella degli attributi.

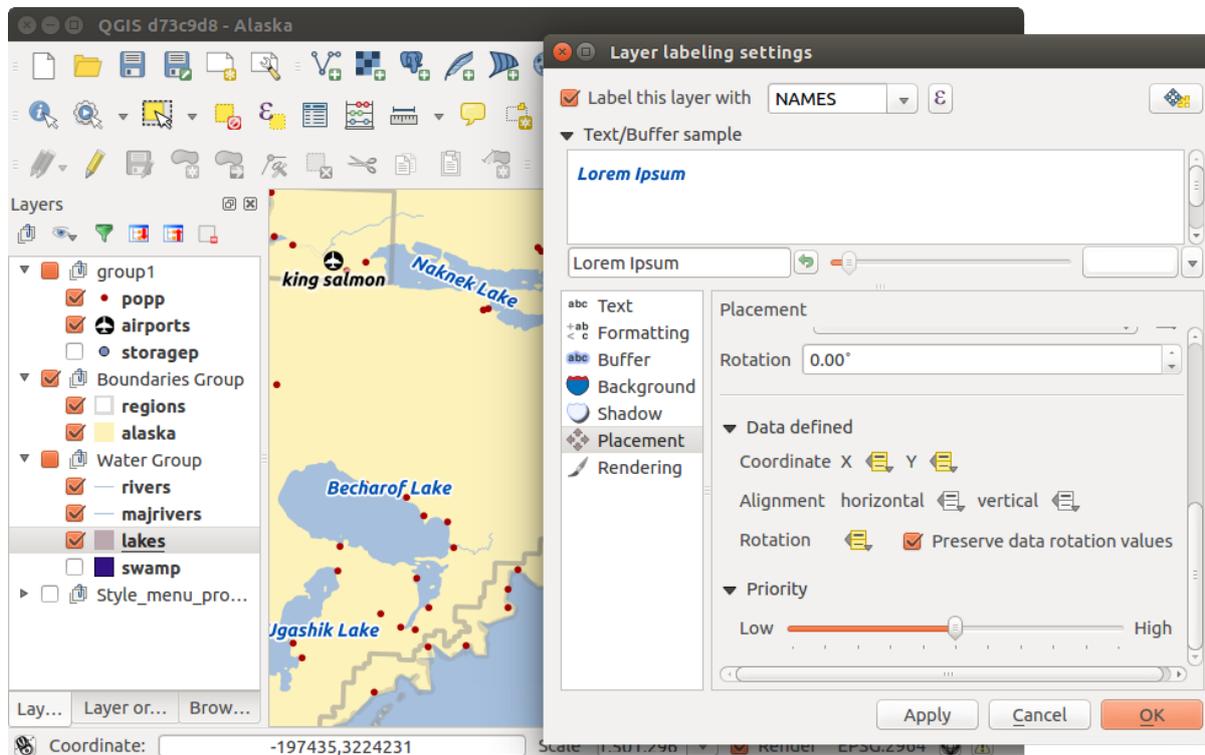


Figura 12.24: Etichettatura di un poligono con proprietà definite dai dati 🐧

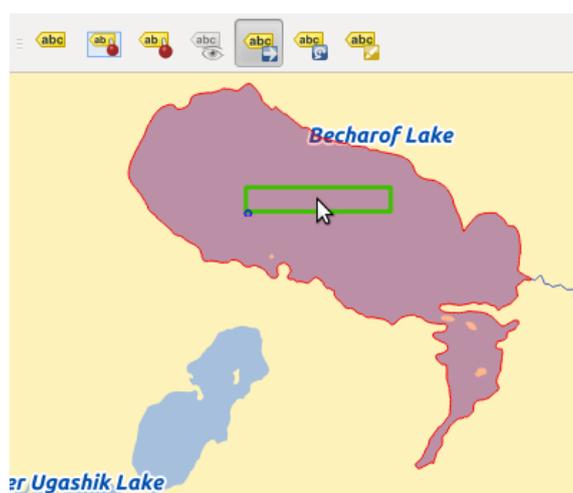


Figura 12.25: Muovi etichette 🐧

12.3.3 Menu Campi

tramite il menu *Campi*, puoi manipolare i campi e gli attributi dell'insieme di dati. Per attivare i pulsanti  Nuova colonna e  Elimina colonna devi entrare in modalità  Modifica.

Widget modifica

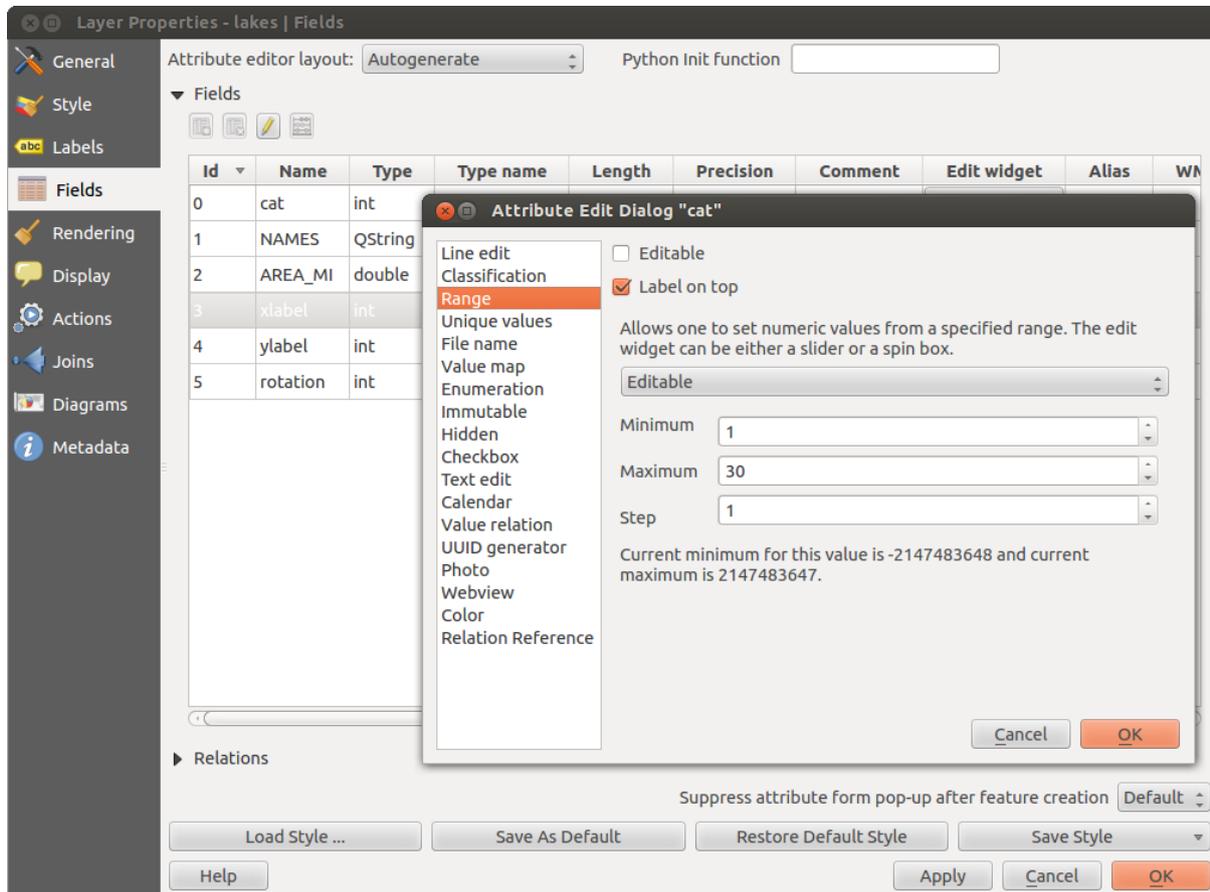


Figura 12.26: Finestra di dialogo per selezionare un widget da modificare per colonna degli attributi 

Nel menu *Campi* hai la possibilità di usare un **widget per la modifica**. In questo modo puoi specificare un tipo particolare di colonna in funzione dei dati che contiene. Se clicchi sul pulsante **[modifica contenuto]** si aprirà una finestra di dialogo in cui puoi scegliere fra diversi widget:

- **Casella di controllo:** mostra una casella di controllo: se spunti la casella allora l'attributo verrà aggiunto alla colonna, altrimenti verrà rimosso.
- **Classificazione:** visualizza un menu a tendina con i valori usati per la classificazione se hai scelto 'Categorizzato' come tipo di visualizzatore nel menu *Stile* delle proprietà del vettore.
- **Colore** virtualizza un pulsante che permette di scegliere un colore da un finestra Selettore di colore
- **** Data / Ora **:** visualizza una linea di campi che possono aprire un calendario per inserire una data. Tipo di colonna deve essere testo. È possibile selezionare un formato personalizzato, pop-up un calendario, etc.
- **Enumerazione:** apre un menu a tendina con i valori che possono essere usati nelle tipologie delle colonne. Attualmente questa funzione è supportata solo da PostgreSQL.
- **Nome file:** seleziona un file dal tuo computer grazie alla finestra dedicata.
- **Nascosto:** rende invisibile la colonna, quindi non potrai vederne il contenuto.
- **Foto:** campo che contiene un percorso ad una foto. Puoi specificare sia la larghezza che l'altezza del campo.

- **Intervallo:** ti permette di impostare dei valori di un preciso intervallo numerico. Il widget può apparire come un cursore o come un campo modificabile.
- **Riferimento alla relazione:** questo widget ti permette di nidificare il modulo di una relazione fra vettori. Per maggiori informazioni vedi *Creating one to many relations*.
- **** Text Edit **** (default): apre un campo di modifica testo che consente di modificare un testo semplice o più righe da utilizzare. Se si sceglie più linee si può anche scegliere i contenuti html.
- **Valori univoci:** puoi selezionare uno dei valori già presenti nella tabella degli attributi. Se la voce 'Modificabile' è attiva, allora comparirà anche una casella con l'autocompletamento, altrimenti verrà visualizzato un menu a tendina.
- **Generatore UUID:** genera un campo UUID (Universally Unique Identifiers) di sola lettura, se il campo è vuoto.
- **Mappa valori:** un menu a tendina con oggetti predefiniti. Puoi associare una descrizione ad valore: nel menu a tendina potrai scegliere fra le varie descrizioni, ma nella tabella degli attributi verrà scritto il valore associato. Puoi specificare i valori manualmente oppure caricarli da un file CSV.
- **Relazione valore:** potrai scegliere i valori da una tabella associata da un menu a tendina. In modo simile al widget 'Mappa valori', scegli un campo dal menu a tendina, ma nella tabella degli attributi verrà scritto il valore corrispondente.
- **Vista web:** il campo contiene un URL di una pagina internet. Puoi aggiustare la larghezza e l'altezza.

Grazie al **Modello dell'editor degli attributi** puoi creare dei moduli specifici per ogni campo (vedi figura [figure_fields_2](#)). Scegli 'Crea maschera di inserimento' e una colonna degli attributi. Usa l'icona  per creare una categoria che verrà visualizzata durante la sessione di digitalizzazione (vedi figura [figure_fields_3](#)). Ora puoi assegnare le colonne alle categorie con l'icona . Puoi creare più categorie e inserire più volte la stessa colonna. Quando crei una nuova categoria, QGIS crea automaticamente una nuova scheda nel modulo degli attributi.

Hai a disposizione anche le opzioni 'Genera automaticamente' e 'Fornisci file UI'. 'Genera automaticamente' crea un modulo semplice per ogni colonna. L'opzione 'Fornisci file UI' ti permette di usare finestre di dialogo, anche molto complesse, create con Qt-Designer. La possibilità di usare un file UI permette di liberare la tua creatività durante la creazione di una finestra di dialogo. Vedi <http://nathanw.net/2011/09/05/qgis-tips-custom-feature-forms-with-python-logic/> per maggiori informazioni.

Le finestre di dialogo di QGIS possono essere accompagnate da una funzione python che viene caricata quando apri la finestra. Usa questa funzione per aggiungere delle logiche extra. Un esempio è (nel modulo MyForms.py):

```
def open(dialog, layer, feature):
    geom = feature.geometry()
    control = dialog.findChild(QWidget, "My line edit")
```

Fa riferimento alla funzione Init di python come:

MyForms.py deve essere presente nel PYTHONPATH, in .qgis2/python, oppure nella cartella del progetto.

12.3.4 Menu Generale



Usa questo menu per gestire le impostazioni principali dei vettori. Hai a disposizione diverse opzioni:

Informazioni del layer

- Cambia il nome visualizzato del vettore in *visualizzato come*
- Specifica la *Sorgente layer* del vettore
- Specifica la *Codifica sorgente dati* per abilitare codifiche specifiche e per poter leggere il file

Sistema di Riferimento

- *Specifica* il sistema di riferimento delle coordinate. Qui puoi vedere o cambiare la proiezione del vettore.

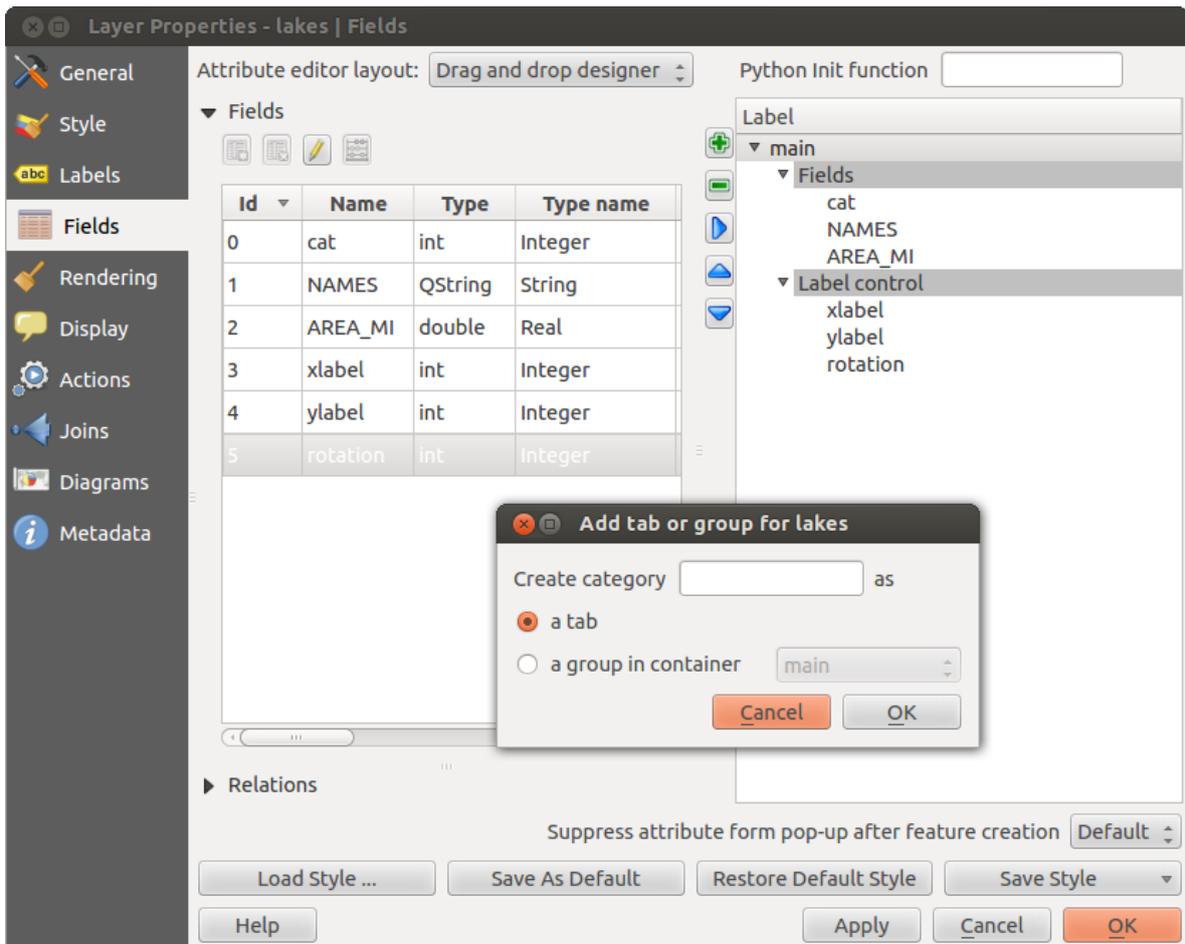


Figura 12.27: Finestra di dialogo per creare categorie con la maschera di inserimento

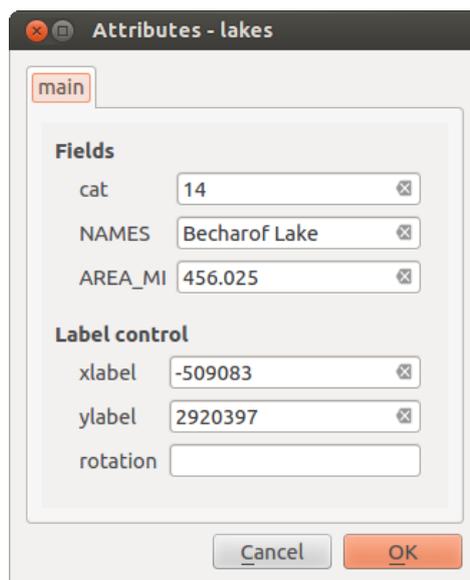


Figura 12.28: Vista di un modulo personalizzato

- Crea indice spaziale (solo per formati supportati da OGR)
- Aggiorna estensione del vettore
- Vedi o cambia la proiezione di un vettore cliccando su *Specifica ...*

Visibilità dipendente dalla scala

- Puoi impostare il *Massimo (incluso)* e *Minimo (escluso)* della scala. Puoi anche impostare la scale con il pulsante **[attuale]**.

Sottoinsieme di geometrie

- Con il pulsante **[Query Builder]** puoi creare un sottoinsieme di elementi nel vettore che saranno visualizzati (consulta anche la sezione *Costruttore di interrogazioni*).

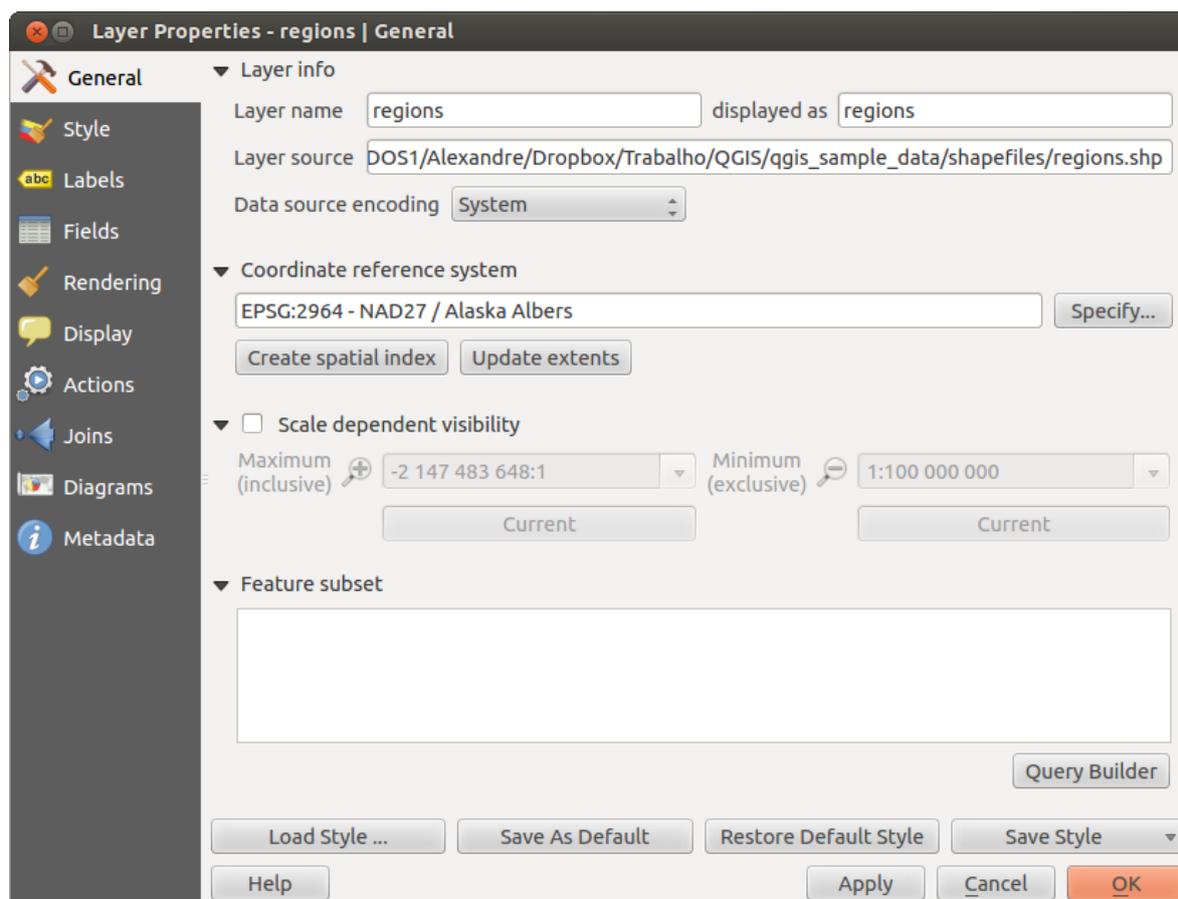


Figura 12.29: Finestra di dialogo Generale 

12.3.5 Menu Visualizzazione

QGIS 2.2 introduce il supporto per la semplificazione ‘al volo’ delle geometrie. Questa funzionalità velocizza la visualizzazione di geometrie specialmente a piccole scale. Puoi attivare o disattivare questa funzionalità dalle proprietà del vettore, usando la casella di controllo *Semplifica geometrie*, ma puoi anche abilitarla in modo predefinito per ogni vettore aggiunto (vedi sezione *Opzioni dell'interfaccia grafica (GUI)*). **Nota:** la semplificazione delle geometrie, in alcuni casi, può creare degli artefatti; ovvero si potranno generare dei poligoni frammentati e la visualizzazione basata su offset potrebbe risultare leggermente inaccurata.

12.3.6 Menu Visualizza

Questo menu serve per visualizzare i Suggerimenti mappa. I suggerimenti vengono visualizzati in HTML. Quando trascini il mouse sopra un elemento, il suggerimento può corrispondere a un *Campo*, ma ora puoi anche inserire del codice HTML. Per attivare i Suggerimenti mappa, attiva l'opzione dal menu *Visualizza* → *Suggerimenti mappa*. La figura Figure Display 1 mostra un esempio di codice HTML.

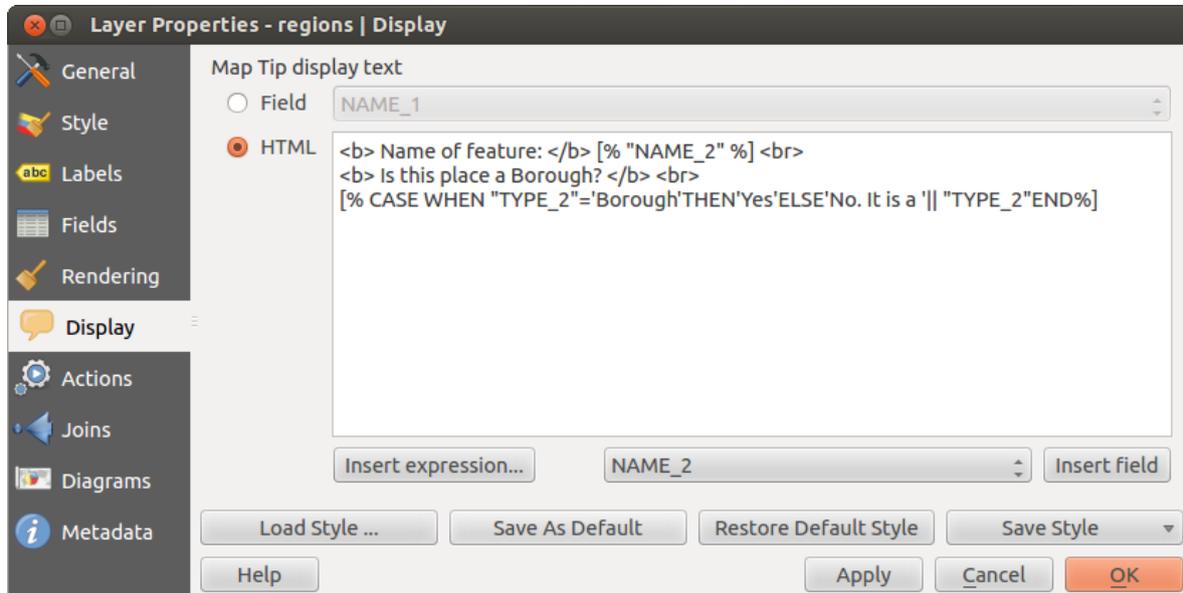


Figura 12.30: Codice HTML per i suggerimenti mappa 🐧

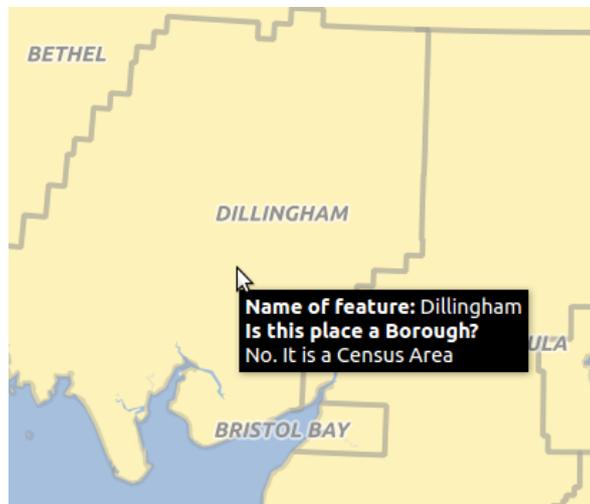


Figura 12.31: Suggerimenti mappa creati con codice HTML 🐧

12.3.7 Menu Azioni

QGIS ti offre la possibilità di creare azioni sulla base degli attributi associati ai singoli elementi del vettore. Potrai così creare un grande numero di azioni, per esempio, avviare un programma con argomenti costituiti dagli attributi di un vettore.

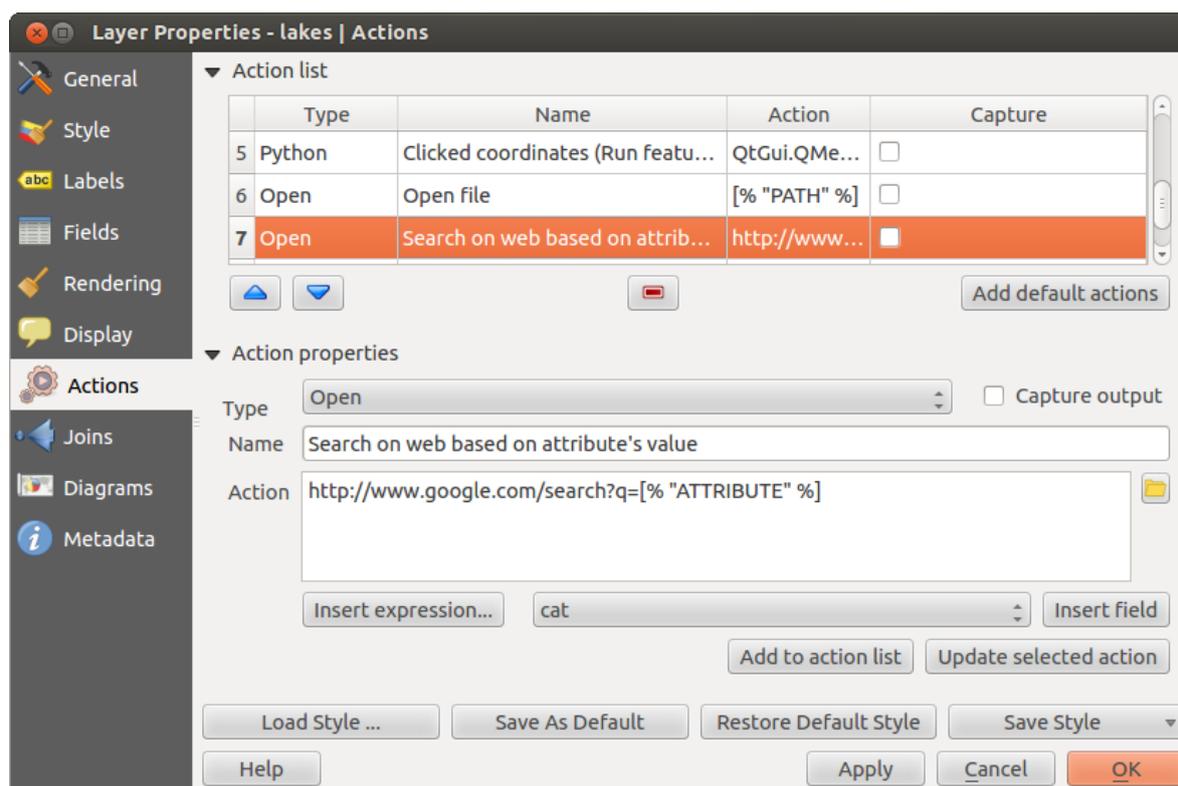


Figura 12.32: Panoramica della finestra di dialogo Azioni 

Le azioni sono utili quando vuoi avviare un'applicazione esterna oppure aprire una pagina web sulla base di uno o più valori associati al vettore. Ci sono sei tipologie di azioni che puoi usare nel seguente modo:

- Le azioni Generic, Mac, Windows e Unix avviano un processo esterno.
- Le azioni python eseguono un'espressione python,
- Le azioni generic e python sono visibili ovunque.
- Le azioni Mac, Windows e Unix sono visibili solo sulle rispettive piattaforme (cioè puoi definire le azioni, ma sari in grado di vedere i risultati solamente sulla piattaforma dalla quale è stato lanciato l'editor).

Ci sono diversi esempi presenti nella finestra di dialogo. Puoi caricarli cliccando su [**Aggiungi azioni predefinite**]. Un esempio è eseguire una ricerca sul web basata su un valore presente nella tabella degli attributi. Di seguito una spiegazione.

Definire le azioni

Puoi definire le azioni dalla finestra di dialogo *Proprietà layer*. Per definire un'azione, apri questa finestra e clicca sul menu *Azioni*. Dal sottomenu *Proprietà azione* scegli 'Generico' come tipo e inserisci un nome per l'azione. L'azione deve contenere il nome dell'applicazione che verrà eseguita quando lancerai l'azione. Puoi aggiungere uno o più campi come argomenti per l'applicazione. L'azione inizierà sempre con il carattere % seguito dal nome del campo che verrà sostituito dal valore del campo stesso. I caratteri speciali %% verranno sostituiti dal valore del campo che hai scelto dalla tabella degli attributi (vedi [using_actions](#)). Puoi usare le doppie virgolette, " per raggruppare il testo in un singolo argomento per il programma, script o comando. Le doppie virgolette saranno ignorate se precedute dal carattere \.

Se sono presenti nomi di campi che possono essere interpretati come sotto-stringhe di altri nomi di campi (ad es. col1 e col10) devi racchiudere il nome (e il carattere %) tra parentesi quadre (es. [%col10]). Questo impedirà che il nome di campo %col10 possa essere confuso con %col1 con uno 0 alla fine. Le virgolette saranno rimosse da QGIS man mano che inserirai i valori del campo. Se vuoi che i campi sostituiti vengano racchiusi entro parentesi quadre, aggiungi una seconda coppia di parentesi quadre: [[%col10]].

La finestra di dialogo *Informazione sui risultati* che compare quando usi lo strumento *Informazioni elementi* ha una voce (*Derivato*) che contiene informazioni che dipendono dal tipo di vettore interrogato. Puoi accedere ai valori di questa voce in modo simile a come accedi ad altri campi della tabella attributi antepoendo al nome del campo (*Derivato*). Per esempio un vettore di punti ha due campi, X e Y, e puoi usare il loro valore nell'azione con l'espressione `% (Derivato) .X e % (Derivato) .Y`. Gli attributi derivati sono disponibili solo nella finestra *Informazione sui risultati* e non nella finestra *Tabella degli attributi*.

Due esempi di azioni sono di seguito indicati:

- konqueror `http://www.google.com/search?q=%nam`
- konqueror `http://www.google.com/search?q=%%`

Nel primo esempio, verrà lanciato il browser konqueror che aprirà un URL. L'URL crea una ricerca Google sul valore del campo `nam` nel vettore. Il programma o lo script richiamato dall'azione deve essere nel path delle variabili d'ambiente altrimenti dovrai specificare il percorso completo del programma. Il primo esempio infatti è accessibile anche con `/opt/kde3/bin/konqueror http://www.google.com/search?q=%nam`. In questo modo sei sicuro che l'applicazione konqueror verrà eseguita quando si richiama l'azione..

Nel secondo esempio viene usata la notazione `%%` che non richiede l'indicazione di un particolare campo. Quando richiami l'azione, il `%%` sarà rimpiazzato dal valore selezionato sia nella finestra *Informazioni risultati* sia nella tabella degli attributi. **Uso delle azioni**

Le azioni possono essere richiamate sia dalla finestra *Informazioni sui risultati* che dalla *Tabella degli attributi* (ricordati che puoi aprire queste finestre rispettivamente cliccando sullo strumento  Informazioni elementi  (ricordati che puoi aprire queste finestre rispettivamente cliccando sullo strumento  Apri tabella degli attributi o  Avvia azione sull'elemento). Per eseguire l'azione, clicca con il tasto destro del mouse sul risultato e scegli l'azione dal menu contestuale. Le azioni sono indicate nel menu a tendina con il nome inserito. Clicca sull'azione che vuoi eseguire.

Se stai richiamando un'azione che usa l'annotazione `%%`, fai click con il tasto destro sul valore del campo nella finestra *Informazioni risultati* oppure dalla finestra *Tabella attributi* e scegli l'applicazione o lo script da assegnare.

In questo altro esempio viene mostrato come estrarre dati da un vettore per inserirli in un file usando il terminale e il comando `echo` (quindi funzionerà su  e forse su  **X**). Il vettore in questione ha i seguenti campi nella tabella attributi: nome della specie `taxon_name`, latitudine `lat` e longitudine `long`. Vuoi eseguire una selezione spaziale delle specie (`taxon`) presenti in determinate posizioni, esportando i risultati in un file di testo (evidenziate in giallo sulla mappa di QGIS). Ecco l'azione giusta per questo scopo:

```
bash -c "echo \"%taxon_name %lat %long\" >> /tmp/species_localities.txt"
```

Selezionando solo alcune posizioni, l'esecuzione dell'azione precedente genera un file di output fatto così:

```
Acacia mearnsii -34.0800000000 150.0800000000
Acacia mearnsii -34.9000000000 150.1200000000
Acacia mearnsii -35.2200000000 149.9300000000
Acacia mearnsii -32.2700000000 150.4100000000
```

Come esercizio puoi creare un'azione che lancia una ricerca su Google in base al vettore `lakes`. Prima di tutto devi impostare l'URL necessario per eseguire una ricerca basata su una parola chiave. Puoi copiare facilmente l'espressione facendo una ricerca semplice dalla pagina di Google. La pagina dei risultati avrà un indirizzo, visibile nella barra indirizzi del browser, del tipo: `http://google.com/search?q=qgis`, in cui QGIS è la parola ricercata. Ora puoi procedere:

1. Assicurarti di aver caricato il vettore `lakes`.
2. Apri la finestra di dialogo *Proprietà layer* facendo doppio click sul vettore o cliccandoci sopra con il tasto destro del mouse e scegliendo *Proprietà* dal menu contestuale.
3. Clicca sul menu *Azioni*.
4. Inserisci un nome descrittivo per l'azione, ad esempio *Ricerca Google*.
5. Devi fornire il nome del programma esterno, in questo caso Firefox. Se il programma non è presente nel tuo path, devi inserire il path assoluto.

6. Dopo il nome dell'applicazione esterna, aggiungi l'URL della ricerca di Google, senza includere il termine della ricerca: `http://google.com/search?q=`
7. A questo punto il testo nel campo *Azioni* dovrebbe apparire così: `firefox http://google.com/search?q=`
8. Clicca sul menu a tendina che contiene i nomi dei campi del vettore *lakes*, posizionato immediatamente a sinistra del pulsante **[Inserisci campo]**.
9. Dal menu a tendina, seleziona 'NAMES' e clicca su **[Inserisci campo]**.
10. Il testo dell'azione dovrebbe ora apparire come segue:
`firefox http://google.com/search?q=%NAMES`
11. Per concludere questa azione, clicca sul pulsante **[Aggiungi alla lista di azioni]**.

Questo ultimo passo completa l'azione che è ora pronta per essere usata. Il testo finale dell'azione dovrebbe apparire così:

```
firefox http://google.com/search?q=%NAMES
```

A questo punto puoi usare l'azione. Chiudi la finestra *Proprietà layer* e usa lo zoom su un'area a scelta. Assicurati che il vettore *lakes* sia attivo ed identifica con l'apposito strumento un lago qualsiasi. Nella finestra risultante dovrebbe essere visibile l'azione:

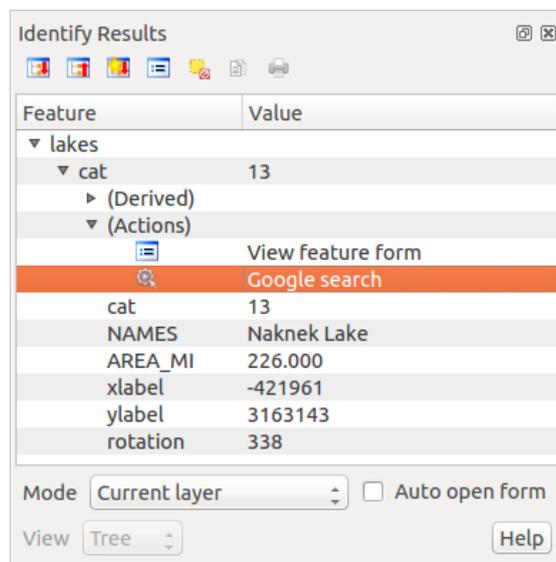


Figura 12.33: Seleziona un elemento e scegli un'azione 

Cliccando sull'azione, verrà lanciato Firefox all'URL <http://www.google.com/search?q=Tustumena>. Puoi anche aggiungere altri campi all'azione, aggiungendo un + alla fine della stringa che definisce l'azione, selezionando quindi un altro campo e cliccando sul pulsante **[Inserisci campo]**. Nel nostro esempio non c'è alcun altro campo sul quale avrebbe senso fare una ricerca.

Puoi definire più di un'azione per ogni vettore, ognuna delle quali verrà mostrata nella finestra *Informazioni sui risultati*.

Puoi creare tantissimi tipi di azione. Per esempio se hai un vettore di punti che fa riferimento alle posizioni dove sono state scattate foto o immagini, insieme al nome stesso del file, puoi creare un'azione per avviare un programma che visualizzerà l'immagine. Puoi usare le azioni anche per lanciare report sul web per uno o più campi della tabella degli attributi, definendole allo stesso modo dell'esempio per la ricerca con Google.

Ci sono esempi anche molto più complicati, per esempio usando le azioni **Python**.

Normalmente quando usi un'azione per aprire un file con un'applicazione esterna, puoi usare un path assoluto o relativo. Nel secondo caso, il path è relativo alla posizione dell'eseguibile dell'applicazione esterna. Ma come

si fa se devi usare un path relativo al vettore selezionato (se è un file, come uno shapefile o Spatialite)? Ecco il trucco:

```
command = "firefox";
imagerelpath = "images_test/test_image.jpg";
layer = qgis.utils.iface.activeLayer();
import os.path;
layerpath = layer.source() if layer.providerType() == 'ogr'
    else (qgis.core.QgsDataSourceURI(layer.source()).database()
    if layer.providerType() == 'spatialite' else None);
path = os.path.dirname(str(layerpath));
image = os.path.join(path, imagerelpath);
import subprocess;
subprocess.Popen( [command, image ] );
```

Ricordati che l'azione è del tipo *Python*, quindi devi cambiare le variabili *command* e *imagerelpath*.

E se il percorso relativo deve essere relativo al file di progetto (salvato)? Il codice per l'azione Python diventa:

```
command="firefox";
imagerelpath="images/test_image.jpg";
projectpath=qgis.core.QgsProject.instance().fileName();
import os.path; path=os.path.dirname(str(projectpath)) if projectpath != '' else None;
image=os.path.join(path, imagerelpath);
import subprocess;
subprocess.Popen( [command, image ] );
```

Un altro esempio di azione python è quello che ti permette di aggiungere nuovi layer al progetto. In questo esempio aggiungeremo sia un vettore che un raster. Il nome dei file da aggiungere al progetto e il nome da assegnare ai layer è specificato dai dati (*filename* e *layname* sono nomi di colonne della tabella dagli attributi del vettore dove l'azione è stata creata):

```
qgis.utils.iface.addVectorLayer('/yourpath/[% "filename" %].shp', '[% "layername" %]',
    'ogr')
```

Per aggiungere un raster (in questo caso un'immagine TIF), diventa:

```
qgis.utils.iface.addRasterLayer('/yourpath/[% "filename" %].tif', '[% "layername" %]
')
```

12.3.8 Menu Join



Il *Join* consente di join una tabella degli attributi caricata a un layer vettoriale caricato. Dopo aver cliccato



viene visualizzata la finestra *Aggiungi vettore da unire (join)*. Come colonne chiave, è necessario definire un: *index:Vettore da unire (join)* che conatterai con il vettore. Quindi, è necessario specificare il campo join che è comune sia al livello di unirsi e il livello di destinazione. Ora è anche possibile specificare un sottoinsieme di campi dallo strato aderito in base alla casellalcheckbox: *guiabel: Seleziona i campi per l'unione*. Come risultato del join, tutte le informazioni dallo strato aderire e il livello di destinazione vengono visualizzati nella tabella attributo del vettore di destinazione come informazione unita. Se è stato specificato un sottoinsieme di campi solo questi campi vengono visualizzati nella tabella degli attributi del livello di destinazione.

QGIS attualmente supporta anche join di tabella non spaziali supportate da OGR (come CSV, DBF e EXcel), file di testo delimitato e PostgreSQL (vedi [figure_joins_1](#)).

Inoltre, la finestra di dialogo aggiungi vettore da unire ti permette di:

- *Layer unito in memoria virtuale*
- *Crea un indice nel campo unito*

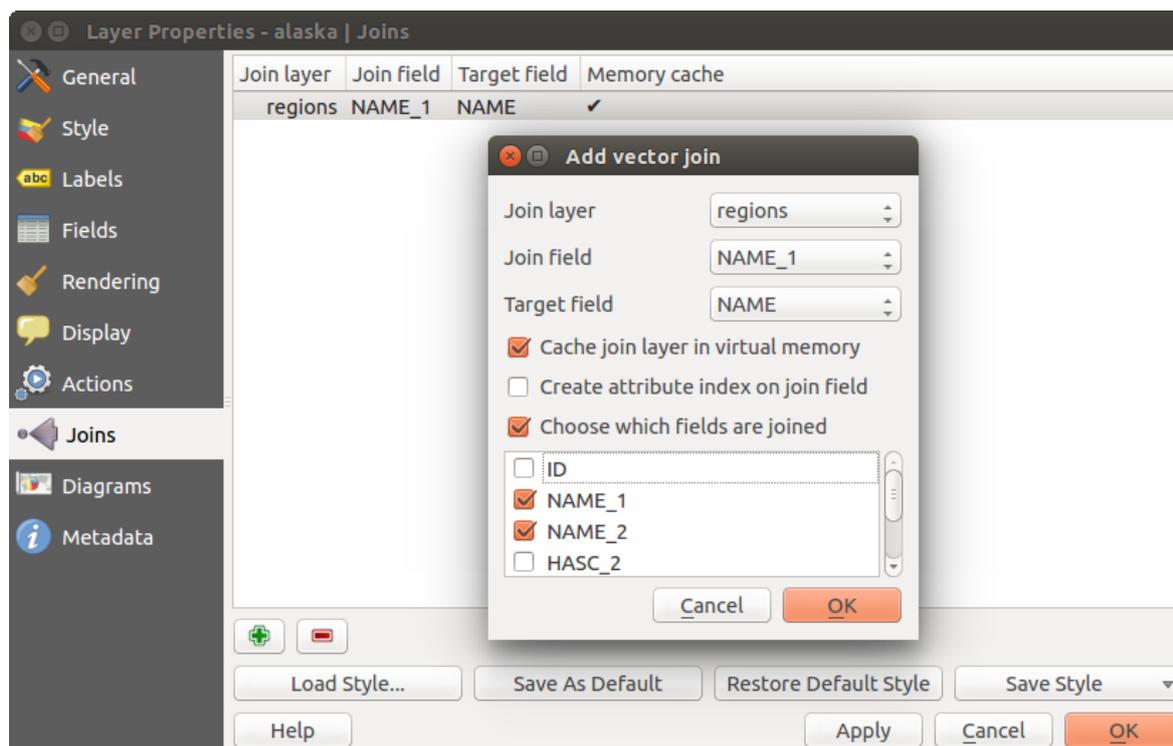


Figura 12.34: Unisci una tabella di attributi con un vettore 🐧

12.3.9 Menu Diagrammi



Il menu *Diagrammi* permette di sovrapporre diagrammi a un vettore (figura [figure_diagrams_1](#)).

L'implementazione nativa per i diagrammi ti permette di creare grafici a torta, diagrammi di testo e istogrammi.

Il menu è suddiviso in quattro schede: *Aspetto*, *Dimensione*, *Posizione* e *Opzioni*.

Nel caso dei grafici a torta e dei diagrammi di testo, gli attributi dei campi scelti per la classificazione verranno visualizzati in un cerchio o in un riquadro. La dimensione dei diagrammi è basata su una scala fissa oppure in base a un attributo. La posizione dei diagrammi interagisce con il nuovo sistema di etichettatura, quindi eventuali conflitti fra etichette e diagrammi sono ora risolti. Inoltre puoi anche posizionare manualmente i diagrammi.

Ecco un esempio di creazione di un diagramma di testo delle temperature sovrapposto al vettore alaska; entrambi i vettori sono disponibili fra i dati campione di QGIS (sezione *Dati campione*).

1. Per prima cosa, clicca su  **Aggiungi vettore** e carica i due file `alaska.shp` e `climate.shp` dall'insieme di dati di QGIS.
2. Fai doppio click sul vettore `climate` nella legenda per aprire la finestra di dialogo *Proprietà layer*.
3. Clicca sulla scheda *Diagrammi*, spunta la casella di controllo *Visualizza diagrammi* e scegli 'Diagramma testo' dal menu a tendina *Tipo di diagramma* .
4. Nella scheda *Aspetto* scegli un blu chiaro come colore di sfondo e nella scheda *Dimensione* imposta 18 mm come dimensione fissa.
5. Nella scheda *Posizione* scegli 'Intorno al punto' come posizionamento.
6. Nel diagramma verranno visualizzati i valori di tre colonne: `T_F_JAN`, `T_F_JUL` e `T_F_MEAN`. Seleziona prima `T_F_JAN` come *Attributo* e clicca sul pulsante  poi esegui la stessa procedura per `T_F_JUL` e `T_F_MEAN`.
7. Clicca su [**Applica**] per visualizzare il diagramma nella finestra principale di QGIS.

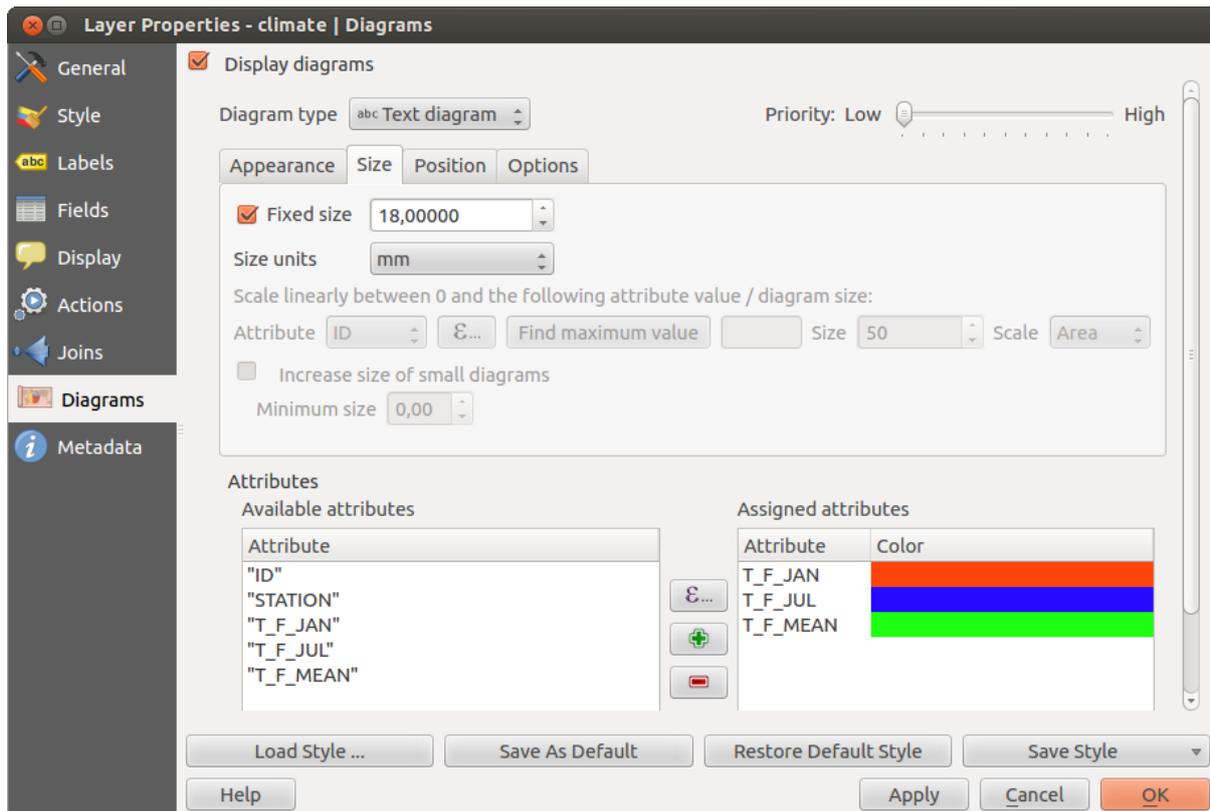


Figura 12.35: Finestra di dialogo dei diagrammi 

8. Puoi aggiustare la dimensione del grafico nella scheda *Dimensione*. Disattiva la casella di controllo *Dimensione fissa* e imposta la dimensione del diagramma in base all'attributo con il pulsante [**Trova valore massimo**]. Se il diagramma è troppo piccolo, spunta la casella di controllo *Aumenta la dimensione dei diagrammi piccoli* e specifica la dimensione minima dei diagrammi.
9. Cambia il colore degli attributi facendo doppio click su colore nel campo *Attributi assegnati*. La figura [Figure_diagrams_2](#) ti dà un'idea del risultato.
10. Clicca su [**OK**].

Ricordati che puoi usare anche l'opzione *Posizione definita da attributo* per posizionare i diagrammi. Inoltre puoi anche impostare la visibilità dei diagrammi in funzione di determinate scale dalla scheda *Aspetto*.

Le dimensioni e gli attributi possono anche essere un'espressione. Usa il pulsante  per aggiungere un'espressione. Vedi il capitolo [Expressions](#) per ulteriori informazioni ed esempi.

12.3.10 Menu Metadati



Il menu *Metadati* ha quattro sezioni: *Descrizione*, *Attribuzione*, *URL Metadati* e *Proprietà*.

Nella sezione *Proprietà* vengono elencate informazioni generali sul vettore, incluse le specifiche sul file, la sorgente del file, il numero di geometrie, il tipo di geometrie e le capabilities. La voce **Estensioni** fornisce l'estensione del vettore e le informazioni sul *Sistema di riferimento spaziale del layer*. Questo è un semplice e rapido metodo per ottenere informazioni sul vettore.

Inoltre puoi aggiungere un titolo e un breve riassunto del vettore nelle sezione *Descrizione*. Puoi anche impostare una *Lista delle parole chiave*. Questa lista può essere usata in un catalogo di metadati. Se vuoi usare un titolo di un file XML, devi aggiungere un collegamento in *URL dati*. Usa la sezione *Attribuzione* per ottenere dati sugli attributi da un catalogo metadati in XML. Nella sezione *URL Metadati* puoi specificare il path generico al

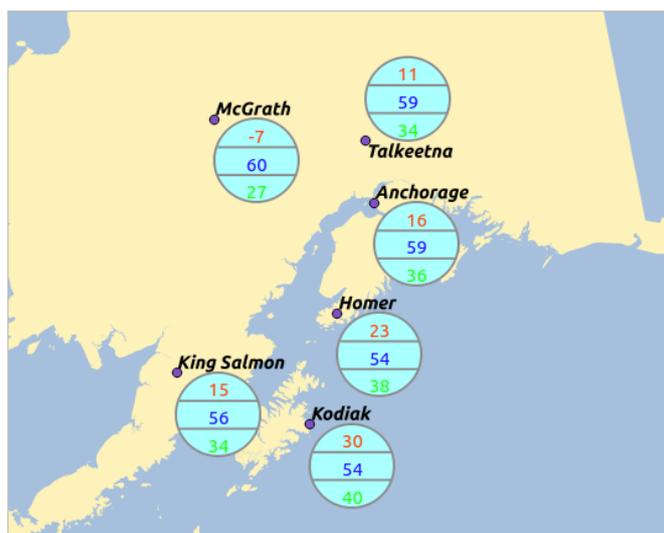


Figura 12.36: Diagrammi di temperatura sovrapposti su una mappa 🐧

catalogo dati XML. Le informazioni verranno salvate nel file di progetto di QGIS per sessioni future, ma lo puoi usare anche su QGIS server.

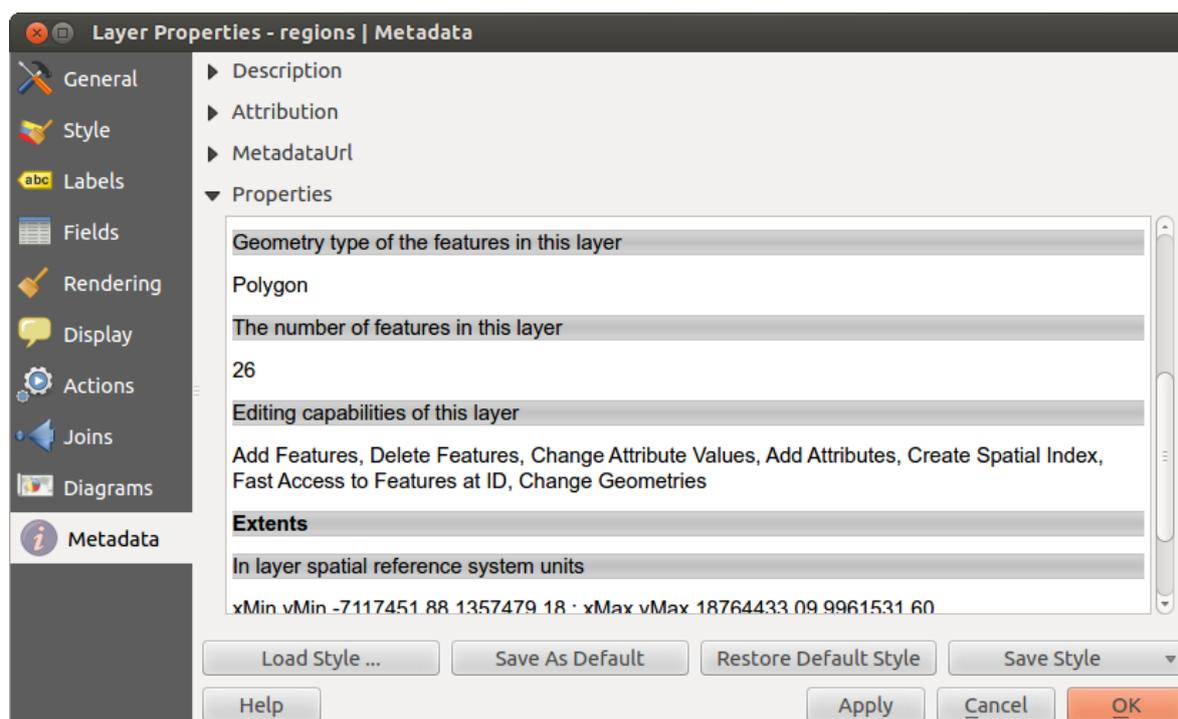


Figura 12.37: Finestra di dialogo dei metadati 🐧

12.4 Expressions

The **Expressions** feature are available through the field calculator or the add a new column button in the attribut table or the Field tab in the Layer properties ; through the graduated, categorized and rule-based rendering in the

Style tab of the Layer properties ; through the expression-based labeling  in the  Labeling core application ; through the feature selection and through the diagram tab of the Layer properties.

There are powerful way to manipulate attribute value in order to dynamically change the final value in order to change the geometry style, the content of the label, the value for diagram, select some feature or create virtual column.

12.4.1 Functions List

The **Function List** contains functions as well as fields and values. View the help function in the **Selected Function Help**. In **Expression** you see the calculation expressions you create with the **Function List**. For the most commonly used operators, see **Operators**.

In the **Function List**, click on *Fields and Values* to view all attributes of the attribute table to be searched. To add an attribute to the Field calculator **Expression** field, double click its name in the *Fields and Values* list. Generally, you can use the various fields, values and functions to construct the calculation expression, or you can just type it into the box. To display the values of a field, you just right click on the appropriate field. You can choose between *Load top 10 unique values* and *Load all unique values*. On the right side, the **Field Values** list opens with the unique values. To add a value to the Field calculator **Expression** box, double click its name in the **Field Values** list.

The *Operators*, *Math*, *Conversions*, *String*, *Geometry* and *Record* groups provide several functions. In *Operators*, you find mathematical operators. Look in *Math* for mathematical functions. The *Conversions* group contains functions that convert one data type to another. The *String* group provides functions for data strings. In the *Geometry* group, you find functions for geometry objects. With *Record* group functions, you can add a numeration to your data set. To add a function to the Field calculator **Expression** box, click on the > and then double click the function.

Operators

This group contains operators (e.g., +, -, *).

a + b	a plus b
a - b	a minus b
a * b	a multiplied by b
a / b	a divided by b
a % b	a modulo b (for example, 7 % 2 = 1, or 2 fits into 7 three times with remainder 1)
a ^ b	a power b (for example, 2^2=4 or 2^3=8)
a = b	a and b are equal
a > b	a is larger than b
a < b	a is smaller than b
a <> b	a and b are not equal
a != b	a and b are not equal
a <= b	a is less than or equal to b
a >= b	a is larger than or equal to b
a ~ b	a matches the regular expression b
+ a	positive sign
- a	negative value of a
	joins two values together into a string 'Hello' ' world'
LIKE	returns 1 if the string matches the supplied pattern
ILIKE	returns 1 if the string matches case-insensitive the supplied pattern (ILIKE can be used instead of LIKE to make the match case-insensitive)
IS	returns 1 if a is the same as b
OR	returns 1 when condition a or b is true
AND	returns 1 when condition a and b are true
NOT	returns 1 if a is not the same as b
column name "column name"	value of the field column name, take care to not be confused with simple

'string'	quote, see below a string value, take care to not be confused with double quote, see above
NULL	null value
a IS NULL	a has no value
a IS NOT NULL	a has a value
a IN (value[,value])	a is below the values listed
a NOT IN (value[,value])	a is not below the values listed

Some example:

- Joins a string and a value from a column name:

```
'My feature's id is: ' || "gid"
```

- Test if the “description” attribute field starts with the ‘Hello’ string in the value (note the position of the % character):

```
"description" LIKE 'Hello%'
```

Conditionals

This group contains functions to handle conditional checks in expressions.

CASE	evaluates multiple expressions and returns a result
CASE ELSE	evaluates multiple expressions and returns a result
coalesce	returns the first non-NULL value from the expression list
regexp_match	returns true if any part of a string matches the supplied regular expression

Some example:

- Send back a value if the first condition is true, else another value:

```
CASE WHEN "software" LIKE '%QGIS%' THEN 'QGIS' ELSE 'Other'
```

Mathematical Functions

This group contains math functions (e.g., square root, sin and cos).

sqrt(a)	square root of a
abs	returns the absolute value of a number
sin(a)	sine of a
cos(a)	cosine of a
tan(a)	tangent of a
asin(a)	arcsin of a
acos(a)	arccos of a
atan(a)	arctan of a
atan2(y,x)	arctan of y/x using the signs of the two arguments to determine the quadrant of the result
exp	exponential of a value
ln	value of the natural logarithm of the passed expression
log10	value of the base 10 logarithm of the passed expression
log	value of the logarithm of the passed value and base
round	round to number of decimal places

rand	random integer within the range specified by the minimum and maximum argument (inclusive)
randf	random float within the range specified by the minimum and maximum argument (inclusive)
max	largest value in a set of values
min	smallest value in a set of values
clamp	restricts an input value to a specified range
scale_linear	transforms a given value from an input domain to an output range using linear interpolation
scale_exp	transforms a given value from an input domain to an output range using an exponential curve
floor	rounds a number downwards
ceil	rounds a number upwards
\$pi	pi as value for calculations

Conversions

This group contains functions to convert one data type to another (e.g., string to integer, integer to string).

toint	converts a string to integer number
toREAL	converts a string to real number
toString	converts number to string
toDateTime	converts a string into Qt data time type
toDate	converts a string into Qt data type
toTime	converts a string into Qt time type
toInterval	converts a string to an interval type (can be used to take days, hours, months, etc. off a date)

Date and Time Functions

This group contains functions for handling date and time data.

\$now	current date and time
age	difference between two dates
year	extract the year part from a date, or the number of years from an interval
month	extract the month part from a date, or the number of months from an interval
week	extract the week number from a date, or the number of weeks from an interval
day	extract the day from a date, or the number of days from an interval
hour	extract the hour from a datetime or time, or the number of hours from an interval
minute	extract the minute from a datetime or time, or the number of minutes from an interval
second	extract the second from a datetime or time, or the number of minutes from an interval

Some example:

- Get the month and the year of today in the format “10/2014”

```
month($now) || '/' || year($now)
```

String Functions

This group contains functions that operate on strings (e.g., that replace, convert to upper case).

lower	convert string a to lower case
upper	convert string a to upper case
title	converts all words of a string to title case (all words lower case with leading capital letter)
trim	removes all leading and trailing white space (spaces, tabs, etc.) from a string
wordwrap	returns a string wrapped to a maximum/minimum number of characters
length	length of string a
replace	returns a string with the supplied string replaced
regexp_replace(a,this,that)	returns a string with the supplied regular expression replaced
regexp_substr	returns the portion of a string which matches a supplied regular expression
substr(*a*,from,len)	returns a part of a string
concat	concatenates several strings to one
strpos	returns the index of a regular expression in a string
left	returns a substring that contains the n leftmost characters of the string
right	returns a substring that contains the n rightmost characters of the string
rpadd	returns a string with supplied width padded using the fill character
lpadd	returns a string with supplied width padded using the fill character
format	formats a string using supplied arguments
format_number	returns a number formatted with the locale separator for thousands (also truncates the number to the number of supplied places)
format_date	formats a date type or string into a custom string format

Color Functions

This group contains functions for manipulating colors.

color_rgb	returns a string representation of a color based on its red, green, and blue components
color_rgba	returns a string representation of a color based on its red, green, blue, and alpha (transparency) components
ramp_color	returns a string representing a color from a color ramp
color_hsl	returns a string representation of a color based on its hue, saturation, and lightness attributes
color_hsla	returns a string representation of a color based on its hue, saturation, lightness and alpha (transparency) attributes
color_hsv	returns a string representation of a color based on its hue, saturation, and value attributes
color_hsva	returns a string representation of a color based on its hue, saturation, value and alpha (transparency) attributes
color_cmyk	returns a string representation of a color based on its cyan, magenta, yellow and black components
color_cmyka	returns a string representation of a color based on its cyan, magenta, yellow, black and alpha (transparency) components

Geometry Functions

This group contains functions that operate on geometry objects (e.g., length, area).

\$geometry	returns the geometry of the current feature (can be used for processing with other functions)
\$area	returns the area size of the current feature
\$length	returns the length size of the current feature
\$perimeter	returns the perimeter length of the current feature
\$x	returns the x coordinate of the current feature
\$y	returns the y coordinate of the current feature
xat	retrieves the nth x coordinate of the current feature. n given as a parameter of the function
yat	retrieves the nth y coordinate of the current feature. n given as a parameter of the function
xmin	returns the minimum x coordinate of a geometry. Calculations are in the Spatial Reference System of this Geometry
xmax	returns the maximum x coordinate of a geometry. Calculations are in the Spatial Reference System of this Geometry
ymin	returns the minimum y coordinate of a geometry. Calculations are in the Spatial Reference System of this Geometry
ymax	returns the maximum y coordinate of a geometry. Calculations are in the Spatial Reference System of this Geometry
geomFromWKT	returns a geometry created from a well-known text (WKT) representation
geomFromGML	returns a geometry from a GML representation of geometry
bbox	
disjoint	returns 1 if the geometries do not share any space together
intersects	returns 1 if the geometries spatially intersect (share any portion of space) and 0 if they don't
touches	returns 1 if the geometries have at least one point in common, but their interiors do not intersect
crosses	returns 1 if the supplied geometries have some, but not all, interior points in common
contains	returns true if and only if no points of b lie in the exterior of a, and at least one point of the interior of b lies in the interior of a
overlaps	returns 1 if the geometries share space, are of the same dimension, but are not completely contained by each other
within	returns 1 if geometry a is completely inside geometry b
buffer	returns a geometry that represents all points whose distance from this geometry is less than or equal to distance
centroid	returns the geometric center of a geometry
bounds	returns a geometry which represents the bounding box of an input geometry. Calculations are in the Spatial Reference System of this Geometry.
bounds_width	returns the width of the bounding box of a geometry. Calculations are in the Spatial Reference System of this Geometry.
bounds_height	returns the height of the bounding box of a geometry. Calculations are in the Spatial Reference System of this Geometry.
convexHull	returns the convex hull of a geometry (this represents the minimum convex geometry that encloses all geometries within the set)
difference	returns a geometry that represents that part of geometry

	a that does not intersect with geometry b
distance	returns the minimum distance (based on spatial ref) between two geometries in projected units
intersection	returns a geometry that represents the shared portion of geometry a and geometry b
symDifference	returns a geometry that represents the portions of a and b that do not intersect
combine	returns the combination of geometry a and geometry b
union	returns a geometry that represents the point set union of the geometries
geomToWKT	returns the well-known text (WKT) representation of the geometry without SRID metadata

Record Functions

This group contains functions that operate on record identifiers.

\$rownum	returns the number of the current row
\$id	returns the feature id of the current row
\$currentfeature	returns the current feature being evaluated. This can be used with the 'attribute' function to evaluate attribute values from the current feature.
\$scale	returns the current scale of the map canvas
\$uuid	generates a Universally Unique Identifier (UUID) for each row. Each UUID is 38 characters long.
getFeature	returns the first feature of a layer matching a given attribute value.
attribute	returns the value of a specified attribute from a feature.
\$map	returns the id of the current map item if the map is being drawn in a composition, or "canvas" if the map is being drawn within the main QGIS window.

Fields and Values

Contains a list of fields from the layer. Sample values can also be accessed via right-click.

Select the field name from the list, then right-click to access a context menu with options to load sample values from the selected field.

Fields name should be double-quoted. Values or string should be simple-quoted.

.

12.5 Modifica

QGIS supports various capabilities for editing OGR, SpatiaLite, PostGIS, MSSQL Spatial and Oracle Spatial vector layers and tables.

Nota: The procedure for editing GRASS layers is different - see section *Digitalizzare e modificare layer vettoriali GRASS* for details.

Suggerimento: Modifiche concorrenti

This version of QGIS does not track if somebody else is editing a feature at the same time as you are. The last person to save their edits wins.

12.5.1 Settare la tolleranza dello snapping e il raggio di ricerca degli elementi

Prima di editare vertici, è molto importante sia impostare il livello di snapping che il valore del raggio di ricerca al fine di gestire in maniera ottimale la modifica delle geometrie di un layer vettoriale.

Tolleranza di snapping

Snapping tolerance is the distance QGIS uses to search for the closest vertex and/or segment you are trying to connect to when you set a new vertex or move an existing vertex. If you aren't within the snapping tolerance, QGIS will leave the vertex where you release the mouse button, instead of snapping it to an existing vertex and/or segment. The snapping tolerance setting affects all tools that work with tolerance.

1. A general, project-wide snapping tolerance can be defined by choosing *Settings* →  *Options*. On Mac, go to *QGIS* →  *Preferences...* On Linux: *Edit* →  *Options*. In the *Digitizing* tab, you can select between 'to vertex', 'to segment' or 'to vertex and segment' as default snap mode. You can also define a default snapping tolerance and a search radius for vertex edits. The tolerance can be set either in map units or in pixels. The advantage of choosing pixels is that the snapping tolerance doesn't have to be changed after zoom operations. In our small digitizing project (working with the Alaska dataset), we define the snapping units in feet. Your results may vary, but something on the order of 300 ft at a scale of 1:10000 should be a reasonable setting.
2. A layer-based snapping tolerance can be defined by choosing *Settings* → (or *File* →) *Snapping options...* to enable and adjust snapping mode and tolerance on a layer basis (see [figure_edit_1](#)).

Note that this layer-based snapping overrides the global snapping option set in the *Digitizing* tab. So, if you need to edit one layer and snap its vertices to another layer, then enable snapping only on the snap to layer, then decrease the global snapping tolerance to a smaller value. Furthermore, snapping will never occur to a layer that is not checked in the snapping options dialog, regardless of the global snapping tolerance. So be sure to mark the checkbox for those layers that you need to snap to.

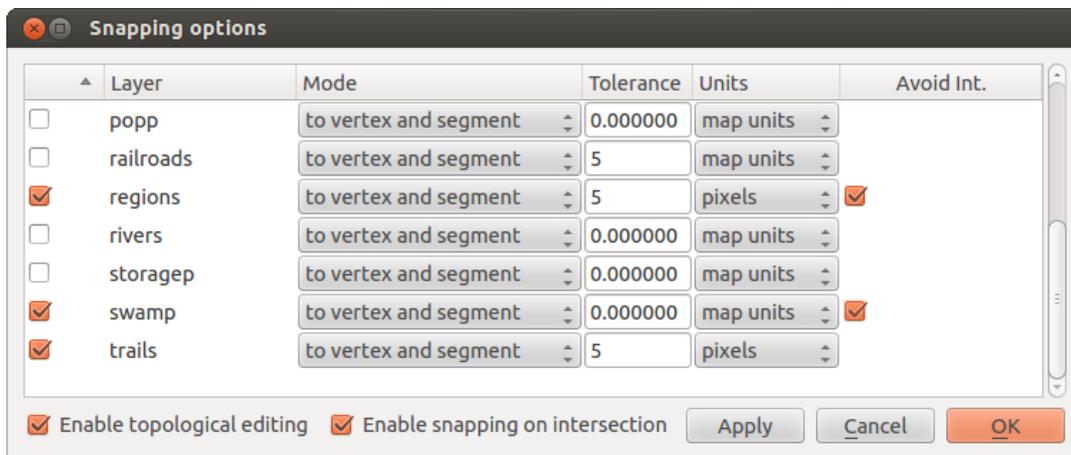


Figura 12.38: Edit snapping options on a layer basis 

Raggio di ricerca

Search radius is the distance QGIS uses to search for the closest vertex you are trying to move when you click on the map. If you aren't within the search radius, QGIS won't find and select any vertex for editing, and it will pop up an annoying warning to that effect. Snap tolerance and search radius are set in map units or pixels, so you may find you need to experiment to get them set right. If you specify too big of a tolerance, QGIS may snap to the wrong vertex, especially if you are dealing with a large number of vertices in close proximity. Set search radius too small, and it won't find anything to move.

The search radius for vertex edits in layer units can be defined in the *Digitizing* tab under *Settings* →  *Options*. This is the same place where you define the general, project- wide snapping tolerance.

12.5.2 Zoom e spostamento

Prima di editare un layer sarebbe opportuno ingrandire la vista mappa su un'area di interesse, al fine di evitare una lunga attesa per la visualizzazione di tutti i vertici della mappa.

Oltre ad utilizzare le icone  Sposta mappa e  Ingrandisci /  Rimpicciolisci è possibile interagire con la mappa con la rotellina del mouse, la barra spaziatrice e i tasti freccia della tastiera.

Zoom e spostamento con la rotella del mouse

While digitizing, you can press the mouse wheel to pan inside of the main window, and you can roll the mouse wheel to zoom in and out on the map. For zooming, place the mouse cursor inside the map area and roll it forward (away from you) to zoom in and backwards (towards you) to zoom out. The mouse cursor position will be the center of the zoomed area of interest. You can customize the behavior of the mouse wheel zoom using the *Map tools* tab under the *Settings* →  *Options* menu.

Spostamento con i tasti freccia

Panning the map during digitizing is possible with the arrow keys. Place the mouse cursor inside the map area, and click on the right arrow key to pan east, left arrow key to pan west, up arrow key to pan north, and down arrow key to pan south.

You can also use the space bar to temporarily cause mouse movements to pan the map. The PgUp and PgDown keys on your keyboard will cause the map display to zoom in or out without interrupting your digitizing session.

12.5.3 Modifiche topologiche

Besides layer-based snapping options, you can also define topological functionalities in the *Snapping options...* dialog in the *Settings* (or *File*) menu. Here, you can define *Enable topological editing*, and/or for polygon layers, you can activate the column *Avoid Int.*, which avoids intersection of new polygons.

Abilitare la modifica topologica

The option *Enable topological editing* is for editing and maintaining common boundaries in polygon mosaics. QGIS 'detects' a shared boundary in a polygon mosaic, so you only have to move the vertex once, and QGIS will take care of updating the other boundary.

Evitare le intersezioni per i nuovi poligoni

The second topological option in the *Avoid Int.* column, called *Avoid intersections of new polygons*, avoids overlaps in polygon mosaics. It is for quicker digitizing of adjacent polygons. If you already have one polygon, it is possible with this option to digitize the second one such that both intersect, and QGIS then cuts the second polygon to the common boundary. The advantage is that you don't have to digitize all vertices of the common boundary.

Enable snapping on intersections

Another option is to use *Enable snapping on intersection*. It allows you to snap on an intersection of background layers, even if there's no vertex on the intersection.

12.5.4 Modifica di un layer esistente

By default, QGIS loads layers read-only. This is a safeguard to avoid accidentally editing a layer if there is a slip of the mouse. However, you can choose to edit any layer as long as the data provider supports it, and the underlying data source is writable (i.e., its files are not read-only).

In general, tools for editing vector layers are divided into a digitizing and an advanced digitizing toolbar, described in section *Digitalizzazione avanzata*. You can select and unselect both under *View* → *Toolbars* →. Using the basic digitizing tools, you can perform the following functions:

Icona	Azione	Icona	Azione
	Current edits		Attiva modifica
	Aggiunge elementi: Inserisci punto		Aggiunge elementi: Inserisci linea
	Aggiunge elementi: Inserisci poligono		Muove elementi
	Strumento vertici		Elimina elementi selezionati
	Taglia elementi		Copia elementi
	Incolla elementi		Save layer edits

Strumenti di base per la modifica di layer vettoriali

All editing sessions start by choosing the  *Toggle editing* option. This can be found in the context menu after right clicking on the legend entry for a given layer.

Alternatively, you can use the *Toggle Editing*  *Toggle editing* button from the digitizing toolbar to start or stop the editing mode. Once the layer is in edit mode, markers will appear at the vertices, and additional tool buttons on the editing toolbar will become available.

Suggerimento: Salvataggio ad intervalli regolari

Remember to  *Save Layer Edits* regularly. This will also check that your data source can accept all the changes.

Aggiungere elementi

È possibile usare gli strumenti  *Inserisci punto*,  *Inserisci linea* o  *Inserisci poligono* per porre il puntatore di QGIS in modalità digitalizzazione.

Per ogni elemento, bisogna dapprima digitalizzare la geometria e successivamente inserire gli attributi. Per digitalizzare la geometria, cliccare con il tasto sinistro del mouse nella vista mappa per creare il primo punto del nuovo elemento.

Per linee e poligoni, continuare a cliccare con il tasto sinistro per ogni ulteriore vertice che si desidera inserire. Quando è terminato l'inserimento dei vertici o dei punti, cliccare con il tasto destro in qualunque punto della mappa per confermare di aver terminato l'inserimento della geometria dell'elemento.

The attribute window will appear, allowing you to enter the information for the new feature. [Figure_edit_2](#) shows setting attributes for a fictitious new river in Alaska. In the *Digitizing* menu under the *Settings* → *Options* menu, you can also activate *Suppress attributes pop-up windows after each created feature* and *Reuse last entered attribute values*.

With the  *Move Feature(s)* icon on the toolbar, you can move existing features.

Suggerimento: Tipologie di attributo

For editing, the attribute types are validated during entry. Because of this, it is not possible to enter a number into a text column in the dialog *Enter Attribute Values* or vice versa. If you need to do so, you should edit the attributes in a second step within the *Attribute table* dialog.

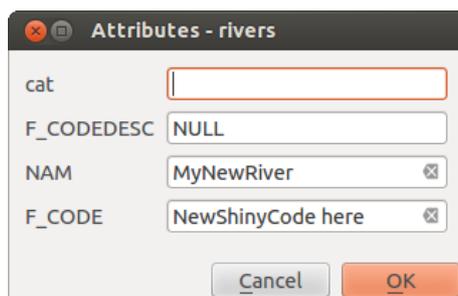


Figura 12.39: Enter Attribute Values Dialog after digitizing a new vector feature 🐧

Current Edits

This feature allows the digitization of multiple layers. Choose  *Save for Selected Layers* to save all changes you made in multiple layers. You also have the opportunity to  *Rollback for Selected Layers*, so that the digitization may be withdrawn for all selected layers. If you want to stop editing the selected layers,  *Cancel for Selected Layer(s)* is an easy way.

The same functions are available for editing all layers of the project.

Strumento vertici

For shapefile-based layers as well as SpatialLite, PostgreSQL/PostGIS, MSSQL Spatial, and Oracle Spatial tables, the  *Node Tool* provides manipulation capabilities of feature vertices similar to CAD programs. It is possible to simply select multiple vertices at once and to move, add or delete them altogether. The node tool also works with ‘on the fly’ projection turned on, and it supports the topological editing feature. This tool is, unlike other tools in QGIS, persistent, so when some operation is done, selection stays active for this feature and tool. If the node tool is unable to find any features, a warning will be displayed.

It is important to set the property *Settings* →  *Options* → *Digitizing* → *Search Radius*: to a number greater than zero (i.e., 10). Otherwise, QGIS will not be able to tell which vertex is being edited.

Suggerimento: Indicatori dei vertici

The current version of QGIS supports three kinds of vertex markers: ‘Semi-transparent circle’, ‘Cross’ and ‘None’. To change the marker style, choose  *Options* from the *Settings* menu, click on the *Digitizing* tab and select the appropriate entry.

Operazioni di base

Attivare lo strumento  *Strumento vertici* e selezionare un elemento cliccandoci sopra: un riquadro rosso apparirà su ogni vertice dell’elemento.

- **Selecting vertices:** You can select vertices by clicking on them one at a time, by clicking on an edge to select the vertices at both ends, or by clicking and dragging a rectangle around some vertices. When a vertex is selected, its color changes to blue. To add more vertices to the current selection, hold down the `Ctrl` key while clicking. Hold down `Ctrl` or `Shift` when clicking to toggle the selection state of vertices (vertices that are currently unselected will be selected as usual, but also vertices that are already selected will become unselected).
- **Adding vertices:** To add a vertex, simply double click near an edge and a new vertex will appear on the edge near to the cursor. Note that the vertex will appear on the edge, not at the cursor position; therefore, it should be moved if necessary.

- **Eliminare vertici:** per eliminare un vertice selezionato basta premere il tasto `Delete`. Si noti che non è possibile eliminare un intero elemento tramite lo strumento  Strumento vertici; QGIS manterrà un numero minimo di vertici per il tipo di elemento su cui si sta lavorando. Per eliminare completamente un elemento, usare lo strumento  Elimina il selezionato.
- **Spostare vertici:** selezionare i vertici di interesse, quindi cliccare e trascinare nella direzione verso la quale si intende spostare i vertici; i vertici saranno spostati tutti insieme. Se lo snap è attivo, l'intera selezione può essere agganciata al vertice e/o linea più vicina.

Each change made with the node tool is stored as a separate entry in the Undo dialog. Remember that all operations support topological editing when this is turned on. On-the-fly projection is also supported, and the node tool provides tooltips to identify a vertex by hovering the pointer over it.

Tagliare, copiare ed incollare elementi

Gli elementi selezionati possono essere tagliati, copiati ed incollati tra layer dello stesso progetto di QGIS a patto che anche per il layer di destinazione sia stata abilitata la modalità di modifica tramite l'opzione  Modifica.

Features can also be pasted to external applications as text. That is, the features are represented in CSV format, with the geometry data appearing in the OGC Well-Known Text (WKT) format.

However, in this version of QGIS, text features from outside QGIS cannot be pasted to a layer within QGIS. When would the copy and paste function come in handy? Well, it turns out that you can edit more than one layer at a time and copy/paste features between layers. Why would we want to do this? Say we need to do some work on a new layer but only need one or two lakes, not the 5,000 on our `big_lakes` layer. We can create a new layer and use copy/paste to plop the needed lakes into it.

As an example, we will copy some lakes to a new layer:

1. Caricare il layer dal quale vogliamo copiare gli elementi (layer sorgente)
2. Caricare o creare il layer nel quale vogliamo incollare gli elementi copiati (layer di destinazione)
3. Impostare entrambi i layer in modalità modifica
4. Rendere attivo il layer sorgente cliccando sul relativo nome nella legenda
5. Attivare lo strumento  Seleziona il singolo elemento per selezionare gli elementi dal layer sorgente
6. Cliccare sullo strumento  Copia elementi
7. Rendere attivo il layer di destinazione cliccando sul relativo nome nella legenda
8. Attivare lo strumento  Incolla elementi
9. Terminare le modifiche e salvare

What happens if the source and target layers have different schemas (field names and types are not the same)? QGIS populates what matches and ignores the rest. If you don't care about the attributes being copied to the target layer, it doesn't matter how you design the fields and data types. If you want to make sure everything - the feature and its attributes - gets copied, make sure the schemas match.

Suggerimento: Congruenza degli elementi incollati

If your source and destination layers use the same projection, then the pasted features will have geometry identical to the source layer. However, if the destination layer is a different projection, then QGIS cannot guarantee the geometry is identical. This is simply because there are small rounding-off errors involved when converting between projections.

Cancellare elementi selezionati

Se si vuole eliminare un intero poligono, è possibile farlo selezionando dapprima l'elemento che intendiamo cancellare con lo strumento  Seleziona il singolo elemento: è possibile anche selezionare più poligoni contemporaneamente. Una volta definita la selezione, usare lo strumento  Elimina il selezionato per cancellare la selezione.

The  Cut Features tool on the digitizing toolbar can also be used to delete features. This effectively deletes the feature but also places it on a “spatial clipboard”. So, we cut the feature to delete. We could then use the  Paste Features tool to put it back, giving us a one-level undo capability. Cut, copy, and paste work on the currently selected features, meaning we can operate on more than one at a time.

Salvare i layer modificati

When a layer is in editing mode, any changes remain in the memory of QGIS. Therefore, they are not committed/saved immediately to the data source or disk. If you want to save edits to the current layer but want to continue editing without leaving the editing mode, you can click the  Save Layer Edits button. When you turn editing mode off with  Toggle editing (or quit QGIS for that matter), you are also asked if you want to save your changes or discard them.

If the changes cannot be saved (e.g., disk full, or the attributes have values that are out of range), the QGIS in-memory state is preserved. This allows you to adjust your edits and try again.

Suggerimento: Integrità dei dati

È buona norma fare un back-up del dato originale prima di procedere alla modifica. Per quando siano stati fatti molti sforzi da parte dei programmatori di QGIS per preservare l'integrità del dato, non vi è alcuna garanzia che ciò avvenga.

12.5.5 Digitalizzazione avanzata

Icona	Azione	Icona	Azione
	Annulla		Ripristina
	Ruota elemento/i		Semplifica geometrie
	Aggiungi buco		Aggiungi una parte
	Fill Ring		Elimina buco
	Elimina parte		Modifica la forma
	Curva di offset		Spezza elemento
	Split Parts		Unisci le geometrie selezionate
	Unisci gli attributi delle geometrie selezionate		Ruota i simboli per i punti

Barra degli strumenti di digitalizzazione avanzata

Annulare e ripristinare

The  Undo and  Redo tools allows you to undo or redo vector editing operations. There is also a dockable widget, which shows all operations in the undo/redo history (see [Figure_edit_3](#)). This widget is not displayed by

default; it can be displayed by right clicking on the toolbar and activating the Undo/Redo checkbox. Undo/Redo is however active, even if the widget is not displayed.

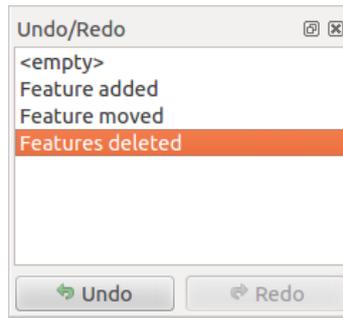


Figura 12.40: Redo and Undo digitizing steps 

When Undo is hit, the state of all features and attributes are reverted to the state before the reverted operation happened. Changes other than normal vector editing operations (for example, changes done by a plugin), may or may not be reverted, depending on how the changes were performed.

To use the undo/redo history widget, simply click to select an operation in the history list. All features will be reverted to the state they were in after the selected operation.

Ruota elemento/i

Use  Rotate Feature(s) to rotate one or multiple selected features in the map canvas. You first need to select the features and then press the  Rotate Feature(s) icon. The centroid of the feature(s) appears and will be the rotation anchor point. If you selected multiple features, the rotation anchor point will be the common center of the features. Press and drag the left mouse button in the desired direction to rotate the selected features.

It's also possible to create a user-defined rotation anchor point around which the selected feature will rotate. Select the features to rotate and activate the  Rotate Feature(s) tool. Press and hold the `Ctrl` button and move the mouse pointer (without pressing the mouse button) to the place where you want the rotation anchor to be moved. Release the `Ctrl` button when the desired rotation anchor point is reached. Now, press and drag the left mouse button in the desired direction to rotate the selected feature(s).

Semplifica geometrie

The  Simplify Feature tool allows you to reduce the number of vertices of a feature, as long as the geometry doesn't change and geometry type is not a multi geometry. First, select a feature. It will be highlighted by a red rubber band and a slider will appear. Moving the slider, the red rubber band will change its shape to show how the feature is being simplified. Click **[OK]** to store the new, simplified geometry. If a feature cannot be simplified (e.g. multi-polygons), a message will appear.

Aggiungi buco

You can create ring polygons using the  Add Ring icon in the toolbar. This means that inside an existing area, it is possible to digitize further polygons that will occur as a 'hole', so only the area between the boundaries of the outer and inner polygons remains as a ring polygon.

Aggiungi una parte

You can  add part polygons to a selected multipolygon. The new part polygon must be digitized outside the selected multi-polygon.

Fill Ring

You can use the  Fill Ring function to add a ring to a polygon and add a new feature to the layer at the same time.

Thus you need not first use the  Add Ring icon and then the  Add feature function anymore.

Elimina buco

The  Delete Ring tool allows you to delete ring polygons inside an existing area. This tool only works with polygon layers. It doesn't change anything when it is used on the outer ring of the polygon. This tool can be used on polygon and multi-polygon features. Before you select the vertices of a ring, adjust the vertex edit tolerance.

Elimina parte

The  Delete Part tool allows you to delete parts from multifeatures (e.g., to delete polygons from a multi-polygon feature). It won't delete the last part of the feature; this last part will stay untouched. This tool works with all multi-part geometries: point, line and polygon. Before you select the vertices of a part, adjust the vertex edit tolerance.

Modifica la forma

You can reshape line and polygon features using the  Reshape Features icon on the toolbar. It replaces the line or polygon part from the first to the last intersection with the original line. With polygons, this can sometimes lead to unintended results. It is mainly useful to replace smaller parts of a polygon, not for major overhauls, and the reshape line is not allowed to cross several polygon rings, as this would generate an invalid polygon.

Per modificare, ad esempio, il bordo di un poligono basta cliccare un primo punto all'interno del poligono, cliccare un secondo punto all'esterno del poligono, tracciare il profilo della nuova forma, rientrare nel poligono e cliccare con il tasto destro del mouse per terminare l'operazione. Lo strumento aggiungerà automaticamente nuovi nodi laddove la nuova linea interseca il bordo del poligono. È, inoltre, possibile rimuovere parte di un poligono iniziando la nuova linea all'esterno del poligono, aggiungendo vertici all'interno e terminando la linea all'esterno con il tasto destro del mouse.

Nota: The reshape tool may alter the starting position of a polygon ring or a closed line. So, the point that is represented 'twice' will not be the same any more. This may not be a problem for most applications, but it is something to consider.

Dividere elementi

The  Offset Curve tool creates parallel shifts of line layers. The tool can be applied to the edited layer (the geometries are modified) or also to background layers (in which case it creates copies of the lines / rings and adds them to the the edited layer). It is thus ideally suited for the creation of distance line layers. The displacement is shown at the bottom left of the taskbar.

To create a shift of a line layer, you must first go into editing mode and then select the feature. You can make the  Offset Curve tool active and drag the cross to the desired distance. Your changes may then be saved with the  Save Layer Edits tool.

QGIS options dialog (Digitizing tab then **Curve offset tools** section) allows you to configure some parameters like **Join style**, **Quadrant segments**, **Miter limit**.

Spezza elemento

È possibile dividere degli elementi tramite lo strumento  Spezza elemento e tracciando una linea attraverso l'elemento di interesse.

Split parts

In QGIS 2.0 it is now possible to split the parts of a multi part feature so that the number of parts is increased. Just draw a line across the part you want to split using the  Split Parts icon.

Unire elementi

The  Merge Selected Features tool allows you to merge features that have common boundaries. A new dialog will allow you to choose which value to choose between each selected features or select a function (Minimum, Maximum, Median, Sum, Skip Attribute) to use for each column.

Unire attributi di elementi

The  Merge Attributes of Selected Features tool allows you to merge attributes of features with common boundaries and attributes without merging their boundaries. First, select several features at once. Then press the  Merge Attributes of Selected Features button. Now QGIS asks you which attributes are to be applied to all selected objects. As a result, all selected objects have the same attribute entries.

Ruota i simboli per i punti

 Rotate Point Symbols allows you to change the rotation of point symbols in the map canvas. You must first define a rotation column from the attribute table of the point layer in the *Advanced* menu of the *Style* menu of the *Layer Properties*. Also, you will need to go into the 'SVG marker' and choose *Data defined properties Activate* *Angle* and choose 'rotation' as field. Without these settings, the tool is inactive.

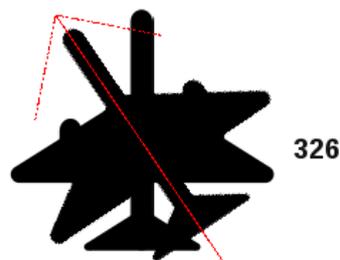


Figura 12.41: Rotate Point Symbols 

To change the rotation, select a point feature in the map canvas and rotate it, holding the left mouse button pressed. A red arrow with the rotation value will be visualized (see [Figure_edit_4](#)). When you release the left mouse button again, the value will be updated in the attribute table.

Nota: Se si tiene premuto il tasto `Ctrl`, la rotazione avverrà per step di 15 gradi.

12.5.6 Creating new Vector layers

QGIS allows you to create new shapefile layers, new Spatialite layers, and new GPX layers. Creation of a new GRASS layer is supported within the GRASS plugin. Please refer to section [Creare un nuovo layer vettoriale GRASS](#) for more information on creating GRASS vector layers.

Creare un nuovo Shapefile

To create a new shape layer for editing, choose *New* →  *New Shapefile Layer...* from the *Layer* menu. The *New Vector Layer* dialog will be displayed as shown in [Figure_edit_5](#). Choose the type of layer (point, line or polygon) and the CRS (coordinate reference system).

Note that QGIS does not yet support creation of 2.5D features (i.e., features with X,Y,Z coordinates).

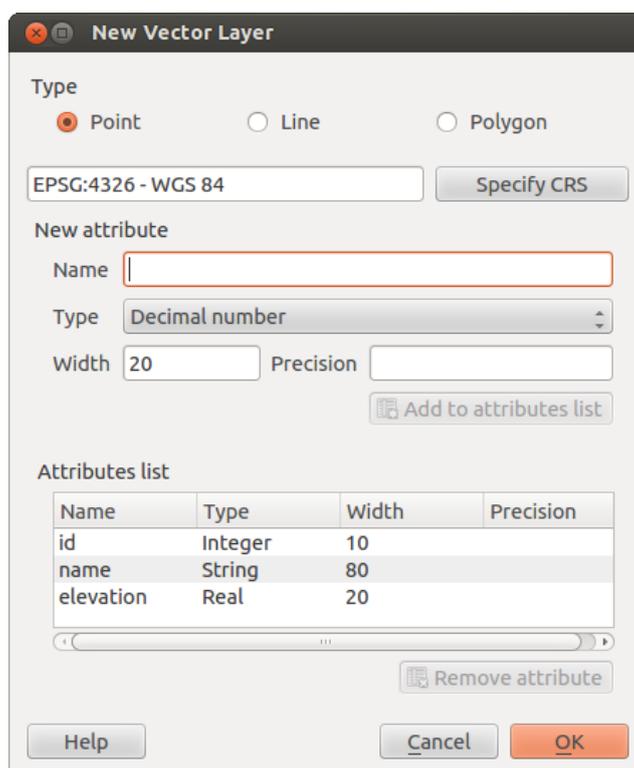


Figura 12.42: Creating a new Shapefile layer Dialog 

To complete the creation of the new shapefile layer, add the desired attributes by clicking on the **[Add to attributes list]** button and specifying a name and type for the attribute. A first 'id' column is added as default but can be removed, if not wanted. Only *Type: real* , *Type: integer* , *Type: string*  and *Type: date*  attributes are supported. Additionally and according to the attribute type, you can also define the width and precision of the new attribute column. Once you are happy with the attributes, click **[OK]** and provide a name for the shapefile. QGIS will automatically add a `.shp` extension to the name you specify. Once the layer has been created, it will be added to the map, and you can edit it in the same way as described in section [Modifica di un layer esistente](#) above.

Creare un nuovo layer SpatiaLite

To create a new SpatiaLite layer for editing, choose *New* →  *New SpatiaLite Layer...* from the *Layer* menu. The *New SpatiaLite Layer* dialog will be displayed as shown in [Figure_edit_6](#).

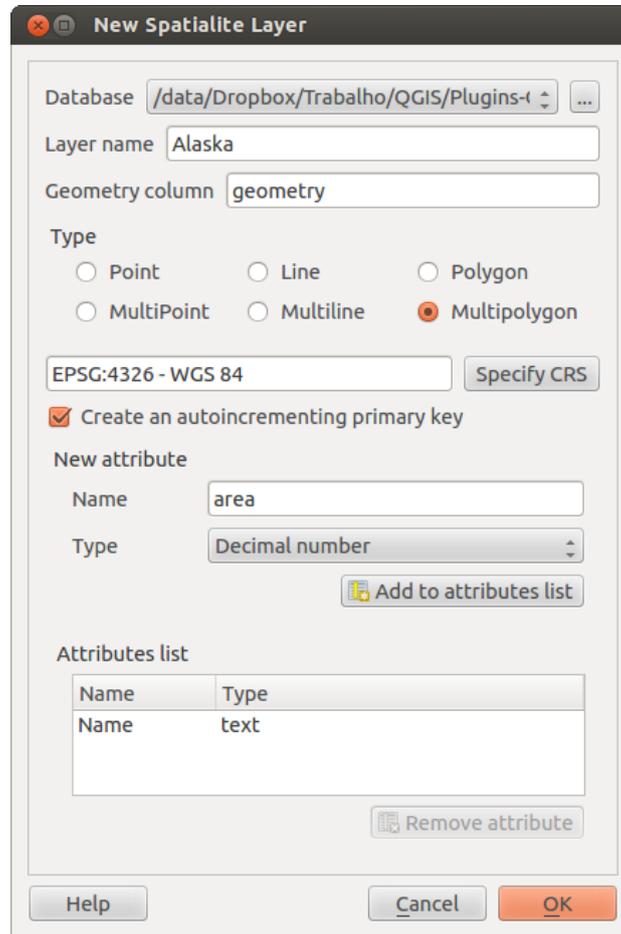


Figura 12.43: Creating a New SpatiaLite layer Dialog 

The first step is to select an existing SpatiaLite database or to create a new SpatiaLite database. This can be done with the browse button  to the right of the database field. Then, add a name for the new layer, define the layer type, and specify the coordinate reference system with [**Specify CRS**]. If desired, you can select *Create an autoincrementing primary key*.

To define an attribute table for the new SpatiaLite layer, add the names of the attribute columns you want to create with the corresponding column type, and click on the [**Add to attribute list**] button. Once you are happy with the attributes, click [**OK**]. QGIS will automatically add the new layer to the legend, and you can edit it in the same way as described in section *Modifica di un layer esistente* above.

Further management of SpatiaLite layers can be done with the DB Manager. See *Plugin DB Manager*.

Creating a new GPX layer

To create a new GPX file, you need to load the GPS plugin first. *Plugins* →  *Plugin Manager...* opens the Plugin Manager Dialog. Activate the *GPS Tools* checkbox.

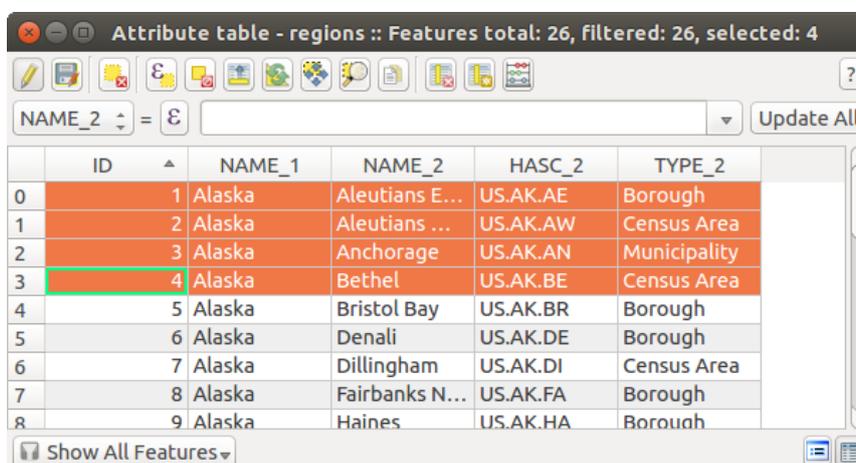
When this plugin is loaded, choose *New* →  *Create new GPX Layer...* from the *Layer* menu. In the *Save new GPX file as* dialog, you can choose where to save the new GPX layer.

12.5.7 Lavorare con la tabella degli attributi

The attribute table displays features of a selected layer. Each row in the table represents one map feature, and each column contains a particular piece of information about the feature. Features in the table can be searched, selected, moved or even edited.

To open the attribute table for a vector layer, make the layer active by clicking on it in the map legend area. Then, from the main *Layer* menu, choose  *Open Attribute Table*. It is also possible to right click on the layer and choose  *Open Attribute Table* from the drop-down menu, and to click on the  *Open Attribute Table* button in the Attributes toolbar.

This will open a new window that displays the feature attributes for the layer (*figure_attributes_1*). The number of features and the number of selected features are shown in the attribute table title.



ID	NAME_1	NAME_2	HASC_2	TYPE_2	
0	1	Alaska	Aleutians E...	US.AK.AE	Borough
1	2	Alaska	Aleutians ...	US.AK.AW	Census Area
2	3	Alaska	Anchorage	US.AK.AN	Municipality
3	4	Alaska	Bethel	US.AK.BE	Census Area
4	5	Alaska	Bristol Bay	US.AK.BR	Borough
5	6	Alaska	Denali	US.AK.DE	Borough
6	7	Alaska	Dillingham	US.AK.DI	Census Area
7	8	Alaska	Fairbanks N...	US.AK.FA	Borough
8	9	Alaska	Haines	US.AK.HA	Borough

Figura 12.44: Attribute Table for regions layer 

Selezionare elementi nella tabella degli attributi

Ogni riga selezionata nella tabella degli attributi mostra gli attributi di un elemento selezionato nel layer. Se l'insieme di elementi selezionati nella vista mappa viene modificato, la selezione viene aggiornata anche nella tabella e viceversa. Allo stesso modo, se cambia l'insieme delle righe selezionate nella tabella degli attributi, cambierà anche la visualizzazione degli elementi nella vista mappa.

Le righe possono essere selezionate cliccando sul numero alla loro sinistra. Si possono selezionare **righe multiple** tenendo premuto il tasto **Ctrl**. È possibile selezionare un **sottoinsieme di righe** tenendo premuto **Shift** e cliccando sulle due righe che delimitano il sottoinsieme di interesse. Spostare il cursore del mouse e cliccare nelle celle della tabella non modifica la selezione. Cambiare la selezione della vista mappa non modifica la posizione del cursore nella tabella.

Le righe della tabella possono essere ordinate in funzione degli attributi. Cliccare sull'intestazione dell'attributo rispetto al quale si intende ordinare la tabella: una piccola freccia a destra del nome dell'attributo indicherà il verso dell'ordinamento (freccia in su per ordinamento crescente, freccia in giù per ordinamento decrescente).

For a **simple search by attributes** on only one column, choose the *Column filter* → from the menu in the bottom left corner. Select the field (column) on which the search should be performed from the drop-down menu, and hit the **[Apply]** button. Then, only the matching features are shown in the attribute table.

To make a selection, you have to use the  Select features using an Expression icon on top of the attribute table. 

Select features using an Expression allows you to define a subset of a table using a *Function List* like in the  Field Calculator (see *Calcolatore di campi*). The query result can then be saved as a new vector layer. For example, if you want to find regions that are boroughs from `regions.shp` of the QGIS sample data, you have to open the *Fields and Values* menu and choose the field that you want to query. Double-click the field 'TYPE_2' and also [**Load all unique values**]. From the list, choose and double-click 'Borough'. In the *Expression* field, the following query appears:

```
"TYPE_2" = 'Borough'
```

Here you can also use the *Function list* → *Recent (Selection)* to make a selection that you used before. The expression builder remembers the last 20 used expressions.

The matching rows will be selected, and the total number of matching rows will appear in the title bar of the attribute table, as well as in the status bar of the main window. For searches that display only selected features on the map, use the Query Builder described in section *Costruttore di interrogazioni*.

To show selected records only, use *Show Selected Features* from the menu at the bottom left.

The other buttons at the top of the attribute table window provide the following functionality:

-  Toggle editing mode to edit single values and to enable functionalities described below (also with `Ctrl+E`)
-  Save Edits (also with `Ctrl+S`)
-  Unselect all (also with `Ctrl+U`)
-  Move selected to top (also with `Ctrl+T`)
-  Invert selection (also with `Ctrl+R`)
-  Copy selected rows to clipboard (also with `Ctrl+C`)
-  Zoom map to the selected rows (also with `Ctrl+J`)
-  Pan map to the selected rows (also with `Ctrl+P`)
-  Delete selected features (also with `Ctrl+D`)
-  New Column for PostGIS layers and for OGR layers with GDAL version ≥ 1.6 (also with `Ctrl+W`)
-  Delete Column for PostGIS layers and for OGR layers with GDAL version ≥ 1.9 (also with `Ctrl+L`)
-  Open field calculator (also with `Ctrl+I`)

Below these buttons is the Field Calculator bar, which allows calculations to be quickly applied attributes visible in the table. This bar uses the same expressions as the  Field Calculator (see *Calcolatore di campi*).

Suggerimento: Geometrie WKT

If you want to use attribute data in external programs (such as Excel), use the  Copy selected rows to clipboard button. You can copy the information without vector geometries if you deactivate *Settings* → *Options* → *Data sources* menu  *Copy geometry in WKT representation from attribute table*.

Salvare elementi selezionati come nuovo layer

The selected features can be saved as any OGR-supported vector format and also transformed into another coordinate reference system (CRS). Just open the right mouse menu of the layer and click on *Save as* to define the name

of the output file, its format and CRS (see section *Legenda*). To save the selection ensure that the  *Save only selected features* is selected. It is also possible to specify OGR creation options within the dialog.

Paste into new layer

Features that are on the clipboard may be pasted into a new layer. To do this, first make a layer editable. Select some features, copy them to the clipboard, and then paste them into a new layer using *Edit* → *Paste Features as* and choosing *New vector layer* or *New memory layer*.

This applies to features selected and copied within QGIS and also to features from another source defined using well-known text (WKT).

Lavorare con tabelle di attributi non spaziali

QGIS allows you also to load non-spatial tables. This currently includes tables supported by OGR and delimited text, as well as the PostgreSQL, MSSQL and Oracle provider. The tables can be used for field lookups or just generally browsed and edited using the table view. When you load the table, you will see it in the legend field. It can be opened with the  *Open Attribute Table* tool and is then editable like any other layer attribute table.

As an example, you can use columns of the non-spatial table to define attribute values, or a range of values that are allowed, to be added to a specific vector layer during digitizing. Have a closer look at the edit widget in section *Menu Campi* to find out more.

12.5.8 Creating one to many relations

Relations are a technique often used in databases. The concept is, that features (rows) of different layers (tables) can belong to each other.

As an example you have a layer with all regions of alaska (polygon) which provides some attributes about its name and region type and a unique id (which acts as primary key).

Foreign keys

Then you get another point layer or table with information about airports that are located in the regions and you also want to keep track of these. If you want to add them to the region layer, you need to create a one to many relation using foreign keys, because there are several airports in most regions.



Figura 12.45: Alaska region with airports 

In addition to the already existing attributes in the airports attribute table another field `fk_region` which acts as a foreign key (if you have a database, you will probably want to define a constraint on it).

This field `fk_region` will always contain an id of a region. It can be seen like a pointer to the region it belongs to. And you can design a custom edit form for the editing and QGIS takes care about the setup. It works with different providers (so you can also use it with shape and csv files) and all you have to do is to tell QGIS the relations between your tables.

Layers

QGIS makes no difference between a table and a vector layer. Basically, a vector layer is a table with a geometry. So can add your table as a vector layer. To demonstrate you can load the 'region' shapefile (with geometries) and the 'airport' csv table (without geometries) and a foreign key (`fk_region`) to the layer region. This means, that each airport belongs to exactly one region while each region can have any number of airports (a typical one to many relation).

Definition (Relation Manager)

The first thing we are going to do is to let QGIS know about the relations between the layer. This is done in *Settings* → *Project Properties*. Open the *Relations* menu and click on *Add*.

- **name** is going to be used as a title. It should be a human readable string, describing, what the relation is used for. We will just call say "Airports" in this case.
- **referencing layer** is the one with the foreign key field on it. In our case this is the airports layer
- **referencing field** will say, which field points to the other layer so this is `fk_region` in this case
- **referenced layer** is the one with the primary key, pointed to, so here it is the regions layer
- **referenced field** is the primary key of the referenced layer so it is `ID`
- **id** will be used for internal purposes and has to be unique. You may need it to build custom forms once this is supported. If you leave it empty, one will be generated for you but you can assign one yourself to get one that is easier to handle.

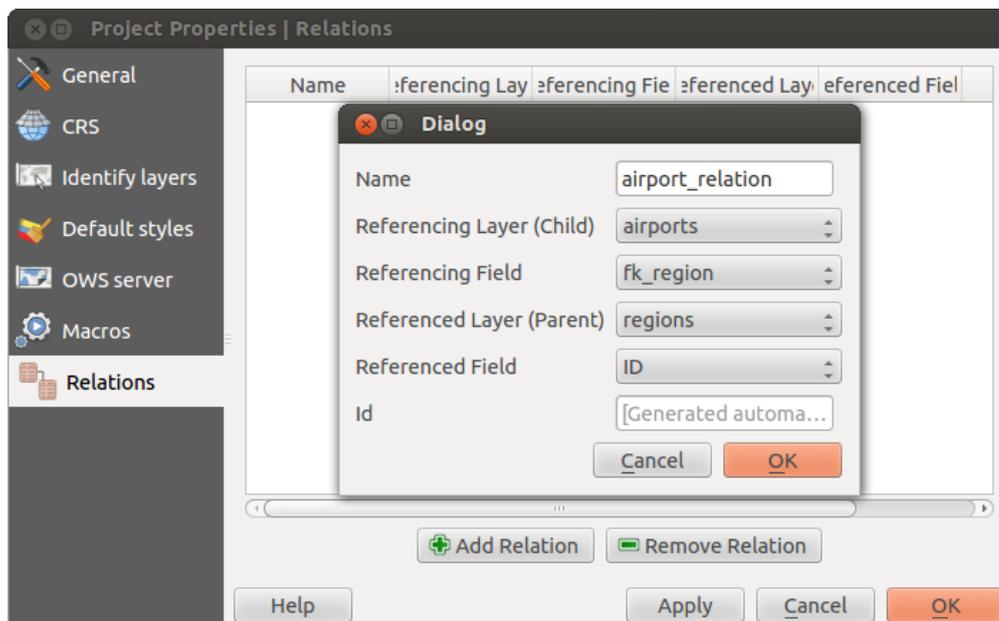


Figura 12.46: Relation Manager 

Forms

Now that QGIS knows about the relation, it will be used to improve the forms it generates. As we did not change the default form method (autogenerated) it will just add a new widget in our form. So let's select the layer region in the legend and use the identify tool. Depending on your settings, the form might open directly or you will have to choose to open it in the identification dialog under actions.

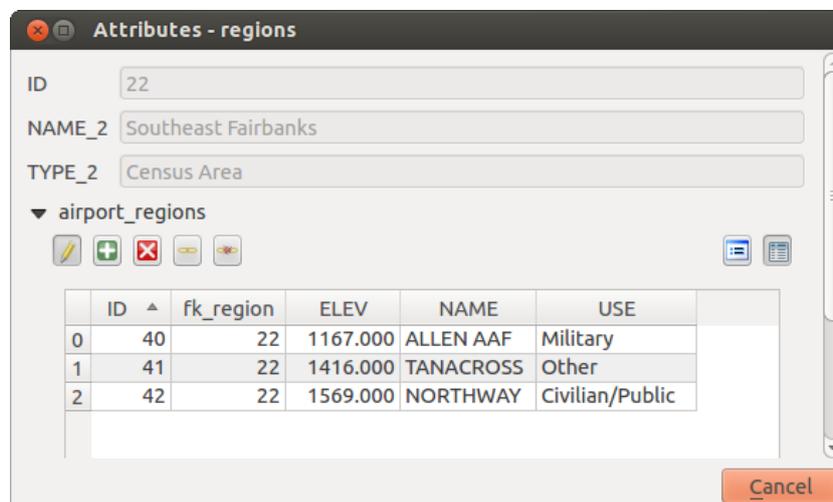


Figura 12.47: Identification dialog regions with relation to airports 🐧

As you can see, the airports assigned to this particular region are all shown in a table. And there are also some buttons available. Let's review them shortly

- The  button is for toggling the edit mode. Be aware that it toggles the edit mode of the airport layer, although we are in the feature form of a feature from the region layer. But the table is representing features of the airport layer.
- The  button will add a new feature to the airport layer. And it will assign the new airport to the current region by default.
- The  button will delete the selected airport permanently.
- The  symbol will open a new dialog where you can select any existing airport which will then be assigned to the current region. This may be handy if you created the airport on the wrong region by accident.
- The  symbol will unlink the selected airport from the current region, leaving them unassigned (the foreign key is set to NULL) effectively.
- The two buttons to the right switch between table view and form view where the later let's you view all the airports in their respective form.

If you work on the airport table, a new widget type is available which lets you embed the feature form of the referenced region on the feature form of the airports. It can be used when you open the layer properties of the airports table, switch to the *Fields* menu and change the widget type of the foreign key field 'fk_region' to Relation Reference.

If you look at the feature dialog now, you will see, that the form of the region is embedded inside the airports form and will even have a combobox, which allows you to assign the current airport to another region.

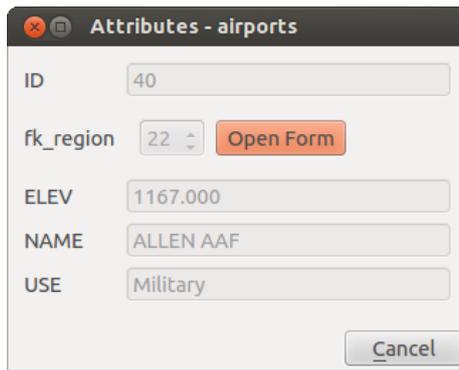


Figura 12.48: Identification dialog airport with relation to regions 

12.6 Costruttore di interrogazioni

Il costruttore di interrogazioni ti permette di creare un sottoinsieme di elementi di una tabella, usando la clausola SQL 'WHERE'. Potrai visualizzare i risultati direttamente sulla mappa oppure li potrai salvare come nuovo vettore.

12.6.1 Interrogazione

Clicca sul pulsante [**Costruttore di interrogazioni**] dal sottomenu *Sottoinsieme di geometrie* presente nel menu *Generale* delle proprietà del vettore. Se hai caricato un vettore *regions* con un campo *TYPE_2* puoi selezionare solamente le regioni appartenenti a un preciso distretto inserendo una specifica clausola nello spazio *Espressioni filtro specifiche del gestore* della finestra del costruttore delle interrogazioni. La figura [Figure_attributes_2](#) mostra un esempio di interrogazione del vettore *regions.shp* dell'insieme di dati campione di QGIS. Le sezioni dei Campi, Valori e Operatori ti aiutano ad inserire una corretta interrogazione in linguaggio SQL.

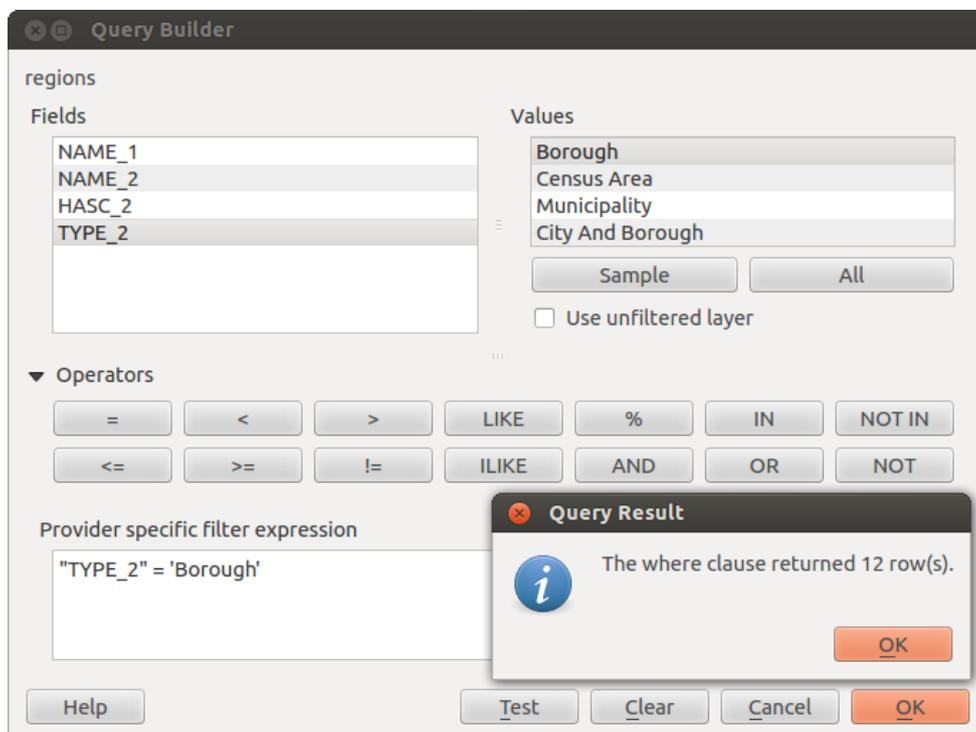


Figura 12.49: Costruttore di interrogazioni 

La sezione **Campi** elenca tutti gli attributi del vettore. Se vuoi aggiungere un attributo nella casella delle clausole SQL WHERE, fai doppio click sul nome dell'attributo, quindi usa le altre sezioni (Valori e Operatori) per completare la clausola. In alternativa puoi scrivere direttamente l'interrogazione nella sezione 'Espressioni filtro specifiche del gestore'.

La sezione **Valori** elenca tutti i valori di un attributo. Clicca il pulsante **[Tutto]** se vuoi visualizzare tutti i valori di un attributo altrimenti clicca sul pulsante **[Campione]** per caricare 25 valori univoci e casuali dell'attributo. Se vuoi aggiungere un valore nella casella delle clausole SQL WHERE fai doppio click sul suo nome nella sezione Valori.

La sezione **Operatori** elenca tutti gli operatori che puoi usare. Per aggiungere un operatore nella casella delle clausole SQL WHERE, clicca sull'operatore che vuoi usare. Sono disponibili operatori relazionali (=, >, ...), operatori per confrontare stringhe di testo (LIKE) ed operatori logici (AND, OR, ...).

Puoi usare il pulsante **[Test]** durante il processo di interrogazione, per visualizzare una finestra con il numero di elementi che soddisfano l'interrogazione effettuata. Il pulsante **[Pulisci]** cancella il testo nella finestra di interrogazione SQL WHERE. Il tasto **[OK]** chiude la finestra dell'interrogazione e seleziona gli elementi che soddisfano la ricerca. Il tasto **[Cancel]** chiude la finestra del costruttore di interrogazioni senza cambiare lo stato attuale della selezione.

QGIS si occupa dei sottoinsieme risultanti come se fosse l'intero livello. Ad esempio, se è stato attivato il filtro per 'Borough', non è possibile visualizzare, interrogare, salvare o modificare Ankorage, perché questa è un 'Municipality' e quindi non fa parte del sottoinsieme.

L'unica eccezione è che se il vostro layer è parte di un database, utilizzando un sottoinsieme vi impedisce di modificare il resto del layer.

12.7 Calcolatore di campi

The  Field Calculator button in the attribute table allows you to perform calculations on the basis of existing attribute values or defined functions, for instance, to calculate length or area of geometry features. The results can be written to a new attribute field, a virtual field, or they can be used to update values in an existing field.

Suggerimento: Virtual Fields

- Virtual fields are not permanent and are not saved.
 - To make a field virtual it must be done when the field is made.
-

The field calculator is now available on any layer that supports edit. When you click on the field calculator icon the dialog opens (see [figure_attributes_3](#)). If the layer is not in edit mode, a warning is displayed and using the field calculator will cause the layer to be put in edit mode before the calculation is made.

The quick field calculation bar in top of the attribute table is only visible if the layer is editable.

In quick field calculation bar, you first select the existing field name then open the expression dialog to create your expression or write it directly in the field then click on **Update All** button.

In the field calculator dialog, you first must select whether you want to only update selected features, create a new attribute field where the results of the calculation will be added or update an existing field.

Per aggiungere un nuovo campo devi creare il nome, il tipo di campo (intero, decimale, testo) e la lunghezza del campo. Per il tipo *Numero decimale* puoi anche definire la precisione, ovvero il numero di cifre dopo la virgola: ad esempio un campo con larghezza 10 e precisione 3 avrà 6 cifre prima della virgola, la virgola stessa è un campo ed infine 3 cifre decimali (vedi [figure_attributes_3](#)).

Questo breve esempio ti mostra come funziona il calcolatore di campi. Vogliamo calcolare la lunghezza in km del vettore `railroads` dell'insieme di dati campione di QGIS.

1. Carica lo shapefile `railroads.shp` e clicca su  Apri tabella attributi.

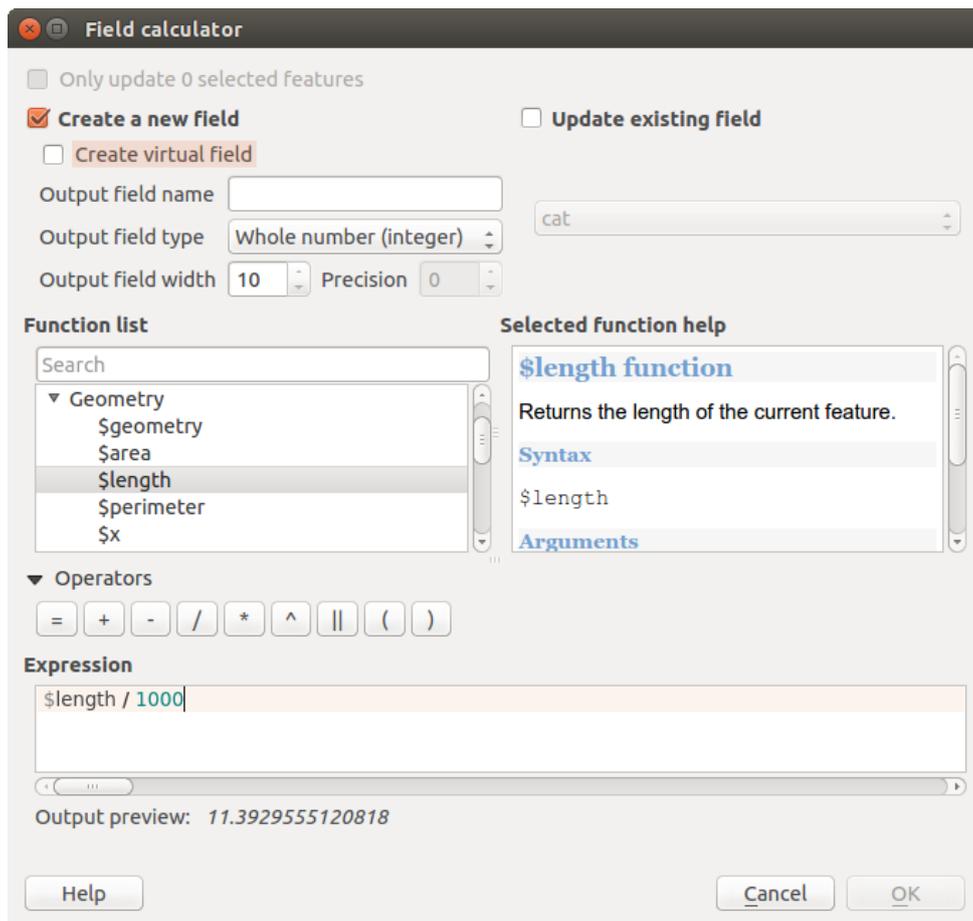


Figura 12.50: Calcolatore di campi 🐧

2. Attiva la modalità  Modifica e apri il  Calcolatore di campi .
3. Spunta la casella di controllo  *Crea un nuovo campo* per attivare la creazione di un nuovo campo.
4. Chiama il campo `length`, imposta Numero decimale (`real`) come tipo, 10 come larghezza e 3 come precisione.
5. Ora fai doppio click sulla funzione `$length` presente nel gruppo *Geometria* per aggiungerla nella sezione **Espressione**.
6. Completa l'espressione digitando `'/ 1000'` nel campo Espressione e clicca [OK].
7. You can now find a new field `length` in the attribute table.

The available functions are listed in [Expressions](#) chapter.

Lavorare con i dati raster

13.1 Lavorare con i dati raster

Questa sezione descrive come visualizzare ed impostare le proprietà dei dati raster. QGIS usa la libreria GDAL per l'accesso alla lettura/scrittura a formati raster, tipo ArcInfo Binary Grid, ArcInfo ASCII Grid, GeoTIFF, ERDAS IMAGINE e molti altri. Il supporto ai raster GRASS è garantito da un plugin nativo di QGIS. Puoi caricare i dati raster in QGIS anche in sola lettura da archivi zip e gzip.

Attualmente, la libreria GDAL supporta più di 100 formati raster (vedi GDAL-SOFTWARE-SUITE *Letteratura e riferimenti web*). La lista completa è disponibile alla pagina web http://www.gdal.org/formats_list.html.

Nota: Per diverse ragioni, non tutti i formati elencati potrebbero funzionare in QGIS. Per esempio, alcuni formati richiedono la presenza di librerie commerciali di terze parti oppure l'installazione di GDAL potrebbe essere avvenuta senza il supporto al formato che intendi usare. Quando carichi un raster in QGIS, solo i formati ben testati appariranno nell'elenco dei tipi di file; altri formati non testati possono essere caricati selezionando l'opzione [GDAL] Tutti i file (*).

Per caricare e lavorare con dati raster di GRASS, fai riferimento alla sezione *Integrazione con GRASS GIS*.

13.1.1 Cosa sono i dati raster?

I dati raster sono matrici di celle discrete che rappresentano elementi della superficie terrestre o dell'ambiente al di sopra o al di sotto di essa. Ogni cella nella matrice ha la stessa dimensione e le celle sono solitamente rettangolari (in QGIS sono sempre rettangolari). Esempi tipici di dati raster sono quelli provenienti dal telerilevamento come le fotografie aeree, le immagini da satellite e dati modellati come le matrici dell'elevazione.

Diversamente dai vettori, i dati raster di solito non hanno associato un database contenente i dati descrittivi di ogni cella e sono geocodificati in base alla risoluzione del pixel e alle coordinate x/y di un angolo del raster. Questo permette a QGIS di posizionare correttamente il dato sulla mappa.

Per posizionare e visualizzare correttamente un raster, QGIS legge le informazioni di georeferenziazione incorporate direttamente nel file raster (ad es. GeoTiff) o gestite da un apposito file (world file).

13.1.2 Caricare dati raster in QGIS

Puoi caricare i raster cliccando sull'icona  Aggiungi raster oppure selezionando l'opzione *Layer* →  *Aggiungi raster*. Puoi caricare più raster contemporaneamente tenendo premuto il tasto `Ctrl` o `Shift` e cliccando su più elementi presenti nella finestra di dialogo *Apri una formato raster supportato da GDAL*.

Una volta caricato il raster puoi cliccare sul suo nome nella legenda con il tasto destro del mouse per selezionare ed attivare opzioni specifiche, o per aprire la finestra per l'impostazione delle proprietà.

Menu contestuale per layer raster

- *Zoom all'estensione del layer*
- *Zoom alla scala migliore (100%)*
- *Stira usando l'estensione attuale*
- *Aggiungi alla panoramica*
- *Rimuovi*
- *Duplica*
- *Set Layer CRS*
- *Imposta il SR del progetto dal layer*
- *Salva come ...*
- *Proprietà*
- *Rinomina*
- *Copia lo stile*
- *Add New Group*
- *Espandi tutto*
- *Comprimi tutto*
- *Aggiorna l'ordine del disegno*

13.2 Proprietà raster

Per visualizzare ed impostare le proprietà di un raster, fai doppio click sul nome del raster nella legenda o cliccaci sopra con il tasto destro e scegli *Proprietà* dal menu contestuale. Si aprirà così la finestra di dialogo *Proprietà del layer* (vedi [figure_raster_1](#)).

Ci sono diversi menu nella finestra di dialogo:

- *Generale*
- *Stile*
- *Trasparenza*
- *Piramidi*
- *Istogramma*
- *Metadati*

13.2.1 Menu Generale

Informazioni del layer

Il menu *Generale* contiene informazioni basilari del raster selezionato, inclusa la sorgente del file, il nome visualizzato nella legenda (che puoi modificare) e il numero di colonne, righe e valori nulli.

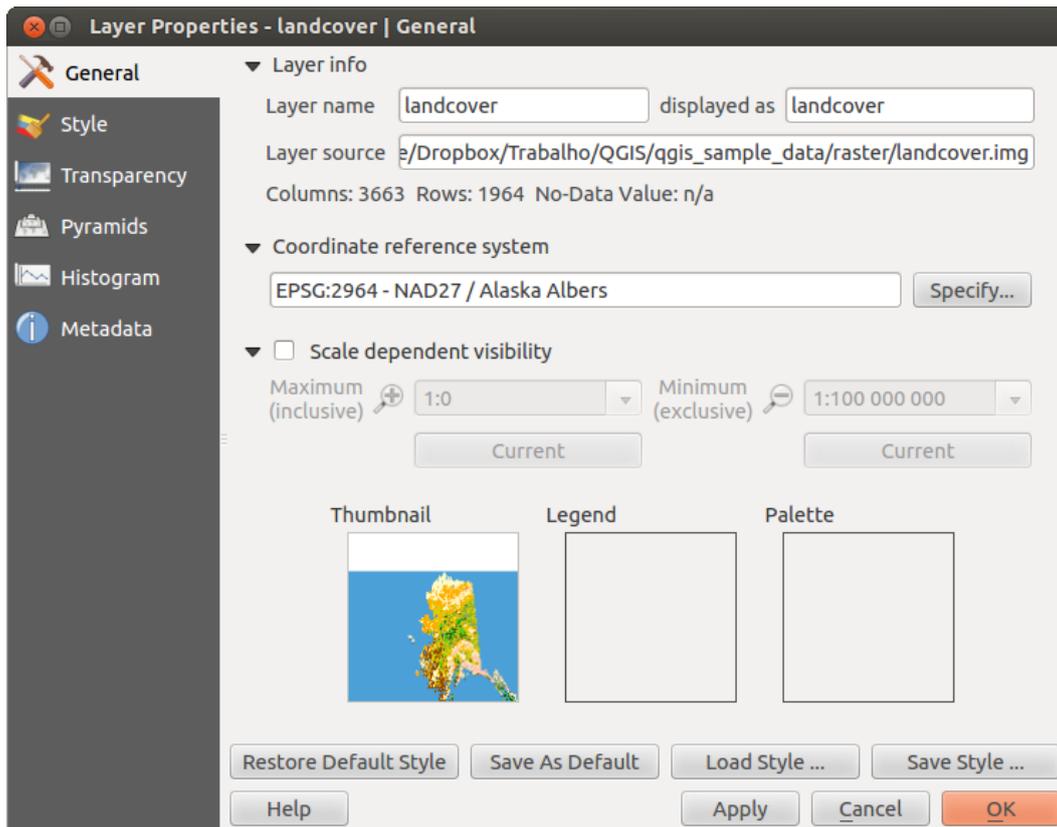


Figura 13.1: Finestra di dialogo proprietà del raster 🐧

Sistema Riferimento Coordinate

Qui puoi trovare il sistema di riferimento spaziale (SR) visualizzato in formato stringa PROJ.4. Se l'impostazione non è corretta la puoi modificare con il pulsante **[Specifica]**.

Visibilità dipendente dalla scala

In questo menu puoi attivare la funzione che imposta la visibilità del raster in funzione della scala. Spuntando la casella di controllo puoi impostare l'intervallo di scala in cui vuoi che il raster venga visualizzato nella mappa.

Nella parte inferiore puoi vedere un'anteprima del raster, la sua simbologia e la tavolozza.

13.2.2 Menu Stile

Visualizzazione banda

QGIS offre quattro tipologie di *Visualizzazione del layer*. La scelta dipende dal tipo di dato.

1. Colori banda multipla - se il file è caricato come multibanda e ha diverse bande di colori (per esempio un'immagine satellitare con molte bande diverse)
2. Tavolozza - se un file ha una tavolozza indicizzata (per esempio una mappa topografica digitale)
3. Banda singola grigia - (una banda) l'immagine verrà visualizzata in grigio: QGIS sceglierà questo tipo di visualizzazione se il file non ha né bande multiple né una tavolozza indicizzata né una tavolozza continua (comune per una mappa dei rilievi)
4. Banda singola falso colore - puoi usare questo visualizzatore per i file che hanno una tavolozza continua o una mappa di colore (per esempio una mappa delle altimetrie)

Colori banda multipla

Con il visualizzatore colore banda multipla verranno visualizzate le tre bande selezionate dell'immagine, ognuna delle quali corrisponde alle componenti rosso, verde e blu che verranno usate per creare i colori dell'immagine stessa. Puoi scegliere fra diversi metodi di *Miglioramento contrasto*: 'Nessun miglioramento', 'Stira a MinMax', 'Stira e taglia a MinMax' e 'Taglia a MinMax'.

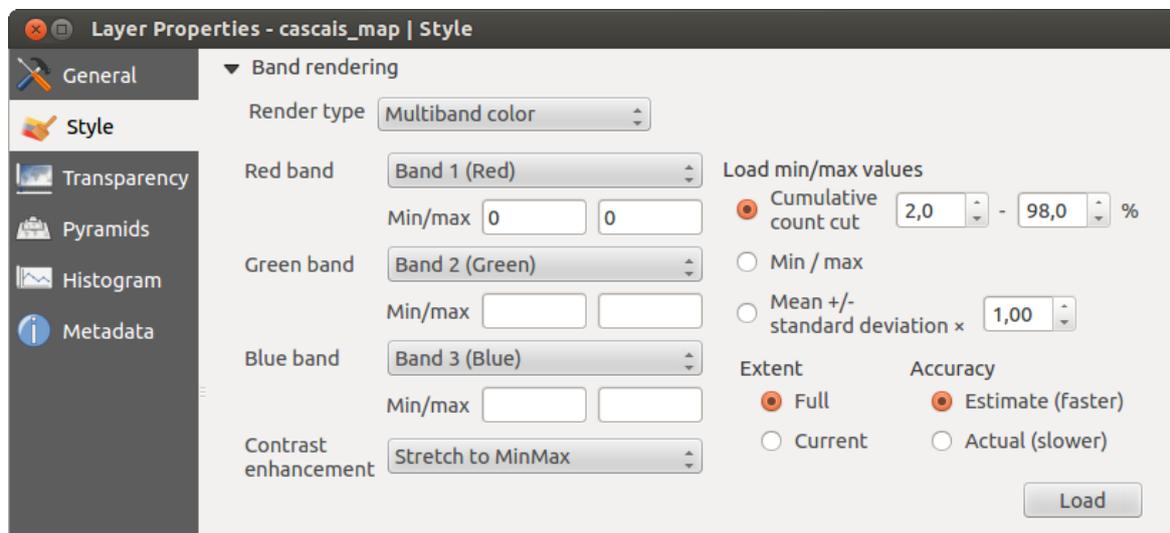


Figura 13.2: Visualizzatore raster - colori banda multipla 🐧

Questa sezione offre un'ampia gamma di opzioni per modificare l'aspetto del tuo raster. Prima di tutto scegli l'estensione dell'immagine da *Estensione* e poi premi il pulsante **[Carica]**. QGIS può scegliere l'*Accuratezza* stimando i valori *Min* e *Max* tramite i pulsanti *Stimato (veloce)* e *Attuale (lento)*.

Ora puoi impostare i colori con l'aiuto della sezione *Carica i valori min/max*. Molte immagini hanno pochi valori estremi. Puoi eliminare questi outlier con l'impostazione *Cumulative count cut*. L'intervallo standard è impostato dal 2% al 98% dei valori del file e può essere adattato manualmente. Con questa impostazione potrebbero sparire i caratteri grigi. Con l'opzione *Min/max*, QGIS crea una tabella di colori con tutti i valori dell'immagine originale (per esempio QGIS crea una tabella di colori con 256 valori, se la tua immagine ha bande a 8 bit). Puoi anche creare la tua tabella dei colori usando l'opzione *Media +/- deviazione standard x* . In questo modo solamente i valori inclusi nella deviazione standard o in multipli della deviazione standard verranno considerati nella tabella dei colori. Questo è molto utile quando hai una o due celle con valori molto grandi che avrebbero un impatto negativo nella visualizzazione del raster.

Le stesse impostazioni sono valide anche per l'estensione *Attuale*.

Suggerimento: Visualizzare una singola banda di un raster multibanda

Se vuoi vedere solamente una banda singola di un'immagine multibanda (per esempio, rossa) potresti pensare di impostare le bande verde e blu come "Non impostato". Ma questo non è il miglior modo di agire. Per visualizzare la banda rossa, seleziona il visualizzatore 'Banda grigia singola' e poi seleziona il rosso come colore da usare al posto del grigio.

Tavolozza

Questo è il visualizzatore standard per i file a banda singola che hanno già una tavola di colori, dove a ogni valore dei pixel è associato un determinato colore. In questo caso, la tavolozza viene visualizzata automaticamente. Se vuoi cambiare i colori assegnati a certi valori fai semplicemente doppio click sul colore e si aprirà così la finestra *Seleziona colore*. In QGIS 2.2 puoi anche assegnare un'etichetta ai valori dei colori. L'etichetta comparirà così nella legenda.

Miglioramento contrasto

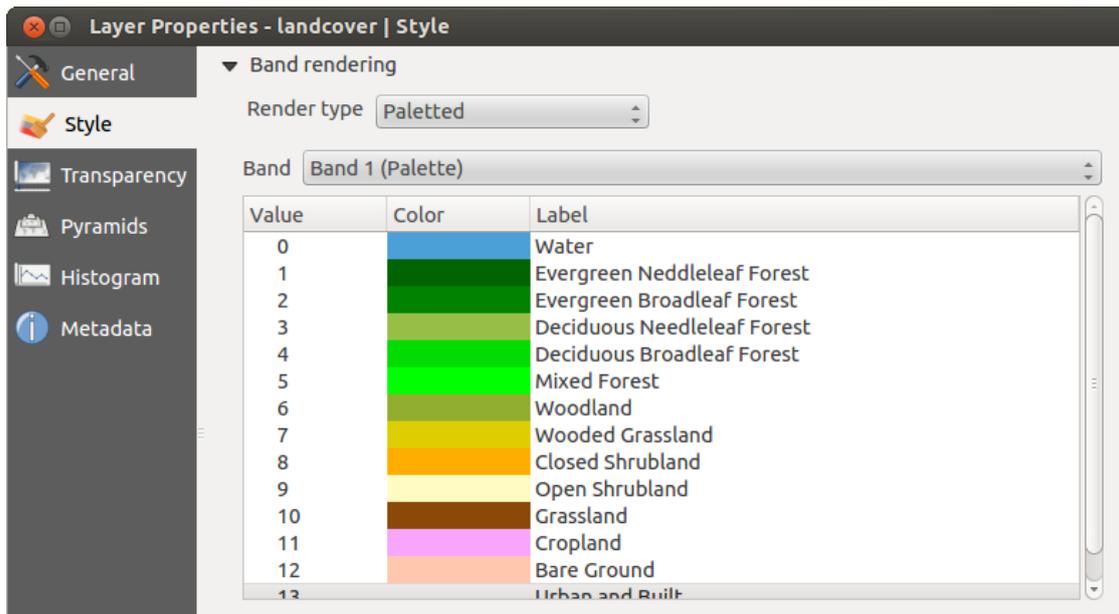


Figura 13.3: Visualizzatore raster - tavolozza 🐧

Nota: Quando aggiungi un raster di GRASS, l'opzione *Miglioramento contrasto* è sempre impostata su *Stira a MinMax* anche se hai impostato altri valori nelle opzioni di QGIS.

Banda singola grigia

Questo visualizzatore ti permette di visualizzare un raster a banda singola con un *Gradiente di colore*: 'Da nero a bianco' o 'Da bianco a nero'. Puoi selezionare il valore *Min* e quello *Max* scegliendo prima l'opzione *Estensione* e poi premendo **[Carica]**. QGIS può scegliere l'*Accuratezza* stimando i valori *Min* e *Max* tramite i pulsanti *Stimato (veloce)* e *Attuale (lento)*.

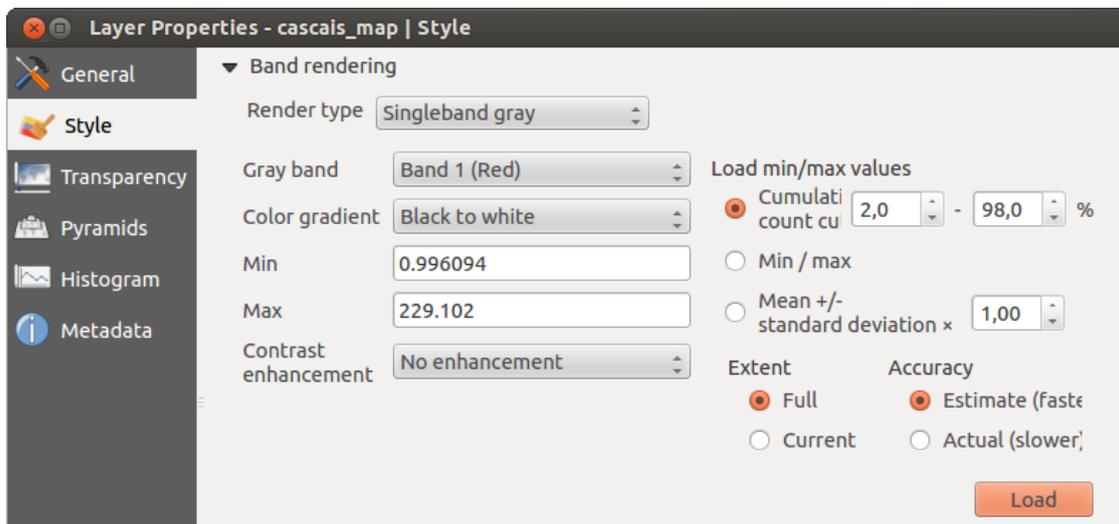


Figura 13.4: Visualizzatore raster - banda singola grigia 🐧

Nella sezione *Carica i valori min/max* puoi scegliere la tabella dei colori. Puoi eliminare questi outlier con l'impostazione *Cumulative count cut*. L'intervallo standard è impostato dal 2% al 98% dei valori del file e può essere adattato manualmente. Con questa impostazione potrebbero sparire i caratteri grigi. Puoi effettuare altri cambiamenti con le impostazioni *Min/max* e *Media +/- deviazione standard x* 1,00. Mentre la prima crea una tabella di colori con tutti i valori dell'immagine originale, la seconda opzione che una tabella di colori in

cui vengono considerati solamente i valori che ricadono all'interno della deviazione standard o a un multiplo di questa. Questo è molto utile quando hai una o due celle con valori molto grandi che avrebbero un impatto negativo nella visualizzazione del raster.

Banda singola falso colore

Questa è l'opzione di visualizzazione per i file a banda singola, inclusa una tavolozza continua. Puoi anche creare mappe di colori singoli per le bande singole. Sono disponibili tre tipologie di interpolazione di colore:

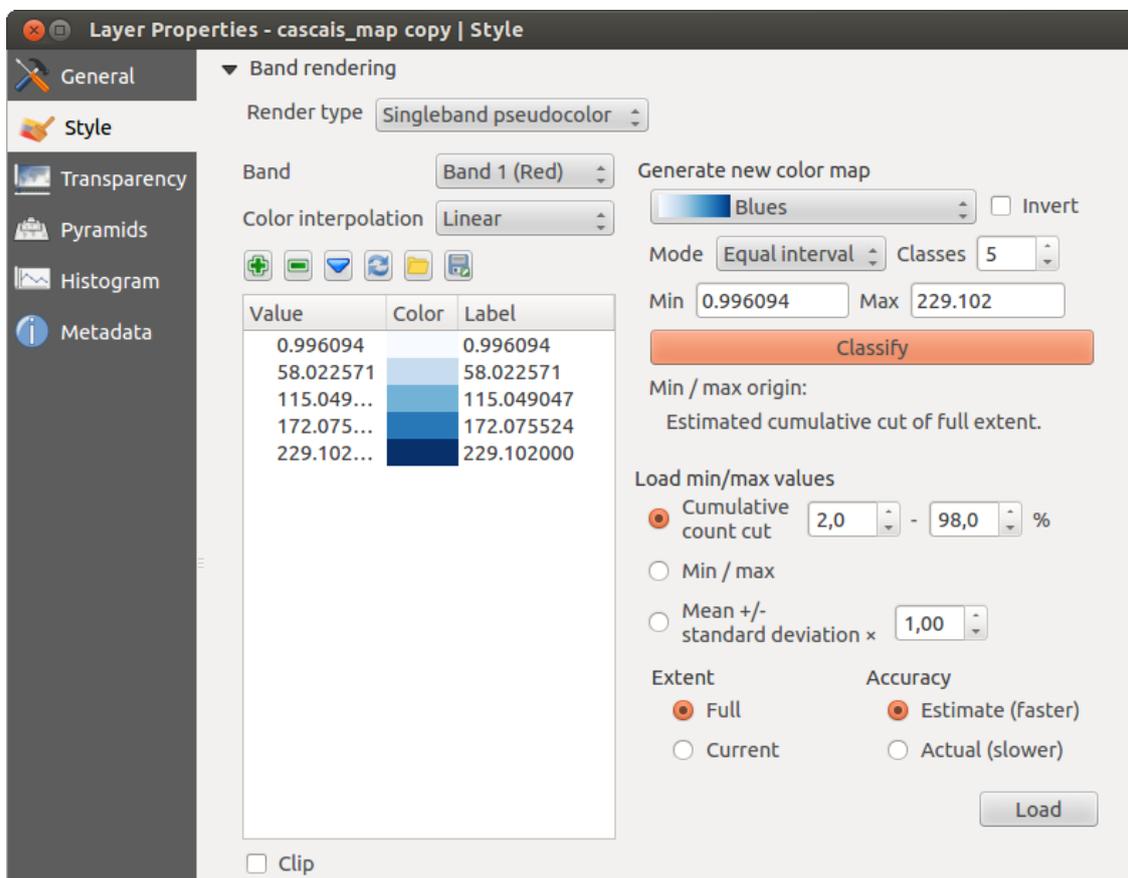


Figura 13.5: Visualizzatore raster - banda singola falso colore 🐧

1. Discreto
2. Lineare
3. Esatto

Nella parte sinistra, il pulsante  *Aggiungi un valore manualmente* aggiunge un valore alla tabella dei colori. Il pulsante  *Rimuovi la riga selezionata* cancella un valore dalla tabella dei colori e il pulsante  *Ordina gli elementi della mappa dei colori* ordina i colori della tabella in funzione dei valori dei pixel e dei valori della colonna. Facendo doppio click sul valore presente nella colonna potrai inserire un valore specifico. Facendo invece doppio click su un colore, potrai scegliere un colore specifico da assegnare a quel valore. Inoltre puoi anche aggiungere un'etichetta per ogni colore, ma questa etichetta non verrà visualizzata quando userai lo strumento Informazioni elementi. Puoi anche cliccare sul pulsante  *Carica mappa colore dalla banda*, che prova a caricare la tabella dalla banda (se questa esiste). Puoi usare i pulsanti  *Carica mappa colore da file* oppure  *Esporta mappa colore su file* per caricare una tabella di colori esistente o per salvarne una per le sessioni successive.

Nella parte destra, la sezione *Genera nuova mappa colore* ti permette di creare mappe di colore categorizzate. Per la *Modalità*  'Intervallo uguale' devi solamente scegliere il *numero di classi* e premere il pulsante *Classifica*. Puoi invertire i colori spuntando la casella di controllo *Inverti*. Se hai scelto la *Modalità* 

‘Continuo’, QGIS crea automaticamente le classi in funzione dei valori *Min* e *Max*. Puoi definire i valori *Min/Max* con l’aiuto della sezione *Carica i valori min/max*. Molte immagini hanno pochi valori estremi. Puoi eliminare questi outlier con l’impostazione *Cumulative count cut*. L’intervallo standard è impostato dal 2% al 98% dei valori del file e può essere adattato manualmente. Con questa impostazione potrebbero sparire i caratteri grigi. Con l’opzione *Min/max*, QGIS crea una tabella di colori con tutti i valori dell’immagine originale (per esempio QGIS crea una tabella di colori con 256 valori, se la tua immagine ha bande a 8 bit). Puoi anche creare la tua tabella dei colori usando l’opzione *Media +/- deviazione standard x* . In questo modo solamente i valori inclusi nella deviazione standard o in multipli della deviazione standard verranno considerati nella tabella dei colori.

Visualizzazione colore

Per ogni *Visualizzazione banda*, è disponibile una *Visualizzazione colore*.

Puoi anche ottenere effetti speciali per i tuoi raster usando una delle modalità fusione (vedi *Proprietà dei vettori*).

Ulteriori impostazioni possono essere fatte modificando la *Luminosità*, la *Saturazione* e il *Contrasto*. Puoi usare anche l’opzione *Scala di grigi* dove puoi scegliere fra ‘Per chiarezza’, ‘Per luminosità’ e ‘Per media’. Puoi modificare la ‘Forza’ per ogni tonalità della tabella dei colori.

Ricampionamento

La sezione *Ricampionamento* ha effetto quando ingrandisci o rimpicciolisci l’immagine. I metodi di ricampionamento ottimizzano l’aspetto della mappa perché calcolano una nuova matrice di grigi attraverso una trasformazione geometrica.

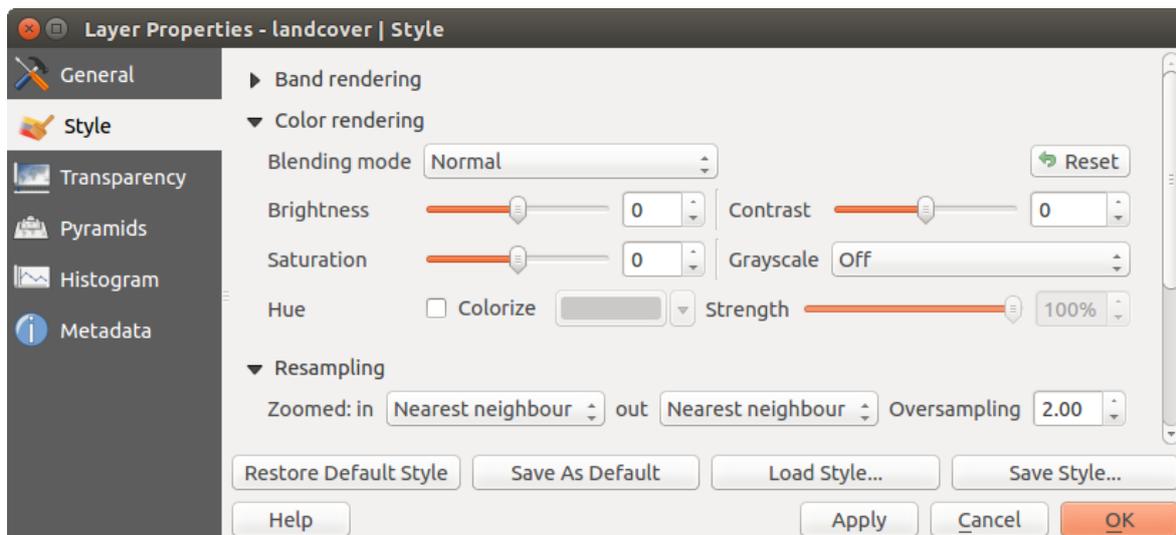


Figura 13.6: Visualizzatore raster - ricampionamento 

Applicando il metodo ‘vicino più prossimo’ la mappa potrebbe avere una struttura con molti pixel quando viene ingrandita. Questo aspetto può essere migliorato usando i metodi ‘Bilineare’ o ‘Cubico’ perché creano delle geometrie più appuntite e offuscate. Il risultato è un’immagine più morbida. Puoi applicare questo metodo, per esempio, a mappe raster topografiche.

13.2.3 Menu Trasparenza

QGIS riesce a visualizzare ogni raster con differenti livelli di trasparenza. Usa il cursore trasparenza  per impostare il livello di trasparenza che desideri. Questa opzione è molto utile se vuoi

sovrapporre diversi raster (per esempio una mappa dei rilievi sovrapposta a un raster classificato). In questo modo puoi simulare un effetto tridimensionale.

Inoltre puoi inserire nel menu *Valori nulli aggiuntivi* un valore che deve essere trattato come *Valore nullo*.

Puoi definire la trasparenza in maniera ancora più dettagliata e personalizzata nella sezione *Opzioni di trasparenza personalizzate*, nella quale puoi impostare il grado di trasparenza di ogni singola cella (o pixel).

Per esempio, vogliamo impostare l'acqua del file `landcover.tif` con una trasparenza del 20%. Questi sono i passi necessari:

1. Carica il file
2. Apri la finestra di dialogo *Proprietà* facendo doppio click sul nome del raster nella legenda o cliccando su di esso con il tasto destro del mouse e scegliendo *Proprietà* dal menu contestuale.
3. Seleziona il menu *Trasparenza*.
4. Scegli 'Nessuno' dal menu *Banda trasparenza*.
5. Clicca sul pulsante  *Aggiungi valori manualmente*. Apparirà così una nuova riga nell'elenco.
6. Inserisci il valore nelle colonne 'Da' e 'A' (nell'esempio viene usato 0) e aggiusta la trasparenza al 20%.
7. Clicca sul pulsante [**Applica**] per visualizzare il risultato.

Ripeti i passaggi 5 e 6 per aggiustare più valori con trasparenze personalizzate.

Come puoi vedere è molto semplice impostare una trasparenza personalizzata, però richiede comunque un po' di lavoro. Proprio per questo puoi usare il pulsante  *Esporta su file* per salvare la lista dei valori su un file esterno. Il pulsante  *Importa da file* ti permette di caricare le impostazioni di trasparenza e applicarle al raster selezionato.

13.2.4 Menu Piramidi

I raster ad alta risoluzione possono rallentare notevolmente il lavoro in QGIS. Creando copie a bassa risoluzione dei dati (piramidi) puoi incrementare notevolmente le prestazioni in quanto QGIS sceglierà la risoluzione migliore in funzione del fattore di zoom.

Per creare piramidi devi avere i permessi di scrittura nella cartella contenente il dato originale: in questa cartella verranno salvate le copie a bassa risoluzione.

Sono disponibili i seguenti metodi di ricampionamento:

- Vicino più prossimo (metodo Nearest Neighbour)
- Media
- Gauss
- Cubico
- Modo
- Nessuno

Se scegli 'Interno (se possibile)' da menu a tendina *Formato panoramica*, QGIS proverà a costruire le piramidi internamente. Puoi anche scegliere 'Esterno' e 'Esterno (immagine Erdas)'.

La costruzione delle piramidi può alterare il dato originale in maniera irreversibile, quindi ti raccomandiamo di fare una copia del raster originale prima di eseguire l'operazione.

13.2.5 Menu Istogramma

Il menu *Istogramma* mostra la distribuzione delle bande di colore del raster. L'istogramma viene automaticamente generato quando apri il menu *Istogramma* e tutte le bande vengono mostrate insieme. Puoi salvare l'istogramma

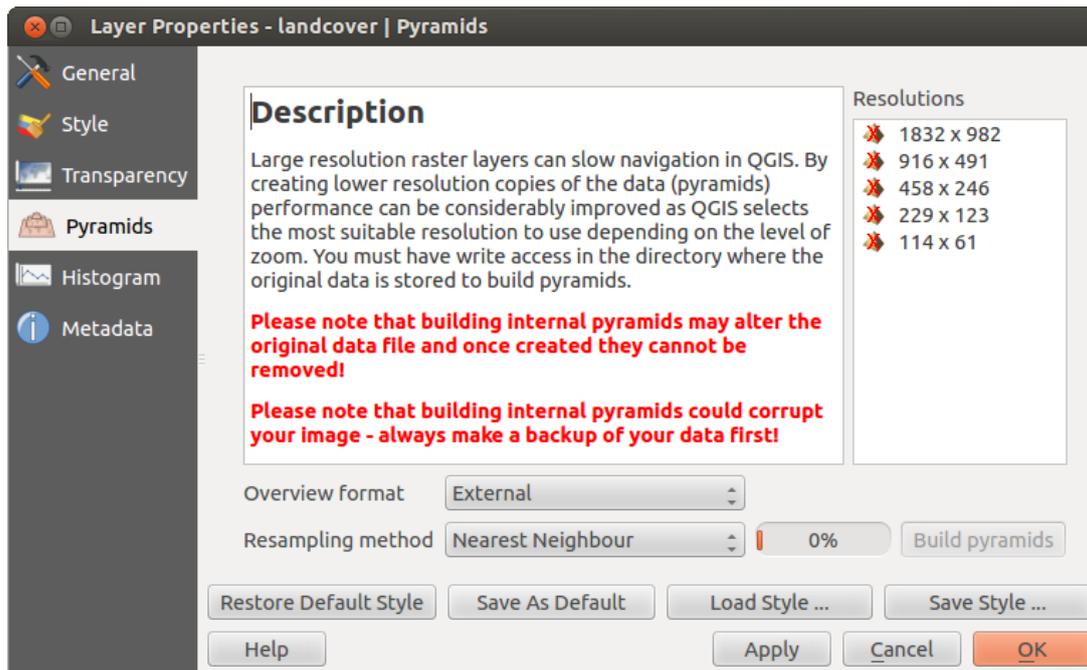


Figura 13.7: Menu Piramidi 🐧

come immagine con il pulsante . L'opzione *Visibilità* del menu  *Preferiti/Azioni* ti permette di visualizzare gli istogrammi delle singole bande. Per farlo, seleziona l'opzione *Mostra la banda selezionata*. L'opzione *Opzioni Min/Max* ti permette di scegliere fra: 'Mostra sempre i marcatori min/max', 'Zoom a min/max' e 'Aggiorna stile a min/max'. Con l'opzione *Azioni* puoi scegliere anche fra 'Ripristina' e 'Ricalcola l'istogramma' dopo aver cambiato le impostazioni *Opzioni Min/Max*.

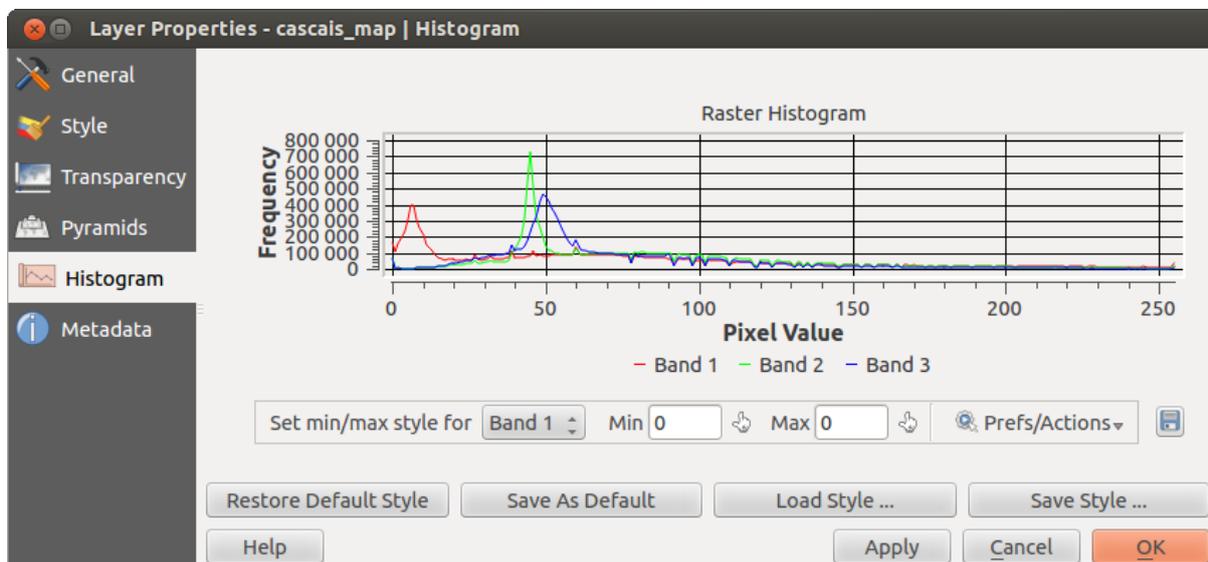


Figura 13.8: Istogramma del raster 🐧

13.2.6 Menu Metadati

La scheda *Metadati* mostra una serie di informazioni sul raster, incluse le statistiche di ogni banda. Da questo menu hai accesso a diverse sezioni: *Descrizione*, *Assegnazione*, *URL Metadati* e *Proprietà*. Nella sezione *Proprietà* le

statistiche sono ottenute da una base ‘che si deve ancora conoscere’, quindi è meglio che le statistiche di questo raster non siano ancora state calcolate.

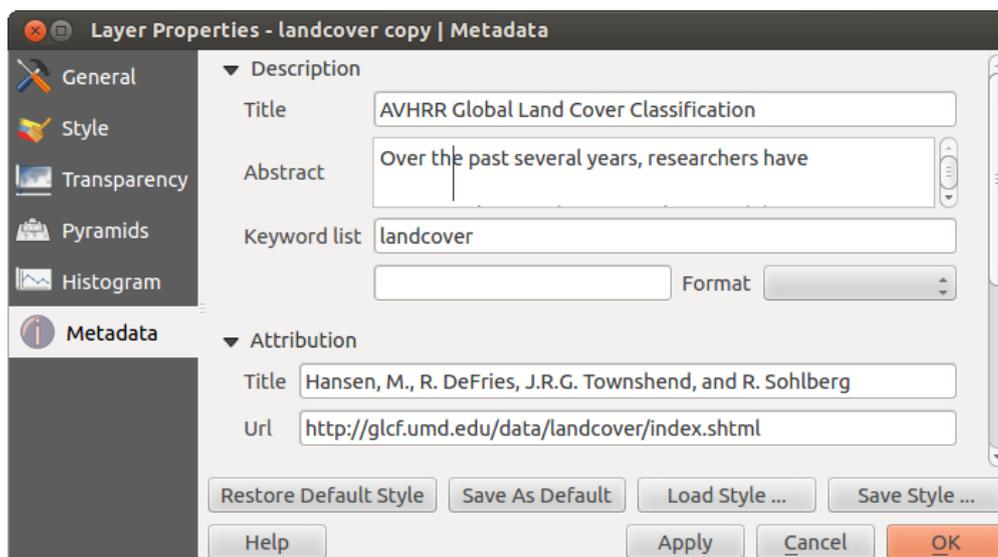


Figura 13.9: Metadati raster 🐧

13.3 Calcolatore raster

Il *Calcolatore Raster* nel menu *Raster* consente di eseguire calcoli sulla base dei valori dei pixel raster esistenti (vedi [figure_raster_10](#)). I risultati vengono scritti in un nuovo raster con un formato supportato GDAL.

La sezione **Bande raster** elenca i raster caricati che possono essere utilizzati. Per aggiungere un raster nella finestra Espressione del calcolatore raster, fai doppio click sul suo nome nella finestra Bande raster. Puoi usare gli operatori per costruire un'espressione oppure puoi scriverla direttamente nella finestra delle Espressione del calcolatore raster.

Nella sezione **Risultato del layer** devi definire il nome del raster in output. Hai diverse opzioni per scegliere l'estensione dell'area di calcolo: sulla base del raster in input, su coordinate X e Y oppure sulla base del numero di righe e colonne. L'estensione che scegli determinerà la risoluzione finale del raster in output. Se il raster in input ha una risoluzione diversa, i valori verranno ricampionati con l'algoritmo del vicino più prossimo.

La finestra **Operatori** elenca tutti gli operatori disponibili. Per aggiungere un operatore alla finestra Espressione del calcolatore raster, clicca sull'icona dell'operatore. Sono disponibili operatori matematici (+, -, * . . .) e funzioni trigonometriche (sin, cos, tan, . . .). Aggiungeremo presto tante altre funzioni!

Selezionando la casella di controllo *Aggiungi al progetto* il raster finale verrà aggiunto alla legenda e lo potrai visualizzare sulla mappa.

13.3.1 Esempi

Convertire unità di misura dell'elevazione da metri a piedi

Per creare un raster con altimetria in piedi a partire da un raster con altimetria in metri devi utilizzare il fattore di conversione di 3,28 piedi per metro. L'espressione è:

```
"elevation@1" * 3.28
```

Utilizzare una maschera

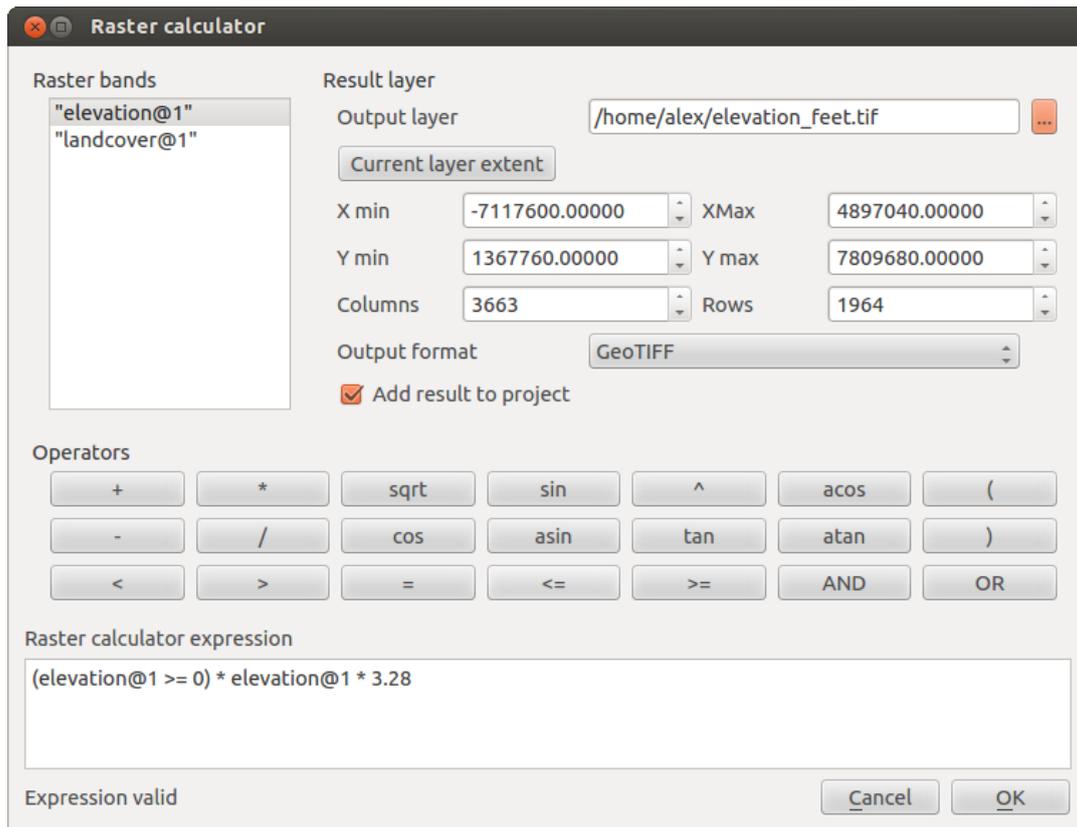


Figura 13.10: Calcolatore raster 🐧

Se vuoi usare una maschera sul raster, perché, per esempio, sei interessato solo ai valori di altezza superiori allo 0, puoi usare l'espressione seguente. Questa creerà una maschera e applicherà i risultati in un solo passaggio.

```
("elevation@1" >= 0) * "elevation@1"
```

In altre parole, per ogni cella maggiore o uguale a 0 imposta il valore uguale a 1, altrimenti imposta 0. In questo modo creerai la maschera al volo.

Se vuoi classificare un raster, ad esempio, in due classi di elevazione, è possibile utilizzare la seguente espressione per creare un raster con due valori 1 e 2 in un solo passo.

```
("elevation@1" < 50) * 1 + ("elevation@1" >= 50) * 2
```

In altre parole, per ogni cella meno di 50 imposta il valore su 1. Per ogni cella maggiore o uguale a 50 imposta il valore su 2.

Lavorare con i dati OGC

14.1 QGIS come client dati OGC

L'Open Geospatial Consortium (OGC), è un'organizzazione internazionale che raggruppa più di 300 organizzazioni commerciali, governative, no-profit ed enti di ricerca. I suoi membri sviluppano e implementano standard per contenuti e servizi geospaziali, analisi GIS e scambio dati.

OGC ha elaborato un numero crescente di specifiche per la descrizione di un modello dati di base per elementi geografici: le specifiche sono orientate a garantire l'interoperabilità nell'ambito della tecnologia geospaziale. Ulteriori informazioni all'indirizzo <http://www.opengeospatial.org/>.

Importanti specifiche OGC supportate da QGIS sono:

- **WMS** — Web Map Service (*Client WMS/WMTS*)
- **WMTS** — Web Map Tile Service (*Client WMS/WMTS*)
- **WFS** — Web Feature Service (*Client WFS e WFS-T*)
- **WFS-T** — Web Feature Service - Transactional (*Client WFS e WFS-T*)
- **WCS** — Web Coverage Service (*Client WCS*)
- **SFS** — Simple Features for SQL (*Vettori PostGIS*)
- **GML** — Geography Markup Language

Ad oggi i servizi OGC sono sempre più di uso comune per scambiare dati geografici fra differenti implementazioni GIS. QGIS ora può gestire tre delle specifiche esposte sopra, **SFS** (tramite il supporto a PostgreSQL/PostGIS, vedi sezione *Vettori PostGIS*), **WFS** e **WMS** come client.

14.1.1 Client WMS/WMTS

Panoramica sul servizio WMS

QGIS può agire come client WMS, nel rispetto delle specifiche 1.1, 1.1.1 e 1.3. È stato testato nei confronti di server accessibili pubblicamente quali DEMIS e JPL OnEarth.

I server WMS rispondono alle richieste da parte dei client (ad es. QGIS) di una mappa raster di una determinata estensione, con un determinato insieme di layer, simboli e trasparenze. Il server WMS quindi consulta le sue risorse (locali o remote), genera il raster e lo invia al client in formato raster, per QGIS tipicamente come immagini JPEG o PNG.

WMS è un servizio REST (Representational State Transfer) piuttosto che un servizio web completo. Quindi puoi prendere l'URL (indirizzo del server con specifiche) generato da QGIS e usarlo in un browser web per ottenere la

stessa immagine che QGIS usa internamente. Questo è utile per identificare le cause di eventuali problemi, dato che esistono vari tipi di server WMS e ciascuno ha la sua propria interpretazione degli standard WMS.

I layer WMS possono essere aggiunti molto semplicemente, una volta disponibile l'indirizzo (URL) per accedere al server WMS, una connessione adatta e posto che il server usi HTTP come meccanismo di trasferimento dati.

Panoramica sul servizio WMTS

QGIS può agire anche come client WMTS. WMTS è uno standard OGC che distribuisce insiemi di mattonelle di dati geospaziali. È un modo più efficace e veloce rispetto a WMS perché gli insiemi di mattonelle vengono già generati e il client deve solamente richiedere la trasmissione di queste mattonelle e non la loro produzione. Una richiesta WMS tipicamente richiede sia la generazione che la trasmissione dei dati. Un esempio molto conosciuto di standard non-OGC è Google Maps.

Per visualizzare i dati a diverse scale, l'insieme delle mattonelle WMTS vengono prodotte con scale molto differenti fra loro in modo che per il client GIS sia più facile effettuare la richiesta.

Questo diagramma mostra il concetto delle mattonelle:

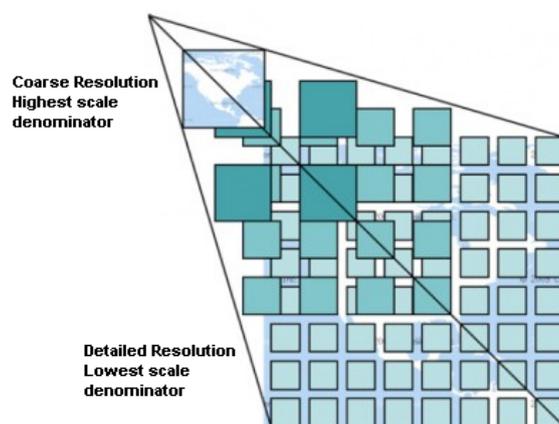


Figura 14.1: Concetto dell'insieme delle mattonelle WMTS

Le due tipologie di interfacce WMTS che supporta QGIS sono via Key-Value-Pairs (KVP) e RESTful. Queste due interfacce sono differenti e devi specificarle in QGIS.

1) In order to access a **WMTS KVP** service, a QGIS user must open the WMS/WMTS interface and add the following string to the URL of the WMTS tile service:

```
"?SERVICE=WMTS&REQUEST=GetCapabilities"
```

Un esempio di questo tipo di indirizzo è:

```
http://opencache.statkart.no/gatekeeper/gk/gk.open_wmts?\  
service=WMTS&request=GetCapabilities
```

Per vedere se il layer topo2 funziona correttamente in questo WMTS, aggiungi la stringa indicata che il servizio WMTS deve usare al posto del servizio WMS.

2. Il servizio **RESTful WMTS** segue un modulo diverso, ovvero un URL diretto. Il formato raccomandato da OGC è:

```
{WMTSBaseURL}/1.0.0/WMTSCapabilities.xml
```

Questo formato ti aiuta a riconoscere che questo è un indirizzo RESTful. Puoi accedere a un RESTful WMTS in QGIS semplicemente aggiungendo il suo indirizzo nel campo dedicato del modulo. Un esempio di questo tipo di indirizzo, per una base austriaca, è <http://maps.wien.gv.at/basemap/1.0.0/WMTSCapabilities.xml>.

Nota: Puoi ancora trovare qualche vecchio servizio chiamato WMS-C. Questi servizi sono

abbastanza simili a WMTS (cioè hanno lo stesso scopo ma lavorano in modo diverso). Li puoi gestire esattamente come i servizi WMTS. Aggiungi `?tiled=true` alla fine dell'URL. Vedi http://wiki.osgeo.org/wiki/Tile_Map_Service_Specification per ulteriori informazioni su questa specifica.

Quando leggi WMTS, puoi anche pensare a WMS-C.

Selezionare server WMS/WMTS

Al primo utilizzo di un servizio WMS in QGIS non sono presenti server predefiniti.

Puoi avviare lo strumento cliccando sul pulsante  **Aggiungi layer WMS** nella barra strumenti, oppure scegliendo *Layer* → *Aggiungi layer WMS...*

Si aprirà la finestra di dialogo: *Aggiungi Layer da server*. Puoi aggiungere alcuni server cliccando sul pulsante **[Aggiungi server predefiniti]**. Verranno quindi aggiunti due server WMS, il server DM Solutions Group ed il server Lizardtech. Per definire un nuovo server WMS nella sezione *Layer*, clicca sul pulsante **[Nuovo]** ed inserisci i parametri di connessione del server WMS desiderato, seguendo le indicazioni della tabella `_OGC_1_`:

Nome	Un nome per la connessione. Questo nome verrà utilizzato nel menù a tendina dei server in modo da distinguere i vari server WMS.
URL	URL del server che fornisce i dati. Deve essere un indirizzo raggiungibile nello stesso formato che verrebbe usato per aprire una connessione telnet o pingare un host.
Username	Nome utente per accedere un WMS protetto. Questo parametro è opzionale.
Password	Password per accedere ad un WMS protetto. Questo parametro è opzionale.
Ignora URI GetMap	<input checked="" type="checkbox"/> <i>Ignora la URI GetMap riportata nelle capabilities.</i> Viene utilizzato l'URI del campo URL precedente.
Ignora URI GetFeatureInfo	<input checked="" type="checkbox"/> <i>Ignora la URI GetFeatureInfo riportata nelle capabilities.</i> Viene utilizzato l'URI del campo URL precedente

Table OGC 1: Parametri di connessione WMS

Se necessario puoi impostare i parametri di un proxy per ricevere i servizi WMS da internet. Seleziona la voce *Impostazioni* → *Opzioni* e clicca sulla scheda *Rete*, nella quale puoi inserire le impostazioni spuntando la casella di controllo *Utilizza un proxy per l'accesso web*. Assicurati di scegliere il tipo di proxy corretto dal menù a tendina *Tipo proxy* .

Una volta creata la connessione al server WMS, verrà memorizzata e sarà disponibile per le successive sessioni di QGIS.

Suggerimento: A PROPOSITO DI INDIRIZZI DEI SERVER WMS

Quando inserisci l'indirizzo URL del server assicurati di usare l'indirizzo di base. Ad esempio non devi inserire frammenti tipo `request=GetCapabilities` o `version=1.0.0` nell'indirizzo.

Caricare layer WMS/WMTS

Una volta riempiti tutti i campi dei parametri richiesti, usa il pulsante **[Connetti]** per caricare le capabilities dei server selezionati. Queste includono: le codifiche delle immagini, i layer, gli stili e le proiezioni. Dal momento che è un'operazione eseguita in rete, la velocità di esecuzione dipende dalla velocità della tua connessione. Mentre vengono scaricati i dati dal server WMS, puoi vedere l'avanzamento nella parte inferiore della finestra di dialogo.

Il vostro schermo adesso deve essere simile a quello rappresentato alla [figura_OGR_1](#), che mostra la risposta fornita dal server WMS di DM Solutions Group.

Codifica immagine

La sezione *Codifica immagine* elenca i formati supportati sia dal client che dal server. La scelta è in funzione dei requisiti di accuratezza.

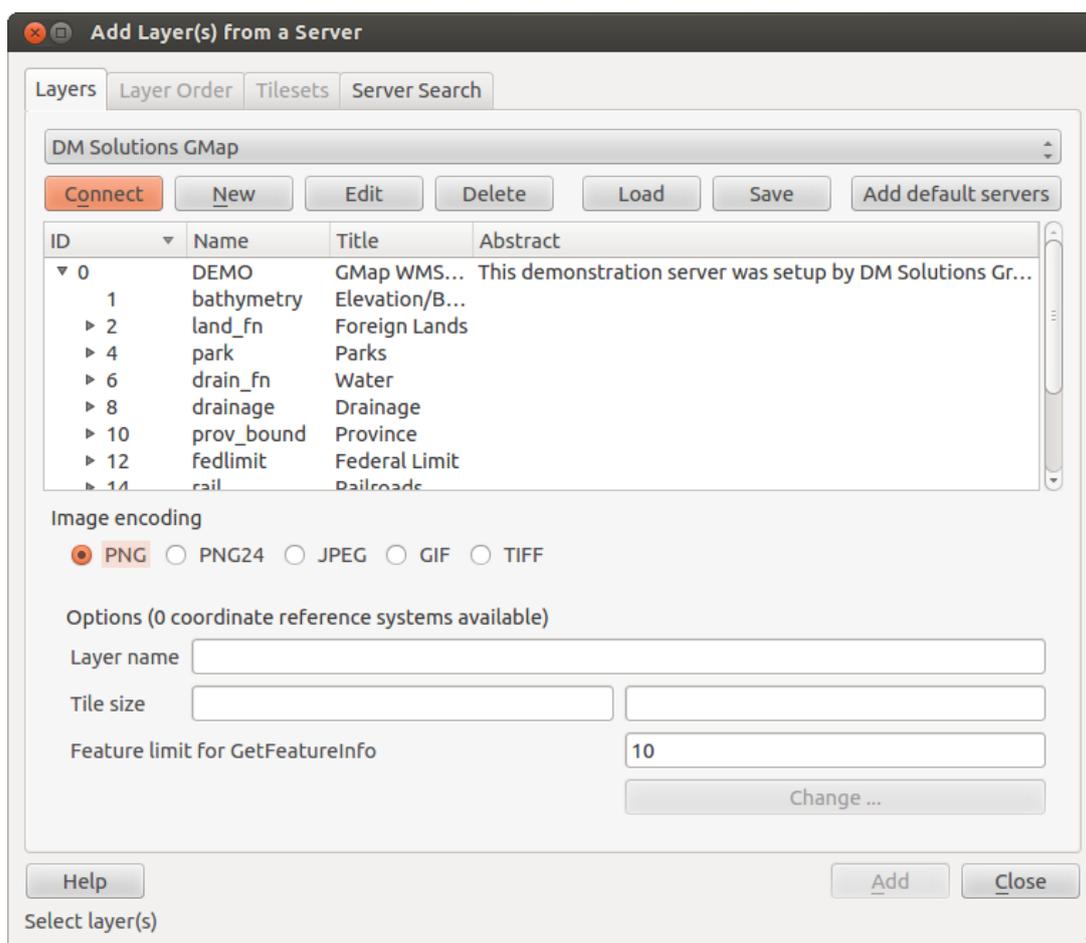


Figura 14.2: Finestra di dialogo per aggiungere un server WMS, mostrando tutti i layer disponibili 🐧

Suggerimento: Codifica immagine

Un server WMS offre normalmente la scelta fra immagini JPEG o PNG. Il formato JPEG è un formato di compressione lossy, mentre il formato PNG riproduce fedelmente i dati raster.

È meglio usare il formato JPEG per dati di natura fotografica e/o se la perdita parziale di qualità dell'immagine non causa problemi. Questa scelta riduce normalmente di cinque volte il volume di dati trasferiti rispetto al formato PNG.

L'uso del formato PNG permette una visualizzazione più precisa ed è da usare se non ci sono problemi per l'incremento dei dati trasferiti.

Opzioni

La sezione Opzioni mette a disposizione il campo testo *Nome layer* per dare un nome al layer WMS. Questo nome apparirà nella legenda a caricamento avvenuto.

Sotto il nome del layer puoi scegliere la *Dimensione delle tile*, (per esempio 256x256) in modo da dividere la richiesta WMS in richieste multiple.

Il campo *Limite di elementi per GetFeatureInfo* definisce quali elementi del server interrogare.

Selezionando un layer WMS dalla lista apparirà un campo con il sistema di proiezione predefinito dal server. Se il pulsante [**Cambia...**] è attivo, allora puoi scegliere un altro SR fornito dal server.

Ordine dei Layer

La scheda *Ordine layer* elenca i diversi layer disponibili sul server WMS a cui sei connesso. Puoi notare che alcuni layer sono espandibili; questo significa che puoi visualizzare quei layer con diversi stili di immagine.

Puoi selezionare più layer in una volta, ma solo uno stile di visualizzazione per layer. Quando più layer sono selezionati questi vengono combinati dal server ed inviati a QGIS in un'unica volta.

Suggerimento: Ordine dei layer WMS

I layer WMS caricati sono sovrapposti in base all'ordine in cui sono elencati nella sezione Layer, ovvero dall'alto verso il basso. Se vuoi cambiare l'ordine di visualizzazione, usa la scheda *Ordine layer*.

Trasparenza

In questa versione di QGIS l'impostazione della *Trasparenza globale* che si imposta da *Proprietà layer* è impostata per essere sempre attiva, se disponibile.

Suggerimento: Trasparenza dei layer WMS

La possibilità di rendere trasparenti i layer WMS dipende dalla codifica tramite la quale sono stati caricati: PNG e GIF gestiscono la trasparenza mentre il JPEG no.

Sistema di Riferimento

Sistema di riferimento delle coordinate (Coordinate Reference System) è il termine OGC per una proiezione in QGIS.

Ogni layer WMS può avere diversi SR, in funzione delle capacità del server.

Per scegliere uno dei SR disponibili, clicca su [**Cambia...**] per fare apparire una finestra simile a quella della figura 3 in *Lavorare con le proiezioni*. La differenza principale è che saranno mostrati solo i SR supportati dal server al quale sei connesso.

Ricerca Server

Con QGIS puoi anche cercare server WMS. La [Figura_OGC_2](#) mostra la nuova scheda *Cerca server* della finestra di dialogo *Aggiungi layer dal server*.

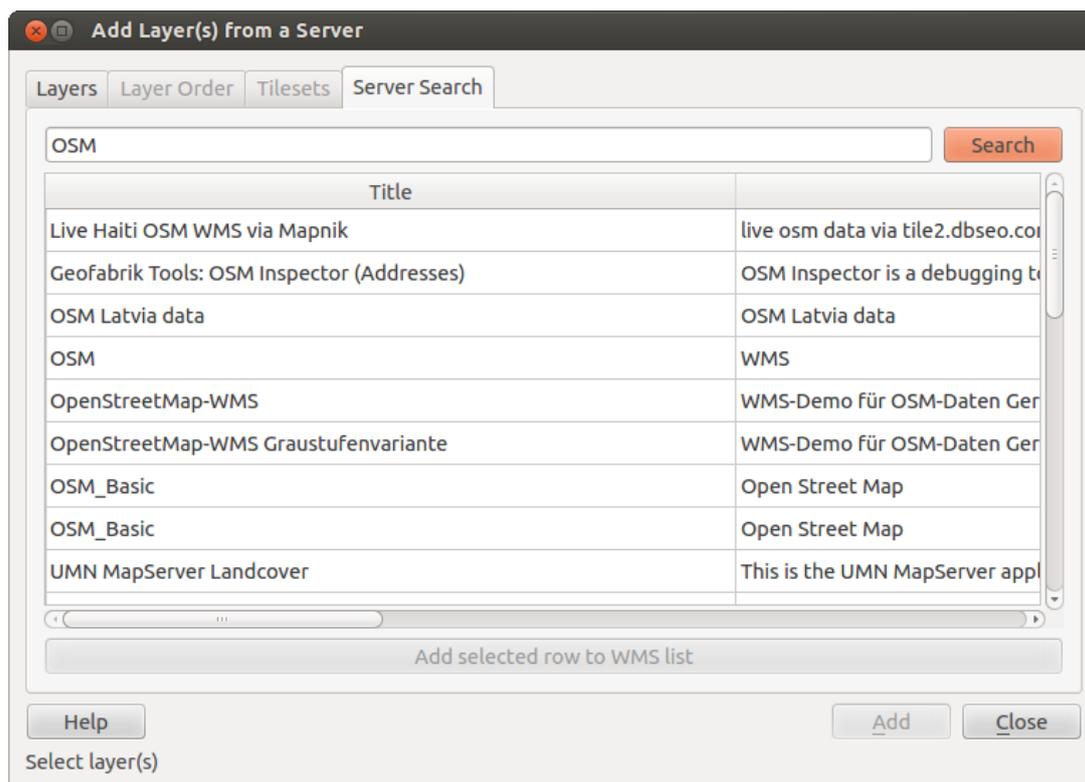


Figura 14.3: Finestra di dialogo della ricerca di server WMS con alcune parole chiave 🐧

Inserire una stringa di ricerca e cliccare sul pulsante [**Cerca**]: i risultati saranno elencati nella sottostante tabella. Per usare uno dei risultati, selezionalo dalla tabella e clicca su [**Aggiungi riga alla lista WMS**]. Il server verrà automaticamente aggiunto alla lista dei server nella scheda *Layer*. Clicca su [**Connetti**] per ottenere la lista di layer forniti dal server. Questa opzione è molto utile quando vuoi cercare delle mappe usando parole chiave.

Si tratta fondamentalmente di un front end alle API di <http://geopole.org>.

Set di tile

Quando usi servizi WMTS (Cached WMS) come

```
http://opencache.statkart.no/gatekeeper/gk/gk.open_wmts?
service=WMTS&request=GetCapabilities
```

puoi cercare grazie alle scheda *Set di tile* dati dai server. Informazioni aggiuntive come la dimensione delle mattonelle, i formati e i SR supportati sono elencati in questa tabella. In combinazione con questa funzionalità, puoi usare il cursore della scala delle mattonelle selezionando *Impostazioni* → *Panelli* (KDE e Windows) o *Visualizza* → *Panelli* (Gnome e MacOSX) e scegliendo 'Scala delle mattonelle'. Potrai usare un comodo cursore per la dimensione delle mattonelle.

Uso dello strumento di identificazione

Una volta aggiunto un server WMS, e se uno dei layer disponibili è interrogabile, puoi usare lo strumento  per selezionare un pixel sulla mappa, interrogando così il server WMS. La formattazione del risultato dell'interrogazione fatta dipende dal server WMS utilizzato. **Selezione formato**

Se il server supporta diversi formati in output, un menu a tendina verrà automaticamente aggiunto alla finestra delle informazioni risultati in modo che i diversi formati possano essere memorizzati nel progetto. **Supporto formato GML**

Lo strumento  **Informazione elementi** supporta anche le risposte WMS (GetFeatureInfo) in formato GML (è chiamato Geometria nell'interfaccia di QGIS in questo contesto). Se il formato "Geometria" è supportato dal server, i risultati dello strumento 'Informazione elementi' sono geometrie proprio come un vettore. Quando selezioni una singola geometria nell'albero, questa viene evidenziata sulla mappa e la puoi copiare e incollare su un altro vettore. Vedi l'esempio UMN Mapserver per il supporto GetFeatureInfo in formato GML.

```
# in layer METADATA add which fields should be included and define geometry (example):

"gml_include_items"    "all"
"ows_geometries"       "mygeom"
"ows_mygeom_type"      "polygon"

# Then there are two possibilities/formats available, see a) and b):

# a) basic (output is generated by Mapserver and does not contain XSD)
# in WEB METADATA define formats (example):
"wms_getfeatureinfo_formatlist" "application/vnd.ogc.gml,text/html"

# b) using OGR (output is generated by OGR, it is send as multipart and contains XSD)
# in MAP define OUTPUTFORMAT (example):
OUTPUTFORMAT
  NAME "OGRGML"
  MIMETYPE "ogr/gml"
  DRIVER "OGR/GML"
  FORMATOPTION "FORM=multipart"
END

# in WEB METADATA define formats (example):
"wms_getfeatureinfo_formatlist" "OGRGML,text/html"
```

Proprietà del server

Una volta aggiunto un server WMS, puoi visualizzarne le proprietà cliccando con il tasto destro sul suo nome nella legenda e selezionando *Proprietà*. **Scheda Metadati**

La scheda *Metadati* mostra molte informazioni sul server WMS: queste informazioni sono fornite dal server stesso in risposta alla richiesta di GetCapabilities fatta da QGIS. Puoi ricavare molte informazioni leggendo gli standard WMS (vedi OPOPEN-GEOSPATIAL-CONSORTIUM *Letteratura e riferimenti web*). Di seguito alcune definizioni utili:

- **Proprietà del server**

- **Versione WMS** — La versione WMS supportata dal server.
- **Formati immagine** — l'elenco dei tipi MIME disponibili sul server durante la visualizzazione della mappa. QGIS permette l'uso di qualunque formato supportato dalle librerie Qt, solitamente sono image/png ed image/jpeg.
- **Interroga formati** — L'elenco dei tipi MIME con i quali il server può fornire risposta quando si usa lo strumento Informazioni elementi. Attualmente QGIS supporta il tipo text-plain.

- **Proprietà layer**

- **Selezionato** — Indica se il layer era selezionato quando il server è stato aggiunto al progetto.
- **Visibilità** — Indica se il layer è stato impostato come visibile in legenda. (funzione non ancora utilizzata in questa versione di QGIS).
- **Può interrogare** — Indica se il layer fornisce o meno informazioni se si usa lo strumento Informazioni elementi.
- **Può essere trasparente** — Indica se il layer può essere o meno reso trasparente. Questa versione di QGIS userà sempre la trasparenza se questa è Sì e se il formato immagine la supporta.

- **Può ingrandire** — Indica se il layer può essere ingrandito dal server. In questa versione di QGIS questa opzione è impostata in modo predefinito su Sì. I layer senza questa impostazione potrebbero essere visualizzati in modo anomalo.
- **Conteggio a cascata** — I server WMS possono fungere da proxy per altri server WMS dai quali ottengono i dati raster per un certo layer. La voce mostra quindi quante richieste per questo layer vengono inoltrate ai nodi per ottenere un risultato.
- **Larghezza fissa, Altezza fissa** — Indica se il layer ha o meno una dimensione fissa dei pixel. In questa versione di QGIS tutti i layer WMS hanno questa impostazione impostata su 'vuoto'. I layer con impostazioni diverse potrebbero essere visualizzati in modo anomalo.
- **Perimetro WGS 84** — Estensione del layer in coordinate WGS84. Alcuni server WMS non impostano questo parametro correttamente (ad es. usano coordinate UTM invece di WGS84). In questo caso sembrerà che la vista iniziale del layer sia ad uno zoom molto ridotto. Bisognerebbe informare di questi errori il webmaster del server WMS, il quale li dovrebbe identificare come elementi WMS XML `LatLonBoundingBox`, `EX_GeographicBoundingBox` o `SR:84 BoundingBox`.
- **Disponibilità in CRS** — Sistemi di riferimento nel quale il layer può essere rappresentato dal server WMS, elencati nel formato nativo WMS.
- **Disponibile in stile** — Stili visuali applicabili al layer dal server WMS.

Mostra la legenda WMS nella legenda e nel compositore di stampe

La sorgente dati WMS di QGIS può visualizzare la legenda nella lista dei layer e nel compositore di stampe. La legenda WMS verrà mostrata solo se il server WMS ha le capability `GetLegendGraphic` e se il layer ha l'url `getCapability` specificato. Potrai anche selezionare uno stile per il layer.

Se `legendGraphic` è disponibile, viene mostrato sotto il layer. È piuttosto piccolo e dovrai cliccarci sopra per aprirlo nelle dimensioni reali (fatto dovuto alle limitazioni `QgsLegendInterface`). Cliccando sulla legenda del layer si aprirà una finestra con la legenda alla massima risoluzione.

Nel compositore di stampe, la legenda verrà mostrata alle sue dimensioni originali (scaricate). Puoi impostare la risoluzione della legenda attraverso le proprietà da `Legenda` -> `LegendGraphic WMS` fino a trovare il risultato che preferisci.

La legenda mostrerà informazioni contestuali riferite alla scala attuale. La legenda WMS verrà mostrata solo se il server WMS ha le capability `GetLegendGraphic` e se il layer ha le capability dell'url `getCapability` specificate, così puoi anche scegliere uno stile per il layer.

Limitazioni del client WMS

Non tutte le funzionalità client WMS sono state incluse in questa versione di QGIS. Le eccezioni più rilevanti sono le seguenti.

Modificare le impostazioni del layer WMS

Una volta completata la procedura mostrata dalla finestra  `Aggiungi layer WMS`, non puoi più modificarne i parametri. Una possibile soluzione è quella di eliminare completamente il layer e ricaricarlo reimpostando i parametri.

Server WMS che richiedono un'autenticazione

Attualmente sono accessibili server pubblici e server protetti. Puoi accedere ai server protetti con autenticazione pubblica. Puoi aggiungere le credenziali (opzionali) quando carichi un server WMS. Vedi sezione *Selezionare server WMS/WMTS* per ulteriori dettagli.

Suggerimento: Accesso ai layer OCG protetti

Se devi accedere a layer protetti con password, puoi usare `InteProxy` come proxy trasparente, che supporta molti metodi di autenticazione diversi. Ulteriori informazioni sono disponibili nel manuale di `InteProxy` nel sito web <http://inteproxy.wald.intevation.org>.

Suggerimento: WMS Mapserver QGIS

A partire dalla versione 1.7.0, in QGIS è stato implementato un server WMS 1.3.0 Mapserver. Ulteriori informazioni nel capitolo *QGIS come server dati OGC*.

14.1.2 Client WCS

Un servizio WCS fornisce accesso a dati raster che sono utili per la visualizzazione lato client, come input per modelli scientifici e per molti altri usi. Quali sono le caratteristiche che distinguono un servizio WCS dai servizi WFS e WMS? Mentre i servizi WFS e WMS sono istanze server, il servizio WCS permette al client di scegliere le porzioni di server che contengono le informazioni desiderate. Queste informazioni possono essere filtrate attraverso limiti spaziali o altre tipologie di interrogazioni.

QGIS ha un servizio WCS nativo e supporta le versioni 1.0 e 1.1 (che sono molto diverse fra di loro). Attualmente QGIS preferisce la versione 1.0, perché la 1.1 ha ancora troppi problemi (ogni server implementa i layer in modi diversi e con molti particolari).

Il supporto nativo WCS si occupa di tutte le richieste di rete e usa tutte le impostazioni standard di QGIS (specialmente i proxy). Puoi anche scegliere la modalità cache ('sempre cache', 'preferisci cache', 'preferisci rete', 'sempre rete') e il fornitore dati supporta anche la posizione temporale se questa è disponibile dal server.

14.1.3 Client WFS e WFS-T

In QGIS, un layer WFS si comporta come un qualsiasi altro vettore. Puoi identificare, selezionare elementi e visualizzare la tabella attributi. A partire da QGIS 1.6.0 puoi modificare il layer se il server lo supporta (WFS-T).

Normalmente la procedura per l'aggiunta di un layer WFS è molto simile a quella vista per i WMS. La differenza sta nel fatto che non ci sono server predefiniti, quindi devi aggiungere manualmente i server noti.

Caricare un layer WFS

Come esempio puoi caricare il server WFS DM Solutions e visualizzare un layer. L'indirizzo da inserire è: http://www2.dmsolutions.ca/cgi-bin/mswfs_gmap

1. Clicca sullo strumento  **Aggiungi layer WFS** nella barra dei layer ed apparirà la finestra di dialogo *Aggiungi layer WFS da server*.
2. Clicca su **[Nuovo]**.
3. Inserisci il nome 'DM Solutions'.
4. Inserisci l'indirizzo precedentemente indicato.
5. Clicca su **[OK]**.
6. Scegli 'DM Solutions' dal menu a tendina *Connessioni server* .
7. Clicca su **[Connetti]**.
8. Aspetta che vengano caricati tutti i layer.
9. Seleziona dalla lista il layer *Parks*.
10. Clicca su **[Applica]** per aggiungere il layer alla mappa.

Ogni impostazione proxy presente nelle preferenze viene automaticamente riconosciuta.

Nota che l'avanzamento della ricezione dei dati viene visualizzato nella parte inferiore sinistra della finestra principale di QGIS. Quando il layer è caricato, puoi identificare e selezionare alcuni elementi e visualizzare la tabella attributi.



Figura 14.4: Aggiungere un layer WFS 

Questo significa che attualmente è supportato solo WFS 1.0.0. Al momento non ci sono stati ancora molti test relativi a versioni di WFS basate su altri server. In caso di problemi con il plugin, non esitare a contattare il team di sviluppo. Vedi la sezione *Aiuto e supporto* per ulteriori informazioni sulle mailinglist.

Suggerimento: Cercare server WFS

Puoi ricercare ulteriori server WFS tramite Google o altro motore di ricerca preferito. Ci sono anche diversi elenchi di URL pubblici, alcuni dei quali aggiornati e altri non più mantenuti.

14.2 QGIS come server dati OGC

QGIS Server è un'implementazione WMS 1.3, WFS 1.0.0 e WCS 1 1.1.1 i con caratteristiche molto avanzate per la produzione di carte tematiche. QGIS Server è un'applicazione FastCGI/CGI (Common Gateway Interface) scritta in C++ e lavora insieme ad un server web (es. Apache, Lighttpd). QGIS Server è supportato dai progetti europei Orchestra e Sany e dalla città di Uster in Svizzera.

Usa QGIS come back end per la restituzione delle mappe e per le operazioni gis. Inoltre viene usata la libreria Qt per la grafica e il linguaggio di programmazione C++ per la piattaforma indipendente. A differenza di altri server WMS, QGIS Server usa regole cartografiche come linguaggio di configurazione, sia per la configurazione a livello server che per le configurazioni utente.

Come QGIS desktop, QGIS Server utilizza le stesse librerie di visualizzazione e le mappe che vengono pubblicate sulla rete hanno lo stesso aspetto che in desktop GIS.

In uno dei prossimi manuali forniremo un esempio per configurare un server QGIS. Per adesso ti consigliamo di leggere i materiali disponibili ad uno dei seguenti URL:

- http://karlinapp.ethz.ch/qgis_wms/
- http://hub.qgis.org/projects/quantum-gis/wiki/QGIS_Server_Tutorial
- <http://linfiniti.com/2010/08/qgis-mapserver-a-wms-server-for-the-masses/>

14.2.1 Installazione di esempio su Debian Squeeze

Di seguito è brevemente descritta un'installazione di esempio su Debian Squeeze. Anche molti altri OS hanno pacchetti precompilati per QGIS Server. Se devi compilare da codice sorgente, fai riferimento agli indirizzi web appena citati.

Oltre a QGIS e qgis-mapserver, devi avere installato un server web, come ad esempio apache2. Puoi installare tutti i pacchetti necessari, e relative dipendenze, con `aptitude` o `apt-get install`. Dopo l'installazione devi verificare il corretto funzionamento del server web e di QGIS Server. Assicurarti che apache sia stato avviato con `/etc/init.d/apache2 start`, apri un browser ed vai sull'URL: `http://localhost`: se hai configurato tutto correttamente dovrebbe apparire il messaggio 'It works!'.

Testa quindi l'installazione di QGIS server. Il file `qgis_mapserv.fcgi`, disponibile in `/usr/lib/cgi-bin/qgis_mapserv.fcgi` fornisce un WMS standard con visualizzati i confini dell'Alaska. Carica il WMS in QGIS con l'URL "`http://localhost/cgi-bin/qgis_mapserv.fcgi`", così come descritto nella Sezione *Selezionare server WMS/WMTS*.

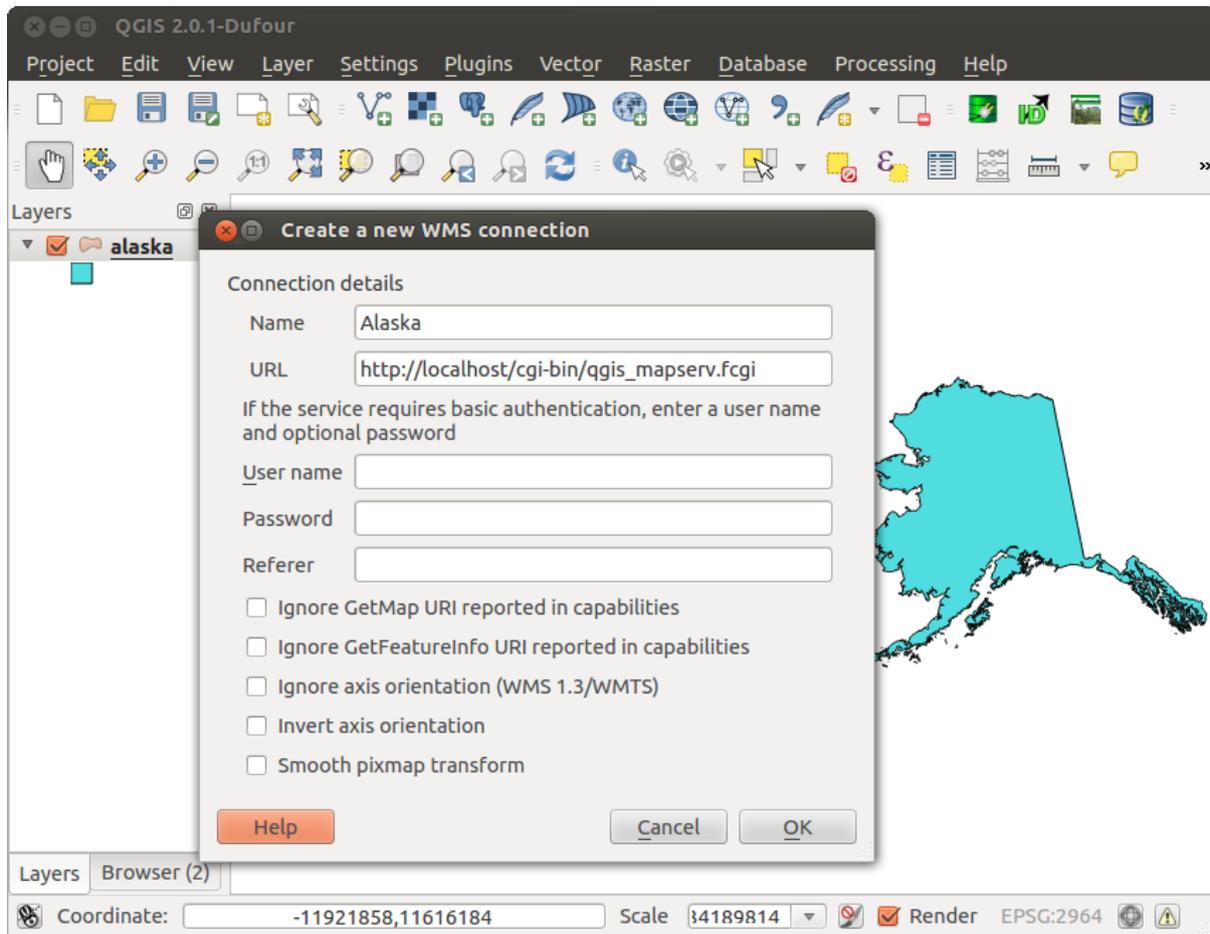


Figura 14.5: WMS standard con confini USA inclusi nel server QGIS (KDE) 

14.2.2 Creare un WMS/WFS/WCS da un progetto QGIS

Abbiamo creato un file di progetto di QGIS proprio per fornire un nuovo Server WMS, WFS o WCS. Abbiamo usato lo shapefile 'Alaska' dei dati campione di QGIS. Scegli i colori, lo stile dei layer e il SR se non sono ancora specificati.

Dalla finestra di dialogo *Progetto* → *Proprietà progetto* scegli il menu *Server OWS* e specifica qualche informazione sull'OWS nei campi *Capabilities del servizio*. Questo è quello che apparirà nelle risposte *GetCapabilities*

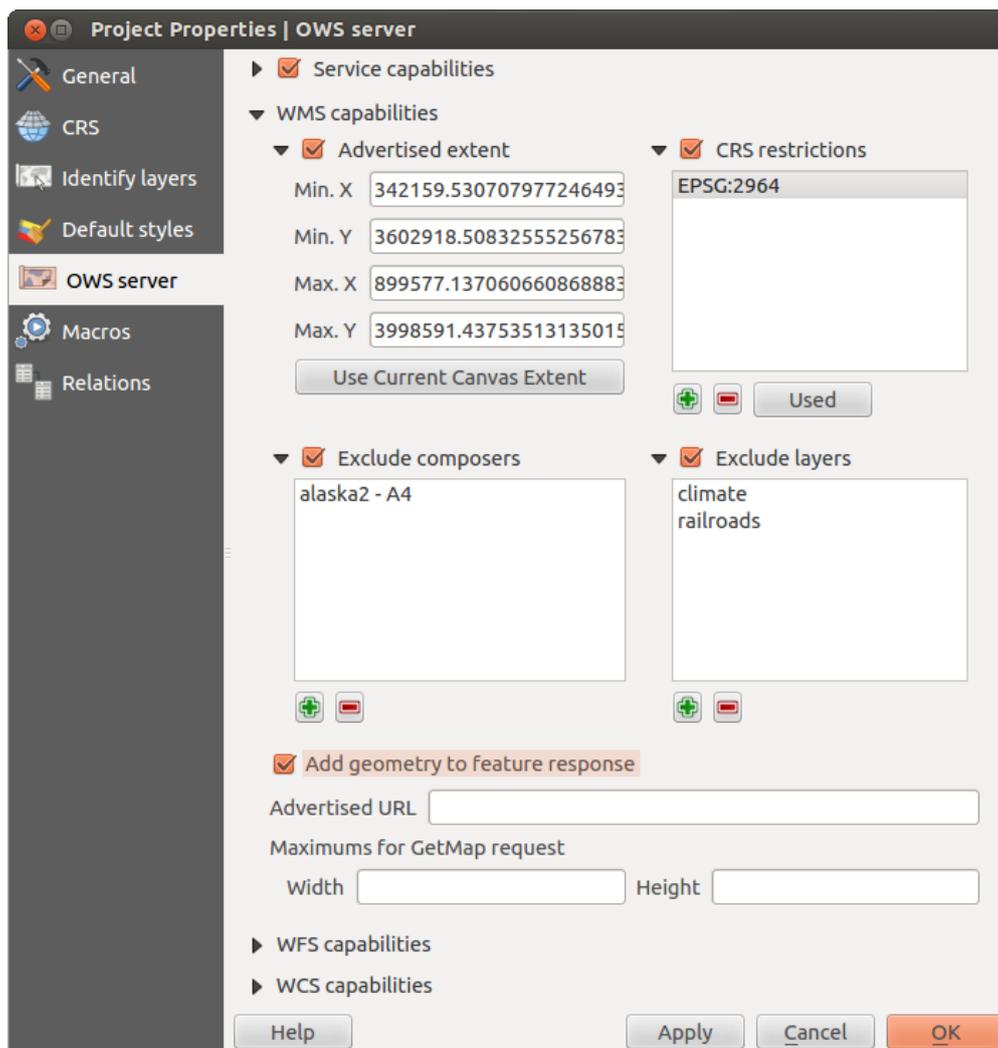


Figura 14.6: Definizioni per un Server WMS/WFS/WCS (KDE)

di WMS, WFS o WCS. Se non spunti la casella di controllo  *Capabilities del servizio*, QGIS Server userà le informazioni fornite dal file `wms_metadata.xml` caricato nella cartella `cgi-bin`.

Capabilities WMS

Nella sezione *Capabilities WMS* puoi scegliere l'estensione massima della risposta *GetCapabilities* inserendo i valori minimi e massimi X e Y nel campo *Estensione pubblicata*. Cliccando su *Imposta alla estensione della mappa* imposterai questi valori all'estensione attuale della mappa di QGIS. Spuntando la casella di controllo 

Restrizioni SR puoi scegliere in quali SR deve venire visualizzata la mappa. Usa il pulsante  per selezionare i SR dal selettore specifico oppure clicca *Usati* per aggiungere i SR usati nel progetto di QGIS.

Se hai impostato delle composizioni di stampa nel tuo progetto, queste verranno elencate nella risposta *GetCapabilities* e li potrai usare con la richiesta *GetPrint* per creare delle stampe, usando una delle composizioni come modello. Questa è un'estensione di QGIS per le specifiche WMS 1.3.0. Se vuoi escludere dalla pubblicazione WMS qualche composizione di stampa, spunta la casella di controllo  *Escludi composizioni* e clicca il pulsante . Poi, seleziona un compositore dalla finestra di dialogo *Scegli il compositore di stampe* in modo da aggiungere alla lista le composizioni che vuoi escludere.

Se vuoi escludere dalla pubblicazione WMS qualche layer o gruppi di layer, spunta la casella di controllo  *Escludi layer* e clicca sul pulsante . In questo modo si aprirà la finestra di dialogo *Seleziona numero limitato di layer e gruppi* che ti permetterà di scegliere i layer e i gruppo che non vuoi pubblicare. Usa i tasti `Shift` o `Ctrl` se vuoi selezionare più voci contemporaneamente.

Puoi ricevere le *GetFeatureInfo* richieste come testo normale, XML e GML. XML è il formato predefinito, il testo o il formato GML dipendono dal formato di output scelto dalla richiesta *GetFeatureInfo*.

Se vuoi, puoi spuntare la casella di controllo  *Aggiungi elemento alla risposta geometrie*. Questo includerà le geometrie degli elementi in testo normale nelle risposte *GetFeatureInfo*. Se vuoi che QGIS Server pubblichi le richieste URL nelle risposte capability WMS, inserisci l'URL nel campo *URL pubblicato*. Inoltre puoi rimpicciolire l'estensione massima della mappa restituita dalla richiesta *GetMap* inserendo la larghezza e l'altezza massima nei rispettivi campi in *Massimi per la richiesta GetMap*.

If one of your layers uses the Map Tip display (i.e. to show text using expressions) this will be listed inside the *GetFeatureInfo* output. If the layer uses a Value Map for one of his attributes, also this information will be shown in the *GetFeatureInfo* output.

Capabilities WFS

Nell'area *Capabilities WFS* puoi scegliere i layer che vuoi pubblicare come WFS e specificare se questi devono includere le opzioni *aggiorna*, *inserisci* ed *elimina*. Se inserisci un URL nel campo *URL pubblicato* della sezione *Capabilities WFS*, QGIS Server pubblicherà l'URL nelle risposte WFS *GetCapabilities*.

Capabilities WCS

Nell'area *Capabilities WCS* puoi scegliere i layer che vuoi pubblicare come WCS. Se inserisci un URL nel campo *URL pubblicato* della sezione *Capabilities WCS*, QGIS Server pubblicherà l'URL nelle risposte WCS *GetCapabilities*.

Salva la sessione nel file di progetto `alaska.qgs`. Per fornire il progetto come WMS/WFS, devi creare un'altra cartella `/usr/lib/cgi-bin/project` con privilegi di amministrazione e aggiungere il file di progetto `alaska.qgs` e una copia del file `qgis_mapserv.fcgi`. Questo è tutto!

Ora non resta che testare i progetti WMS, WFS e WCS. Aggiungi i WMS, WFS e WCS come descritto nella sezione *Caricare layer WMS/WFS* e *Client WCS* e carica i dati. L'URL è:

```
http://localhost/cgi-bin/project/qgis_mapserv.fcgi
```

OWS impostato correttamente

Per i vettori, il menu *Campi* della finestra di dialogo *Layer* → *Proprietà* ti permette di definire quali attributi vuoi pubblicare. In modo predefinito, tutti gli attributi vengono pubblicati. Se vuoi escludere qualche attributo dalla pubblicazione, togli la spunta alla casella di controllo corrispondente nelle colonne *WMS* o *WFS*.

Puoi sovrapporre i watermark sulla mappa creati dal WMS aggiungendo note testuali o note SVG al progetto. Vedi la sezione Note Testuali in *Strumenti generali* per capire come creare le note. Per fare in modo che le note vengano visualizzate come watermark nell'output WMS, la casella di controllo *Posizione fissa sulla mappa* non deve essere spuntata. Puoi accedere a questa opzione facendo doppio click sulla note quando lo strumento note è attivo. Per le note SVG, devi impostare il salvataggio del progetto con un percorso assoluto (nel menu *Generale* della finestra di dialogo *Progetto* → *Proprietà progetto*) oppure devi modificare manualmente il percorso all'immagine SVG in modo da ottenere un percorso relativo.

Parametri extra supportati dalla richiesta WMS GetMap

Nella richiesta WMS GetMap, QGIS Server accetta alcuni parametri extra ai parametri standard pur rispettando le specifiche OGC WMS 1.3.0:

- parametro **MAPPA**: in modo simile a MapServer, puoi usare il parametro `MAPPA` per specificare il percorso al file di progetto di QGIS. Puoi scegliere fra un percorso assoluto o uno relativo all'eseguibile del server (`qgis_mapserv.fcgi`). Se non specificato, il server QGIS cerca i file `.qgs` nella cartella dove è presente l'eseguibile del server.

Esempio:

```
http://localhost/cgi-bin/qgis_mapserv.fcgi?\nREQUEST=GetMap&MAP=/home/qgis/mymap.qgs&...
```

- Parametri **DPI**: puoi usare i parametri `DPI` per specificare la risoluzione di output.

Esempio:

```
http://localhost/cgi-bin/qgis_mapserv.fcgi?REQUEST=GetMap&DPI=300&...
```

- parametro **OPACITÀ**: puoi impostare l'opacità per un singolo layer o per un gruppo di layer. I valori vanno da 0 (completamente trasparente) a 255 (completamente opaco).

Esempio:

```
http://localhost/cgi-bin/qgis_mapserv.fcgi?\nREQUEST=GetMap&LAYERS=mylayer1,mylayer2&OPACITIES=125,200&...
```

QGIS Server logging

To log requests send to server, set the following environment variables:

- **QGIS_SERVER_LOG_FILE**: Specify path and filename. Make sure that server has proper permissions for writing to file. File should be created automatically, just send some requests to server. If it's not there, check permissions.
- **QGIS_SERVER_LOG_LEVEL**: Specify desired log level. Available values are:
 - 0 INFO (log all requests),
 - 1 WARNING,
 - 2 CRITICAL (log just critical errors, suitable for production purposes).

Esempio:

```
SetEnv QGIS_SERVER_LOG_FILE /var/tmp/qgislog.txt\nSetEnv QGIS_SERVER_LOG_LEVEL 0
```

Note

- When using Fcgid module use FcgidInitialEnv instead of SetEnv!
- Server logging is enabled also if executable is compiled in release mode.

Environment variables

- **QGIS_OPTIONS_PATH**: The variable specifies path to directory with settings. It works the same ways as QGIS application `-optionspath` option. It is looking for settings file in `<QGIS_OPTIONS_PATH>/QGIS/QGIS2.ini`. For exaple, to set QGIS server on Apache to use `/path/to/config/QGIS/QGIS2.ini` settings file, add to Apache config:

```
SetEnv QGIS_OPTIONS_PATH "/path/to/config/"
```

Lavorare con i dati GPS

15.1 Plugin GPS

15.1.1 Cos'è un GPS?

Il GPS, il Sistema di Posizionamento Globale, è un sistema satellitare che permette, a chiunque sia dotato di un ricevitore GPS, di individuare la sua posizione in qualunque parte del mondo. Il GPS è utilizzato come supporto per la navigazione, per esempio sugli aeroplani, sulle barche e anche dagli escursionisti. Il ricevitore GPS usa i segnali inviati dai satelliti per calcolare la propria latitudine, longitudine e (qualche volta) l'altitudine. Molti ricevitori possono memorizzare: la posizione (detta **waypoints**), la sequenza delle posizioni che formano una **route** e il tracciato o **track** dei movimenti che il ricevitore ha compiuto nel tempo. Waypoints, routes e tracks sono i tre principali elementi dei dati GPS. QGIS mostra i waypoints in un vettore puntuale, mentre routes e tracks sono mostrati in un vettore lineare.

15.1.2 Caricamento dei dati GPS da file

Ci sono una dozzina di formati di file diversi per memorizzare dati GPS. Il formato utilizzato da QGIS è chiamato GPX (GPS eXchange format), il quale è un formato standard di interscambio che può contenere waypoint, route e track nello stesso file.

Per caricare un file GPX, per prima cosa è necessario installare il plugin aprendo la finestra di dialogo Gestisci e installa Plugin. *Plugins* →  *Gestisci e installa Plugin* Spuntare la casella  *Strumenti GPS*. Una volta installato il plugin, verranno visualizzati nella barra degli strumenti due pulsanti che rappresentano dei piccoli dispositivi GPS.

-  Crea un nuovo vettore GPX
-  Strumenti GPS

Per lavorare con i dati GPS è fornito un file GPX di esempio, il quale è disponibile nell'insieme di dati di QGIS: file: *qgis_sample_data/gps/national_monuments.gpx*. Vedere la sezione *Dati campione* per maggiori informazioni sull'insieme di dati a disposizione.

1. Selezionare *Vettore* → *GPS* → *Strumenti GPS* o cliccare sull'icona  GPS Tools presente sulla barra degli strumenti e aprire la scheda *Carica file GPX* (vedere [figure_GPS_1](#))
2. Navigare all'interno della cartella *qgis_sample_data/gps/*, selezionare il file *GPX national_monuments.gpx* e cliccare **[Apri]**.

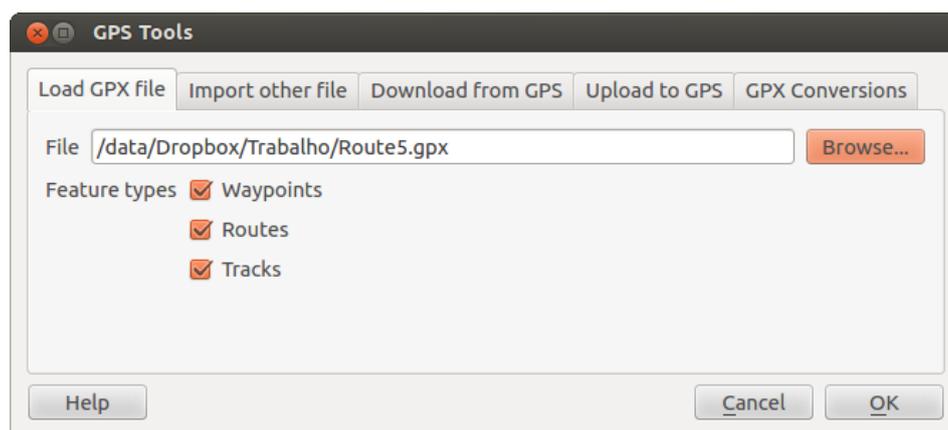


Figura 15.1: Finestra di dialogo degli *Strumenti GPS* 

Utilizzare il pulsante [**Sfoggia...**] per selezionare il file GPX, e poi spuntare le caselle per selezionare il tipo di dati che si desidera caricare dal file GPX. Ogni tipo di dato sarà caricato in layer separati quando viene premuto il pulsante [**OK**]. Il file `national_monuments.gpx` include solamente dati puntuali.

Nota: I dispositivi GPS ti consentono di registrare i dati in differenti sistemi di coordinate. Quando scarichi un file GPX (dal dispositivo GPS o da un sito internet) e lo carichi in QGIS, assicurati che i dati presenti nel file GPX siano in WGS 84 (latitudine/longitudine). QGIS si aspetta questo SR, così come indicato dalle specifiche ufficiali dei formato GPX. Vedere <http://www.topografix.com/GPX/1/1/>.

15.1.3 GPSTabel

Dato che QGIS usa i file GPX, ti occorrerà un modo per convertire gli altri formati GPS in GPX. Questa operazione potrà essere compiuta, sulla maggior parte dei formati utilizzati, con il programma gratuito GPSTabel, il quale è disponibile sul sito <http://www.gpsbabel.org>. Questo programma può anche trasferire i dati GPS dal tuo computer al dispositivo GPS e viceversa. QGIS usa GPSTabel per effettuare tutte queste operazioni, perciò è consigliato installarlo. Tuttavia, se vuoi soltanto caricare dati GPS contenuti in files GPX, non è necessario avere il programma installato. La versione 1.2.3 di GPSTabel funziona con QGIS, ma potrai utilizzare versioni successive senza problemi.

15.1.4 Importare dati GPS

Per importare dei dati che non sono dei file GPX, utilizza lo strumento *Importa altro file* presente nella finestra di dialogo degli Strumenti GPS. Qui, puoi scegliere il file da importare (e il formato dei files), il tipo di dato da estrarre da esso, dove vuoi salvare il file una volta convertito in GPX e il nome che vuoi dare al nuovo vettore. Nota che non tutti i formati di dati GPS supportano tutti e tre i tipi di dato, così per alcuni formati potrai scegliere solo uno e due tipi.

15.1.5 Scaricare dati GPS da un dispositivo

QGIS utilizza GPSTabel per scaricare dati da un dispositivo GPS, e caricarli direttamente come nuovi vettori. Per fare questo si utilizza la scheda *Scarica dal GPS* presente nella finestra di dialogo degli Strumenti GPS (vedi [Figure_GPS_2](#)). Qui, è possibile selezionare la periferica GPS, la porta alla quale il dispositivo è connesso (o la porta USB se il GPS supporta questa opzione), il tipo di dato che si desidera scaricare, il file GPX in cui i dati dovranno essere memorizzati, e il nome del nuovo layer.

GPSTabel comunica con il GPS in base al tipo di dispositivo che viene selezionato nel menu. Se nessuna delle opzioni disponibili è compatibile con il proprio dispositivo GPS è possibile creare un nuovo tipo (vedi sezione [:ref: defining-new-device](#)).

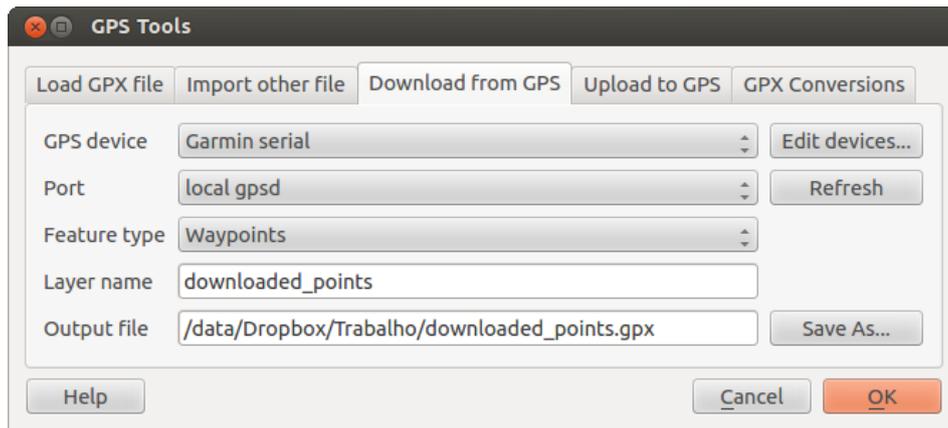


Figura 15.2: Lo strumento di scaricamento

La porta potrebbe essere il nome del file o qualche altro termine che il tuo sistema operativo riconosce come porta fisica alla quale è connesso il dispositivo GPS. Essa potrebbe essere un'uscita USB, nel caso di dispositivi abilitati per l'USB.

-  Nei sistemi Linux è qualcosa di simile a `/dev/ttyS0` or `/dev/ttyS1`
-  In Windows è COM1 or COM2.

Quando si clicca su **[OK]** i dati saranno scaricati dal dispositivo e appariranno come nuovi vettori in QGIS.

15.1.6 Caricare dati GPS sul dispositivo

Puoi anche caricare direttamente dei dati vettoriali sulla periferica GPS utilizzando la scheda *Carica sul GPS* presente nella finestra di dialogo Strumenti GPS. Per effettuare questa operazione devi semplicemente selezionare il layer che desideri caricare (che deve essere un layer GPX), la periferica GPS utilizzata, e la porta (o l'usb) alla quale la periferica è collegata. Come per lo scaricamento dei dati, anche in questo caso è possibile specificare un nuovo tipo di periferica, nel caso non fosse presente nella lista proposta.

Questo strumento è molto utile in combinazione con le capacità di editing dei dati vettoriali di QGIS. Permette di caricare una mappa, creare dei waypoints e delle routes, e successivamente caricarli nel dispositivo per poi utilizzarli.

15.1.7 Definire un nuovo tipo di dispositivo

Ci sono molteplici tipologie di dispositivi GPS. Gli sviluppatori di QGIS non possono configurare tutti i dispositivi, perciò se hai un dispositivo non compatibile con quelli messi a disposizione nelle schede *Scarica dal GPS* e *Carica su GPS*, ti puoi configurare il tuo dispositivo. Per fare questo puoi utilizzare l'editor dei dispositivi GPS, che si avvia cliccando sul bottone **[Modifica periferiche]** sia nella scheda di download che in quella di upload dei dati.

Per configurare una nuova periferica devi semplicemente cliccare il pulsante **[Nuovo]**, inserire il nome del dispositivo, i vari comandi di download e di upload, e cliccare il pulsante **[Aggiorna]**. Il nome del dispositivo sarà inserito nella lista delle periferiche GPS presenti nelle schede di download e upload dei dati. Il comando di download è il comando che viene utilizzato per scaricare i dati dal dispositivo come file GPX. Solitamente è un comando di GPSTools, ma puoi utilizzare qualsiasi altro programma a linea di comando che permette di creare un file GPX. QGIS rimpiazzerà i tasti `%type`, `%in`, e `%out` quando elaborerà il comando.

`%type` sarà sostituito da `-w` se stai scaricando dei waypoints, `-r` se stai scaricando delle routes e `-t` se stai scaricando dei tracks. Queste sono le opzioni che comunicano a GPSTools quali elementi scaricare.

`%in` sarà sostituito dal nome della porta selezionata nella finestra di download, `%out` sarà sostituito dal nome del file GPX nel quale verranno salvati i dati. Quindi se configuri una nuova periferica con il comando `gpsbabel %type -i garmin -o gpx %in %out` (questo è il comando di download per i dispositivi predefiniti

‘Garmin serial’) e lo usi per scaricare dei waypoints tramite una porta `/dev/ttyS0` in un file GPX denominato `output.gpx`, QGIS leggerà la stringa e avvierà il comando `gpsbabel -w -i garmin -o gpx /dev/ttyS0 output.gpx`.

Il comando di upload è il comando che viene utilizzato per caricare dati sul dispositivo. Vengono utilizzati i stessi tasti, ma `%in` è utilizzato per indicare il nome del file GPX che contiene il layer in caricamento, e `%out` viene sostituito dal nome della porta.

Puoi avere maggiori informazioni su GPSBabel e sulle opzioni utilizzabili tramite linea di comando sul sito <http://www.gpsbabel.org>.

Una volta che avrai creato una nuova periferica, essa apparirà nella lista dei dispositivi presente sia nella scheda Scarica dal GPS sia nella scheda Carica sul GPS.

15.1.8 Scaricare points/tracks dall’unità GPS

Come descritto nei paragrafi precedenti QGIS usa GPSBabel per scaricare punti/track direttamente nel progetto. QGIS si configura con impostazioni predefinite per scaricare da dispositivi Garmin. Purtroppo vi è un *bug* #6318 <<http://hub.qgis.org/issues/6318>> che non permette di creare altre impostazioni, così al momento è illimitato solo alle unità USB Garmi scaricare direttamente QGIS utilizzando gli strumenti GPS.

Garmin GPSMAP 60cs

MS Windows

Installare i drivers Garmin USB dal sito http://www8.garmin.com/support/download_details.jsp?id=591

Connettere l’unità. Aprire gli Strumenti GPS e impostare *Periferica GPS=garmin serial* e ‘Porta=usb:’. Riempire i campi `:guilabel: ‘Nome layer’` and `:guilabel: ‘File di output’`. A volte si possono avere dei problemi nel salvataggio dei dati in certe cartelle, si consiglia di utilizzare un percorso del tipo `‘c:\temp`.

Ubuntu/Mint GNU/Linux

Per prima cosa occorre risolvere un problema inerente i permessi di accesso alla periferica, seguendo quanto scritto qui https://wiki.openstreetmap.org/wiki/USB_Garmin_on_GNU/Linux. Puoi provare a creare un file `/etc/udev/rules.d/51-garmin.rules` contenente il seguente codice:

```
ATTRS{idVendor}=="091e", ATTRS{idProduct}=="0003", MODE="666"
```

Successivamente occorre essere sicuri che il modulo del kernel ‘garmin_gps’ non sia caricato

```
rmmod garmin_gps
```

e quindi puoi utilizzare gli strumenti GPS. Purtroppo sembra che ci sia un *bug* # 7182 <<http://hub.qgis.org/issues/7182>> _ e di solito QGIS si blocca più volte prima della fine.

Data logger BTGP-38KM (solo Bluetooth)

MS Windows

Il baco già discusso non consente di scaricare i dati tramite QGIS, per cui è necessario utilizzare GPSBabel dalla riga di comando o tramite la sua interfaccia. Il comando da eseguire è

```
gpsbabel -t -i skytraq,baud=9600,initbaud=9600 -f COM9 -o gpx -F C:/GPX/aaa.gpx
```

Ubuntu/Mint GNU/Linux

Utilizzare lo stesso comando (o gli stessi parametri, se usate la GUI di GPSBabel). Su Linux potrebbe capitare di vedere un messaggio tipo

```
skytraq: Too many read errors on serial port
```

Si tratta solo di spegnere e riaccendere il data logger e ritentare

BlueMax GPS-4044 datalogger (sia BT che USB)

MS Windows

Nota: Ha bisogno di installare i propri driver prima di essere utilizzato su Windows 7. Si veda il sito del costruttore per il file corretto da scaricare.

Scaricando con GSPBabel, sia con USB che BT, si ottiene sempre un errore tipo

```
gpsbabel -t -i mtk -f COM12 -o gpx -F C:/temp/test.gpx
mtk_logger: Can't create temporary file data.bin
Error running gpsbabel: Process exited unsuccessfully with code 1
```

Ubuntu/Mint GNU/Linux

con USB

Dopo aver collegato il cavo, usare il comando `dmesg` per capire quale porta viene utilizzata, ad esempio `/dev/ttyACM3`. Poi, come al solito, utilizzare GPSBabel dalla riga di comando o dalla GUI

```
gpsbabel -t -i mtk -f /dev/ttyACM3 -o gpx -F /home/user/bluemax.gpx
```

Con Bluetooth

Utilizzare il Gestore di dispositivi Blueman per accoppiare il dispositivo e renderlo disponibile tramite una porta di sistema, poi eseguire GPSBabel

```
gpsbabel -t -i mtk -f /dev/rfcomm0 -o gpx -F /home/user/bluemax_bt.gpx
```

15.2 Tracciamento live GPS

To activate live GPS tracking in QGIS, you need to select *Settings* → *Panels*  *GPS information*. You will get a new docked window on the left side of the canvas.

There are four possible screens in this GPS tracking window:

-  GPS position coordinates and an interface for manually entering vertices and features
-  GPS signal strength of satellite connections
-  GPS polar screen showing number and polar position of satellites
-  GPS options screen (see [figure_gps_options](#))

With a plugged-in GPS receiver (has to be supported by your operating system), a simple click on **[Connect]** connects the GPS to QGIS. A second click (now on **[Disconnect]**) disconnects the GPS receiver from your computer. For GNU/Linux, `gpsd` support is integrated to support connection to most GPS receivers. Therefore, you first have to configure `gpsd` properly to connect QGIS to it.

Avvertimento: If you want to record your position to the canvas, you have to create a new vector layer first and switch it to editable status to be able to record your track.

15.2.1 Posizione e attributi aggiuntivi

 If the GPS is receiving signals from satellites, you will see your position in latitude, longitude and altitude together with additional attributes.

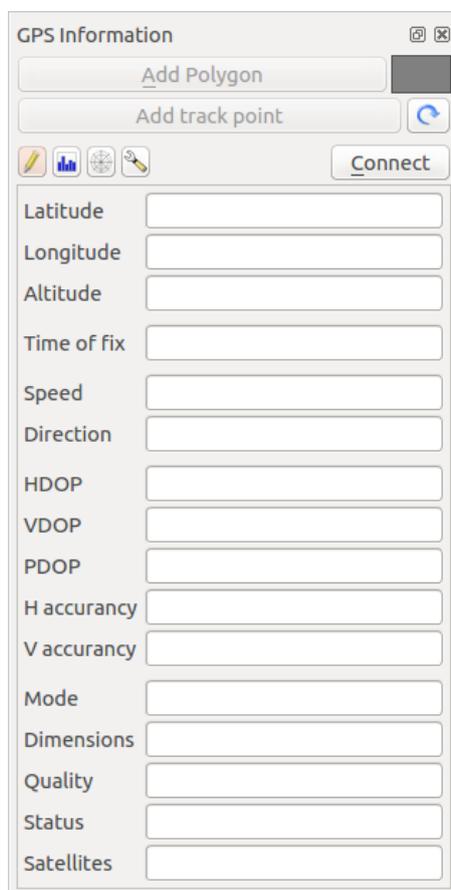


Figura 15.3: GPS tracking position and additional attributes 

15.2.2 Potenza del segnale GPS

 Here, you can see the signal strength of the satellites you are receiving signals from.

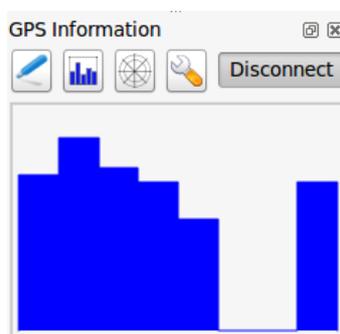


Figura 15.4: GPS tracking signal strength 

15.2.3 Finestra delle coordinate polari del GPS

 Se si desidera visualizzare la posizione dei satelliti nel cielo, è necessario passare alla visualizzazione polare. E' possibile anche vedere il numero identificativo ID dei satelliti dai cui il ricevitore sta ricevendo il segnale.

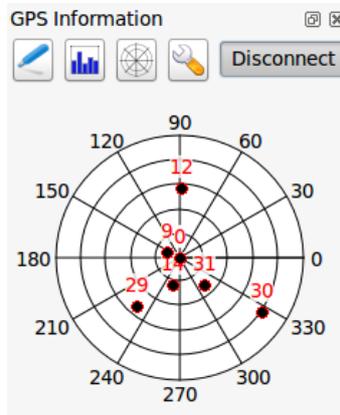


Figura 15.5: GPS tracking polar window 

15.2.4 Opzioni GPS

 In case of connection problems, you can switch between:

- *Autodetect*
- *Internal*
- *Serial device*
- *gpsd* (selecting the Host, Port and Device your GPS is connected to)

Cliccare nuovamente [**Connect**] per iniziare la connessione al ricevitore GPS.

You can activate *Automatically save added features* when you are in editing mode. Or you can activate *Automatically add points* to the map canvas with a certain width and color.

Activating *Cursor*, you can use a slider  to shrink and grow the position cursor on the canvas.

Activating *Map centering* allows you to decide in which way the canvas will be updated. This includes 'always', 'when leaving', if your recorded coordinates start to move out of the canvas, or 'never', to keep map extent.

Finally, you can activate *Log file* and define a path and a file where log messages about the GPS tracking are logged.

If you want to set a feature manually, you have to go back to  *Position* and click on [**Add Point**] or [**Add track point**].

15.2.5 Connect to a Bluetooth GPS for live tracking

With QGIS you can connect a Bluetooth GPS for field data collection. To perform this task you need a GPS Bluetooth device and a Bluetooth receiver on your computer.

At first you must let your GPS device be recognized and paired to the computer. Turn on the GPS, go to the Bluetooth icon on your notification area and search for a New Device.

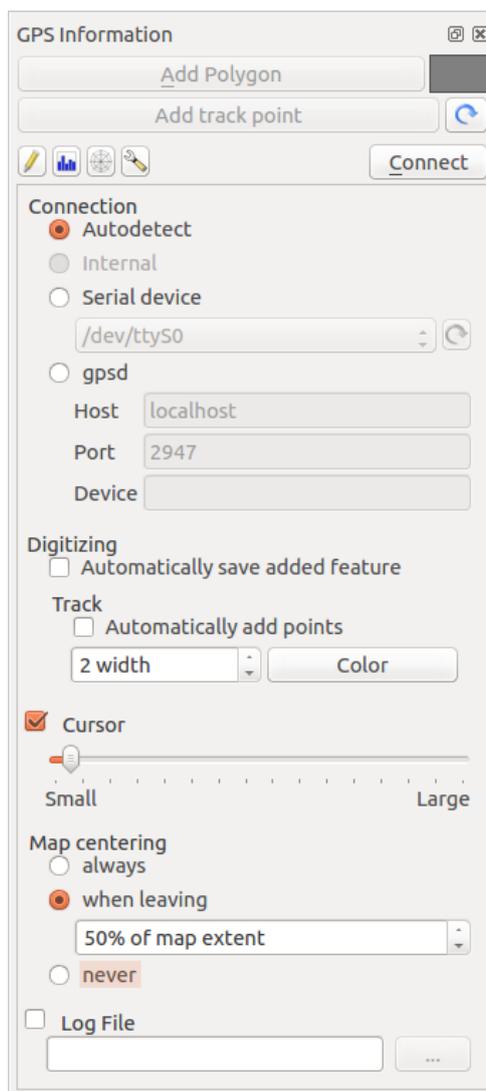


Figura 15.6: GPS tracking options window 

On the right side of the Device selection mask make sure that all devices are selected so your GPS unit will probably appear among those available. In the next step a serial connection service should be available, select it and click on **[Configure]** button.

Remember the number of the COM port assigned to the GPS connection as resulting by the Bluetooth properties.

After the GPS has been recognized, make the pairing for the connection. Usually the authorization code is 0000.

Now open *GPS information* panel and switch to  GPS options screen. Select the COM port assigned to the GPS connection and click the **[Connect]**. After a while a cursor indicating your position should appear.

If QGIS can't receive GPS data, then you should restart your GPS device, wait 5-10 seconds then try to connect again. Usually this solution work. If you receive again a connection error make sure you don't have another Bluetooth receiver near you, paired with the same GPS unit.

15.2.6 Using GPSPMAP 60cs

MS Windows

Easiest way to make it work is to use a middleware (freeware, not open) called [GPSGate](#).

Launch the program, make it scan for GPS devices (works for both USB and BT ones) and then in QGIS just click **[Connect]** in the Live tracking panel using the  *Autodetect* mode.

Ubuntu/Mint GNU/Linux

As for Windows the easiest way is to use a server in the middle, in this case GPSPD, so

```
sudo apt-get install gpsd
```

Then load the `garmin_gps` kernel module

```
sudo modprobe garmin_gps
```

And then connect the unit. Then check with `dmesg` the actual device being used by the unit, for example `/dev/ttyUSB0`. Now you can launch `gpsd`

```
gpsd /dev/ttyUSB0
```

And finally connect with the QGIS live tracking tool.

15.2.7 Using BTGP-38KM datalogger (only Bluetooth)

Using GPSPD (under Linux) or GPSGate (under Windows) is effortless.

15.2.8 Using BlueMax GPS-4044 datalogger (both BT and USB)

MS Windows

The live tracking works for both USB and BT modes, by using GPSGate or even without it, just use the  *Autodetect* mode, or point the tool the right port.

Ubuntu/Mint GNU/Linux

For USB

The live tracking works both with GPSPD

```
gpsd /dev/ttyACM3
```

or without it, by connecting the QGIS live tracking tool directly to the device (for example `/dev/ttyACM3`).

For Bluetooth

The live tracking works both with GPSD

```
gpsd /dev/rfcomm0
```

or without it, by connecting the QGIS live tracking tool directly to the device (for example `/dev/rfcomm0`).

.

Integrazione con GRASS GIS

The GRASS plugin provides access to GRASS GIS databases and functionalities (see GRASS-PROJECT in *Letteratura e riferimenti web*). This includes visualizing GRASS raster and vector layers, digitizing vector layers, editing vector attributes, creating new vector layers and analysing GRASS 2-D and 3-D data with more than 400 GRASS modules.

In this section, we'll introduce the plugin functionalities and give some examples of managing and working with GRASS data. The following main features are provided with the toolbar menu when you start the GRASS plugin, as described in section [sec_starting_grass](#):

-  Apri mapset
-  Nuovo mapset
-  Chiudi mapset
-  Aggiungi vettore GRASS
-  Aggiungi raster GRASS
-  Crea nuovo vettore GRASS
-  Modifica vettore GRASS
-  Apri strumenti GRASS
-  Visualizza la regione di GRASS attuale
-  Modifica la regione di GRASS attuale

16.1 Avviare il plugin GRASS

To use GRASS functionalities and/or visualize GRASS vector and raster layers in QGIS, you must select and load the GRASS plugin with the Plugin Manager. Therefore, go to the menu *Plugins* →  *Manage Plugins*, select *GRASS* and click [OK].

You can now start loading raster and vector layers from an existing GRASS LOCATION (see section [sec_load_grassdata](#)). Or, you can create a new GRASS LOCATION with QGIS (see section [Creare una nuova LOCATION GRASS](#)) and import some raster and vector data (see section [Importare dati nelle LOCATION GRASS](#)) for further analysis with the GRASS Toolbox (see section [The GRASS Toolbox](#)).

16.2 Caricare layer raster e vettoriali GRASS

With the GRASS plugin, you can load vector or raster layers using the appropriate button on the toolbar menu. As an example, we will use the QGIS Alaska dataset (see section *Dati campione*). It includes a small sample GRASS LOCATION with three vector layers and one raster elevation map.

1. Create a new folder called `grassdata`, download the QGIS 'Alaska' dataset `qgis_sample_data.zip` from <http://download.osgeo.org/qgis/data/> and unzip the file into `grassdata`.
2. Start QGIS.
3. If not already done in a previous QGIS session, load the GRASS plugin clicking on *Plugins* →  *Manage Plugins* and activate  *GRASS*. The GRASS toolbar appears in the QGIS main window.
4. Nella barra degli strumenti di GRASS, selezionare il pulsante  *Apri mapset* per evidenziare il *MAPSET* wizard.
5. For `GISDBASE`, browse and select or enter the path to the newly created folder `grassdata`.
6. Dovrebbe ora essere possibile selezionare la *LOCATION*  `alaska` e il *MAPSET*  `demo`.
7. Cliccare su **[OK]**. Si noti che ora alcuni degli strumenti precedentemente disabilitati sono divenuti attivi.
8. Cliccare su  *Aggiungi raster GRASS*, scegliere la mappa denominata `gtopo30` e cliccare su **[OK]**. Verrà visualizzato il layer delle quote del terreno.
9. Click on  *Add GRASS vector layer*, choose the map name `alaska` and click **[OK]**. The Alaska boundary vector layer will be overlaid on top of the `gtopo30` map. You can now adapt the layer properties as described in chapter *Proprietà dei vettori* (e.g., change opacity, fill and outline color).
10. Also load the other two vector layers, `rivers` and `airports`, and adapt their properties.

As you see, it is very simple to load GRASS raster and vector layers in QGIS. See the following sections for editing GRASS data and creating a new LOCATION. More sample GRASS LOCATIONS are available at the GRASS website at <http://grass.osgeo.org/download/sample-data/>.

Suggerimento: Caricare dati GRASS

If you have problems loading data or QGIS terminates abnormally, check to make sure you have loaded the GRASS plugin properly as described in section *Avviare il plugin GRASS*.

16.3 LOCATION e MAPSET in GRASS

GRASS data are stored in a directory referred to as GISDBASE. This directory, often called `grassdata`, must be created before you start working with the GRASS plugin in QGIS. Within this directory, the GRASS GIS data are organized by projects stored in subdirectories called *LOCATIONS*. Each *LOCATION* is defined by its coordinate system, map projection and geographical boundaries. Each *LOCATION* can have several *MAPSETS* (subdirectories of the *LOCATION*) that are used to subdivide the project into different topics or subregions, or as workspaces for individual team members (see Neteler & Mitasova 2008 in *Letteratura e riferimenti web*). In order to analyze vector and raster layers with GRASS modules, you must import them into a GRASS *LOCATION*. (This is not strictly true – with the GRASS modules `r.external` and `v.external` you can create read-only links to external GDAL/OGR-supported datasets without importing them. But because this is not the usual way for beginners to work with GRASS, this functionality will not be described here.)

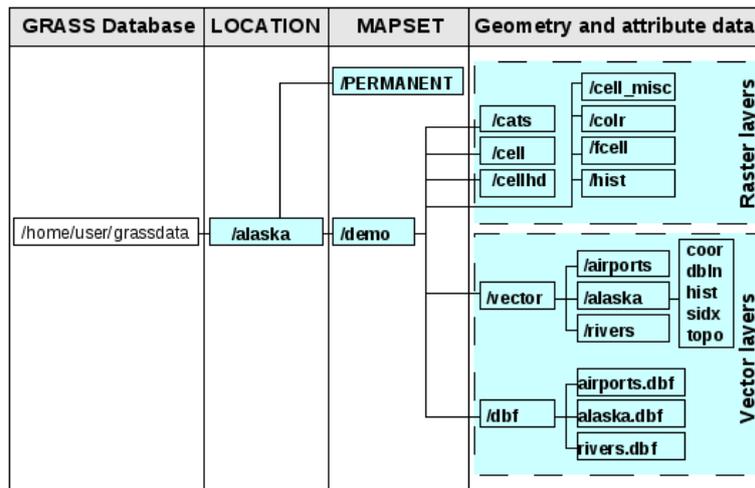


Figura 16.1: Dati di GRASS all'interno della LOCATION Alaska

16.3.1 Creare una nuova LOCATION GRASS

As an example, here is how the sample GRASS LOCATION `alaska`, which is projected in Albers Equal Area projection with unit feet was created for the QGIS sample dataset. This sample GRASS LOCATION `alaska` will be used for all examples and exercises in the following GRASS-related sections. It is useful to download and install the dataset on your computer (see *Dati campione*).

1. Start QGIS and make sure the GRASS plugin is loaded.
2. Visualize the `alaska.shp` shapefile (see section *Caricare uno shapefile*) from the QGIS Alaska dataset (see *Dati campione*).
3. In the GRASS toolbar, click on the  **New mapset** icon to bring up the *MAPSET* wizard.
4. Select an existing GRASS database (GISDBASE) folder `grassdata`, or create one for the new LOCATION using a file manager on your computer. Then click **[Next]**.
5. We can use this wizard to create a new MAPSET within an existing LOCATION (see section *Aggiungere un nuovo MAPSET*) or to create a new LOCATION altogether. Select *Create new location* (see *figure_grass_location_2*).
6. Enter a name for the LOCATION – we used ‘alaska’ – and click **[Next]**.
7. Define the projection by clicking on the radio button *Projection* to enable the projection list.
8. We are using Albers Equal Area Alaska (feet) projection. Since we happen to know that it is represented by the EPSG ID 2964, we enter it in the search box. (Note: If you want to repeat this process for another LOCATION and projection and haven’t memorized the EPSG ID, click on the  **CRS Status** icon in the lower right-hand corner of the status bar (see section *Lavorare con le proiezioni*)).
9. In *Filter*, insert 2964 to select the projection.
10. Cliccare su **[Next]**.
11. To define the default region, we have to enter the LOCATION bounds in the north, south, east, and west directions. Here, we simply click on the button **[Set current lqgl extent]**, to apply the extent of the loaded layer `alaska.shp` as the GRASS default region extent.
12. Cliccare su **[Next]**.
13. We also need to define a MAPSET within our new LOCATION (this is necessary when creating a new LOCATION). You can name it whatever you like - we used ‘demo’. GRASS automatically creates a special MAPSET called `PERMANENT`, designed to store the core data for the project, its default spatial extent and coordinate system definitions (see Neteler & Mitasova 2008 in *Letteratura e riferimenti web*).

14. Controllare il riassunto per assicurarsi che le impostazioni siano corrette e cliccare su **[Finish]**.
15. The new LOCATION, 'alaska', and two MAPSETs, 'demo' and 'PERMANENT', are created. The currently opened working set is 'demo', as you defined.
16. Si noti che alcuni strumenti della barra di GRASS precedentemente disabilitati sono ora attivi.

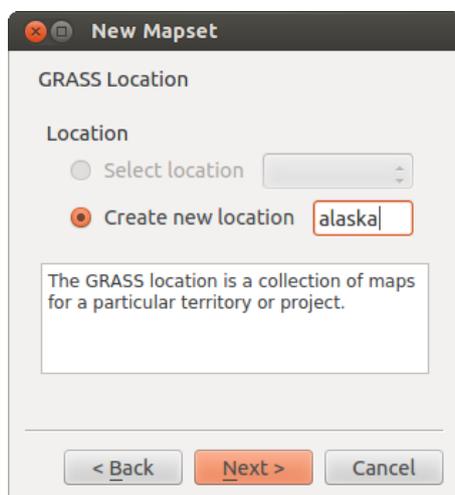


Figura 16.2: Creating a new GRASS LOCATION or a new MAPSET in QGIS

If that seemed like a lot of steps, it's really not all that bad and a very quick way to create a LOCATION. The LOCATION 'alaska' is now ready for data import (see section *Importare dati nelle LOCATION GRASS*). You can also use the already-existing vector and raster data in the sample GRASS LOCATION 'alaska', included in the QGIS 'Alaska' dataset *Dati campione*, and move on to section *Il modello dati vettoriale di GRASS*.

16.3.2 Aggiungere un nuovo MAPSET

A user has write access only to a GRASS MAPSET he or she created. This means that besides access to your own MAPSET, you can read maps in other users' MAPSETs (and they can read yours), but you can modify or remove only the maps in your own MAPSET.

All MAPSETs include a WIND file that stores the current boundary coordinate values and the currently selected raster resolution (see Neteler & Mitasova 2008 in *Letteratura e riferimenti web*, and section *Lo strumento Regione di GRASS*).

1. Start QGIS and make sure the GRASS plugin is loaded.
2. In the GRASS toolbar, click on the  New mapset icon to bring up the MAPSET wizard.
3. Select the GRASS database (GISDBASE) folder `grassdata` with the LOCATION 'alaska', where we want to add a further MAPSET called 'test'.
4. Cliccare su **[Next]**.
5. We can use this wizard to create a new MAPSET within an existing LOCATION or to create a new LOCATION altogether. Click on the radio button *Select location* (see [figure_grass_location_2](#)) and click **[Next]**.
6. Enter the name `test` for the new MAPSET. Below in the wizard, you see a list of existing MAPSETs and corresponding owners.
7. Cliccare su **[Next]**, controllare il riassunto per assicurarsi che le impostazioni siano corrette e cliccare su **[Finish]**.

16.4 Importare dati nelle LOCATION GRASS

This section gives an example of how to import raster and vector data into the 'alaska' GRASS LOCATION provided by the QGIS 'Alaska' dataset. Therefore, we use the landcover raster map `landcover.img` and the vector GML file `lakes.gml` from the QGIS 'Alaska' dataset (see *Dati campione*).

1. Start QGIS and make sure the GRASS plugin is loaded.
2. In the GRASS toolbar, click the  Open MAPSET icon to bring up the MAPSET wizard.
3. Select as GRASS database the folder `grassdata` in the QGIS Alaska dataset, as LOCATION 'alaska', as MAPSET 'demo' and click [OK].
4. Cliccare ora sullo strumento  Apri strumenti GRASS. Apparirà la finestra degli strumenti di GRASS (sezione *The GRASS Toolbox*).
5. To import the raster map `landcover.img`, click the module `r.in.gdal` in the *Modules Tree* tab. This GRASS module allows you to import GDAL-supported raster files into a GRASS LOCATION. The module dialog for `r.in.gdal` appears.
6. Browse to the folder `raster` in the QGIS 'Alaska' dataset and select the file `landcover.img`.
7. As raster output name, define `landcover_grass` and click [Run]. In the *Output* tab, you see the currently running GRASS command `r.in.gdal -o input=/path/to/landcover.img output=landcover_grass`.
8. When it says **Successfully finished**, click [View output]. The `landcover_grass` raster layer is now imported into GRASS and will be visualized in the QGIS canvas.
9. To import the vector GML file `lakes.gml`, click the module `v.in.ogr` in the *Modules Tree* tab. This GRASS module allows you to import OGR-supported vector files into a GRASS LOCATION. The module dialog for `v.in.ogr` appears.
10. Browse to the folder `gml` in the QGIS 'Alaska' dataset and select the file `lakes.gml` as OGR file.
11. As vector output name, define `lakes_grass` and click [Run]. You don't have to care about the other options in this example. In the *Output* tab you see the currently running GRASS command `v.in.ogr -o dsn=/path/to/lakes.gml output=lakes_grass`.
12. When it says **Successfully finished**, click [View output]. The `lakes_grass` vector layer is now imported into GRASS and will be visualized in the QGIS canvas.

16.5 Il modello dati vettoriale di GRASS

It is important to understand the GRASS vector data model prior to digitizing.

In generale, GRASS usa un modello dati vettoriale topologico.

Questo significa che le aree non sono rappresentate con poligoni chiusi singoli, ma da uno o più contorni (boundary). Un contorno tra due aree adiacenti è digitalizzato una sola volta e condiviso da entrambe le aree. Perché un'area sia topologicamente corretta, i contorni devono essere connessi senza soluzione di continuità. Un'area è identificata (ed etichettata) dal suo **centroide**.

Besides boundaries and centroids, a vector map can also contain points and lines. All these geometry elements can be mixed in one vector and will be represented in different so-called 'layers' inside one GRASS vector map. So in GRASS, a layer is not a vector or raster map but a level inside a vector layer. This is important to distinguish carefully. (Although it is possible to mix geometry elements, it is unusual and, even in GRASS, only used in special cases such as vector network analysis. Normally, you should prefer to store different geometry elements in different layers.)

It is possible to store several 'layers' in one vector dataset. For example, fields, forests and lakes can be stored in one vector. An adjacent forest and lake can share the same boundary, but they have separate attribute tables. It is

also possible to attach attributes to boundaries. An example might be the case where the boundary between a lake and a forest is a road, so it can have a different attribute table.

The 'layer' of the feature is defined by the 'layer' inside GRASS. 'Layer' is the number which defines if there is more than one layer inside the dataset (e.g., if the geometry is forest or lake). For now, it can be only a number. In the future, GRASS will also support names as fields in the user interface.

Attributes can be stored inside the GRASS LOCATION as dBase or SQLite3 or in external database tables, for example, PostgreSQL, MySQL, Oracle, etc.

Gli attributi contenuti nelle tabelle del database sono collegati alla geometria per il tramite di un valore 'category'. 'Category' (key, ID) è un valore intero collegato alle primitive geometriche ed è usato come collegamento ad una colonna chiave nella tabella del database.

Suggerimento: Conoscere il modello dati vettoriale di GRASS

The best way to learn the GRASS vector model and its capabilities is to download one of the many GRASS tutorials where the vector model is described more deeply. See <http://grass.osgeo.org/documentation/manuals/> for more information, books and tutorials in several languages.

16.6 Creare un nuovo layer vettoriale GRASS

To create a new GRASS vector layer with the GRASS plugin, click the  Create new GRASS vector toolbar icon. Enter a name in the text box, and you can start digitizing point, line or polygon geometries following the procedure described in section *Digitalizzare e modificare layer vettoriali GRASS*.

In GRASS, it is possible to organize all sorts of geometry types (point, line and area) in one layer, because GRASS uses a topological vector model, so you don't need to select the geometry type when creating a new GRASS vector. This is different from shapefile creation with QGIS, because shapefiles use the Simple Feature vector model (see section *Creating new Vector layers*).

Suggerimento: Creare una tabella attributi per un nuovo layer vettoriale GRASS

Se si desidera assegnare attributi alla geometria digitalizzata, accertarsi di definire lo schema della tabella prima di iniziare a digitalizzare (figura [figure_grass_digitizing_5](#)).

16.7 Digitalizzare e modificare layer vettoriali GRASS

Gli strumenti di digitalizzazione per i layer vettoriali GRASS sono accessibili con l'icona  Modifica vettore GRASS nella barra degli strumenti GRASS. Assicurarsi di caricare un vettoriale GRASS e che esso sia selezionato nella legenda prima di attivare lo strumento di digitalizzazione. La figura [figure_grass_digitizing_2](#) mostra la finestra di dialogo degli strumenti di digitalizzazione GRASS che viene mostrata quando si clicca sullo strumento di modifica. Gli strumenti e le impostazioni di questa barra saranno discussi nelle sezioni seguenti.

Suggerimento: Digitalizzare poligoni in GRASS

If you want to create a polygon in GRASS, you first digitize the boundary of the polygon, setting the mode to 'No category'. Then you add a centroid (label point) into the closed boundary, setting the mode to 'Next not used'. The reason for this is that a topological vector model links the attribute information of a polygon always to the centroid and not to the boundary.

Barra degli strumenti di digitalizzazione

In [figure_grass_digitizing_1](#), you see the GRASS digitizing toolbar icons provided by the GRASS plugin. Table [table_grass_digitizing_1](#) explains the available functionalities.



Figura 16.3: GRASS Digitizing Toolbar

Icona	Strumento	Azione
	Nuovo punto	Digitalizza un nuovo punto
	Nuova linea	Digitalizza una nuova linea
	Nuovo contorno	Digitalizza nuovo contorno (annullare selezionando un altro strumento)
	Nuovo centroide	Digitalizza un nuovo centroide (imposta l'etichetta per un'area esistente)
	Sposta vertice	Sposta un vertice di una linea o contorno esistente in una nuova posizione
	Aggiungi vertice	Aggiunge un vertice ad una linea o contorno esistente
	Elimina vertice	ancella vertici da linee e contorni esistenti (confermare l'eliminazione del vertice selezionato cliccando una seconda volta)
	Sposta elemento	Sposta il contorno, la linea, il punto o il centroide selezionato in una nuova posizione
	Dividi linea	Split an existing line into two parts
	Elimina elemento	Elimina un contorno, una linea, un punto o un centroide esistente (confermare cliccando una seconda volta)
	Modifica attributi	Modifica gli attributi dell'elemento selezionato (si noti che ad un elemento possono essere associati più attributi, si veda sopra)
	Chiudi	Chiude la sessione e salva lo stato attuale (ricostruisce la topologia)

Tabella Strumenti per la digitalizzazione in GRASS

Scheda Categoria

La scheda *Category* consente di definire il modo in cui i valori della categoria verranno assegnati al nuovo elemento geometrico.

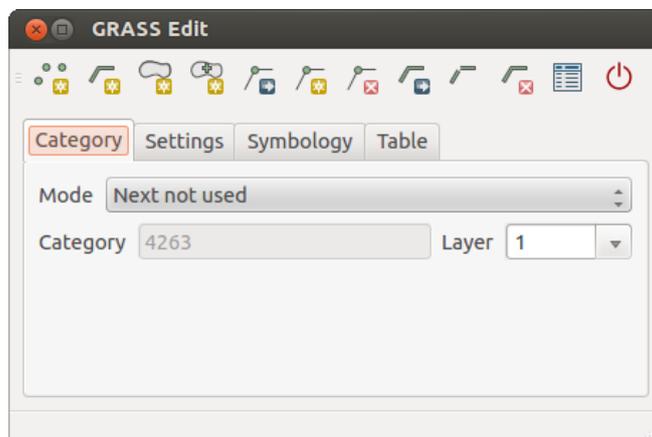


Figura 16.4: GRASS Digitizing Category Tab

- **Mode:** The category value that will be applied to new geometry elements.

- Next not used - Apply next not yet used category value to geometry element.
- Manual entry - Manually define the category value for the geometry element in the 'Category' entry field.
- No category - Do not apply a category value to the geometry element. This is used, for instance, for area boundaries, because the category values are connected via the centroid.
- **Category** - The number (ID) that is attached to each digitized geometry element. It is used to connect each geometry element with its attributes.
- **Field (layer)** - Each geometry element can be connected with several attribute tables using different GRASS geometry layers. The default layer number is 1.

Suggerimento: Creating an additional GRASS 'layer' with lqgl

If you would like to add more layers to your dataset, just add a new number in the 'Field (layer)' entry box and press return. In the Table tab, you can create your new table connected to your new layer.

Scheda Preferenze

La scheda *Preferenze* consente di impostare la tolleranza per l'aggancio automatico tra elementi (snapping) in pixels dello schermo. La soglia definisce a quale distanza massima nuovi punti o linee sono agganciati ad altri nodi esistenti. Ciò aiuta ad evitare interruzioni o incroci tra contorni. Il valore preimpostato è 10 pixels.

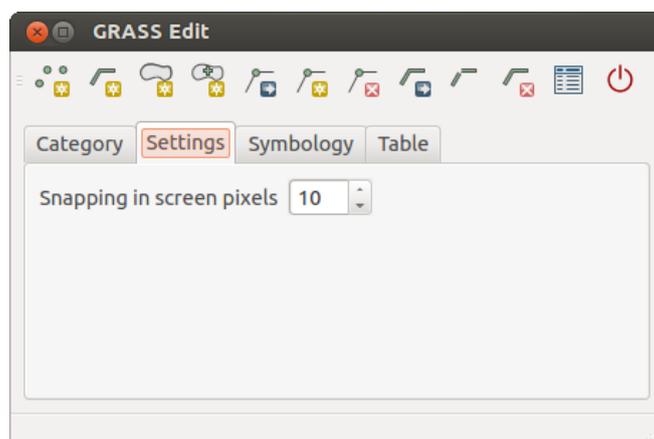


Figura 16.5: GRASS Digitizing Settings Tab

Scheda Simbologia

The *Symbology* tab allows you to view and set symbology and color settings for various geometry types and their topological status (e.g., closed / opened boundary).

Scheda Tabella

The *Table* tab provides information about the database table for a given 'layer'. Here, you can add new columns to an existing attribute table, or create a new database table for a new GRASS vector layer (see section [Creare un nuovo layer vettoriale GRASS](#)).

Suggerimento: Permessi di modifica in GRASS

You must be the owner of the GRASS MAPSET you want to edit. It is impossible to edit data layers in a MAPSET that is not yours, even if you have write permission.

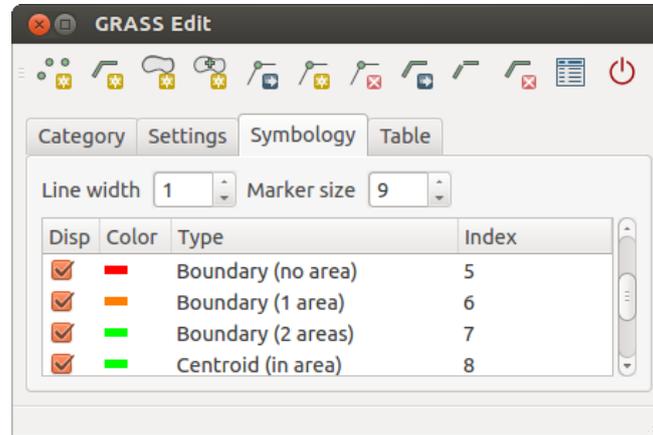


Figura 16.6: GRASS Digitizing Symbology Tab

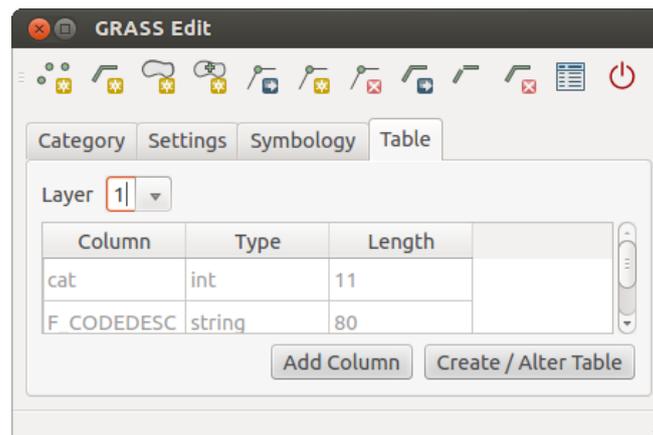


Figura 16.7: GRASS Digitizing Table Tab

16.8 Lo strumento Regione di GRASS

The region definition (setting a spatial working window) in GRASS is important for working with raster layers. Vector analysis is by default not limited to any defined region definitions. But all newly created rasters will have the spatial extension and resolution of the currently defined GRASS region, regardless of their original extension and resolution. The current GRASS region is stored in the `$LOCATION/$MAPSET/WIND` file, and it defines north, south, east and west bounds, number of columns and rows, horizontal and vertical spatial resolution.

It is possible to switch on and off the visualization of the GRASS region in the QGIS canvas using the  Display current GRASS region button.

With the  Edit current GRASS region icon, you can open a dialog to change the current region and the symbology of the GRASS region rectangle in the QGIS canvas. Type in the new region bounds and resolution, and click [OK]. The dialog also allows you to select a new region interactively with your mouse on the QGIS canvas. Therefore, click with the left mouse button in the QGIS canvas, open a rectangle, close it using the left mouse button again and click [OK].

The GRASS module `g.region` provides a lot more parameters to define an appropriate region extent and resolution for your raster analysis. You can use these parameters with the GRASS Toolbox, described in section [The GRASS Toolbox](#).

16.9 The GRASS Toolbox

The  Open GRASS Tools box provides GRASS module functionalities to work with data inside a selected GRASS LOCATION and MAPSET. To use the GRASS Toolbox you need to open a LOCATION and MAPSET that you have write permission for (usually granted, if you created the MAPSET). This is necessary, because new raster or vector layers created during analysis need to be written to the currently selected LOCATION and MAPSET.

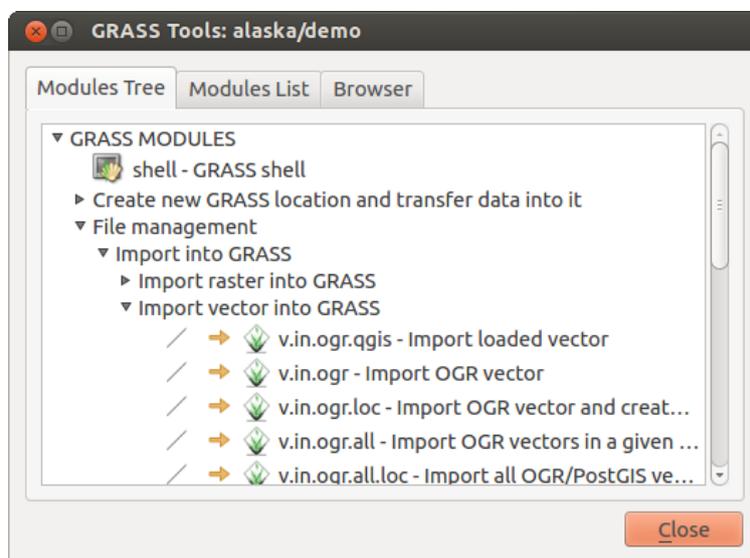


Figura 16.8: GRASS Toolbox and Module Tree 

16.9.1 Lavorare con i moduli GRASS

The GRASS shell inside the GRASS Toolbox provides access to almost all (more than 300) GRASS modules in a command line interface. To offer a more user-friendly working environment, about 200 of the available GRASS modules and functionalities are also provided by graphical dialogs within the GRASS plugin Toolbox.

A complete list of GRASS modules available in the graphical Toolbox in QGIS version 2.6 is available in the GRASS wiki at http://grass.osgeo.org/wiki/GRASS-QGIS_relevant_module_list.

It is also possible to customize the GRASS Toolbox content. This procedure is described in section *Personalizzare gli strumenti GRASS*.

As shown in [figure_grass_toolbox_1](#), you can look for the appropriate GRASS module using the thematically grouped *Modules Tree* or the searchable *Modules List* tab.

By clicking on a graphical module icon, a new tab will be added to the Toolbox dialog, providing three new sub-tabs: *Options*, *Output* and *Manual*.

Opzioni

The *Options* tab provides a simplified module dialog where you can usually select a raster or vector layer visualized in the QGIS canvas and enter further module-specific parameters to run the module.

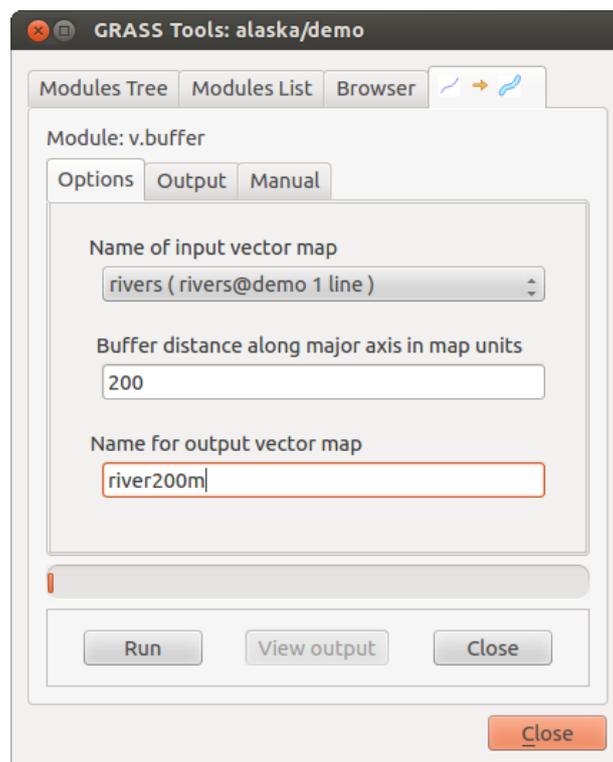


Figura 16.9: GRASS Toolbox Module Options 

The provided module parameters are often not complete to keep the dialog clear. If you want to use further module parameters and flags, you need to start the GRASS shell and run the module in the command line.

A new feature since QGIS 1.8 is the support for a *Show Advanced Options* button below the simplified module dialog in the *Options* tab. At the moment, it is only added to the module `v.in.ascii` as an example of use, but it will probably be part of more or all modules in the GRASS Toolbox in future versions of QGIS. This allows you to use the complete GRASS module options without the need to switch to the GRASS shell.

Output

The *Output* tab provides information about the output status of the module. When you click the **[Run]** button, the module switches to the *Output* tab and you see information about the analysis process. If all works well, you will finally see a `Successfully finished` message.

Manuale

The *Manual* tab shows the HTML help page of the GRASS module. You can use it to check further module parameters and flags or to get a deeper knowledge about the purpose of the module. At the end of each module

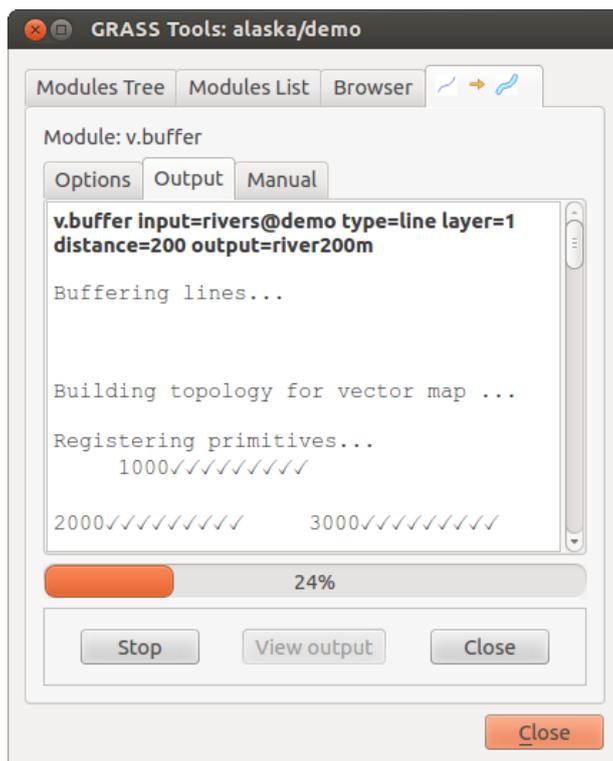


Figura 16.10: GRASS Toolbox Module Output 

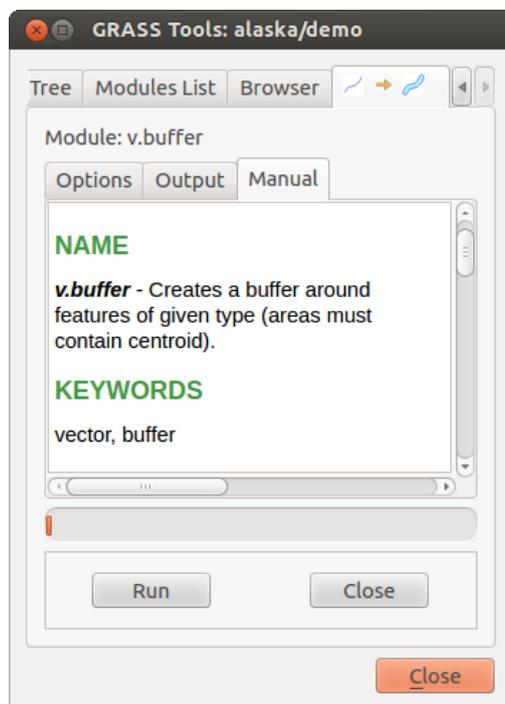


Figura 16.11: GRASS Toolbox Module Manual 

manual page, you see further links to the `Main Help index`, the `Thematic index` and the `Full index`. These links provide the same information as the module `g.manual`.

Suggerimento: Mostrare i risultati immediatamente

Se si desidera visualizzare il risultato di un'analisi immediatamente nella vista mappa, è possibile cliccare sul pulsante `Visualizza Output` nella porzione inferiore della scheda.

16.9.2 Esempi di utilizzo di moduli GRASS

Gli esempi che seguono mostrano le potenzialità di alcuni moduli GRASS.

Creare curve di livello

The first example creates a vector contour map from an elevation raster (DEM). Here, it is assumed that you have the Alaska `LOCATION` set up as explained in section *Importare dati nelle LOCATION GRASS*.

- First, open the location by clicking the  `Open mapset` button and choosing the Alaska location.
- Caricare il DEM `gtopo30` cliccando su  `Aggiungi raster GRASS` e selezionare `gtopo30` dal mapset demo.
- Aprire gli strumenti GRASS con  `Apri strumenti GRASS`.
- In the list of tool categories, double-click `Raster` → `Surface Management` → `Generate vector contour lines`.
- Now a single click on the tool `r.contour` will open the tool dialog as explained above (see *Lavorare con i moduli GRASS*). The `gtopo30` raster should appear as the *Name of input raster*.
- Inserire in *Incrementa fra le isoipse* il valore 100. (per creare curve di livello ad intervalli di 100 metri)
- Inserire in *Nome del vettoriale in output* il nome `ctour_100`.
- Click **[Run]** to start the process. Wait for several moments until the message `Successfully finished` appears in the output window. Then click **[View Output]** and **[Close]**.

Dal momento che la regione è piuttosto estesa, il comando richiede del tempo. Una volta terminata l'operazione è possibile modificare le proprietà del nuovo layer vettoriale come descritto in *Proprietà dei vettori*.

Next, zoom in to a small, mountainous area in the center of Alaska. Zooming in close, you will notice that the contours have sharp corners. GRASS offers the `v.generalize` tool to slightly alter vector maps while keeping their overall shape. The tool uses several different algorithms with different purposes. Some of the algorithms (i.e., Douglas Peuker and Vertex Reduction) simplify the line by removing some of the vertices. The resulting vector will load faster. This process is useful when you have a highly detailed vector, but you are creating a very small-scale map, so the detail is unnecessary.

Suggerimento: Semplifica geometrie

Note that the QGIS fTools plugin has a *Simplify geometries* → tool that works just like the GRASS `v.generalize` Douglas-Peuker algorithm.

However, the purpose of this example is different. The contour lines created by `r.contour` have sharp angles that should be smoothed. Among the `v.generalize` algorithms, there is Chaiken's, which does just that (also Hermite splines). Be aware that these algorithms can **add** additional vertices to the vector, causing it to load even more slowly.

- Open the GRASS Toolbox and double-click the categories `Vector` → `Develop map` → `Generalization`, then click on the `v.generalize` module to open its options window.
- Controllare che `'ctour_100'` appaia come *Nome della mappa vettoriale in input*.

- From the list of algorithms, choose Chaiken's. Leave all other options at their default, and scroll down to the last row to enter in the field *Name for output vector map* 'ctour_100_smooth', and click **[Run]**.
- The process takes several moments. Once *Successfully finished* appears in the output windows, click **[View output]** and then **[Close]**.
- È possibile modificare il colore del layer vettoriale in modo da renderlo ben visibile sul raster di sfondo. Si potrà notare come le curve di livello ora appaiano meno spigolose.

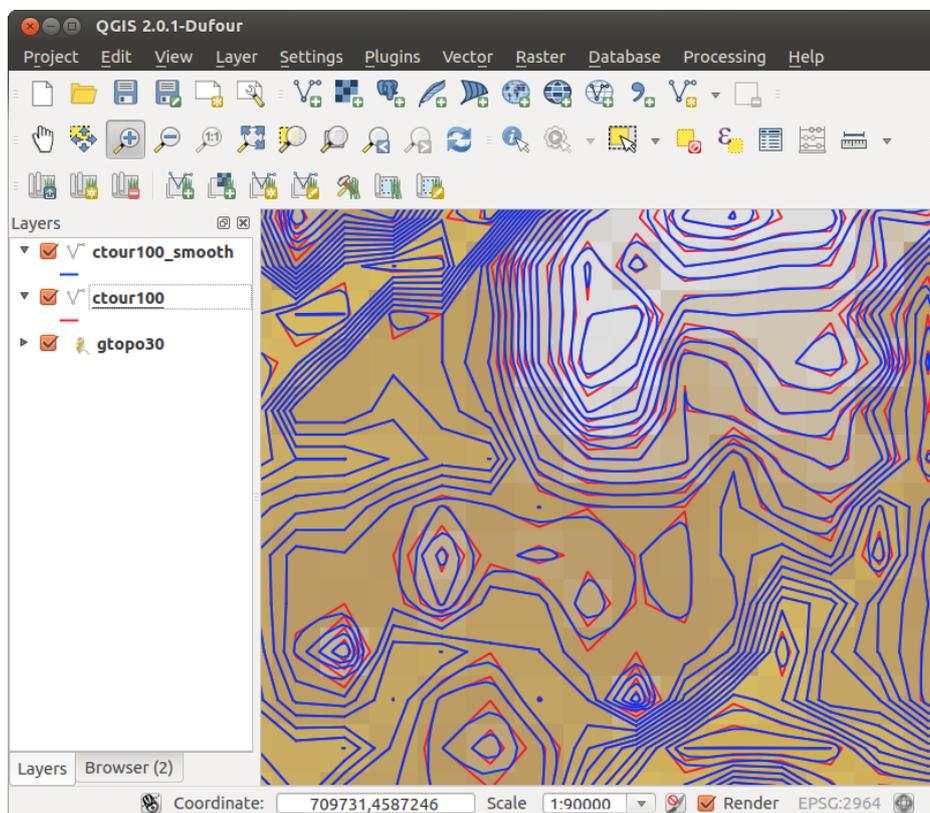


Figura 16.12: GRASS module v.generalize to smooth a vector map 🐧

Suggerimento: Altri usi di r.contour

The procedure described above can be used in other equivalent situations. If you have a raster map of precipitation data, for example, then the same method will be used to create a vector map of isohyetal (constant rainfall) lines.

Creating a Hillshade 3-D effect

Several methods are used to display elevation layers and give a 3-D effect to maps. The use of contour lines, as shown above, is one popular method often chosen to produce topographic maps. Another way to display a 3-D effect is by hillshading. The hillshade effect is created from a DEM (elevation) raster by first calculating the slope and aspect of each cell, then simulating the sun's position in the sky and giving a reflectance value to each cell. Thus, you get sun-facing slopes lighted; the slopes facing away from the sun (in shadow) are darkened.

- Begin this example by loading the `gtopo30` elevation raster. Start the GRASS Toolbox, and under the Raster category, double-click to open *Spatial analysis* → *Terrain analysis*.
- Cliccare **r.shaded.relief** per aprire il modulo.
- Impostare l'*azimuth* da 270 a 315.
- Inserire `gtopo30_shade` per il nuovo raster delle ombreggiature e cliccare su **[Esegui]**.
- Quando il processo sarà completato, aggiungere il raster ombreggiatura alla vista mappa.

- To view both the hillshading and the colors of the `gtopo30` together, move the hillshade map below the `gtopo30` map in the table of contents, then open the *Properties* window of `gtopo30`, switch to the *Transparency* tab and set its transparency level to about 25%.

Si dovrebbe vedere `gtopo30` sopra la mappa di ombreggiatura in scala di grigi. Per riuscire a visualizzare appieno gli effetti dell'ombreggiatura, deselezionare `gtopo30_shade`.

Usare la shell di GRASS

The GRASS plugin in QGIS is designed for users who are new to GRASS and not familiar with all the modules and options. As such, some modules in the Toolbox do not show all the options available, and some modules do not appear at all. The GRASS shell (or console) gives the user access to those additional GRASS modules that do not appear in the Toolbox tree, and also to some additional options to the modules that are in the Toolbox with the simplest default parameters. This example demonstrates the use of an additional option in the `r.shaded.relief` module that was shown above.

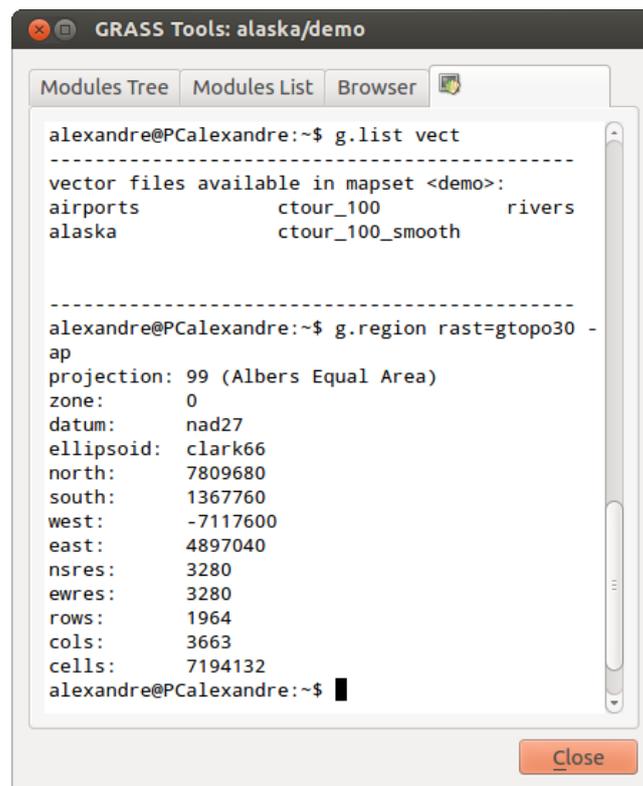


Figura 16.13: The GRASS shell, `r.shaded.relief` module 🐧

The module `r.shaded.relief` can take a parameter `zmult`, which multiplies the elevation values relative to the X-Y coordinate units so that the hillshade effect is even more pronounced.

- Load the `gtopo30` elevation raster as above, then start the GRASS Toolbox and click on the GRASS shell. In the shell window, type the command `r.shaded.relief map=gtopo30 shade=gtopo30_shade2 azimuth=315 zmult=3` and press **[Enter]**.
- After the process finishes, shift to the *Browse* tab and double-click on the new `gtopo30_shade2` raster to display it in QGIS.
- As explained above, move the shaded relief raster below the `gtopo30` raster in the table of contents, then check the transparency of the colored `gtopo30` layer. You should see that the 3-D effect stands out more strongly compared with the first shaded relief map.

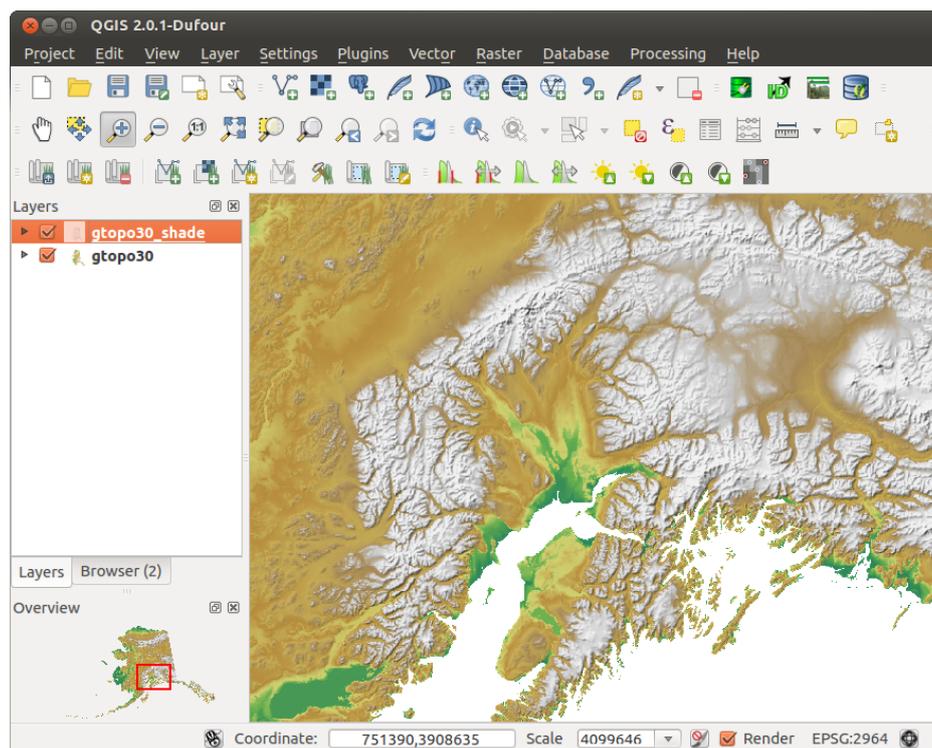


Figura 16.14: Displaying shaded relief created with the GRASS module `r.shaded.relief` 

Statistiche raster in una mappa vettoriale

Il prossimo esempio tratta di un modulo GRASS che può aggregare dati raster ed aggiungere colonne di statistiche per ogni poligono di una mappa vettoriale.

- Importare in GRASS lo shapefile `trees` nella cartella `shapefiles` *Importare dati nelle LOCATION GRASS*.
- Now an intermediate step is required: centroids must be added to the imported trees map to make it a complete GRASS area vector (including both boundaries and centroids).
- From the Toolbox, choose *Vector* → *Manage features*, and open the module `v.centroids`.
- Inserire come *Nome del vettoriale in output* `forest_areas` e lanciare il modulo.
- Now load the `forest_areas` vector and display the types of forests - deciduous, evergreen, mixed - in different colors: In the layer *Properties* window, *Symbology* tab, choose from *Legend type*  'Unique value' and set the *Classification field* to 'VEGDESC'. (Refer to the explanation of the symbology tab in *Menu Stile* of the vector section.)
- Next, reopen the GRASS Toolbox and open *Vector* → *Vector update* by other maps.
- Click on the `v.rast.stats` module. Enter `gtopo30` and `forest_areas`.
- Only one additional parameter is needed: Enter *column prefix* `elev`, and click **[Run]**. This is a computationally heavy operation, which will run for a long time (probably up to two hours).
- Finally, open the `forest_areas` attribute table, and verify that several new columns have been added, including `elev_min`, `elev_max`, `elev_mean`, etc., for each forest polygon.

16.9.3 Lavorare con il browser delle LOCATION GRASS

Another useful feature inside the GRASS Toolbox is the GRASS LOCATION browser. In [figure_grass_module_7](#), you can see the current working LOCATION with its MAPSETS.

In the left browser windows, you can browse through all MAPSETs inside the current LOCATION. The right browser window shows some meta-information for selected raster or vector layers (e.g., resolution, bounding box, data source, connected attribute table for vector data, and a command history).

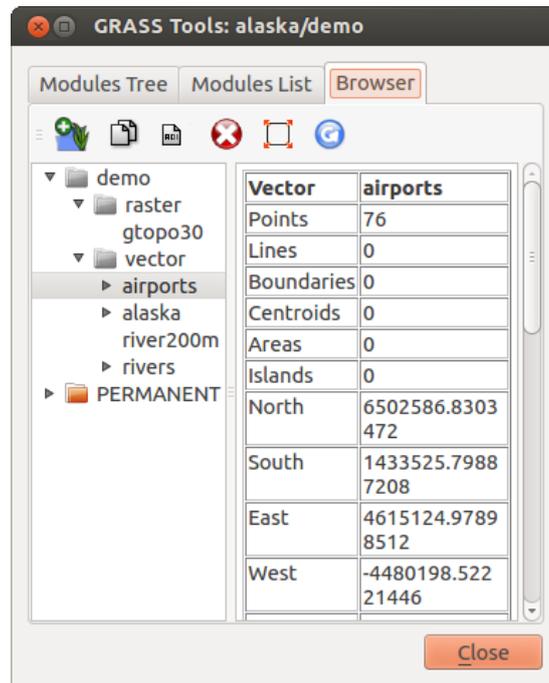


Figura 16.15: GRASS LOCATION browser 

The toolbar inside the *Browser* tab offers the following tools to manage the selected LOCATION:

-  *Aggiungi la mappa selezionata all'area di lavoro*
-  *Copia la mappa selezionata*
-  *Rinomina la mappa selezionata*
-  *Elimina la mappa selezionata*
-  *Imposta la regione corrente con la mappa selezionata*
-  *Aggiorna*

Gli strumenti  *Rinomina la mappa selezionata* e  *Elimina la mappa selezionata* funzionano solo su mappe contenute nel MAPSET attivo. Tutti gli altri strumenti funzionano anche con layer raster e vettoriali di altri MAPSET.

16.9.4 Personalizzare gli strumenti GRASS

Nearly all GRASS modules can be added to the GRASS Toolbox. An XML interface is provided to parse the pretty simple XML files that configure the modules' appearance and parameters inside the Toolbox.

Un esempio di file XML che genera il modulo `v.buffer` (`v.buffer.qgm`) ha il seguente aspetto:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE qgisgrassmodule SYSTEM "http://mrcc.com/qgisgrassmodule.dtd">

<qgisgrassmodule label="Vector buffer" module="v.buffer">
```

```
<option key="input" typeoption="type" layeroption="layer" />
<option key="buffer"/>
<option key="output" />
</qgisgrassmodule>
```

The parser reads this definition and creates a new tab inside the Toolbox when you select the module. A more detailed description for adding new modules, changing a module's group, etc., can be found on the QGIS wiki at http://hub.qgis.org/projects/quantum-gis/wiki/Adding_New_Tools_to_the_GRASS_Toolbox.

QGIS L'ambiente Processing

17.1 Introduzione

Questo capitolo introduce l'ambiente QGIS Processing, un ambiente di elaborazione di dati geografici grazie al quale potrai usare algoritmi nativi di QGIS e algoritmi di terze parti. In questo modo le attività di analisi spaziale saranno molto più produttive e facili da realizzare.

Nella sezione seguente esamineremo come usare gli elementi grafici di questo ambiente e come ottenere il massimo da ciascuno di essi.

Ci sono quattro elementi base nell'interfaccia grafica dell'ambiente, che vengono usati per lanciare algoritmi con differenti finalità. La scelta tra uno strumento e un altro dipende dal tipo di analisi che vuoi effettuare e dalle caratteristiche particolari dei singoli utenti e progetti. Tutti questi elementi (esclusa l'interfaccia dei processi in serie, che come vedremo viene caricata da strumenti) possono essere selezionati dal menu *Processing* (vedrai più di quattro voci. Le altre non sono usate per eseguire algoritmi e saranno spiegate nel seguito del capitolo).

- **Strumenti.** È l'elemento principale dell'interfaccia grafica. Lo potrai usare per eseguire un singolo algoritmo o per lanciare un processo in serie basato sullo stesso algoritmo.

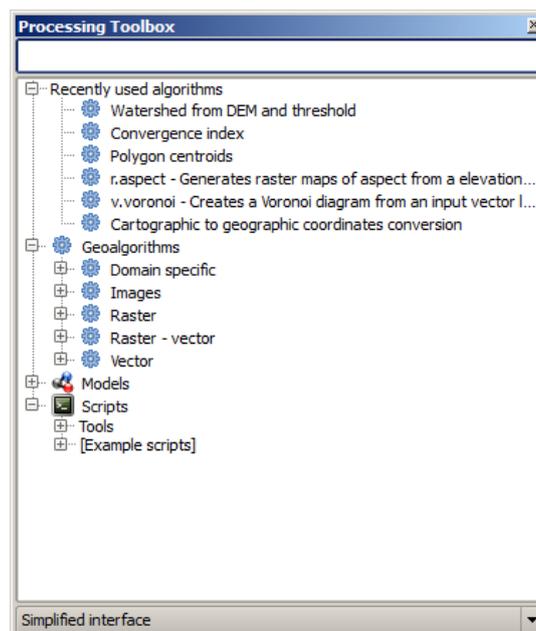


Figura 17.1: Strumenti di Processing 

- Modellatore grafico. Puoi combinare molti algoritmi usando il modellatore grafico. In questo modo potrai definire un flusso di lavoro anche molto complesso, creando un singolo processo che in realtà coinvolge più sotto-processi.

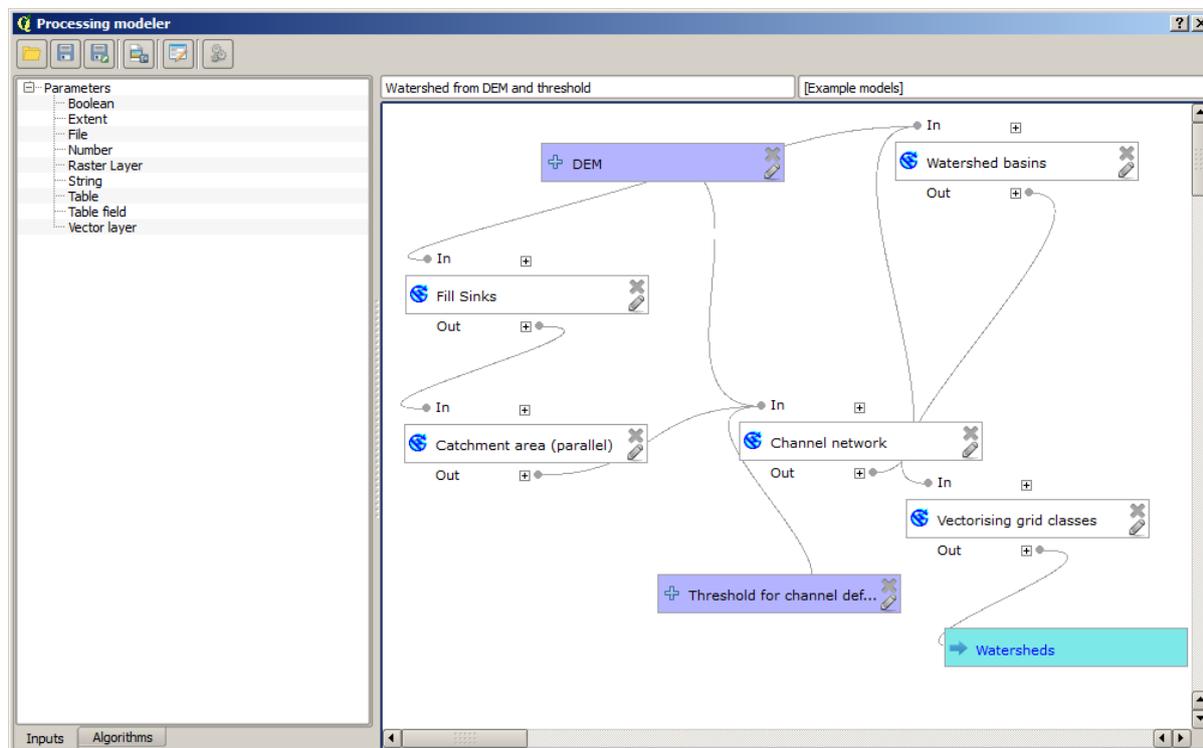


Figura 17.2: Modellatore Grafico di Processing

- Storico e log. Tutte le azioni eseguite usando uno qualsiasi degli elementi sono raccolte in un file di cronologia e le potrai facilmente richiamare in un secondo momento.
- Intercaccia del processo in serie. Questa interfaccia ti permette di eseguire processi in serie e di automatizzare l'esecuzione di un particolare algoritmo su più insiemi di dati.

Nelle sezioni seguenti rivedremo in dettaglio ciascuno di questi elementi.

17.2 Strumenti

La finestra *Strumenti* è l'elemento principale della interfaccia grafica di Processing, ed è l'elemento che probabilmente userai quotidianamente. La finestra ti mostra l'elenco degli algoritmi disponibili raggruppati in differenti sezioni ed è il punto di partenza per lanciare gli algoritmi sia come singoli processi che come processi in serie, ovvero processi che implicano l'esecuzione dello stesso algoritmo su diversi insiemi di dati.

Strumenti contiene tutti gli algoritmi disponibili, divisi in gruppi predefiniti. Tutti questi gruppi si trovano nello stesso insieme chiamato *Geoalgorithms*.

Inoltre, puoi trovare altre due voci, chiamate *Modelli* e *Scripts*. Queste comprendono gli algoritmi creati dall'utente e permettono di definire i propri flussi di lavoro e le attività di elaborazione. Gli dedicheremo una sezione completa in seguito.

Nella parte superiore degli Strumenti, potrai trovare spazio in cui inserire del testo. Per facilitare la ricerca di un algoritmo, puoi inserire qui una parola o una frase specifica: vedrai subito che il numero degli algoritmi si ridurrà al numero di quelli che contengono il testo che hai inserito.

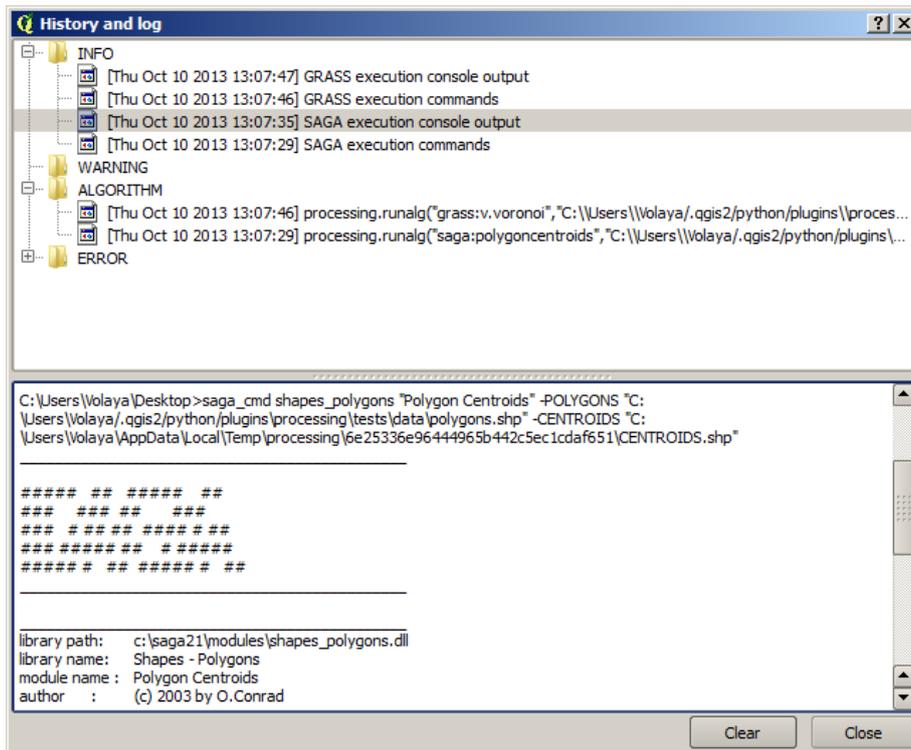


Figura 17.3: Storico e log di Processing

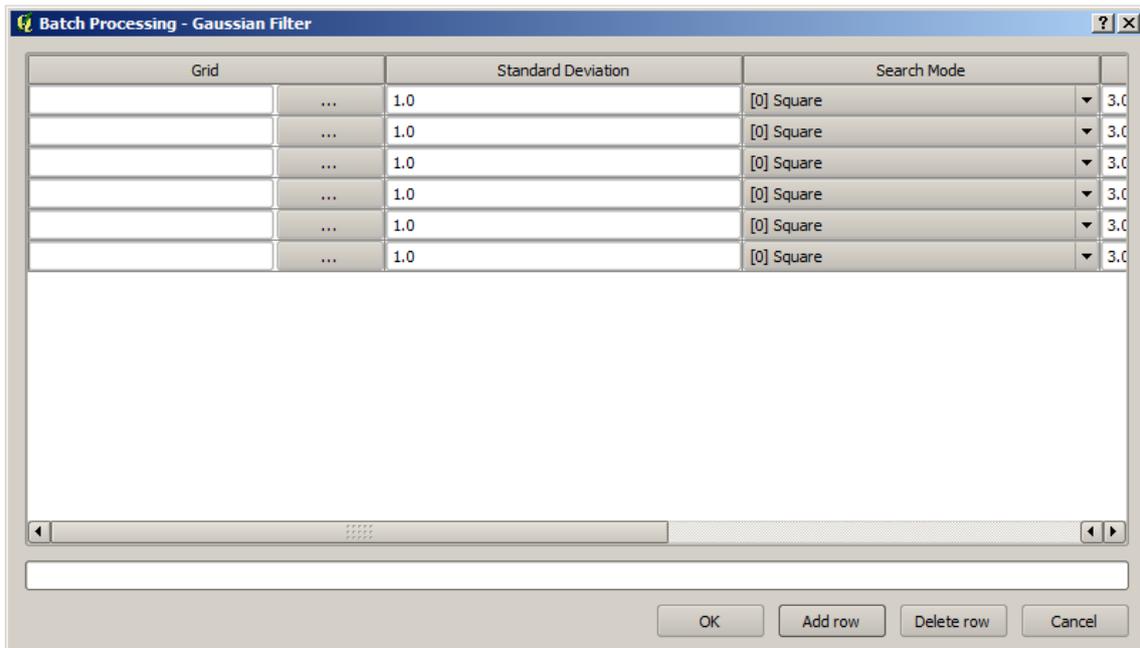


Figura 17.4: Interfaccia del processo in serie

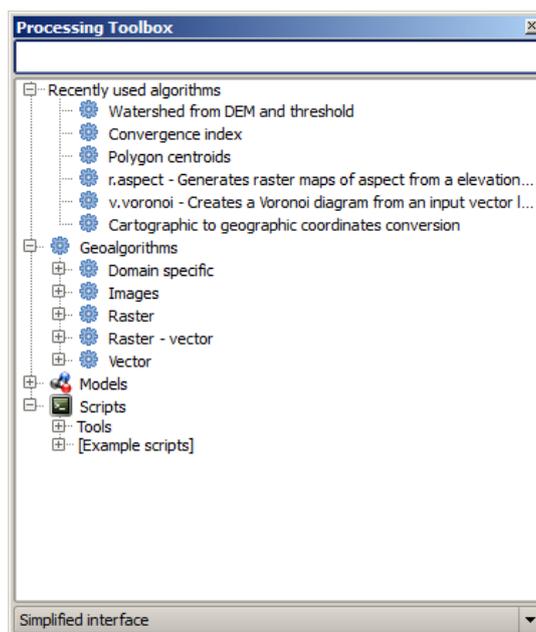


Figura 17.5: Strumenti di Processing

Nella parte inferiore puoi trovare un menu a tendina che ti permette di scegliere tra l'interfaccia semplificata della lista di algoritmi (spiegata sopra) e l'interfaccia avanzata. Se scegli la visualizzazione avanzata, lo strumento avrà questo aspetto:

Nell'interfaccia avanzata, ogni gruppo rappresenta un 'fornitore di algoritmi', ovvero identifica un insieme di algoritmi provenienti dallo stesso fornitore, per esempio, un programma esterno, con capacità di elaborazione di dati geografici. Alcuni di questi gruppi rappresentano algoritmi provenienti da una di queste applicazioni (come SAGA, GRASS o R), mentre altri gruppi contengono algoritmi associati direttamente a Processing, visto che non serve nessun software aggiuntivo.

Questa modalità di visualizzazione è consigliata agli utenti che hanno una certa conoscenza dei programmi che supportano tali algoritmi, dal momento che questi saranno indicati con i loro nomi originali e gruppi di appartenenza.

Inoltre, altri algoritmi sono disponibili solo nell'interfaccia avanzata, come ad esempio strumenti LiDAR o script basati sul software di analisi statistica R. I plugin QGIS indipendenti che aggiungono nuovi algoritmi alla finestra degli strumenti saranno visibili solo nell'interfaccia avanzata.

In particolare, la visualizzazione semplificata contiene algoritmi supportati dai seguenti fornitori:

- GRASS
- SAGA
- OTB
- Algoritmi QGIS nativi

Nel caso particolare di esecuzione di QGIS su Windows, questi algoritmi sono pienamente funzionanti con la nuova installazione di QGIS e li puoi eseguire senza che sia necessaria nessuna ulteriore installazione. Non devi conoscere i programmi per eseguire questi algoritmi; in questo modo è tutto più semplice da usare per i nuovi utenti.

Se vuoi usare un algoritmo non supportato da nessuno dei fornitori sopra elencati, seleziona nella parte in basso dello strumento l'interfaccia avanzata.

Per eseguire un algoritmo è sufficiente un doppio click con il mouse sul suo nome.

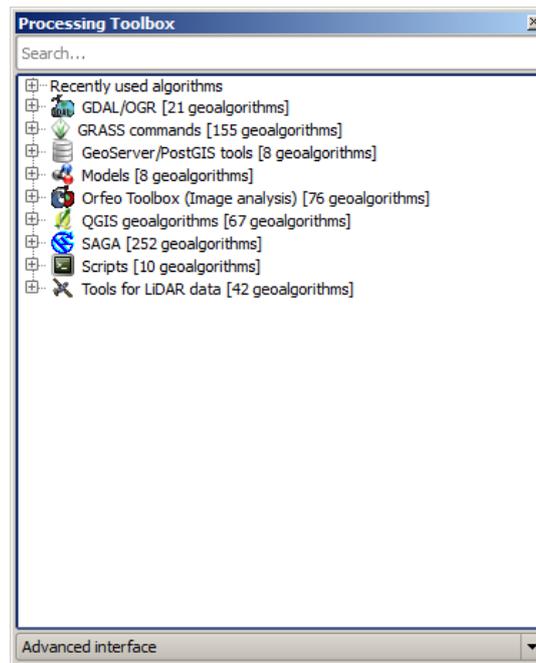


Figura 17.6: Strumenti di Processing (visualizzazione avanzata) 

17.2.1 La finestra di configurazione di un algoritmo

Una volta fatto doppio click sul nome dell'algoritmo che vuoi eseguire, apparirà una finestra di dialogo come quella mostrata qui sotto (in questo caso si tratta della finestra di dialogo corrispondente all'algoritmo 'indice di convergenza' di SAGA).

In questa finestra dovrai inserire tutti gli input necessari per l'esecuzione dell'algoritmo. Vedrai una tabella dove devi definire i valori di input ed i vari parametri di configurazione. Ogni algoritmo avrà la sua finestra di dialogo con voci sempre differenti. Nella parte sinistra vedrai il nome del parametro, nella parte destra invece vedrai il valore corrispondente.

Anche se il numero e tipo dei parametri dipende dal tipo di algoritmo, la struttura di base è simile per tutti. I parametri della tabella possono essere uno dei seguenti tipi.

- Un raster, da selezionare da una lista di tutti quelli disponibili (attualmente caricati) in QGIS. Il selettore contiene anche un pulsante nella parte destra, che ti permette di selezionare i raster non caricati in QGIS.
- Un vettore, da selezionare da una lista di tutti quelli caricati in QGIS. Puoi selezionare i vettori non caricati in QGIS, come nel caso dei raster, solamente se l'algoritmo non richiede un campo che dipende dalla tabella degli attributi di un vettore caricato in QGIS. In questo caso, puoi selezionare solo i vettori caricati; infatti, QGIS riesce a recuperare i campi della tabella degli attributi solo in questo modo.

Vedrete un pulsante per ciascun selezionatore di vettori come mostrato nella figura.

Se l'algoritmo ne contiene molti, potrai selezionarne anche solamente uno. Se il pulsante corrispondente a un vettore in input è attivo, allora l'algoritmo verrà eseguito iterativamente su tutte le sue geometrie, invece di una sola volta per tutto il vettore. Il numero di output dipende da quante volte eseguirai l'algoritmo. Questo ti permette di automatizzare il processo quando tutte le geometrie in un vettore devono essere elaborate separatamente.

- Una tabella, da selezionare da una lista di tutte quelle caricate in QGIS. Le tabelle non-spaziali caricate in QGIS vengono trattate come se fossero un vettore. Attualmente, la lista delle possibili tabelle che puoi vedere durante l'esecuzione di un algoritmo è limitata a tabelle provenienti da file in formati dBase (.dbf) o Comma-Separated Values (.csv).
- Opzione, da scegliere in una lista di possibili opzioni.
- Un valore numerico, da inserire in una casella di testo. Cliccando sul pulsante laterale, comparirà una finestra di dialogo che ti permetterà di inserire un'espressione matematica. In questo modo hai la possibilità

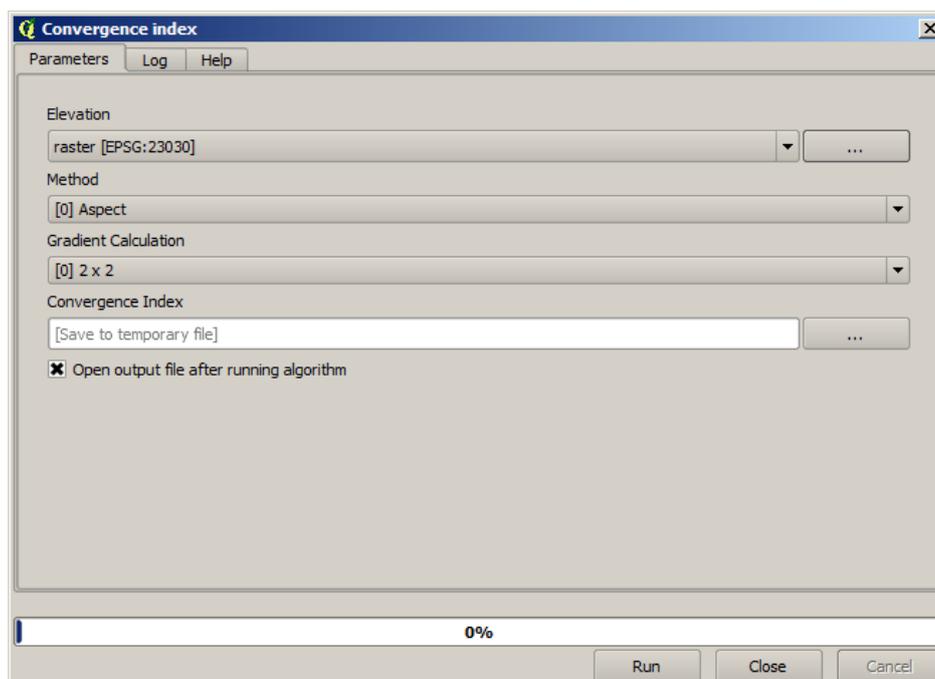


Figura 17.7: La finestra di impostazione dei Parametri



Figura 17.8: Vector iterator button 

di usare una vera e propria calcolatrice. Puoi anche aggiungere all'espressione alcune utili variabili che fanno riferimento ai dati caricati in QGIS, così che potrai selezionare un valore numerico come la risoluzione del layer o la coordinata più a nord di un altro layer.

- Un intervallo, con i valori minimo e massimo da inserire in due corrispondenti caselle di testo.
- Una stringa di testo, da inserire in un'apposita casella.
- Un campo, da scegliere dalla tabella degli attributi di un vettore, o da una singola tabella selezionata con un altro parametro.
- Un sistema di riferimento di coordinate. Puoi inserire direttamente il codice EPSG o selezionarlo dalla finestra di dialogo dei SR che puoi aprire cliccando sul pulsante nella parte destra.
- Regione, da inserire mediante i suoi estremi x_{min} , x_{max} , y_{min} , y_{max} . Premendo il pulsante sulla destra apparirà una finestra che ti permette di scegliere fra: usare la regione attualmente presente nella mappa oppure modificare gli estremi selezionando un rettangolo direttamente con il mouse.

Se scegliete la prima opzione verrà mostrata la seguente finestra di scelta rapida.

Se scegliete la seconda opzione, la finestra dei parametri sparirà, così potrai scegliere l'area tramite il mouse. Una volta selezionato il rettangolo, la finestra di dialogo riapparirà completa dei limiti della regione appena definita.

- Una lista di elementi (siano essi raster, vettori o tabelle), da selezionare dalla lista di quelli caricati in QGIS. Per selezionarli, clicca il pulsante sul lato sinistro della riga corrispondente in modo da aprire una finestra di dialogo come la seguente.
- Una piccola tabella da completare a cura dell'utente. Queste tabelle sono usate per definire, ad esempio, parametri tipo tabelle di riferimento o kernel di convoluzione.

Cliccate sul bottone sul lato destro per vedere la tabella e aggiornare i suoi valori.

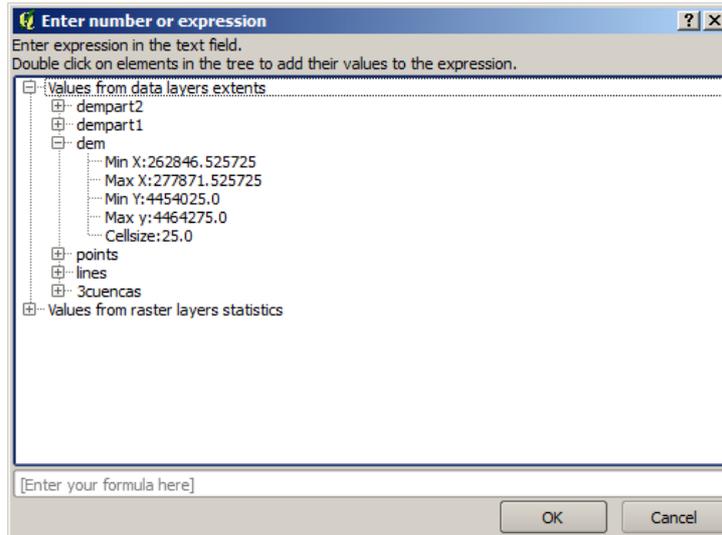


Figura 17.9: Selettore di Numero

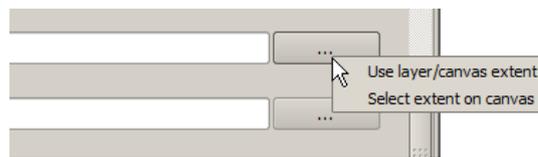


Figura 17.10: Selettore di regione

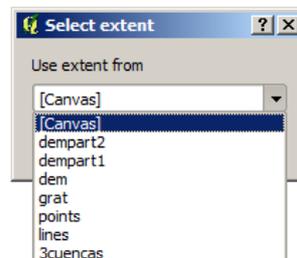


Figura 17.11: Lista di selezione di una regione

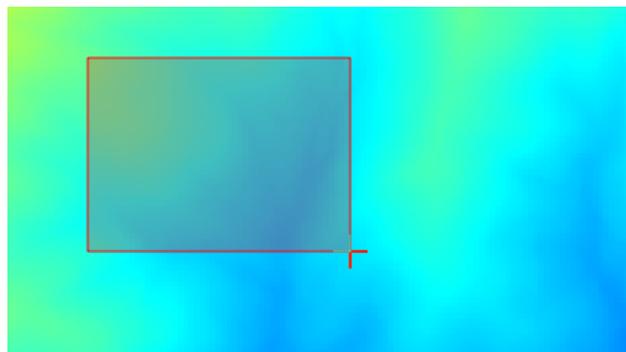


Figura 17.12: Trascinamento del mouse e selezione di una regione

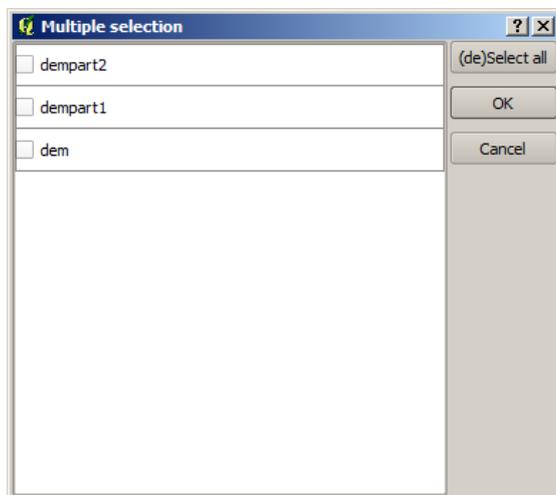


Figura 17.13: Selezioni multiple

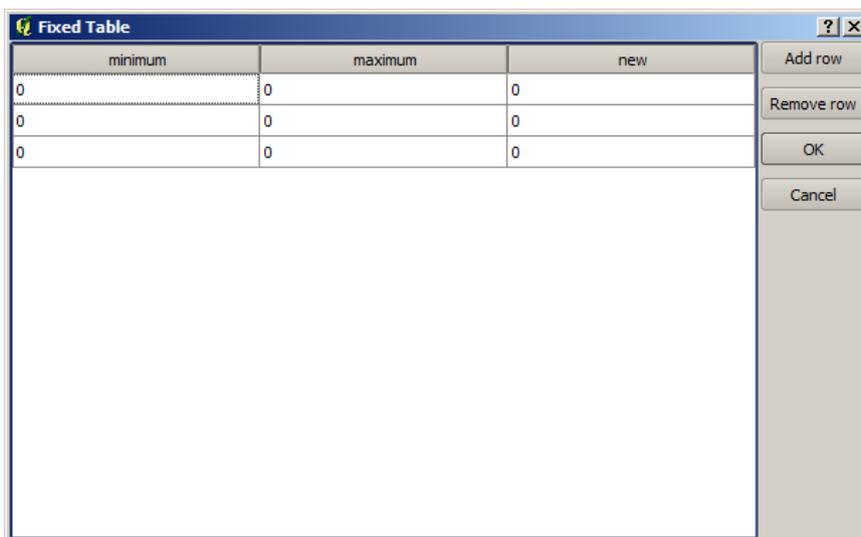


Figura 17.14: Tabella

A seconda dell'algoritmo, potrai modificare il numero delle righe, usando i pulsanti sul lato destro della finestra.

È anche presente una scheda **[Help]** nella finestra di dialogo dell'algoritmo. Se l'aiuto è disponibile, allora avrai maggiori informazioni sull'algoritmo e spiegazioni dettagliate di ciascun parametro da inserire. Purtroppo, la maggior parte degli algoritmi non hanno ancora una buona documentazione, quindi se hai voglia di contribuire al progetto, questo è un ottimo punto di partenza!

Nota sulle proiezioni

Gli algoritmi lanciati da Processing – e anche la maggior parte delle applicazioni esterne attraverso cui sono caricati gli algoritmi – non riproiettano i layer in input perché viene dato per scontato che tutti i layer siano già in un sistema di coordinate comune e pronti per essere analizzati. Ogni volta che si utilizza più di un layer in input per un algoritmo, vettore o raster, spetta a te assicurarti che siano tutti nello stesso sistema di coordinate.

Grazie alla capacità di riproiezione al volo di QGIS, due layer possono sembrare sovrapposti, ma in realtà non hanno lo stesso sistema di riferimento. Sarebbe meglio riproiettare a mano i vari layer e lanciare dopo l'algoritmo. Puoi anche usare un algoritmo di Processing per riproiettare i layer.

In modo predefinito, la finestra di impostazione dei parametri mostra, oltre al nome del layer anche il suo SR in modo da facilitare la selezione di altri layer con lo stesso SR. Se non vuoi questa informazione aggiuntiva, puoi disabilitare questa funzionalità nella finestra Opzioni e Configurazione di Processing, togliendo la spunta alla casella di controllo *Show CRS*.

Se provi a lanciare un algoritmo usando due o più layer con diversi SR, comparirà una finestra di avviso.

Potrai comunque eseguire l'algoritmo, ma sappi che nella maggior parte dei casi ciò produrrà cattivi risultati, come ad esempio layer di uscita inconsistenti, proprio perché questi non sono sovrapposti.

17.2.2 Dati generati dagli algoritmi

I dati generati da un algoritmo possono apparire a una delle seguenti tipologie:

- Raster
- Un vettore
- Tabella
- File HTML (usato per risultati testuali e grafici)

Questi vengono tutti memorizzati su disco, e la tabella dei parametri conterrà una casella di testo per ciascuno di questi output in cui potrai specificare il canale di output per la memorizzazione. Un canale di output contiene le informazioni necessarie per memorizzare l'oggetto risultante da qualche parte. È molto probabile che tu voglia salvare l'output su un file, ma ci sono altri modi di memorizzazione. Per esempio, puoi salvare un vettore in un database o lo puoi anche caricare su un server remoto usando un servizio WMS-T. Anche se queste soluzioni non sono ancora implementate, Processing è in grado di gestirle e presto aggiungeremo tutte queste nuove caratteristiche.

Per selezionare un canale di uscita, basta fare clic sul pulsante a destra della casella di testo. Si aprirà una finestra di impostazione file, dove potrai selezionare il percorso del file da memorizzare. I formati dei file supportati vengono visualizzati nel selettore della finestra, a seconda del tipo di output e di algoritmo.

Il formato di output è definito dall'estensione file. I formati supportati dipendono da quelli supportati dall'algoritmo stesso. Per selezionare un formato, basta selezionare la corrispondente estensione file (o specificarla manualmente se stai inserendo direttamente il percorso file). Se l'estensione del percorso file inserito non corrisponde a nessuno dei formati supportati allora un'estensione predefinita (di solito `.dbf` per le tabelle, `.tif` per i raster e `.shp` per i vettori) verrà aggiunta al percorso file e il formato file corrispondente a quell'estensione verrà usato per memorizzare il layer o la tabella.

Se non inserisci nessun nome file, il risultato verrà salvato in un file temporaneo nel corrispondente formato predefinito e sarà cancellato una volta uscito da QGIS (fai quindi attenzione nel caso in cui memorizzi il tuo progetto con layer temporanei).

Puoi impostare una cartella predefinita in cui memorizzare i dati in output. Vai nella finestra di dialogo delle configurazioni (la puoi aprire da menu *Processing*) e nel gruppo *General* troverai l'opzione *Output folder*. Questa cartella verrà usata come percorso predefinito nel caso tu inserisca un nome del file senza percorso (i.e., `myfile.shp`) durante l'esecuzione dell'algoritmo.

Durante l'esecuzione di un algoritmo che usa un vettore in modo iterativo, il percorso del file inserito è usato come percorso di base per tutti i file generati, i quali sono nominati usando il nome del vettore e aggiungendo poi un numero che rappresenta l'indice di iterazione. L'estensione del file (e il formato) viene usata per tutti i file generati.

Oltre che raster e tabelle, gli algoritmi generano anche grafici e testo in formato HTML. I risultati vengono mostrati al termine dell'esecuzione dell'algoritmo in una nuova finestra di dialogo. Questa finestra di dialogo conserverà i risultati prodotti da ciascun algoritmo durante la sessione corrente, e potrai visualizzarla in qualsiasi momento selezionando *Processing* → *Visualizzatore risultati* dal menu di QGIS.

Alcune applicazioni esterne possono avere dei file in output (senza alcuna restrizione particolare come estensione) come output, che non appartengono ad alcuna delle categorie sopra elencate. Questi file di output non verranno processati da QGIS (aperti o inclusi nel progetto QGIS corrente), poiché la maggior parte delle volte corrispondono a formati file o elementi non supportati da QGIS. Questo, per esempio, è il caso dei LAS file usati per i dati LIDAR. I file vengono creati, ma non vedrai nulla all'interno della vostra sessione di lavoro QGIS.

Per tutti gli altri tipi di output, troverai una casella di controllo che potrai usare per indicare se caricare o meno il file una volta che è stato generato dall'algoritmo. Come impostazione predefinita, tutti i file vengono aperti.

Output opzionali non sono supportati, ovvero tutti gli output sono creati. Tuttavia puoi deselezionare la corrispondente casella di controllo se non sei interessato a un certo output, il che lo fa comportare come un output opzionale (in altre parole, il layer viene comunque generato, ma se lasci la casella di testo vuota, il layer viene salvato su un file temporaneo e cancellato una volta chiuso QGIS).

17.2.3 Configurazione dell'ambiente di elaborazione

Come è stato detto, il menu di configurazione dà accesso ad una nuova finestra di dialogo in cui puoi configurare la modalità con la quale funzionano gli algoritmi. I parametri di configurazione sono strutturati in blocchi separati che puoi selezionare sul lato sinistro della finestra di dialogo.

Insieme alla già citata voce *Output folder*, il blocco *General* contiene parametri per l'impostazione di default degli stili di rappresentazione dei layer di output (dei layer cioè, che sono generati usando una qualsiasi delle componenti delle GUI dell'ambiente). Appena crei lo stile che vuoi usare in QGIS, memorizzalo in un file, e successivamente inserisci il percorso del file nelle impostazioni, così che l'algoritmo sia in grado di usarlo. Ogni volta che un layer è caricato da QGIS Processing e aggiunto nell'area grafica di `dlqgl`, sarà rappresentato con questo stile.

Gli stili di visualizzazione possono essere configurati individualmente per ogni algoritmo e per ciascuno dei suoi output. Basta fare click con il pulsante destro del mouse sul nome dell'algoritmo nella casella degli strumenti e selezionare la voce *Modifica gli stili di visualizzazione per l'output*. Vedrete una finestra come quella mostrata di seguito.

Selezionate il file di stile (`.qml`) che vuoi assegnare ad ogni risultato e premi **[OK]**.

Altri parametri di configurazione nel gruppo *Generale* sono elencati di seguito:

- *Use filename as layer name*. Il nome di ogni layer risultante creato da un algoritmo è definito dall'algoritmo stesso. In alcuni casi puoi usare un nome unico, il che significa che verrà utilizzato lo stesso nome i output, a prescindere dal layer in input. In altri casi, il nome potrebbe dipendere dal nome del layer in input o da alcuni parametri utilizzati dell'algoritmo. Se hai selezionato la casella di controllo, il nome verrà preso invece dal nome del file di output. Nota che, se l'output viene salvato in un file temporaneo, il nome di questo file temporaneo è di solito lungo e privo di significato in moda da evitare conflitti con altri nomi di file già esistenti.
- *Use only selected features*. Se hai selezionato questa opzione, l'esecuzione dell'algoritmo avverrà solamente sulle geometrie selezionate. Se non hai selezionato nessuna geometria, allora l'algoritmo agirà sull'intero vettore.

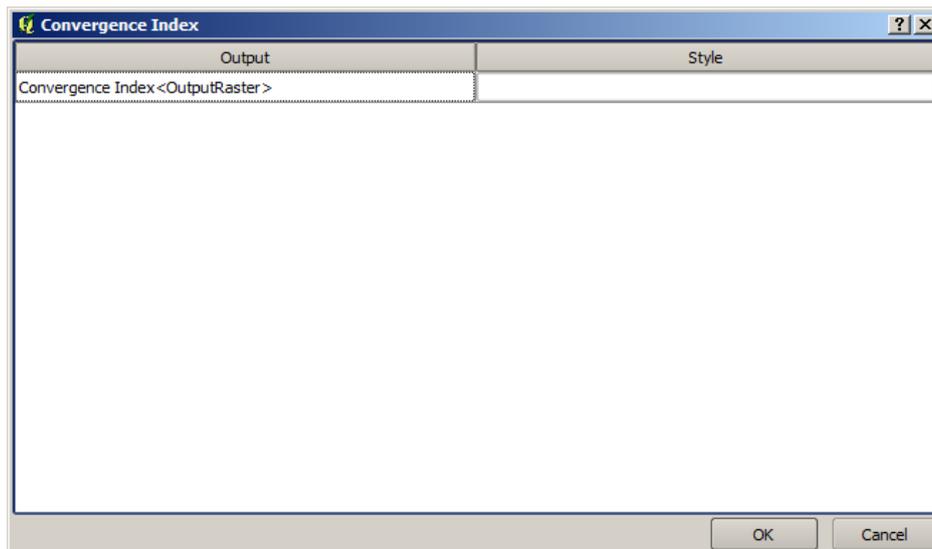


Figura 17.15: Stili di visualizzazione

- *Pre-execution script file* e *Post-execution script file*. Questi parametri fanno riferimento alla possibilità di scrivere script e sono spiegati nella sezione che tratta lo scripting e la console.

Oltre alla sezione *Generale* nella finestra di dialogo, ne troverai uno per ogni fornitore di algoritmi. Ogni programma contiene una casella di controllo *Activate* che puoi usare per far apparire o meno gli algoritmi di quel programma negli Strumenti. Inoltre alcuni fornitori di algoritmi hanno elementi specifici di configurazione che ti spiegheremo successivamente.

17.3 Modellatore grafico

Il *modellatore grafico* ti permette di creare modelli complessi grazie a un'interfaccia molto facile ed intuitiva. Quando lavori con un GIS, molte operazioni di analisi non sono isolate, ma sono inserite in un processo di operazioni. Grazie al modellatore grafico puoi unire i singoli processi in un unico grande modello; in questo modo velocizzerai molto il flusso di lavoro in quanto non dovrai più specificare ogni volta ogni singola operazione. Indipendentemente dal numero di passaggi e dai differenti algoritmi coinvolti, il modello funziona come se fosse un singolo algoritmo, facendoti risparmiare tempo e fatica, specialmente nel caso di modelli di grandi dimensioni.

Puoi aprire il modellatore grafico dal menu di Processing.

Il modellatore ha un'area grafica di lavoro dove sono visualizzati la struttura del modello ed il flusso delle operazioni che lo rappresenta. Si può usare un pannello con due riquadri sulla sinistra della finestra per aggiungere nuovi elementi al modello.

La creazione di un modello comporta due passaggi:

1. *Definizione degli input necessari*. Tutti gli input verranno aggiunti alla finestra dei parametri, in questo modo puoi impostare i valori durante l'esecuzione del modello. Il modello stesso è un algoritmo, quindi la finestra dei parametri viene automaticamente generata come accade con tutti gli algoritmi disponibili.
2. *Definizione del flusso di lavoro*. Usando i dati in input del modello, il flusso di lavoro è definito aggiungendo algoritmi e scegliendo come questi devono usare gli input o gli output generati da altri algoritmi già presenti nel modello.

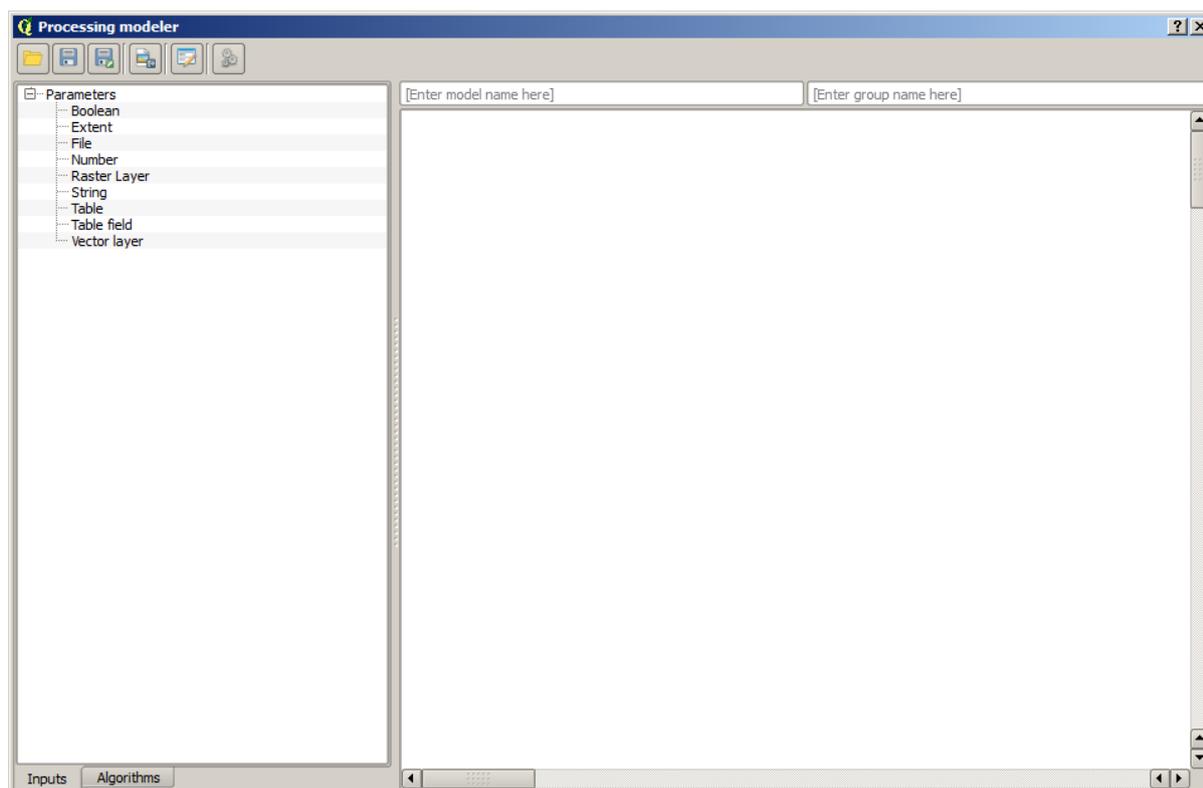


Figura 17.16: Modellatore grafico 

17.3.1 Definizione dei dati di ingresso

Il primo passo per creare un modello è quello di definire tutti gli input. Nella parte sinistra del modellatore trovi la scheda *Inputs* dove potrai scegliere i vari elementi.

- Raster
- Vettori
- Stringhe di testo
- Campi di tabelle
- Tabelle
- Estensione
- Numero
- Booleano
- File

Facendo doppio click su uno di questi elementi, apparirà una finestra di dialogo che servirà a definire le sue caratteristiche. A seconda del parametro, la finestra conterrà un solo elemento (la descrizione, ovvero quello che vedrai durante l'esecuzione del modello) oppure più elementi. Per esempio, aggiungendo un valore numerico, come puoi vedere nella figura seguente, oltre alla descrizione del parametro, dovrai definire un valore di partenza ed un intervallo di valori validi.

Per ogni dato di ingresso aggiunto, appare un nuovo elemento nel pannello grafico del modellatore.

Puoi inoltre aggiungere input trascinandolo dall'elenco e rilasciandolo nella mappa del modellatore, nella posizione in cui vuoi posizionarlo.

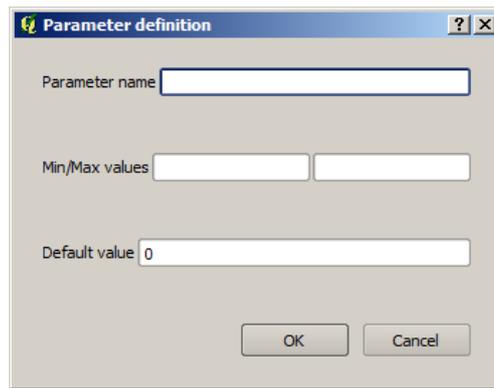


Figura 17.17: Parametri del modellatore

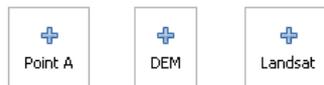


Figura 17.18: Parametri del modellatore

17.3.2 Definizione del flusso operativo

Una volta definiti gli input, devi scegliere gli algoritmi che intendi usare. Gli algoritmi si trovano nella casella *Algoritmi*, raggruppati allo stesso modo come in Strumenti.

Anche l'aspetto degli Strumenti ha due voci: semplificato e avanzato. Tuttavia, non puoi cambiare l'aspetto nel modello, lo devi fare direttamente dagli Strumenti. La voce selezionata sarà quella che verrà usata nell'elenco degli algoritmi del modellatore.

Per aggiungere un algoritmo a un modello, fai doppio click sul suo nome o trascinalo, proprio come hai fatto durante l'aggiunta di input. Apparirà una finestra di dialogo, simile a quella che viene visualizzata quando si esegue l'algoritmo dagli strumenti. Quella mostrata di seguito corrisponde all'algoritmo SAGA 'indice di convergenza', lo stesso esempio che abbiamo visto nella sezione dedicata agli Strumenti.

Come puoi vedere, ci sono diverse differenze. Al posto della casella di output dove potevi specificare il percorso dei layer e tabelle, qui c'è una semplice casella di testo. Se il layer generato da questo algoritmo è solamente un layer temporaneo che deve essere usato come input da un altro algoritmo, allora non inserire niente in questa casella. Se riempi questa casella significa che il risultato è finale e che il testo che hai inserito corrisponderà all'output finale.

Anche il valore di ogni parametro è leggermente diverso, dal momento che ci sono importanti differenze fra il contesto del modello e quello degli Strumenti. Vediamo come inserire i valori per ogni tipologia di parametro.

- Layer (raster e vettori) e tabelle. Li devi selezionare da un elenco, ma in questo caso, quello che puoi scegliere non sono i layer o le tabelle caricati in QGIS, ma semplicemente l'elenco dei tipi di file o altri layer e tabelle generati da algoritmi già presenti al modello.
- Valori numerici. Puoi inserire questi valori direttamente nella casella di testo. Questa casella però funziona anche come elenco in modo che tu possa selezionare ogni valore numerico del modello. In questo caso, il parametro considererà il valore che hai inserito durante l'esecuzione del modello.
- Stringa. Come nel caso dei valori numerici, puoi inserire una stringa o scegliere un stringa fra quelle disponibili.
- Table filed. Il parametro *parent layer* dipende da altri layer caricati nel modello e quindi lo potrai definire solamente in un secondo momento. Inserisci prima il nome del parametro e poi scegli il *parent layer* dal menu a tendina. Nel menu avrai a disposizione tutti i layer caricati nel modello.

In tutti i casi troverai un parametro aggiunti chiamato *Parent algorithms*, non disponibile quando avvii un algoritmo dagli Strumenti. Questo parametro ti permette di definire l'ordine in cui gli algoritmi vengono eseguiti, in

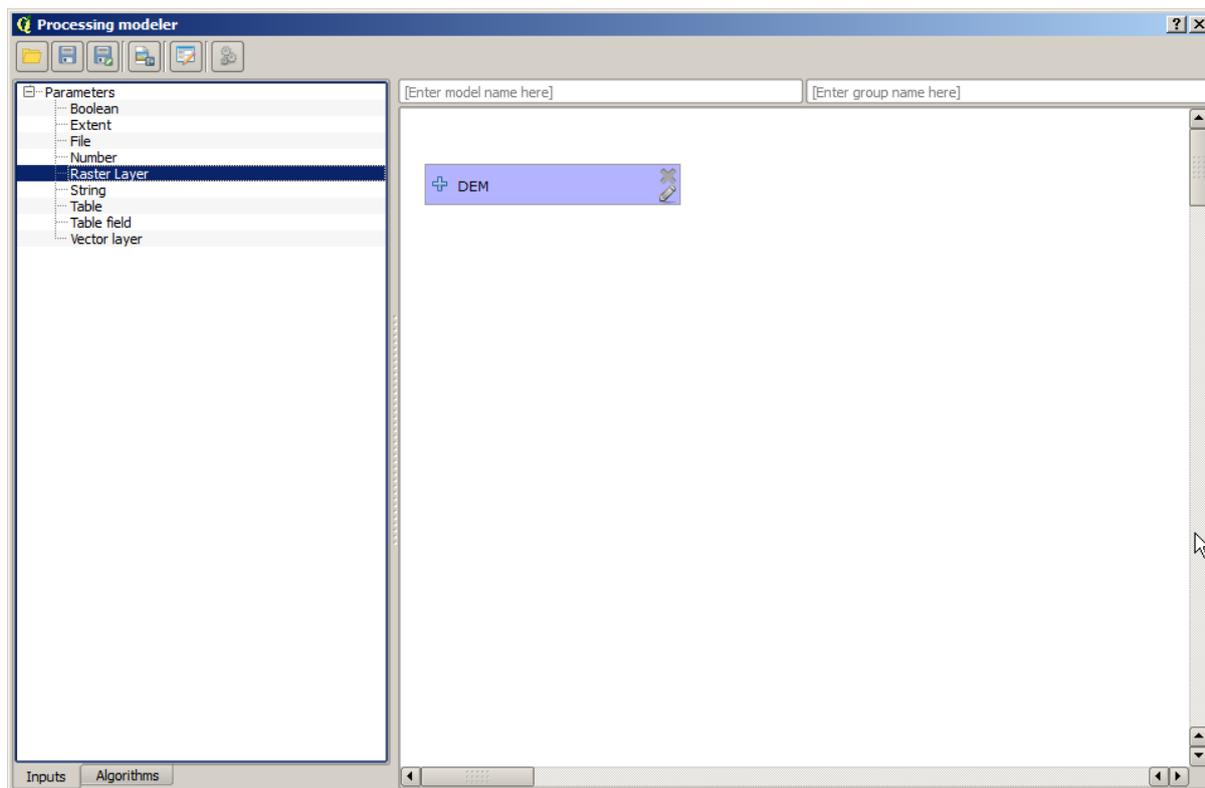


Figura 17.19: Parametri del modellatore

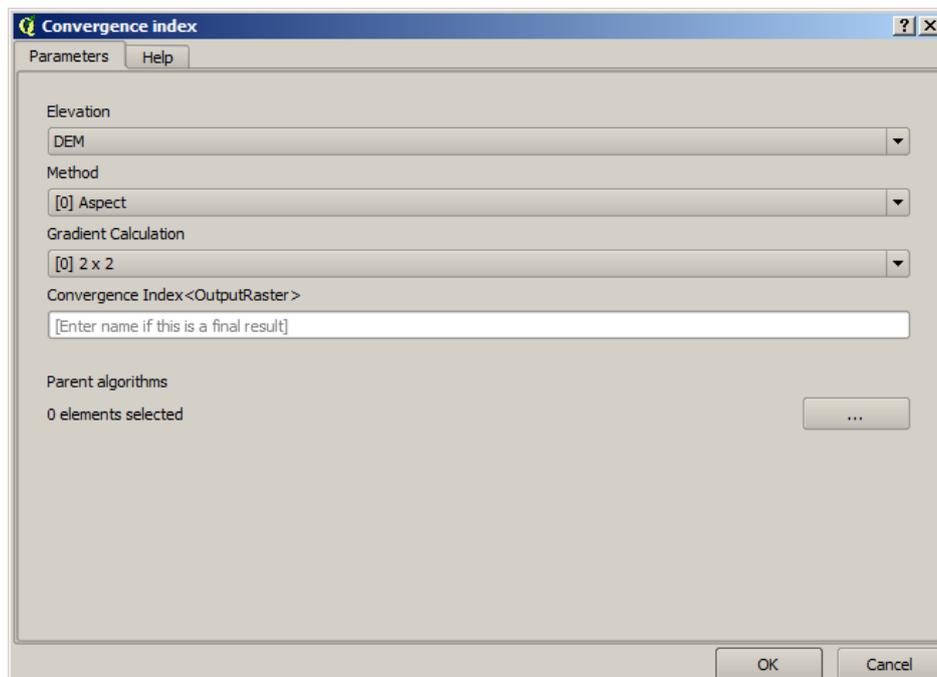


Figura 17.20: Parametri del modellatore

quanto scegli tu la nidificazione degli algoritmi. In altre parole, puoi forzare l'esecuzione di un algoritmo prima di un altro.

Quando usi l'output di un algoritmo come input per un altro algoritmo, allora il primo algoritmo è implicitamente trattato come *Parent algorithm* del secondo (e lo puoi anche vedere dalla direzione della freccia nella finestra del modello). Tuttavia, in alcuni casi un algoritmo può dipendere da un altro algoritmo anche se non richiede output (per esempio, un algoritmo che esegue un'interrogazione SQL su un database PostGIS e un altro che importa un layer nel database stesso). In questo caso, basta che selezioni l'algoritmo precedente nel parametro *Parent algorithms* e i due passaggi verranno eseguiti nell'ordine corretto.

Una volta assegnati valori corretti a tutti i parametri, premi **[OK]** e l'algoritmo verrà aggiunto alla finestra. L'algoritmo sarà collegato a tutti gli altri elementi del pannello, sia algoritmi sia dati in input, che creano risultati usati come input dall'algoritmo.

Gli elementi possono essere trascinati in un'altra posizione all'interno della mappa, per cambiare il modo in cui viene visualizzata la struttura del modulo e renderlo più chiaro e intuitivo. I collegamenti tra gli elementi vengono aggiornati automaticamente. Puoi ingrandire e ridurre utilizzando la rotellina del mouse.

Puoi eseguire l'algoritmo in ogni momento premendo il pulsante **[Run]**. Tuttavia, per poterlo usare dagli Strumenti, devi prima salvarlo e poi chiudere la finestra di dialogo del modellatore in modo da consentire al sistema di aggiornare la configurazione.

17.3.3 Salvataggio e caricamento di modelli

Usa il pulsante **[Salva]** per salvare il modello ed il pulsante **[Apri]** per aprire un modello esistente. I modelli sono salvati con l'estensione `.model`. Se hai già salvato il modello non ti verrà più chiesto il nome del file da salvare, infatti il modello verrà automaticamente sovrascritto sul file già esistente.

Prima di salvare un modello, devi inserire un nome ed un gruppo di appartenenza usando le caselle di testo nella parte alta della finestra.

I modelli salvati nella cartella `modelli` (cartella predefinita dove vengono salvati i modelli) appariranno in Strumenti, nel gruppo corrispondente. Quando apri gli Strumenti, vengono cercati tutti i file con estensione `.model` e poi caricati nella finestra. Visto che un modello è inteso come un algoritmo, lo puoi aggiungere agli Strumenti come un algoritmo qualsiasi.

Puoi specificare la cartella del modellatore nella finestra di configurazione di Processing, presente nel gruppo *Modellatore*.

I modelli caricati dalla cartella `models` appariranno non solo negli Strumenti, ma anche nell'albero degli algoritmi della scheda *Algorithms* presente nella finestra del modellatore. Questo significa che puoi includere un modello all'interno di un altro modello, proprio come se fosse un semplice algoritmo.

In alcuni casi, Processing potrebbe non essere in grado di caricare un modello perché non tutti gli algoritmi inclusi nel modello sono disponibili. Se hai usato un determinato algoritmo nel tuo modello, questo deve anche essere disponibile (cioè deve apparire negli Strumenti), altrimenti il modello non verrà caricato. Se disattivi un gestore di algoritmi nella finestra di configurazione di Processing, renderai inutilizzabili tutti i suoi algoritmi, creando problemi durante il caricamento dei modelli. Ricordati di questa procedura se hai problemi nel caricare o eseguire un modello.

17.3.4 Aggiornare il modello

Puoi modificare il modello mentre lo stai creando, ridefinendo il flusso e le relazioni fra gli algoritmi e i dati che definiscono il modello stesso.

Se premi il tasto destro del mouse su un algoritmo nel grafico che rappresenta il modello, apparirà un menu come quello sotto riportato:

Selezionando l'opzione *Remove* rimuoverai l'algoritmo selezionato. Non potrai rimuovere un algoritmo se altri algoritmi dipendono da lui, ovvero quando un algoritmo vuole come input il risultato di un altro algoritmo. Se provi a rimuovere comunque uno di questi algoritmi, apparirà una finestra di avviso.

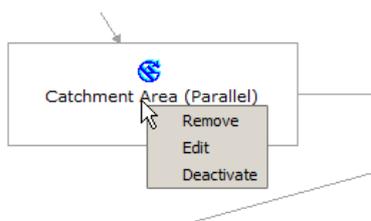


Figura 17.21: Click con il tasto destro

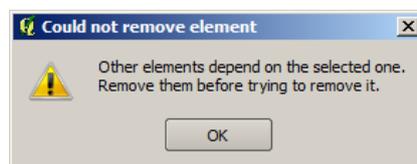


Figura 17.22: Impossibile rimuovere l'algorithmo

Selezionando l'opzione *Modifica* o semplicemente premendo due volte sull'icona dell'algorithmo apparirà la finestra dei parametri dell'algorithmo in modo che possiate cambiarne i valori. Non tutti i valori disponibili nel modello appariranno in questo caso come dati disponibili. Layer o valori generati ad un passaggio successivo del flusso del modello che possono causare dipendenze circolari non saranno disponibili.

Selezionate i nuovi valori e premete il pulsante **[OK]** come al solito. La connessione fra gli elementi del modello cambieranno di conseguenza anche nel pannello grafico.

17.3.5 Informazioni ed aiuto per l'aggiornamento del modello

Puoi anche aggiungere una documentazione ai tuoi modelli. Premi il pulsante **[Edit model help]** per aprire la finestra di dialogo corrispondente.

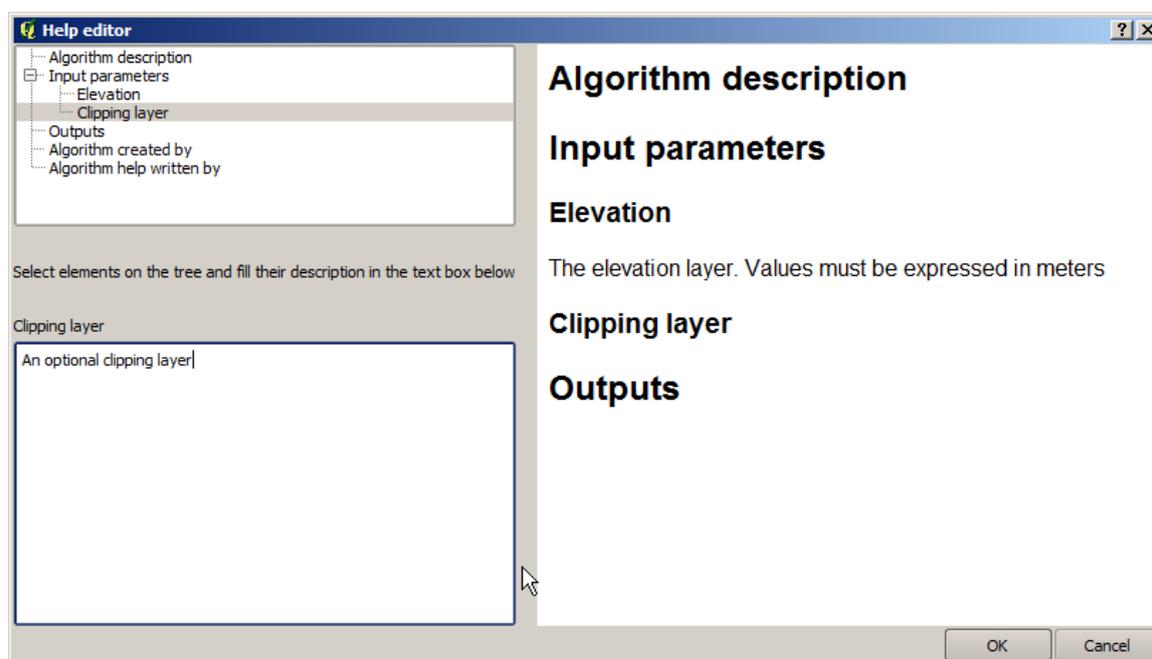


Figura 17.23: Documentazione dell'algorithmo

Sulla parte destra vedrai una semplice pagina HTML, creata usando la descrizione dei parametri di input e di output dell'algorithmo insieme ad alcuni parametri aggiuntivi come la sua descrizione e l'autore. La prima volta

che apri l'editor di aiuto, vedrai tutti i campi vuoti. Li puoi riempire usando le voci corrispondenti presenti nella parte sinistra della finestra di dialogo. Seleziona un elemento nella parte superiore e poi inserisci la descrizione nella casella di testo.

L'informazione di aiuto del modello è salvata nella stessa cartella insieme al modello stesso. Non devi preoccuparti del suo salvataggio perché viene eseguito automaticamente.

17.3.6 A proposito degli algoritmi disponibili

Potrai notare che alcuni algoritmi eseguibili dagli Strumenti non appaiono nell'elenco di quelli disponibili quando stai creando un modello. Per essere incluso in un modello, un algoritmo deve avere una semantica corretta in modo da poter essere correttamente collegato a tutti gli altri nel flusso di lavoro. Se un algoritmo non ha una semantica corretta (ad esempio, se il numero dei layer generati non può essere conosciuto in anticipo) allora non lo potrai usare all'interno del modello e quindi non apparirà nell'elenco di quelli disponibili.

Inoltre, ci sono altri algoritmi presenti nel modellatore grafico ma non negli Strumenti. Questi sono gli algoritmi pensati per essere usati solamente come parte di un modello e non hanno senso in altri contesti. Un esempio è l'algoritmo 'Calcolatore'. È infatti una semplice calcolatrice che puoi usare per cambiare valori numerici (inseriti da te o creati da qualche algoritmo). Questo strumento è molto utile in un modello ma non ha molto senso al di fuori di questo contesto.

17.4 L'interfaccia per i processi in serie

17.4.1 Introduzione

Puoi eseguire come processi in serie tutti gli algoritmi (compresi i modelli). Questo significa che puoi eseguire ogni algoritmo usando non solo un singolo input, ma anche più di uno. Questa funzionalità è particolarmente utile quando hai bisogno di processare grandi quantità di dati; non dovrai più eseguire l'algoritmo singolarmente ogni volta.

Per eseguire un algoritmo come un processo in serie, selezionarlo e col pulsante di destra del mouse scegliere la voce *Execute as batch process* dal menu che apparirà.

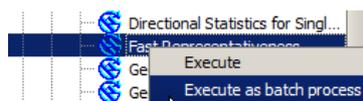


Figura 17.24: Click con tasto destro 

17.4.2 La tabella dei parametri

Eseguire un processo in serie è un'operazione simile ad un'esecuzione singola di un algoritmo. Devi definire i valori dei parametri, ma in questo caso, devi definire non solo un singolo valore per ciascuno di essi, ma un insieme di valori, uno per ogni volta che l'algoritmo verrà eseguito. I valori sono introdotti per mezzo di una tabella del tipo di quella mostrata oltre.

Ogni riga della tabella rappresenta una singola esecuzione dell'algoritmo mentre ogni cella contiene il valore di uno dei parametri caratteristici dell'algoritmo. In un certo senso, è simile alla finestra di dialogo dei parametri utilizzata quando si lancia un algoritmo da Strumenti, ma organizzata in maniera differente.

Come impostazione predefinita, la tabella contiene solo due righe. Puoi aggiungere o cancellare righe utilizzando i pulsanti della parte inferiore della finestra.

Una volta definita la dimensione della tabella, la devi riempire con i valori desiderati.

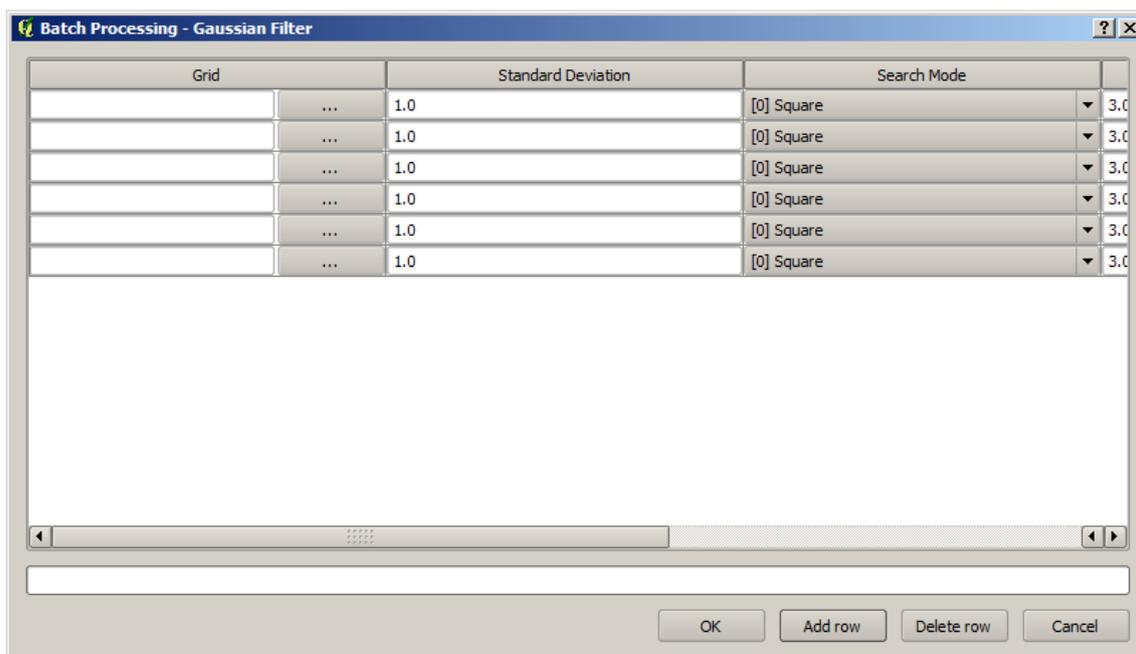


Figura 17.25: Finestra dei processi in serie

17.4.3 Compilazione della tabella dei parametri

Per la maggior parte dei parametri, la selezione del valore corretto è banale. Basta semplicemente scrivere il valore o selezionarlo dalla lista delle opzioni disponibili, a seconda del tipo di parametro.

Le principali differenze riguardano i parametri che rappresentano layer o tabelle riguarda il percorso di salvataggio dei risultati. Per quanto riguarda gli layer e tabelle in input, quando un algoritmo viene eseguito come parte di un processo in serie, i dati di input sono caricati direttamente da file e non dai dati già caricati in QGIS. Per questo motivo, puoi eseguire qualunque algoritmo in serie anche se non hai ancora caricato nessun dato e l'algoritmo non può essere eseguito dagli Strumenti.

Puoi inserire direttamente i nomi dei file per i dati in input semplicemente scrivendoli, oppure cliccando su pulsante subito a destra della cella. Si aprirà così una classica finestra di dialogo da dove potrai scegliere i layer. Puoi anche selezionare più file contemporaneamente. Se i parametri di input fanno riferimento a un dato singolo e se hai selezionato più file, ognuno di questi verrà caricato su una riga separata. Se il parametro fa riferimento a input multipli, allora tutti i file selezionati verranno aggiunti in una cella singola, separati da un punto e virgola (;).

I risultati sono sempre salvati su un file e, a differenza di quando un algoritmo viene eseguito dagli Strumenti, non puoi salvarlo come file temporaneo. Puoi digitare direttamente il nome del file di output oppure utilizzare la finestra di dialogo compare quando clicchi sul pulsante corrispondente.

Una volta selezionato il file di output, appare una nuova finestra di dialogo che permette l'autocompletamento delle altre celle nella stessa colonna (stesso parametro).

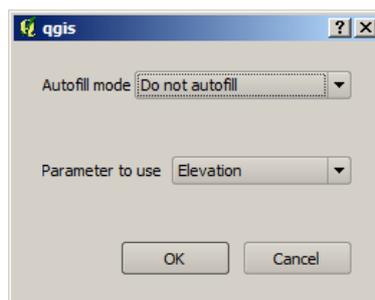


Figura 17.26: Salvataggio di processi in serie

Se mantieni il valore predefinito ('Do not autocomplete'), Processing metterà il nome del file selezionato nella cella selezionata dalla tabella dei parametri. Se selezioni una qualunque delle altre opzioni, tutte le celle sottostanti a quella selezionata saranno riempite automaticamente basandosi sul criterio definito. In questo modo, è molto più agevole riempire la tabella e puoi definire il processo in serie con meno fatica.

Puoi eseguire il riempimento automatico semplicemente aggiungendo numeri correlati al percorso del file selezionato oppure aggiungendo il valore di un altro campo alla stessa riga. Questo è particolarmente utile per dare un nome agli output che ricordi quello degli input.

Slope
C:\Documents and Settings\usuario\Mis documentos\slope1.tif
C:\Documents and Settings\usuario\Mis documentos\slope2.tif
C:\Documents and Settings\usuario\Mis documentos\slope3.tif
C:\Documents and Settings\usuario\Mis documentos\slope4.tif

Figura 17.27: Percorso dei file del processo in serie 

17.4.4 Esecuzione di un processo in serie

Per eseguire un processo in serie, una volta introdotti tutti i valori necessari, clicca semplicemente su **[OK]**. Processing mostrerà l'avanzamento globale del processo in serie nella barra di avanzamento nella parte inferiore della finestra di dialogo.

17.5 Usare gli algoritmi di Processing dalla console dei comandi

La console permette ad utenti esperti di aumentare la propria produttività e di eseguire operazioni complesse che non possono essere eseguite utilizzando uno qualsiasi degli altri elementi dell'interfaccia grafica di Processing. I modelli che richiamano diversi algoritmi possono essere definiti utilizzando l'interfaccia della riga di comando, e le operazioni aggiuntive, come i loop e le frasi condizionali possono essere aggiunte per creare flussi di lavoro più flessibili e potenti.

Non c'è una console di Processing in QGIS, ma tutti i comandi di Processing sono disponibili al posto di quelli nativi di QGIS della console di python. Questo significa che puoi incorporare questi comandi nella tua console di lavoro e collegare gli algoritmi di Processing a tutte le altre funzionalità (inclusi i metodi della API di QGIS) disponibili.

Il codice che puoi eseguire dalla console di python, anche se non richiama un metodo specifico di Processing, può essere convertito in un nuovo algoritmo che potrai richiamare in seguito dagli Strumenti, dal Modellatore grafico o da qualunque altra parte, proprio come ogni altro algoritmo. Alcuni algoritmi che trovi in Strumenti sono in effetti degli script semplici.

In questa sezione verrà spiegato come usare gli algoritmi di Processing dalla console di python e anche come scrivere un algoritmo usando python.

17.5.1 Richiamare algoritmi dalla console di python

La prima cosa da fare è importare le funzioni di Processing con la seguente istruzione:

```
>>> import processing
```

Fondamentalmente, c'è solo una cosa (interessante) che puoi fare dalla console: eseguire un algoritmo. Questo viene fatto usando il comando `runalg()`, che prende il nome dell'algoritmo da eseguire come primo parametro, e poi un numero variabile di parametri aggiuntivi che dipendono da ciò che è richiesto dall'algoritmo. Quindi la prima cosa che devi sapere è il nome dell'algoritmo da eseguire. Questo non è il nome che è riportato in Strumenti, ma un nome univoco da richiamare nella riga di comando. Per trovare il nome corretto dell'algoritmo, puoi usare il comando `algslist()`. Inserisci il seguente comando nella console:

```
>>> processing.alglist()
```

Il risultato dovrebbe essere.

```
Accumulated Cost (Anisotropic)----->saga:accumulatedcost (anisotropic)
Accumulated Cost (Isotropic)----->saga:accumulatedcost (isotropic)
Add Coordinates to points----->saga:addcoordinatestopoints
Add Grid Values to Points----->saga:addgridvaluestopoints
Add Grid Values to Shapes----->saga:addgridvaluestoshapes
Add Polygon Attributes to Points----->saga:addpolygonattributestopoints
Aggregate----->saga:aggregate
Aggregate Point Observations----->saga:aggregatepointobservations
Aggregation Index----->saga:aggregationindex
Analytical Hierarchy Process----->saga:analyticalhierarchyprocess
Analytical Hillshading----->saga:analyticalhillshading
Average With Mask 1----->saga:averagewithmask1
Average With Mask 2----->saga:averagewithmask2
Average With Threshold 1----->saga:averagewiththreshold1
Average With Threshold 2----->saga:averagewiththreshold2
Average With Threshold 3----->saga:averagewiththreshold3
B-Spline Approximation----->saga:b-splineapproximation
...
```

Questa è l'elenco di tutti gli algoritmi disponibili in ordine alfabetico, con il corrispondente nome da utilizzare nella riga di comando.

Puoi usare una stringa come parametro per questo comando. Invece di restituire l'elenco completo degli algoritmi, verranno visualizzati solo quelli che includono tale stringa. Se, per esempio, stai cercando un algoritmo per calcolare la pendenza da un DEM, inserisci `alglist ("slope")` per ottenere il seguente risultato:

```
DTM Filter (slope-based)----->saga:dtmfilter (slope-based)
Downslope Distance Gradient----->saga:downslopedistancegradient
Relative Heights and Slope Positions----->saga:relativeheightsandslopepositions
Slope Length----->saga:slopelength
Slope, Aspect, Curvature----->saga:slopeaspectcurvature
Upslope Area----->saga:upslopearea
Vegetation Index[slope based]----->saga:vegetationindex[slopebased]
```

Il risultato potrebbe cambiare a seconda degli algoritmi disponibili.

Ora è più facile ora trovare sia l'algoritmo che cercavi sia il suo nome da utilizzare nella riga di comando, in questo caso `saga: slopeaspectcurvature`.

Una volta che sai qual è il nome dell'algoritmo da utilizzare nella riga di comando, la prossima cosa da fare è conoscere la giusta sintassi per eseguirlo. Questo significa conoscere quali sono i parametri necessari e l'ordine in cui questi devono essere dichiarati quando si esegue il comando `runalg ()`. Processing ha un comando per descrivere un algoritmo in dettaglio, che puoi usare per ottenere un elenco dei parametri che un algoritmo richiede e gli output che genererà. Puoi usare il comando `alghelp (nome_algoritmo)`. Usa solo il nome dell'algoritmo nella riga di comando, non il nome descrittivo completo.

Chiamando il metodo `saga: slopeaspectcurvature` come parametro, otterrai la seguente descrizione:

```
>>> processing.alghelp("saga:slopeaspectcurvature")
ALGORITHM: Slope, Aspect, Curvature
  ELEVATION <ParameterRaster>
  METHOD <ParameterSelection>
  SLOPE <OutputRaster>
  ASPECT <OutputRaster>
  CURV <OutputRaster>
  HCURV <OutputRaster>
  VCURV <OutputRaster>
```

Ora hai tutto il necessario per eseguire qualsiasi algoritmo. Come già accennato, c'è solo un unico comando per eseguire algoritmi: “`runalg ()`”. La sua sintassi è la seguente:

```
>>> processing.runalg(name_of_the_algorithm, param1, param2, ..., paramN,
    Output1, Output2, ..., OutputN)
```

L'elenco dei parametri e degli output da aggiungere dipende dall'algoritmo che vuoi eseguire, ed è esattamente la lista che il comando `alghelp()` restituisce, nello stesso ordine, come mostrato.

A seconda del tipo di parametro, i valori sono inseriti in maniera diversa. Il seguente elenco dà una rapida panoramica di come inserire valori per ogni tipo di parametro in input:

- Raster, vettore o tabella. Basta usare una stringa con il nome che identifica l'oggetto da usare (il nome che ha nella legenda di QGIS) o il nome di un file (se il layer corrispondente non è aperto, sarà aperto, ma non aggiunto alla mappa). Se hai un'istanza di un oggetto di QGIS che rappresenta il layer, puoi anche usarla come parametro. Se l'input è opzionale e non vuoi usare alcun dato, scegli `None`.
- Selezione. Se un algoritmo ha un parametro di selezione, dovresti inserire il valore di questo parametro usando un valore intero. Per conoscere le opzioni disponibili puoi usare il comando `algorithms()` come mostrato nel seguente esempio:

```
>>> processing.algorithms("saga:slopeaspectcurvature")
METHOD (Method)
0 - [0] Maximum Slope (Travis et al. 1975)
1 - [1] Maximum Triangle Slope (Tarboton 1997)
2 - [2] Least Squares Fitted Plane (Horn 1981, Costa-Cabral & Burgess 1996)
3 - [3] Fit 2.Degree Polynom (Bauer, Rohdenburg, Bork 1985)
4 - [4] Fit 2.Degree Polynom (Heerdegen & Beran 1982)
5 - [5] Fit 2.Degree Polynom (Zevenbergen & Thorne 1987)
6 - [6] Fit 3.Degree Polynom (Haralick 1983)
```

In questo caso, l'algoritmo ha uno di questi parametri con sette opzioni, ordinate partendo da zero.

- Input multipli. Il valore è una stringa con descrittori in input separati da un punto e virgola (;). Come nel caso di layer singoli o tabelle, ogni descrittore in input può essere il nome dell'oggetto o il suo percorso.
- Campo di una Tabella da XXX. Inserisci una stringa con il nome del campo da usare. Il parametro è sensibile alle lettere maiuscole.
- Tabella fissa. Inserisci l'elenco di tutti i valori delle tabelle separati da una virgola (,) e racchiusi fra virgolette ("). I valori partono dalla riga in alto e proseguono da sinistra verso destra. Puoi usare un array 2-D per i valori che rappresentano la tabella.
- SR. Inserisci il codice EPSG del SR desiderato.
- Estensione. Usa un stringa con valori `xmin`, `xmax`, `ymin` e `ymax` separati da virgole (,).

Parametri booleani, di file, di stringa e numerici non hanno bisogno di ulteriori spiegazioni.

I parametri di input, come stringhe, booleani, o valori numerici hanno valori predefiniti. Per utilizzarli, specifica "None" nella corrispondente voce di parametro.

Per salvare i dati in output, digita il percorso del file da utilizzare, così come viene fatto in Strumenti. Se vuoi salvare il risultato in un file temporaneo, utilizzare `None`. L'estensione del file determina il formato del file. Se inserisci un'estensione del file non inclusa tra quelle supportate dall'algoritmo, verrà utilizzato il formato di file predefinito per il tipo di output e sarà aggiunta al percorso del file specificato la sua estensione corrispondente.

A differenza di quando un algoritmo viene eseguito dagli Strumenti, gli output non vengono aggiunti alla mappa quando esegui lo stesso algoritmo dalla console python. Se vuoi aggiungere l'output alla mappa lo devi fare da solo dopo l'esecuzione dell'algoritmo. Per farlo, puoi usare i comandi delle API di QGIS oppure, più facilmente, usare uno dei seguenti metodi.

Il metodo `runalg` restituisce un dizionario con i nomi degli output (quelli visti nella descrizione dell'algoritmo) come chiavi e il percorso degli output come valori. Puoi caricare questi layer con il metodo `load()`.

17.5.2 Funzioni aggiuntive per la gestione dei dati

Accanto alle funzioni usate per richiamare gli algoritmi, importare il pacchetto `processing` significa anche importare funzioni aggiuntive che permettono di lavorare con maggiore facilità con i dati, specialmente con i vettori. Sono semplicemente funzioni che incorporano le funzionalità delle API di QGIS, ma con una sintassi più semplice. Dovresti usare queste funzioni quando sviluppi nuovi algoritmi proprio perché è più facile lavorare con i dati in input.

Di sotto trovi un elenco di alcuni di questi comandi. Ulteriori informazioni possono essere trovate sotto le classi del pacchetto `processing/tools` e anche negli script di esempio forniti con QGIS.

- `getObject(obj)`: restituisce un oggetto QGIS (layer o tabella) da precedenti, che può essere un nome di file o il nome dell'oggetto nella legenda QGIS.
- `values(layer, fields)`: restituisce i valori nella tabella degli attributi di un vettore dei campi interessati. Puoi eseguire i campi come nome o come indici basati su zero. Restituisce un dict di elenchi, con gli identificatori dei campi come chiavi. Considera la selezione esistente.
- `features(layer)`: ti restituisce un iteratore sulla geometria di vettore, considerando la selezione esistente.
- `uniqueValues(layer, field)`: restituisce un elenco di valori unici per un dato attributo. Gli attributi possono essere un nome di campo o di un indice con base zero. Considera la selezione esistente.

17.5.3 Creare script ed eseguirli da Strumenti

Puoi creare i tuoi algoritmi scrivendo il codice python corrispondente e aggiungendo solo poche righe extra che forniscono le informazioni sulla semantica dell'algoritmo. In *Strumenti* puoi trovare il menu *Crea nuovo script* nel gruppo *Script*. Se fai doppio click su questo menu di aprirà una finestra di dialogo dove potrai inserire il codice. Salvando lo script nella cartella `scripts` (cartella predefinita per il salvataggio degli script) con l'estensione `.py` verrà automaticamente creato l'algoritmo corrispondente.

Il nome dell'algoritmo (quello che vedrai in Strumenti) viene creato dal nome del file, rimuovendo l'estensione del file e sostituendo i trattini bassi con spazi vuoti.

Questo di seguito è il codice che calcola l'Indice di Umidità Topografica (Topographic Wetness Index, TWI) direttamente da un DEM.

```
##dem=raster
##twi=output
ret_slope = processing.runalg("saga:slopeaspectcurvature", dem, 0, None,
                             None, None, None, None)
ret_area = processing.runalg("saga:catchmentarea(mass-fluxmethod)", dem,
                             0, False, False, False, None, None, None, None)
processing.runalg("saga:topographicwetnessindex(twi)", ret_slope['SLOPE'],
                 ret_area['AREA'], None, 1, 0, twi)
```

Come puoi vedere, durante il calcolo vengono usati tre algoritmi, tutti di SAGA. L'ultimo calcola il TWI, ma richiede un raster delle pendenze e uno di accumulo dei flussi. Anche se non hai questi layer, li puoi ricavare direttamente dal DEM grazie agli algoritmi di SAGA.

La parte del codice in cui avviene questo processo non è difficile da capire, una volta lette le sezioni precedenti di questo capitolo. Le prime linee, tuttavia, hanno bisogno di qualche spiegazione aggiuntiva. Queste forniscono a Processing le informazioni di cui ha bisogno per trasformare il codice in un algoritmo che potrai eseguire da uno dei suoi componenti, come gli Strumenti o il modellatore grafico.

Queste righe iniziano con un doppio commento di python (##) e hanno la seguente struttura:

```
[parameter_name]=[parameter_type] [optional_values]
```

Ecco un elenco di tutti i tipi di parametri supportati dagli script di Processing, la loro sintassi ed alcuni esempi.

- `raster`. Un raster.
- `vector`. Un vettore.

- `table`. Una tabella.
- `number`. Un valore numerico Devi fornire un valore predefinito, per esempio, `depth=number 2.4`.
- `string`. una stringa. Come per i valori numerici, devi fornire un valore predefinito, per esempio, `name=string Victor`.
- `boolean`. un valore booleano. Aggiungi `True` o `False` dopo per scegliere il valore predefinito. Per esempio, `verbose=boolean True`.
- `multiple raster`. Un insieme di raster in input.
- `multiple vector`. A set of input vector layers.
- `field`. Un campo nella tabella degli attributi di un vettore. Il nome del vettore deve essere aggiunto dopo il tag `field`. Ad esempio, una volta chiamato il vettore in input con `mylayer=vector`, puoi usare `myfield=field mylayer` per aggiungere come parametro un campo di quel vettore.
- `folder`. Una cartella.
- `file`. Un nome di un file.

Il nome del parametro è il nome che ti verrà mostrato durante l'esecuzione dell'algoritmo, ed è anche il nome della variabile da usare nel codice dello script. Il valore che hai inserito per quel parametro sarà assegnato a una variabile con quel nome.

Quando viene mostrato il nome del parametro, il nome verrà modificato per migliorare l'aspetto, sostituendo i trattini con degli spazi. Quindi, per esempio, se vuoi visualizzare il parametro `A numerical value`, puoi inserire il nome della variabile anche così: `A_numerical_value`.

I valori dei layer e delle tabelle sono stringhe che contengono il percorso dell'oggetto corrispondente. Per trasformarlo in un oggetto QGIS puoi usare la funzione `processing.getObjectFromUri()`. Anche gli input multipli hanno valori di stringhe che contengono il percorso dell'oggetto selezionato, separati da punto e virgola

Gli output sono definiti in maniera simile, usando i seguenti tag:

- `output raster`
- `output vector`
- `output table`
- `output html`
- `output file`
- `output number`
- `output string`

Il valore assegnati alle variabili in output è sempre una stringa con il percorso del file. Corrisponderà a un percorso temporaneo nel caso in cui non hai inserito il nome del file.

Quando dichiarati un output, l'algoritmo cercherà di aggiungerlo a QGIS una volta eseguito. Questa è la ragione per cui, anche se il comando `runalg()` non carica i layer finali, il layer TWI (nel caso dell'esempio precedente) verrà caricato, poiché è salvato nel file che hai scelto, ovvero il valore dell'output corrispondente.

Non usare il comando `load()` negli algoritmi degli script, usalo solo quando lavori con la riga di comando. Se un layer viene creato come output di un algoritmo, dovrebbe essere dichiarato come tale. Altrimenti non potrai usare l'algoritmo nel modellatore grafico dal momento che la sua sintassi (come definita dai tag spiegato sopra) non corrisponde a ciò che l'algoritmo in realtà crea.

Gli output nascosti (numeri e stringhe) non hanno un valore. Sei tu che devi assegnarli un valore. Per farlo, basta impostare il valore di una variabile con il nome utilizzato per dichiarare quell'output. Per esempio, se hai usato questa dichiarazione,

```
##average=output number
```

la linea seguente imposterà il valore dell'output a 5:

```
average = 5
```

Oltre ai tag per i parametri e gli output, è anche possibile definire il gruppo in cui verrà mostrato l'algoritmo, utilizzando il tag `group`.

Se l'algoritmo impiega molto tempo per essere eseguito, è una buona idea fornire questa informazione. Hai a disposizione due comandi globali denominati `progress` con due metodi disponibili: `setText(text)` e `setPercentage(percent)` per modificare il testo e la barra di avanzamento.

Ti abbiamo fornito diversi esempi. Controllali per vedere alcuni esempi reali di come creare algoritmi che utilizzano queste classi di Processing. Puoi fare clic con il tasto destro su un qualsiasi algoritmo e selezionare *Edit script* per modificare il codice o solo per vederlo.

17.5.4 Documentare gli script

Come nel caso dei modelli, puoi creare una documentazione aggiuntiva per gli script, per spiegare che cosa fanno e come usarli. Nella finestra di modifica dello script trovi il pulsante **[Edit script help]**. Cliccaci per aprire una finestra di editing dell'help. Controlla il capitolo sul modellatore grafico per sapere di più su questa finestra di dialogo e come usarla.

Gli help file vengono salvati nella stessa cartella dello stesso script, aggiungendo l'estensione `.help` al nome del file. Puoi modificare la guida dello script prima di salvarlo per la prima volta. Se in seguito chiudi la finestra di modifica dello script senza salvarlo (cioè lo scarti), perderai il contenuto già scritto della guida. Se hai già salvato lo script e se questo è associato ad un nome di file, il salvataggio è fatto automaticamente.

17.5.5 Script agganciati pre e post esecuzione

Gli script possono essere usati come agganci pre e post esecuzione di funzioni prima e dopo che un algoritmo venga eseguito. Li puoi usare per automatizzare dei compiti da espletare all'esecuzione di un algoritmo.

La sintassi è identica alla sintassi spiegato sopra, ma hai a disposizione anche una variabile globale chiamata `alg` che rappresenta l'algoritmo che è appena (o che sarà) stato eseguito.

Nel gruppo *General* del menu di configurazione di Processing trovi due voci chiamate *Pre-execution script file* e *Post-execution script file* dove puoi inserire il nome del file dello script che deve essere eseguito.

17.6 Il gestore della cronologia di Processing

17.6.1 La cronologia di Processing

Ogni volta che esegui un algoritmo, le informazioni sul processo sono salvate dal gestore della cronologia. Vengono salvati anche i parametri usati, la data ed il tempo di esecuzione.

In questo modo puoi tenere traccia e controllare tutto il lavoro eseguito tramite Processing e riprodurlo in tutta facilità.

Il gestore della cronologia è un insieme di registri raggruppati per data di esecuzione: in questo modo è molto facile trovare l'informazione su uno specifico algoritmo eseguito in un particolare momento.

L'informazione sul processo è conservata come una espressione a riga di comando, anche se l'algoritmo è stato avviato tramite Strumenti. Questo è molto utile se stai imparando ad usare la riga di comando perché puoi vedere il comando di un algoritmo eseguito tramite Strumenti.

Oltre a poter scorrere l'elenco del registro, puoi rieseguire i processi semplicemente facendo doppio click sul loro nome.

Durante l'esecuzione degli algoritmi, Processing comunica altre informazioni che puoi visualizzare nel registro: *Error*, *Warnings* e *Information*. Nel caso qualcosa non funzioni correttamente, dando un'occhiata a *Error* potresti

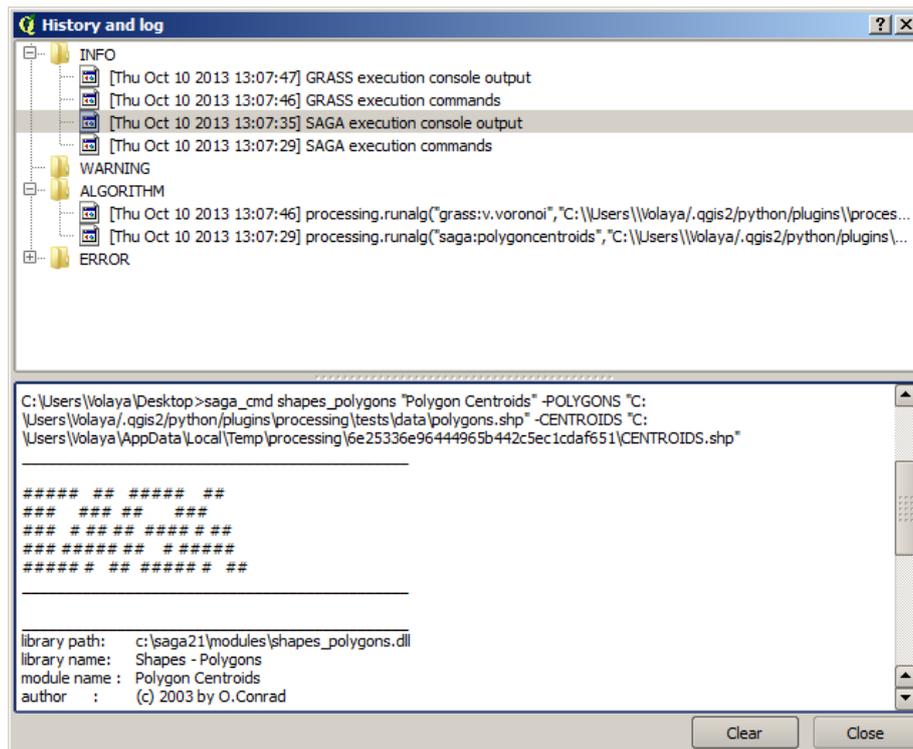


Figura 17.28: Cronologia 

capire cosa è successo. Queste informazioni sono molto utili agli sviluppatori nel momento in cui vuoi segnalare un problema o un errore.

Gli algoritmi di terze parti vengono eseguiti richiamando la loro interfaccia di riga di comando e che viene normalmente visualizzata. Anche se la console non è visualizzata, una copia completa è salvata nel gruppo *Informazioni* ogni volta che esegui quell'algoritmo. Se per esempio, hai dei problemi eseguendo un algoritmo di SAGA, cerca nella voce 'SAGA execution console output' per controllare tutti i messaggi generati da SAGA e cercare di capire dov'è il problema.

Alcuni algoritmi, anche se vengono correttamente eseguiti e forniscono un risultato, possono aggiungere delle informazioni nel gruppo *Warning* se rilevano potenziali problemi con i dati. Guarda i messaggi se vedi che il risultato non è quello atteso.

17.7 Writing new Processing algorithms as python scripts

You can create your own algorithms by writing the corresponding Python code and adding a few extra lines to supply additional information needed to define the semantics of the algorithm. You can find a *Create new script* menu under the *Tools* group in the *Script* algorithms block of the toolbox. Double-click on it to open the script edition dialog. That's where you should type your code. Saving the script from there in the *scripts* folder (the default one when you open the save file dialog), with *.py* extension, will automatically create the corresponding algorithm.

The name of the algorithm (the one you will see in the toolbox) is created from the filename, removing its extension and replacing low hyphens with blank spaces.

Let's have the following code, which calculates the Topographic Wetness Index (TWI) directly from a DEM

```
##dem=raster
##twi=output raster
ret_slope = processing.runalg("saga:slopeaspectcurvature", dem, 0, None,
                             None, None, None, None)
ret_area = processing.runalg("saga:catchmentarea", dem,
```

```
0, False, False, False, False, None, None, None, None, None)
processing.runalg("saga:topographicwetnessindextwi, ret_slope['SLOPE'],
ret_area['AREA'], None, 1, 0, twi)
```

As you can see, it involves 3 algorithms, all of them coming from SAGA. The last one of them calculates the TWI, but it needs a slope layer and a flow accumulation layer. We do not have these ones, but since we have the DEM, we can calculate them calling the corresponding SAGA algorithms.

The part of the code where this processing takes place is not difficult to understand if you have read the previous chapter. The first lines, however, need some additional explanation. They provide the information that is needed to turn your code into an algorithm that can be run from any of the GUI components, like the toolbox or the graphical modeler.

These lines start with a double Python comment symbol (##) and have the following structure

```
[parameter_name]=[parameter_type] [optional_values]
```

Here is a list of all the parameter types that are supported in processign scripts, their syntax and some examples.

- `raster`. A raster layer
- `vector`. A vector layer
- `table`. A table
- `number`. A numerical value. A default value must be provided. For instance, `depth=number 2.4`
- `string`. A text string. As in the case of numerical values, a default value must be added. For instance, `name=string Victor`
- `longstring`. Same as `string`, but a larger text box will be shown, so it is better suited for long strings, such as for a script expecting a small code snippet.
- `boolean`. A boolean value. Add `True` or `False` after it to set the default value. For example, `verbose=boolean True`.
- `multiple raster`. A set of input raster layers.
- `multiple vector`. A set of input vector layers.
- `field`. A field in the attributes table of a vector layer. The name of the layer has to be added after the `field` tag. For instance, if you have declared a vector input with `mylayer=vector`, you could use `myfield=field mylayer` to add a field from that layer as parameter.
- `folder`. A folder
- `file`. A filename
- `crs`. A Coordinate Reference System

The parameter name is the name that will be shown to the user when executing the algorithm, and also the variable name to use in the script code. The value entered by the user for that parameter will be assigned to a variable with that name.

When showing the name of the parameter to the user, the name will be edited it to improve its appearance, replacing low hyphens with spaces. So, for instance, if you want the user to see a parameter named `A numerical value`, you can use the variable name `A_numerical_value`.

Layers and tables values are strings containing the filepath of the corresponding object. To turn them into a QGIS object, you can use the `processing.getObjectFromUri()` function. Multiple inputs also have a string value, which contains the filepaths to all selected objects, separated by semicolons (;).

Outputs are defined in a similar manner, using the following tags:

- `output raster`
- `output vector`
- `output table`

- `output html`
- `output file`
- `output number`
- `output string`
- `output extent`

The value assigned to the output variables is always a string with a filepath. It will correspond to a temporary filepath in case the user has not entered any output filename.

In addition to the tags for parameters and outputs, you can also define the group under which the algorithm will be shown, using the `group` tag.

The last tag that you can use in your script header is `##nomodeler`. Use that when you do not want your algorithm to be shown in the modeler window. This should be used for algorithms that do not have a clear syntax (for instance, if the number of layers to be created is not known in advance, at design time), which make them unsuitable for the graphical modeler

17.8 Handing data produced by the algorithm

When you declare an output representing a layer (raster, vector or table), the algorithm will try to add it to QGIS once it is finished. That is the reason why, although the `runalg()` method does not load the layers it produces, the final *TWI* layer will be loaded, since it is saved to the file entered by the user, which is the value of the corresponding output.

Do not use the `load()` method in your script algorithms, but just when working with the console line. If a layer is created as output of an algorithm, it should be declared as such. Otherwise, you will not be able to properly use the algorithm in the modeler, since its syntax (as defined by the tags explained above) will not match what the algorithm really creates.

Hidden outputs (numbers and strings) do not have a value. Instead, it is you who has to assign a value to them. To do so, just set the value of a variable with the name you used to declare that output. For instance, if you have used this declaration,

```
##average=output number
```

the following line will set the value of the output to 5:

```
average = 5
```

17.9 Communicating with the user

If your algorithm takes a long time to process, it is a good idea to inform the user. You have a global named `progress` available, with two available methods: `setText(text)` and `setPercentage(percent)` to modify the progress text and the progress bar.

If you have to provide some information to the user, not related to the progress of the algorithm, you can use the `setInfo(text)` method, also from the `progress` object.

If your script has some problem, the correct way of propagating it is to raise an exception of type `GeoAlgorithmExecutionException()`. You can pass a message as argument to the constructor of the exception. Processing will take care of handling it and communicating with the user, depending on where the algorithm is being executed from (toolbox, modeler, Python console...)

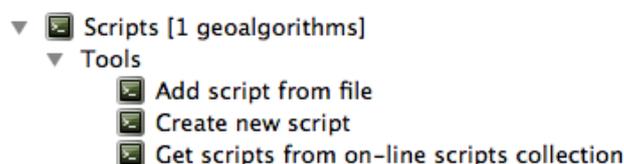
17.10 Documenting your scripts

As in the case of models, you can create additional documentation for your script, to explain what they do and how to use them. In the script editing dialog you will find a **[Edit script help]** button. Click on it and it will take you to the help editing dialog. Check the chapter about the graphical modeler to know more about this dialog and how to use it.

Help files are saved in the same folder as the script itself, adding the `.help` extension to the filename. Notice that you can edit your script's help before saving it for the first time. If you later close the script editing dialog without saving the script (i.e. you discard it), the help content you wrote will be lost. If your script was already saved and is associated to a filename, saving is done automatically.

17.11 Example scripts

Several examples are available in the on-line collection of scripts, which you can access by selecting the *Get script from on-line script collection* tool under the *Scripts/tools* entry in the toolbox.



Please, check them to see real examples of how to create algorithms using the processing framework classes. You can right-click on any script algorithm and select *Edit script* to edit its code or just to see it.

17.12 Best practices for writing script algorithms

Here's a quick summary of ideas to consider when creating your script algorithms and, especially, if you want to share with other QGIS users. Following these simple rules will ensure consistency across the different Processing elements such as the toolbox, the modeler or the batch processing interface.

- Do not load resulting layers. Let Processing handle your results and load your layers if needed.
- Always declare the outputs your algorithm creates. Avoid things such as declaring one output and then using the destination filename set for that output to create a collection of them. That will break the correct semantics of the algorithm and make it impossible to use it safely in the modeler. If you have to write an algorithm like that, make sure you add the `##nomodeler` tag.
- Do not show message boxes or use any GUI element from the script. If you want to communicate with the user, use the `setInfo()` method or throw an `GeoAlgorithmExecutionException`
- As a rule of thumb, do not forget that your algorithm might be executed in a context other than the Processing toolbox.

17.13 Pre- and post-execution script hooks

Scripts can also be used to set pre- and post-execution hooks that are run before and after an algorithm is run. This can be used to automate tasks that should be performed whenever an algorithm is executed.

The syntax is identical to the syntax explained above, but an additional global variable named `alg` is available, representing the algorithm that has just been (or is about to be) executed.

In the *General* group of the processing config dialog you will find two entries named *Pre-execution script file* and *Post-execution script file* where the filename of the scripts to be run in each case can be entered.

17.14 Configurazione di applicazioni esterne

The processing framework can be extended using additional applications. Currently, SAGA, GRASS, OTB (Orfeo Toolbox) and R are supported, along with some other command-line applications that provide spatial data analysis functionalities. Algorithms relying on an external application are managed by their own algorithm provider.

This section will show you how to configure the processing framework to include these additional applications, and it will explain some particular features of the algorithms based on them. Once you have correctly configured the system, you will be able to execute external algorithms from any component like the toolbox or the graphical modeler, just like you do with any other geospatial algorithm.

Per default, tutti gli algoritmi che si affidano ad una applicazione esterna non fornita con QGIS, non sono abilitati. E' possibile abilitarli nella finestra di configurazione. Siate sicuri che la corrispondente applicazione sia già installata nel vostro sistema. Abilitando un fornitore di algoritmi senza aver installato l'applicazione per esso necessaria, farà sì che l'algoritmo appaia tra gli strumenti, ma un errore verrà generato quando si cercherà di eseguirlo.

This is because the algorithm descriptions (needed to create the parameters dialog and provide the information needed about the algorithm) are not included with each application, but with QGIS instead. That is, they are part of QGIS, so you have them in your installation even if you have not installed any other software. Running the algorithm, however, needs the application binaries to be installed in your system.

17.14.1 Nota per gli utenti Windows

If you are not an advanced user and you are running QGIS on Windows, you might not be interested in reading the rest of this chapter. Make sure you install QGIS in your system using the standalone installer. That will automatically install SAGA, GRASS and OTB in your system and configure them so they can be run from QGIS. All the algorithms in the simplified view of the toolbox will be ready to be run without needing any further configuration. If installing through OSGeo4W application, make sure you select for installation SAGA and OTB as well.

If you want to know more about how these providers work, or if you want to use some algorithms not included in the simplified toolbox (such as R scripts), keep on reading.

17.14.2 Nota sui formati dei file

When using an external software, opening a file in QGIS does not mean that it can be opened and processed as well in that other software. In most cases, other software can read what you have opened in QGIS, but in some cases, that might not be true. When using databases or uncommon file formats, whether for raster or vector layers, problems might arise. If that happens, try to use well-known file formats that you are sure are understood by both programs, and check the console output (in the history and log dialog) to know more about what is going wrong.

L'uso di layer raster GRASS è, per esempio, un caso in cui potreste avere problemi e non essere in grado di completare il vostro lavoro se lanciate un algoritmo esterno usando in input questo layer. Per questa ragione questi layer non appariranno disponibili agli algoritmi

You should, however, find no problems at all with vector layers, since QGIS automatically converts from the original file format to one accepted by the external application before passing the layer to it. This adds extra processing time, which might be significant if the layer has a large size, so do not be surprised if it takes more time to process a layer from a DB connection than it does to process one of a similar size stored in a shapefile.

Fornitori di algoritmi che non usano applicazioni esterne possono elaborare qualsiasi layer aperto in QGIS, poichè essi lo aprono in analisi tramite QGIS.

Regarding output formats, all formats supported by QGIS as output can be used, both for raster and vector layers. Some providers do not support certain formats, but all can export to common raster layer formats that can later

be transformed by QGIS automatically. As in the case of input layers, if this conversion is needed, that might increase the processing time.

If the extension of the filename specified when calling an algorithm does not match the extension of any of the formats supported by QGIS, then a suffix will be added to set a default format. In the case of raster layers, the `.tif` extension is used, while `.shp` is used for vector layers.

17.14.3 Nota sulla selezione di layer vettore

External applications may also be made aware of the selections that exist in vector layers within QGIS. However, that requires rewriting all input vector layers, just as if they were originally in a format not supported by the external application. Only when no selection exists, or the *Use only selected features* option is not enabled in the processing general configuration, can a layer be directly passed to an external application.

In altri casi, è necessaria l'esportazione delle sole caratteristiche selezionate, che comporta un allungamento dei tempi di esecuzione

SAGA

Gli algoritmi di SAGA possono essere lanciati da QGIS se avete SAGA installato nel vostro sistema e se configurate l'ambiente di processing in modo che possa ritrovare l'eseguibile SAGA. In particolare, è necessario l'eseguibile SAGA da riga di comando, per lanciare gli algoritmi SAGA.

If you are running Windows, both the stand-alone installer and the OSGeo4W installer include SAGA along with QGIS, and the path is automatically configured, so there is no need to do anything else.

If you have installed SAGA yourself (remember, you need version 2.1), the path to the SAGA executable must be configured. To do this, open the configuration dialog. In the SAGA block, you will find a setting named *SAGA Folder*. Enter the path to the folder where SAGA is installed. Close the configuration dialog, and now you are ready to run SAGA algorithms from QGIS.

If you are running Linux, SAGA binaries are not included with SEXTANTE, so you have to download and install the software yourself. Please check the SAGA website for more information. SAGA 2.1 is needed.

In this case, there is no need to configure the path to the SAGA executable, and you will not see those folders. Instead, you must make sure that SAGA is properly installed and its folder is added to the PATH environment variable. Just open a console and type `saga_cmd` to check that the system can find where the SAGA binaries are located.

17.14.4 Le limitazioni del sistema di griglia di SAGA

Most SAGA algorithms that require several input raster layers require them to have the same grid system. That is, they must cover the same geographic area and have the same cell size, so their corresponding grids match. When calling SAGA algorithms from QGIS, you can use any layer, regardless of its cell size and extent. When multiple raster layers are used as input for a SAGA algorithm, QGIS resamples them to a common grid system and then passes them to SAGA (unless the SAGA algorithm can operate with layers from different grid systems).

The definition of that common grid system is controlled by the user, and you will find several parameters in the SAGA group of the settings window to do so. There are two ways of setting the target grid system:

- Setting it manually. You define the extent by setting the values of the following parameters:
 - *Resampling min X*
 - *Resampling max X*
 - *Resampling min Y*
 - *Resampling max Y*
 - *Resampling cellsize*

Attenzione: QGIS ricamperà il layer in ingresso a questa griglia anche se non coincidente.

- Setting it automatically from input layers. To select this option, just check the *Use min covering grid system for resampling* option. All the other settings will be ignored and the minimum extent that covers all the input layers will be used. The cell size of the target layer is the maximum of all cell sizes of the input layers.

Per gli algoritmi che non operano su molteplici layers o che non necessitano di un'unica griglia, non verrà operato alcun campionamento.

17.14.5 Limitazioni per i raster multi-banda

Unlike QGIS, SAGA has no support for multi-band layers. If you want to use a multiband layer (such as an RGB or multispectral image), you first have to split it into single-banded images. To do so, you can use the 'SAGA/Grid - Tools/Split RGB image' algorithm (which creates three images from an RGB image) or the 'SAGA/Grid - Tools/Extract band' algorithm (to extract a single band).

17.14.6 Limitations in cell size

SAGA assumes that raster layers have the same cell size in the X and Y axis. If you are working with a layer with different values for horizontal and vertical cell size, you might get unexpected results. In this case, a warning will be added to the processing log, indicating that an input layer might not be suitable to be processed by SAGA.

17.14.7 Registrazioni di controllo

When QGIS calls SAGA, it does so using its command-line interface, thus passing a set of commands to perform all the required operations. SAGA shows its progress by writing information to the console, which includes the percentage of processing already done, along with additional content. This output is filtered and used to update the progress bar while the algorithm is running.

Both the commands sent by QGIS and the additional information printed by SAGA can be logged along with other processing log messages, and you might find them useful to track in detail what is going on when QGIS runs a SAGA algorithm. You will find two settings, namely *Log console output* and *Log execution commands*, to activate that logging mechanism.

Molti altri fornitori di algoritmi che usano applicazioni esterne e le chiamano tramite la linea di comando hanno simili opzioni, così che troverete esse anche in altre posizioni nella lista di impostazioni di processing.

R. Creating R scripts

R integration in QGIS is different from that of SAGA in that there is not a predefined set of algorithms you can run (except for a few examples). Instead, you should write your scripts and call R commands, much like you would do from R, and in a very similar manner to what we saw in the section dedicated to processing scripts. This section shows you the syntax to use to call those R commands from QGIS and how to use QGIS objects (layers, tables) in them.

The first thing you have to do, as we saw in the case of SAGA, is to tell QGIS where your R binaries are located. You can do this using the *R folder* entry in the processing configuration dialog. Once you have set that parameter, you can start creating and executing your own R scripts.

Ancora una volta, la cosa è diversa in Linux e dovete solo assicurarvi che la cartella di R sia inclusa nella variabile di ambiente PATH; se eseguendo il comando R in una console, R si avvia, allora siete pronti per partire.

Per aggiungere un nuovo algoritmo che chiama una funzione di R (o un più complesso script che avete sviluppato e vorreste averlo disponibile in QGIS) dovete creare uno script che dica a QGIS come eseguire quell'operazione ed attivare i corrispondenti comandi di R.

R script files have the extension `.rsx`, and creating them is pretty easy if you just have a basic knowledge of R syntax and R scripting. They should be stored in the R scripts folder. You can set this folder in the *R settings* group (available from the processing settings dialog), just like you do with the folder for regular processing scripts.

Let's have a look at a very simple script file, which calls the R method `spsample` to create a random grid within the boundary of the polygons in a given polygon layer. This method belongs to the `mapprools` package. Since almost all the algorithms that you might like to incorporate into QGIS will use or generate spatial data, knowledge of spatial packages like `mapprools` and, especially, `sp`, is mandatory.

```
##polyg=vector
##numpoints=number 10
##output=output vector
##sp=group
pts=spsample(polyg,numpoints,type="random")
output=SpatialPointsDataFrame(pts, as.data.frame(pts))
```

The first lines, which start with a double Python comment sign (`##`), tell QGIS the inputs of the algorithm described in the file and the outputs that it will generate. They work with exactly the same syntax as the SEXTANTE scripts that we have already seen, so they will not be described here again.

When you declare an input parameter, QGIS uses that information for two things: creating the user interface to ask the user for the value of that parameter and creating a corresponding R variable that can later be used as input for R commands.

In the above example, we are declaring an input of type `vector` named `polyg`. When executing the algorithm, QGIS will open in R the layer selected by the user and store it in a variable also named `polyg`. So, the name of a parameter is also the name of the variable that we can use in R for accessing the value of that parameter (thus, you should avoid using reserved R words as parameter names).

Spatial elements such as vector and raster layers are read using the `readOGR()` and `brick()` commands (you do not have to worry about adding those commands to your description file – QGIS will do it), and they are stored as `Spatial*DataFrame` objects. Table fields are stored as strings containing the name of the selected field.

Tables are opened using the `read.csv()` command. If a table entered by the user is not in CSV format, it will be converted prior to importing it into R.

Additionally, raster files can be read using the `readGDAL()` command instead of `brick()` by using the `##userreadgdal`.

Se siete degli utenti avanzati e non volete che sia QGIS a creare l'oggetto che rappresenta il layer, potete usare l'etichetta `##passfilename` per indicare che invece, preferite una stringa con il nome del file. In questo caso è di vostra competenza l'apertura del file prima di eseguire le operazioni sui dati in esso contenuti.

Dalla precedente informazione, è possibile capire la prima riga del nostro primo file script di esempio (prima riga che non inizia con un commento Python).

```
pts=spsample(polyg,numpoints,type="random")
```

La variabile `polyg` contiene un oggetto `SpatialPolygonsDataFrame` che può essere usato per chiamare la funzione `spsample` o similmente la funzione `numpoints` che indica il numero di punti da aggiungere alla griglia creata.

Since we have declared an output of type `vector` named `out`, we have to create a variable named `out` and store a `Spatial*DataFrame` object in it (in this case, a `SpatialPointsDataFrame`). You can use any name for your intermediate variables. Just make sure that the variable storing your final result has the same name that you used to declare it, and that it contains a suitable value.

In questo caso il risultato ottenuto dal metodo `spsample` deve essere convertito esplicitamente in un oggetto `SpatialPointsDataFrame`, poichè è esso stesso un oggetto di classe `ppp`, che non è una classe adatta ad essere restituita a QGIS.

If your algorithm generates raster layers, the way they are saved will depend on whether or not you have used the `#dontuserasterpackage` option. In you have used it, layers are saved using the `writeGDAL()` method. If not, the `writeRaster()` method from the `raster` package will be used.

Se avete usato l'opzione `#passfilename`, gli output sono generati usando il pacchetto `raster` (con `writeRaster()`), anche se non esso non è usato per gli input.

If your algorithm does not generate any layer, but rather a text result in the console instead, you have to indicate that you want the console to be shown once the execution is finished. To do so, just start the command lines that

produce the results you want to print with the > ('greater') sign. The output of all other lines will not be shown. For instance, here is the description file of an algorithm that performs a normality test on a given field (column) of the attributes of a vector layer:

```
##layer=vector
##field=field layer
##nortest=group
library(nortest)
>lillie.test(layer[[field]])
```

The output of the last line is printed, but the output of the first is not (and neither are the outputs from other command lines added automatically by QGIS).

Se il vostro algoritmo produce qualche tipo di risultato grafico (usando la funzione `plot()`) dovete aggiungere la linea seguente:

```
##showplots
```

This will cause QGIS to redirect all R graphical outputs to a temporary file, which will be opened once R execution has finished.

Sia i risultati grafici che quelli da console saranno mostrati nel gestore risultati di processing.

For more information, please check the script files provided with SEXTANTE. Most of them are rather simple and will greatly help you understand how to create your own scripts.

Nota: `rgdal` and `maptools` libraries are loaded by default, so you do not have to add the corresponding `library()` commands (you just have to make sure that those two packages are installed in your R distribution). However, other additional libraries that you might need have to be explicitly loaded. Just add the necessary commands at the beginning of your script. You also have to make sure that the corresponding packages are installed in the R distribution used by QGIS. The processing framework will not take care of any package installation. If you run a script that requires a package that is not installed, the execution will fail, and SEXTANTE will try to detect which packages are missing. You must install those missing libraries manually before you can run the algorithm.

GRASS

La configurazione di GRASS non è molto differente da quella di SAGA. Per prima cosa occorre definire, ma solo nel caso di Windows, il percorso della cartella di GRASS; Inoltre occorre definire la shell di interfaccia normalmente `msys.exe` che si trova in molte distribuzioni di GRASS per Windows) ed il suo percorso.

By default, the processing framework tries to configure its GRASS connector to use the GRASS distribution that ships along with QGIS. This should work without problems in most systems, but if you experience problems, you might have to configure the GRASS connector manually. Also, if you want to use a different GRASS installation, you can change that setting and point to the folder where the other version is installed. GRASS 6.4 is needed for algorithms to work correctly.

Se state usando Linux dovete solo assicurarvi che GRASS è correttamente installato e che può essere attivato senza problemi da una console.

GRASS algorithms use a region for calculations. This region can be defined manually using values similar to the ones found in the SAGA configuration, or automatically, taking the minimum extent that covers all the input layers used to execute the algorithm each time. If the latter approach is the behaviour you prefer, just check the *Use min covering region* option in the GRASS configuration parameters.

The last parameter that has to be configured is related to the mapset. A mapset is needed to run GRASS, and the processing framework creates a temporary one for each execution. You have to specify if the data you are working with uses geographical (lat/lon) coordinates or projected ones.

GDAL

No additional configuration is needed to run GDAL algorithms. Since they are already incorporated into QGIS, the algorithms can infer their configuration from it.

Orfeo Toolbox

Orfeo Toolbox (OTB) algorithms can be run from QGIS if you have OTB installed in your system and you have configured QGIS properly, so it can find all necessary files (command-line tools and libraries).

As in the case of SAGA, OTB binaries are included in the stand-alone installer for Windows, but they are not included if you are running Linux, so you have to download and install the software yourself. Please check the OTB website for more information.

Once OTB is installed, start QGIS, open the processing configuration dialog and configure the OTB algorithm provider. In the *Orfeo Toolbox (image analysis)* block, you will find all settings related to OTB. First, ensure that algorithms are enabled.

Then, configure the path to the folder where OTB command-line tools and libraries are installed:

-  Usually *OTB applications folder* points to `/usr/lib/otb/applications` and *OTB command line tools folder* is `/usr/bin`.
-  If you use the OSGeo4W installer, then install `otb-bin` package and enter `C:\OSGeo4W\apps\orfeotoolbox\applications` as *OTB applications folder* and `C:\OSGeo4W\bin` as *OTB command line tools folder*. These values should be configured by default, but if you have a different OTB installation, configure them to the corresponding values in your system.

TauDEM

To use this provider, you need to install TauDEM command line tools.

17.14.8 Windows

Please visit the [TauDEM homepage](#) for installation instructions and precompiled binaries for 32-bit and 64-bit systems. **IMPORTANT:** You need TauDEM 5.0.6 executables. Version 5.2 is currently not supported.

17.14.9 Linux

There are no packages for most Linux distributions, so you should compile TauDEM by yourself. As TauDEM uses MPICH2, first install it using your favorite package manager. Alternatively, TauDEM works fine with Open MPI, so you can use it instead of MPICH2.

Download TauDEM 5.0.6 [source code](#) and extract the files in some folder.

Open the `linearpart.h` file, and after line

```
#include "mpi.h"
```

add a new line with

```
#include <stdint.h>
```

così avrete

```
#include "mpi.h"  
#include <stdint.h>
```

Save the changes and close the file. Now open `tiffIO.h`, find line `#include "stdint.h"` and replace quotes (" ") with `<>`, so you'll get

```
#include <stdint.h>
```

Save the changes and close the file. Create a build directory and `cd` into it

```
mkdir build
cd build
```

Configure your build with the command

```
CXX=mpicxx cmake -DCMAKE_INSTALL_PREFIX=/usr/local ..
```

e quindi compilare

```
make
```

Finally, to install TauDEM into `/usr/local/bin`, run

```
sudo make install
```

17.15 The QGIS Commander

Processing includes a practical tool that allows you to run algorithms without having to use the toolbox, but just by typing the name of the algorithm you want to run.

This tool is known as the *QGIS commander*, and it is just a simple text box with autocompletion where you type the command you want to run.

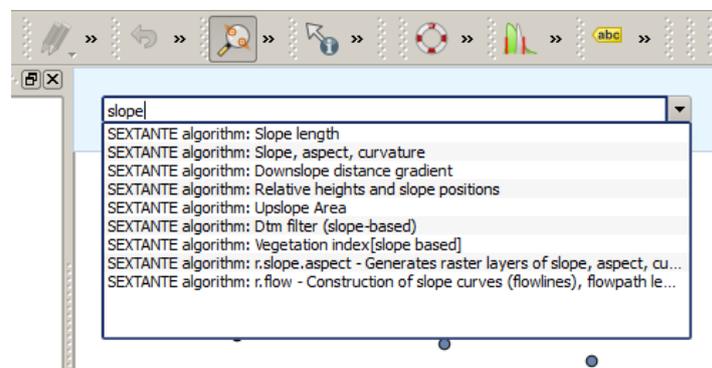


Figura 17.29: The QGIS Commander

The Commander is started from the *Analysis* menu or, more practically, by pressing `Shift + Ctrl + M` (you can change that default keyboard shortcut in the QGIS configuration if you prefer a different one). Apart from executing Processing algorithms, the Commander gives you access to most of the functionality in QGIS, which means that it gives you a practical and efficient way of running QGIS tasks and allows you to control QGIS with reduced usage of buttons and menus.

Inoltre, la Linea di comando è completamente configurabile. Puoi infatti personalizzare i comandi in modo da poterli eseguire con una breve sequenza di tasti. Questo ti permette di avere uno strumento molto potente che velocizzerà notevolmente il tuo lavoro con QGIS.

17.15.1 Comandi disponibili

I comandi disponibili nella Linea di comando ricadono nelle seguenti categorie:

- Processing algorithms. These are shown as Processing algorithm: <name of the algorithm>.
- Voci di menu. Queste sono visualizzate come Menu item: <menu entry text>. Sono disponibili tutte le voci dei menu dell'interfaccia di QGIS anche se queste sono incluse in sottomenu.
- Funzioni python. Puoi creare brevi funzioni python che verranno incluse nella lista dei comandi disponibili. Verranno mostrate come Function: <function name>.

Per eseguire un comando qualsiasi elencato sopra, inizia inserendo il nome del comando e poi seleziona l'elemento corrispondente dalla lista che mostrerà automaticamente i comandi filtrati con quello che hai inserito.

Se stai richiamando una funzione di python, puoi selezionare la voce dall'elenco. Questa avrà il prefisso Function: (per esempio Function: removeall) oppure puoi digitare direttamente il nome della funzione (removeall nell'esempio precedente). No devi aggiungere parentesi dopo il nome della funzione.

17.15.2 Creare funzioni personalizzate

Puoi aggiungere funzioni personalizzate inserendo il codice corrispondente nel file `commands.py` che puoi trovare nella cartella `.qgis2/processing/commander`. Questo è semplicemente un piccolo file python dove puoi aggiungere le funzioni di cui hai bisogno.

Il file viene creato con alcune funzioni di esempio la prima volta che avvierai la linea di comando. Se non hai ancora avviato la riga di comando puoi creare tu stesso il file. Per modificare il file usa un qualsiasi editor di testo. Puoi anche usare un editor presente in QGIS richiamando il comando `edit` dalla riga di comando. In questo modo si aprirà un editor con il file dei comandi e potrai modificarlo direttamente salvando tutti i cambiamenti.

Per esempio, puoi aggiungere la funzione seguente che rimuove tutti i layer:

```
from qgis.gui import *

def removeall():
    mapreg = QgsMapLayerRegistry.instance()
    mapreg.removeAllMapLayers()
```

Una volta che hai aggiunto la funzione, questa sarà disponibile nella linea di comando e la potrai richiamare con il comando `removeall`. Non devi fare altro che scrivere la funzione.

Le funzioni possono avere parametri aggiuntivi. Aggiungi `*args` alla definizione della funzione in modo da aggiungere parametri. Richiamando la funzione dalla linea di comando, i parametri devono essere separati da spazi.

Qui un esempio di funzione che carica un layer e prende un parametro con il nome del file del layer da caricare.

```
import processing

def load(*args):
    processing.load(args[0])
```

Se vuoi caricare il vettore `/home/myuser/points.shp`, inserisci `load /home/myuser/points.shp` nella linea di comando.

Processing providers and algorithms

18.1 GDAL algorithm provider

GDAL (Geospatial Data Abstraction Library) is a translator library for raster and vector geospatial data formats.

18.1.1 GDAL analysis

Aspect

Description

<put algorithm description here>

Parameters

Input layer [raster] <put parameter description here>

Band number [number] <put parameter description here>

Default: *1*

Compute edges [boolean] <put parameter description here>

Default: *False*

Use Zevenbergen&Thorne formula (instead of the Horn's one) [boolean] <put parameter description here>

Default: *False*

Return trigonometric angle (instead of azimuth) [boolean] <put parameter description here>

Default: *False*

Return 0 for flat (instead of -9999) [boolean] <put parameter description here>

Default: *False*

Outputs

Output file [raster] <put output description here>

Console usage

```
processing.runalg('gdalogr:aspect', input, band, compute_edges, zevenbergen, trig_angle, zero_flat)
```

See also

Color relief

Description

<put algorithm description here>

Parameters

Input layer [raster] <put parameter description here>

Band number [number] <put parameter description here>

Default: *1*

Compute edges [boolean] <put parameter description here>

Default: *False*

Color configuration file [file] <put parameter description here>

Matching mode [selection] <put parameter description here>

Options:

- 0 — “0,0,0,0” RGBA
- 1 — Exact color
- 2 — Nearest color

Default: *0*

Outputs

Output file [raster] <put output description here>

Console usage

```
processing.runalg('gdalogr:colorrelief', input, band, compute_edges, color_table, match_mode, outp
```

See also

Fill nodata

Description

<put algorithm description here>

Parameters

Input layer [raster] <put parameter description here>

Search distance [number] <put parameter description here>

Default: *100*

Smooth iterations [number] <put parameter description here>

Default: *0*

Band to operate on [number] <put parameter description here>

Default: *1*

Validity mask [raster] Optional.

<put parameter description here>

Do not use default validity mask [boolean] <put parameter description here>

Default: *False*

Outputs

Output layer [raster] <put output description here>

Console usage

```
processing.runalg('gdalogr:fillnodata', input, distance, iterations, band, mask, no_default_mask,
```

See also

Grid (Moving average)

Description

<put algorithm description here>

Parameters

Input layer [vector: point] <put parameter description here>

Z field [tablefield: numeric] Optional.

<put parameter description here>

Radius 1 [number] <put parameter description here>

Default: *0.0*

Radius 2 [number] <put parameter description here>

Default: *0.0*

Min points [number] <put parameter description here>

Default: *0.0*

Angle [number] <put parameter description here>

Default: *0.0*

Nodata [number] <put parameter description here>

Default: 0.0

Output raster type [selection] <put parameter description here>

Options:

- 0 — Byte
- 1 — Int16
- 2 — UInt16
- 3 — UInt32
- 4 — Int32
- 5 — Float32
- 6 — Float64
- 7 — CInt16
- 8 — CInt32
- 9 — CFloat32
- 10 — CFloat64

Default: 5

Outputs

Output file [raster] <put output description here>

Console usage

```
processing.runalg('gdalogr:gridaverage', input, z_field, radius_1, radius_2, min_points, angle, n
```

See also

Grid (Data metrics)

Description

<put algorithm description here>

Parameters

Input layer [vector: point] <put parameter description here>

Z field [tablefield: numeric] Optional.

<put parameter description here>

Metrics [selection] <put parameter description here>

Options:

- 0 — Minimum
- 1 — Maximum
- 2 — Range

- 3 — Count
- 4 — Average distance
- 5 — Average distance between points

Default: 0

Radius 1 [number] <put parameter description here>

Default: 0.0

Radius 2 [number] <put parameter description here>

Default: 0.0

Min points [number] <put parameter description here>

Default: 0.0

Angle [number] <put parameter description here>

Default: 0.0

Nodata [number] <put parameter description here>

Default: 0.0

Output raster type [selection] <put parameter description here>

Options:

- 0 — Byte
- 1 — Int16
- 2 — UInt16
- 3 — UInt32
- 4 — Int32
- 5 — Float32
- 6 — Float64
- 7 — CInt16
- 8 — CInt32
- 9 — CFloat32
- 10 — CFloat64

Default: 5

Outputs

Output file [raster] <put output description here>

Console usage

```
processing.runalg('gdalogr:griddatametrics', input, z_field, metric, radius_1, radius_2, min_point
```

See also

Grid (Inverse distance to a power)

Description

<put algorithm description here>

Parameters

Input layer [**vector: point**] <put parameter description here>

Z field [**tablefield: numeric**] Optional.

<put parameter description here>

Power [**number**] <put parameter description here>

Default: 2.0

Smoothing [**number**] <put parameter description here>

Default: 0.0

Radius 1 [**number**] <put parameter description here>

Default: 0.0

Radius 2 [**number**] <put parameter description here>

Default: 0.0

Max points [**number**] <put parameter description here>

Default: 0.0

Min points [**number**] <put parameter description here>

Default: 0.0

Angle [**number**] <put parameter description here>

Default: 0.0

Nodata [**number**] <put parameter description here>

Default: 0.0

Output raster type [**selection**] <put parameter description here>

Options:

- 0 — Byte
- 1 — Int16
- 2 — UInt16
- 3 — UInt32
- 4 — Int32
- 5 — Float32
- 6 — Float64
- 7 — CInt16
- 8 — CInt32
- 9 — CFloat32

- 10 — CFloat64

Default: 5

Outputs

Output file [raster] <put output description here>

Console usage

```
processing.runalg('gdalogr:gridinvdist', input, z_field, power, smothing, radius_1, radius_2, max.
```

See also

Grid (Nearest neighbor)

Description

<put algorithm description here>

Parameters

Input layer [vector: point] <put parameter description here>

Z field [tablefield: numeric] Optional.

<put parameter description here>

Radius 1 [number] <put parameter description here>

Default: 0.0

Radius 2 [number] <put parameter description here>

Default: 0.0

Angle [number] <put parameter description here>

Default: 0.0

Nodata [number] <put parameter description here>

Default: 0.0

Output raster type [selection] <put parameter description here>

Options:

- 0 — Byte
- 1 — Int16
- 2 — UInt16
- 3 — UInt32
- 4 — Int32
- 5 — Float32
- 6 — Float64
- 7 — CInt16
- 8 — CInt32

- 9 — CFloat32
- 10 — CFloat64

Default: 5

Outputs

Output file [raster] <put output description here>

Console usage

```
processing.runalg('gdalogr:gridnearestneighbor', input, z_field, radius_1, radius_2, angle, nodata)
```

See also

Hillshade

Description

<put algorithm description here>

Parameters

Input layer [raster] <put parameter description here>

Band number [number] <put parameter description here>

Default: 1

Compute edges [boolean] <put parameter description here>

Default: *False*

Use Zevenbergen&Thorne formula (instead of the Horn's one) [boolean] <put parameter description here>

Default: *False*

Z factor (vertical exaggeration) [number] <put parameter description here>

Default: 1.0

Scale (ratio of vert. units to horiz.) [number] <put parameter description here>

Default: 1.0

Azimuth of the light [number] <put parameter description here>

Default: 315.0

Altitude of the light [number] <put parameter description here>

Default: 45.0

Outputs

Output file [raster] <put output description here>

Console usage

```
processing.runalg('gdalogr:hillshade', input, band, compute_edges, zevenbergen, z_factor, scale, a
```

See also

Near black

Description

<put algorithm description here>

Parameters

Input layer [raster] <put parameter description here>

How far from black (white) [number] <put parameter description here>

Default: *15*

Search for nearly white pixels instead of nearly black [boolean] <put parameter description here>

Default: *False*

Outputs

Output layer [raster] <put output description here>

Console usage

```
processing.runalg('gdalogr:nearblack', input, near, white, output)
```

See also

Proximity (raster distance)

Description

<put algorithm description here>

Parameters

Input layer [raster] <put parameter description here>

Values [string] <put parameter description here>

Default: *(not set)*

Dist units [selection] <put parameter description here>

Options:

- 0 — GEO
- 1 — PIXEL

Default: 0

Max dist (negative value to ignore) [number] <put parameter description here>

Default: -1

No data (negative value to ignore) [number] <put parameter description here>

Default: -1

Fixed buf val (negative value to ignore) [number] <put parameter description here>

Default: -1

Output raster type [selection] <put parameter description here>

Options:

- 0 — Byte
- 1 — Int16
- 2 — UInt16
- 3 — UInt32
- 4 — Int32
- 5 — Float32
- 6 — Float64
- 7 — CInt16
- 8 — CInt32
- 9 — CFloat32
- 10 — CFloat64

Default: 5

Outputs

Output layer [raster] <put output description here>

Console usage

```
processing.runalg('gdalogr:proximity', input, values, units, max_dist, nodata, buf_val, rtype, out)
```

See also

Roughness

Description

<put algorithm description here>

Parameters

Input layer [raster] <put parameter description here>

Band number [number] <put parameter description here>

Default: 1

Compute edges [boolean] <put parameter description here>

Default: *False*

Outputs

Output file [raster] <put output description here>

Console usage

```
processing.runalg('gdalogr:roughness', input, band, compute_edges, output)
```

See also

Sieve

Description

<put algorithm description here>

Parameters

Input layer [raster] <put parameter description here>

Threshold [number] <put parameter description here>

Default: 2

Pixel connection [selection] <put parameter description here>

Options:

- 0 — 4
- 1 — 8

Default: 0

Outputs

Output layer [raster] <put output description here>

Console usage

```
processing.runalg('gdalogr:sieve', input, threshold, connections, output)
```

See also

Slope

Description

<put algorithm description here>

Parameters

Input layer [raster] <put parameter description here>

Band number [number] <put parameter description here>

Default: *1*

Compute edges [boolean] <put parameter description here>

Default: *False*

Use Zevenbergen&Thorne formula (instead of the Horn's one) [boolean] <put parameter description here>

Default: *False*

Slope expressed as percent (instead of degrees) [boolean] <put parameter description here>

Default: *False*

Scale (ratio of vert. units to horiz.) [number] <put parameter description here>

Default: *1.0*

Outputs

Output file [raster] <put output description here>

Console usage

```
processing.runalg('gdalogr:slope', input, band, compute_edges, zevenbergen, as_percent, scale, out)
```

See also

TPI (Topographic Position Index)

Description

<put algorithm description here>

Parameters

Input layer [raster] <put parameter description here>

Band number [number] <put parameter description here>

Default: *1*

Compute edges [boolean] <put parameter description here>

Default: *False*

Outputs

Output file [raster] <put output description here>

Console usage

```
processing.runalg('gdalogr:tpitopographicpositionindex', input, band, compute_edges, output)
```

See also

TRI (Terrain Ruggedness Index)

Description

<put algorithm description here>

Parameters

Input layer [raster] <put parameter description here>

Band number [number] <put parameter description here>

Default: *1*

Compute edges [boolean] <put parameter description here>

Default: *False*

Outputs

Output file [raster] <put output description here>

Console usage

```
processing.runalg('gdalogr:triterrainruggednessindex', input, band, compute_edges, output)
```

See also

.

18.1.2 GDAL conversion

gdal2xyz

Description

<put algorithm description here>

Parameters

Input layer [raster] <put parameter description here>

Band number [number] <put parameter description here>

Default: *1*

Outputs

Output file [table] <put output description here>

Console usage

```
processing.runalg('gdalogr:gdal2xyz', input, band, output)
```

See also

PCT to RGB

Description

<put algorithm description here>

Parameters

Input layer [raster] <put parameter description here>

Band to convert [selection] <put parameter description here>

Options:

- 0 — 1
- 1 — 2
- 2 — 3
- 3 — 4
- 4 — 5
- 5 — 6
- 6 — 7
- 7 — 8
- 8 — 9
- 9 — 10
- 10 — 11
- 11 — 12
- 12 — 13
- 13 — 14
- 14 — 15
- 15 — 16
- 16 — 17
- 17 — 18
- 18 — 19
- 19 — 20
- 20 — 21

- 21 — 22
- 22 — 23
- 23 — 24
- 24 — 25

Default: 0

Outputs

Output layer [raster] <put output description here>

Console usage

```
processing.runalg('gdalogr:pcttorgb', input, nband, output)
```

See also

Polygonize (raster to vector)

Description

<put algorithm description here>

Parameters

Input layer [raster] <put parameter description here>

Output field name [string] <put parameter description here>

Default: *DN*

Outputs

Output layer [vector] <put output description here>

Console usage

```
processing.runalg('gdalogr:polygonize', input, field, output)
```

See also

Rasterize (vector to raster)

Description

<put algorithm description here>

Parameters

Input layer [vector: any] <put parameter description here>

Attribute field [tablefield: any] <put parameter description here>

Write values inside an existing raster layer (*) [boolean] <put parameter description here>

Default: *False*

Set output raster size (ignored if above option is checked) [selection] <put parameter description here>

Options:

- 0 — Output size in pixels
- 1 — Output resolution in map units per pixel

Default: *1*

Horizontal [number] <put parameter description here>

Default: *100.0*

Vertical [number] <put parameter description here>

Default: *100.0*

Raster type [selection] <put parameter description here>

Options:

- 0 — Byte
- 1 — Int16
- 2 — UInt16
- 3 — UInt32
- 4 — Int32
- 5 — Float32
- 6 — Float64
- 7 — CInt16
- 8 — CInt32
- 9 — CFloat32
- 10 — CFloat64

Default: *0*

Outputs

Output layer: mandatory to choose an existing raster layer if the (*) option is selected
<put output description here>

Console usage

```
processing.runalg('gdalogr:rasterize', input, field, writeover, dimensions, width, height, rtype,
```

See also

RGB to PCT

Description

<put algorithm description here>

Parameters

Input layer [raster] <put parameter description here>

Number of colors [number] <put parameter description here>

Default: 2

Outputs

Output layer [raster] <put output description here>

Console usage

```
processing.runalg('gdalogr:rgbtocpct', input, ncolors, output)
```

See also

Translate (convert format)

Description

<put algorithm description here>

Parameters

Input layer [raster] <put parameter description here>

Set the size of the output file (In pixels or %) [number] <put parameter description here>

Default: 100

Output size is a percentage of input size [boolean] <put parameter description here>

Default: *True*

Nodata value, leave as none to take the nodata value from input [string] <put parameter description here>

Default: *none*

Expand [selection] <put parameter description here>

Options:

- 0 — none
- 1 — gray
- 2 — rgb

- 3 — rgba

Default: 0

Output projection for output file [leave blank to use input projection] [crs]
<put parameter description here>

Default: *None*

Subset based on georeferenced coordinates [extent] <put parameter description here>

Default: *0,1,0,1*

Copy all subdatasets of this file to individual output files [boolean] <put parameter description here>

Default: *False*

Additional creation parameters [string] Optional.

<put parameter description here>

Default: *(not set)*

Output raster type [selection] <put parameter description here>

Options:

- 0 — Byte
- 1 — Int16
- 2 — UInt16
- 3 — UInt32
- 4 — Int32
- 5 — Float32
- 6 — Float64
- 7 — CInt16
- 8 — CInt32
- 9 — CFloat32
- 10 — CFloat64

Default: 5

Outputs

Output layer [raster] <put output description here>

Console usage

```
processing.runalg('gdalogr:translate', input, outsize, outsize_perc, no_data, expand, srs, projwin)
```

See also

.

18.1.3 GDAL extraction

Clip raster by extent

Description

<put algorithm description here>

Parameters

Input layer [raster] <put parameter description here>

Nodata value, leave as none to take the nodata value from input [string] <put parameter description here>

Default: *none*

Clipping extent [extent] <put parameter description here>

Default: *0,1,0,1*

Additional creation parameters [string] Optional.

<put parameter description here>

Default: *(not set)*

Outputs

Output layer [raster] <put output description here>

Console usage

```
processing.runalg('gdalogr:cliprasterbyextent', input, no_data, projwin, extra, output)
```

See also

Clip raster by mask layer

Description

<put algorithm description here>

Parameters

Input layer [raster] <put parameter description here>

Mask layer [vector: polygon] <put parameter description here>

Nodata value, leave as none to take the nodata value from input [string] <put parameter description here>

Default: *none*

Create and output alpha band [boolean] <put parameter description here>

Default: *False*

Keep resolution of output raster [boolean] <put parameter description here>

Default: *False*

Additional creation parameters [string] Optional.

<put parameter description here>

Default: *(not set)*

Outputs

Output layer [raster] <put output description here>

Console usage

```
processing.runalg('gdalogr:cliprasterbymasklayer', input, mask, no_data, alpha_band, keep_resolut
```

See also

Contour

Description

<put algorithm description here>

Parameters

Input layer [raster] <put parameter description here>

Interval between contour lines [number] <put parameter description here>

Default: *10.0*

Attribute name (if not set, no elevation attribute is attached) [string]

Optional.

<put parameter description here>

Default: *ELEV*

Additional creation parameters [string] Optional.

<put parameter description here>

Default: *(not set)*

Outputs

Output file for contour lines (vector) [vector] <put output description here>

Console usage

```
processing.runalg('gdalogr:contour', input_raster, interval, field_name, extra, output_vector)
```

See also

.

18.1.4 GDAL miscellaneous**Build Virtual Raster****Description**

<put algorithm description here>

Parameters**Input layers** [**multipleinput: rasters**] <put parameter description here>**Resolution** [**selection**] <put parameter description here>

Options:

- 0 — average
- 1 — highest
- 2 — lowest

Default: *0***Layer stack** [**boolean**] <put parameter description here>Default: *True***Allow projection difference** [**boolean**] <put parameter description here>Default: *False***Outputs****Output layer** [**raster**] <put output description here>**Console usage**

```
processing.runalg('gdalogr:buildvirtualraster', input, resolution, separate, proj_difference, outp
```

See also**Merge****Description**

<put algorithm description here>

Parameters

Input layers [**multipleinput: rasters**] <put parameter description here>

Grab pseudocolor table from first layer [**boolean**] <put parameter description here>

Default: *False*

Layer stack [**boolean**] <put parameter description here>

Default: *False*

Output raster type [**selection**] <put parameter description here>

Options:

- 0 — Byte
- 1 — Int16
- 2 — UInt16
- 3 — UInt32
- 4 — Int32
- 5 — Float32
- 6 — Float64
- 7 — CInt16
- 8 — CInt32
- 9 — CFloat32
- 10 — CFloat64

Default: 5

Outputs

Output layer [**raster**] <put output description here>

Console usage

```
processing.runalg('gdalogr:merge', input, pct, separate, rtype, output)
```

See also

Build overviews (pyramids)

Description

<put algorithm description here>

Parameters

Input layer [**raster**] <put parameter description here>

Overview levels [**string**] <put parameter description here>

Default: 2 4 8 16

Remove all existing overviews [boolean] <put parameter description here>

Default: *False*

Resampling method [selection] <put parameter description here>

Options:

- 0 — nearest
- 1 — average
- 2 — gauss
- 3 — cubic
- 4 — average_mp
- 5 — average_magphase
- 6 — mode

Default: *0*

Overview format [selection] <put parameter description here>

Options:

- 0 — Internal (if possible)
- 1 — External (GTiff .ovr)
- 2 — External (ERDAS Imagine .aux)

Default: *0*

Outputs

Output layer [raster] <put output description here>

Console usage

```
processing.runalg('gdalogr:overviews', input, levels, clean, resampling_method, format)
```

See also

Information

Description

<put algorithm description here>

Parameters

Input layer [raster] <put parameter description here>

Suppress GCP info [boolean] <put parameter description here>

Default: *False*

Suppress metadata info [boolean] <put parameter description here>

Default: *False*

Outputs

Layer information [html] <put output description here>

Console usage

```
processing.runalg('gdalorg:rasterinfo', input, nogcp, nometadata, output)
```

See also

.

18.1.5 GDAL projections

Extract projection

Description

<put algorithm description here>

Parameters

Input file [raster] <put parameter description here>

Create also .prj file [boolean] <put parameter description here>

Default: *False*

Outputs

Console usage

```
processing.runalg('gdalorg:extractprojection', input, prj_file)
```

See also

Warp (reproject)

Description

<put algorithm description here>

Parameters

Input layer [raster] <put parameter description here>

Source SRS (EPSG Code) [crs] <put parameter description here>

Default: *EPSG:4326*

Destination SRS (EPSG Code) [crs] <put parameter description here>

Default: *EPSG:4326*

Output file resolution in target georeferenced units (leave 0 for no change) [number]
 <put parameter description here>

Default: *0.0*

Resampling method [selection] <put parameter description here>

Options:

- 0 — near
- 1 — bilinear
- 2 — cubic
- 3 — cubicspline
- 4 — lanczos

Default: *0*

Additional creation parameters [string] Optional.

<put parameter description here>

Default: *(not set)*

Output raster type [selection] <put parameter description here>

Options:

- 0 — Byte
- 1 — Int16
- 2 — UInt16
- 3 — UInt32
- 4 — Int32
- 5 — Float32
- 6 — Float64
- 7 — CInt16
- 8 — CInt32
- 9 — CFloat32
- 10 — CFloat64

Default: *5*

Outputs

Output layer [raster] <put output description here>

Console usage

```
processing.runalg('gdal:warpreproject', input, source_srs, dest_srs, tr, method, extra, rtype,
```

See also

.

18.1.6 OGR conversion

Convert format

Description

<put algorithm description here>

Parameters

Input layer [vector: any] <put parameter description here>

Destination Format [selection] <put parameter description here>

Options:

- 0 — ESRI Shapefile
- 1 — GeoJSON
- 2 — GeoRSS
- 3 — SQLite
- 4 — GMT
- 5 — MapInfo File
- 6 — INTERLIS 1
- 7 — INTERLIS 2
- 8 — GML
- 9 — Geoconcept
- 10 — DXF
- 11 — DGN
- 12 — CSV
- 13 — BNA
- 14 — S57
- 15 — KML
- 16 — GPX
- 17 — PGDump
- 18 — GPSTrackMaker
- 19 — ODS
- 20 — XLSX
- 21 — PDF

Default: 0

Creation Options [string] Optional.

<put parameter description here>

Default: *(not set)*

Outputs

Output layer [vector] <put output description here>

Console usage

```
processing.runalg('gdalogr:convertformat', input_layer, format, options, output_layer)
```

See also

.

18.1.7 OGR geoprocessing

Clip vectors by extent

Description

<put algorithm description here>

Parameters

Input layer [vector: any] <put parameter description here>

Clip extent [extent] <put parameter description here>

Default: *0,1,0,1*

Additional creation Options [string] Optional.

<put parameter description here>

Default: *(not set)*

Outputs

Output layer [vector] <put output description here>

Console usage

```
processing.runalg('gdalogr:clipvectorsbyextent', input_layer, clip_extent, options, output_layer)
```

See also

Clip vectors by polygon

Description

<put algorithm description here>

Parameters

Input layer [vector: any] <put parameter description here>

Clip layer [vector: polygon] <put parameter description here>

Additional creation Options [string] Optional.

<put parameter description here>

Default: *(not set)*

Outputs

Output layer [vector] <put output description here>

Console usage

```
processing.runalg('gdalogr:clipvectorsbypolygon', input_layer, clip_layer, options, output_layer)
```

See also

.

18.1.8 OGR miscellaneous

Execute SQL

Description

<put algorithm description here>

Parameters

Input layer [vector: any] <put parameter description here>

SQL [string] <put parameter description here>

Default: *(not set)*

Outputs

SQL result [vector] <put output description here>

Console usage

```
processing.runalg('gdalogr:executesql', input, sql, output)
```

See also**Import Vector into PostGIS database (available connections)****Description**

<put algorithm description here>

Parameters

Database (connection name) [selection] <put parameter description here>

Options:

- 0 — local

Default: 0

Input layer [vector: any] <put parameter description here>

Output geometry type [selection] <put parameter description here>

Options:

- 0 —
- 1 — NONE
- 2 — GEOMETRY
- 3 — POINT
- 4 — LINESTRING
- 5 — POLYGON
- 6 — GEOMETRYCOLLECTION
- 7 — MULTIPOINT
- 8 — MULTIPOLYGON
- 9 — MULTILINESTRING

Default: 5

Input CRS (EPSG Code) [crs] <put parameter description here>

Default: *EPSG:4326*

Output CRS (EPSG Code) [crs] <put parameter description here>

Default: *EPSG:4326*

Schema name [string] Optional.

<put parameter description here>

Default: *public*

Table name, leave blank to use input name [string] Optional.

<put parameter description here>

Default: *(not set)*

Primary Key [string] Optional.

<put parameter description here>

Default: *id*

Geometry column name [string] Optional.

<put parameter description here>

Default: *geom*

Vector dimensions [selection] <put parameter description here>

Options:

- 0 — 2
- 1 — 3

Default: *0*

Distance tolerance for simplification [string] Optional.

<put parameter description here>

Default: *(not set)*

Maximum distance between 2 nodes (densification) [string] Optional.

<put parameter description here>

Default: *(not set)*

Select features by extent (defined in input layer CRS) [extent] <put parameter description here>

Default: *0,1,0,1*

Clip the input layer using the above (rectangle) extent [boolean] <put parameter description here>

Default: *False*

Select features using a SQL "WHERE" statement (Ex: column="value") [string] Optional.

<put parameter description here>

Default: *(not set)*

Group "n" features per transaction (Default: 20000) [string] Optional.

<put parameter description here>

Default: *(not set)*

Overwrite existing table? [boolean] <put parameter description here>

Default: *True*

Append to existing table? [boolean] <put parameter description here>

Default: *False*

Append and add new fields to existing table? [boolean] <put parameter description here>

Default: *False*

Do not launder columns/table name/s? [boolean] <put parameter description here>

Default: *False*

Do not create Spatial Index? [boolean] <put parameter description here>

Default: *False*

Continue after a failure, skipping the failed feature [boolean] <put parameter description here>

Default: *False*

Additional creation options [string] Optional.

<put parameter description here>

Default: *(not set)*

Outputs

Console usage

```
processing.runalg('gdalogr:importvectorintopostgisdatabaseavailableconnections', database, input_
```

See also

Import Vector into PostGIS database (new connection)

Description

<put algorithm description here>

Parameters

Input layer [vector: any] <put parameter description here>

Output geometry type [selection] <put parameter description here>

Options:

- 0 —
- 1 — NONE
- 2 — GEOMETRY
- 3 — POINT
- 4 — LINESTRING
- 5 — POLYGON
- 6 — GEOMETRYCOLLECTION
- 7 — MULTIPOINT
- 8 — MULTIPOLYGON
- 9 — MULTILINESTRING

Default: 5

Input CRS (EPSG Code) [crs] <put parameter description here>

Default: *EPSG:4326*

Output CRS (EPSG Code) [crs] <put parameter description here>

Default: *EPSG:4326*

Host [string] <put parameter description here>

Default: *localhost*

Port [string] <put parameter description here>

Default: *5432*

Username [string] <put parameter description here>

Default: *(not set)*

Database Name [string] <put parameter description here>

Default: *(not set)*

Password [string] <put parameter description here>

Default: *(not set)*

Schema name [string] Optional.

<put parameter description here>

Default: *public*

Table name, leave blank to use input name [string] Optional.

<put parameter description here>

Default: *(not set)*

Primary Key [string] Optional.

<put parameter description here>

Default: *id*

Geometry column name [string] Optional.

<put parameter description here>

Default: *geom*

Vector dimensions [selection] <put parameter description here>

Options:

- 0 — 2
- 1 — 3

Default: *0*

Distance tolerance for simplification [string] Optional.

<put parameter description here>

Default: *(not set)*

Maximum distance between 2 nodes (densification) [string] Optional.

<put parameter description here>

Default: *(not set)*

Select features by extent (defined in input layer CRS) [extent] <put parameter description here>

Default: *0,1,0,1*

Clip the input layer using the above (rectangle) extent [boolean] <put parameter description here>

Default: *False*

Select features using a SQL "WHERE" statement (Ex: column="value") [string] Optional.

<put parameter description here>

Default: *(not set)*

Group "n" features per transaction (Default: 20000) [string] Optional.

<put parameter description here>

Default: *(not set)*

Overwrite existing table? [boolean] <put parameter description here>

Default: *True*

Append to existing table? [boolean] <put parameter description here>

Default: *False*

Append and add new fields to existing table? [boolean] <put parameter description here>

Default: *False*

Do not launder columns/table name/s? [boolean] <put parameter description here>

Default: *False*

Do not create Spatial Index? [boolean] <put parameter description here>

Default: *False*

Continue after a failure, skipping the failed feature [boolean] <put parameter description here>

Default: *False*

Additional creation options [string] Optional.

<put parameter description here>

Default: *(not set)*

Outputs

Console usage

```
processing.runalg('gdalogr:importvectorintopostgisdatabasewconnection', input_layer, gtype, s_s
```

See also

Information

Description

<put algorithm description here>

Parameters

Input layer [vector: any] <put parameter description here>

Outputs

Layer information [html] <put output description here>

Console usage

```
processing.runalg('gdalogr:information', input, output)
```

See also

.

18.2 LAStools

LAStools is a collection of highly efficient, multicore command line tools for LiDAR data processing.

18.2.1 las2las_filter

Description

<put algorithm description here>

Parameters

verbose [**boolean**] <put parameter description here>

Default: *False*

input LAS/LAZ file [**file**] Optional.

<put parameter description here>

filter (by return, classification, flags) [**selection**] <put parameter description here>

Options:

- 0 — —
- 1 — keep_last
- 2 — keep_first
- 3 — keep_middle
- 4 — keep_single
- 5 — drop_single
- 6 — keep_double
- 7 — keep_class 2
- 8 — keep_class 2 8
- 9 — keep_class 8
- 10 — keep_class 6
- 11 — keep_class 9
- 12 — keep_class 3 4 5
- 13 — keep_class 2 6
- 14 — drop_class 7
- 15 — drop_withheld

Default: 0

second filter (by return, classification, flags) [selection] <put parameter description here>

Options:

- 0 — —
- 1 — keep_last
- 2 — keep_first
- 3 — keep_middle
- 4 — keep_single
- 5 — drop_single
- 6 — keep_double
- 7 — keep_class 2
- 8 — keep_class 2 8
- 9 — keep_class 8
- 10 — keep_class 6
- 11 — keep_class 9
- 12 — keep_class 3 4 5
- 13 — keep_class 2 6
- 14 — drop_class 7
- 15 — drop_withheld

Default: 0

filter (by coordinate, intensity, GPS time, ...) [selection] <put parameter description here>

Options:

- 0 — —
- 1 — clip_x_above
- 2 — clip_x_below
- 3 — clip_y_above
- 4 — clip_y_below
- 5 — clip_z_above
- 6 — clip_z_below
- 7 — drop_intensity_above
- 8 — drop_intensity_below
- 9 — drop_gps_time_above
- 10 — drop_gps_time_below
- 11 — drop_scan_angle_above
- 12 — drop_scan_angle_below
- 13 — keep_point_source
- 14 — drop_point_source
- 15 — drop_point_source_above

- 16 — drop_point_source_below
- 17 — keep_user_data
- 18 — drop_user_data
- 19 — drop_user_data_above
- 20 — drop_user_data_below
- 21 — keep_every_nth
- 22 — keep_random_fraction
- 23 — thin_with_grid

Default: 0

value for filter (by coordinate, intensity, GPS time, ...) [string] <put parameter description here>

Default: *(not set)*

second filter (by coordinate, intensity, GPS time, ...) [selection] <put parameter description here>

Options:

- 0 — —
- 1 — clip_x_above
- 2 — clip_x_below
- 3 — clip_y_above
- 4 — clip_y_below
- 5 — clip_z_above
- 6 — clip_z_below
- 7 — drop_intensity_above
- 8 — drop_intensity_below
- 9 — drop_gps_time_above
- 10 — drop_gps_time_below
- 11 — drop_scan_angle_above
- 12 — drop_scan_angle_below
- 13 — keep_point_source
- 14 — drop_point_source
- 15 — drop_point_source_above
- 16 — drop_point_source_below
- 17 — keep_user_data
- 18 — drop_user_data
- 19 — drop_user_data_above
- 20 — drop_user_data_below
- 21 — keep_every_nth
- 22 — keep_random_fraction
- 23 — thin_with_grid

Default: 0

value for second filter (by coordinate, intensity, GPS time, ...) [string] <put parameter description here>

Default: *(not set)*

Outputs

output LAS/LAZ file [file] <put output description here>

Console usage

```
processing.runalg('lidartools:las2lasfilter', verbose, input_laslaz, filter_return_class_flags1, :
```

See also

18.2.2 las2las_project

Description

<put algorithm description here>

Parameters

verbose [boolean] <put parameter description here>

Default: *False*

input LAS/LAZ file [file] Optional.

<put parameter description here>

source projection [selection] <put parameter description here>

Options:

- 0 — —
- 1 — utm
- 2 — sp83
- 3 — sp27
- 4 — longlat
- 5 — latlong

Default: *0*

source utm zone [selection] <put parameter description here>

Options:

- 0 — —
- 1 — 1 (north)
- 2 — 2 (north)
- 3 — 3 (north)
- 4 — 4 (north)
- 5 — 5 (north)
- 6 — 6 (north)

- 7 — 7 (north)
- 8 — 8 (north)
- 9 — 9 (north)
- 10 — 10 (north)
- 11 — 11 (north)
- 12 — 12 (north)
- 13 — 13 (north)
- 14 — 14 (north)
- 15 — 15 (north)
- 16 — 16 (north)
- 17 — 17 (north)
- 18 — 18 (north)
- 19 — 19 (north)
- 20 — 20 (north)
- 21 — 21 (north)
- 22 — 22 (north)
- 23 — 23 (north)
- 24 — 24 (north)
- 25 — 25 (north)
- 26 — 26 (north)
- 27 — 27 (north)
- 28 — 28 (north)
- 29 — 29 (north)
- 30 — 30 (north)
- 31 — 31 (north)
- 32 — 32 (north)
- 33 — 33 (north)
- 34 — 34 (north)
- 35 — 35 (north)
- 36 — 36 (north)
- 37 — 37 (north)
- 38 — 38 (north)
- 39 — 39 (north)
- 40 — 40 (north)
- 41 — 41 (north)
- 42 — 42 (north)
- 43 — 43 (north)
- 44 — 44 (north)
- 45 — 45 (north)

- 46 — 46 (north)
- 47 — 47 (north)
- 48 — 48 (north)
- 49 — 49 (north)
- 50 — 50 (north)
- 51 — 51 (north)
- 52 — 52 (north)
- 53 — 53 (north)
- 54 — 54 (north)
- 55 — 55 (north)
- 56 — 56 (north)
- 57 — 57 (north)
- 58 — 58 (north)
- 59 — 59 (north)
- 60 — 60 (north)
- 61 — 1 (south)
- 62 — 2 (south)
- 63 — 3 (south)
- 64 — 4 (south)
- 65 — 5 (south)
- 66 — 6 (south)
- 67 — 7 (south)
- 68 — 8 (south)
- 69 — 9 (south)
- 70 — 10 (south)
- 71 — 11 (south)
- 72 — 12 (south)
- 73 — 13 (south)
- 74 — 14 (south)
- 75 — 15 (south)
- 76 — 16 (south)
- 77 — 17 (south)
- 78 — 18 (south)
- 79 — 19 (south)
- 80 — 20 (south)
- 81 — 21 (south)
- 82 — 22 (south)
- 83 — 23 (south)
- 84 — 24 (south)

- 85 — 25 (south)
- 86 — 26 (south)
- 87 — 27 (south)
- 88 — 28 (south)
- 89 — 29 (south)
- 90 — 30 (south)
- 91 — 31 (south)
- 92 — 32 (south)
- 93 — 33 (south)
- 94 — 34 (south)
- 95 — 35 (south)
- 96 — 36 (south)
- 97 — 37 (south)
- 98 — 38 (south)
- 99 — 39 (south)
- 100 — 40 (south)
- 101 — 41 (south)
- 102 — 42 (south)
- 103 — 43 (south)
- 104 — 44 (south)
- 105 — 45 (south)
- 106 — 46 (south)
- 107 — 47 (south)
- 108 — 48 (south)
- 109 — 49 (south)
- 110 — 50 (south)
- 111 — 51 (south)
- 112 — 52 (south)
- 113 — 53 (south)
- 114 — 54 (south)
- 115 — 55 (south)
- 116 — 56 (south)
- 117 — 57 (south)
- 118 — 58 (south)
- 119 — 59 (south)
- 120 — 60 (south)

Default: 0

source state plane code [selection] <put parameter description here>

Options:

- 0 — —
- 1 — AK_10
- 2 — AK_2
- 3 — AK_3
- 4 — AK_4
- 5 — AK_5
- 6 — AK_6
- 7 — AK_7
- 8 — AK_8
- 9 — AK_9
- 10 — AL_E
- 11 — AL_W
- 12 — AR_N
- 13 — AR_S
- 14 — AZ_C
- 15 — AZ_E
- 16 — AZ_W
- 17 — CA_I
- 18 — CA_II
- 19 — CA_III
- 20 — CA_IV
- 21 — CA_V
- 22 — CA_VI
- 23 — CA_VII
- 24 — CO_C
- 25 — CO_N
- 26 — CO_S
- 27 — CT
- 28 — DE
- 29 — FL_E
- 30 — FL_N
- 31 — FL_W
- 32 — GA_E
- 33 — GA_W
- 34 — HI_1
- 35 — HI_2
- 36 — HI_3
- 37 — HI_4
- 38 — HI_5

- 39 — IA_N
- 40 — IA_S
- 41 — ID_C
- 42 — ID_E
- 43 — ID_W
- 44 — IL_E
- 45 — IL_W
- 46 — IN_E
- 47 — IN_W
- 48 — KS_N
- 49 — KS_S
- 50 — KY_N
- 51 — KY_S
- 52 — LA_N
- 53 — LA_S
- 54 — MA_I
- 55 — MA_M
- 56 — MD
- 57 — ME_E
- 58 — ME_W
- 59 — MI_C
- 60 — MI_N
- 61 — MI_S
- 62 — MN_C
- 63 — MN_N
- 64 — MN_S
- 65 — MO_C
- 66 — MO_E
- 67 — MO_W
- 68 — MS_E
- 69 — MS_W
- 70 — MT_C
- 71 — MT_N
- 72 — MT_S
- 73 — NC
- 74 — ND_N
- 75 — ND_S
- 76 — NE_N
- 77 — NE_S

- 78 — NH
- 79 — NJ
- 80 — NM_C
- 81 — NM_E
- 82 — NM_W
- 83 — NV_C
- 84 — NV_E
- 85 — NV_W
- 86 — NY_C
- 87 — NY_E
- 88 — NY_LI
- 89 — NY_W
- 90 — OH_N
- 91 — OH_S
- 92 — OK_N
- 93 — OK_S
- 94 — OR_N
- 95 — OR_S
- 96 — PA_N
- 97 — PA_S
- 98 — PR
- 99 — RI
- 100 — SC_N
- 101 — SC_S
- 102 — SD_N
- 103 — SD_S
- 104 — St.Croix
- 105 — TN
- 106 — TX_C
- 107 — TX_N
- 108 — TX_NC
- 109 — TX_S
- 110 — TX_SC
- 111 — UT_C
- 112 — UT_N
- 113 — UT_S
- 114 — VA_N
- 115 — VA_S
- 116 — VT

- 117 — WA_N
- 118 — WA_S
- 119 — WI_C
- 120 — WI_N
- 121 — WI_S
- 122 — WV_N
- 123 — WV_S
- 124 — WY_E
- 125 — WY_EC
- 126 — WY_W
- 127 — WY_WC

Default: 0

target projection [selection] <put parameter description here>

Options:

- 0 — —
- 1 — utm
- 2 — sp83
- 3 — sp27
- 4 — longlat
- 5 — latlong

Default: 0

target utm zone [selection] <put parameter description here>

Options:

- 0 — —
- 1 — 1 (north)
- 2 — 2 (north)
- 3 — 3 (north)
- 4 — 4 (north)
- 5 — 5 (north)
- 6 — 6 (north)
- 7 — 7 (north)
- 8 — 8 (north)
- 9 — 9 (north)
- 10 — 10 (north)
- 11 — 11 (north)
- 12 — 12 (north)
- 13 — 13 (north)
- 14 — 14 (north)
- 15 — 15 (north)

- 16 — 16 (north)
- 17 — 17 (north)
- 18 — 18 (north)
- 19 — 19 (north)
- 20 — 20 (north)
- 21 — 21 (north)
- 22 — 22 (north)
- 23 — 23 (north)
- 24 — 24 (north)
- 25 — 25 (north)
- 26 — 26 (north)
- 27 — 27 (north)
- 28 — 28 (north)
- 29 — 29 (north)
- 30 — 30 (north)
- 31 — 31 (north)
- 32 — 32 (north)
- 33 — 33 (north)
- 34 — 34 (north)
- 35 — 35 (north)
- 36 — 36 (north)
- 37 — 37 (north)
- 38 — 38 (north)
- 39 — 39 (north)
- 40 — 40 (north)
- 41 — 41 (north)
- 42 — 42 (north)
- 43 — 43 (north)
- 44 — 44 (north)
- 45 — 45 (north)
- 46 — 46 (north)
- 47 — 47 (north)
- 48 — 48 (north)
- 49 — 49 (north)
- 50 — 50 (north)
- 51 — 51 (north)
- 52 — 52 (north)
- 53 — 53 (north)
- 54 — 54 (north)

- 55 — 55 (north)
- 56 — 56 (north)
- 57 — 57 (north)
- 58 — 58 (north)
- 59 — 59 (north)
- 60 — 60 (north)
- 61 — 1 (south)
- 62 — 2 (south)
- 63 — 3 (south)
- 64 — 4 (south)
- 65 — 5 (south)
- 66 — 6 (south)
- 67 — 7 (south)
- 68 — 8 (south)
- 69 — 9 (south)
- 70 — 10 (south)
- 71 — 11 (south)
- 72 — 12 (south)
- 73 — 13 (south)
- 74 — 14 (south)
- 75 — 15 (south)
- 76 — 16 (south)
- 77 — 17 (south)
- 78 — 18 (south)
- 79 — 19 (south)
- 80 — 20 (south)
- 81 — 21 (south)
- 82 — 22 (south)
- 83 — 23 (south)
- 84 — 24 (south)
- 85 — 25 (south)
- 86 — 26 (south)
- 87 — 27 (south)
- 88 — 28 (south)
- 89 — 29 (south)
- 90 — 30 (south)
- 91 — 31 (south)
- 92 — 32 (south)
- 93 — 33 (south)

- 94 — 34 (south)
- 95 — 35 (south)
- 96 — 36 (south)
- 97 — 37 (south)
- 98 — 38 (south)
- 99 — 39 (south)
- 100 — 40 (south)
- 101 — 41 (south)
- 102 — 42 (south)
- 103 — 43 (south)
- 104 — 44 (south)
- 105 — 45 (south)
- 106 — 46 (south)
- 107 — 47 (south)
- 108 — 48 (south)
- 109 — 49 (south)
- 110 — 50 (south)
- 111 — 51 (south)
- 112 — 52 (south)
- 113 — 53 (south)
- 114 — 54 (south)
- 115 — 55 (south)
- 116 — 56 (south)
- 117 — 57 (south)
- 118 — 58 (south)
- 119 — 59 (south)
- 120 — 60 (south)

Default: 0

target state plane code [selection] <put parameter description here>

Options:

- 0 — —
- 1 — AK_10
- 2 — AK_2
- 3 — AK_3
- 4 — AK_4
- 5 — AK_5
- 6 — AK_6
- 7 — AK_7
- 8 — AK_8

- 9 — AK_9
- 10 — AL_E
- 11 — AL_W
- 12 — AR_N
- 13 — AR_S
- 14 — AZ_C
- 15 — AZ_E
- 16 — AZ_W
- 17 — CA_I
- 18 — CA_II
- 19 — CA_III
- 20 — CA_IV
- 21 — CA_V
- 22 — CA_VI
- 23 — CA_VII
- 24 — CO_C
- 25 — CO_N
- 26 — CO_S
- 27 — CT
- 28 — DE
- 29 — FL_E
- 30 — FL_N
- 31 — FL_W
- 32 — GA_E
- 33 — GA_W
- 34 — HI_1
- 35 — HI_2
- 36 — HI_3
- 37 — HI_4
- 38 — HI_5
- 39 — IA_N
- 40 — IA_S
- 41 — ID_C
- 42 — ID_E
- 43 — ID_W
- 44 — IL_E
- 45 — IL_W
- 46 — IN_E
- 47 — IN_W

- 48 — KS_N
- 49 — KS_S
- 50 — KY_N
- 51 — KY_S
- 52 — LA_N
- 53 — LA_S
- 54 — MA_I
- 55 — MA_M
- 56 — MD
- 57 — ME_E
- 58 — ME_W
- 59 — MI_C
- 60 — MI_N
- 61 — MI_S
- 62 — MN_C
- 63 — MN_N
- 64 — MN_S
- 65 — MO_C
- 66 — MO_E
- 67 — MO_W
- 68 — MS_E
- 69 — MS_W
- 70 — MT_C
- 71 — MT_N
- 72 — MT_S
- 73 — NC
- 74 — ND_N
- 75 — ND_S
- 76 — NE_N
- 77 — NE_S
- 78 — NH
- 79 — NJ
- 80 — NM_C
- 81 — NM_E
- 82 — NM_W
- 83 — NV_C
- 84 — NV_E
- 85 — NV_W
- 86 — NY_C

- 87 — NY_E
- 88 — NY_LI
- 89 — NY_W
- 90 — OH_N
- 91 — OH_S
- 92 — OK_N
- 93 — OK_S
- 94 — OR_N
- 95 — OR_S
- 96 — PA_N
- 97 — PA_S
- 98 — PR
- 99 — RI
- 100 — SC_N
- 101 — SC_S
- 102 — SD_N
- 103 — SD_S
- 104 — St.Croix
- 105 — TN
- 106 — TX_C
- 107 — TX_N
- 108 — TX_NC
- 109 — TX_S
- 110 — TX_SC
- 111 — UT_C
- 112 — UT_N
- 113 — UT_S
- 114 — VA_N
- 115 — VA_S
- 116 — VT
- 117 — WA_N
- 118 — WA_S
- 119 — WI_C
- 120 — WI_N
- 121 — WI_S
- 122 — WV_N
- 123 — WV_S
- 124 — WY_E
- 125 — WY_EC

- 126 — WY_W
- 127 — WY_WC

Default: 0

Outputs

output LAS/LAZ file [file] <put output description here>

Console usage

```
processing.runalg('lidartools:las2lasproject', verbose, input_laslaz, source_projection, source_u
```

See also

18.2.3 las2las_transform

Description

<put algorithm description here>

Parameters

verbose [boolean] <put parameter description here>

Default: *False*

input LAS/LAZ file [file] Optional.

<put parameter description here>

transform (coordinates) [selection] <put parameter description here>

Options:

- 0 — —
- 1 — translate_x
- 2 — translate_y
- 3 — translate_z
- 4 — scale_x
- 5 — scale_y
- 6 — scale_z
- 7 — clamp_z_above
- 8 — clamp_z_below

Default: 0

value for transform (coordinates) [string] <put parameter description here>

Default: *(not set)*

second transform (coordinates) [selection] <put parameter description here>

Options:

- 0 — —

- 1 — translate_x
- 2 — translate_y
- 3 — translate_z
- 4 — scale_x
- 5 — scale_y
- 6 — scale_z
- 7 — clamp_z_above
- 8 — clamp_z_below

Default: 0

value for second transform (coordinates) [string] <put parameter description here>

Default: *(not set)*

transform (intensities, scan angles, GPS times, ...) [selection] <put parameter description here>

Options:

- 0 — —
- 1 — scale_intensity
- 2 — translate_intensity
- 3 — clamp_intensity_above
- 4 — clamp_intensity_below
- 5 — scale_scan_angle
- 6 — translate_scan_angle
- 7 — translate_gps_time
- 8 — set_classification
- 9 — set_user_data
- 10 — set_point_source
- 11 — scale_rgb_up
- 12 — scale_rgb_down
- 13 — repair_zero_returns

Default: 0

value for transform (intensities, scan angles, GPS times, ...) [string] <put parameter description here>

Default: *(not set)*

second transform (intensities, scan angles, GPS times, ...) [selection] <put parameter description here>

Options:

- 0 — —
- 1 — scale_intensity
- 2 — translate_intensity
- 3 — clamp_intensity_above
- 4 — clamp_intensity_below

- 5 — scale_scan_angle
- 6 — translate_scan_angle
- 7 — translate_gps_time
- 8 — set_classification
- 9 — set_user_data
- 10 — set_point_source
- 11 — scale_rgb_up
- 12 — scale_rgb_down
- 13 — repair_zero_returns

Default: 0

value for second transform (intensities, scan angles, GPS times, ...) [string]
 <put parameter description here>

Default: *(not set)*

operations (first 7 need an argument) [selection] <put parameter description here>

Options:

- 0 — —
- 1 — set_point_type
- 2 — set_point_size
- 3 — set_version_minor
- 4 — set_version_major
- 5 — start_at_point
- 6 — stop_at_point
- 7 — remove_vlr
- 8 — auto_reoffset
- 9 — week_to_adjusted
- 10 — adjusted_to_week
- 11 — scale_rgb_up
- 12 — scale_rgb_down
- 13 — remove_all_vlrs
- 14 — remove_extra
- 15 — clip_to_bounding_box

Default: 0

argument for operation [string] <put parameter description here>

Default: *(not set)*

Outputs

output LAS/LAZ file [file] <put output description here>

Console usage

```
processing.runalg('lidartools:las2lastransform', verbose, input_laslaz, transform_coordinate1, tr
```

See also

18.2.4 las2txt

Description

<put algorithm description here>

Parameters

verbose [boolean] <put parameter description here>

Default: *False*

input LAS/LAZ file [file] Optional.

<put parameter description here>

parse_string [string] <put parameter description here>

Default: *xyz*

Outputs

Output ASCII file [file] <put output description here>

Console usage

```
processing.runalg('lidartools:las2txt', verbose, input_laslaz, parse_string, output)
```

See also

18.2.5 lasindex

Description

<put algorithm description here>

Parameters

verbose [boolean] <put parameter description here>

Default: *False*

input LAS/LAZ file [file] Optional.

<put parameter description here>

is mobile or terrestrial LiDAR (not airborne) [boolean] <put parameter description here>

Default: *False*

Outputs

Console usage

```
processing.runalg('lidartools:lasindex', verbose, input_laslaz, mobile_or_terrestrial)
```

See also

18.2.6 lasinfo

Description

<put algorithm description here>

Parameters

verbose [boolean] <put parameter description here>

Default: *False*

input LAS/LAZ file [file] Optional.

<put parameter description here>

Outputs

Output ASCII file [file] <put output description here>

Console usage

```
processing.runalg('lidartools:lasinfo', verbose, input_laslaz, output)
```

See also

18.2.7 lasmerge

Description

<put algorithm description here>

Parameters

verbose [boolean] <put parameter description here>

Default: *False*

files are flightlines [boolean] <put parameter description here>

Default: *True*

input LAS/LAZ file [file] Optional.

<put parameter description here>

2nd file [file] Optional.

<put parameter description here>

3rd file [file] Optional.

<put parameter description here>

4th file [file] Optional.

<put parameter description here>

5th file [file] Optional.

<put parameter description here>

6th file [file] Optional.

<put parameter description here>

7th file [file] Optional.

<put parameter description here>

Outputs

output LAS/LAZ file [file] <put output description here>

Console usage

```
processing.runalg('lidartools:lasmerge', verbose, files_are_flightlines, input_laslaz, file2, file3)
```

See also

18.2.8 lasprecision

Description

<put algorithm description here>

Parameters

verbose [boolean] <put parameter description here>

Default: *False*

input LAS/LAZ file [file] Optional.

<put parameter description here>

Outputs

Output ASCII file [file] <put output description here>

Console usage

```
processing.runalg('lidartools:lasprecision', verbose, input_laslaz, output)
```

See also

18.2.9 lasquery

Description

<put algorithm description here>

Parameters

verbose [**boolean**] <put parameter description here>

Default: *False*

area of interest [**extent**] <put parameter description here>

Default: *0,1,0,1*

Outputs

Console usage

```
processing.runalg('lidartools:lasquery', verbose, aoi)
```

See also

18.2.10 lasvalidate

Description

<put algorithm description here>

Parameters

verbose [**boolean**] <put parameter description here>

Default: *False*

input LAS/LAZ file [**file**] Optional.

<put parameter description here>

Outputs

Output XML file [**file**] <put output description here>

Console usage

```
processing.runalg('lidartools:lasvalidate', verbose, input_laslaz, output)
```

See also

18.2.11 laszip

Description

<put algorithm description here>

Parameters

verbose [**boolean**] <put parameter description here>

Default: *False*

input LAS/LAZ file [**file**] Optional.

<put parameter description here>

only report size [**boolean**] <put parameter description here>

Default: *False*

Outputs

output LAS/LAZ file [**file**] <put output description here>

Console usage

```
processing.runalg('lidartools:laszip', verbose, input_laslaz, report_size, output_laslaz)
```

See also

18.2.12 txt2las

Description

<put algorithm description here>

Parameters

verbose [**boolean**] <put parameter description here>

Default: *False*

Input ASCII file [**file**] Optional.

<put parameter description here>

parse lines as [**string**] <put parameter description here>

Default: *xyz*

skip the first n lines [**number**] <put parameter description here>

Default: *0*

resolution of x and y coordinate [**number**] <put parameter description here>

Default: *0.01*

resolution of z coordinate [number] <put parameter description here>

Default: *0.01*

Outputs

output LAS/LAZ file [file] <put output description here>

Console usage

```
processing.runalg('lidartools:txt2las', verbose, input, parse_string, skip, scale_factor_xy, scale_factor_z)
```

See also

.

18.3 Modeler Tools

18.3.1 Calculator

Description

<put algorithm description here>

Parameters

Formula [string] <put parameter description here>

Default: *(not set)*

dummy [number] <put parameter description here>

Default: *0.0*

dummy [number] <put parameter description here>

Default: 0.0

dummy [number] <put parameter description here>

Default: 0.0

Outputs

Result [number] <put output description here>

Console usage

```
processing.runalg('modelertools:calculator', formula, number0, number1, number2, number3, number4,
```

See also

18.3.2 Raster layer bounds

Description

<put algorithm description here>

Parameters

Layer [raster] <put parameter description here>

Outputs

min X [number] <put output description here>

max X [number] <put output description here>

min Y [number] <put output description here>

max Y [number] <put output description here>

Extent [extent] <put output description here>

Console usage

```
processing.runalg('modelertools:rasterlayerbounds', layer)
```

See also

18.3.3 Vector layer bounds

Description

<put algorithm description here>

Parameters

Layer [vector: any] <put parameter description here>

Outputs

min X [number] <put output description here>

max X [number] <put output description here>

min Y [number] <put output description here>

max Y [number] <put output description here>

Extent [extent] <put output description here>

Console usage

```
processing.runalg('modelertools:vectorlayerbounds', layer)
```

See also

.

18.4 OrfeoToolbox algorithm provider

Orfeo ToolBox (OTB) is an open source library of image processing algorithms. OTB is based on the medical image processing library ITK and offers particular functionalities for remote sensing image processing in general and for high spatial resolution images in particular. Targeted algorithms for high resolution optical images (Pleiades, SPOT, QuickBird, WorldView, Landsat, Ikonos), hyperspectral sensors (Hyperion) or SAR (TerraSarX, ERS, Palsar) are available.

Nota: Please remember that Processing contains only the interface description, so you need to install OTB by yourself and configure Processing properly.

.

18.4.1 Calibration

Optical calibration

Description

<put algorithm description here>

Parameters

Input [raster] <put parameter description here>

Available RAM (Mb) [number] <put parameter description here>

Default: *128*

Calibration Level [selection] <put parameter description here>

Options:

- 0 — toa

Default: *0*

Convert to milli reflectance [boolean] <put parameter description here>

Default: *True*

Clamp of reflectivity values between [0, 100] [boolean] <put parameter description here>

Default: *True*

Relative Spectral Response File [file] Optional.

<put parameter description here>

Outputs

Output [raster] <put output description here>

Console usage

```
processing.runalg('otb:opticalcalibration', -in, -ram, -level, -milli, -clamp, -rsr, -out)
```

See also

.

18.4.2 Feature extrcation

BinaryMorphologicalOperation (closing)

Description

<put algorithm description here>

Parameters

Input Image [raster] <put parameter description here>

Selected Channel [number] <put parameter description here>

Default: *1*

Available RAM (Mb) [number] <put parameter description here>

Default: *128*

Structuring Element Type [selection] <put parameter description here>

Options:

- 0 — ball

Default: *0*

The Structuring Element Radius [number] <put parameter description here>

Default: *5*

Morphological Operation [selection] <put parameter description here>

Options:

- 0 — closing

Default: *0*

Outputs

Feature Output Image [raster] <put output description here>

Console usage

`processing.runalg('otb:binarymorphologicaloperationclosing', -in, -channel, -ram, -structype, -st`

See also

BinaryMorphologicalOperation (dilate)

Description

<put algorithm description here>

Parameters

Input Image [raster] <put parameter description here>

Selected Channel [number] <put parameter description here>

Default: *1*

Available RAM (Mb) [number] <put parameter description here>

Default: *128*

Structuring Element Type [selection] <put parameter description here>

Options:

- 0 — ball

Default: *0*

The Structuring Element Radius [number] <put parameter description here>

Default: *5*

Morphological Operation [selection] <put parameter description here>

Options:

- 0 — dilate

Default: *0*

Foreground Value [number] <put parameter description here>

Default: *1*

Background Value [number] <put parameter description here>

Default: *0*

Outputs

Feature Output Image [raster] <put output description here>

Console usage

```
processing.runalg('otb:binarymorphologicaloperationdilate', -in, -channel, -ram, -structype, -stru
```

See also

BinaryMorphologicalOperation (erode)

Description

<put algorithm description here>

Parameters

Input Image [raster] <put parameter description here>

Selected Channel [number] <put parameter description here>

Default: *1*

Available RAM (Mb) [number] <put parameter description here>

Default: *128*

Structuring Element Type [selection] <put parameter description here>

Options:

- 0 — ball

Default: *0*

The Structuring Element Radius [number] <put parameter description here>

Default: *5*

Morphological Operation [selection] <put parameter description here>

Options:

- 0 — erode

Default: *0*

Outputs

Feature Output Image [raster] <put output description here>

Console usage

```
processing.runalg('otb:binarymorphologicaloperationerode', -in, -channel, -ram, -structype, -stru
```

See also

BinaryMorphologicalOperation (opening)

Description

<put algorithm description here>

Parameters

Input Image [raster] <put parameter description here>

Selected Channel [number] <put parameter description here>

Default: *1*

Available RAM (Mb) [number] <put parameter description here>

Default: *128*

Structuring Element Type [selection] <put parameter description here>

Options:

- 0 — ball

Default: *0*

The Structuring Element Radius [number] <put parameter description here>

Default: *5*

Morphological Operation [selection] <put parameter description here>

Options:

- 0 — opening

Default: *0*

Outputs

Feature Output Image [raster] <put output description here>

Console usage

```
processing.runalg('otb:binarymorphologicaloperationopening', -in, -channel, -ram, -structype, -st
```

See also

EdgeExtraction (gradient)

Description

<put algorithm description here>

Parameters

Input Image [raster] <put parameter description here>

Selected Channel [number] <put parameter description here>

Default: *1*

Available RAM (Mb) [number] <put parameter description here>

Default: *128*

Edge feature [selection] <put parameter description here>

Options:

- 0 — gradient

Default: 0

Outputs

Feature Output Image [raster] <put output description here>

Console usage

```
processing.runalg('otb:edgeextractiongradient', -in, -channel, -ram, -filter, -out)
```

See also

EdgeExtraction (sobel)

Description

<put algorithm description here>

Parameters

Input Image [raster] <put parameter description here>

Selected Channel [number] <put parameter description here>

Default: 1

Available RAM (Mb) [number] <put parameter description here>

Default: 128

Edge feature [selection] <put parameter description here>

Options:

- 0 — sobel

Default: 0

Outputs

Feature Output Image [raster] <put output description here>

Console usage

```
processing.runalg('otb:edgeextractionsobel', -in, -channel, -ram, -filter, -out)
```

See also

EdgeExtraction (touzi)

Description

<put algorithm description here>

Parameters

Input Image [raster] <put parameter description here>

Selected Channel [number] <put parameter description here>

Default: *1*

Available RAM (Mb) [number] <put parameter description here>

Default: *128*

Edge feature [selection] <put parameter description here>

Options:

- 0 — touzi

Default: *0*

The Radius [number] <put parameter description here>

Default: *1*

Outputs

Feature Output Image [raster] <put output description here>

Console usage

```
processing.runalg('otb:edgeextractiontouzi', -in, -channel, -ram, -filter, -filter.touzi.xradius,
```

See also

GrayScaleMorphologicalOperation (closing)

Description

<put algorithm description here>

Parameters

Input Image [raster] <put parameter description here>

Selected Channel [number] <put parameter description here>

Default: *1*

Available RAM (Mb) [number] <put parameter description here>

Default: *128*

Structuring Element Type [selection] <put parameter description here>

Options:

- 0 — ball

Default: *0*

The Structuring Element Radius [number] <put parameter description here>

Default: *5*

Morphological Operation [selection] <put parameter description here>

Options:

- 0 — closing

Default: 0

Outputs

Feature Output Image [raster] <put output description here>

Console usage

```
processing.runalg('otb:grayscalemorphologicaloperationclosing', -in, -channel, -ram, -structype, ...)
```

See also

GrayScaleMorphologicalOperation (dilate)

Description

<put algorithm description here>

Parameters

Input Image [raster] <put parameter description here>

Selected Channel [number] <put parameter description here>

Default: 1

Available RAM (Mb) [number] <put parameter description here>

Default: 128

Structuring Element Type [selection] <put parameter description here>

Options:

- 0 — ball

Default: 0

The Structuring Element Radius [number] <put parameter description here>

Default: 5

Morphological Operation [selection] <put parameter description here>

Options:

- 0 — dilate

Default: 0

Outputs

Feature Output Image [raster] <put output description here>

Console usage

```
processing.runalg('otb:grayscalemorphologicaloperationdilate', -in, -channel, -ram, -structype, -s
```

See also

GrayScaleMorphologicalOperation (erode)

Description

<put algorithm description here>

Parameters

Input Image [raster] <put parameter description here>

Selected Channel [number] <put parameter description here>

Default: *1*

Available RAM (Mb) [number] <put parameter description here>

Default: *128*

Structuring Element Type [selection] <put parameter description here>

Options:

- 0 — ball

Default: *0*

The Structuring Element Radius [number] <put parameter description here>

Default: *5*

Morphological Operation [selection] <put parameter description here>

Options:

- 0 — erode

Default: *0*

Outputs

Feature Output Image [raster] <put output description here>

Console usage

```
processing.runalg('otb:grayscalemorphologicaloperationerode', -in, -channel, -ram, -structype, -s
```

See also

GrayScaleMorphologicalOperation (opening)

Description

<put algorithm description here>

Parameters

Input Image [raster] <put parameter description here>

Selected Channel [number] <put parameter description here>

Default: *1*

Available RAM (Mb) [number] <put parameter description here>

Default: *128*

Structuring Element Type [selection] <put parameter description here>

Options:

- 0 — ball

Default: *0*

The Structuring Element Radius [number] <put parameter description here>

Default: *5*

Morphological Operation [selection] <put parameter description here>

Options:

- 0 — opening

Default: *0*

Outputs

Feature Output Image [raster] <put output description here>

Console usage

```
processing.runalg('otb:grayscalemorphologicaloperationopening', -in, -channel, -ram, -structype, .
```

See also

Haralick Texture Extraction

Description

<put algorithm description here>

Parameters

Input Image [raster] <put parameter description here>

Selected Channel [number] <put parameter description here>

Default: *1*

Available RAM (Mb) [number] <put parameter description here>

Default: *128*

X Radius [number] <put parameter description here>

Default: *2*

Y Radius [number] <put parameter description here>

Default: 2

X Offset [number] <put parameter description here>

Default: 1

Y Offset [number] <put parameter description here>

Default: 1

Image Minimum [number] <put parameter description here>

Default: 0

Image Maximum [number] <put parameter description here>

Default: 255

Histogram number of bin [number] <put parameter description here>

Default: 8

Texture Set Selection [selection] <put parameter description here>

Options:

- 0 — simple
- 1 — advanced
- 2 — higher

Default: 0

Outputs

Output Image [raster] <put output description here>

Console usage

```
processing.runalg('otb:haralicktextureextraction', -in, -channel, -ram, -parameters.xrad, -parameter
```

See also

Line segment detection

Description

<put algorithm description here>

Parameters

Input Image [raster] <put parameter description here>

No rescaling in [0, 255] [boolean] <put parameter description here>

Default: *True*

Outputs

Output **Detected lines** [**vector**] <put output description here>

Console usage

```
processing.runalg('otb:linesegmentdetection', -in, -norescale, -out)
```

See also

Local Statistic Extraction

Description

<put algorithm description here>

Parameters

Input Image [**raster**] <put parameter description here>

Selected Channel [**number**] <put parameter description here>

Default: *1*

Available RAM (Mb) [**number**] <put parameter description here>

Default: *128*

Neighborhood radius [**number**] <put parameter description here>

Default: *3*

Outputs

Feature Output Image [**raster**] <put output description here>

Console usage

```
processing.runalg('otb:localstatisticextraction', -in, -channel, -ram, -radius, -out)
```

See also

Multivariate alteration detector

Description

<put algorithm description here>

Parameters

Input Image 1 [raster] <put parameter description here>

Input Image 2 [raster] <put parameter description here>

Available RAM (Mb) [number] <put parameter description here>

Default: *128*

Outputs

Change Map [raster] <put output description here>

Console usage

```
processing.runalg('otb:multivariatealterationdetector', -in1, -in2, -ram, -out)
```

See also

Radiometric Indices

Description

<put algorithm description here>

Parameters

Input Image [raster] <put parameter description here>

Available RAM (Mb) [number] <put parameter description here>

Default: *128*

Blue Channel [number] <put parameter description here>

Default: *1*

Green Channel [number] <put parameter description here>

Default: *1*

Red Channel [number] <put parameter description here>

Default: *1*

NIR Channel [number] <put parameter description here>

Default: *1*

Mir Channel [number] <put parameter description here>

Default: *1*

Available Radiometric Indices [selection] <put parameter description here>

Options:

- 0 — ndvi
- 1 — tndvi
- 2 — rvi

- 3 — savi
- 4 — tsavi
- 5 — msavi
- 6 — msavi2
- 7 — gemi
- 8 — ipvi
- 9 — ndwi
- 10 — ndwi2
- 11 — mndwi
- 12 — ndpi
- 13 — ndti
- 14 — ri
- 15 — ci
- 16 — bi
- 17 — bi2

Default: 0

Outputs

Output Image [raster] <put output description here>

Console usage

```
processing.runalg('otb:radiometricindices', -in, -ram, -channels.blue, -channels.green, -channels
```

See also

.

18.4.3 Geometry

Image Envelope

Description

<put algorithm description here>

Parameters

Input Image [raster] <put parameter description here>

Sampling Rate [number] <put parameter description here>

Default: 0

Projection [string] Optional.

<put parameter description here>

Default: *None*

Outputs

Output Vector Data [vector] <put output description here>

Console usage

```
processing.runalg('otb:imageenvelope', -in, -sr, -proj, -out)
```

See also

OrthoRectification (epsg)

Description

<put algorithm description here>

Parameters

Input Image [raster] <put parameter description here>

Output Cartographic Map Projection [selection] <put parameter description here>

Options:

- 0 — epsg

Default: *0*

EPSG Code [number] <put parameter description here>

Default: *4326*

Parameters estimation modes [selection] <put parameter description here>

Options:

- 0 — autosize
- 1 — autospacing

Default: *0*

Default pixel value [number] <put parameter description here>

Default: *0*

Default elevation [number] <put parameter description here>

Default: *0*

Interpolation [selection] <put parameter description here>

Options:

- 0 — bco
- 1 — nn
- 2 — linear

Default: 0

Radius for bicubic interpolation [number] <put parameter description here>

Default: 2

Available RAM (Mb) [number] <put parameter description here>

Default: 128

Resampling grid spacing [number] <put parameter description here>

Default: 4

Outputs

Output Image [raster] <put output description here>

Console usage

```
processing.runalg('otb:orthorectificationeps', -io.in, -map, -map.epsg.code, -outputs.mode, -outp
```

See also

OrthoRectification (fit-to-ortho)

Description

<put algorithm description here>

Parameters

Input Image [raster] <put parameter description here>

Parameters estimation modes [selection] <put parameter description here>

Options:

- 0 — orthofit

Default: 0

Model ortho-image [raster] Optional.

<put parameter description here>

Default pixel value [number] <put parameter description here>

Default: 0

Default elevation [number] <put parameter description here>

Default: 0

Interpolation [selection] <put parameter description here>

Options:

- 0 — bco
- 1 — nn
- 2 — linear

Default: 0

Radius for bicubic interpolation [number] <put parameter description here>

Default: 2

Available RAM (Mb) [number] <put parameter description here>

Default: 128

Resampling grid spacing [number] <put parameter description here>

Default: 4

Outputs

Output Image [raster] <put output description here>

Console usage

```
processing.runalg('otb:orthorectificationfittoortho', -io.in, -outputs.mode, -outputs.ortho, -outp
```

See also

OrthoRectification (lambert-WGS84)

Description

<put algorithm description here>

Parameters

Input Image [raster] <put parameter description here>

Output Cartographic Map Projection [selection] <put parameter description here>

Options:

- 0 — lambert2
- 1 — lambert93
- 2 — wgs

Default: 0

Parameters estimation modes [selection] <put parameter description here>

Options:

- 0 — autosize
- 1 — autospacing

Default: 0

Default pixel value [number] <put parameter description here>

Default: 0

Default elevation [number] <put parameter description here>

Default: 0

Interpolation [selection] <put parameter description here>

Options:

- 0 — bco
- 1 — nn
- 2 — linear

Default: 0

Radius for bicubic interpolation [number] <put parameter description here>

Default: 2

Available RAM (Mb) [number] <put parameter description here>

Default: 128

Resampling grid spacing [number] <put parameter description here>

Default: 4

Outputs

Output Image [raster] <put output description here>

Console usage

```
processing.runalg('otb:orthorectificationlambertwgs84', -io.in, -map, -outputs.mode, -outputs.def
```

See also

OrthoRectification (utm)

Description

<put algorithm description here>

Parameters

Input Image [raster] <put parameter description here>

Output Cartographic Map Projection [selection] <put parameter description here>

Options:

- 0 — utm

Default: 0

Zone number [number] <put parameter description here>

Default: 31

Northern Hemisphere [boolean] <put parameter description here>

Default: *True*

Parameters estimation modes [selection] <put parameter description here>

Options:

- 0 — autosize
- 1 — autospacing

Default: 0

Default pixel value [number] <put parameter description here>

Default: 0

Default elevation [number] <put parameter description here>

Default: 0

Interpolation [selection] <put parameter description here>

Options:

- 0 — bco
- 1 — nn
- 2 — linear

Default: 0

Radius for bicubic interpolation [number] <put parameter description here>

Default: 2

Available RAM (Mb) [number] <put parameter description here>

Default: 128

Resampling grid spacing [number] <put parameter description here>

Default: 4

Outputs

Output Image [raster] <put output description here>

Console usage

```
processing.runalg('otb:orthorectificationutm', -io.in, -map, -map.utm.zone, -map.utm.northhem, -o
```

See also

Pansharpening (bayes)

Description

<put algorithm description here>

Parameters

Input PAN Image [raster] <put parameter description here>

Input XS Image [raster] <put parameter description here>

Algorithm [selection] <put parameter description here>

Options:

- 0 — bayes

Default: 0

Weight [number] <put parameter description here>

Default: 0.9999

S coefficient [number] <put parameter description here>

Default: 1

Available RAM (Mb) [number] <put parameter description here>

Default: 128

Outputs

Output image [raster] <put output description here>

Console usage

```
processing.runalg('otb:pansharpeningbayes', -inp, -inxs, -method, -method.bayes.lambda, -method.b
```

See also

Pansharpening (lmvm)

Description

<put algorithm description here>

Parameters

Input PAN Image [raster] <put parameter description here>

Input XS Image [raster] <put parameter description here>

Algorithm [selection] <put parameter description here>

Options:

- 0 — lmvm

Default: 0

X radius [number] <put parameter description here>

Default: 3

Y radius [number] <put parameter description here>

Default: 3

Available RAM (Mb) [number] <put parameter description here>

Default: 128

Outputs

Output image [raster] <put output description here>

Console usage

```
processing.runalg('otb:pansharpeninglmvm', -inp, -inxs, -method, -method.lmvm.radiusx, -method.lm
```

See also

Pansharpening (rcs)

Description

<put algorithm description here>

Parameters

Input PAN Image [raster] <put parameter description here>

Input XS Image [raster] <put parameter description here>

Algorithm [selection] <put parameter description here>

Options:

- 0 — rcs

Default: 0

Available RAM (Mb) [number] <put parameter description here>

Default: 128

Outputs

Output image [raster] <put output description here>

Console usage

```
processing.runalg('otb:pansharpeningrcs', -inp, -inxs, -method, -ram, -out)
```

See also

RigidTransformResample (id)

Description

<put algorithm description here>

Parameters

Input image [raster] <put parameter description here>

Type of transformation [selection] <put parameter description here>

Options:

- 0 — id

Default: 0

X scaling [number] <put parameter description here>

Default: 1

Y scaling [number] <put parameter description here>

Default: 1

Interpolation [selection] <put parameter description here>

Options:

- 0 — nn
- 1 — linear
- 2 — bco

Default: 2

Radius for bicubic interpolation [number] <put parameter description here>

Default: 2

Available RAM (Mb) [number] <put parameter description here>

Default: 128

Outputs

Output image [raster] <put output description here>

Console usage

```
processing.runalg('otb:rigidtransformresampleid', -in, -transform.type, -transform.type.id.scalex
```

See also

RigidTransformResample (rotation)

Description

<put algorithm description here>

Parameters

Input image [raster] <put parameter description here>

Type of transformation [selection] <put parameter description here>

Options:

- 0 — rotation

Default: 0

Rotation angle [number] <put parameter description here>

Default: 0

X scaling [number] <put parameter description here>

Default: 1

Y scaling [number] <put parameter description here>

Default: 1

Interpolation [selection] <put parameter description here>

Options:

- 0 — nn
- 1 — linear
- 2 — bco

Default: 2

Radius for bicubic interpolation [number] <put parameter description here>

Default: 2

Available RAM (Mb) [number] <put parameter description here>

Default: 128

Outputs

Output image [raster] <put output description here>

Console usage

```
processing.runalg('otb:rigidtransformresamplerotation', -in, -transform.type, -transform.type.rot
```

See also

RigidTransformResample (translation)

Description

<put algorithm description here>

Parameters

Input image [raster] <put parameter description here>

Type of transformation [selection] <put parameter description here>

Options:

- 0 — translation

Default: 0

The X translation (in physical units) [number] <put parameter description here>

Default: 0

The Y translation (in physical units) [number] <put parameter description here>

Default: 0

X scaling [number] <put parameter description here>

Default: 1

Y scaling [number] <put parameter description here>

Default: 1

Interpolation [selection] <put parameter description here>

Options:

- 0 — nn
- 1 — linear
- 2 — bco

Default: 2

Radius for bicubic interpolation [number] <put parameter description here>

Default: 2

Available RAM (Mb) [number] <put parameter description here>

Default: 128

Outputs

Output image [raster] <put output description here>

Console usage

```
processing.runalg('otb:rigidtransformresampletranslation', -in, -transform.type, -transform.type.
```

See also

Superimpose sensor

Description

<put algorithm description here>

Parameters

Reference input [raster] <put parameter description here>

The image to reproject [raster] <put parameter description here>

Default elevation [number] <put parameter description here>

Default: 0

Spacing of the deformation field [number] <put parameter description here>

Default: 4

Interpolation [selection] <put parameter description here>

Options:

- 0 — bco
- 1 — nn
- 2 — linear

Default: 0

Radius for bicubic interpolation [number] <put parameter description here>

Default: 2

Available RAM (Mb) [number] <put parameter description here>

Default: *128*

Outputs

Output image [raster] <put output description here>

Console usage

```
processing.runalg('otb:superimposesensor', -inr, -inm, -elev.default, -lms, -interpolator, -inter
```

See also

.

18.4.4 Image filtering

DimensionalityReduction (ica)

Description

<put algorithm description here>

Parameters

Input Image [raster] <put parameter description here>

Algorithm [selection] <put parameter description here>

Options:

- 0 — ica

Default: *0*

number of iterations [number] <put parameter description here>

Default: *20*

Give the increment weight of W in [0, 1] [number] <put parameter description here>

Default: *1*

Number of Components [number] <put parameter description here>

Default: *0*

Normalize [boolean] <put parameter description here>

Default: *True*

Outputs

Output Image [raster] <put output description here>

“Inverse Output Image“ [raster] <put output description here>

Transformation matrix output [file] <put output description here>

Console usage

```
processing.runalg('otb:dimensionalityreductionica', -in, -method, -method.ica.iter, -method.ica.m
```

See also

DimensionalityReduction (maf)

Description

<put algorithm description here>

Parameters

Input Image [raster] <put parameter description here>

Algorithm [selection] <put parameter description here>

Options:

- 0 — maf

Default: 0

Number of Components. [number] <put parameter description here>

Default: 0

Normalize. [boolean] <put parameter description here>

Default: *True*

Outputs

Output Image [raster] <put output description here>

Transformation matrix output [file] <put output description here>

Console usage

```
processing.runalg('otb:dimensionalityreductionmaf', -in, -method, -nbcomp, -normalize, -out, -out
```

See also

DimensionalityReduction (napca)

Description

<put algorithm description here>

Parameters

Input Image [raster] <put parameter description here>

Algorithm [selection] <put parameter description here>

Options:

- 0 — napca

Default: 0

Set the x radius of the sliding window. [number] <put parameter description here>

Default: 1

Set the y radius of the sliding window. [number] <put parameter description here>

Default: 1

Number of Components. [number] <put parameter description here>

Default: 0

Normalize. [boolean] <put parameter description here>

Default: *True*

Outputs

Output Image [raster] <put output description here>

“**Inverse Output Image**“ [raster] <put output description here>

Transformation matrix output [file] <put output description here>

Console usage

```
processing.runalg('otb:dimensionalityreductionnapca', -in, -method, -method.napca.radiusx, -method
```

See also

DimensionalityReduction (pca)

Description

<put algorithm description here>

Parameters

Input Image [raster] <put parameter description here>

Algorithm [selection] <put parameter description here>

Options:

- 0 — pca

Default: 0

Number of Components. [number] <put parameter description here>

Default: 0

Normalize. [boolean] <put parameter description here>

Default: *True*

Outputs

Output Image [raster] <put output description here>

“ **Inverse Output Image**“ [raster] <put output description here>

Transformation matrix output [file] <put output description here>

Console usage

```
processing.runalg('otb:dimensionalityreductionpca', -in, -method, -nbcomp, -normalize, -out, -out.
```

See also

Mean Shift filtering (can be used as Exact Large-Scale Mean-Shift segmentation, step 1)

Description

<put algorithm description here>

Parameters

Input Image [raster] <put parameter description here>

Spatial radius [number] <put parameter description here>

Default: *5*

Range radius [number] <put parameter description here>

Default: *15*

Mode convergence threshold [number] <put parameter description here>

Default: *0.1*

Maximum number of iterations [number] <put parameter description here>

Default: *100*

Range radius coefficient [number] <put parameter description here>

Default: *0*

Mode search. [boolean] <put parameter description here>

Default: *True*

Outputs

Filtered output [raster] <put output description here>

Spatial image [raster] <put output description here>

Console usage

```
processing.runalg('otb:meanshiftfilteringcanbeusedasexactlargescalemeanshiftsegmentationstep1', -
```

See also

Smoothing (anidif)

Description

<put algorithm description here>

Parameters

Input Image [raster] <put parameter description here>

Available RAM (Mb) [number] <put parameter description here>

Default: *128*

Smoothing Type [selection] <put parameter description here>

Options:

- 0 — anidif

Default: *2*

Time Step [number] <put parameter description here>

Default: *0.125*

Nb Iterations [number] <put parameter description here>

Default: *10*

Outputs

Output Image [raster] <put output description here>

Console usage

```
processing.runalg('otb:smoothinganidif', -in, -ram, -type, -type.anidif.timestep, -type.anidif.nb
```

See also

Smoothing (gaussian)

Description

<put algorithm description here>

Parameters

Input Image [raster] <put parameter description here>

Available RAM (Mb) [number] <put parameter description here>

Default: 128

Smoothing Type [selection] <put parameter description here>

Options:

- 0 — gaussian

Default: 2

Radius [number] <put parameter description here>

Default: 2

Outputs

Output Image [raster] <put output description here>

Console usage

```
processing.runalg('otb:smoothinggaussian', -in, -ram, -type, -type.gaussian.radius, -out)
```

See also

Smoothing (mean)

Description

<put algorithm description here>

Parameters

Input Image [raster] <put parameter description here>

Available RAM (Mb) [number] <put parameter description here>

Default: 128

Smoothing Type [selection] <put parameter description here>

Options:

- 0 — mean

Default: 2

Radius [number] <put parameter description here>

Default: 2

Outputs

Output Image [raster] <put output description here>

Console usage

```
processing.runalg('otb:smoothingmean', -in, -ram, -type, -type.mean.radius, -out)
```

See also

.

18.4.5 Image manipulation

ColorMapping (continuous)

Description

<put algorithm description here>

Parameters

Input Image [raster] <put parameter description here>

Available RAM (Mb) [number] <put parameter description here>

Default: *128*

Operation [selection] <put parameter description here>

Options:

- 0 — labeltocolour

Default: *0*

Color mapping method [selection] <put parameter description here>

Options:

- 0 — continuous

Default: *0*

Look-up tables [selection] <put parameter description here>

Options:

- 0 — red
- 1 — green
- 2 — blue
- 3 — grey
- 4 — hot
- 5 — cool
- 6 — spring
- 7 — summer
- 8 — autumn
- 9 — winter
- 10 — copper

- 11 — jet
- 12 — hsv
- 13 — overunder
- 14 — relief

Default: 0

Mapping range lower value [number] <put parameter description here>

Default: 0

Mapping range higher value [number] <put parameter description here>

Default: 255

Outputs

Output Image [raster] <put output description here>

Console usage

```
processing.runalg('otb:colormappingcontinuous', -in, -ram, -op, -method, -method.continuous.lut, ...
```

See also

ColorMapping (custom)

Description

<put algorithm description here>

Parameters

Input Image [raster] <put parameter description here>

Available RAM (Mb) [number] <put parameter description here>

Default: 128

Operation [selection] <put parameter description here>

Options:

- 0 — labeltcolor

Default: 0

Color mapping method [selection] <put parameter description here>

Options:

- 0 — custom

Default: 0

Look-up table file [file] <put parameter description here>

Outputs

Output Image [raster] <put output description here>

Console usage

```
processing.runalg('otb:colormappingcustom', -in, -ram, -op, -method, -method.custom.lut, -out)
```

See also

ColorMapping (image)

Description

<put algorithm description here>

Parameters

Input Image [raster] <put parameter description here>

Available RAM (Mb) [number] <put parameter description here>

Default: *128*

Operation [selection] <put parameter description here>

Options:

- 0 — labeltocolor

Default: *0*

Color mapping method [selection] <put parameter description here>

Options:

- 0 — image

Default: *0*

Support Image [raster] <put parameter description here>

NoData value [number] <put parameter description here>

Default: *0*

lower quantile [number] <put parameter description here>

Default: *2*

upper quantile [number] <put parameter description here>

Default: *2*

Outputs

Output Image [raster] <put output description here>

Console usage

```
processing.runalg('otb:colormappingimage', -in, -ram, -op, -method, -method.image.in, -method.ima
```

See also

ColorMapping (optimal)

Description

<put algorithm description here>

Parameters

Input Image [raster] <put parameter description here>

Available RAM (Mb) [number] <put parameter description here>

Default: *128*

Operation [selection] <put parameter description here>

Options:

- 0 — labeltocolor

Default: *0*

Color mapping method [selection] <put parameter description here>

Options:

- 0 — optimal

Default: *0*

Background label [number] <put parameter description here>

Default: *0*

Outputs

Output Image [raster] <put output description here>

Console usage

```
processing.runalg('otb:colormappingoptimal', -in, -ram, -op, -method, -method.optimal.background,
```

See also

ExtractROI (fit)

Description

<put algorithm description here>

Parameters

Input Image [raster] <put parameter description here>

Available RAM (Mb) [number] <put parameter description here>

Default: 128

Extraction mode [selection] <put parameter description here>

Options:

- 0 — fit

Default: 0

Reference image [raster] <put parameter description here>

Default elevation [number] <put parameter description here>

Default: 0

Outputs

Output Image [raster] <put output description here>

Console usage

```
processing.runalg('otb:extractroi', -in, -ram, -mode, -mode.fit.ref, -mode.fit.elev.default, -
```

See also

ExtractROI (standard)

Description

<put algorithm description here>

Parameters

Input Image [raster] <put parameter description here>

Available RAM (Mb) [number] <put parameter description here>

Default: 128

Extraction mode [selection] <put parameter description here>

Options:

- 0 — standard

Default: 0

Start X [number] <put parameter description here>

Default: 0

Start Y [number] <put parameter description here>

Default: 0

Size X [number] <put parameter description here>

Default: 0

Size Y [number] <put parameter description here>

Default: 0

Outputs

Output Image [raster] <put output description here>

Console usage

```
processing.runalg('otb:extractroistandard', -in, -ram, -mode, -startx, -starty, -sizex, -sizey, -o
```

See also

Images Concatenation

Description

<put algorithm description here>

Parameters

Input images list [multipleinput: rasters] <put parameter description here>

Available RAM (Mb) [number] <put parameter description here>

Default: 128

Outputs

Output Image [raster] <put output description here>

Console usage

```
processing.runalg('otb:imagesconcatenation', -il, -ram, -out)
```

See also

Image Tile Fusion

Description

<put algorithm description here>

Parameters

Input Tile Images [**multipleinput: rasters**] <put parameter description here>

Number of tile columns [**number**] <put parameter description here>

Default: *0*

Number of tile rows [**number**] <put parameter description here>

Default: *0*

Outputs

Output Image [**raster**] <put output description here>

Console usage

```
processing.runalg('otb:imagetilefusion', -il, -cols, -rows, -out)
```

See also

Read image information

Description

<put algorithm description here>

Parameters

Input Image [**raster**] <put parameter description here>

Display the OSSIM keywordlist [**boolean**] <put parameter description here>

Default: *True*

GCPs Id [**string**] <put parameter description here>

Default: *None*

GCPs Info [**string**] <put parameter description here>

Default: *None*

GCPs Image Coordinates [**string**] <put parameter description here>

Default: *None*

GCPs Geographic Coordinates [**string**] <put parameter description here>

Default: *None*

Outputs

Console usage

```
processing.runalg('otb:readimageinformation', -in, -keywordlist, -gcp.ids, -gcp.info, -gcp.imcoord
```

See also

Rescale Image

Description

<put algorithm description here>

Parameters

Input Image [raster] <put parameter description here>

Available RAM (Mb) [number] <put parameter description here>

Default: 128

Output min value [number] <put parameter description here>

Default: 0

Output max value [number] <put parameter description here>

Default: 255

Outputs

Output Image [raster] <put output description here>

Console usage

```
processing.runalg('otb:rescaleimage', -in, -ram, -outmin, -outmax, -out)
```

See also

Split Image

Description

<put algorithm description here>

Parameters

Input Image [raster] <put parameter description here>

Available RAM (Mb) [number] <put parameter description here>

Default: 128

Outputs

Output Image [file] <put output description here>

Console usage

```
processing.runalg('otb:splitimage', -in, -ram, -out)
```

See also

.

18.4.6 Learning

Classification Map Regularization

Description

<put algorithm description here>

Parameters

Input classification image [raster] <put parameter description here>

Structuring element radius (in pixels) [number] <put parameter description here>

Default: *1*

Multiple majority: Undecided(X)/Original [boolean] <put parameter description here>

Default: *True*

Label for the NoData class [number] <put parameter description here>

Default: *0*

Label for the Undecided class [number] <put parameter description here>

Default: *0*

Available RAM (Mb) [number] <put parameter description here>

Default: *128*

Outputs

Output regularized image [raster] <put output description here>

Console usage

```
processing.runalg('otb:classificationmapregularization', -io.in, -ip.radius, -ip.suvbool, -ip.nod
```

See also

ComputeConfusionMatrix (raster)

Description

<put algorithm description here>

Parameters

Input Image [raster] <put parameter description here>

Ground truth [selection] <put parameter description here>

Options:

- 0 — raster

Default: 0

Input reference image [raster] <put parameter description here>

Value for nodata pixels [number] <put parameter description here>

Default: 0

Available RAM (Mb) [number] <put parameter description here>

Default: 128

Outputs

Matrix output [file] <put output description here>

Console usage

```
processing.runalg('otb:computeconfusionmatrixraster', -in, -ref, -ref.raster.in, -nodatalabel, -r
```

See also

ComputeConfusionMatrix (vector)

Description

<put algorithm description here>

Parameters

Input Image [raster] <put parameter description here>

Ground truth [selection] <put parameter description here>

Options:

- 0 — vector

Default: 0

Input reference vector data [file] <put parameter description here>

Field name [string] Optional.

<put parameter description here>

Default: *Class*

Value for nodata pixels [number] <put parameter description here>

Default: 0

Available RAM (Mb) [number] <put parameter description here>

Default: *128*

Outputs

Matrix output [file] <put output description here>

Console usage

```
processing.runalg('otb:computeconfusionmatrixvector', -in, -ref, -ref.vector.in, -ref.vector.field)
```

See also

Compute Images second order statistics

Description

<put algorithm description here>

Parameters

Input images [multipleinput: rasters] <put parameter description here>

Background Value [number] <put parameter description here>

Default: *0.0*

Outputs

Output XML file [file] <put output description here>

Console usage

```
processing.runalg('otb:computeimagessecondorderstatistics', -il, -bv, -out)
```

See also

FusionOfClassifications (dempstershafer)

Description

<put algorithm description here>

Parameters

Input classifications [multipleinput: rasters] <put parameter description here>

Fusion method [selection] <put parameter description here>

Options:

- 0 — dempstershafer

Default: 0

Confusion Matrices [**multipleinput: files**] <put parameter description here>

Mass of belief measurement [**selection**] <put parameter description here>

Options:

- 0 — precision
- 1 — recall
- 2 — accuracy
- 3 — kappa

Default: 0

Label for the NoData class [**number**] <put parameter description here>

Default: 0

Label for the Undecided class [**number**] <put parameter description here>

Default: 0

Outputs

The output classification image [**raster**] <put output description here>

Console usage

```
processing.runalg('otb:fusionofclassificationsdempstershafer', -il, -method, -method.dempstershafer)
```

See also

FusionOfClassifications (majorityvoting)

Description

<put algorithm description here>

Parameters

Input classifications [**multipleinput: rasters**] <put parameter description here>

Fusion method [**selection**] <put parameter description here>

Options:

- 0 — majorityvoting

Default: 0

Label for the NoData class [**number**] <put parameter description here>

Default: 0

Label for the Undecided class [**number**] <put parameter description here>

Default: 0

Outputs

The output classification image [raster] <put output description here>

Console usage

```
processing.runalg('otb:fusionofclassificationsmajorityvoting', -il, -method, -nodatalabel, -undec
```

See also

Image Classification

Description

<put algorithm description here>

Parameters

Input Image [raster] <put parameter description here>

Input Mask [raster] Optional.

<put parameter description here>

Model file [file] <put parameter description here>

Statistics file [file] Optional.

<put parameter description here>

Available RAM (Mb) [number] <put parameter description here>

Default: 128

Outputs

Output Image [raster] <put output description here>

Console usage

```
processing.runalg('otb:imageclassification', -in, -mask, -model, -imstat, -ram, -out)
```

See also

SOM Classification

Description

<put algorithm description here>

Parameters

InputImage [raster] <put parameter description here>

ValidityMask [raster] Optional.

<put parameter description here>

TrainingProbability [number] <put parameter description here>

Default: 1

TrainingSetSize [number] <put parameter description here>

Default: 0

StreamingLines [number] <put parameter description here>

Default: 0

SizeX [number] <put parameter description here>

Default: 32

SizeY [number] <put parameter description here>

Default: 32

NeighborhoodX [number] <put parameter description here>

Default: 10

NeighborhoodY [number] <put parameter description here>

Default: 10

NumberIteration [number] <put parameter description here>

Default: 5

BetaInit [number] <put parameter description here>

Default: 1

BetaFinal [number] <put parameter description here>

Default: 0.1

InitialValue [number] <put parameter description here>

Default: 0

Available RAM (Mb) [number] <put parameter description here>

Default: 128

set user defined seed [number] <put parameter description here>

Default: 0

Outputs

OutputImage [raster] <put output description here>

SOM Map [raster] <put output description here>

Console usage

```
processing.runalg('otb:somclassification', -in, -vm, -tp, -ts, -sl, -sx, -sy, -nx, -ny, -ni, -bi,
```

See also

TrainImagesClassifier (ann)

Description

<put algorithm description here>

Parameters

Input Image List [**multipleinput: rasters**] <put parameter description here>

Input Vector Data List [**multipleinput: any vectors**] <put parameter description here>

Input XML image statistics file [**file**] Optional.

<put parameter description here>

Default elevation [**number**] <put parameter description here>

Default: 0

Maximum training sample size per class [**number**] <put parameter description here>

Default: 1000

Maximum validation sample size per class [**number**] <put parameter description here>

Default: 1000

On edge pixel inclusion [**boolean**] <put parameter description here>

Default: True

Training and validation sample ratio [**number**] <put parameter description here>

Default: 0.5

Name of the discrimination field [**string**] <put parameter description here>

Default: Class

Classifier to use for the training [**selection**] <put parameter description here>

Options:

- 0 — ann

Default: 0

Train Method Type [**selection**] <put parameter description here>

Options:

- 0 — reg
- 1 — back

Default: 0

Number of neurons in each intermediate layer [**string**] <put parameter description here>

Default: None

Neuron activation function type [**selection**] <put parameter description here>

Options:

- 0 — ident
- 1 — sig

- 2 — gau

Default: 1

Alpha parameter of the activation function [number] <put parameter description here>

Default: 1

Beta parameter of the activation function [number] <put parameter description here>

Default: 1

Strength of the weight gradient term in the BACKPROP method [number] <put parameter description here>

Default: 0.1

Strength of the momentum term (the difference between weights on the 2 previous iterations) [number] <put parameter description here>

Default: 0.1

Initial value Delta_0 of update-values Delta_{ij} in RPROP method [number] <put parameter description here>

Default: 0.1

Update-values lower limit Delta_{min} in RPROP method [number] <put parameter description here>

Default: 1e-07

Termination criteria [selection] <put parameter description here>

Options:

- 0 — iter
- 1 — eps
- 2 — all

Default: 2

Epsilon value used in the Termination criteria [number] <put parameter description here>

Default: 0.01

Maximum number of iterations used in the Termination criteria [number] <put parameter description here>

Default: 1000

set user defined seed [number] <put parameter description here>

Default: 0

Outputs

Output confusion matrix [file] <put output description here>

Output model [file] <put output description here>

Console usage

```
processing.runalg('otb:trainimagesclassifierann', -io.il, -io.vd, -io.imstat, -elev.default, -sam
```

See also

TrainImagesClassifier (bayes)

Description

<put algorithm description here>

Parameters

Input Image List [**multipleinput: rasters**] <put parameter description here>

Input Vector Data List [**multipleinput: any vectors**] <put parameter description here>

Input XML image statistics file [**file**] Optional.

<put parameter description here>

Default elevation [**number**] <put parameter description here>

Default: 0

Maximum training sample size per class [**number**] <put parameter description here>

Default: 1000

Maximum validation sample size per class [**number**] <put parameter description here>

Default: 1000

On edge pixel inclusion [**boolean**] <put parameter description here>

Default: True

Training and validation sample ratio [**number**] <put parameter description here>

Default: 0.5

Name of the discrimination field [**string**] <put parameter description here>

Default: Class

Classifier to use for the training [**selection**] <put parameter description here>

Options:

- 0 — bayes

Default: 0

set user defined seed [**number**] <put parameter description here>

Default: 0

Outputs

Output confusion matrix [**file**] <put output description here>

Output model [**file**] <put output description here>

Console usage

```
processing.runalg('otb:trainimagesclassifierbayes', -io.il, -io.vd, -io.imstat, -elev.default, -s
```

See also

TrainImagesClassifier (boost)

Description

<put algorithm description here>

Parameters

Input Image List [**multipleinput: rasters**] <put parameter description here>

Input Vector Data List [**multipleinput: any vectors**] <put parameter description here>

Input XML image statistics file [**file**] Optional.

<put parameter description here>

Default elevation [**number**] <put parameter description here>

Default: 0

Maximum training sample size per class [**number**] <put parameter description here>

Default: 1000

Maximum validation sample size per class [**number**] <put parameter description here>

Default: 1000

On edge pixel inclusion [**boolean**] <put parameter description here>

Default: True

Training and validation sample ratio [**number**] <put parameter description here>

Default: 0.5

Name of the discrimination field [**string**] <put parameter description here>

Default: Class

Classifier to use for the training [**selection**] <put parameter description here>

Options:

- 0 — boost

Default: 0

Boost Type [**selection**] <put parameter description here>

Options:

- 0 — discrete
- 1 — real
- 2 — logit
- 3 — gentle

Default: 1

Weak count [**number**] <put parameter description here>

Default: 100

Weight Trim Rate [**number**] <put parameter description here>

Default: 0.95

Maximum depth of the tree [number] <put parameter description here>

Default: *1*

set user defined seed [number] <put parameter description here>

Default: *0*

Outputs

Output confusion matrix [file] <put output description here>

Output model [file] <put output description here>

Console usage

```
processing.runalg('otb:trainimagesclassifierboost', -io.il, -io.vd, -io.imstat, -elev.default, -s
```

See also

TrainImagesClassifier (dt)

Description

<put algorithm description here>

Parameters

Input Image List [multipleinput: rasters] <put parameter description here>

Input Vector Data List [multipleinput: any vectors] <put parameter description here>

Input XML image statistics file [file] Optional.

<put parameter description here>

Default elevation [number] <put parameter description here>

Default: *0*

Maximum training sample size per class [number] <put parameter description here>

Default: *1000*

Maximum validation sample size per class [number] <put parameter description here>

Default: *1000*

On edge pixel inclusion [boolean] <put parameter description here>

Default: *True*

Training and validation sample ratio [number] <put parameter description here>

Default: *0.5*

Name of the discrimination field [string] <put parameter description here>

Default: *Class*

Classifier to use for the training [selection] <put parameter description here>

Options:

- 0 — dt

Default: 0

Maximum depth of the tree [number] <put parameter description here>

Default: 65535

Minimum number of samples in each node [number] <put parameter description here>

Default: 10

Termination criteria for regression tree [number] <put parameter description here>

Default: 0.01

Cluster possible values of a categorical variable into K <= cat clusters to find a suboptimal
<put parameter description here>

Default: 10

K-fold cross-validations [number] <put parameter description here>

Default: 10

Set UseSelfRule flag to false [boolean] <put parameter description here>

Default: True

Set TruncatePrunedTree flag to false [boolean] <put parameter description here>

Default: True

set user defined seed [number] <put parameter description here>

Default: 0

Outputs

Output confusion matrix [file] <put output description here>

Output model [file] <put output description here>

Console usage

```
processing.runalg('otb:trainimagesclassifierdt', -io.il, -io.vd, -io.imstat, -elev.default, -samp
```

See also

TrainImagesClassifier (gbt)

Description

<put algorithm description here>

Parameters

Input Image List [multipleinput: rasters] <put parameter description here>

Input Vector Data List [multipleinput: any vectors] <put parameter description here>

Input XML image statistics file [file] Optional.

<put parameter description here>

Default elevation [number] <put parameter description here>

Default: *0*

Maximum training sample size per class [number] <put parameter description here>

Default: *1000*

Maximum validation sample size per class [number] <put parameter description here>

Default: *1000*

On edge pixel inclusion [boolean] <put parameter description here>

Default: *True*

Training and validation sample ratio [number] <put parameter description here>

Default: *0.5*

Name of the discrimination field [string] <put parameter description here>

Default: *Class*

Classifier to use for the training [selection] <put parameter description here>

Options:

- 0 — gbt

Default: *0*

Number of boosting algorithm iterations [number] <put parameter description here>

Default: *200*

Regularization parameter [number] <put parameter description here>

Default: *0.01*

Portion of the whole training set used for each algorithm iteration [number]

<put parameter description here>

Default: *0.8*

Maximum depth of the tree [number] <put parameter description here>

Default: *3*

set user defined seed [number] <put parameter description here>

Default: *0*

Outputs

Output confusion matrix [file] <put output description here>

Output model [file] <put output description here>

Console usage

```
processing.runalg('otb:trainimagesclassifiergbt', -io.il, -io.vd, -io.imstat, -elev.default, -sam
```

See also

TrainImagesClassifier (knn)

Description

<put algorithm description here>

Parameters

Input Image List [**multipleinput: rasters**] <put parameter description here>

Input Vector Data List [**multipleinput: any vectors**] <put parameter description here>

Input XML image statistics file [**file**] Optional.

<put parameter description here>

Default elevation [**number**] <put parameter description here>

Default: 0

Maximum training sample size per class [**number**] <put parameter description here>

Default: 1000

Maximum validation sample size per class [**number**] <put parameter description here>

Default: 1000

On edge pixel inclusion [**boolean**] <put parameter description here>

Default: *True*

Training and validation sample ratio [**number**] <put parameter description here>

Default: 0.5

Name of the discrimination field [**string**] <put parameter description here>

Default: *Class*

Classifier to use for the training [**selection**] <put parameter description here>

Options:

- 0 — knn

Default: 0

Number of Neighbors [**number**] <put parameter description here>

Default: 32

set user defined seed [**number**] <put parameter description here>

Default: 0

Outputs

Output confusion matrix [**file**] <put output description here>

Output model [**file**] <put output description here>

Console usage

```
processing.runalg('otb:trainimagesclassifierknn', -io.il, -io.vd, -io.imstat, -elev.default, -sam
```

See also

TrainImagesClassifier (libsvm)

Description

<put algorithm description here>

Parameters

Input Image List [**multipleinput: rasters**] <put parameter description here>

Input Vector Data List [**multipleinput: any vectors**] <put parameter description here>

Input XML image statistics file [**file**] Optional.

<put parameter description here>

Default elevation [**number**] <put parameter description here>

Default: *0*

Maximum training sample size per class [**number**] <put parameter description here>

Default: *1000*

Maximum validation sample size per class [**number**] <put parameter description here>

Default: *1000*

On edge pixel inclusion [**boolean**] <put parameter description here>

Default: *True*

Training and validation sample ratio [**number**] <put parameter description here>

Default: *0.5*

Name of the discrimination field [**string**] <put parameter description here>

Default: *Class*

Classifier to use for the training [**selection**] <put parameter description here>

Options:

- 0 — libsvm

Default: *0*

SVM Kernel Type [**selection**] <put parameter description here>

Options:

- 0 — linear
- 1 — rbf
- 2 — poly
- 3 — sigmoid

Default: *0*

Cost parameter C [number] <put parameter description here>

Default: *1*

Parameters optimization [boolean] <put parameter description here>

Default: *True*

set user defined seed [number] <put parameter description here>

Default: *0*

Outputs

Output confusion matrix [file] <put output description here>

Output model [file] <put output description here>

Console usage

```
processing.runalg('otb:trainimagesclassifierlibsvm', -io.il, -io.vd, -io.imstat, -elev.default, -
```

See also

TrainImagesClassifier (rf)

Description

<put algorithm description here>

Parameters

Input Image List [multipleinput: rasters] <put parameter description here>

Input Vector Data List [multipleinput: any vectors] <put parameter description here>

Input XML image statistics file [file] Optional.

<put parameter description here>

Default elevation [number] <put parameter description here>

Default: *0*

Maximum training sample size per class [number] <put parameter description here>

Default: *1000*

Maximum validation sample size per class [number] <put parameter description here>

Default: *1000*

On edge pixel inclusion [boolean] <put parameter description here>

Default: *True*

Training and validation sample ratio [number] <put parameter description here>

Default: *0.5*

Name of the discrimination field [string] <put parameter description here>

Default: *Class*

Classifier to use for the training [selection] <put parameter description here>

Options:

- 0 — rf

Default: 0

Maximum depth of the tree [number] <put parameter description here>

Default: 5

Minimum number of samples in each node [number] <put parameter description here>

Default: 10

Termination Criteria for regression tree [number] <put parameter description here>

Default: 0

Cluster possible values of a categorical variable into K <= cat clusters to find a suboptimal <put parameter description here>

Default: 10

Size of the randomly selected subset of features at each tree node [number] <put parameter description here>

Default: 0

Maximum number of trees in the forest [number] <put parameter description here>

Default: 100

Sufficient accuracy (OOB error) [number] <put parameter description here>

Default: 0.01

set user defined seed [number] <put parameter description here>

Default: 0

Outputs

Output confusion matrix [file] <put output description here>

Output model [file] <put output description here>

Console usage

```
processing.runalg('otb:trainimagesclassifierrf', -io.il, -io.vd, -io.imstat, -elev.default, -samp
```

See also

TrainImagesClassifier (svm)

Description

<put algorithm description here>

Parameters

Input Image List [**multipleinput: rasters**] <put parameter description here>

Input Vector Data List [**multipleinput: any vectors**] <put parameter description here>

Input XML image statistics file [**file**] Optional.

<put parameter description here>

Default elevation [**number**] <put parameter description here>

Default: *0*

Maximum training sample size per class [**number**] <put parameter description here>

Default: *1000*

Maximum validation sample size per class [**number**] <put parameter description here>

Default: *1000*

On edge pixel inclusion [**boolean**] <put parameter description here>

Default: *True*

Training and validation sample ratio [**number**] <put parameter description here>

Default: *0.5*

Name of the discrimination field [**string**] <put parameter description here>

Default: *Class*

Classifier to use for the training [**selection**] <put parameter description here>

Options:

- 0 — svm

Default: *0*

SVM Model Type [**selection**] <put parameter description here>

Options:

- 0 — csvc
- 1 — nusvc
- 2 — oneclass

Default: *0*

SVM Kernel Type [**selection**] <put parameter description here>

Options:

- 0 — linear
- 1 — rbf
- 2 — poly
- 3 — sigmoid

Default: *0*

Cost parameter C [**number**] <put parameter description here>

Default: *1*

Parameter nu of a SVM optimization problem (NU_SVC / ONE_CLASS) [**number**] <put parameter description here>

Default: *0*

Parameter coef0 of a kernel function (POLY / SIGMOID) [number] <put parameter description here>

Default: 0

Parameter gamma of a kernel function (POLY / RBF / SIGMOID) [number] <put parameter description here>

Default: 1

Parameter degree of a kernel function (POLY) [number] <put parameter description here>

Default: 1

Parameters optimization [boolean] <put parameter description here>

Default: *True*

set user defined seed [number] <put parameter description here>

Default: 0

Outputs

Output confusion matrix [file] <put output description here>

Output model [file] <put output description here>

Console usage

```
processing.runalg('otb:trainimagesclassifiersvm', -io.il, -io.vd, -io.imstat, -elev.default, -sam
```

See also

Unsupervised KMeans image classification

Description

<put algorithm description here>

Parameters

Input Image [raster] <put parameter description here>

Available RAM (Mb) [number] <put parameter description here>

Default: 128

Validity Mask [raster] Optional.

<put parameter description here>

Training set size [number] <put parameter description here>

Default: 100

Number of classes [number] <put parameter description here>

Default: 5

Maximum number of iterations [number] <put parameter description here>

Default: 1000

Convergence threshold [number] <put parameter description here>

Default: *0.0001*

Outputs

Output Image [raster] <put output description here>

Centroid filename [file] <put output description here>

Console usage

```
processing.runalg('otb:unsupervisedkmeansimageclassification', -in, -ram, -vm, -ts, -nc, -maxit, ...
```

See also

.

18.4.7 Miscellaneous

Band Math

Description

<put algorithm description here>

Parameters

Input image list [multipleinput: rasters] <put parameter description here>

Available RAM (Mb) [number] <put parameter description here>

Default: *128*

Expression [string] <put parameter description here>

Default: *None*

Outputs

Output Image [raster] <put output description here>

Console usage

```
processing.runalg('otb:bandmath', -il, -ram, -exp, -out)
```

See also

ComputeModulusAndPhase-one (OneEntry)

Description

<put algorithm description here>

Parameters

Number Of inputs [selection] <put parameter description here>

Options:

- 0 — one

Default: 0

Input image [raster] <put parameter description here>

Available RAM (Mb) [number] <put parameter description here>

Default: 128

Outputs

Modulus [raster] <put output description here>

Phase [raster] <put output description here>

Console usage

```
processing.runalg('otb:computemodulusandphaseoneentry', -nbininput, -nbininput.one.in, -ram, -mod,
```

See also

ComputeModulusAndPhase-two (TwoEntries)

Description

<put algorithm description here>

Parameters

Number Of inputs [selection] <put parameter description here>

Options:

- 0 — two

Default: 0

Real part input [raster] <put parameter description here>

Imaginary part input [raster] <put parameter description here>

Available RAM (Mb) [number] <put parameter description here>

Default: 128

Outputs

Modulus [raster] <put output description here>

Phase [raster] <put output description here>

Console usage

```
processing.runalg('otb:computemodulusandphasetwoentries', -nbinput, -nbinput.two.re, -nbinput.t
```

See also

Images comparaison

Description

<put algortithm description here>

Parameters

Reference image [raster] <put parameter description here>

Reference image channel [number] <put parameter description here>

Default: *1*

Measured image [raster] <put parameter description here>

Measured image channel [number] <put parameter description here>

Default: *1*

Start X [number] <put parameter description here>

Default: *0*

Start Y [number] <put parameter description here>

Default: *0*

Size X [number] <put parameter description here>

Default: *0*

Size Y [number] <put parameter description here>

Default: *0*

Outputs

Console usage

```
processing.runalg('otb:imagescomparaison', -ref.in, -ref.channel, -meas.in, -meas.channel, -roi.st
```

See also

Image to KMZ Export

Description

<put algortithm description here>

Parameters

Input image [raster] <put parameter description here>

Tile Size [number] <put parameter description here>

Default: *512*

Image logo [raster] Optional.

<put parameter description here>

Image legend [raster] Optional.

<put parameter description here>

Default elevation [number] <put parameter description here>

Default: *0*

Outputs

Output .kmz product [file] <put output description here>

Console usage

```
processing.runalg('otb:imagetokmzexport', -in, -tilesize, -logo, -legend, -elev.default, -out)
```

See also

.

18.4.8 Segmentation

Connected Component Segmentation

Description

<put algorithm description here>

Parameters

Input Image [raster] <put parameter description here>

Mask expression [string] Optional.

<put parameter description here>

Default: *None*

Connected Component Expression [string] <put parameter description here>

Default: *None*

Minimum Object Size [number] <put parameter description here>

Default: *2*

OBIA Expression [string] Optional.

<put parameter description here>

Default: *None*

Default elevation [number] <put parameter description here>

Default: *0*

Outputs

Output Shape [vector] <put output description here>

Console usage

```
processing.runalg('otb:connectedcomponentsegmentation', -in, -mask, -expr, -minsize, -obia, -elev
```

See also

Exact Large-Scale Mean-Shift segmentation, step 2

Description

<put algorithm description here>

Parameters

Filtered image [raster] <put parameter description here>

Spatial image [raster] Optional.

<put parameter description here>

Range radius [number] <put parameter description here>

Default: *15*

Spatial radius [number] <put parameter description here>

Default: *5*

Minimum Region Size [number] <put parameter description here>

Default: *0*

Size of tiles in pixel (X-axis) [number] <put parameter description here>

Default: *500*

Size of tiles in pixel (Y-axis) [number] <put parameter description here>

Default: *500*

Directory where to write temporary files [file] Optional.

<put parameter description here>

Temporary files cleaning [boolean] <put parameter description here>

Default: *True*

Outputs

Output Image [raster] <put output description here>

Console usage

```
processing.runalg('otb:exactlargescalemeanshiftsegmentationstep2', -in, -inpos, -ranger, -spatial,
```

See also

Exact Large-Scale Mean-Shift segmentation, step 3 (optional)

Description

<put algorithm description here>

Parameters

Input image [raster] <put parameter description here>

Segmented image [raster] <put parameter description here>

Minimum Region Size [number] <put parameter description here>

Default: *50*

Size of tiles in pixel (X-axis) [number] <put parameter description here>

Default: *500*

Size of tiles in pixel (Y-axis) [number] <put parameter description here>

Default: *500*

Outputs

Output Image [raster] <put output description here>

Console usage

```
processing.runalg('otb:exactlargescalemeanshiftsegmentationstep3optional', -in, -inseg, -minsize,
```

See also

Exact Large-Scale Mean-Shift segmentation, step 4

Description

<put algorithm description here>

Parameters

Input Image [raster] <put parameter description here>

Segmented image [raster] <put parameter description here>

Size of tiles in pixel (X-axis) [number] <put parameter description here>

Default: *500*

Size of tiles in pixel (Y-axis) [number] <put parameter description here>

Default: *500*

Outputs

Output GIS vector file [vector] <put output description here>

Console usage

```
processing.runalg('otb:exactlargescalemeanshiftsegmentationstep4', -in, -inseg, -tilesizex, -tile
```

See also

Hoover compare segmentation

Description

<put algorithm description here>

Parameters

Input ground truth [raster] <put parameter description here>

Input machine segmentation [raster] <put parameter description here>

Background label [number] <put parameter description here>

Default: *0*

Overlapping threshold [number] <put parameter description here>

Default: *0.75*

Correct detection score [number] <put parameter description here>

Default: *0.0*

Over-segmentation score [number] <put parameter description here>

Default: *0.0*

Under-segmentation score [number] <put parameter description here>

Default: *0.0*

Missed detection score [number] <put parameter description here>

Default: *0.0*

Outputs

Colored ground truth output [raster] <put output description here>

Colored machine segmentation output [raster] <put output description here>

Console usage

```
processing.runalg('otb:hoovercomparesegmentation', -ingt, -inms, -bg, -th, -rc, -rf, -ra, -rm, -o
```

See also

Segmentation (cc)

Description

<put algorithm description here>

Parameters

Input Image [raster] <put parameter description here>

Segmentation algorithm [selection] <put parameter description here>

Options:

- 0 — cc

Default: 0

Condition [string] <put parameter description here>

Default: *None*

Processing mode [selection] <put parameter description here>

Options:

- 0 — vector

Default: 0

Writing mode for the output vector file [selection] <put parameter description here>

Options:

- 0 — ulco
- 1 — ovw
- 2 — ulovw
- 3 — ulu

Default: 0

Mask Image [raster] Optional.

<put parameter description here>

8-neighbor connectivity [boolean] <put parameter description here>

Default: *True*

Stitch polygons [**boolean**] <put parameter description here>

Default: *True*

Minimum object size [**number**] <put parameter description here>

Default: *1*

Simplify polygons [**number**] <put parameter description here>

Default: *0.1*

Layer name [**string**] <put parameter description here>

Default: *layer*

Geometry index field name [**string**] <put parameter description here>

Default: *DN*

Tiles size [**number**] <put parameter description here>

Default: *1024*

Starting geometry index [**number**] <put parameter description here>

Default: *1*

OGR options for layer creation [**string**] Optional.

<put parameter description here>

Default: *None*

Outputs

Output vector file [**vector**] <put output description here>

Console usage

```
processing.runalg('otb:segmentationcc', -in, -filter, -filter.cc.expr, -mode, -mode.vector.outmode)
```

See also

Segmentation (edison)

Description

<put algorithm description here>

Parameters

Input Image [**raster**] <put parameter description here>

Segmentation algorithm [**selection**] <put parameter description here>

Options:

- 0 — edison

Default: *0*

Spatial radius [**number**] <put parameter description here>

Default: *5*

Range radius [number] <put parameter description here>

Default: *15*

Minimum region size [number] <put parameter description here>

Default: *100*

Scale factor [number] <put parameter description here>

Default: *1*

Processing mode [selection] <put parameter description here>

Options:

- 0 — vector

Default: *0*

Writing mode for the output vector file [selection] <put parameter description here>

Options:

- 0 — ulco
- 1 — ovw
- 2 — ulovw
- 3 — ulu

Default: *0*

Mask Image [raster] Optional.

<put parameter description here>

8-neighbor connectivity [boolean] <put parameter description here>

Default: *True*

Stitch polygons [boolean] <put parameter description here>

Default: *True*

Minimum object size [number] <put parameter description here>

Default: *1*

Simplify polygons [number] <put parameter description here>

Default: *0.1*

Layer name [string] <put parameter description here>

Default: *layer*

Geometry index field name [string] <put parameter description here>

Default: *DN*

Tiles size [number] <put parameter description here>

Default: *1024*

Starting geometry index [number] <put parameter description here>

Default: *1*

OGR options for layer creation [string] Optional.

<put parameter description here>

Default: *None*

Outputs

Output vector file [vector] <put output description here>

Console usage

```
processing.runalg('otb:segmentationedison', -in, -filter, -filter.edison.spatialr, -filter.edison
```

See also

Segmentation (meanshift)

Description

<put algorithm description here>

Parameters

Input Image [raster] <put parameter description here>

Segmentation algorithm [selection] <put parameter description here>

Options:

- 0 — meanshift

Default: 0

Spatial radius [number] <put parameter description here>

Default: 5

Range radius [number] <put parameter description here>

Default: 15

Mode convergence threshold [number] <put parameter description here>

Default: 0.1

Maximum number of iterations [number] <put parameter description here>

Default: 100

Minimum region size [number] <put parameter description here>

Default: 100

Processing mode [selection] <put parameter description here>

Options:

- 0 — vector

Default: 0

Writing mode for the output vector file [selection] <put parameter description here>

Options:

- 0 — ulco
- 1 — ovw
- 2 — ulovw

- 3 — ulu

Default: *0*

Mask Image [raster] Optional.

<put parameter description here>

8-neighbor connectivity [boolean] <put parameter description here>

Default: *True*

Stitch polygons [boolean] <put parameter description here>

Default: *True*

Minimum object size [number] <put parameter description here>

Default: *1*

Simplify polygons [number] <put parameter description here>

Default: *0.1*

Layer name [string] <put parameter description here>

Default: *layer*

Geometry index field name [string] <put parameter description here>

Default: *DN*

Tiles size [number] <put parameter description here>

Default: *1024*

Starting geometry index [number] <put parameter description here>

Default: *1*

OGR options for layer creation [string] Optional.

<put parameter description here>

Default: *None*

Outputs

Output vector file [vector] <put output description here>

Console usage

```
processing.runalg('otb:segmentationmeanshift', -in, -filter, -filter.meanshift.spatialr, -filter.
```

See also

Segmentation (mprofiles)

Description

<put algorithm description here>

Parameters

Input Image [raster] <put parameter description here>

Segmentation algorithm [selection] <put parameter description here>

Options:

- 0 — mprofiles

Default: 0

Profile Size [number] <put parameter description here>

Default: 5

Initial radius [number] <put parameter description here>

Default: 1

Radius step. [number] <put parameter description here>

Default: 1

Threshold of the final decision rule [number] <put parameter description here>

Default: 1

Processing mode [selection] <put parameter description here>

Options:

- 0 — vector

Default: 0

Writing mode for the output vector file [selection] <put parameter description here>

Options:

- 0 — ulco
- 1 — ovw
- 2 — ulovw
- 3 — ulu

Default: 0

Mask Image [raster] Optional.

<put parameter description here>

8-neighbor connectivity [boolean] <put parameter description here>

Default: *True*

Stitch polygons [boolean] <put parameter description here>

Default: *True*

Minimum object size [number] <put parameter description here>

Default: 1

Simplify polygons [number] <put parameter description here>

Default: 0.1

Layer name [string] <put parameter description here>

Default: *layer*

Geometry index field name [string] <put parameter description here>

Default: *DN*

Tiles size [number] <put parameter description here>

Default: *1024*

Starting geometry index [number] <put parameter description here>

Default: *1*

OGR options for layer creation [string] Optional.

<put parameter description here>

Default: *None*

Outputs

Output vector file [vector] <put output description here>

Console usage

```
processing.runalg('otb:segmentationmprofiles', -in, -filter, -filter.mprofiles.size, -filter.mpro
```

See also

Segmentation (watershed)

Description

<put algorithm description here>

Parameters

Input Image [raster] <put parameter description here>

Segmentation algorithm [selection] <put parameter description here>

Options:

- 0 — watershed

Default: *0*

Depth Threshold [number] <put parameter description here>

Default: *0.01*

Flood Level [number] <put parameter description here>

Default: *0.1*

Processing mode [selection] <put parameter description here>

Options:

- 0 — vector

Default: *0*

Writing mode for the output vector file [selection] <put parameter description here>

Options:

- 0 — ulco
- 1 — ovw
- 2 — ulovw
- 3 — ulu

Default: 0

Mask Image [raster] Optional.

<put parameter description here>

8-neighbor connectivity [boolean] <put parameter description here>

Default: *True*

Stitch polygons [boolean] <put parameter description here>

Default: *True*

Minimum object size [number] <put parameter description here>

Default: *1*

Simplify polygons [number] <put parameter description here>

Default: *0.1*

Layer name [string] <put parameter description here>

Default: *layer*

Geometry index field name [string] <put parameter description here>

Default: *DN*

Tiles size [number] <put parameter description here>

Default: *1024*

Starting geometry index [number] <put parameter description here>

Default: *1*

OGR options for layer creation [string] Optional.

<put parameter description here>

Default: *None*

Outputs

Output vector file [vector] <put output description here>

Console usage

```
processing.runalg('otb:segmentationwatershed', -in, -filter, -filter.watershed.threshold, -filter
```

See also

.

18.4.9 Stereo

Stereo Framework

Description

<put algorithm description here>

Parameters

Input images list [**multipleinput: rasters**] <put parameter description here>

Couples list [**string**] Optional.

<put parameter description here>

Default: *None*

Image channel used for the block matching [**number**] <put parameter description here>

Default: *1*

Default elevation [**number**] <put parameter description here>

Default: *0*

Output resolution [**number**] <put parameter description here>

Default: *1*

NoData value [**number**] <put parameter description here>

Default: *-32768*

Method to fuse measures in each DSM cell [**selection**] <put parameter description here>

Options:

- 0 — max
- 1 — min
- 2 — mean
- 3 — acc

Default: *0*

Parameters estimation modes [**selection**] <put parameter description here>

Options:

- 0 — fit
- 1 — user

Default: *0*

Upper Left X [**number**] <put parameter description here>

Default: *0.0*

Upper Left Y [**number**] <put parameter description here>

Default: *0.0*

Size X [**number**] <put parameter description here>

Default: *0*

Size Y [number] <put parameter description here>

Default: 0

Pixel Size X [number] <put parameter description here>

Default: 0.0

Pixel Size Y [number] <put parameter description here>

Default: 0.0

Output Cartographic Map Projection [selection] <put parameter description here>

Options:

- 0 — utm
- 1 — lambert2
- 2 — lambert93
- 3 — wgs
- 4 — epsg

Default: 3

Zone number [number] <put parameter description here>

Default: 31

Northern Hemisphere [boolean] <put parameter description here>

Default: *True*

EPSG Code [number] <put parameter description here>

Default: 4326

Step of the deformation grid (in pixels) [number] <put parameter description here>

Default: 16

Sub-sampling rate for epipolar grid inversion [number] <put parameter description here>

Default: 10

Block-matching metric [selection] <put parameter description here>

Options:

- 0 — ssdmean
- 1 — ssd
- 2 — ncc
- 3 — lp

Default: 0

p value [number] <put parameter description here>

Default: 1

Radius of blocks for matching filter (in pixels) [number] <put parameter description here>

Default: 2

Minimum altitude offset (in meters) [number] <put parameter description here>

Default: -20

Maximum altitude offset (in meters) [number] <put parameter description here>

Default: *20*

Use bijection consistency in block matching strategy [boolean] <put parameter description here>

Default: *True*

Use median disparities filtering [boolean] <put parameter description here>

Default: *True*

Correlation metric threshold [number] <put parameter description here>

Default: *0.6*

Input left mask [raster] Optional.

<put parameter description here>

Input right mask [raster] Optional.

<put parameter description here>

Discard pixels with low local variance [number] <put parameter description here>

Default: *50*

Available RAM (Mb) [number] <put parameter description here>

Default: *128*

Outputs

Output DSM [raster] <put output description here>

Console usage

```
processing.runalg('otb:stereoframework', -input.il, -input.co, -input.channel, -elev.default, -out
```

See also

.

18.4.10 Vector

Concatenate

Description

<put algorithm description here>

Parameters

Input VectorDatas to concatenate [multipleinput: any vectors] <put parameter description here>

Outputs

Concatenated VectorData [**vector**] <put output description here>

Console usage

```
processing.runalg('otb:concatenate', -vd, -out)
```

See also

.

18.5 QGIS algorithm provider

QGIS algorithm provider implements various analysis and geoprocessing operations using mostly only QGIS API. So almost all algorithms from this provider will work “out of the box” without any additional configuration.

This provider incorporates fTools functionality, some algorithms from mmQGIS plugin and also adds its own algorithms.

.

18.5.1 Database

Import into PostGIS

Description

<put algorithm description here>

Parameters

Layer to import [**vector: any**] <put parameter description here>

Database (connection name) [**selection**] <put parameter description here>

Options:

- 0 — local

Default: 0

Schema (schema name) [**string**] <put parameter description here>

Default: *public*

Table to import to (leave blank to use layer name) [**string**] <put parameter description here>

Default: *(not set)*

Primary key field [**tablefield: any**] Optional.

<put parameter description here>

Geometry column [**string**] <put parameter description here>

Default: *geom*

Overwrite [boolean] <put parameter description here>

Default: *True*

Create spatial index [boolean] <put parameter description here>

Default: *True*

Convert field names to lowercase [boolean] <put parameter description here>

Default: *True*

Drop length constraints on character fields [boolean] <put parameter description here>

Default: *False*

Outputs

Console usage

```
processing.runalg('qgis:importintopostgis', input, database, schema, tablename, primary_key, geom
```

See also

PostGIS execute SQL

Description

<put algorithm description here>

Parameters

Database [string] <put parameter description here>

Default: *(not set)*

SQL query [string] <put parameter description here>

Default: *(not set)*

Outputs

Console usage

```
processing.runalg('qgis:postgisexecutesql', database, sql)
```

See also

.

18.5.2 Raster general

Set style for raster layer

Description

<put algorithm description here>

Parameters

Raster layer [**raster**] <put parameter description here>

Style file [**file**] <put parameter description here>

Outputs

Styled layer [**raster**] <put output description here>

Console usage

```
processing.runalg('qgis:setstyleforrasterlayer', input, style)
```

See also

.

18.5.3 Raster

Hypsometric curves

Description

Calculate hypsometric curves for features of polygon layer and save them as CSV file for further processing.

Parameters

DEM to analyze [**raster**] DEM to use for calculating altitudes.

Boundary layer [**vector: polygon**] Polygonal vector layer with boundaries of areas used to calculate hypsometric curves.

Step [**number**] Distance between curves.

Default: *100.0*

Use % of area instead of absolute value [**boolean**] Write area percentage to “Area” field of the CSV file instead of absolute area value.

Default: *False*

Outputs

Output directory [directory] Directory where output will be saved. For each feature from input vector layer CSV file with area and altitude values will be created.

File name consists of prefix `hystogram_` followed by layer name and feature ID.

Console usage

```
processing.runalg('qgis:hypsometriccurves', input_dem, boundary_layer, step, use_percentage, output_directory)
```

See also

Raster layer statistics

Description

Calculates basic statistics of the raster layer.

Parameters

Input layer [raster] Raster to analyze.

Outputs

Statistics [html] Analysis results in HTML format.

Minimum value [number] Minimum cell value.

Maximum value [number] Maximum cell value.

Sum [number] Sum of all cells values.

Mean value [number] Mean cell value.

valid cells count [number] Number of cell with data.

No-data cells count [number] Number of NODATA cells.

Standard deviation [number] Standard deviation of cells values.

Console usage

```
processing.runalg('qgis:rasterlayerstatistics', input_raster, output_html_file)
```

See also

Zonal Statistics

Description

Calculates some statistics values for pixels of input raster inside certain zones, defined as polygon layer.

Following values calculated for each zone:

- minimum

- maximum
- sum
- count
- mean
- standard deviation
- number of unique values
- range
- variance

Parameters

Raster layer [**raster**] Raster to analyze.

Raster band [**number**] Number of raster band to analyze.

Default: *1*

Vector layer containing zones [**vector: polygon**] Layer with zones boundaries.

Output column prefix [**string**] Prefix for output fields.

Default: *_*

Load whole raster in memory [**boolean**] Determines if raster band will be loaded in memory (**True**) or readed by chunks (**False**). Useful only when disk IO or raster scanning inefficiencies are your limiting factor.

Default: *True*

Outputs

Output layer [**vector**] The resulting layer. Basically this is same layer as zones layer with new columns containing statistics added.

Console usage

```
processing.runalg('qgis:zonalstatistics', input_raster, raster_band, input_vector, column_prefix,
```

See also

.

18.5.4 Table

Frequency analysis

Description

<put algorithm description here>

Parameters

input [vector: any] <put parameter description here>

fields [string] <put parameter description here>

Default: *(not set)*

Outputs

output [table] <put output description here>

Console usage

```
processing.runalg('qgis:frequencyanalysis', input, fields, output)
```

See also

.

18.5.5 Vector analysis

Count points in polygon

Description

Counts the number of points present in each feature of a polygon layer.

Parameters

Polygons [vector: polygon] Polygons layer.

Points [vector: point] Points layer.

Count field name [string] The name of the attribute table column containing the points number.

Default: *NUMPOINTS*

Outputs

Result [vector] Resulting layer with the attribute table containing the new column of the points count.

Console usage

```
processing.runalg('qgis:countpointsinpolygon', polygons, points, field, output)
```

See also

Count points in polygon (weighted)

Description

Counts the number of points in each feature of a polygon layer and calculates the mean of the selected field for each feature of the polygon layer. These values will be added to the attribute table of the resulting polygon layer.

Parameters

Polygons [vector: polygon] Polygons layer.

Points [vector: point] Points layer.

Weight field [tablefield: any] Weight field of the points attribute table.

Count field name [string] Name of the column for the new weighted field.

Default: *NUMPOINTS*

Outputs

Result [vector] The resulting polygons layer.

Console usage

```
processing.runalg('qgis:countpointsinpolygonweighted', polygons, points, weight, field, output)
```

See also

Count unique points in polygon

Description

Counts the number of unique values of a points in a polygons layer. Creates a new polygons layer with an extra column in the attribute table containing the count of unique values for each feature.

Parameters

Polygons [vector: polygon] Polygons layer.

Points [vector: point] Points layer.

Class field [tablefield: any] Points layer column name of the unique value chosen.

Count field name [string] Column name containing the count of unique values in the resulting polygons layer.

Default: *NUMPOINTS*

Outputs

Result [vector] The resulting polygons layer.

Console usage

```
processing.runalg('qgis:countuniquepointsinpolygon', polygons, points, classfield, field, output)
```

See also

Distance matrix

Description

<put algorithm description here>

Parameters

Input point layer [**vector: point**] <put parameter description here>

Input unique ID field [**tablefield: any**] <put parameter description here>

Target point layer [**vector: point**] <put parameter description here>

Target unique ID field [**tablefield: any**] <put parameter description here>

Output matrix type [**selection**] <put parameter description here>

Options:

- 0 — Linear (N*k x 3) distance matrix
- 1 — Standard (N x T) distance matrix
- 2 — Summary distance matrix (mean, std. dev., min, max)

Default: 0

Use only the nearest (k) target points [**number**] <put parameter description here>

Default: 0

Outputs

Distance matrix [**table**] <put output description here>

Console usage

```
processing.runalg('qgis:distancematrix', input_layer, input_field, target_layer, target_field, ma
```

See also

Distance to nearest hub

Description

<put algorithm description here>

Parameters

Source points layer [**vector: any**] <put parameter description here>

Destination hubs layer [**vector: any**] <put parameter description here>

Hub layer name attribute [**tablefield: any**] <put parameter description here>

Output shape type [**selection**] <put parameter description here>

Options:

- 0 — Point
- 1 — Line to hub

Default: 0

Measurement unit [**selection**] <put parameter description here>

Options:

- 0 — Meters
- 1 — Feet
- 2 — Miles
- 3 — Kilometers
- 4 — Layer units

Default: 0

Outputs

Output [**vector**] <put output description here>

Console usage

```
processing.runalg('qgis:distancetonearesthub', points, hubs, field, geometry, unit, output)
```

See also

Generate points (pixel centroids) along line

Description

<put algorithm description here>

Parameters

Raster layer [**raster**] <put parameter description here>

Vector layer [**vector: line**] <put parameter description here>

Outputs

Output layer [**vector**] <put output description here>

Console usage

```
processing.runalg('qgis:generatepointspixelcentroidsalongline', input_raster, input_vector, output)
```

See also

Generate points (pixel centroids) inside polygons

Description

<put algorithm description here>

Parameters

Raster layer [raster] <put parameter description here>

Vector layer [vector: polygon] <put parameter description here>

Outputs

Output layer [vector] <put output description here>

Console usage

```
processing.runalg('qgis:generatepointspixelcentroidsinsidepolygons', input_raster, input_vector, output)
```

See also

Hub lines

Description

Creates hub and spoke diagrams with lines drawn from points on the `Spoke Point` layer to matching points in the `Hub Point` layer. Determination of which hub goes with each point is based on a match between the `Hub ID field` on the hub points and the `Spoke ID field` on the spoke points.

Parameters

Hub point layer [vector: any] <put parameter description here>

Hub ID field [tablefield: any] <put parameter description here>

Spoke point layer [vector: any] <put parameter description here>

Spoke ID field [tablefield: any] <put parameter description here>

Outputs

Output [vector] The resulting layer.

Console usage

```
processing.runalg('qgis:hublines', hubs, hub_field, spokes, spoke_field, output)
```

See also

Mean coordinate(s)

Description

Calculates the mean of the coordinates of a layer starting from a field of the attribute table.

Parameters

Input layer [**vector: any**] <put parameter description here>

Weight field [**tablefield: numeric**] Optional.

Field to use if you want to perform a weighted mean.

Unique ID field [**tablefield: numeric**] Optional.

Unique field on which the calculation of the mean will be made.

Outputs

Result [**vector**] The resulting points layer.

Console usage

```
processing.runalg('qgis:meancoordinates', points, weight, uid, output)
```

See also

Nearest neighbour analysis

Description

<put algorithm description here>

Parameters

Points [**vector: point**] <put parameter description here>

Outputs

Result [**html**] <put output description here>

Observed mean distance [**number**] <put output description here>

Expected mean distance [**number**] <put output description here>

Nearest neighbour index [**number**] <put output description here>

Number of points [number] <put output description here>

Z-Score [number] <put output description here>

Console usage

```
processing.runalg('qgis:nearestneighbouranalysis', points, output)
```

See also

Sum line lengths

Description

<put algorithm description here>

Parameters

Lines [vector: line] <put parameter description here>

Polygons [vector: polygon] <put parameter description here>

Lines length field name [string] <put parameter description here>

Default: *LENGTH*

Lines count field name [string] <put parameter description here>

Default: *COUNT*

Outputs

Result [vector] <put output description here>

Console usage

```
processing.runalg('qgis:sumlinelengths', lines, polygons, len_field, count_field, output)
```

See also

18.5.6 Vector creation

Create grid

Description

Creates a grid.

Parameters

Grid type [selection] Grid type.

Options:

- 0 — Rectangle (line)
- 1 — Rectangle (polygon)
- 2 — Diamond (polygon)
- 3 — Hexagon (polygon)

Default: *0*

Width [number] Horizontal extent of the grid.

Default: *360.0*

Height [number] Vertical extent of the grid.

Default: *180.0*

Horizontal spacing [number] X-axes spacing between the lines.

Default: *10.0*

Vertical spacing [number] Y-axes spacing between the lines.

Default: *10.0*

Center X [number] X-coordinate of the grid center.

Default: *0.0*

Center Y [number] Y-coordinate of the grid center.

Default: *0.0*

Output CRS [crs] Coordinate reference system for grid.

Default: *EPSG:4326*

Outputs

Output [vector] The resulting grid layer (lines or polygons).

Console usage

```
processing.runalg('qgis:creategrid', type, width, height, hspacing, vspacing, centerx, centery, c
```

See also

Points layer from table

Description

Creates points layer from geometryless table with columns that contain point coordinates.

Parameters

Input layer [table] Input table

X field [tablefield: any] Table column containing the X coordinate.

Y field [tablefield: any] Table column containing the Y coordinate.

Target CRS [crs] Coordinate reference system to use for layer.

Default: *EPSG:4326*

Outputs

Output layer [vector] The resulting layer.

Console usage

```
processing.runalg('qgis:pointslayerfromtable', input, xfield, yfield, target_crs, output)
```

See also

Points to path

Description

<put algorithm description here>

Parameters

Input point layer [vector: point] <put parameter description here>

Group field [tablefield: any] <put parameter description here>

Order field [tablefield: any] <put parameter description here>

Date format (if order field is DateTime) [string] Optional.

<put parameter description here>

Default: *(not set)*

Outputs

Paths [vector] <put output description here>

Directory [directory] <put output description here>

Console usage

```
processing.runalg('qgis:pointstopath', vector, group_field, order_field, date_format, output_line)
```

See also

Random points along line

Description

<put algorithm description here>

Parameters

Input layer [vector: line] <put parameter description here>

Number of points [number] <put parameter description here>

Default: 1

Minimum distance [number] <put parameter description here>

Default: 0.0

Outputs

Random points [vector] <put output description here>

Console usage

```
processing.runalg('qgis:randompointsalongline', vector, point_number, min_distance, output)
```

See also

Random points in extent

Description

<put algorithm description here>

Parameters

Input extent [extent] <put parameter description here>

Default: 0,1,0,1

Points number [number] <put parameter description here>

Default: 1

Minimum distance [number] <put parameter description here>

Default: 0.0

Outputs

Random points [vector] <put output description here>

Console usage

```
processing.runalg('qgis:randompointsinextent', extent, point_number, min_distance, output)
```

See also

Random points in layer bounds

Description

<put algorithm description here>

Parameters

Input layer [vector: polygon] <put parameter description here>

Points number [number] <put parameter description here>

Default: 1

Minimum distance [number] <put parameter description here>

Default: 0.0

Outputs

Random points [vector] <put output description here>

Console usage

```
processing.runalg('qgis:randompointsinlayerbounds', vector, point_number, min_distance, output)
```

See also

Random points inside polygons (fixed)

Description

<put algorithm description here>

Parameters

Input layer [vector: polygon] <put parameter description here>

Sampling strategy [selection] <put parameter description here>

Options:

- 0 — Points count
- 1 — Points density

Default: 0

Number or density of points [number] <put parameter description here>

Default: *1.0*

Minimum distance [number] <put parameter description here>

Default: *0.0*

Outputs

Random points [vector] <put output description here>

Console usage

```
processing.runalg('qgis:randompointsinsidepolygonsfixed', vector, strategy, value, min_distance, ...)
```

See also

Random points inside polygons (variable)

Description

<put algorithm description here>

Parameters

Input layer [vector: polygon] <put parameter description here>

Sampling strategy [selection] <put parameter description here>

Options:

- 0 — Points count
- 1 — Points density

Default: *0*

Number field [tablefield: numeric] <put parameter description here>

Minimum distance [number] <put parameter description here>

Default: *0.0*

Outputs

Random points [vector] <put output description here>

Console usage

```
processing.runalg('qgis:randompointsinsidepolygonsvariable', vector, strategy, field, min_distance, ...)
```

See also

Regular points

Description

<put algorithm description here>

Parameters

Input extent [extent] <put parameter description here>

Default: *0,1,0,1*

Point spacing/count [number] <put parameter description here>

Default: *0.0001*

Initial inset from corner (LH side) [number] <put parameter description here>

Default: *0.0*

Apply random offset to point spacing [boolean] <put parameter description here>

Default: *False*

Use point spacing [boolean] <put parameter description here>

Default: *True*

Outputs

Regular points [vector] <put output description here>

Console usage

```
processing.runalg('qgis:regularpoints', extent, spacing, inset, randomize, is_spacing, output)
```

See also

Vector grid

Description

<put algorithm description here>

Parameters

Grid extent [extent] <put parameter description here>

Default: *0,1,0,1*

X spacing [number] <put parameter description here>

Default: *0.0001*

Y spacing [number] <put parameter description here>

Default: *0.0001*

Grid type [selection] <put parameter description here>

Options:

- 0 — Output grid as polygons
- 1 — Output grid as lines

Default: 0

Outputs

Grid [vector] <put output description here>

Console usage

```
processing.runalg('qgis:vectorgrid', extent, step_x, step_y, type, output)
```

See also

.

18.5.7 Vector general

Delete duplicate geometries

Description

<put algorithm description here>

Parameters

Input layer [vector: any] <put parameter description here>

Outputs

Output [vector] <put output description here>

Console usage

```
processing.runalg('qgis:deleteduplicategeometries', input, output)
```

See also

Join attributes by location

Description

<put algorithm description here>

Parameters

Target vector layer [**vector: any**] <put parameter description here>

Join vector layer [**vector: any**] <put parameter description here>

Attribute summary [**selection**] <put parameter description here>

Options:

- 0 — Take attributes of the first located feature
- 1 — Take summary of intersecting features

Default: 0

Statistics for summary (comma separated) [**string**] <put parameter description here>

Default: *sum,mean,min,max,median*

Output table [**selection**] <put parameter description here>

Options:

- 0 — Only keep matching records
- 1 — Keep all records (including non-matching target records)

Default: 0

Outputs

Output layer [**vector**] <put output description here>

Console usage

```
processing.runalg('qgis:joinattributesbylocation', target, join, summary, stats, keep, output)
```

See also

Join attributes table

Description

<put algorithm description here>

Parameters

Input layer [**vector: any**] <put parameter description here>

Input layer 2 [**table**] <put parameter description here>

Table field [**tablefield: any**] <put parameter description here>

Table field 2 [**tablefield: any**] <put parameter description here>

Outputs

Output layer [**vector**] <put output description here>

Console usage

```
processing.runalg('qgis:joinattributetable', input_layer, input_layer_2, table_field, table_fiel
```

See also

Merge vector layers

Description

<put algorithm description here>

Parameters

Input layer 1 [vector: any] <put parameter description here>

Input layer 2 [vector: any] <put parameter description here>

Outputs

Output [vector] <put output description here>

Console usage

```
processing.runalg('qgis:mergevectorlayers', layer1, layer2, output)
```

See also

Polygon from layer extent

Description

<put algorithm description here>

Parameters

Input layer [vector: any] <put parameter description here>

Calculate extent for each feature separately [boolean] <put parameter description here>

Default: *False*

Outputs

Output layer [vector] <put output description here>

Console usage

```
processing.runalg('qgis:polygonfromlayerextent', input_layer, by_feature, output)
```

See also

Reproject layer

Description

Reprojects a vector layer in a different CRS.

Parameters

Input layer [vector: any] Layer to reproject.

Target CRS [crs] Destination coordinate reference system.

Default: *EPSG:4326*

Outputs

Reprojected layer [vector] The resulting layer.

Console usage

```
processing.runalg('qgis:reprojectlayer', input, target_crs, output)
```

See also

Salva le geometrie selezionate

Descrizione

Salva le geometrie selezionate come un nuovo layer.

Parametri

Input layer [vector: any] Layer to process.

Output

Output layer with selected features [vector] Layer risultante

Console usage

```
processing.runalg('qgis:savesselectedfeatures', input_layer, output_layer)
```

Vedi anche

Set style for vector layer

Description

<put algorithm description here>

Parameters

Vector layer [vector: any] <put parameter description here>

Style file [file] <put parameter description here>

Outputs

Styled layer [vector] <put output description here>

Console usage

```
processing.runalg('qgis:setstyleforvectorlayer', input, style)
```

See also

Snap points to grid

Description

<put algorithm description here>

Parameters

Input Layer [vector: any] <put parameter description here>

Horizontal spacing [number] <put parameter description here>

Default: *0.1*

Vertical spacing [number] <put parameter description here>

Default: *0.1*

Outputs

Output [vector] <put output description here>

Console usage

```
processing.runalg('qgis:snappointstogrid', input, hspacing, vspacing, output)
```

See also

Split vector layer

Description

<put algorithm description here>

Parameters

Input layer [vector: any] <put parameter description here>

Unique ID field [tablefield: any] <put parameter description here>

Outputs

Output directory [directory] <put output description here>

Console usage

```
processing.runalg('qgis:splitvectorlayer', input, field, output)
```

See also

.

18.5.8 Vector geometry

Concave hull

Description

<put algorithm description here>

Parameters

Input point layer [vector: point] <put parameter description here>

Threshold (0-1, where 1 is equivalent with Convex Hull) [number] <put parameter description here>

Default: *0.3*

Allow holes [boolean] <put parameter description here>

Default: *True*

Split multipart geometry into singleparts geometries [boolean] <put parameter description here>

Default: *False*

Outputs

Concave hull [vector] <put output description here>

Console usage

```
processing.runalg('qgis:concavehull', input, alpha, holes, no_multigeometry, output)
```

See also

Convert geometry type

Description

Converts a geometry type to another one.

Parameters

Input layer [vector: any] Layer in input.

New geometry type [selection] Type of conversion to perform.

Options:

- 0 — Centroids
- 1 — Nodes
- 2 — Linestrings
- 3 — Multilinestrings
- 4 — Polygons

Default: 0

Outputs

Output [vector] The resulting layer.

Console usage

```
processing.runalg('qgis:convertgeometrytype', input, type, output)
```

See also

Convex hull

Description

<put algorithm description here>

Parameters

Input layer [**vector: any**] <put parameter description here>

Field (optional, only used if creating convex hulls by classes) [**tablefield: any**]

Optional.

<put parameter description here>

Method [**selection**] <put parameter description here>

Options:

- 0 — Create single minimum convex hull
- 1 — Create convex hulls based on field

Default: 0

Outputs

Convex hull [**vector**] <put output description here>

Console usage

```
processing.runalg('qgis:convexhull', input, field, method, output)
```

See also

Create points along lines

Description

<put algorithm description here>

Parameters

lines [**vector: any**] <put parameter description here>

distance [**number**] <put parameter description here>

Default: 1

startpoint [**number**] <put parameter description here>

Default: 0

endpoint [**number**] <put parameter description here>

Default: 0

Outputs

output [**vector**] <put output description here>

Console usage

```
processing.runalg('qgis:createpointsalonglines', lines, distance, startpoint, endpoint, output)
```

See also

Delaunay triangulation

Description

<put algorithm description here>

Parameters

Input layer [vector: point] <put parameter description here>

Outputs

Delaunay triangulation [vector] <put output description here>

Console usage

```
processing.runalg('qgis:delaunaytriangulation', input, output)
```

See also

Densify geometries given an interval

Description

<put algorithm description here>

Parameters

Input layer [vector: polygon, line] <put parameter description here>

Interval between Vertices to add [number] <put parameter description here>

Default: 1.0

Outputs

Densified layer [vector] <put output description here>

Console usage

```
processing.runalg('qgis:densifygeometriesgivenaninterval', input, interval, output)
```

See also

Densify geometries

Description

<put algorithm description here>

Parameters

Input layer [vector: polygon, line] <put parameter description here>

Vertices to add [number] <put parameter description here>

Default: 1

Outputs

Densified layer [vector] <put output description here>

Console usage

```
processing.runalg('qgis:densifygeometries', input, vertices, output)
```

See also

Dissolve

Description

<put algorithm description here>

Parameters

Input layer [vector: polygon, line] <put parameter description here>

Dissolve all (do not use field) [boolean] <put parameter description here>

Default: *True*

Unique ID field [tablefield: any] Optional.

<put parameter description here>

Outputs

Dissolved [vector] <put output description here>

Console usage

```
processing.runalg('qgis:dissolve', input, dissolve_all, field, output)
```

See also

Eliminate sliver polygons

Description

<put algorithm description here>

Parameters

Input layer [**vector: polygon**] <put parameter description here>

Use current selection in input layer (works only if called from toolbox) [**boolean**]
<put parameter description here>

Default: *False*

Selection attribute [**tablefield: any**] <put parameter description here>

Comparison [**selection**] <put parameter description here>

Options:

- 0 — ==
- 1 — !=
- 2 — >
- 3 — >=
- 4 — <
- 5 — <=
- 6 — begins with
- 7 — contains

Default: *0*

Value [**string**] <put parameter description here>

Default: *0*

Merge selection with the neighbouring polygon with the [**selection**] <put parameter description here>

Options:

- 0 — Largest area
- 1 — Smallest Area
- 2 — Largest common boundary

Default: *0*

Outputs

Cleaned layer [**vector**] <put output description here>

Console usage

```
processing.runalg('qgis:eliminatesliverpolygons', input, keepselection, attribute, comparison, co
```

See also

Explode lines

Description

<put algorithm description here>

Parameters

Input layer [vector: line] <put parameter description here>

Outputs

Output layer [vector] <put output description here>

Console usage

```
processing.runalg('qgis:explodelines', input, output)
```

See also

Extract nodes

Description

<put algorithm description here>

Parameters

Input layer [vector: polygon, line] <put parameter description here>

Outputs

Output layer [vector] <put output description here>

Console usage

```
processing.runalg('qgis:extractnodes', input, output)
```

See also

Fill holes

Description

<put algorithm description here>

Parameters

Polygons [vector: any] <put parameter description here>

Max area [number] <put parameter description here>

Default: *100000*

Outputs

Results [vector] <put output description here>

Console usage

```
processing.runalg('qgis:fillholes', polygons, max_area, results)
```

See also

Fixed distance buffer

Description

<put algorithm description here>

Parameters

Input layer [vector: any] <put parameter description here>

Distance [number] <put parameter description here>

Default: *10.0*

Segments [number] <put parameter description here>

Default: *5*

Dissolve result [boolean] <put parameter description here>

Default: *False*

Outputs

Buffer [vector] <put output description here>

Console usage

```
processing.runalg('qgis:fixeddistancebuffer', input, distance, segments, dissolve, output)
```

See also

Keep n biggest parts

Description

<put algorithm description here>

Parameters

Polygons [vector: polygon] <put parameter description here>

To keep [number] <put parameter description here>

Default: 1

Outputs

Results [vector] <put output description here>

Console usage

```
processing.runalg('qgis:keepnbiggestparts', polygons, to_keep, results)
```

See also

Lines to polygons

Description

<put algorithm description here>

Parameters

Input layer [vector: line] <put parameter description here>

Outputs

Output layer [vector] <put output description here>

Console usage

```
processing.runalg('qgis:linestopolygons', input, output)
```

See also

Multipart to singleparts

Description

<put algorithm description here>

Parameters

Input layer [vector: any] <put parameter description here>

Outputs

Output layer [vector] <put output description here>

Console usage

```
processing.runalg('qgis:multiparttosingleparts', input, output)
```

See also

Points displacement

Description

Moves overlapped points at small distance, that they all become visible. The result is very similar to the output of the “Point displacement” renderer but it is permanent.

Parameters

Input layer [vector: point] Layer with overlapped points.

Displacement distance [number] Desired displacement distance **NOTE:** displacement distance should be in same units as layer.

Default: *0.00015*

Horizontal distribution for two point case [boolean] Controls distribution direction in case of two overlapped points. If *True* points will be distributed horizontally, otherwise they will be distributed vertically.

Default: *True*

Outputs

Output layer [vector] The resulting layer with shifted overlapped points.

Console usage

```
processing.runalg('qgis:pointsdisplacement', input_layer, distance, horizontal, output_layer)
```

See also

Polygon centroids

Description

<put algorithm description here>

Parameters

Input layer [**vector: polygon**] <put parameter description here>

Outputs

Output layer [**vector**] <put output description here>

Console usage

```
processing.runalg('qgis:polygoncentroids', input_layer, output_layer)
```

See also

Polygonize

Description

<put algorithm description here>

Parameters

Input layer [**vector: line**] <put parameter description here>

Keep table structure of line layer [**boolean**] <put parameter description here>

Default: *False*

Create geometry columns [**boolean**] <put parameter description here>

Default: *True*

Outputs

Output layer [**vector**] <put output description here>

Console usage

```
processing.runalg('qgis:polygonize', input, fields, geometry, output)
```

See also

Polygons to lines

Description

<put algorithm description here>

Parameters

Input layer [vector: polygon] <put parameter description here>

Outputs

Output layer [vector] <put output description here>

Console usage

```
processing.runalg('qgis:polygontolines', input, output)
```

See also

Simplify geometries

Description

<put algorithm description here>

Parameters

Input layer [vector: polygon, line] <put parameter description here>

Tolerance [number] <put parameter description here>

Default: *1.0*

Outputs

Simplified layer [vector] <put output description here>

Console usage

```
processing.runalg('qgis:simplifygeometries', input, tolerance, output)
```

See also

Singleparts to multipart

Description

<put algorithm description here>

Parameters

Input layer [vector: any] <put parameter description here>

Unique ID field [tablefield: any] <put parameter description here>

Outputs

Output layer [vector] <put output description here>

Console usage

```
processing.runalg('qgis:singlepartstomultipart', input, field, output)
```

See also

Variable distance buffer

Description

<put algorithm description here>

Parameters

Input layer [vector: any] <put parameter description here>

Distance field [tablefield: any] <put parameter description here>

Segments [number] <put parameter description here>

Default: 5

Dissolve result [boolean] <put parameter description here>

Default: *False*

Outputs

Buffer [vector] <put output description here>

Console usage

```
processing.runalg('qgis:variabledistancebuffer', input, field, segments, dissolve, output)
```

See also

Voronoi polygons

Description

<put algorithm description here>

Parameters

Input layer [vector: point] <put parameter description here>

Buffer region [number] <put parameter description here>

Default: *0.0*

Outputs

Voronoi polygons [vector] <put output description here>

Console usage

```
processing.runalg('qgis:voronoipolygons', input, buffer, output)
```

See also

.

18.5.9 Vector overlay

Clip

Description

<put algorithm description here>

Parameters

Input layer [vector: any] <put parameter description here>

Clip layer [vector: any] <put parameter description here>

Outputs

Clipped [vector] <put output description here>

Console usage

```
processing.runalg('qgis:clip', input, overlay, output)
```

See also

Difference

Description

<put algorithm description here>

Parameters

Input layer [vector: any] <put parameter description here>

Difference layer [vector: any] <put parameter description here>

Outputs

Difference [vector] <put output description here>

Console usage

```
processing.runalg('qgis:difference', input, overlay, output)
```

See also

Intersection

Description

<put algorithm description here>

Parameters

Input layer [vector: any] <put parameter description here>

Intersect layer [vector: any] <put parameter description here>

Outputs

Intersection [vector] <put output description here>

Console usage

```
processing.runalg('qgis:intersection', input, input2, output)
```

See also

Line intersections

Description

<put algorithm description here>

Parameters

Input layer [vector: line] <put parameter description here>

Intersect layer [vector: line] <put parameter description here>

Input unique ID field [tablefield: any] <put parameter description here>

Intersect unique ID field [tablefield: any] <put parameter description here>

Outputs

Output layer [vector] <put output description here>

Console usage

```
processing.runalg('qgis:lineintersections', input_a, input_b, field_a, field_b, output)
```

See also

Symmetrical difference

Description

<put algorithm description here>

Parameters

Input layer [vector: any] <put parameter description here>

Difference layer [vector: any] <put parameter description here>

Outputs

Symmetrical difference [vector] <put output description here>

Console usage

```
processing.runalg('qgis:symmetricaldifference', input, overlay, output)
```

See also

Union

Description

<put algorithm description here>

Parameters

Input layer [vector: any] <put parameter description here>

Input layer 2 [vector: any] <put parameter description here>

Outputs

Union [vector] <put output description here>

Console usage

```
processing.runalg('qgis:union', input, input2, output)
```

See also

.

18.5.10 Vector selection

Extract by attribute

Description

<put algorithm description here>

Parameters

Input Layer [vector: any] <put parameter description here>

Selection attribute [tablefield: any] <put parameter description here>

Operator [selection] <put parameter description here>

Options:

- 0 — =
- 1 — !=
- 2 — >
- 3 — >=
- 4 — <
- 5 — <=
- 6 — begins with

- 7 — contains

Default: 0

Value [string] <put parameter description here>

Default: *(not set)*

Outputs

Output [vector] <put output description here>

Console usage

```
processing.runalg('qgis:extractbyattribute', input, field, operator, value, output)
```

See also

Extract by location

Description

<put algorithm description here>

Parameters

Layer to select from [vector: any] <put parameter description here>

Additional layer (intersection layer) [vector: any] <put parameter description here>

Include input features that touch the selection features [boolean] <put parameter description here>

Default: *False*

Include input features that overlap/cross the selection features [boolean] <put parameter description here>

Default: *False*

Include input features completely within the selection features [boolean] <put parameter description here>

Default: *False*

Outputs

Selection [vector] <put output description here>

Console usage

```
processing.runalg('qgis:extractbylocation', input, intersect, touches, overlaps, within, output)
```

See also

Random extract

Description

<put algorithm description here>

Parameters

Input layer [**vector: any**] <put parameter description here>

Method [**selection**] <put parameter description here>

Options:

- 0 — Number of selected features
- 1 — Percentage of selected features

Default: 0

Number/percentage of selected features [**number**] <put parameter description here>

Default: 10

Outputs

Selection [**vector**] <put output description here>

Console usage

```
processing.runalg('qgis:randomextract', input, method, number, output)
```

See also

Random extract within subsets

Description

<put algorithm description here>

Parameters

Input layer [**vector: any**] <put parameter description here>

ID Field [**tablefield: any**] <put parameter description here>

Method [**selection**] <put parameter description here>

Options:

- 0 — Number of selected features
- 1 — Percentage of selected features

Default: 0

Number/percentage of selected features [number] <put parameter description here>

Default: *10*

Outputs

Selection [vector] <put output description here>

Console usage

```
processing.runalg('qgis:randomextractwithinsubsets', input, field, method, number, output)
```

See also

Random selection

Description

<put algorithm description here>

Parameters

Input layer [vector: any] <put parameter description here>

Method [selection] <put parameter description here>

Options:

- 0 — Number of selected features
- 1 — Percentage of selected features

Default: *0*

Number/percentage of selected features [number] <put parameter description here>

Default: *10*

Outputs

Selection [vector] <put output description here>

Console usage

```
processing.runalg('qgis:randomselection', input, method, number)
```

See also

Random selection within subsets

Description

<put algorithm description here>

Parameters

Input layer [**vector: any**] <put parameter description here>

ID Field [**tablefield: any**] <put parameter description here>

Method [**selection**] <put parameter description here>

Options:

- 0 — Number of selected features
- 1 — Percentage of selected features

Default: 0

Number/percentage of selected features [**number**] <put parameter description here>

Default: 10

Outputs

Selection [**vector**] <put output description here>

Console usage

```
processing.runalg('qgis:randomselectionwithinsubsets', input, field, method, number)
```

See also

Select by attribute

Description

Selects and saves as new layer all features from input layer that satisfy condition.

NOTE: algorithm is case-sensitive (“qgis” is different from “Qgis” and “QGIS”)

Parameters

Input Layer [**vector: any**] Layer to process.

Selection attribute [**tablefield: any**] Field on which perform the selection.

Operator [**selection**] Comparison operator.

Options:

- 0 — =
- 1 — !=
- 2 — >
- 3 — >=
- 4 — <
- 5 — <=
- 6 — begins with
- 7 — contains

Default: 0

Value [string] Value to compare.

Default: *(not set)*

Outputs

Output [vector] The resulting layer.

Console usage

```
processing.runalg('qgis:selectbyattribute', input, field, operator, value, output)
```

See also

Select by expression

Description

<put algorithm description here>

Parameters

Input Layer [vector: any] <put parameter description here>

Expression [string] <put parameter description here>

Default: *(not set)*

Modify current selection by [selection] <put parameter description here>

Options:

- 0 — creating new selection
- 1 — adding to current selection
- 2 — removing from current selection

Default: 0

Outputs

Output [vector] <put output description here>

Console usage

```
processing.runalg('qgis:selectbyexpression', layername, expression, method)
```

See also

Select by location

Description

<put algorithm description here>

Parameters

Layer to select from [vector: any] <put parameter description here>

Additional layer (intersection layer) [vector: any] <put parameter description here>

Include input features that touch the selection features [boolean] <put parameter description here>

Default: *False*

Include input features that overlap/cross the selection features [boolean] <put parameter description here>

Default: *False*

Include input features completely within the selection features [boolean] <put parameter description here>

Default: *False*

Modify current selection by [selection] <put parameter description here>

Options:

- 0 — creating new selection
- 1 — adding to current selection
- 2 — removing from current selection

Default: *0*

Outputs

Selection [vector] <put output description here>

Console usage

```
processing.runalg('qgis:selectbylocation', input, intersect, touches, overlaps, within, method)
```

See also

.

18.5.11 Vector table

Add autoincremental field

Description

<put algorithm description here>

Parameters

Input layer [vector: any] <put parameter description here>

Outputs

Output layer [vector] <put output description here>

Console usage

```
processing.runalg('qgis:addautoincrementalfield', input, output)
```

See also

Add field to attributes table

Description

<put algorithm description here>

Parameters

Input layer [vector: any] <put parameter description here>

Field name [string] <put parameter description here>

Default: *(not set)*

Field type [selection] <put parameter description here>

Options:

- 0 — Integer
- 1 — Float
- 2 — String

Default: *0*

Field length [number] <put parameter description here>

Default: *10*

Field precision [number] <put parameter description here>

Default: *0*

Outputs

Output layer [vector] <put output description here>

Console usage

```
processing.runalg('qgis:addfieldtoattributetable', input_layer, field_name, field_type, field_length)
```

See also

Advanced Python field calculator

Description

<put algorithm description here>

Parameters

Input layer [vector: any] <put parameter description here>

Result field name [string] <put parameter description here>

Default: *NewField*

Field type [selection] <put parameter description here>

Options:

- 0 — Integer
- 1 — Float
- 2 — String

Default: *0*

Field length [number] <put parameter description here>

Default: *10*

Field precision [number] <put parameter description here>

Default: *0*

Global expression [string] Optional.

<put parameter description here>

Default: *(not set)*

Formula [string] <put parameter description here>

Default: *value =*

Outputs

Output layer [vector] <put output description here>

Console usage

```
processing.runalg('qgis:advancedpythonfieldcalculator', input_layer, field_name, field_type, field
```

See also

Basic statistics for numeric fields

Description

<put algorithm description here>

Parameters

Input vector layer [**vector: any**] <put parameter description here>

Field to calculate statistics on [**tablefield: numeric**] <put parameter description here>

Outputs

Statistics for numeric field [**html**] <put output description here>

Coefficient of Variation [**number**] <put output description here>

Minimum value [**number**] <put output description here>

Maximum value [**number**] <put output description here>

Sum [**number**] <put output description here>

Mean value [**number**] <put output description here>

Count [**number**] <put output description here>

Range [**number**] <put output description here>

Median [**number**] <put output description here>

Number of unique values [**number**] <put output description here>

Standard deviation [**number**] <put output description here>

Console usage

```
processing.runalg('qgis:basicstatisticsfornumericfields', input_layer, field_name, output_html_fi
```

See also

Basic statistics for text fields

Description

<put algorithm description here>

Parameters

Input vector layer [vector: any] <put parameter description here>

Field to calculate statistics on [tablefield: string] <put parameter description here>

Outputs

Statistics for text field [html] <put output description here>

Minimum length [number] <put output description here>

Maximum length [number] <put output description here>

Mean length [number] <put output description here>

Count [number] <put output description here>

Number of empty values [number] <put output description here>

Number of non-empty values [number] <put output description here>

Number of unique values [number] <put output description here>

Console usage

```
processing.runalg('qgis:basicstatisticsfortextfields', input_layer, field_name, output_html_file)
```

See also

Create equivalent numerical field

Description

<put algorithm description here>

Parameters

Input layer [vector: any] <put parameter description here>

Class field [tablefield: any] <put parameter description here>

Outputs

Output layer [vector] <put output description here>

Console usage

```
processing.runalg('qgis:createequivalentnumericalfield', input, field, output)
```

See also

Delete column

Description

<put algorithm description here>

Parameters

Input layer [vector: any] <put parameter description here>

Field to delete [tablefield: any] <put parameter description here>

Outputs

Output [vector] <put output description here>

Console usage

```
processing.runalg('qgis:deletecolumn', input, column, output)
```

See also

Export/Add geometry columns

Description

<put algorithm description here>

Parameters

Input layer [vector: any] <put parameter description here>

Calculate using [selection] <put parameter description here>

Options:

- 0 — Layer CRS
- 1 — Project CRS
- 2 — Ellipsoidal

Default: 0

Outputs

Output layer [vector] <put output description here>

Console usage

```
processing.runalg('qgis:exportaddgeometrycolumns', input, calc_method, output)
```

See also

Field calculator

Description

<put algorithm description here>

Parameters

Input layer [vector: any] <put parameter description here>

Result field name [string] <put parameter description here>

Default: *(not set)*

Field type [selection] <put parameter description here>

Options:

- 0 — Float
- 1 — Integer
- 2 — String
- 3 — Date

Default: *0*

Field length [number] <put parameter description here>

Default: *10*

Field precision [number] <put parameter description here>

Default: *3*

Create new field [boolean] <put parameter description here>

Default: *True*

Formula [string] <put parameter description here>

Default: *(not set)*

Outputs

Output layer [vector] <put output description here>

Console usage

```
processing.runalg('qgis:fieldcalculator', input_layer, field_name, field_type, field_length, field...
```

See also

List unique values

Description

Lists unique values of an attribute table field and counts their number.

Parameters

Input layer [vector: any] Layer to analyze.

Target field [tablefield: any] Field to analyze.

Outputs

Unique values [html] Analysis results in HTML format.

Total unique values [number] Total number of unique values in given field.

Unique values [string] List of all unique values in given field.

Console usage

```
processing.runalg('qgis:listuniquevalues', input_layer, field_name, output)
```

See also

Number of unique values in classes

Description

<put algorithm description here>

Parameters

input [vector: any] <put parameter description here>

class field [tablefield: any] <put parameter description here>

value field [tablefield: any] <put parameter description here>

Outputs

output [vector] <put output description here>

Console usage

```
processing.runalg('qgis:numberofuniquevaluesinclasses', input, class_field, value_field, output)
```

See also

Statistics by categories

Description

<put algorithm description here>

Parameters

Input vector layer [vector: any] <put parameter description here>

Field to calculate statistics on [tablefield: numeric] <put parameter description here>

Field with categories [tablefield: any] <put parameter description here>

Outputs

Statistics [table] <put output description here>

Console usage

```
processing.runalg('qgis:statisticsbycategories', input_layer, values_field_name, categories_field)
```

See also

Text to float

Description

<put algorithm description here>

Parameters

Input Layer [vector: any] <put parameter description here>

Text attribute to convert to float [tablefield: string] <put parameter description here>

Outputs

Output [vector] <put output description here>

Console usage

```
processing.runalg('qgis:texttofloat', input, field, output)
```

See also

.

18.6 R algorithm provider

R also called GNU S, is a strongly functional language and environment to statistically explore data sets, make many graphical displays of data from custom data sets

Nota: Please remember that Processing contains only R scripts, so you need to install R by yourself and configure Processing properly.

18.6.1 Basic statistics

Frequency table

Description

<put algorithm description here>

Parameters

Layer [vector: any] <put parameter description here>

Field [tablefield: any] <put parameter description here>

Outputs

R Console Output [html] <put output description here>

Console usage

```
processing.runalg('r:frequencytable', layer, field, r_console_output)
```

See also

Kolmogrov-Smirnov test

Description

<put algorithm description here>

Parameters

Layer [vector: any] <put parameter description here>

Field [tablefield: any] <put parameter description here>

Outputs

R Console Output [html] <put output description here>

Console usage

```
processing.runalg('r:kolmogrovsmirnovtest', layer, field, r_console_output)
```

See also

Summary statistics

Description

<put algorithm description here>

Parameters

Layer [vector: any] <put parameter description here>

Field [tablefield: any] <put parameter description here>

Outputs

R Console Output [html] <put output description here>

Console usage

```
processing.runalg('r:summarystatistics', layer, field, r_console_output)
```

See also

.

18.6.2 Home range

Characteristic hull method

Description

<put algorithm description here>

Parameters

Layer [vector: any] <put parameter description here>

Field [tablefield: any] <put parameter description here>

Outputs

Home_ranges [vector] <put output description here>

Console usage

```
processing.runalg('r:characteristichullmethod', layer, field, home_ranges)
```

See also

Kernel h ref

Description

<put algorithm description here>

Parameters

Layer [vector: any] <put parameter description here>

Field [tablefield: any] <put parameter description here>

Grid [number] <put parameter description here>

Default: *10.0*

Percentage [number] <put parameter description here>

Default: *10.0*

Folder [directory] Optional.

<put parameter description here>

Outputs

Home_ranges [vector] <put output description here>

Console usage

```
processing.runalg('r:kernelhref', layer, field, grid, percentage, folder, home_ranges)
```

See also

Minimum convex polygon

Description

<put algorithm description here>

Parameters

Layer [vector: any] <put parameter description here>

Percentage [number] <put parameter description here>

Default: *10.0*

Field [tablefield: any] <put parameter description here>

Outputs

Home_ranges [vector] <put output description here>

Console usage

```
processing.runalg('r:minimumconvexpolygon', layer, percentage, field, home_ranges)
```

See also

Single-linkage cluster analysis

Description

<put algorithm description here>

Parameters

Layer [vector: any] <put parameter description here>

Field [tablefield: any] <put parameter description here>

Percentage [number] <put parameter description here>

Default: *10.0*

Outputs

R Plots [html] <put output description here>

Home_ranges [vector] <put output description here>

Console usage

```
processing.runalg('r:singlelinkageclusteranalysis', layer, field, percentage, rplots, home_ranges)
```

See also

.

18.6.3 Point pattern

F function

Description

<put algorithm description here>

Parameters

Layer [vector: any] <put parameter description here>

Nsim [number] <put parameter description here>

Default: *10.0*

Outputs

R Plots [html] <put output description here>

Console usage

```
processing.runalg('r:ffunction', layer, nsim, rplots)
```

See also

G function

Description

<put algorithm description here>

Parameters

Layer [vector: any] <put parameter description here>

Nsim [number] <put parameter description here>

Default: *10.0*

Outputs

R Plots [html] <put output description here>

Console usage

```
processing.runalg('r:gfunction', layer, nsim, rplots)
```

See also

Monte-Carlo spatial randomness

Description

<put algorithm description here>

Parameters

Layer [vector: any] <put parameter description here>

Simulations [number] <put parameter description here>

Default: *100.0*

Optional plot name [string] <put parameter description here>

Default: *(not set)*

Outputs

R Plots [html] <put output description here>

R Console Output [html] <put output description here>

Console usage

```
processing.runalg('r:montecarlospatialrandomness', layer, simulations, optional_plot_name, rplots)
```

See also

Quadrat analysis

Description

<put algorithm description here>

Parameters

Layer [vector: any] <put parameter description here>

Outputs

R Plots [html] <put output description here>

R Console Output [html] <put output description here>

Console usage

```
processing.runalg('r:quadratanalysis', layer, rplots, r_console_output)
```

See also

Random sampling grid

Description

<put algorithm description here>

Parameters

Layer [vector: any] <put parameter description here>

Size [number] <put parameter description here>

Default: *10.0*

Outputs

Output [vector] <put output description here>

Console usage

```
processing.runalg('r:randomsamplinggrid', layer, size, output)
```

See also

Regular sampling grid

Description

<put algorithm description here>

Parameters

Layer [vector: any] <put parameter description here>

Size [number] <put parameter description here>

Default: *10.0*

Outputs

Output [vector] <put output description here>

Console usage

```
processing.runalg('r:regularsamplinggrid', layer, size, output)
```

See also

Relative distribution (distance covariate)

Description

<put algorithm description here>

Parameters

Layer [vector: any] <put parameter description here>

Covariate [vector: any] <put parameter description here>

Covariate name [string] <put parameter description here>

Default: *mandatory_covariate_name_(no_spaces)*

x label [string] <put parameter description here>

Default: *(not set)*

Plot name [string] <put parameter description here>

Default: *(not set)*

Legend position [string] <put parameter description here>

Default: *float*

Outputs

R Plots [html] <put output description here>

Console usage

```
processing.runalg('r:relativedistributiondistancecovariate', layer, covariate, covariate_name, x_
```

See also

Relative distribution (raster covariate)

Description

<put algorithm description here>

Parameters

points [vector: any] <put parameter description here>

covariate [raster] <put parameter description here>

covariate name [string] <put parameter description here>

Default: *mandatory_covariate_name_(no_spaces)*

x label [string] <put parameter description here>

Default: *(not set)*

plot name [string] <put parameter description here>

Default: *(not set)*

legend position [string] <put parameter description here>

Default: *float*

Outputs

R Plots [html] <put output description here>

Console usage

```
processing.runalg('r:relativedistributionrastercovariate', points, covariate, covariate_name, x_l
```

See also

Ripley - Rasson spatial domain

Description

<put algorithm description here>

Parameters

Layer [vector: any] <put parameter description here>

Outputs

Output [vector] <put output description here>

Console usage

```
processing.runalg('r:ripleyrassonspatialdomain', layer, output)
```

See also

.

18.6.4 Raster processing

Advanced raster histogram

Description

<put algorithm description here>

Parameters

Layer [raster] <put parameter description here>

Dens or Hist [string] <put parameter description here>

Default: *Hist*

Outputs

R Plots [html] <put output description here>

Console usage

```
processing.runalg('r:advancedrasterhistogram', layer, dens_or_hist, rplots)
```

See also

Raster histogram

Description

<put algorithm description here>

Parameters

Layer [raster] <put parameter description here>

Outputs

R Plots [html] <put output description here>

Console usage

```
processing.runalg('r:rasterhistogram', layer, rplots)
```

See also

.

18.6.5 Vector processing

Histogram

Description

<put algorithm description here>

Parameters

Layer [vector: any] <put parameter description here>

Field [tablefield: any] <put parameter description here>

Outputs

R Plots [html] <put output description here>

Console usage

```
processing.runalg('r:histogram', layer, field, rplots)
```

See also

.

18.7 SAGA algorithm provider

SAGA (System for Automated Geoscientific Analyses) is a free, hybrid, cross-platform GIS software. SAGA provides many geoscientific methods which are bundled in so-called module libraries.

Nota: Please remember that Processing contains only the interface description, so you need to install SAGA by yourself and configure Processing properly.

18.7.1 Geostatistics

Directional statistics for single grid

Description

<put algorithm description here>

Parameters

Grid [**raster**] <put parameter description here>

Points [**vector: any**] Optional.

<put parameter description here>

Direction [**Degree**] [**number**] <put parameter description here>

Default: *0.0*

Tolerance [**Degree**] [**number**] <put parameter description here>

Default: *0.0*

Maximum Distance [**Cells**] [**number**] <put parameter description here>

Default: *0*

Distance Weighting [**selection**] <put parameter description here>

Options:

- 0 — [0] no distance weighting
- 1 — [1] inverse distance to a power
- 2 — [2] exponential
- 3 — [3] gaussian weighting

Default: *0*

Inverse Distance Weighting Power [number] <put parameter description here>

Default: *1*

Inverse Distance Offset [boolean] <put parameter description here>

Default: *True*

Gaussian and Exponential Weighting Bandwidth [number] <put parameter description here>

Default: *1.0*

Outputs

Arithmetic Mean [raster] <put output description here>

Difference from Arithmetic Mean [raster] <put output description here>

Minimum [raster] <put output description here>

Maximum [raster] <put output description here>

Range [raster] <put output description here>

Variance [raster] <put output description here>

Standard Deviation [raster] <put output description here>

Mean less Standard Deviation [raster] <put output description here>

Mean plus Standard Deviation [raster] <put output description here>

Deviation from Arithmetic Mean [raster] <put output description here>

Percentile [raster] <put output description here>

Directional Statistics for Points [vector] <put output description here>

Console usage

```
processing.runalg('saga:directionalstatisticsforsinglegrid', grid, points, direction, tolerance, n)
```

See also

Fast representativeness

Description

<put algorithm description here>

Parameters

Input [raster] <put parameter description here>

Level of Generalisation [number] <put parameter description here>

Default: *16*

Outputs

Output [raster] <put output description here>

Output **Lod** [raster] <put output description here>

Output **Seeds** [raster] <put output description here>

Console usage

```
processing.runalg('saga:fastrepresentativeness', input, lod, result, result_lod, seeds)
```

See also

Geographically weighted multiple regression (points/grids)

Description

<put algorithm description here>

Parameters

Predictors [multipleinput: rasters] <put parameter description here>

Output of Regression Parameters [boolean] <put parameter description here>

Default: *True*

Points [vector: point] <put parameter description here>

Dependent Variable [tablefield: any] <put parameter description here>

Distance Weighting [selection] <put parameter description here>

Options:

- 0 — [0] no distance weighting
- 1 — [1] inverse distance to a power
- 2 — [2] exponential
- 3 — [3] gaussian weighting

Default: *0*

Inverse Distance Weighting Power [number] <put parameter description here>

Default: *1*

Inverse Distance Offset [boolean] <put parameter description here>

Default: *True*

Gaussian and Exponential Weighting Bandwidth [number] <put parameter description here>

Default: *1.0*

Search Range [selection] <put parameter description here>

Options:

- 0 — [0] search radius (local)
- 1 — [1] no search radius (global)

Default: 0

Search Radius [number] <put parameter description here>

Default: 100

Search Mode [selection] <put parameter description here>

Options:

- 0 — [0] all directions
- 1 — [1] quadrants

Default: 0

Number of Points [selection] <put parameter description here>

Options:

- 0 — [0] maximum number of observations
- 1 — [1] all points

Default: 0

Maximum Number of Observations [number] <put parameter description here>

Default: 10

Minimum Number of Observations [number] <put parameter description here>

Default: 4

Outputs

Regression [raster] <put output description here>

Coefficient of Determination [raster] <put output description here>

Regression Parameters [raster] <put output description here>

Residuals [vector] <put output description here>

Console usage

```
processing.runalg('saga:geographicallyweightedmultipleregressionpointsgrids', predictors, parameter)
```

See also

Geographically weighted multiple regression (points)

Description

<put algorithm description here>

Parameters

Points [vector: any] <put parameter description here>

Dependent Variable [tablefield: any] <put parameter description here>

Distance Weighting [selection] <put parameter description here>

Options:

- 0 — [0] no distance weighting
- 1 — [1] inverse distance to a power
- 2 — [2] exponential
- 3 — [3] gaussian weighting

Default: 0

Inverse Distance Weighting Power [number] <put parameter description here>

Default: 1

Inverse Distance Offset [boolean] <put parameter description here>

Default: *True*

Gaussian and Exponential Weighting Bandwidth [number] <put parameter description here>

Default: 1.0

Search Range [selection] <put parameter description here>

Options:

- 0 — [0] search radius (local)
- 1 — [1] no search radius (global)

Default: 0

Search Radius [number] <put parameter description here>

Default: 100

Search Mode [selection] <put parameter description here>

Options:

- 0 — [0] all directions
- 1 — [1] quadrants

Default: 0

Number of Points [selection] <put parameter description here>

Options:

- 0 — [0] maximum number of observations
- 1 — [1] all points

Default: 0

Maximum Number of Observations [number] <put parameter description here>

Default: 10

Minimum Number of Observations [number] <put parameter description here>

Default: 4

Outputs

Regression [vector] <put output description here>

Console usage

```
processing.runalg('saga:geographicallyweightedmultipleregressionpoints', points, dependent, distan
```

See also

Geographically weighted multiple regression

Description

<put algorithm description here>

Parameters

Points [vector: point] <put parameter description here>

Dependent Variable [tablefield: any] <put parameter description here>

Target Grids [selection] <put parameter description here>

Options:

- 0 — [0] user defined

Default: 0

Distance Weighting [selection] <put parameter description here>

Options:

- 0 — [0] no distance weighting
- 1 — [1] inverse distance to a power
- 2 — [2] exponential
- 3 — [3] gaussian weighting

Default: 0

Inverse Distance Weighting Power [number] <put parameter description here>

Default: 1

Inverse Distance Offset [boolean] <put parameter description here>

Default: *True*

Gaussian and Exponential Weighting Bandwidth [number] <put parameter description here>

Default: 1

Search Range [selection] <put parameter description here>

Options:

- 0 — [0] search radius (local)
- 1 — [1] no search radius (global)

Default: 0

Search Radius [number] <put parameter description here>

Default: 100

Search Mode [selection] <put parameter description here>

Options:

- 0 — [0] all directions
- 1 — [1] quadrants

Default: 0

Number of Points [selection] <put parameter description here>

Options:

- 0 — [0] maximum number of observations
- 1 — [1] all points

Default: 0

Maximum Number of Observations [number] <put parameter description here>

Default: 10

Minimum Number of Observations [number] <put parameter description here>

Default: 4

Output extent [extent] <put parameter description here>

Default: 0,1,0,1

Cellsize [number] <put parameter description here>

Default: 100.0

Outputs

Quality [raster] <put output description here>

Intercept [raster] <put output description here>

Quality [raster] <put output description here>

Intercept [raster] <put output description here>

Console usage

```
processing.runalg('saga:geographicallyweightedmultipleregression', points, dependent, target, dist
```

See also

Geographically weighted regression (points/grid)

Description

<put algorithm description here>

Parameters

Predictor [raster] <put parameter description here>

Points [vector: point] <put parameter description here>

Dependent Variable [tablefield: any] <put parameter description here>

Distance Weighting [selection] <put parameter description here>

Options:

- 0 — [0] no distance weighting
- 1 — [1] inverse distance to a power
- 2 — [2] exponential
- 3 — [3] gaussian weighting

Default: 0

Inverse Distance Weighting Power [number] <put parameter description here>

Default: 1

Inverse Distance Offset [boolean] <put parameter description here>

Default: *True*

Gaussian and Exponential Weighting Bandwidth [number] <put parameter description here>

Default: 1.0

Search Range [selection] <put parameter description here>

Options:

- 0 — [0] search radius (local)
- 1 — [1] no search radius (global)

Default: 0

Search Radius [number] <put parameter description here>

Default: 0

Search Mode [selection] <put parameter description here>

Options:

- 0 — [0] all directions
- 1 — [1] quadrants

Default: 0

Number of Points [selection] <put parameter description here>

Options:

- 0 — [0] maximum number of observations
- 1 — [1] all points

Default: 0

Maximum Number of Observations [number] <put parameter description here>

Default: 10

Minimum Number of Observations [number] <put parameter description here>

Default: 4

Outputs

Regression [raster] <put output description here>

Coefficient of Determination [raster] <put output description here>

Intercept [raster] <put output description here>

Slope [raster] <put output description here>

Residuals [vector] <put output description here>

Console usage

```
processing.runalg('saga:geographicallyweightedregressionpointsgrid', predictor, points, dependent,
```

See also

Geographically weighted regression

Description

<put algorithm description here>

Parameters

Points [**vector: point**] <put parameter description here>

Dependent Variable [**tablefield: any**] <put parameter description here>

Predictor [**tablefield: any**] <put parameter description here>

Target Grids [**selection**] <put parameter description here>

Options:

- 0 — [0] user defined

Default: *0*

Distance Weighting [**selection**] <put parameter description here>

Options:

- 0 — [0] no distance weighting
- 1 — [1] inverse distance to a power
- 2 — [2] exponential
- 3 — [3] gaussian weighting

Default: *0*

Inverse Distance Weighting Power [**number**] <put parameter description here>

Default: *0*

Inverse Distance Offset [**boolean**] <put parameter description here>

Default: *True*

Gaussian and Exponential Weighting Bandwidth [**number**] <put parameter description here>

Default: *0.0*

Search Range [**selection**] <put parameter description here>

Options:

- 0 — [0] search radius (local)
- 1 — [1] no search radius (global)

Default: *0*

Search Radius [**number**] <put parameter description here>

Default: *100*

Search Mode [selection] <put parameter description here>

Options:

- 0 — [0] all directions
- 1 — [1] quadrants

Default: 0

Number of Points [selection] <put parameter description here>

Options:

- 0 — [0] maximum number of observations
- 1 — [1] all points

Default: 0

Maximum Number of Observations [number] <put parameter description here>

Default: 10

Minimum Number of Observations [number] <put parameter description here>

Default: 4

Output extent [extent] <put parameter description here>

Default: 0,1,0,1

Cellsize [number] <put parameter description here>

Default: 100.0

Outputs

Grid [raster] <put output description here>

Quality [raster] <put output description here>

Intercept [raster] <put output description here>

Slope [raster] <put output description here>

Console usage

```
processing.runalg('saga:geographicallyweightedregression', points, dependent, predictor, target, 0)
```

See also

Global moran's i for grids

Description

<put algorithm description here>

Parameters

Grid [raster] <put parameter description here>

Case of contiguity [selection] <put parameter description here>

Options:

- 0 — [0] Rook
 - 1 — [1] Queen
- Default: 0

Outputs

Result [table] <put output description here>

Console usage

```
processing.runalg('saga:globalmoransiforgrids', grid, contiguity, result)
```

See also

Minimum distance analysis

Description

Performs a complete distance analysis of a point layer:

- minimum distance of points
- maximum distance of points
- average distance of all the points
- standard deviation of the distance
- duplicated points

Parameters

Points [vector: point] Layer to analyze.

Outputs

Minimum Distance Analysis [table] The resulting table.

Console usage

```
processing.runalg('saga:minimumdistanceanalysis', points, table)
```

See also

Multi-band variation

Description

<put algorithm description here>

Parameters

Grids [**multipleinput: rasters**] <put parameter description here>

Radius [**Cells**] [**number**] <put parameter description here>

Default: *1*

Distance Weighting [**selection**] <put parameter description here>

Options:

- 0 — [0] no distance weighting
- 1 — [1] inverse distance to a power
- 2 — [2] exponential
- 3 — [3] gaussian weighting

Default: *0*

Inverse Distance Weighting Power [**number**] <put parameter description here>

Default: *1*

Inverse Distance Offset [**boolean**] <put parameter description here>

Default: *True*

Gaussian and Exponential Weighting Bandwidth [**number**] <put parameter description here>

Default: *1.0*

Outputs

Mean Distance [**raster**] <put output description here>

Standard Deviation [**raster**] <put output description here>

Distance [**raster**] <put output description here>

Console usage

```
processing.runalg('saga:multibandvariation', bands, radius, distance_weighting_weighting, distance_offset)
```

See also

Multiple regression analysis (grid/grids)

Description

<put algorithm description here>

Parameters

Dependent [**raster**] <put parameter description here>

Grids [**multipleinput: rasters**] <put parameter description here>

Grid Interpolation [**selection**] <put parameter description here>

Options:

- 0 — [0] Nearest Neighbor
- 1 — [1] Bilinear Interpolation
- 2 — [2] Inverse Distance Interpolation
- 3 — [3] Bicubic Spline Interpolation
- 4 — [4] B-Spline Interpolation

Default: 0

Include X Coordinate [boolean] <put parameter description here>

Default: *True*

Include Y Coordinate [boolean] <put parameter description here>

Default: *True*

Method [selection] <put parameter description here>

Options:

- 0 — [0] include all
- 1 — [1] forward
- 2 — [2] backward
- 3 — [3] stepwise

Default: 0

P in [number] <put parameter description here>

Default: 5

P out [number] <put parameter description here>

Default: 5

Outputs

Regression [raster] <put output description here>

Residuals [raster] <put output description here>

Details: Coefficients [table] <put output description here>

Details: Model [table] <put output description here>

Details: Steps [table] <put output description here>

Console usage

```
processing.runalg('saga:multipleregressionanalysisgridgrids', dependent, grids, interpol, coord_x,
```

See also

Multiple regression analysis (points/grids)

Description

<put algorithm description here>

Parameters

Grids [**multipleinput: rasters**] <put parameter description here>

Shapes [**vector: any**] <put parameter description here>

Attribute [**tablefield: any**] <put parameter description here>

Grid Interpolation [**selection**] <put parameter description here>

Options:

- 0 — [0] Nearest Neighbor
- 1 — [1] Bilinear Interpolation
- 2 — [2] Inverse Distance Interpolation
- 3 — [3] Bicubic Spline Interpolation
- 4 — [4] B-Spline Interpolation

Default: 0

Include X Coordinate [**boolean**] <put parameter description here>

Default: *True*

Include Y Coordinate [**boolean**] <put parameter description here>

Default: *True*

Method [**selection**] <put parameter description here>

Options:

- 0 — [0] include all
- 1 — [1] forward
- 2 — [2] backward
- 3 — [3] stepwise

Default: 0

P in [**number**] <put parameter description here>

Default: 5

P out [**number**] <put parameter description here>

Default: 5

Outputs

Details: **Coefficients** [**table**] <put output description here>

Details: **Model** [**table**] <put output description here>

Details: **Steps** [**table**] <put output description here>

Residuals [**vector**] <put output description here>

Regression [**raster**] <put output description here>

Console usage

```
processing.runalg('saga:multipleregressionanalysispointsgrids', grids, shapes, attribute, interpo
```

See also

Polynomial regression

Description

<put algorithm description here>

Parameters

Points [vector: any] <put parameter description here>

Attribute [tablefield: any] <put parameter description here>

Polynom [selection] <put parameter description here>

Options:

- 0 — [0] simple planar surface
- 1 — [1] bi-linear saddle
- 2 — [2] quadratic surface
- 3 — [3] cubic surface
- 4 — [4] user defined

Default: 0

Maximum X Order [number] <put parameter description here>

Default: 4

Maximum Y Order [number] <put parameter description here>

Default: 4

Maximum Total Order [number] <put parameter description here>

Default: 4

Trend Surface [selection] <put parameter description here>

Options:

- 0 — [0] user defined

Default: 0

Output extent [extent] <put parameter description here>

Default: 0,1,0,1

Cellsize [number] <put parameter description here>

Default: 100.0

Outputs

Residuals [vector] <put output description here>

Grid [raster] <put output description here>

Console usage

```
processing.runalg('saga:polynomialregression', points, attribute, polynom, xorder, yorder, torder)
```

See also

Radius of variance (grid)

Description

<put algorithm description here>

Parameters

Grid [raster] <put parameter description here>

Standard Deviation [number] <put parameter description here>

Default: *1.0*

Maximum Search Radius (cells) [number] <put parameter description here>

Default: *20*

Type of Output [selection] <put parameter description here>

Options:

- 0 — [0] Cells
- 1 — [1] Map Units

Default: *0*

Outputs

Variance Radius [raster] <put output description here>

Console usage

```
processing.runalg('saga:radiusofvariancegrid', input, variance, radius, output, result)
```

See also

Regression analysis

Description

<put algorithm description here>

Parameters

Grid [raster] <put parameter description here>

Shapes [vector: any] <put parameter description here>

Attribute [tablefield: any] <put parameter description here>

Grid Interpolation [selection] <put parameter description here>

Options:

- 0 — [0] Nearest Neighbor
- 1 — [1] Bilinear Interpolation
- 2 — [2] Inverse Distance Interpolation
- 3 — [3] Bicubic Spline Interpolation
- 4 — [4] B-Spline Interpolation

Default: 0

Regression Function [selection] <put parameter description here>

Options:

- 0 — [0] $Y = a + b * X$ (linear)
- 1 — [1] $Y = a + b / X$
- 2 — [2] $Y = a / (b - X)$
- 3 — [3] $Y = a * X^b$ (power)
- 4 — [4] $Y = a e^{(b * X)}$ (exponential)
- 5 — [5] $Y = a + b * \ln(X)$ (logarithmic)

Default: 0

Outputs

Regression [raster] <put output description here>

Residuals [vector] <put output description here>

Console usage

```
processing.runalg('saga:regressionanalysis', grid, shapes, attribute, interpol, method, regression)
```

See also

Representativeness

Description

<put algorithm description here>

Parameters

Grid [raster] <put parameter description here>

Radius (Cells) [number] <put parameter description here>

Default: *10*

Exponent [number] <put parameter description here>

Default: *1*

Outputs

Representativeness [raster] <put output description here>

Console usage

```
processing.runalg('saga:representativeness', input, radius, exponent, result)
```

See also

Residual analysis

Description

<put algorithm description here>

Parameters

Grid [raster] <put parameter description here>

Radius (Cells) [number] <put parameter description here>

Default: *7*

Distance Weighting [selection] <put parameter description here>

Options:

- 0 — [0] no distance weighting
- 1 — [1] inverse distance to a power
- 2 — [2] exponential
- 3 — [3] gaussian weighting

Default: *0*

Inverse Distance Weighting Power [number] <put parameter description here>

Default: *1*

Inverse Distance Offset [boolean] <put parameter description here>

Default: *True*

Gaussian and Exponential Weighting Bandwidth [number] <put parameter description here>

Default: *1.0*

Outputs

Mean Value [raster] <put output description here>
Difference from Mean Value [raster] <put output description here>
Standard Deviation [raster] <put output description here>
Value Range [raster] <put output description here>
Minimum Value [raster] <put output description here>
Maximum Value [raster] <put output description here>
Deviation from Mean Value [raster] <put output description here>
Percentile [raster] <put output description here>

Console usage

```
processing.runalg('saga:residualanalysis', grid, radius, distance_weighting_weighting, distance_w
```

See also

Spatial point pattern analysis

Description

<put algorithm description here>

Parameters

Points [vector: point] <put parameter description here>
Vertex Distance [Degree] [number] <put parameter description here>
Default: 5

Outputs

Mean Centre [vector] <put output description here>
Standard Distance [vector] <put output description here>
Bounding Box [vector] <put output description here>

Console usage

```
processing.runalg('saga:spatialpointpatternanalysis', points, step, centre, stddist, bbox)
```

See also

Statistics for grids

Description

<put algorithm description here>

Parameters

Grids [**multipleinput: rasters**] <put parameter description here>

Outputs

Arithmetic Mean [**raster**] <put output description here>

Minimum [**raster**] <put output description here>

Maximum [**raster**] <put output description here>

Variance [**raster**] <put output description here>

Standard Deviation [**raster**] <put output description here>

Mean less Standard Deviation [**raster**] <put output description here>

Mean plus Standard Deviation [**raster**] <put output description here>

Console usage

```
processing.runalg('saga:statisticsforgrids', grids, mean, min, max, var, stddev, stddevlo, stddevhi)
```

See also

Variogram cloud

Description

<put algorithm description here>

Parameters

Points [**vector: point**] <put parameter description here>

Attribute [**tablefield: any**] <put parameter description here>

Maximum Distance [**number**] <put parameter description here>

Default: *0.0*

Skip Number [**number**] <put parameter description here>

Default: *1*

Outputs

Variogram Cloud [**table**] <put output description here>

Console usage

```
processing.runalg('saga:variogramcloud', points, field, distmax, nskip, result)
```

See also

Variogram surface

Description

<put algorithm description here>

Parameters

Points [vector: point] <put parameter description here>

Attribute [tablefield: any] <put parameter description here>

Number of Distance Classes [number] <put parameter description here>

Default: 10

Skip Number [number] <put parameter description here>

Default: 1

Outputs

Number of Pairs [raster] <put output description here>

Variogram Surface [raster] <put output description here>

Covariance Surface [raster] <put output description here>

Console usage

```
processing.runalg('saga:variogramsurface', points, field, distcount, nskip, count, variance, covar
```

See also

Zonal grid statistics

Description

<put algorithm description here>

Parameters

Zone Grid [raster] <put parameter description here>

Categorical Grids [multipleinput: rasters] Optional.

<put parameter description here>

Grids to analyse [multipleinput: rasters] Optional.

<put parameter description here>

Aspect [raster] Optional.

<put parameter description here>

Short Field Names [boolean] <put parameter description here>

Default: *True*

Outputs

Zonal Statistics [table] <put output description here>

Console usage

```
processing.runalg('saga:zonalgridstatistics', zones, catlist, statlist, aspect, shortnames, outta
```

See also

.

18.7.2 Grid analysis

Accumulated cost (anisotropic)

Description

<put algorithm description here>

Parameters

Cost Grid [raster] <put parameter description here>

Direction of max cost [raster] <put parameter description here>

Destination Points [raster] <put parameter description here>

k factor [number] <put parameter description here>

Default: *1*

Threshold for different route [number] <put parameter description here>

Default: *0*

Outputs

Accumulated Cost [raster] <put output description here>

Console usage

```
processing.runalg('saga:accumulatedcostanisotropic', cost, direction, points, k, threshold, accco
```

See also

Accumulated cost (isotropic)

Description

<put algorithm description here>

Parameters

Cost Grid [raster] <put parameter description here>

Destination Points [raster] <put parameter description here>

Threshold for different route [number] <put parameter description here>

Default: 0.0

Outputs

Accumulated Cost [raster] <put output description here>

Closest Point [raster] <put output description here>

Console usage

```
processing.runalg('saga:accumulatedcostisotropic', cost, points, threshold, acccost, closestpt)
```

See also

Aggregation index

Description

<put algorithm description here>

Parameters

Input Grid [raster] <put parameter description here>

Max. Number of Classes [number] <put parameter description here>

Default: 5

Outputs

Result [table] <put output description here>

Console usage

```
processing.runalg('saga:aggregationindex', input, maxnumclass, result)
```

See also

Analytical hierarchy process

Description

<put algorithm description here>

Parameters

Input Grids [**multipleinput: rasters**] <put parameter description here>

Pairwise Comparisons Table [**table**] <put parameter description here>

Outputs

Output Grid [**raster**] <put output description here>

Console usage

```
processing.runalg('saga:analyticalhierarchyprocess', grids, table, output)
```

See also

Cross-classification and tabulation

Description

<put algorithm description here>

Parameters

Input Grid 1 [**raster**] <put parameter description here>

Input Grid 2 [**raster**] <put parameter description here>

Max. Number of Classes [**number**] <put parameter description here>

Default: 5

Outputs

Cross-Classification Grid [**raster**] <put output description here>

Cross-Tabulation Table [**table**] <put output description here>

Console usage

```
processing.runalg('saga:crossclassificationandtabulation', input, input2, maxnumclass, resultgrid)
```

See also

Fragmentation (alternative)

Description

<put algorithm description here>

Parameters

Classification [raster] <put parameter description here>

Class Identifier [number] <put parameter description here>

Default: 1

Neighborhood Min [number] <put parameter description here>

Default: 1

Neighborhood Max [number] <put parameter description here>

Default: 1

Level Aggregation [selection] <put parameter description here>

Options:

- 0 — [0] average
- 1 — [1] multiplicative

Default: 0

Add Border [boolean] <put parameter description here>

Default: *True*

Connectivity Weighting [number] <put parameter description here>

Default: 1.1

Minimum Density [Percent] [number] <put parameter description here>

Default: 10

Minimum Density for Interior Forest [Percent] [number] <put parameter description here>

Default: 99

Search Distance Increment [number] <put parameter description here>

Default: 0.0

Density from Neighbourhood [boolean] <put parameter description here>

Default: *True*

Outputs

Density [Percent] [raster] <put output description here>

Connectivity [Percent] [raster] <put output description here>

Fragmentation [raster] <put output description here>

Summary [table] <put output description here>

Console usage

```
processing.runalg('saga:fragmentationalternative', classes, class, neighborhood_min, neighborhood_max)
```

See also

Fragmentation classes from density and connectivity

Description

<put algorithm description here>

Parameters

Density [Percent] [raster] <put parameter description here>

Connectivity [Percent] [raster] <put parameter description here>

Add Border [boolean] <put parameter description here>

Default: *True*

Connectivity Weighting [number] <put parameter description here>

Default: *0*

Minimum Density [Percent] [number] <put parameter description here>

Default: *10*

Minimum Density for Interior Forest [Percent] [number] <put parameter description here>

Default: *99*

Outputs

Fragmentation [raster] <put output description here>

Console usage

```
processing.runalg('saga:fragmentationclassesfromdensityandconnectivity', density, connectivity, border)
```

See also

Fragmentation (standard)

Description

<put algorithm description here>

Parameters

Classification [raster] <put parameter description here>

Class Identifier [number] <put parameter description here>

Default: 1

Neighborhood Min [number] <put parameter description here>

Default: 1

Neighborhood Max [number] <put parameter description here>

Default: 3

Level Aggregation [selection] <put parameter description here>

Options:

- 0 — [0] average
- 1 — [1] multiplicative

Default: 0

Add Border [boolean] <put parameter description here>

Default: *True*

Connectivity Weighting [number] <put parameter description here>

Default: 1.1

Minimum Density [Percent] [number] <put parameter description here>

Default: 10

Minimum Density for Interior Forest [Percent] [number] <put parameter description here>

Default: 99

Neighborhood Type [selection] <put parameter description here>

Options:

- 0 — [0] square
- 1 — [1] circle

Default: 0

Include diagonal neighbour relations [boolean] <put parameter description here>

Default: *True*

Outputs

Density [Percent] [raster] <put output description here>

Connectivity [Percent] [raster] <put output description here>

Fragmentation [raster] <put output description here>

Summary [table] <put output description here>

Console usage

```
processing.runalg('saga:fragmentationstandard', classes, class, neighborhood_min, neighborhood_max)
```

See also

Layer of extreme value

Description

<put algorithm description here>

Parameters

Grids [**multipleinput: rasters**] <put parameter description here>

Method [**selection**] <put parameter description here>

Options:

- 0 — [0] Maximum
- 1 — [1] Minimum

Default: 0

Outputs

Result [**raster**] <put output description here>

Console usage

```
processing.runalg('saga:layerofextremevalue', grids, criteria, result)
```

See also

Least cost paths

Description

<put algorithm description here>

Parameters

Source Point (s) [**vector: point**] <put parameter description here>

Accumulated cost [**raster**] <put parameter description here>

Values [**multipleinput: rasters**] Optional.

<put parameter description here>

Outputs

Profile (points) [**vector**] <put output description here>

Profile (lines) [**vector**] <put output description here>

Console usage

```
processing.runalg('saga:leastcostpaths', source, dem, values, points, line)
```

See also

Ordered Weighted Averaging

Description

<put algorithm description here>

Parameters

Input Grids [**multipleinput: rasters**] <put parameter description here>

Weights [**fixedtable**] <put parameter description here>

Outputs

Output Grid [**raster**] <put output description here>

Console usage

```
processing.runalg('saga:orderedweightedaveraging', grids, weights, output)
```

See also

Pattern analysis

Description

<put algorithm description here>

Parameters

Input Grid [**raster**] <put parameter description here>

Size of Analysis Window [**selection**] <put parameter description here>

Options:

- 0 — [0] 3 X 3
- 1 — [1] 5 X 5
- 2 — [2] 7 X 7

Default: 0

Max. Number of Classes [**number**] <put parameter description here>

Default: 0

Outputs

Relative Richness [raster] <put output description here>

Diversity [raster] <put output description here>

Dominance [raster] <put output description here>

Fragmentation [raster] <put output description here>

Number of Different Classes [raster] <put output description here>

Center Versus Neighbours [raster] <put output description here>

Console usage

```
processing.runalg('saga:patternanalysis', input, winsize, maxnumclass, relative, diversity, domin
```

See also

Soil texture classification

Description

<put algorithm description here>

Parameters

Sand [raster] Optional.

<put parameter description here>

Silt [raster] Optional.

<put parameter description here>

Clay [raster] Optional.

<put parameter description here>

Outputs

Soil Texture [raster] <put output description here>

Sum [raster] <put output description here>

Console usage

```
processing.runalg('saga:soiltextureclassification', sand, silt, clay, texture, sum)
```

See also

.

18.7.3 Grid calculus

Function

Description

<put algorithm description here>

Parameters

xmin [number] <put parameter description here>

Default: *0.0*

xmax [number] <put parameter description here>

Default: *0.0*

ymin [number] <put parameter description here>

Default: *0.0*

ymax [number] <put parameter description here>

Default: *0.0*

Formula [string] <put parameter description here>

Default: *(not set)*

Outputs

Function [raster] <put output description here>

Console usage

```
processing.runalg('saga:function', xmin, xmax, ymin, ymax, formul, result)
```

See also

Fuzzify

Description

<put algorithm description here>

Parameters

Grid [raster] <put parameter description here>

A [number] <put parameter description here>

Default: *0.0*

B [number] <put parameter description here>

Default: *0.0*

C [number] <put parameter description here>

Default: *0.0*

D [number] <put parameter description here>

Default: *0.0*

Membership Function Type [selection] <put parameter description here>

Options:

- 0 — [0] linear
- 1 — [1] sigmoidal
- 2 — [2] j-shaped

Default: *0*

Adjust to Grid [boolean] <put parameter description here>

Default: *True*

Outputs

Fuzzified Grid [raster] <put output description here>

Console usage

```
processing.runalg('saga:fuzzify', input, a, b, c, d, type, autofit, output)
```

See also

Fuzzy intersection (and)

Description

<put algorithm description here>

Parameters

Grids [multipleinput: rasters] <put parameter description here>

Operator Type [selection] <put parameter description here>

Options:

- 0 — [0] min(a, b) (non-interactive)
- 1 — [1] a * b
- 2 — [2] max(0, a + b - 1)

Default: *0*

Outputs

Intersection [raster] <put output description here>

Console usage

```
processing.runalg('saga:fuzzyintersectionand', grids, type, and)
```

See also

Fuzzy union (or)

Description

<put algorithm description here>

Parameters

Grids [**multipleinput: rasters**] <put parameter description here>

Operator Type [**selection**] <put parameter description here>

Options:

- 0 — [0] $\max(a, b)$ (non-interactive)
- 1 — [1] $a + b - a * b$
- 2 — [2] $\min(1, a + b)$

Default: 0

Outputs

Union [**raster**] <put output description here>

Console usage

```
processing.runalg('saga:fuzzyunionor', grids, type, or)
```

See also

Geometric figures

Description

Draws simple geometric figures.

Parameters

Cell Count [**number**] Number of cells to use.

Default: 0

Cell Size [**number**] Size of the single cell.

Default: 0

Figure [selection] Type of the figure.

Options:

- 0 — [0] Cone (up)
- 1 — [1] Cone (down)
- 2 — [2] Plane

Default: 0

Direction of Plane [Degree] [number] Rotation factor in degrees.

Default: 0

Outputs

Result [raster] The resulting layer.

Console usage

```
processing.runalg('saga:geometricfigures', cell_count, cell_size, figure, plane, result)
```

See also

Gradient vector from cartesian to polar coordinates

Description

<put algorithm description here>

Parameters

X Component [raster] <put parameter description here>

Y Component [raster] <put parameter description here>

Polar Angle Units [selection] <put parameter description here>

Options:

- 0 — [0] radians
- 1 — [1] degree

Default: 0

Polar Coordinate System [selection] <put parameter description here>

Options:

- 0 — [0] mathematical
- 1 — [1] geographical
- 2 — [2] user defined

Default: 0

User defined Zero Direction [number] <put parameter description here>

Default: 0.0

User defined Orientation [selection] <put parameter description here>

Options:

- 0 — [0] clockwise
- 1 — [1] counterclockwise

Default: 0

Outputs

Direction [raster] <put output description here>

Length [raster] <put output description here>

Console usage

```
processing.runalg('saga:gradientvectorfromcartesiantopolarcoordinates', dx, dy, units, system, sy
```

See also

Gradient vector from polar to cartesian coordinates

Description

<put algorithm description here>

Parameters

Direction [raster] <put parameter description here>

Length [raster] <put parameter description here>

Polar Angle Units [selection] <put parameter description here>

Options:

- 0 — [0] radians
- 1 — [1] degree

Default: 0

Polar Coordinate System [selection] <put parameter description here>

Options:

- 0 — [0] mathematical
- 1 — [1] geographical
- 2 — [2] user defined

Default: 0

User defined Zero Direction [number] <put parameter description here>

Default: 0.0

User defined Orientation [selection] <put parameter description here>

Options:

- 0 — [0] clockwise

- 1 — [1] counterclockwise

Default: 0

Outputs

X Component [raster] <put output description here>

Y Component [raster] <put output description here>

Console usage

```
processing.runalg('saga:gradientvectorfrompolarcoordinates', dir, len, units, system, ...)
```

See also

Grid difference

Description

Creates a new grid layer as the result of the difference between two other grid layers.

Parameters

A [raster] First layer.

B [raster] Second layer.

Outputs

Difference (A - B) [raster] The resulting layer.

Console usage

```
processing.runalg('saga:griddifference', a, b, c)
```

See also

Grid division

Description

Creates a new grid layer as the result of the division between two other grid layers.

Parameters

Dividend [raster] First layer.

Divisor [raster] Second layer.

Outputs

Quotient [raster] The resulting layer.

Console usage

```
processing.runalg('saga:griddivision', a, b, c)
```

See also

Grid normalisation

Description

Normalises the grid values according to minimum and maximum values chosen.

Parameters

Grid [raster] Grid to normalize.

Target Range (min) [number] Minimum value.

Default: *0*

Target Range (max) [number] Maximum value.

Default: *1*

Outputs

Normalised Grid [raster] The resulting layer.

Console usage

```
processing.runalg('saga:gridnormalisation', input, range_min, range_max, output)
```

See also

Grids product

Description

<put algorithm description here>

Parameters

Grids [multipleinput: rasters] <put parameter description here>

Outputs

Product [raster] <put output description here>

Console usage

```
processing.runalg('saga:gridsproduct', grids, result)
```

See also

Grids sum

Description

Creates a new grid layer as the result of the sum of two or more grid layers.

Parameters

Grids [multipleinput: rasters] Grid layers to sum

Outputs

Sum [raster] The resulting layer.

Console usage

```
processing.runalg('saga:gridssum', grids, result)
```

See also

Grid standardisation

Description

Standardises the grid layer values.

Parameters

Grid [raster] Grid to process.

Stretch Factor [number] stretching factor.

Default: *1.0*

Outputs

Standardised Grid [raster] The resulting layer.

Console usage

```
processing.runalg('saga:gridstandardisation', input, stretch, output)
```

See also

Grid volume

Description

<put algorithm description here>

Parameters

Grid [raster] <put parameter description here>

Method [selection] <put parameter description here>

Options:

- 0 — [0] Count Only Above Base Level
- 1 — [1] Count Only Below Base Level
- 2 — [2] Subtract Volumes Below Base Level
- 3 — [3] Add Volumes Below Base Level

Default: 0

Base Level [number] <put parameter description here>

Default: 0.0

Outputs

Console usage

```
processing.runalg('saga:gridvolume', grid, method, level)
```

See also

Metric conversions

Description

Performs numerical conversions of the grid values.

Parameters

Grid [raster] Grid to process.

Conversion [selection] Conversion type.

Options:

- 0 — [0] radians to degree
- 1 — [1] degree to radians
- 2 — [2] Celsius to Fahrenheit
- 3 — [3] Fahrenheit to Celsius

Default: 0

Outputs

Converted Grid [raster] The resulting layer.

Console usage

```
processing.runalg('saga:metricconversions', grid, conversion, conv)
```

See also

Polynomial trend from grids

Description

<put algorithm description here>

Parameters

Dependent Variables [multipleinput: rasters] <put parameter description here>

Independent Variable (per Grid and Cell) [multipleinput: rasters] Optional.

<put parameter description here>

Independent Variable (per Grid) [fixedtable] <put parameter description here>

Type of Approximated Function [selection] <put parameter description here>

Options:

- 0 — [0] first order polynom (linear regression)
- 1 — [1] second order polynom
- 2 — [2] third order polynom
- 3 — [3] fourth order polynom
- 4 — [4] fifth order polynom

Default: 0

Outputs

Polynomial Coefficients [raster] <put output description here>

Coefficient of Determination [raster] <put output description here>

Console usage

```
processing.runalg('saga:polynomialtrendfromgrids', grids, y_grids, y_table, polynom, parms, quality)
```

See also

Random field

Description

Generates a random grid layer.

Parameters

Width (Cells) [number] Width of the layer in cells.

Default: *100*

Height (Cells) [number] Height of the layer in cells.

Default: *100*

Cellsize [number] Cell size to use.

Default: *100.0*

West [number] West coordinate of the bottom-left corner of the grid.

Default: *0.0*

South [number] South coordinate of the bottom-left corner of the grid.

Default: *0.0*

Method [selection] Statistical method used for the calculation.

Options:

- 0 — [0] Uniform
- 1 — [1] Gaussian

Default: *0*

Range Min [number] Minimum cell value to use.

Default: *0.0*

Range Max [number] Maximum cell value to use.

Default: *1.0*

Arithmetic Mean [number] Mean of all the cell values to use.

Default: *0.0*

Standard Deviation [number] Standard deviation of all the cell values to use.

Default: *1.0*

Outputs

Random Field [raster] The resulting layer.

Console usage

```
processing.runalg('saga:randomfield', nx, ny, cellsize, xmin, ymin, method, range_min, range_max,
```

See also

Random terrain generation

Description

<put algorithm description here>

Parameters

Radius (cells) [number] <put parameter description here>

Default: *10*

Iterations [number] <put parameter description here>

Default: *10*

Target Dimensions [selection] <put parameter description here>

Options:

- 0 — [0] User defined

Default: *0*

Grid Size [number] <put parameter description here>

Default: *1.0*

Cols [number] <put parameter description here>

Default: *100*

Rows [number] <put parameter description here>

Default: *100*

Outputs

Grid [raster] <put output description here>

Console usage

```
processing.runalg('saga:randomterraingeneration', radius, iterations, target_type, user_cell_size)
```

See also

Raster calculator

Description

<put algorithm description here>

Parameters

Main input layer [raster] <put parameter description here>

Additional layers [multipleinput: rasters] Optional.

<put parameter description here>

Formula [string] <put parameter description here>

Default: *(not set)*

Outputs

Result [raster] <put output description here>

Console usage

```
processing.runalg('saga:rastercalculator', grids, xgrids, formula, result)
```

See also

.

18.7.4 Grid filter

Dtm filter (slope-based)

Description

<put algorithm description here>

Parameters

Grid to filter [raster] <put parameter description here>

Search Radius [number] <put parameter description here>

Default: 2

Approx. Terrain Slope [number] <put parameter description here>

Default: 30.0

Use Confidence Interval [boolean] <put parameter description here>

Default: *True*

Outputs

Bare Earth [raster] <put output description here>

Removed Objects [raster] <put output description here>

Console usage

```
processing.runalg('saga:dtmfilterslopebased', input, radius, terrainslope, stddev, ground, nongro
```

See also

Filter clumps

Description

<put algorithm description here>

Parameters

Input Grid [raster] <put parameter description here>

Min. Size [number] <put parameter description here>

Default: 10

Outputs

Filtered Grid [raster] <put output description here>

Console usage

```
processing.runalg('saga:filterclumps', grid, threshold, output)
```

See also

Gaussian filter

Description

<put algorithm description here>

Parameters

Grid [raster] <put parameter description here>

Standard Deviation [number] <put parameter description here>

Default: 1

Search Mode [selection] <put parameter description here>

Options:

- 0 — [0] Square
- 1 — [1] Circle

Default: 0

Search Radius [number] <put parameter description here>

Default: 3

Outputs

Filtered Grid [raster] <put output description here>

Console usage

```
processing.runalg('saga:gaussianfilter', input, sigma, mode, radius, result)
```

See also

Laplacian filter

Description

<put algorithm description here>

Parameters

Grid [raster] <put parameter description here>

Method [selection] <put parameter description here>

Options:

- 0 — [0] standard kernel 1
- 1 — [1] standard kernel 2
- 2 — [2] Standard kernel 3
- 3 — [3] user defined kernel

Default: 0

Standard Deviation (Percent of Radius) [number] <put parameter description here>

Default: 0

Radius [number] <put parameter description here>

Default: 1

Search Mode [selection] <put parameter description here>

Options:

- 0 — [0] square
- 1 — [1] circle

Default: 0

Outputs

Filtered Grid [raster] <put output description here>

Console usage

```
processing.runalg('saga:laplacianfilter', input, method, sigma, radius, mode, result)
```

See also

Majority filter

Description

<put algorithm description here>

Parameters

Grid [raster] <put parameter description here>

Search Mode [selection] <put parameter description here>

Options:

- 0 — [0] Square
- 1 — [1] Circle

Default: 0

Radius [number] <put parameter description here>

Default: 1

Threshold [Percent] [number] <put parameter description here>

Default: 0

Outputs

Filtered Grid [raster] <put output description here>

Console usage

```
processing.runalg('saga:majorityfilter', input, mode, radius, threshold, result)
```

See also

Morphological filter

Description

<put algorithm description here>

Parameters

Grid [raster] <put parameter description here>

Search Mode [selection] <put parameter description here>

Options:

- 0 — [0] Square
- 1 — [1] Circle

Default: 0

Radius [number] <put parameter description here>

Default: *1*

Method [selection] <put parameter description here>

Options:

- 0 — [0] Dilation
- 1 — [1] Erosion
- 2 — [2] Opening
- 3 — [3] Closing

Default: *0*

Outputs

Filtered Grid [raster] <put output description here>

Console usage

```
processing.runalg('saga:morphologicalfilter', input, mode, radius, method, result)
```

See also

Multi direction lee filter

Description

<put algorithm description here>

Parameters

Grid [raster] <put parameter description here>

Estimated Noise (absolute) [number] <put parameter description here>

Default: *1.0*

Estimated Noise (relative) [number] <put parameter description here>

Default: *1.0*

Weighted [boolean] <put parameter description here>

Default: *True*

Method [selection] <put parameter description here>

Options:

- 0 — [0] noise variance given as absolute value
- 1 — [1] noise variance given relative to mean standard deviation
- 2 — [2] original calculation (Ringeler)

Default: *0*

Outputs

Filtered Grid [raster] <put output description here>

Minimum Standard Deviation [raster] <put output description here>

Direction of Minimum Standard Deviation [raster] <put output description here>

Console usage

```
processing.runalg('saga:multidirectionleefilter', input, noise_abs, noise_rel, weighted, method, ...)
```

See also

Rank filter

Description

<put algorithm description here>

Parameters

Grid [raster] <put parameter description here>

Search Mode [selection] <put parameter description here>

Options:

- 0 — [0] Square
- 1 — [1] Circle

Default: 0

Radius [number] <put parameter description here>

Default: 1

Rank [Percent] [number] <put parameter description here>

Default: 50

Outputs

Filtered Grid [raster] <put output description here>

Console usage

```
processing.runalg('saga:rankfilter', input, mode, radius, rank, result)
```

See also

Simple filter

Description

<put algorithm description here>

Parameters

Grid [raster] <put parameter description here>

Search Mode [selection] <put parameter description here>

Options:

- 0 — [0] Square
- 1 — [1] Circle

Default: 0

Filter [selection] <put parameter description here>

Options:

- 0 — [0] Smooth
- 1 — [1] Sharpen
- 2 — [2] Edge

Default: 0

Radius [number] <put parameter description here>

Default: 2

Outputs

Filtered Grid [raster] <put output description here>

Console usage

```
processing.runalg('saga:simplefilter', input, mode, method, radius, result)
```

See also

User defined filter

Description

<put algorithm description here>

Parameters

Grid [raster] <put parameter description here>

Filter Matrix [table] Optional.

<put parameter description here>

Default Filter Matrix (3x3) [fixedtable] <put parameter description here>

Outputs

Filtered Grid [raster] <put output description here>

Console usage

```
processing.runalg('saga:userdefinedfilter', input, filter, filter_3x3, result)
```

See also

.

18.7.5 Grid gridding

Inverse distance weighted

Description

Inverse distance grid interpolation from irregular distributed points.

Parameters

Points [**vector:** **point**] <put parameter description here>

Attribute [**tablefield:** **any**] <put parameter description here>

Target Grid [**selection**] <put parameter description here>

Options:

- 0 — [0] user defined

Default: 0

Distance Weighting [**selection**] <put parameter description here>

Options:

- 0 — [0] inverse distance to a power
- 1 — [1] linearly decreasing within search radius
- 2 — [2] exponential weighting scheme
- 3 — [3] gaussian weighting scheme

Default: 0

Inverse Distance Power [**number**] <put parameter description here>

Default: 2

Exponential and Gaussian Weighting Bandwidth [**number**] <put parameter description here>

Default: 1

Search Range [**selection**] <put parameter description here>

Options:

- 0 — [0] search radius (local)
- 1 — [1] no search radius (global)

Default: 0

Search Radius [**number**] <put parameter description here>

Default: 100.0

Search Mode [selection] <put parameter description here>

Options:

- 0 — [0] all directions
- 1 — [1] quadrants

Default: 0

Number of Points [selection] <put parameter description here>

Options:

- 0 — [0] maximum number of points
- 1 — [1] all points

Default: 0

Maximum Number of Points [number] <put parameter description here>

Default: 10

Output extent [extent] <put parameter description here>

Default: 0,1,0,1

Cellsize [number] <put parameter description here>

Default: 100.0

Outputs

Grid [raster] <put output description here>

Console usage

```
processing.runalg('saga:inversedistanceweighted', shapes, field, target, weighting, power, bandwi
```

See also

Kernel density estimation

Description

<put algorithm description here>

Parameters

Points [vector: point] <put parameter description here>

Weight [tablefield: any] <put parameter description here>

Radius [number] <put parameter description here>

Default: 10

Kernel [selection] <put parameter description here>

Options:

- 0 — [0] quartic kernel
- 1 — [1] gaussian kernel

Default: 0

Target Grid [selection] <put parameter description here>

Options:

- 0 — [0] user defined

Default: 0

Output extent [extent] <put parameter description here>

Default: 0,1,0,1

Cellsize [number] <put parameter description here>

Default: 100.0

Outputs

Grid [raster] <put output description here>

Console usage

```
processing.runalg('saga:kerneldensityestimation', points, population, radius, kernel, target, outp
```

See also

Modifed quadratic shepard

Description

<put algorithm description here>

Parameters

Points [vector: point] <put parameter description here>

Attribute [tablefield: any] <put parameter description here>

Target Grid [selection] <put parameter description here>

Options:

- 0 — [0] user defined

Default: 0

Quadratic Neighbors [number] <put parameter description here>

Default: 13

Weighting Neighbors [number] <put parameter description here>

Default: 19

Left [number] <put parameter description here>

Default: 0.0

Right [number] <put parameter description here>

Default: 0.0

Bottom [number] <put parameter description here>

Default: *0.0*

Top [number] <put parameter description here>

Default: *0.0*

Cellsize [number] <put parameter description here>

Default: *100.0*

Outputs

Grid [raster] <put output description here>

Console usage

```
processing.runalg('saga:modifiedquadraticshepard', shapes, field, target, quadratic_neighbors, wei
```

See also

Natural neighbour

Description

<put algorithm description here>

Parameters

Points [vector: point] <put parameter description here>

Attribute [tablefield: any] <put parameter description here>

Target Grid [selection] <put parameter description here>

Options:

- 0 — [0] user defined

Default: *0*

Sibson [boolean] <put parameter description here>

Default: *True*

Output extent [extent] <put parameter description here>

Default: *0,1,0,1*

Cellsize [number] <put parameter description here>

Default: *100.0*

Outputs

Grid [raster] <put output description here>

Console usage

```
processing.runalg('saga:naturalneighbour', shapes, field, target, sibson, output_extent, user_size)
```

See also

Nearest neighbour

Description

<put algorithm description here>

Parameters

Points [vector: point] <put parameter description here>

Attribute [tablefield: any] <put parameter description here>

Target Grid [selection] <put parameter description here>

Options:

- 0 — [0] user defined

Default: 0

Output extent [extent] <put parameter description here>

Default: 0,1,0,1

Cellsize [number] <put parameter description here>

Default: 100.0

Outputs

Grid [raster] <put output description here>

Console usage

```
processing.runalg('saga:nearestneighbour', shapes, field, target, output_extent, user_size, user_size)
```

See also

Shapes to grid

Description

<put algorithm description here>

Parameters

Shapes [vector: any] <put parameter description here>

Attribute [tablefield: any] <put parameter description here>

Method for Multiple Values [selection] <put parameter description here>

Options:

- 0 — [0] first
- 1 — [1] last
- 2 — [2] minimum
- 3 — [3] maximum
- 4 — [4] mean

Default: 0

Method for Lines [selection] <put parameter description here>

Options:

- 0 — [0] thin
- 1 — [1] thick

Default: 0

Preferred Target Grid Type [selection] <put parameter description here>

Options:

- 0 — [0] Integer (1 byte)
- 1 — [1] Integer (2 byte)
- 2 — [2] Integer (4 byte)
- 3 — [3] Floating Point (4 byte)
- 4 — [4] Floating Point (8 byte)

Default: 0

Output extent [extent] <put parameter description here>

Default: 0,1,0,1

Cellsize [number] <put parameter description here>

Default: 100.0

Outputs

Grid [raster] <put output description here>

Console usage

```
processing.runalg('saga:shapestogrid', input, field, multiple, line_type, grid_type, output_extent)
```

See also

Triangulation

Description

<put algorithm description here>

Parameters

Points [vector: point] <put parameter description here>

Attribute [tablefield: any] <put parameter description here>

Target Grid [selection] <put parameter description here>

Options:

- 0 — [0] user defined

Default: 0

Output extent [extent] <put parameter description here>

Default: 0,1,0,1

Cellsize [number] <put parameter description here>

Default: 100.0

Outputs

Grid [raster] <put output description here>

Console usage

```
processing.runalg('saga:triangulation', shapes, field, target, output_extent, user_size, user_grid)
```

See also

.

18.7.6 Grid spline

B-spline approximation

Description

<put algorithm description here>

Parameters

Points [vector: point] <put parameter description here>

Attribute [tablefield: any] <put parameter description here>

Target Grid [selection] <put parameter description here>

Options:

- 0 — [0] user defined

Default: 0

Resolution [number] <put parameter description here>

Default: 1.0

Output extent [extent] <put parameter description here>

Default: 0,1,0,1

Cellsize [number] <put parameter description here>

Default: 100.0

Outputs

Grid [raster] <put output description here>

Console usage

```
processing.runalg('saga:bsplineapproximation', shapes, field, target, level, output_extent, user_
```

See also

Cubic spline approximation

Description

<put algorithm description here>

Parameters

Points [vector: point] <put parameter description here>

Attribute [tablefield: any] <put parameter description here>

Target Grid [selection] <put parameter description here>

Options:

- 0 — [0] user defined

Default: 0

Minimal Number of Points [number] <put parameter description here>

Default: 3

Maximal Number of Points [number] <put parameter description here>

Default: 20

Points per Square [number] <put parameter description here>

Default: 5

Tolerance [number] <put parameter description here>

Default: 140.0

Output extent [extent] <put parameter description here>

Default: 0,1,0,1

Cellsize [number] <put parameter description here>

Default: 100.0

Outputs

Grid [raster] <put output description here>

Console usage

```
processing.runalg('saga:cubicsplineapproximation', shapes, field, target, npmin, npmax, nppc, k, c
```

See also

Multilevel b-spline interpolation (from grid)

Description

<put algorithm description here>

Parameters

Grid [raster] <put parameter description here>

Target Grid [selection] <put parameter description here>

Options:

- 0 — [0] user defined

Default: 0

Method [selection] <put parameter description here>

Options:

- 0 — [0] without B-spline refinement
- 1 — [1] with B-spline refinement

Default: 0

Threshold Error [number] <put parameter description here>

Default: 0.0001

Maximum Level [number] <put parameter description here>

Default: 11.0

Data Type [selection] <put parameter description here>

Options:

- 0 — [0] same as input grid
- 1 — [1] floating point

Default: 0

Output extent [extent] <put parameter description here>

Default: 0,1,0,1

Cellsize [number] <put parameter description here>

Default: 100.0

Outputs

Grid [raster] <put output description here>

Console usage

```
processing.runalg('saga:multilevelbsplineinterpolationfromgrid', gridpoints, target, method, epsi
```

See also

Multilevel b-spline interpolation

Description

<put algorithm description here>

Parameters

Points [vector: point] <put parameter description here>

Attribute [tablefield: any] <put parameter description here>

Target Grid [selection] <put parameter description here>

Options:

- 0 — [0] user defined

Default: 0

Method [selection] <put parameter description here>

Options:

- 0 — [0] without B-spline refinement
- 1 — [1] with B-spline refinement

Default: 0

Threshold Error [number] <put parameter description here>

Default: 0.0001

Maximum Level [number] <put parameter description here>

Default: 11.0

Output extent [extent] <put parameter description here>

Default: 0,1,0,1

Cellsize [number] <put parameter description here>

Default: *100.0*

Outputs

Grid [raster] <put output description here>

Console usage

```
processing.runalg('saga:multilevelbsplineinterpolation', shapes, field, target, method, epsilon, .
```

See also

Thin plate spline (global)

Description

<put algorithm description here>

Parameters

Points [vector: point] <put parameter description here>

Attribute [tablefield: any] <put parameter description here>

Target Grid [selection] <put parameter description here>

Options:

- 0 — [0] user defined

Default: *0*

Regularisation [number] <put parameter description here>

Default: *0.0*

Output extent [extent] <put parameter description here>

Default: *0,1,0,1*

Cellsize [number] <put parameter description here>

Default: *100.0*

Outputs

Grid [raster] <put output description here>

Console usage

```
processing.runalg('saga:thinplatesplineglobal', shapes, field, target, regul, output_extent, user.
```

See also

Thin plate spline (local)

Description

<put algorithm description here>

Parameters

Points [vector: point] <put parameter description here>

Attribute [tablefield: any] <put parameter description here>

Target Grid [selection] <put parameter description here>

Options:

- 0 — [0] user defined

Default: 0

Regularisation [number] <put parameter description here>

Default: 0.0001

Search Radius [number] <put parameter description here>

Default: 100.0

Search Mode [selection] <put parameter description here>

Options:

- 0 — [0] all directions
- 1 — [1] quadrants

Default: 0

Points Selection [selection] <put parameter description here>

Options:

- 0 — [0] all points in search radius
- 1 — [1] maximum number of points

Default: 0

Maximum Number of Points [number] <put parameter description here>

Default: 10

Output extent [extent] <put parameter description here>

Default: 0,1,0,1

Cellsize [number] <put parameter description here>

Default: 100.0

Outputs

Grid [raster] <put output description here>

Console usage

```
processing.runalg('saga:thinplatesplinelocal', shapes, field, target, regul, radius, mode, select,
```

See also

Thin plate spline (tin)

Description

<put algorithm description here>

Parameters

Points [vector: point] <put parameter description here>

Attribute [tablefield: any] <put parameter description here>

Target Grid [selection] <put parameter description here>

Options:

- 0 — [0] user defined

Default: 0

Regularisation [number] <put parameter description here>

Default: 0.0

Neighbourhood [selection] <put parameter description here>

Options:

- 0 — [0] immediate
- 1 — [1] level 1
- 2 — [2] level 2

Default: 0

Add Frame [boolean] <put parameter description here>

Default: True

Output extent [extent] <put parameter description here>

Default: 0,1,0,1

Cellsize [number] <put parameter description here>

Default: 100.0

Outputs

Grid [raster] <put output description here>

Console usage

```
processing.runalg('saga:thinplatesplinetin', shapes, field, target, regul, level, frame, output_e
```

See also

.

18.7.7 Grid tools

Aggregate

Description

<put algorithm description here>

Parameters

Grid [**raster**] <put parameter description here>

Aggregation Size [**number**] <put parameter description here>

Default: 3

Method [**selection**] <put parameter description here>

Options:

- 0 — [0] Sum
- 1 — [1] Min
- 2 — [2] Max

Default: 0

Outputs

Console usage

```
processing.runalg('saga:aggregate', input, size, method)
```

See also

Change grid values

Description

<put algorithm description here>

Parameters

Grid [**raster**] <put parameter description here>

Replace Condition [**selection**] <put parameter description here>

Options:

- 0 — [0] Grid value equals low value
- 1 — [1] Low value < grid value < high value

- 2 — [2] Low value \leq grid value $<$ high value

Default: 0

Lookup Table [fixedtable] <put parameter description here>

Outputs

Changed Grid [raster] <put output description here>

Console usage

```
processing.runalg('saga:changegridvalues', grid_in, method, lookup, grid_out)
```

See also

Close gaps

Description

<put algorithm description here>

Parameters

Grid [raster] <put parameter description here>

Mask [raster] Optional.

<put parameter description here>

Tension Threshold [number] <put parameter description here>

Default: 0.1

Outputs

Changed Grid [raster] <put output description here>

Console usage

```
processing.runalg('saga:closegaps', input, mask, threshold, result)
```

See also

Close gaps with spline

Description

<put algorithm description here>

Parameters

Grid [raster] <put parameter description here>

Mask [raster] Optional.

<put parameter description here>

Only Process Gaps with Less Cells [number] <put parameter description here>

Default: *0*

Maximum Points [number] <put parameter description here>

Default: *1000*

Number of Points for Local Interpolation [number] <put parameter description here>

Default: *10*

Extended Neighbourhood [boolean] <put parameter description here>

Default: *True*

Neighbourhood [selection] <put parameter description here>

Options:

- 0 — [0] Neumann
- 1 — [1] Moore

Default: *0*

Radius (Cells) [number] <put parameter description here>

Default: *0*

Relaxation [number] <put parameter description here>

Default: *0.0*

Outputs

Closed Gaps Grid [raster] <put output description here>

Console usage

```
processing.runalg('saga:closegapswithspline', grid, mask, maxgapcells, maxpoints, localpoints, ex
```

See also

Close one cell gaps

Description

<put algorithm description here>

Parameters

Grid [raster] <put parameter description here>

Outputs

Changed Grid [raster] <put output description here>

Console usage

```
processing.runalg('saga:closeonecellgaps', input, result)
```

See also

Convert data storage type

Description

<put algorithm description here>

Parameters

Grid [raster] <put parameter description here>

Data storage type [selection] <put parameter description here>

Options:

- 0 — [0] bit
- 1 — [1] unsigned 1 byte integer
- 2 — [2] signed 1 byte integer
- 3 — [3] unsigned 2 byte integer
- 4 — [4] signed 2 byte integer
- 5 — [5] unsigned 4 byte integer
- 6 — [6] signed 4 byte integer
- 7 — [7] 4 byte floating point number
- 8 — [8] 8 byte floating point number

Default: 0

Outputs

Converted Grid [raster] <put output description here>

Console usage

```
processing.runalg('saga:convertdatastoragetype', input, type, output)
```

See also

Crop to data

Description

<put algorithm description here>

Parameters

Input layer [raster] <put parameter description here>

Outputs

Cropped layer [raster] <put output description here>

Console usage

```
processing.runalg('saga:croptodata', input, output)
```

See also

Grid buffer

Description

<put algorithm description here>

Parameters

Features Grid [raster] <put parameter description here>

Distance [number] <put parameter description here>

Default: *1000*

Buffer Distance [selection] <put parameter description here>

Options:

- 0 — [0] Fixed
- 1 — [1] Cell value

Default: *0*

Outputs

Buffer Grid [raster] <put output description here>

Console usage

```
processing.runalg('saga:gridbuffer', features, dist, buffertype, buffer)
```

See also

Grid masking

Description

<put algorithm description here>

Parameters

Grid [raster] <put parameter description here>

Mask [raster] <put parameter description here>

Outputs

Masked Grid [raster] <put output description here>

Console usage

```
processing.runalg('saga:gridmasking', grid, mask, masked)
```

See also

Grid orientation

Description

<put algorithm description here>

Parameters

Grid [raster] <put parameter description here>

Method [selection] <put parameter description here>

Options:

- 0 — [0] Copy
- 1 — [1] Flip
- 2 — [2] Mirror
- 3 — [3] Invert

Default: 0

Outputs

Changed Grid [raster] <put output description here>

Console usage

```
processing.runalg('saga:gridorientation', input, method, result)
```

See also

Grid proximity buffer

Description

<put algorithm description here>

Parameters

Source Grid [raster] <put parameter description here>

Buffer distance [number] <put parameter description here>

Default: *500.0*

Equidistance [number] <put parameter description here>

Default: *100.0*

Outputs

Distance Grid [raster] <put output description here>

Allocation Grid [raster] <put output description here>

Buffer Grid [raster] <put output description here>

Console usage

```
processing.runalg('saga:gridproximitybuffer', source, dist, ival, distance, alloc, buffer)
```

See also

Grid shrink/expand

Description

<put algorithm description here>

Parameters

Grid [raster] <put parameter description here>

Operation [selection] <put parameter description here>

Options:

- 0 — [0] Shrink
- 1 — [1] Expand

Default: 0

Search Mode [selection] <put parameter description here>

Options:

- 0 — [0] Square
- 1 — [1] Circle

Default: 0

Radius [number] <put parameter description here>

Default: 1

Method [selection] <put parameter description here>

Options:

- 0 — [0] min
- 1 — [1] max
- 2 — [2] mean
- 3 — [3] majority

Default: 0

Outputs

Result Grid [raster] <put output description here>

Console usage

```
processing.runalg('saga:gridshrinkexpand', input, operation, mode, radius, method_expand, result)
```

See also

Invert data/no-data

Description

<put algorithm description here>

Parameters

Grid [raster] <put parameter description here>

Outputs

Result [raster] <put output description here>

Console usage

```
processing.runalg('saga:invertdatanodata', input, output)
```

See also

Merge raster layers

Description

<put algorithm description here>

Parameters

Grids to Merge [**multipleinput: rasters**] <put parameter description here>

Preferred data storage type [**selection**] <put parameter description here>

Options:

- 0 — [0] 1 bit
- 1 — [1] 1 byte unsigned integer
- 2 — [2] 1 byte signed integer
- 3 — [3] 2 byte unsigned integer
- 4 — [4] 2 byte signed integer
- 5 — [5] 4 byte unsigned integer
- 6 — [6] 4 byte signed integer
- 7 — [7] 4 byte floating point
- 8 — [8] 8 byte floating point

Default: 0

Interpolation [**selection**] <put parameter description here>

Options:

- 0 — [0] Nearest Neighbor
- 1 — [1] Bilinear Interpolation
- 2 — [2] Inverse Distance Interpolation
- 3 — [3] Bicubic Spline Interpolation
- 4 — [4] B-Spline Interpolation

Default: 0

Overlapping Cells [**selection**] <put parameter description here>

Options:

- 0 — [0] mean value
- 1 — [1] first value in order of grid list

Default: 0

Outputs

Merged Grid [**raster**] <put output description here>

Console usage

```
processing.runalg('saga:mergerasterlayers', grids, type, interpol, overlap, merged)
```

See also

Patching

Description

<put algorithm description here>

Parameters

Grid [raster] <put parameter description here>

Patch Grid [raster] <put parameter description here>

Interpolation Method [selection] <put parameter description here>

Options:

- 0 — [0] Nearest Neighbor
- 1 — [1] Bilinear Interpolation
- 2 — [2] Inverse Distance Interpolation
- 3 — [3] Bicubic Spline Interpolation
- 4 — [4] B-Spline Interpolation

Default: 0

Outputs

Completed Grid [raster] <put output description here>

Console usage

```
processing.runalg('saga:patching', original, additional, interpolation, completed)
```

See also

Proximity grid

Description

<put algorithm description here>

Parameters

Features [raster] <put parameter description here>

Outputs

Distance [raster] <put output description here>

Direction [raster] <put output description here>

Allocation [raster] <put output description here>

Console usage

```
processing.runalg('saga:proximitygrid', features, distance, direction, allocation)
```

See also

Reclassify grid values

Description

<put algorithm description here>

Parameters

Grid [raster] <put parameter description here>

Method [selection] <put parameter description here>

Options:

- 0 — [0] single
- 1 — [1] range
- 2 — [2] simple table

Default: 0

old value (for single value change) [number] <put parameter description here>

Default: 0.0

new value (for single value change) [number] <put parameter description here>

Default: 1.0

operator (for single value change) [selection] <put parameter description here>

Options:

- 0 — [0] =
- 1 — [1] <
- 2 — [2] <=
- 3 — [3] >=
- 4 — [4] >

Default: 0

minimum value (for range) [number] <put parameter description here>

Default: 0.0

maximum value (for range) [number] <put parameter description here>

Default: *1.0*

new value (for range) [number] <put parameter description here>

Default: *2.0*

operator (for range) [selection] <put parameter description here>

Options:

- 0 — [0] <=
- 1 — [1] <

Default: *0*

Lookup Table [fixedtable] <put parameter description here>

operator (for table) [selection] <put parameter description here>

Options:

- 0 — [0] min <= value < max
- 1 — [1] min <= value <= max
- 2 — [2] min < value <= max
- 3 — [3] min < value < max

Default: *0*

replace no data values [boolean] <put parameter description here>

Default: *True*

new value for no data values [number] <put parameter description here>

Default: *0.0*

replace other values [boolean] <put parameter description here>

Default: *True*

new value for other values [number] <put parameter description here>

Default: *0.0*

Outputs

Reclassified Grid [raster] <put output description here>

Console usage

```
processing.runalg('saga:reclassifygridvalues', input, method, old, new, soperator, min, max, rnew,
```

See also

Resampling

Description

<put algorithm description here>

Parameters

Grid [raster] <put parameter description here>

Preserve Data Type [boolean] <put parameter description here>

Default: *True*

Target Grid [selection] <put parameter description here>

Options:

- 0 — [0] user defined

Default: *0*

Interpolation Method (Scale Up) [selection] <put parameter description here>

Options:

- 0 — [0] Nearest Neighbor
- 1 — [1] Bilinear Interpolation
- 2 — [2] Inverse Distance Interpolation
- 3 — [3] Bicubic Spline Interpolation
- 4 — [4] B-Spline Interpolation
- 5 — [5] Mean Value
- 6 — [6] Mean Value (cell area weighted)
- 7 — [7] Minimum Value
- 8 — [8] Maximum Value
- 9 — [9] Majority

Default: *0*

Interpolation Method (Scale Down) [selection] <put parameter description here>

Options:

- 0 — [0] Nearest Neighbor
- 1 — [1] Bilinear Interpolation
- 2 — [2] Inverse Distance Interpolation
- 3 — [3] Bicubic Spline Interpolation
- 4 — [4] B-Spline Interpolation

Default: *0*

Output extent [extent] <put parameter description here>

Default: *0,1,0,1*

Cellsize [number] <put parameter description here>

Default: *100.0*

Outputs

Grid [raster] <put output description here>

Console usage

```
processing.runalg('saga:resampling', input, keep_type, target, scale_up_method, scale_down_method)
```

See also

Sort grid

Description

<put algorithm description here>

Parameters

Input Grid [raster] <put parameter description here>

Down sort [boolean] <put parameter description here>

Default: *True*

Outputs

Sorted Grid [raster] <put output description here>

Console usage

```
processing.runalg('saga:sortgrid', grid, down, output)
```

See also

Split RGB bands

Description

<put algorithm description here>

Parameters

Input layer [raster] <put parameter description here>

Outputs

Output R band layer [raster] <put output description here>

Output G band layer [raster] <put output description here>

Output B band layer [raster] <put output description here>

Console usage

```
processing.runalg('saga:splitrgrbbands', input, r, g, b)
```

See also

Threshold buffer

Description

<put algorithm description here>

Parameters

Features Grid [raster] <put parameter description here>

Value Grid [raster] <put parameter description here>

Threshold Grid [raster] Optional.

<put parameter description here>

Threshold [number] <put parameter description here>

Default: *0.0*

Threshold Type [selection] <put parameter description here>

Options:

- 0 — [0] Absolute
- 1 — [1] Relative from cell value

Default: *0*

Outputs

Buffer Grid [raster] <put output description here>

Console usage

```
processing.runalg('saga:thresholdbuffer', features, value, thresholdgrid, threshold, thresholdtype)
```

See also

.

18.7.8 Grid visualization

Histogram surface

Description

<put algorithm description here>

Parameters

Grid [**raster**] <put parameter description here>

Method [**selection**] <put parameter description here>

Options:

- 0 — [0] rows
- 1 — [1] columns
- 2 — [2] circle

Default: 0

Outputs

Histogram [**raster**] <put output description here>

Console usage

```
processing.runalg('saga:histogramsurface', grid, method, hist)
```

See also

Rgb composite

Description

<put algorithm description here>

Parameters

R [**raster**] <put parameter description here>

G [**raster**] <put parameter description here>

B [**raster**] <put parameter description here>

Method for R value [**selection**] <put parameter description here>

Options:

- 0 — 0 - 255
- 1 — Rescale to 0 - 255
- 2 — User defined rescale
- 3 — Percentiles
- 4 — Percentage of standard deviation

Default: 0

Method for G value [**selection**] <put parameter description here>

Options:

- 0 — 0 - 255
- 1 — Rescale to 0 - 255

- 2 — User defined rescale
- 3 — Percentiles
- 4 — Percentage of standard deviation

Default: 0

Method for B value [selection] <put parameter description here>

Options:

- 0 — 0 - 255
- 1 — Rescale to 0 - 255
- 2 — User defined rescale
- 3 — Percentiles
- 4 — Percentage of standard deviation

Default: 0

Rescale Range for RED min [number] <put parameter description here>

Default: 0

Rescale Range for RED max [number] <put parameter description here>

Default: 255

Percentiles Range for RED max [number] <put parameter description here>

Default: 1

Percentiles Range for RED max [number] <put parameter description here>

Default: 99

Percentage of standard deviation for RED [number] <put parameter description here>

Default: 150.0

Rescale Range for GREEN min [number] <put parameter description here>

Default: 0

Rescale Range for GREEN max [number] <put parameter description here>

Default: 255

Percentiles Range for GREEN max [number] <put parameter description here>

Default: 1

Percentiles Range for GREEN max [number] <put parameter description here>

Default: 99

Percentage of standard deviation for GREEN [number] <put parameter description here>

Default: 150.0

Rescale Range for BLUE min [number] <put parameter description here>

Default: 0

Rescale Range for BLUE max [number] <put parameter description here>

Default: 255

Percentiles Range for BLUE max [number] <put parameter description here>

Default: 1

Percentiles Range for BLUE max [number] <put parameter description here>

Default: *99*

Percentage of standard deviation for BLUE [number] <put parameter description here>

Default: *150.0*

Outputs

Output RGB [raster] <put output description here>

Console usage

```
processing.runalg('saga:rgbcomposite', grid_r, grid_g, grid_b, r_method, g_method, b_method, r_ra
```

See also

.

18.7.9 Imagery classification

Change detection

Description

<put algorithm description here>

Parameters

Initial State [raster] <put parameter description here>

Look-up Table [table] Optional.

<put parameter description here>

Value [tablefield: any] <put parameter description here>

Value (Maximum) [tablefield: any] <put parameter description here>

Name [tablefield: any] <put parameter description here>

Final State [raster] <put parameter description here>

Look-up Table [table] Optional.

<put parameter description here>

Value [tablefield: any] <put parameter description here>

Value (Maximum) [tablefield: any] <put parameter description here>

Name [tablefield: any] <put parameter description here>

Report Unchanged Classes [boolean] <put parameter description here>

Default: *True*

Output as... [selection] <put parameter description here>

Options:

- 0 — [0] cells
- 1 — [1] percent
- 2 — [2] area

Default: 0

Outputs

Changes [raster] <put output description here>

Changes [table] <put output description here>

Console usage

```
processing.runalg('saga:changedetection', initial, ini_lut, ini_lut_min, ini_lut_max, ini_lut_name,
```

See also

Cluster analysis for grids

Description

<put algorithm description here>

Parameters

Grids [multipleinput: rasters] <put parameter description here>

Method [selection] <put parameter description here>

Options:

- 0 — [0] Iterative Minimum Distance (Forgy 1965)
- 1 — [1] Hill-Climbing (Rubin 1967)
- 2 — [2] Combined Minimum Distance / Hillclimbing

Default: 0

Clusters [number] <put parameter description here>

Default: 5

Normalise [boolean] <put parameter description here>

Default: *True*

Old Version [boolean] <put parameter description here>

Default: *True*

Outputs

Clusters [raster] <put output description here>

Statistics [table] <put output description here>

Console usage

```
processing.runalg('saga:clusteranalysisforgrids', grids, method, ncluster, normalise, oldversion,
```

See also

Supervised classification

Description

<put algorithm description here>

Parameters

Grids [**multipleinput: rasters**] <put parameter description here>

Training Areas [**vector: polygon**] <put parameter description here>

Class Identifier [**tablefield: any**] <put parameter description here>

Method [**selection**] <put parameter description here>

Options:

- 0 — [0] Binary Encoding
- 1 — [1] Parallelepiped
- 2 — [2] Minimum Distance
- 3 — [3] Mahalanobis Distance
- 4 — [4] Maximum Likelihood
- 5 — [5] Spectral Angle Mapping
- 6 — [6] Winner Takes All

Default: *0*

Normalise [**boolean**] <put parameter description here>

Default: *True*

Distance Threshold [**number**] <put parameter description here>

Default: *0.0*

Probability Threshold (Percent) [**number**] <put parameter description here>

Default: *0.0*

Probability Reference [**selection**] <put parameter description here>

Options:

- 0 — [0] absolute
- 1 — [1] relative

Default: *0*

Spectral Angle Threshold (Degree) [**number**] <put parameter description here>

Default: *0.0*

Outputs

Class Information [table] <put output description here>

Classification [raster] <put output description here>

Quality [raster] <put output description here>

Console usage

```
processing.runalg('saga:supervisedclassification', grids, roi, roi_id, method, normalise, thresho
```

See also

.

18.7.10 Imagery RGA

Fast region growing algorithm

Description

<put algorithm description here>

Parameters

Input Grids [multipleinput: rasters] <put parameter description here>

Seeds Grid [raster] <put parameter description here>

Smooth Rep [raster] Optional.

<put parameter description here>

Outputs

Segmente [raster] <put output description here>

Mean [raster] <put output description here>

Console usage

```
processing.runalg('saga:fastregiongrowingalgorithm', input, start, rep, result, mean)
```

See also

.

18.7.11 Imagery segmentation

Grid skeletonization

Description

<put algorithm description here>

Parameters

Grid [raster] <put parameter description here>

Method [selection] <put parameter description here>

Options:

- 0 — [0] Standard
- 1 — [1] Hilditch's Algorithm
- 2 — [2] Channel Skeleton

Default: 0

Initialisation [selection] <put parameter description here>

Options:

- 0 — [0] Less than
- 1 — [1] Greater than

Default: 0

Threshold (Init.) [number] <put parameter description here>

Default: 0.0

Convergence [number] <put parameter description here>

Default: 3.0

Outputs

Skeleton [raster] <put output description here>

Skeleton [vector] <put output description here>

Console usage

```
processing.runalg('saga:gridskeletonization', input, method, init_method, init_threshold, converg
```

See also

Seed generation

Description

<put algorithm description here>

Parameters

Features [**multipleinput: rasters**] <put parameter description here>

Bandwidth (Cells) [**number**] <put parameter description here>

Default: 2

Type of Surface [**selection**] <put parameter description here>

Options:

- 0 — [0] smoothed surface
- 1 — [1] variance (a)
- 2 — [2] variance (b)

Default: 0

Extraction of... [**selection**] <put parameter description here>

Options:

- 0 — [0] minima
- 1 — [1] maxima
- 2 — [2] minima and maxima

Default: 0

Feature Aggregation [**selection**] <put parameter description here>

Options:

- 0 — [0] additive
- 1 — [1] multiplicative

Default: 0

Normalized [**boolean**] <put parameter description here>

Default: *True*

Outputs

Surface [**raster**] <put output description here>

Seeds Grid [**raster**] <put output description here>

Seeds [**vector**] <put output description here>

Console usage

```
processing.runalg('saga:seedgeneration', grids, factor, type_surface, type_seeds, type_merge, norm
```

See also

Simple region growing

Description

<put algorithm description here>

Parameters

Seeds [**raster**] <put parameter description here>

Features [**multipleinput: rasters**] <put parameter description here>

Method [**selection**] <put parameter description here>

Options:

- 0 — [0] feature space and position
- 1 — [1] feature space

Default: *0*

Neighbourhood [**selection**] <put parameter description here>

Options:

- 0 — [0] 4 (von Neumann)
- 1 — [1] 8 (Moore)

Default: *0*

Variance in Feature Space [**number**] <put parameter description here>

Default: *1.0*

Variance in Position Space [**number**] <put parameter description here>

Default: *1.0*

Threshold - Similarity [**number**] <put parameter description here>

Default: *0.0*

Refresh [**boolean**] <put parameter description here>

Default: *True*

Leaf Size (for Speed Optimisation) [**number**] <put parameter description here>

Default: *256*

Outputs

Segments [**raster**] <put output description here>

Similarity [**raster**] <put output description here>

Seeds [**table**] <put output description here>

Console usage

```
processing.runalg('saga:simpleregiongrowing', seeds, features, method, neighbour, sig_1, sig_2, t
```

See also

Watershed segmentation

Description

<put algorithm description here>

Parameters

Grid [raster] <put parameter description here>

Output [selection] <put parameter description here>

Options:

- 0 — [0] Seed Value
- 1 — [1] Segment ID

Default: 0

Method [selection] <put parameter description here>

Options:

- 0 — [0] Minima
- 1 — [1] Maxima

Default: 0

Join Segments based on Threshold Value [selection] <put parameter description here>

Options:

- 0 — [0] do not join
- 1 — [1] seed to saddle difference
- 2 — [2] seeds difference

Default: 0

Threshold [number] <put parameter description here>

Default: 0

Allow Edge Pixels to be Seeds [boolean] <put parameter description here>

Default: *True*

Borders [boolean] <put parameter description here>

Default: *True*

Outputs

Segments [raster] <put output description here>

Seed Points [vector] <put output description here>

Borders [raster] <put output description here>

Console usage

```
processing.runalg('saga:watershedsegmentation', grid, output, down, join, threshold, edge, bborder)
```

See also

.

18.7.12 Imagery tools

Vegetation index[distance based]

Description

<put algorithm description here>

Parameters

Near Infrared Band [raster] <put parameter description here>

Red Band [raster] <put parameter description here>

Slope of the soil line [number] <put parameter description here>

Default: *0.0*

Intercept of the soil line [number] <put parameter description here>

Default: *0.0*

Outputs

PVI (Richardson and Wiegand) [raster] <put output description here>

PVI (Perry & Lautenschlager) [raster] <put output description here>

PVI (Walther & Shabaani) [raster] <put output description here>

PVI (Qi, et al) [raster] <put output description here>

Console usage

```
processing.runalg('saga:vegetationindexdistancebased', nir, red, slope, intercept, pvi, pvi1, pvi2)
```

See also

Vegetation index[slope based]

Description

<put algorithm description here>

Parameters

Near Infrared Band [raster] <put parameter description here>

Red Band [raster] <put parameter description here>

Outputs

Normalized Difference Vegetation Index [raster] <put output description here>

Ratio Vegetation Index [raster] <put output description here>

Transformed Vegetation Index [raster] <put output description here>

Corrected Transformed Vegetation Index [raster] <put output description here>

Thiam's Transformed Vegetation Index [raster] <put output description here>

Normalized Ratio Vegetation Index [raster] <put output description here>

Console usage

```
processing.runalg('saga:vegetationindexslopebased', nir, red, ndvi, ratio, tvi, ctvi, ttvi, nratio)
```

See also

.

18.7.13 Kriging

Ordinary kriging (global)

Description

<put algorithm description here>

Parameters

Points [vector: point] <put parameter description here>

Attribute [tablefield: any] <put parameter description here>

Create Variance Grid [boolean] <put parameter description here>

Default: *True*

Target Grid [selection] <put parameter description here>

Options:

- 0 — [0] user defined

Default: *0*

Variogram Model [selection] <put parameter description here>

Options:

- 0 — [0] Spherical Model
- 1 — [1] Exponential Model
- 2 — [2] Gaussian Model
- 3 — [3] Linear Regression
- 4 — [4] Exponential Regression
- 5 — [5] Power Function Regression

Default: 0

Block Kriging [boolean] <put parameter description here>

Default: *True*

Block Size [number] <put parameter description here>

Default: 100

Logarithmic Transformation [boolean] <put parameter description here>

Default: *True*

Nugget [number] <put parameter description here>

Default: 0.0

Sill [number] <put parameter description here>

Default: 0.0

Range [number] <put parameter description here>

Default: 0.0

Linear Regression [number] <put parameter description here>

Default: 1.0

Exponential Regression [number] <put parameter description here>

Default: 0.1

Power Function - A [number] <put parameter description here>

Default: 1.0

Power Function - B [number] <put parameter description here>

Default: 0.5

Grid Size [number] <put parameter description here>

Default: 1.0

Fit Extent [boolean] <put parameter description here>

Default: *True*

Output extent [extent] <put parameter description here>

Default: 0,1,0,1

Outputs

Grid [raster] <put output description here>

Variance [raster] <put output description here>

Console usage

```
processing.runalg('saga:ordinarykrigingglobal', shapes, field, bvariance, target, model, block, d
```

See also

Ordinary kriging

Description

<put algorithm description here>

Parameters

Points [**vector: point**] <put parameter description here>

Attribute [**tablefield: any**] <put parameter description here>

Create Variance Grid [**boolean**] <put parameter description here>

Default: *True*

Target Grid [**selection**] <put parameter description here>

Options:

- 0 — [0] user defined

Default: *0*

Variogram Model [**selection**] <put parameter description here>

Options:

- 0 — [0] Spherical Model
- 1 — [1] Exponential Model
- 2 — [2] Gaussian Model
- 3 — [3] Linear Regression
- 4 — [4] Exponential Regression
- 5 — [5] Power Function Regression

Default: *0*

Block Kriging [**boolean**] <put parameter description here>

Default: *True*

Block Size [**number**] <put parameter description here>

Default: *100*

Logarithmic Transformation [**boolean**] <put parameter description here>

Default: *True*

Nugget [**number**] <put parameter description here>

Default: *0.0*

Sill [**number**] <put parameter description here>

Default: *10.0*

Range [**number**] <put parameter description here>

Default: *100.0*

Linear Regression [**number**] <put parameter description here>

Default: *1.0*

Exponential Regression [number] <put parameter description here>

Default: *0.1*

Power Function - A [number] <put parameter description here>

Default: *1*

Power Function - B [number] <put parameter description here>

Default: *0.5*

Maximum Search Radius (map units) [number] <put parameter description here>

Default: *1000.0*

Min. Number of m_Points [number] <put parameter description here>

Default: *4*

Max. Number of m_Points [number] <put parameter description here>

Default: *20*

Grid Size [number] <put parameter description here>

Default: *1.0*

Fit Extent [boolean] <put parameter description here>

Default: *True*

Output extent [extent] <put parameter description here>

Default: *0,1,0,1*

Outputs

Grid [raster] <put output description here>

Variance [raster] <put output description here>

Console usage

```
processing.runalg('saga:ordinarykriging', shapes, field, bvariance, target, model, block, dblock,
```

See also

Universal kriging (global)

Description

<put algorithm description here>

Parameters

Points [vector: point] <put parameter description here>

Attribute [tablefield: any] <put parameter description here>

Create Variance Grid [boolean] <put parameter description here>

Default: *True*

Target Grid [selection] <put parameter description here>

Options:

- 0 — [0] user defined

Default: 0

Variogram Model [selection] <put parameter description here>

Options:

- 0 — [0] Spherical Model
- 1 — [1] Exponential Model
- 2 — [2] Gaussian Model
- 3 — [3] Linear Regression
- 4 — [4] Exponential Regression
- 5 — [5] Power Function Regression

Default: 0

Block Kriging [boolean] <put parameter description here>

Default: *True*

Block Size [number] <put parameter description here>

Default: 100

Logarithmic Transformation [boolean] <put parameter description here>

Default: *True*

Nugget [number] <put parameter description here>

Default: 0.0

Sill [number] <put parameter description here>

Default: 0.0

Range [number] <put parameter description here>

Default: 0.0

Linear Regression [number] <put parameter description here>

Default: 1

Exponential Regression [number] <put parameter description here>

Default: 0.5

Power Function - A [number] <put parameter description here>

Default: 1.0

Power Function - B [number] <put parameter description here>

Default: 0.1

Grids [multipleinput: rasters] <put parameter description here>

Grid Interpolation [selection] <put parameter description here>

Options:

- 0 — [0] Nearest Neighbor
- 1 — [1] Bilinear Interpolation
- 2 — [2] Inverse Distance Interpolation

- 3 — [3] Bicubic Spline Interpolation
- 4 — [4] B-Spline Interpolation

Default: *0*

Grid Size [number] <put parameter description here>

Default: *1.0*

Fit Extent [boolean] <put parameter description here>

Default: *True*

Output extent [extent] <put parameter description here>

Default: *0,1,0,1*

Outputs

Grid [raster] <put output description here>

Variance [raster] <put output description here>

Console usage

```
processing.runalg('saga:universalkrigingglobal', shapes, field, bvariance, target, model, block, ...)
```

See also

Universal kriging

Description

<put algorithm description here>

Parameters

Points [vector: point] <put parameter description here>

Attribute [tablefield: any] <put parameter description here>

Create Variance Grid [boolean] <put parameter description here>

Default: *True*

Target Grid [selection] <put parameter description here>

Options:

- 0 — [0] user defined

Default: *0*

Variogram Model [selection] <put parameter description here>

Options:

- 0 — [0] Spherical Model
- 1 — [1] Exponential Model
- 2 — [2] Gaussian Model
- 3 — [3] Linear Regression

- 4 — [4] Exponential Regression
- 5 — [5] Power Function Regression

Default: 0

Block Kriging [boolean] <put parameter description here>

Default: *True*

Block Size [number] <put parameter description here>

Default: 100

Logarithmic Transformation [boolean] <put parameter description here>

Default: *True*

Nugget [number] <put parameter description here>

Default: 0.0

Sill [number] <put parameter description here>

Default: 0.0

Range [number] <put parameter description here>

Default: 0.0

Linear Regression [number] <put parameter description here>

Default: 1.0

Exponential Regression [number] <put parameter description here>

Default: 0.1

Power Function - A [number] <put parameter description here>

Default: 1

Power Function - B [number] <put parameter description here>

Default: 0.5

Grids [multipleinput: rasters] <put parameter description here>

Grid Interpolation [selection] <put parameter description here>

Options:

- 0 — [0] Nearest Neighbor
- 1 — [1] Bilinear Interpolation
- 2 — [2] Inverse Distance Interpolation
- 3 — [3] Bicubic Spline Interpolation
- 4 — [4] B-Spline Interpolation

Default: 0

Min.Number of m_Points [number] <put parameter description here>

Default: 4

Max. Number of m_Points [number] <put parameter description here>

Default: 20

Maximum Search Radius (map units) [number] <put parameter description here>

Default: 1000.0

Grid Size [number] <put parameter description here>

Default: *1.0*

Fit Extent [boolean] <put parameter description here>

Default: *True*

Output extent [extent] <put parameter description here>

Default: *0,1,0,1*

Outputs

Grid [raster] <put output description here>

Variance [raster] <put output description here>

Console usage

```
processing.runalg('saga:universalkriging', shapes, field, bvariance, target, model, block, dblock,
```

See also

.

18.7.14 Shapes grid

Add grid values to points

Description

Creates a new vector layer as a result of the union of a points layer with the interpolated value of one or more base background grid layer(s). This way, the new layer created will have a new column in the attribute table that reflects the interpolated value of the background grid.

Parameters

Points [vector: point] Input layer.

Grids [multipleinput: rasters] Background grid layer(s)

Interpolation [selection] interpolation method to use.

Options:

- 0 — [0] Nearest Neighbor
- 1 — [1] Bilinear Interpolation
- 2 — [2] Inverse Distance Interpolation
- 3 — [3] Bicubic Spline Interpolation
- 4 — [4] B-Spline Interpolation

Default: *0*

Outputs

Result [vector] The resulting layer.

Console usage

```
processing.runalg('saga:addgridvaluestopoints', shapes, grids, interpol, result)
```

See also

Add grid values to shapes

Description

<put algorithm description here>

Parameters

Shapes [vector: any] <put parameter description here>

Grids [multipleinput: rasters] <put parameter description here>

Interpolation [selection] <put parameter description here>

Options:

- 0 — [0] Nearest Neighbor
- 1 — [1] Bilinear Interpolation
- 2 — [2] Inverse Distance Interpolation
- 3 — [3] Bicubic Spline Interpolation
- 4 — [4] B-Spline Interpolation

Default: 0

Outputs

Result [vector] <put output description here>

Console usage

```
processing.runalg('saga:addgridvaluestoshapes', shapes, grids, interpol, result)
```

See also

Clip grid with polygon

Description

<put algorithm description here>

Parameters

Input [raster] <put parameter description here>

Polygons [vector: polygon] <put parameter description here>

Outputs

Output [raster] <put output description here>

Console usage

```
processing.runalg('saga:clipgridwithpolygon', input, polygons, output)
```

See also

Contour lines from grid

Description

<put algorithm description here>

Parameters

Grid [raster] <put parameter description here>

Minimum Contour Value [number] <put parameter description here>

Default: *0.0*

Maximum Contour Value [number] <put parameter description here>

Default: *10000.0*

Equidistance [number] <put parameter description here>

Default: *100.0*

Outputs

Contour Lines [vector] <put output description here>

Console usage

```
processing.runalg('saga:contourlinesfromgrid', input, zmin, zmax, zstep, contour)
```

See also

Gradient vectors from directional components

Description

<put algorithm description here>

Parameters

X Component [raster] <put parameter description here>

Y Component [raster] <put parameter description here>

Step [number] <put parameter description here>

Default: *1*

Size Range Min [number] <put parameter description here>

Default: *25.0*

Size Range Max [number] <put parameter description here>

Default: *100.0*

Aggregation [selection] <put parameter description here>

Options:

- 0 — [0] nearest neighbour
- 1 — [1] mean value

Default: *0*

Style [selection] <put parameter description here>

Options:

- 0 — [0] simple line
- 1 — [1] arrow
- 2 — [2] arrow (centered to cell)

Default: *0*

Outputs

Gradient Vectors [vector] <put output description here>

Console usage

```
processing.runalg('saga:gradientvectorsfromdirectionalcomponents', x, y, step, size_min, size_max)
```

See also

Gradient vectors from direction and length

Description

<put algorithm description here>

Parameters

Direction [raster] <put parameter description here>

Length [raster] <put parameter description here>

Step [number] <put parameter description here>

Default: 1

Size Range Min [number] <put parameter description here>

Default: 25.0

Size Range Max [number] <put parameter description here>

Default: 100.0

Aggregation [selection] <put parameter description here>

Options:

- 0 — [0] nearest neighbour
- 1 — [1] mean value

Default: 0

Style [selection] <put parameter description here>

Options:

- 0 — [0] simple line
- 1 — [1] arrow
- 2 — [2] arrow (centered to cell)

Default: 0

Outputs

Gradient Vectors [vector] <put output description here>

Console usage

```
processing.runalg('saga:gradientvectorsfromdirectionandlength', dir, len, step, size_min, size_max)
```

See also

Gradient vectors from surface

Description

<put algorithm description here>

Parameters

Surface [raster] <put parameter description here>

Step [number] <put parameter description here>

Default: 1

Size Range Min [number] <put parameter description here>

Default: 25.0

Size Range Max [number] <put parameter description here>

Default: 100.0

Aggregation [selection] <put parameter description here>

Options:

- 0 — [0] nearest neighbour
- 1 — [1] mean value

Default: 0

Style [selection] <put parameter description here>

Options:

- 0 — [0] simple line
- 1 — [1] arrow
- 2 — [2] arrow (centered to cell)

Default: 0

Outputs

Gradient Vectors [vector] <put output description here>

Console usage

```
processing.runalg('saga:gradientvectorsfromsurface', surface, step, size_min, size_max, aggr, sty
```

See also

Grid statistics for polygons

Description

<put algorithm description here>

Parameters

Grids [multipleinput: rasters] <put parameter description here>

Polygons [vector: polygon] <put parameter description here>

Number of Cells [boolean] <put parameter description here>

Default: *True*

Minimum [boolean] <put parameter description here>

Default: *True*

Maximum [boolean] <put parameter description here>

Default: *True*

Range [boolean] <put parameter description here>

Default: *True*

Sum [boolean] <put parameter description here>

Default: *True*

Mean [boolean] <put parameter description here>

Default: *True*

Variance [boolean] <put parameter description here>

Default: *True*

Standard Deviation [boolean] <put parameter description here>

Default: *True*

Quantiles [number] <put parameter description here>

Default: *0*

Outputs

Statistics [vector] <put output description here>

Console usage

```
processing.runalg('saga:gridstatisticsforpolygons', grids, polygons, count, min, max, range, sum,
```

See also

Grid values to points (randomly)

Description

<put algorithm description here>

Parameters

Grid [raster] <put parameter description here>

Frequency [number] <put parameter description here>

Default: *100*

Outputs

Points [vector] <put output description here>

Console usage

```
processing.runalg('saga:gridvaluestopointsrandomly', grid, freq, points)
```

See also

Grid values to points

Description

<put algorithm description here>

Parameters

Grids [**multipleinput: rasters**] <put parameter description here>

Polygons [**vector: any**] Optional.

<put parameter description here>

Exclude NoData Cells [**boolean**] <put parameter description here>

Default: *True*

Type [**selection**] <put parameter description here>

Options:

- 0 — [0] nodes
- 1 — [1] cells

Default: *0*

Outputs

Shapes [**vector**] <put output description here>

Console usage

```
processing.runalg('saga:gridvaluestopoints', grids, polygons, nodata, type, shapes)
```

See also

Local minima and maxima

Description

<put algorithm description here>

Parameters

Grid [**raster**] <put parameter description here>

Outputs

Minima [**vector**] <put output description here>

Maxima [**vector**] <put output description here>

Console usage

```
processing.runalg('saga:localminimaandmaxima', grid, minima, maxima)
```

See also

Vectorising grid classes

Description

<put algorithm description here>

Parameters

Grid [raster] <put parameter description here>

Class Selection [selection] <put parameter description here>

Options:

- 0 — [0] one single class specified by class identifier
- 1 — [1] all classes

Default: 0

Class Identifier [number] <put parameter description here>

Default: 0

Vectorised class as... [selection] <put parameter description here>

Options:

- 0 — [0] one single (multi-)polygon object
- 1 — [1] each island as separated polygon

Default: 0

Outputs

Polygons [vector] <put output description here>

Console usage

```
processing.runalg('saga:vectorisinggridclasses', grid, class_all, class_id, split, polygons)
```

See also

.

18.7.15 Shapes lines

Convert points to line(s)

Description

Converts points to lines.

Parameters

Points [vector: point] Points to convert.

Order by... [tablefield: any] Lines will be ordered following this field.

Separate by... [tablefield: any] Lines will be grouped according to this field.

Outputs

Lines [vector] The resulting layer.

Console usage

```
processing.runalg('saga:convertpointstolines', points, order, separate, lines)
```

See also

Convert polygons to lines

Description

Creates lines from polygons.

Parameters

Polygons [vector: polygon] Layer to process.

Outputs

Lines [vector] The resulting layer.

Console usage

```
processing.runalg('saga:convertpolygonstolines', polygons, lines)
```

See also

Line dissolve

Description

<put algorithm description here>

Parameters

Lines [vector: any] <put parameter description here>

1. **Attribute** [tablefield: any] <put parameter description here>
2. **Attribute** [tablefield: any] <put parameter description here>

3. Attribute [tablefield: any] <put parameter description here>

Dissolve... [selection] <put parameter description here>

Options:

- 0 — [0] lines with same attribute value(s)
- 1 — [1] all lines

Default: 0

Outputs

Dissolved Lines [vector] <put output description here>

Console usage

```
processing.runalg('saga:linedissolve', lines, field_1, field_2, field_3, all, dissolved)
```

See also

Line-polygon intersection

Description

<put algorithm description here>

Parameters

Lines [vector: line] <put parameter description here>

Polygons [vector: polygon] <put parameter description here>

Output [selection] <put parameter description here>

Options:

- 0 — [0] one multi-line per polygon
- 1 — [1] keep original line attributes

Default: 0

Outputs

Intersection [vector] <put output description here>

Console usage

```
processing.runalg('saga:linepolygonintersection', lines, polygons, method, intersect)
```

See also

Line properties

Description

Calculates some information on each line of the layer.

Parameters

Lines [**vector: line**] Layer to analyze.

Number of Parts [**boolean**] Determines whether to calculate number of segments in line.

Default: *True*

Number of Vertices [**boolean**] Determines whether to calculate number of vertices in line.

Default: *True*

Length [**boolean**] Determines whether to calculate total line length.

Default: *True*

Outputs

Lines with Property Attributes [**vector**] The resulting layer.

Console usage

```
processing.runalg('saga:lineproperties', lines, bparts, bpoints, blength, output)
```

See also

Line simplification

Description

Simplifies the geometry of a lines layer.

Parameters

Lines [**vector: line**] Layer to process.

Tolerance [**number**] Simplification tolerance.

Default: *1.0*

Outputs

Simplified Lines [**vector**] The resulting layer.

Console usage

```
processing.runalg('saga:linesimplification', lines, tolerance, output)
```

See also

.

18.7.16 Shapes points

Add coordinates to points

Description

Adds the X and Y coordinates of feature in the attribute table of input layer.

Parameters

Points [vector: point] Input layer.

Outputs

Output [vector] Resulting layer with the updated attribute table.

Console usage

```
processing.runalg('saga:addcoordinatestopoints', input, output)
```

See also

Add polygon attributes to points

Description

Adds the specified field of the polygons layer to the attribute table of the points layer. The new attributes added for each point depend on the value of the background polygon layer.

Parameters

Points [vector: point] Points layer.

Polygons [vector: polygon] Background polygons layer.

Attribute [tablefield: any] Attribute of the polygons layer that will be added to the points layer.

Outputs

Result [vector] The resulting layer.

Console usage

```
processing.runalg('saga:addpolygonattributestopoints', input, polygons, field, output)
```

See also

Aggregate point observations

Description

<put algorithm description here>

Parameters

Reference Points [vector: any] <put parameter description here>

ID [tablefield: any] <put parameter description here>

Observations [table] <put parameter description here>

X [tablefield: any] <put parameter description here>

Y [tablefield: any] <put parameter description here>

Track [tablefield: any] <put parameter description here>

Date [tablefield: any] <put parameter description here>

Time [tablefield: any] <put parameter description here>

Parameter [tablefield: any] <put parameter description here>

Maximum Time Span (Seconds) [number] <put parameter description here>

Default: *60.0*

Maximum Distance [number] <put parameter description here>

Default: *0.002*

Outputs

Aggregated [table] <put output description here>

Console usage

```
processing.runalg('saga:aggregatepointobservations', reference, reference_id, observations, x, y,
```

See also

Clip points with polygons

Description

<put algorithm description here>

Parameters

Points [vector: point] <put parameter description here>

Polygons [vector: polygon] <put parameter description here>

Add Attribute to Clipped Points [tablefield: any] <put parameter description here>

Clipping Options [selection] <put parameter description here>

Options:

- 0 — [0] one layer for all points
- 1 — [1] separate layer for each polygon

Default: 0

Outputs

Clipped Points [vector] <put output description here>

Console usage

```
processing.runalg('saga:clippointswithpolygons', points, polygons, field, method, clips)
```

See also

Convert lines to points

Description

Converts lines layer into a points.

Parameters

Lines [vector: line] Lines layer to convert.

Insert Additional Points [boolean] Determines whether to add additional nodes or not.

Default: *True*

Insert Distance [number] Distance between the additional points.

Default: *1.0*

Outputs

Points [vector] The resulting layer.

Console usage

```
processing.runalg('saga:convertlinestopoints', lines, add, dist, points)
```

See also

Convert multipoints to points

Description

<put algorithm description here>

Parameters

Multipoints [**vector: point**] <put parameter description here>

Outputs

Points [**vector**] <put output description here>

Console usage

```
processing.runalg('saga:convertmultipointstopoints', multipoints, points)
```

See also

Convex hull

Description

<put algorithm description here>

Parameters

Points [**vector: point**] <put parameter description here>

Hull Construction [**selection**] <put parameter description here>

Options:

- 0 — [0] one hull for all shapes
- 1 — [1] one hull per shape
- 2 — [2] one hull per shape part

Default: 0

Outputs

Convex Hull [**vector**] <put output description here>

Minimum Bounding Box [**vector**] <put output description here>

Console usage

```
processing.runalg('saga:convexhull', shapes, polypoints, hulls, boxes)
```

See also

Distance matrix

Description

Generates a distance matrix between each point of the input layer. A unique ID will be created in the first row of the resulting matrix (symmetric matrix), while every other cell reflects the distance between the points.

Parameters

Points [vector: point] Input layer.

Outputs

Distance Matrix Table [table] The resulting table.

Console usage

```
processing.runalg('saga:distancematrix', points, table)
```

See also

Fit n points to shape

Description

<put algorithm description here>

Parameters

Shapes [vector: polygon] <put parameter description here>

Number of points [number] <put parameter description here>

Default: 10

Outputs

Points [vector] <put output description here>

Console usage

```
processing.runalg('saga:fitnpointstoshape', shapes, numpoints, points)
```

See also

Points filter

Description

<put algorithm description here>

Parameters

Points [**vector: point**] <put parameter description here>

Attribute [**tablefield: any**] <put parameter description here>

Radius [**number**] <put parameter description here>

Default: *1*

Minimum Number of Points [**number**] <put parameter description here>

Default: *0*

Maximum Number of Points [**number**] <put parameter description here>

Default: *0*

Quadrants [**boolean**] <put parameter description here>

Default: *True*

Filter Criterion [**selection**] <put parameter description here>

Options:

- 0 — [0] keep maxima (with tolerance)
- 1 — [1] keep minima (with tolerance)
- 2 — [2] remove maxima (with tolerance)
- 3 — [3] remove minima (with tolerance)
- 4 — [4] remove below percentile
- 5 — [5] remove above percentile

Default: *0*

Tolerance [**number**] <put parameter description here>

Default: *0.0*

Percentile [**number**] <put parameter description here>

Default: *50*

Outputs

Filtered Points [**vector**] <put output description here>

Console usage

```
processing.runalg('saga:pointsfilter', points, field, radius, minnum, maxnum, quadrants, method, t
```

See also

Points thinning

Description

<put algorithm description here>

Parameters

Points [**vector: point**] <put parameter description here>

Attribute [**tablefield: any**] <put parameter description here>

Resolution [**number**] <put parameter description here>

Default: *1.0*

Outputs

Thinned Points [**vector**] <put output description here>

Console usage

```
processing.runalg('saga:pointsthinning', points, field, resolution, thinned)
```

See also

Remove duplicate points

Description

<put algorithm description here>

Parameters

Points [**vector: any**] <put parameter description here>

Attribute [**tablefield: any**] <put parameter description here>

Point to Keep [**selection**] <put parameter description here>

Options:

- 0 — [0] first point
- 1 — [1] last point
- 2 — [2] point with minimum attribute value
- 3 — [3] point with maximum attribute value

Default: *0*

Numeric Attribute Values [**selection**] <put parameter description here>

Options:

- 0 — [0] take value from the point to be kept

- 1 — [1] minimum value of all duplicates
- 2 — [2] maximum value of all duplicates
- 3 — [3] mean value of all duplicates

Default: 0

Outputs

Result [vector] <put output description here>

Console usage

```
processing.runalg('saga:removeduplicatepoints', points, field, method, numeric, result)
```

See also

Separate points by direction

Description

<put algorithm description here>

Parameters

Points [vector: point] <put parameter description here>

Number of Directions [number] <put parameter description here>

Default: 4

Tolerance (Degree) [number] <put parameter description here>

Default: 5

Outputs

Output [vector] <put output description here>

Console usage

```
processing.runalg('saga:separatepointsbydirection', points, directions, tolerance, output)
```

See also

.

18.7.17 Shapes polygons

Convert lines to polygons

Description

Converts lines to polygons.

Parameters

Lines [vector: line] Lines to convert.

Outputs

Polygons [vector] The resulting layer.

Console usage

```
processing.runalg('saga:convertlinestopolygons', lines, polygons)
```

See also

Convert polygon/line vertices to points

Description

Converts the line or polygon vertices into points.

Parameters

Shapes [vector: any] Layer to process.

Outputs

Points [vector] The resulting layer.

Console usage

```
processing.runalg('saga:convertpolygonlineverticestopoints', shapes, points)
```

See also

Polygon centroids

Description

Calculates the centroids of polygons.

Parameters

Polygons [vector: polygon] Input layer.

Centroids for each part [boolean] Determines whether centroids should be calculated for each part of multipart polygon or not.

Default: *True*

Outputs

Centroids [vector] The resulting layer.

Console usage

```
processing.runalg('saga:polygoncentroids', polygons, method, centroids)
```

See also

Polygon dissolve

Description

<put algorithm description here>

Parameters

Polygons [vector: polygon] <put parameter description here>

1. **Attribute** [tablefield: any] Optional.

<put parameter description here>

2. **Attribute** [tablefield: any] Optional.

<put parameter description here>

3. **Attribute** [tablefield: any] Optional.

<put parameter description here>

Dissolve... [selection] <put parameter description here>

Options:

- 0 — [0] polygons with same attribute value
- 1 — [1] all polygons
- 2 — [2] polygons with same attribute value (keep inner boundaries)
- 3 — [3] all polygons (keep inner boundaries)

Default: *0*

Outputs

Dissolved Polygons [vector] <put output description here>

Console usage

```
processing.runalg('saga:polygondissolve', polygons, field_1, field_2, field_3, dissolve, dissolve)
```

See also

Polygon-line intersection

Description

<put algorithm description here>

Parameters

Polygons [vector: polygon] <put parameter description here>

Lines [vector: line] <put parameter description here>

Outputs

Intersection [vector] <put output description here>

Console usage

```
processing.runalg('saga:polygonlineintersection', polygons, lines, intersect)
```

See also

Polygon parts to separate polygons

Description

<put algorithm description here>

Parameters

Polygons [vector: polygon] <put parameter description here>

Ignore Lakes [boolean] <put parameter description here>

Default: *True*

Outputs

Polygon Parts [vector] <put output description here>

Console usage

```
processing.runalg('saga:polygonpartstoseparatepolygons', polygons, lakes, parts)
```

See also

Polygon properties

Description

<put algorithm description here>

Parameters

Polygons [**vector: polygon**] <put parameter description here>

Number of Parts [**boolean**] <put parameter description here>

Default: *True*

Number of Vertices [**boolean**] <put parameter description here>

Default: *True*

Perimeter [**boolean**] <put parameter description here>

Default: *True*

Area [**boolean**] <put parameter description here>

Default: *True*

Outputs

Polygons with Property Attributes [**vector**] <put output description here>

Console usage

```
processing.runalg('saga:polygonproperties', polygons, bparts, bpoints, blength, barea, output)
```

See also

Polygon shape indices

Description

Calculates spatial statistics for polygons. This includes:

- area
- perimeter
- perimeter / area
- perimeter / square root of the area
- maximum distance
- maximum distance / area
- maximum distance / square root of the area
- shape index

Parameters

Shapes [vector: polygon] Layer to analyze.

Outputs

Shape Index [vector] The resulting layer.

Console usage

```
processing.runalg('saga:polygonshapeindices', shapes, index)
```

See also

Polygons to edges and nodes

Description

Extracts boundaries and nodes of polygons in separate files.

Parameters

Polygons [vector: polygon] Input layer.

Outputs

Edges [vector] Resulting line layer with polygons boundaries.

Nodes [vector] Resulting line layer with polygons nodes.

Console usage

```
processing.runalg('saga:polygonstoedgesandnodes', polygons, edges, nodes)
```

See also

.

18.7.18 Shapes tools

Create graticule

Description

Creates a grid.

Parameters

Extent [**vector: any**] Optional.

Grid will be created according to the selected layer.

Output extent [**extent**] Extent of the grid.

Default: *0,1,0,1*

Division Width [**number**] X-axes spacing between the lines.

Default: *1.0*

Division Height [**number**] Y-axes spacing between the lines.

Default: *1.0*

Type [**selection**] Geometry type of the resulting grid.

Options:

- 0 — [0] Lines
- 1 — [1] Rectangles

Default: *0*

Outputs

Graticule [**vector**] The resulting layer.

Console usage

```
processing.runalg('saga:creategraticule', extent, output_extent, distx, disty, type, graticule)
```

See also

Cut shapes layer

Description

<put algorithm description here>

Parameters

Vector layer to cut [**vector: any**] <put parameter description here>

Method [**selection**] <put parameter description here>

Options:

- 0 — [0] completely contained
- 1 — [1] intersects
- 2 — [2] center

Default: *0*

Cutting polygons [**vector: any**] <put parameter description here>

Outputs

Result [vector] <put output description here>

Extent [vector] <put output description here>

Console usage

```
processing.runalg('saga:cutshapeslayer', shapes, method, polygons_polygons, cut, extent)
```

See also

Get shapes extents

Description

Creates polygons according to the extent of the input layer features.

Parameters

Shapes [vector: any] Input layer.

Parts [boolean] Determines whether create polygon for each feature (`True`) or just create single polygon for whole layer (`False`).

Default: *True*

Outputs

Extents [vector] The resulting layer.

Console usage

```
processing.runalg('saga:getshapesextents', shapes, parts, extents)
```

See also

Merge shapes layers

Description

Merges two or more input layer into a unique resulting layer. You can merge together only layer of the same type (polygons with polygons, lines with lines, points with points).

The attribute table of the resulting layer will include only the attributes of the first input layer. Two additional columns will be added: one corresponding to the ID of every merged layer and the other one corresponding to the original name of the merged layer.

Parameters

Main Layer [vector: any] Initial layer.

Additional Layers [multipleinput: any vectors] Optional.

Layer(s) to merge with.

Outputs

Merged Layer [vector] The resulting layer.

Console usage

```
processing.runalg('saga:mergeshapelayers', main, layers, out)
```

See also

Polar to cartesian coordinates

Description

<put algorithm description here>

Parameters

Polar Coordinates [vector: any] <put parameter description here>

Exaggeration [tablefield: any] <put parameter description here>

Exaggeration Factor [number] <put parameter description here>

Default: *1*

Radius [number] <put parameter description here>

Default: *6371000.0*

Degree [boolean] <put parameter description here>

Default: *True*

Outputs

Cartesian Coordinates [vector] <put output description here>

Console usage

```
processing.runalg('saga:polartocartesiancoordinates', polar, f_exagg, d_exagg, radius, degree, ca
```

See also

Quadtree structure to shapes

Description

<put algorithm description here>

Parameters

Shapes [**vector: any**] <put parameter description here>

Attribute [**tablefield: any**] <put parameter description here>

Outputs

Polygons [**vector**] <put output description here>

Lines [**vector**] <put output description here>

Duplicated Points [**vector**] <put output description here>

Console usage

```
processing.runalg('saga:quadtreestructuretoshapes', shapes, attribute, polygons, lines, points)
```

See also

Shapes buffer

Description

Creates buffer around features based on fixed distance or distance field.

Parameters

Shapes [**vector: any**] Input layer.

Buffer Distance [**selection**] Buffering method.

Options:

- 0 — [0] fixed value
- 1 — [1] attribute field

Default: 0

Buffer Distance (Fixed) [**number**] Buffer distance for “fixed value” method.

Default: 100.0

Buffer Distance (Attribute) [**tablefield: any**] Name of the distance field for “attribute field” method.

Scaling Factor for Attribute Value [**number**] <put parameter description here>

Default: 1.0

Number of Buffer Zones [number] Number of buffer(s) to generate.

Default: *1.0*

Circle Point Distance [Degree] [number] Smoothness of the buffer borders: great numbers means rough borders.

Default: *5.0*

Dissolve Buffers [boolean] Determines whether to dissolve results or not.

Default: *True*

Outputs

Buffer [vector] The resulting layer.

Console usage

```
processing.runalg('saga:shapesbuffer', shapes, buf_type, buf_dist, buf_field, buf_scale, buf_zones)
```

See also

Split shapes layer randomly

Description

Splits the input layer randomly in two parts.

Parameters

Shapes [vector: any] Layer to split.

Split ratio (%) [number] Split ratio between the resulting layers.

Default: *50*

Outputs

Group A [vector] First resulting layer.

Group B [vector] Second resulting layer.

Console usage

```
processing.runalg('saga:splitshapeslayerrandomly', shapes, percent, a, b)
```

See also

Transform shapes

Description

<put algorithm description here>

Parameters

Shapes [**vector: any**] <put parameter description here>

dX [**number**] <put parameter description here>

Default: *0.0*

dY [**number**] <put parameter description here>

Default: *0.0*

Angle [**number**] <put parameter description here>

Default: *0.0*

Scale Factor X [**number**] <put parameter description here>

Default: *1.0*

Scale Factor Y [**number**] <put parameter description here>

Default: *1.0*

X [**number**] <put parameter description here>

Default: *0.0*

Y [**number**] <put parameter description here>

Default: *0.0*

Outputs

Output [**vector**] <put output description here>

Console usage

```
processing.runalg('saga:transformshapes', in, dx, dy, angle, scalex, scaley, anchorx, anchory, out)
```

See also

.

18.7.19 Shapes transect

Transect through polygon shapefile

Description

<put algorithm description here>

Parameters

Line Transect (s) [**vector: line**] <put parameter description here>

Theme [**vector: any**] <put parameter description here>

Theme Field [**tablefield: any**] <put parameter description here>

Outputs

Result table [table] <put output description here>

Console usage

```
processing.runalg('saga:transectthroughpolygonshapefile', transect, theme, theme_field, transect_
```

See also

.

18.7.20 Simulation fire

Fire risk analysis

Description

<put algorithm description here>

Parameters

DEM [raster] <put parameter description here>

Fuel Model [raster] <put parameter description here>

Wind Speed [raster] <put parameter description here>

Wind Direction [raster] <put parameter description here>

Dead Fuel Moisture 1H [raster] <put parameter description here>

Dead Fuel Moisture 10H [raster] <put parameter description here>

Dead Fuel Moisture 100H [raster] <put parameter description here>

Herbaceous Fuel Moisture [raster] <put parameter description here>

Wood Fuel Moisture [raster] <put parameter description here>

Value [raster] Optional.

<put parameter description here>

Base Probability [raster] Optional.

<put parameter description here>

Number of Events [number] <put parameter description here>

Default: *1000*

Fire Length [number] <put parameter description here>

Default: *100*

Outputs

Danger [raster] <put output description here>

Compound Probability [raster] <put output description here>

Priority Index [raster] <put output description here>

Console usage

```
processing.runalg('saga:fireriskanalysis', dem, fuel, windspd, winddir, m1h, m10h, m100h, mherb, m
```

See also

Simulation

Description

<put algorithm description here>

Parameters

DEM [raster] <put parameter description here>

Fuel Model [raster] <put parameter description here>

Wind Speed [raster] <put parameter description here>

Wind Direction [raster] <put parameter description here>

Dead Fuel Moisture 1H [raster] <put parameter description here>

Dead Fuel Moisture 10H [raster] <put parameter description here>

Dead Fuel Moisture 100H [raster] <put parameter description here>

Herbaceous Fuel Moisture [raster] <put parameter description here>

Wood Fuel Moisture [raster] <put parameter description here>

Ignition Points [raster] <put parameter description here>

Update View [boolean] <put parameter description here>

Default: *True*

Outputs

Time [raster] <put output description here>

Flame Length [raster] <put output description here>

Intensity [raster] <put output description here>

Console usage

```
processing.runalg('saga:simulation', dem, fuel, windspd, winddir, m1h, m10h, m100h, mherb, mwood,
```

See also

18.7.21 Simulation hydrology

Overland flow - kinematic wave d8

Description

<put algorithm description here>

Parameters

Elevation [raster] <put parameter description here>

Gauges [vector: any] Optional.

<put parameter description here>

Simulation Time [h] [number] <put parameter description here>

Default: 24

Simulation Time Step [h] [number] <put parameter description here>

Default: 0.1

Manning's Roughness [number] <put parameter description here>

Default: 0.03

Max. Iterations [number] <put parameter description here>

Default: 100

Epsilon [number] <put parameter description here>

Default: 0.0001

Precipitation [selection] <put parameter description here>

Options:

- 0 — [0] Homogenous
- 1 — [1] Above Elevation
- 2 — [2] Left Half

Default: 0

Threshold Elevation [number] <put parameter description here>

Default: 0.0

Outputs

Runoff [raster] <put output description here>

Flow at Gauges [table] <put output description here>

Console usage

```
processing.runalg('saga:overlandflowkinematicwaved8', dem, gauges, time_span, time_step, roughnes
```

See also

Water retention capacity

Description

<put algorithm description here>

Parameters

Plot Holes [vector: any] <put parameter description here>

DEM [raster] <put parameter description here>

Outputs

Final Parameters [vector] <put output description here>

Water Retention Capacity [raster] <put output description here>

Console usage

```
processing.runalg('saga:waterretentioncapacity', shapes, dem, output, retention)
```

See also

.

18.7.22 Table calculus

Fill gaps in records

Description

<put algorithm description here>

Parameters

Table [table] <put parameter description here>

Order [tablefield: any] <put parameter description here>

Interpolation [selection] <put parameter description here>

Options:

- 0 — [0] Nearest Neighbour
- 1 — [1] Linear

- 2 — [2] Spline

Default: 0

Outputs

Table without Gaps [table] <put output description here>

Console usage

```
processing.runalg('saga:fillgapsinrecords', table, order, method, nogaps)
```

See also

Principle components analysis

Description

<put algorithm description here>

Parameters

Table [table] <put parameter description here>

Method [selection] <put parameter description here>

Options:

- 0 — [0] correlation matrix
- 1 — [1] variance-covariance matrix
- 2 — [2] sums-of-squares-and-cross-products matrix

Default: 0

Number of Components [number] <put parameter description here>

Default: 3

Outputs

Principle Components [table] <put output description here>

Console usage

```
processing.runalg('saga:principlecomponentsanalysis', table, method, nfirst, pca)
```

See also

Running average

Description

<put algorithm description here>

Parameters

Input [table] <put parameter description here>

Attribute [tablefield: any] <put parameter description here>

Number of Records [number] <put parameter description here>

Default: 10

Outputs

Output [table] <put output description here>

Console usage

```
processing.runalg('saga:runningaverage', input, field, count, output)
```

See also

.

18.7.23 Table tools

Change date format

Description

Converts the date format of the input layer.

Parameters

Table [table] Input table.

Date Field [tablefield: any] Attribute the date.

Input Format [selection] Input date format.

Options:

- 0 — [0] dd.mm.yy
- 1 — [1] yy.mm.dd
- 2 — [2] dd:mm:yy
- 3 — [3] yy:mm:dd
- 4 — [4] ddmmyyyy, fix size
- 5 — [5] yyyymmdd, fix size
- 6 — [6] ddmmyy, fix size
- 7 — [7] yymmdd, fix size
- 8 — [8] Julian Day

Default: 0

Output Format [selection] Output date format.

Options:

- 0 — [0] dd.mm.yy
- 1 — [1] yy.mm.dd
- 2 — [2] dd:mm:yy
- 3 — [3] yy:mm:dd
- 4 — [4] ddmmyyyy, fix size
- 5 — [5] yyyyymmdd, fix size
- 6 — [6] ddmmyy, fix size
- 7 — [7] yymmdd, fix size
- 8 — [8] Julian Day

Default: 0

Outputs

Output [table] The resulting table.

Console usage

```
processing.runalg('saga:changedateformat', table, field, fmt_in, fmt_out, output)
```

See also

Change time format

Description

Converts the time format of the input layer.

Parameters

Table [table] Input table.

Time Field [tablefield: any] Attribute with time.

Input Format [selection] Input time format.

Options:

- 0 — [0] hh.mm.ss
- 1 — [1] hh:mm:ss
- 2 — [2] hhmmss, fix size
- 3 — [3] hours
- 4 — [4] minutes
- 5 — [5] seconds

Default: 0

Output Format [selection] Output time format.

Options:

- 0 — [0] hh.mm.ss
- 1 — [1] hh:mm:ss
- 2 — [2] hhmmss, fix size
- 3 — [3] hours
- 4 — [4] minutes
- 5 — [5] seconds

Default: 0

Outputs

Output [table] The resulting table.

Console usage

```
processing.runalg('saga:changetimeformat', table, field, fmt_in, fmt_out, output)
```

See also

.

18.7.24 Terrain channels

Channel network and drainage basins

Description

<put algorithm description here>

Parameters

Elevation [raster] <put parameter description here>

Threshold [number] <put parameter description here>

Default: 5.0

Outputs

Flow Direction [raster] <put output description here>

Flow Connectivity [raster] <put output description here>

Strahler Order [raster] <put output description here>

Drainage Basins [raster] <put output description here>

Channels [vector] <put output description here>

Drainage Basins [vector] <put output description here>

Junctions [vector] <put output description here>

Console usage

```
processing.runalg('saga:channelnetworkanddrainagebasins', dem, threshold, direction, connection, c
```

See also

Channel network

Description

<put algorithm description here>

Parameters

Elevation [raster] <put parameter description here>

Flow Direction [raster] Optional.

<put parameter description here>

Initiation Grid [raster] <put parameter description here>

Initiation Type [selection] <put parameter description here>

Options:

- 0 — [0] Less than
- 1 — [1] Equals
- 2 — [2] Greater than

Default: 0

Initiation Threshold [number] <put parameter description here>

Default: 0.0

Divergence [raster] Optional.

<put parameter description here>

Tracing: Max. Divergence [number] <put parameter description here>

Default: 10

Tracing: Weight [raster] Optional.

<put parameter description here>

Min. Segment Length [number] <put parameter description here>

Default: 10

Outputs

Channel Network [raster] <put output description here>

Channel Direction [raster] <put output description here>

Channel Network [vector] <put output description here>

Console usage

```
processing.runalg('saga:channelnetwork', elevation, sinkroute, init_grid, init_method, init_value)
```

See also

Overland flow distance to channel network

Description

<put algorithm description here>

Parameters

Elevation [raster] <put parameter description here>

Channel Network [raster] <put parameter description here>

Flow Algorithm [selection] <put parameter description here>

Options:

- 0 — [0] D8
- 1 — [1] MFD

Default: 0

Outputs

Overland Flow Distance [raster] <put output description here>

Vertical Overland Flow Distance [raster] <put output description here>

Horizontal Overland Flow Distance [raster] <put output description here>

Console usage

```
processing.runalg('saga:overlandflowdistancetochannelnetwork', elevation, channels, method, distanceto)
```

See also

Strahler order

Description

<put algorithm description here>

Parameters

Elevation [raster] <put parameter description here>

Outputs

Strahler Order [raster] <put output description here>

Console usage

```
processing.runalg('saga:strahlerorder', dem, strahler)
```

See also

Vertical distance to channel network

Description

<put algorithm description here>

Parameters

Elevation [raster] <put parameter description here>

Channel Network [raster] <put parameter description here>

Tension Threshold [Percentage of Cell Size] [number] <put parameter description here>

Default: *1*

Keep Base Level below Surface [boolean] <put parameter description here>

Default: *True*

Outputs

Vertical Distance to Channel Network [raster] <put output description here>

Channel Network Base Level [raster] <put output description here>

Console usage

```
processing.runalg('saga:verticaldistancetochannelnetwork', elevation, channels, threshold, nounde
```

See also

Watershed basins

Description

<put algorithm description here>

Parameters

Elevation [raster] <put parameter description here>

Channel Network [raster] <put parameter description here>

Sink Route [raster] Optional.

<put parameter description here>

Min. Size [number] <put parameter description here>

Default: 0

Outputs

Watershed Basins [raster] <put output description here>

Console usage

```
processing.runalg('saga:watershedbasins', elevation, channels, sinkroute, minsize, basins)
```

See also

.

18.7.25 Terrain hydrology

Burn stream network into dem

Description

<put algorithm description here>

Parameters

DEM [raster] <put parameter description here>

Streams [raster] <put parameter description here>

Method [selection] <put parameter description here>

Options:

- 0 — [0] simply decrease cell's value by epsilon
- 1 — [1] lower cell's value to neighbours minimum value minus epsilon

Default: 0

Epsilon [number] <put parameter description here>

Default: 1.0

Outputs

Processed DEM [raster] <put output description here>

Console usage

```
processing.runalg('saga:burnstreamnetworkintodem', dem, stream, method, epsilon, burn)
```

See also

Catchment area (flow tracing)

Description

<put algorithm description here>

Parameters

Elevation [raster] <put parameter description here>

Sink Routes [raster] Optional.

<put parameter description here>

Weight [raster] Optional.

<put parameter description here>

Material [raster] Optional.

<put parameter description here>

Target [raster] Optional.

<put parameter description here>

Step [number] <put parameter description here>

Default: *1*

Method [selection] <put parameter description here>

Options:

- 0 — [0] Rho 8
- 1 — [1] Kinematic Routing Algorithm
- 2 — [2] DEMON

Default: *0*

DEMON - Min. DQV [number] <put parameter description here>

Default: *0.0*

Flow Correction [boolean] <put parameter description here>

Default: *True*

Outputs

Catchment Area [raster] <put output description here>

Catchment Height [raster] <put output description here>

Catchment Slope [raster] <put output description here>

Total accumulated Material [raster] <put output description here>

Accumulated Material from _left_ side [raster] <put output description here>

Accumulated Material from _right_ side [raster] <put output description here>

Console usage

```
processing.runalg('saga:catchmentareaflowtracing', elevation, sinkroute, weight, material, target,
```

See also

Catchment area (recursive)

Description

<put algorithm description here>

Parameters

Elevation [raster] <put parameter description here>

Sink Routes [raster] Optional.

<put parameter description here>

Weight [raster] Optional.

<put parameter description here>

Material [raster] Optional.

<put parameter description here>

Target [raster] Optional.

<put parameter description here>

Step [number] <put parameter description here>

Default: *1*

Target Areas [raster] Optional.

<put parameter description here>

Method [selection] <put parameter description here>

Options:

- 0 — [0] Deterministic 8
- 1 — [1] Rho 8
- 2 — [2] Deterministic Infinity
- 3 — [3] Multiple Flow Direction

Default: *0*

Convergence [number] <put parameter description here>

Default: *1.1*

Outputs

Catchment Area [raster] <put output description here>

Catchment Height [raster] <put output description here>

Catchment Slope [raster] <put output description here>

Total accumulated Material [raster] <put output description here>

Accumulated Material from _left_ side [raster] <put output description here>

Accumulated Material from _right_ side [raster] <put output description here>

Flow Path Length [raster] <put output description here>

Console usage

```
processing.runalg('saga:catchmentarearecursive', elevation, sinkroute, weight, material, target, ...)
```

See also

Catchment Area

Description

<put algorithm description here>

Parameters

Elevation [raster] <put parameter description here>

Method [selection] <put parameter description here>

Options:

- 0 — [0] Deterministic 8
- 1 — [1] Rho 8
- 2 — [2] Braunschweiger Reliefmodell
- 3 — [3] Deterministic Infinity
- 4 — [4] Multiple Flow Direction
- 5 — [5] Multiple Triangular Flow Directon

Default: 0

Outputs

Catchment Area [raster] <put output description here>

Console usage

```
processing.runalg('saga:catchmentarea', elevation, method, carea)
```

See also

Cell balance

Description

<put algorithm description here>

Parameters

Elevation [raster] <put parameter description here>

Parameter [raster] Optional.

<put parameter description here>

Default Weight [number] <put parameter description here>

Default: 1.0

Method [selection] <put parameter description here>

Options:

- 0 — [0] Deterministic 8
- 1 — [1] Multiple Flow Direction

Default: 0

Outputs

Cell Balance [raster] <put output description here>

Console usage

```
processing.runalg('saga:cellbalance', dem, weights, weight, method, balance)
```

See also

Edge contamination

Description

<put algorithm description here>

Parameters

Elevation [raster] <put parameter description here>

Outputs

Edge Contamination [raster] <put output description here>

Console usage

```
processing.runalg('saga:edgecontamination', dem, contamination)
```

See also

Fill Sinks

Description

<put algorithm description here>

Parameters

DEM [raster] <put parameter description here>

Minimum Slope [Degree] [number] <put parameter description here>

Default: *0.01*

Outputs

Filled DEM [raster] <put output description here>

Console usage

```
processing.runalg('saga:fillsinks', dem, minslope, result)
```

See also

Fill sinks (wang & liu)

Description

<put algorithm description here>

Parameters

DEM [raster] <put parameter description here>

Minimum Slope [Degree] [number] <put parameter description here>

Default: *0.01*

Outputs

Filled DEM [raster] <put output description here>

Flow Directions [raster] <put output description here>

Watershed Basins [raster] <put output description here>

Console usage

```
processing.runalg('saga:fillsinkswangliu', elev, minslope, filled, fdir, wshed)
```

See also

Fill sinks xxi (wang & liu)

Description

<put algorithm description here>

Parameters

DEM [raster] <put parameter description here>

Minimum Slope [Degree] [number] <put parameter description here>

Default: *0.01*

Outputs

Filled DEM [raster] <put output description here>

Console usage

```
processing.runalg('saga:fillsinksxxlwangliu', elev, minslope, filled)
```

See also

Flat detection

Description

<put algorithm description here>

Parameters

DEM [raster] <put parameter description here>

Flat Area Values [selection] <put parameter description here>

Options:

- 0 — [0] elevation
- 1 — [1] enumeration

Default: *0*

Outputs

No Flats [raster] <put output description here>

Flat Areas [raster] <put output description here>

Console usage

```
processing.runalg('saga:flatdetection', dem, flat_output, noflats, flats)
```

See also

Flow path length

Description

<put algorithm description here>

Parameters

Elevation [raster] <put parameter description here>

Seeds [raster] Optional.

<put parameter description here>

Seeds Only [boolean] <put parameter description here>

Default: *True*

Flow Routing Algorithm [selection] <put parameter description here>

Options:

- 0 — [0] Deterministic 8 (D8)
- 1 — [1] Multiple Flow Direction (FD8)

Default: *0*

Convergence (FD8) [number] <put parameter description here>

Default: *1.1*

Outputs

Flow Path Length [raster] <put output description here>

Console usage

```
processing.runalg('saga:flowpathlength', elevation, seed, seeds_only, method, convergence, length)
```

See also

Flow width and specific catchment area

Description

<put algorithm description here>

Parameters

Elevation [raster] <put parameter description here>

Total Catchment Area (TCA) [raster] Optional.

<put parameter description here>

Method [selection] <put parameter description here>

Options:

- 0 — [0] Deterministic 8
- 1 — [1] Multiple Flow Direction (Quinn et al. 1991)
- 2 — [2] Aspect

Default: 0

Outputs

Flow Width [raster] <put output description here>

Specific Catchment Area (SCA) [raster] <put output description here>

Console usage

```
processing.runalg('saga:flowwidthandspecificcatchmentarea', dem, tca, method, width, sca)
```

See also

Lake flood

Description

<put algorithm description here>

Parameters

DEM [raster] <put parameter description here>

Seeds [raster] <put parameter description here>

Absolute Water Levels [boolean] <put parameter description here>

Default: *True*

Outputs

Lake [raster] <put output description here>

Surface [raster] <put output description here>

Console usage

```
processing.runalg('saga:lakeflood', elev, seeds, level, outdepth, outlevel)
```

See also

Ls factor

Description

<put algorithm description here>

Parameters

Slope [raster] <put parameter description here>

Catchment Area [raster] <put parameter description here>

Area to Length Conversion [selection] <put parameter description here>

Options:

- 0 — [0] no conversion (areas already given as specific catchment area)
- 1 — [1] 1 / cell size (specific catchment area)
- 2 — [2] square root (catchment length)

Default: 0

Method (LS) [selection] <put parameter description here>

Options:

- 0 — [0] Moore et al. 1991
- 1 — [1] Desmet & Govers 1996
- 2 — [2] Bochner & Selige 2006

Default: 0

Rill/Interrill Erosivity [number] <put parameter description here>

Default: 0.0

Stability [selection] <put parameter description here>

Options:

- 0 — [0] stable
- 1 — [1] instable (thawing)

Default: 0

Outputs

LS Factor [raster] <put output description here>

Console usage

```
processing.runalg('saga:lsfactor', slope, area, conv, method, erosivity, stability, ls)
```

See also

Saga wetness index

Description

<put algorithm description here>

Parameters

Elevation [raster] <put parameter description here>

t [number] <put parameter description here>

Default: *10*

Outputs

Catchment area [raster] <put output description here>

Catchment slope [raster] <put output description here>

Modified catchment area [raster] <put output description here>

Wetness index [raster] <put output description here>

Console usage

```
processing.runalg('saga:sagawetnessindex', dem, t, c, gn, cs, sb)
```

See also

Sink drainage route detection

Description

<put algorithm description here>

Parameters

Elevation [raster] <put parameter description here>

Threshold [boolean] <put parameter description here>

Default: *True*

Threshold Height [number] <put parameter description here>

Default: *100.0*

Outputs

Sink Route [raster] <put output description here>

Console usage

```
processing.runalg('saga:sinkdrainageroutedetection', elevation, threshold, thrsheight, sinkroute)
```

See also

Sink removal

Description

<put algorithm description here>

Parameters

DEM [raster] <put parameter description here>

Sink Route [raster] Optional.

<put parameter description here>

Method [selection] <put parameter description here>

Options:

- 0 — [0] Deepen Drainage Routes
- 1 — [1] Fill Sinks

Default: *0*

Threshold [boolean] <put parameter description here>

Default: *True*

Threshold Height [number] <put parameter description here>

Default: *100.0*

Outputs

Preprocessed DEM [raster] <put output description here>

Console usage

```
processing.runalg('saga:sinkremoval', dem, sinkroute, method, threshold, thrsheight, dem_preproc)
```

See also

Slope length

Description

<put algorithm description here>

Parameters

Elevation [raster] <put parameter description here>

Outputs

Slope Length [raster] <put output description here>

Console usage

```
processing.runalg('saga:slopelength', dem, length)
```

See also

Stream power index

Description

<put algorithm description here>

Parameters

Slope [raster] <put parameter description here>

Catchment Area [raster] <put parameter description here>

Area Conversion [selection] <put parameter description here>

Options:

- 0 — [0] no conversion (areas already given as specific catchment area)
- 1 — [1] 1 / cell size (pseudo specific catchment area)

Default: 0

Outputs

Stream Power Index [raster] <put output description here>

Console usage

```
processing.runalg('saga:streampowerindex', slope, area, conv, spi)
```

See also

Topographic wetness index (twi)

Description

<put algorithm description here>

Parameters

Slope [raster] <put parameter description here>

Catchment Area [raster] <put parameter description here>

Transmissivity [raster] Optional.

<put parameter description here>

Area Conversion [selection] <put parameter description here>

Options:

- 0 — [0] no conversion (areas already given as specific catchment area)
- 1 — [1] 1 / cell size (pseudo specific catchment area)

Default: 0

Method (TWI) [selection] <put parameter description here>

Options:

- 0 — [0] Standard
- 1 — [1] TOPMODEL

Default: 0

Outputs

Topographic Wetness Index [raster] <put output description here>

Console usage

```
processing.runalg('saga:topographicwetnessindextwi', slope, area, trans, conv, method, twi)
```

See also

Upslope Area

Description

<put algorithm description here>

Parameters

Target Area [raster] Optional.

<put parameter description here>

Target X coordinate [number] <put parameter description here>

Default: *0.0*

Target Y coordinate [number] <put parameter description here>

Default: *0.0*

Elevation [raster] <put parameter description here>

Sink Routes [raster] Optional.

<put parameter description here>

Method [selection] <put parameter description here>

Options:

- 0 — [0] Deterministic 8
- 1 — [1] Deterministic Infinity
- 2 — [2] Multiple Flow Direction

Default: *0*

Convergence [number] <put parameter description here>

Default: *1.1*

Outputs

Upslope Area [raster] <put output description here>

Console usage

```
processing.runalg('saga:upslopearea', target, target_pt_x, target_pt_y, elevation, sinkroute, met
```

See also

.

18.7.26 Terrain lighting

Analytical hillshading

Description

<put algorithm description here>

Parameters

Elevation [**raster**] <put parameter description here>

Shading Method [**selection**] <put parameter description here>

Options:

- 0 — [0] Standard
- 1 — [1] Standard (max. 90Degree)
- 2 — [2] Combined Shading
- 3 — [3] Ray Tracing

Default: 0

Azimuth [**Degree**] [**number**] <put parameter description here>

Default: 315.0

Declination [**Degree**] [**number**] <put parameter description here>

Default: 45.0

Exaggeration [**number**] <put parameter description here>

Default: 4.0

Outputs

Analytical Hillshading [**raster**] <put output description here>

Console usage

```
processing.runalg('saga:analyticalhillshading', elevation, method, azimuth, declination, exaggeration)
```

See also

Sky view factor

Description

<put algorithm description here>

Parameters

Elevation [**raster**] <put parameter description here>

Maximum Search Radius [**number**] <put parameter description here>

Default: 10000

Method [**selection**] <put parameter description here>

Options:

- 0 — [0] multi scale
- 1 — [1] sectors

Default: 0

Multi Scale Factor [number] <put parameter description here>

Default: 3

Number of Sectors [number] <put parameter description here>

Default: 8

Outputs

Visible Sky [raster] <put output description here>

Sky View Factor [raster] <put output description here>

Sky View Factor (Simplified) [raster] <put output description here>

Terrain View Factor [raster] <put output description here>

Console usage

```
processing.runalg('saga:skyviewfactor', dem, maxradius, method, level_inc, ndirs, visible, svf, s
```

See also

Topographic correction

Description

<put algorithm description here>

Parameters

Elevation [raster] <put parameter description here>

Original Image [raster] <put parameter description here>

Azimuth [number] <put parameter description here>

Default: 180.0

Height [number] <put parameter description here>

Default: 45.0

Method [selection] <put parameter description here>

Options:

- 0 — [0] Cosine Correction (Teillet et al. 1982)
- 1 — [1] Cosine Correction (Civco 1989)
- 2 — [2] Minnaert Correction
- 3 — [3] Minnaert Correction with Slope (Riano et al. 2003)
- 4 — [4] Minnaert Correction with Slope (Law & Nichol 2004)
- 5 — [5] C Correction
- 6 — [6] Normalization (after Civco, modified by Law & Nichol)

Default: 0

Minnaert Correction [number] <put parameter description here>

Default: 0.5

Maximum Cells (C Correction Analysis) [number] <put parameter description here>

Default: 1000

Value Range [selection] <put parameter description here>

Options:

- 0 — [0] 1 byte (0-255)
- 1 — [1] 2 byte (0-65535)

Default: 0

Outputs

Corrected Image [raster] <put output description here>

Console usage

```
processing.runalg('saga:topographiccorrection', dem, original, azi, hgt, method, minnaert, maxcel.
```

See also

.

18.7.27 Terrain morphometry

Convergence index

Description

Calculates an index of convergence/divergence regarding to overland flow. By its meaning it is similar to plan or horizontal curvature, but gives much smoother results. The calculation uses the aspects of surrounding cells, i.e. it looks to which degree surrounding cells point to the center cell. The result is given as percentages, negative values correspond to convergent, positive to divergent flow conditions. Minus 100 would be like a peak of a cone, plus 100 a pit, and 0 an even slope.

Parameters

Elevation [raster] <put parameter description here>

Method [selection] <put parameter description here>

Options:

- 0 — [0] Aspect
- 1 — [1] Gradient

Default: 0

Gradient Calculation [selection] <put parameter description here>

Options:

- 0 — [0] 2 x 2

- 1 — [1] 3 x 3

Default: 0

Outputs

Convergence Index [raster] <put output description here>

Console usage

```
processing.runalg('saga:convergenceindex', elevation, method, neighbours, result)
```

See also

- Koethe, R. / Lehmeier, F. (1996): 'SARA, System zur Automatischen Relief-Analyse', Benutzerhandbuch, 2. Auflage [Geogr. Inst. Univ. Goettingen, unpublished]

Convergence index (search radius)

Description

<put algorithm description here>

Parameters

Elevation [raster] <put parameter description here>

Radius [Cells] [number] <put parameter description here>

Default: 10

Distance Weighting [selection] <put parameter description here>

Options:

- 0 — [0] no distance weighting
- 1 — [1] inverse distance to a power
- 2 — [2] exponential
- 3 — [3] gaussian weighting

Default: 0

Inverse Distance Weighting Power [number] <put parameter description here>

Default: 1

Inverse Distance Offset [boolean] <put parameter description here>

Default: *True*

Gaussian and Exponential Weighting Bandwidth [number] <put parameter description here>

Default: 1

Gradient [boolean] <put parameter description here>

Default: *True*

Difference [selection] <put parameter description here>

Options:

- 0 — [0] direction to the center cell
- 1 — [1] center cell's aspect direction

Default: 0

Outputs

Convergence Index [raster] <put output description here>

Console usage

```
processing.runalg('saga:convergenceindexsearchradius', elevation, radius, distance_weighting_weight)
```

See also

Curvature classification

Description

<put algorithm description here>

Parameters

Plan Curvature [raster] <put parameter description here>

Profile Curvature [raster] <put parameter description here>

Threshold for plane [number] <put parameter description here>

Default: 0.001

Outputs

Curvature Classification [raster] <put output description here>

Console usage

```
processing.runalg('saga:curvatureclassification', cplan, cprof, threshold, class)
```

See also

Diurnal anisotropic heating

Description

<put algorithm description here>

Parameters

Elevation [raster] <put parameter description here>

Alpha Max (Degree) [number] <put parameter description here>

Default: *202.5*

Outputs

Diurnal Anisotropic Heating [raster] <put output description here>

Console usage

```
processing.runalg('saga:diurnalanisotropicheating', dem, alpha_max, dah)
```

See also

Downslope distance gradient

Description

<put algorithm description here>

Parameters

Elevation [raster] <put parameter description here>

Vertical Distance [number] <put parameter description here>

Default: *10*

Output [selection] <put parameter description here>

Options:

- 0 — [0] distance
- 1 — [1] gradient (tangens)
- 2 — [2] gradient (degree)

Default: *0*

Outputs

Gradient [raster] <put output description here>

Gradient Difference [raster] <put output description here>

Console usage

```
processing.runalg('saga:downslopedistancegradient', dem, distance, output, gradient, difference)
```

See also

Effective air flow heights

Description

<put algorithm description here>

Parameters

Elevation [raster] <put parameter description here>

Wind Direction [raster] Optional.

<put parameter description here>

Wind Speed [raster] Optional.

<put parameter description here>

Constant Wind Direction [Degree] [number] <put parameter description here>

Default: *135*

Old Version [boolean] <put parameter description here>

Default: *True*

Search Distance [km] [number] <put parameter description here>

Default: *300*

Acceleration [number] <put parameter description here>

Default: *1.5*

Use Pyramids with New Version [boolean] <put parameter description here>

Default: *True*

Lee Factor [number] <put parameter description here>

Default: *0.5*

Luv Factor [number] <put parameter description here>

Default: *1.0*

Wind Direction Units [selection] <put parameter description here>

Options:

- 0 — [0] radians
- 1 — [1] degree

Default: *0*

Wind Speed Scale Factor [number] <put parameter description here>

Default: *1.0*

Outputs

Effective Air Flow Heights [raster] <put output description here>

Console usage

```
processing.runalg('saga:effectiveairflowheights', dem, dir, len, dir_const, oldver, maxdist, acce
```

See also

Hypsometry

Description

<put algorithm description here>

Parameters

Elevation [raster] <put parameter description here>

Number of Classes [number] <put parameter description here>

Default: *100.0*

Sort [selection] <put parameter description here>

Options:

- 0 — [0] up
- 1 — [1] down

Default: *0*

Classification Constant [selection] <put parameter description here>

Options:

- 0 — [0] height
- 1 — [1] area

Default: *0*

Use Z-Range [boolean] <put parameter description here>

Default: *True*

Z-Range Min [number] <put parameter description here>

Default: *0.0*

Z-Range Max [number] <put parameter description here>

Default: *1000.0*

Outputs

Hypsometry [table] <put output description here>

Console usage

```
processing.runalg('saga:hypsometry', elevation, count, sorting, method, bzrange, zrange_min, zran
```

See also

Land surface temperature

Description

<put algorithm description here>

Parameters

Elevation [m] [raster] <put parameter description here>

Short Wave Radiation [kW/m2] [raster] <put parameter description here>

Leaf Area Index [raster] <put parameter description here>

Elevation at Reference Station [m] [number] <put parameter description here>

Default: *0.0*

Temperature at Reference Station [Deg.Celsius] [number] <put parameter description here>

Default: *0.0*

Temperature Gradient [Deg.Celsius/km] [number] <put parameter description here>

Default: *6.5*

C Factor [number] <put parameter description here>

Default: *1.0*

Outputs

Land Surface Temperature [Deg.Celsius] [raster] <put output description here>

Console usage

```
processing.runalg('saga:landsurfacetemperature', dem, swr, lai, z_reference, t_reference, t_gradient)
```

See also

Mass balance index

Description

<put algorithm description here>

Parameters

Elevation [raster] <put parameter description here>

Vertical Distance to Channel Network [raster] Optional.

<put parameter description here>

T Slope [number] <put parameter description here>

Default: *15.0*

T Curvature [number] <put parameter description here>

Default: *0.01*

T Vertical Distance to Channel Network [number] <put parameter description here>

Default: *15.0*

Outputs

Mass Balance Index [raster] <put output description here>

Console usage

```
processing.runalg('saga:massbalanceindex', dem, hrel, tslope, tcurve, threl, mbi)
```

See also

Morphometric protection index

Description

<put algorithm description here>

Parameters

Elevation [raster] <put parameter description here>

Radius [number] <put parameter description here>

Default: *2000.0*

Outputs

Protection Index [raster] <put output description here>

Console usage

```
processing.runalg('saga:morphometricprotectionindex', dem, radius, protection)
```

See also

Multiresolution index of valley bottom flatness (mrvbf)

Description

<put algorithm description here>

Parameters

Elevation [raster] <put parameter description here>

Initial Threshold for Slope [number] <put parameter description here>

Default: *16*

Threshold for Elevation Percentile (Lowness) [number] <put parameter description here>

Default: *0.4*

Threshold for Elevation Percentile (Upness) [number] <put parameter description here>

Default: *0.35*

Shape Parameter for Slope [number] <put parameter description here>

Default: *4.0*

Shape Parameter for Elevation Percentile [number] <put parameter description here>

Default: *3.0*

Update Views [boolean] <put parameter description here>

Default: *True*

Classify [boolean] <put parameter description here>

Default: *True*

Maximum Resolution (Percentage) [number] <put parameter description here>

Default: *100*

Outputs

MRVBF [raster] <put output description here>

MRRTF [raster] <put output description here>

Console usage

```
processing.runalg('saga:multiresolutionindexofvalleybottomflatnessmrvbf', dem, t_slope, t_pctl_v,
```

See also

Real area calculation

Description

<put algorithm description here>

Parameters

Elevation [raster] <put parameter description here>

Outputs

Real Area Grid [raster] <put output description here>

Console usage

```
processing.runalg('saga:realareacalculation', dem, area)
```

See also

Relative heights and slope positions

Description

<put algorithm description here>

Parameters

Elevation [raster] <put parameter description here>

w [number] <put parameter description here>

Default: *0.5*

t [number] <put parameter description here>

Default: *10.0*

e [number] <put parameter description here>

Default: *2.0*

Outputs

Slope Height [raster] <put output description here>

Valley Depth [raster] <put output description here>

Normalized Height [raster] <put output description here>

Standardized Height [raster] <put output description here>

Mid-Slope Positon [raster] <put output description here>

Console usage

```
processing.runalg('saga:relativeheightsandslopepositions', dem, w, t, e, ho, hu, nh, sh, ms)
```

See also

Slope, aspect, curvature

Description

<put algorithm description here>

Parameters

Elevation [raster] <put parameter description here>

Method [selection] <put parameter description here>

Options:

- 0 — [0] Maximum Slope (Travis et al. 1975)
- 1 — [1] Maximum Triangle Slope (Tarboton 1997)
- 2 — [2] Least Squares Fitted Plane (Horn 1981, Costa-Cabral & Burgess 1996)
- 3 — [3] Fit 2.Degree Polynom (Bauer, Rohdenburg, Bork 1985)
- 4 — [4] Fit 2.Degree Polynom (Heerdegen & Beran 1982)
- 5 — [5] Fit 2.Degree Polynom (Zevenbergen & Thorne 1987)
- 6 — [6] Fit 3.Degree Polynom (Haralick 1983)

Default: 5

Outputs

Slope [raster] <put output description here>

Aspect [raster] <put output description here>

Curvature [raster] <put output description here>

Plan Curvature [raster] <put output description here>

Profile Curvature [raster] <put output description here>

Console usage

```
processing.runalg('saga:slopeaspectcurvature', elevation, method, slope, aspect, curv, hcurv, vcurv)
```

See also

Surface specific points

Description

<put algorithm description here>

Parameters

Elevation [raster] <put parameter description here>

Method [selection] <put parameter description here>

Options:

- 0 — [0] Mark Highest Neighbour
- 1 — [1] Opposite Neighbours
- 2 — [2] Flow Direction
- 3 — [3] Flow Direction (up and down)

- 4 — [4] Peucker & Douglas

Default: 0

Threshold [number] <put parameter description here>

Default: 2.0

Outputs

Result [raster] <put output description here>

Console usage

```
processing.runalg('saga:surfacespecificpoints', elevation, method, threshold, result)
```

See also

Terrain ruggedness index (tri)

Description

<put algorithm description here>

Parameters

Elevation [raster] <put parameter description here>

Radius (Cells) [number] <put parameter description here>

Default: 1

Distance Weighting [selection] <put parameter description here>

Options:

- 0 — [0] no distance weighting
- 1 — [1] inverse distance to a power
- 2 — [2] exponential
- 3 — [3] gaussian weighting

Default: 0

Inverse Distance Weighting Power [number] <put parameter description here>

Default: 1

Inverse Distance Offset [boolean] <put parameter description here>

Default: *True*

Gaussian and Exponential Weighting Bandwidth [number] <put parameter description here>

Default: 1.0

Outputs

Terrain Ruggedness Index (TRI) [raster] <put output description here>

Console usage

```
processing.runalg('saga:terrainruggednessindextri', dem, radius, distance_weighting_weighting, di
```

See also

Topographic position index (tpi)

Description

<put algorithm description here>

Parameters

Elevation [raster] <put parameter description here>

Standardize [boolean] <put parameter description here>

Default: *True*

Min Radius [number] <put parameter description here>

Default: *0.0*

Max Radius [number] <put parameter description here>

Default: *100.0*

Distance Weighting [selection] <put parameter description here>

Options:

- 0 — [0] no distance weighting
- 1 — [1] inverse distance to a power
- 2 — [2] exponential
- 3 — [3] gaussian weighting

Default: *0*

Inverse Distance Weighting Power [number] <put parameter description here>

Default: *1*

Inverse Distance Offset [boolean] <put parameter description here>

Default: *True*

Gaussian and Exponential Weighting Bandwidth [number] <put parameter description here>

Default: *75.0*

Outputs

Topographic Position Index [raster] <put output description here>

Console usage

```
processing.runalg('saga:topographicpositionindextpi', dem, standard, radius_min, radius_max, dist
```

See also

Tpi based landform classification

Description

<put algorithm description here>

Parameters

Elevation [raster] <put parameter description here>

Min Radius A [number] <put parameter description here>

Default: 0

Max Radius A [number] <put parameter description here>

Default: 100

Min Radius B [number] <put parameter description here>

Default: 0

Max Radius B [number] <put parameter description here>

Default: 1000

Distance Weighting [selection] <put parameter description here>

Options:

- 0 — [0] no distance weighting
- 1 — [1] inverse distance to a power
- 2 — [2] exponential
- 3 — [3] gaussian weighting

Default: 0

Inverse Distance Weighting Power [number] <put parameter description here>

Default: 1

Inverse Distance Offset [boolean] <put parameter description here>

Default: True

Gaussian and Exponential Weighting Bandwidth [number] <put parameter description here>

Default: 75.0

Outputs

Landforms [raster] <put output description here>

Console usage

```
processing.runalg('saga:tpibasedlandformclassification', dem, radius_a_min, radius_a_max, radius_b_min, radius_b_max, distance_weighting, inverse_distance_weighting_power, inverse_distance_offset, gaussian_and_exponential_weighting_bandwidth)
```

See also

Vector ruggedness measure (vrm)

Description

<put algorithm description here>

Parameters

Elevation [raster] <put parameter description here>

Radius (Cells) [number] <put parameter description here>

Default: *1*

Distance Weighting [selection] <put parameter description here>

Options:

- 0 — [0] no distance weighting
- 1 — [1] inverse distance to a power
- 2 — [2] exponential
- 3 — [3] gaussian weighting

Default: *0*

Inverse Distance Weighting Power [number] <put parameter description here>

Default: *1*

Inverse Distance Offset [boolean] <put parameter description here>

Default: *True*

Gaussian and Exponential Weighting Bandwidth [number] <put parameter description here>

Default: *1*

Outputs

Vector Terrain Ruggedness (VRM) [raster] <put output description here>

Console usage

```
processing.runalg('saga:vectorruggednessmeasurevrm', dem, radius, distance_weighting_weighting, d
```

See also

Wind effect

Description

<put algorithm description here>

Parameters

Elevation [raster] <put parameter description here>

Wind Direction [raster] Optional.

<put parameter description here>

Wind Speed [raster] Optional.

<put parameter description here>

Constant Wind Direction [Degree] [number] <put parameter description here>

Default: *135*

Old Version [boolean] <put parameter description here>

Default: *True*

Search Distance [km] [number] <put parameter description here>

Default: *300.0*

Acceleration [number] <put parameter description here>

Default: *1.5*

Use Pyramids [boolean] <put parameter description here>

Default: *True*

Wind Direction Units [selection] <put parameter description here>

Options:

- 0 — [0] radians
- 1 — [1] degree

Default: *0*

Wind Speed Scale Factor [number] <put parameter description here>

Default: *1.0*

Outputs

Wind Effect [raster] <put output description here>

Windward Effect [raster] <put output description here>

Leeward Effect [raster] <put output description here>

Console usage

```
processing.runalg('saga:windeffect', dem, dir, len, dir_const, oldver, maxdist, accel, pyramids, o
```

See also

.

18.7.28 Terrain profiles

Cross profiles

Description

<put algorithm description here>

Parameters

DEM [raster] <put parameter description here>

Lines [vector: line] <put parameter description here>

Profile Distance [number] <put parameter description here>

Default: *10.0*

Profile Length [number] <put parameter description here>

Default: *10.0*

Profile Samples [number] <put parameter description here>

Default: *10.0*

Outputs

Cross Profiles [vector] <put output description here>

Console usage

```
processing.runalg('saga:crossprofiles', dem, lines, dist_line, dist_profile, num_profile, profile)
```

See also

Profile from points table

Description

<put algorithm description here>

Parameters

Grid [raster] <put parameter description here>

Input [table] <put parameter description here>

X [tablefield: any] <put parameter description here>

Y [tablefield: any] <put parameter description here>

Outputs

Result [table] <put output description here>

Console usage

```
processing.runalg('saga:profilefrompointstable', grid, table, x, y, result)
```

See also

Profiles from lines

Description

<put algorithm description here>

Parameters

DEM [raster] <put parameter description here>

Values [multipleinput: rasters] Optional.

<put parameter description here>

Lines [vector: line] <put parameter description here>

Name [tablefield: any] <put parameter description here>

Each Line as new Profile [boolean] <put parameter description here>

Default: *True*

Outputs

Profiles [vector] <put output description here>

Profiles [vector] <put output description here>

Console usage

```
processing.runalg('saga:profilesfromlines', dem, values, lines, name, split, profile, profiles)
```

See also

.

18.8 TauDEM algorithm provider

TauDEM (Terrain Analysis Using Digital Elevation Models) is a set of Digital Elevation Model (DEM) tools for the extraction and analysis of hydrologic information from topography as represented by a DEM. This is software developed at Utah State University (USU) for hydrologic digital elevation model analysis and watershed delineation.

TauDEM is distributed as a set of standalone command line executable programs for a Windows and source code for compiling and use on other systems.

Nota: Please remember that Processing contains only the interface description, so you need to install TauDEM 5.0.6 by yourself and configure Processing properly.

Documentation for TauDEM algorithms derived from official [TauDEM documentation](#)

18.8.1 Basic Grid Analysis

D8 Contributing Area

Description

Calculates a grid of contributing areas using the single direction D8 flow model. The contribution of each grid cell is taken as one (or when the optional weight grid is used, the value from the weight grid). The contributing area for each grid cell is taken as its own contribution plus the contribution from upslope neighbors that drain in to it according to the D8 flow model.

If the optional outlet point shapefile is used, only the outlet cells and the cells upslope (by the D8 flow model) of them are in the domain to be evaluated.

By default, the tool checks for edge contamination. This is defined as the possibility that a contributing area value may be underestimated due to grid cells outside of the domain not being counted. This occurs when drainage is inwards from the boundaries or areas with “no data” values for elevation. The algorithm recognizes this and reports “no data” for the contributing area. It is common to see streaks of “no data” values extending inwards from boundaries along flow paths that enter the domain at a boundary. This is the desired effect and indicates that contributing area for these grid cells is unknown due to it being dependent on terrain outside of the domain of data available. Edge contamination checking may be turned off in cases where you know this is not an issue or want to ignore these problems, if for example, the DEM has been clipped along a watershed outline.

Parameters

D8 Flow Direction Grid [raster] A grid of D8 flow directions which are defined, for each cell, as the direction of the one of its eight adjacent or diagonal neighbors with the steepest downward slope. This grid can be obtained as the output of the “**D8 Flow Directions**” tool.

Outlets Shapefile [vector: point] Optional.

A point shapefile defining the outlets of interest. If this input file is used, only the cells upslope of these outlet cells are considered to be within the domain being evaluated.

Weight Grid [raster] Optional.

A grid giving contribution to flow for each cell. These contributions (also sometimes referred to as weights or loadings) are used in the contributing area accumulation. If this input file is not used, the contribution to flow will assumed to be one for each grid cell.

Check for edge contamination [boolean] A flag that indicates whether the tool should check for edge contamination. Edge contamination is defined as the possibility that a contributing area value may be underestimated due to the fact that grid cells outside of the domain have not been evaluated. This occurs when drainage is inwards from the boundaries or areas with NODATA values for elevation. The algorithm recognizes this and reports NODATA for the impated cells. It is common to see streaks of NODATA values extending inwards from boundaries along flow paths that enter the domain at a boundary. This is the desired effect and indicates that contributing area for these grid cells is unknown due to it being dependent on terrain outside of the domain of available data. Edge contamination checking may be turned off in cases where you know this is not an issue, or want to ignore these problems, if for example, the DEM has been clipped along a watershed outline.

Default: *True*

Outputs

D8 Contributing Area Grid [raster] A grid of contributing area values calculated as the cells own contribution plus the contribution from upslope neighbors that drain in to it according to the D8 flow model.

Console usage

```
processing.runalg('taudem:d8contributingarea', -p, -o, -wg, -nc, -ad8)
```

See also

D8 Flow Directions

Description

Creates 2 grids. The first contains the flow direction from each grid cell to one of its adjacent or diagonal neighbors, calculated using the direction of steepest descent. The second contain the slope, as evaluated in the direction of steepest descent, and is reported as drop/distance, i.e. tan of the angle. Flow direction is reported as NODATA for any grid cell adjacent to the edge of the DEM domain, or adjacent to a NODATA value in the DEM. In flat areas, flow directions are assigned away from higher ground and towards lower ground using the method of Garbrecht and Martz (1997). The D8 flow direction algorithm may be applied to a DEM that has not had its pits filled, but it will then result in NODATA values for flow direction and slope at the lowest point of each pit.

D8 Flow Direction Coding:

- 1 — East
- 2 — Northeast
- 3 — North
- 4 — Northwest
- 5 — West
- 6 — Southwest
- 7 — South
- 8 — Southeast

The flow direction routing across flat areas is performed according to the method described by Garbrecht, J. and L. W. Martz, (1997), “The Assignment of Drainage Direction Over Flat Surfaces in Raster Digital Elevation Models”, *Journal of Hydrology*, 193: 204-213.

Parameters

Pit Filled Elevation Grid [raster] A grid of elevation values. This is usually the output of the “**Pit Remove**” tool, in which case it is elevations with pits removed. Pits are low elevation areas in digital elevation models (DEMs) that are completely surrounded by higher terrain. They are generally taken to be artifacts of the digitation process that interfere with the processing of flow across DEMs. So they are removed by raising their elevation to the point where they just drain off the domain. This step is not essential if you have reason to believe that the pits in your DEM are real. If a few pits actually exist and so should not be removed, while at the same time others are believed to be artifacts that need to be removed, the actual pits should have NODATA elevation values inserted at their lowest point. NODATA values serve to define edges of the domain in the flow field, and elevations are only raised to where flow is off an edge, so an internal NODATA value will stop a pit from being removed, if necessary.

Outputs

D8 Flow Direction Grid [raster] A grid of D8 flow directions which are defined, for each cell, as the direction of the one of its eight adjacent or diagonal neighbors with the steepest downward slope.

D8 Slope Grid [raster] A grid giving slope in the D8 flow direction. This is measured as drop/distance.

Console usage

```
processing.runalg('taudem:d8flowdirections', -fel, -p, -sd8)
```

See also

D-Infinity Contributing Area

Description

Calculates a grid of specific catchment area which is the contributing area per unit contour length using the multiple flow direction D-infinity approach. D-infinity flow direction is defined as steepest downward slope on planar triangular facets on a block centered grid. The contribution at each grid cell is taken as the grid cell length (or when the optional weight grid input is used, from the weight grid). The contributing area of each grid cell is then taken as its own contribution plus the contribution from upslope neighbors that have some fraction draining to it according to the D-infinity flow model. The flow from each cell either all drains to one neighbor, if the angle falls along a cardinal ($0, \pi/2, \pi, 3\pi/2$) or ordinal ($\pi/4, 3\pi/4, 5\pi/4, 7\pi/4$) direction, or is on an angle falling between the direct angle to two adjacent neighbors. In the latter case the flow is proportioned between these two neighbor cells according to how close the flow direction angle is to the direct angle to those cells. The contour length used here is the grid cell size. The resulting units of the specific catchment area are length units the same as those of the grid cell size.

When the optional weight grid is not used, the result is reported in terms of specific catchment area, the upslope area per unit contour length, taken here as the number of cells times grid cell length (cell area divided by cell length). This assumes that grid cell length is the effective contour length, in the definition of specific catchment area and does not distinguish any difference in contour length dependent upon the flow direction. When the optional weight grid is used, the result is reported directly as a summation of weights, without any scaling.

If the optional outlet point shapefile is used, only the outlet cells and the cells upslope (by the D-infinity flow model) of them are in the domain to be evaluated.

By default, the tool checks for edge contamination. This is defined as the possibility that a contributing area value may be underestimated due to grid cells outside of the domain not being counted. This occurs when drainage is inwards from the boundaries or areas with “no data” values for elevation. The algorithm recognizes this and reports “no data” for the contributing area. It is common to see streaks of “no data” values extending inwards from boundaries along flow paths that enter the domain at a boundary. This is the desired effect and indicates that contributing area for these grid cells is unknown due to it being dependent on terrain outside of the domain of data available. Edge contamination checking may be turned off in cases where you know it is not an issue or want to ignore these problems, if for example, the DEM has been clipped along a watershed outline.

Parameters

D-Infinity Flow Direction Grid [raster] A grid of flow directions based on the D-infinity flow method using the steepest slope of a triangular facet. Flow direction is determined as the direction of the steepest downward slope on the 8 triangular facets of a 3x3 block centered grid. Flow direction is encoded as an angle in radians, counter-clockwise from east as a continuous (floating point) quantity between 0 and 2π . The resulting flow in a grid is then usually interpreted as being proportioned between the two neighboring cells that define the triangular facet with the steepest downward slope.

Outlets Shapefile [vector: point] Optional.

A point shapefile defining the outlets of interest. If this input file is used, only the cells upslope of these outlet cells are considered to be within the domain being evaluated.

Weight Grid [raster] Optional.

A grid giving contribution to flow for each cell. These contributions (also sometimes referred to as weights or loadings) are used in the contributing area accumulation. If this input file is not used, the result is reported in terms of specific catchment area (the upslope area per unit contour length) taken as the number of cells times grid cell length (cell area divided by cell length).

Check for edge contamination [boolean] A flag that indicates whether the tool should check for edge contamination. Edge contamination is defined as the possibility that a contributing area value may be underestimated due to the fact that grid cells outside of the domain have not been evaluated. This occurs when drainage is inwards from the boundaries or areas with NODATA values for elevation. The algorithm recognizes this and reports NODATA for the impated cells. It is common to see streaks of NODATA values extending inwards from boundaries along flow paths that enter the domain at a boundary. This is the desired effect and indicates that contributing area for these grid cells is unknown due to it being dependent on terrain outside of the domain of available data. Edge contamination checking may be turned off in cases where you know this is not an issue, or want to ignore these problems, if for example, the DEM has been clipped along a watershed outline.

Default: *True*

Outputs

D-Infinity Specific Catchment Area Grid [raster] A grid of specific catchment area which is the contributing area per unit contour length using the multiple flow direction D-infinity approach. The contributing area of each grid cell is then taken as its own contribution plus the contribution from upslope neighbors that have some fraction draining to it according to the D-infinity flow model.

Console usage

```
processing.runalg('taudem:dinfinitecontributingarea', -ang, -o, -wg, -nc, -sca)
```

See also

D-Infinity Flow Directions

Description

Assigns a flow direction based on the D-infinity flow method using the steepest slope of a triangular facet (Tarboton, 1997, "A New Method for the Determination of Flow Directions and Contributing Areas in Grid Digital Elevation Models", Water Resources Research, 33(2): 309-319). Flow direction is defined as steepest downward slope on planar triangular facets on a block centered grid. Flow direction is encoded as an angle in radians counter-clockwise from east as a continuous (floating point) quantity between 0 and 2π . The flow direction angle is determined as the direction of the steepest downward slope on the eight triangular facets formed in a 3 x 3 grid cell window centered on the grid cell of interest. The resulting flow in a grid is then usually interpreted as being proportioned between the two neighboring cells that define the triangular facet with the steepest downward slope.

A block-centered representation is used with each elevation value taken to represent the elevation of the center of the corresponding grid cell. Eight planar triangular facets are formed between each grid cell and its eight neighbors. Each of these has a downslope vector which when drawn outwards from the center may be at an angle that lies within or outside the 45 degree ($\pi/4$ radian) angle range of the facet at the center point. If the slope vector

angle is within the facet angle, it represents the steepest flow direction on that facet. If the slope vector angle is outside a facet, the steepest flow direction associated with that facet is taken along the steepest edge. The slope and flow direction associated with the grid cell is taken as the magnitude and direction of the steepest downslope vector from all eight facets. Slope is measured as drop/distance, i.e. tan of the slope angle.

In the case where no slope vectors are positive (downslope), the flow direction is set using the method of Garbrecht and Martz (1997) for the determination of flow across flat areas. This makes flat areas drain away from high ground and towards low ground. The flow path grid to enforce drainage along existing streams is an optional input, and if used, takes precedence over elevations for the setting of flow directions.

The D-infinity flow direction algorithm may be applied to a DEM that has not had its pits filled, but it will then result in “no data” values for the D-infinity flow direction and slope associated with the lowest point of the pit.

Parameters

Pit Filled Elevation Grid [raster] A grid of elevation values. This is usually the output of the “**Pit Remove**” tool, in which case it is elevations with pits removed.

Outputs

D-Infinity Flow Directions Grid [raster] A grid of flow directions based on the D-infinity flow method using the steepest slope of a triangular facet. Flow direction is determined as the direction of the steepest downward slope on the 8 triangular facets of a 3x3 block centered grid. Flow direction is encoded as an angle in radians, counter-clockwise from east as a continuous (floating point) quantity between 0 and 2π . The resulting flow in a grid is then usually interpreted as being proportioned between the two neighboring cells that define the triangular facet with the steepest downward slope.

D-Infinity Slope Grid [raster] A grid of slope evaluated using the D-infinity method described in Tarboton, D. G., (1997), “A New Method for the Determination of Flow Directions and Contributing Areas in Grid Digital Elevation Models”, *Water Resources Research*, 33(2): 309-319. This is the steepest outwards slope on one of eight triangular facets centered at each grid cell, measured as drop/distance, i.e. tan of the slope angle.

Console usage

```
processing.runalg('taudem:dinfiniteflowdirections', -fel, -ang, -slp)
```

See also

Grid Network

Description

Creates 3 grids that contain for each grid cell: 1) the longest path, 2) the total path, and 3) the Strahler order number. These values are derived from the network defined by the D8 flow model.

The longest upslope length is the length of the flow path from the furthest cell that drains to each cell. The total upslope path length is the length of the entire grid network upslope of each grid cell. Lengths are measured between cell centers taking into account cell size and whether the direction is adjacent or diagonal.

Strahler order is defined as follows: A network of flow paths is defined by the D8 Flow Direction grid. Source flow paths have a Strahler order number of one. When two flow paths of different order join the order of the downstream flow path is the order of the highest incoming flow path. When two flow paths of equal order join the downstream flow path order is increased by 1. When more than two flow paths join the downstream flow path order is calculated as the maximum of the highest incoming flow path order or the second highest incoming flow path order + 1. This generalizes the common definition to cases where more than two flow paths join at a point.

Where the optional mask grid and threshold value are input, the function is evaluated only considering grid cells that lie in the domain with mask grid value greater than or equal to the threshold value. Source (first order) grid cells are taken as those that do not have any other grid cells from inside the domain draining in to them, and only when two of these flow paths join is order propagated according to the ordering rules. Lengths are also only evaluated counting paths within the domain greater than or equal to the threshold.

If the optional outlet point shapefile is used, only the outlet cells and the cells upslope (by the D8 flow model) of them are in the domain to be evaluated.

Parameters

D8 Flow Direction Grid [raster] A grid of D8 flow directions which are defined, for each cell, as the direction of the one of its eight adjacent or diagonal neighbors with the steepest downward slope. This grid can be obtained as the output of the “**D8 Flow Directions**” tool.

Outlets Shapefile [vector: point] Optional.

A point shapefile defining the outlets of interest. If this input file is used, only the cells upslope of these outlet cells are considered to be within the domain being evaluated.

Mask Grid [raster] Optional.

A grid that is used to determine the domain to be analyzed. If the mask grid value \geq mask threshold (see below), then the cell will be included in the domain. While this tool does not have an edge contamination flag, if edge contamination analysis is needed, then a mask grid from a function like “**D8 Contributing Area**” that does support edge contamination can be used to achieve the same result.

Mask Threshold [number] This input parameter is used in the calculation mask grid value \geq mask threshold to determine if the grid cell is in the domain to be analyzed.

Default: *100*

Outputs

Longest Upslope Length Grid [raster] A grid that gives the length of the longest upslope D8 flow path terminating at each grid cell. Lengths are measured between cell centers taking into account cell size and whether the direction is adjacent or diagonal.

Total Upslope Length Grid [raster] The total upslope path length is the length of the entire D8 flow grid network upslope of each grid cell. Lengths are measured between cell centers taking into account cell size and whether the direction is adjacent or diagonal.

Strahler Network Order Grid [raster] A grid giving the Strahler order number for each cell. A network of flow paths is defined by the D8 Flow Direction grid. Source flow paths have a Strahler order number of one. When two flow paths of different order join the order of the downstream flow path is the order of the highest incoming flow path. When two flow paths of equal order join the downstream flow path order is increased by 1. When more than two flow paths join the downstream flow path order is calculated as the maximum of the highest incoming flow path order or the second highest incoming flow path order + 1. This generalizes the common definition to cases where more than two flow paths join at a point.

Console usage

```
processing.runalg('taudem:gridnetwork', d8_flow_dir_grid, outlets_shape, mask_grid, threshold, lo
```

See also

Pit Remove

Description

Identifies all pits in the DEM and raises their elevation to the level of the lowest pour point around their edge. Pits are low elevation areas in digital elevation models (DEMs) that are completely surrounded by higher terrain. They are generally taken to be artifacts that interfere with the routing of flow across DEMs, so are removed by raising their elevation to the point where they drain off the edge of the domain. The pour point is the lowest point on the boundary of the “watershed” draining to the pit. This step is not essential if you have reason to believe that the pits in your DEM are real. If a few pits actually exist and so should not be removed, while at the same time others are believed to be artifacts that need to be removed, the actual pits should have NODATA elevation values inserted at their lowest point. NODATA values serve to define edges in the domain, and elevations are only raised to where flow is off an edge, so an internal NODATA value will stop a pit from being removed, if necessary.

Parameters

Elevation Grid [raster] A digital elevation model (DEM) grid to serve as the base input for the terrain analysis and stream delineation.

Outputs

Pit Removed Elevation Grid [raster] A grid of elevation values with pits removed so that flow is routed off of the domain.

Console usage

```
processing.runalg('taudem:pitremove', -z, -fel)
```

See also

.

18.8.2 Specialized Grid Analysis

D8 Distance To Streams

Description

Computes the horizontal distance to stream for each grid cell, moving downslope according to the D8 flow model, until a stream grid cell is encountered.

Parameters

D8 Flow Direction Grid [raster] This input is a grid of flow directions that are encoded using the D8 method where all flow from a cells goes to a single neighboring cell in the direction of steepest descent. This grid can be obtained as the output of the “**D8 Flow Directions**” tool.

Stream Raster Grid [raster] A grid indicating streams. Such a grid can be created by several of the tools in the “**Stream Network Analysis**” toolset. However, the tools in the “**Stream Network Analysis**” toolset only create grids with a value of 0 for no stream, or 1 for stream cells. This tool can also accept grids with values greater than 1, which can be used in conjunction with the `Threshold` parameter to determine the location of streams. This allows Contributing Area grids to be used to define streams as well as the normal Stream Raster grids. This grid expects integer (long integer) values and any non-integer values will be truncated to an integer before being evaluated.

Threshold [number] This value acts as threshold on the `Stream Raster Grid` to determine the location of streams. Cells with a `Stream Raster Grid` value greater than or equal to the `Threshold` value are interpreted as streams.

Default: 50

Outputs

Output Distance to Streams [raster] A grid giving the horizontal distance along the flow path as defined by the D8 Flow Directions Grid to the streams in the Stream Raster Grid.

Console usage

```
processing.runalg('taudem:d8distancetostreams', -p, -src, -thresh, -dist)
```

See also

D-Infinity Avalanche Runout

Description

Identifies an avalanche’s affected area and the flow path length to each cell in that affected area. All cells downslope from each source area cell, up to the point where the slope from the source to the affected area is less than a threshold angle called the Alpha Angle can be in the affected area. This tool uses the D-infinity multiple flow direction method for determining flow direction. This will likely cause very small amounts of flow to be dispersed to some downslope cells that might overstate the affected area, so a threshold proportion can be set to avoid this excess dispersion. The flow path length is the distance from the cell in question to the source cell that has the highest angle.

All points downslope from the source area are potentially in the affected area, but not beyond a point where the slope from the source to the affected area is less than a threshold angle called the Alpha Angle.

Slope is to be measured using the straight line distance from source point to evaluation point.

It makes more physical sense to me for the angle to be measured along the flow path. Nevertheless it is equally easy to code straight line angles as angles along the flow path, so an option that allows switching will be provided. The most practical way to evaluate avalanche runout is to keep track of the source point with the greatest angle to each point. Then the recursive upslope flow algebra approach will look at a grid cell and all its upslope neighbors that flow to it. Information from the upslope neighbors will be used to calculate the angle to the grid cell in question and retain it in the runout zone if the angle exceeds the alpha angle. This procedure makes the assumption that the maximum angle at a grid cell will be from the set of cells that have maximum angles to the inflowing neighbors. This will always be true of angle is calculated along a flow path, but I can conceive of cases where flow paths bend back on themselves where this would not be the case for straight line angles.

The D-infinity multiple flow direction field assigns flow from each grid cell to multiple downslope neighbors using proportions (P_{ik}) that vary between 0 and 1 and sum to 1 for all flows out of a grid cell. It may be desirable to specify a threshold T that this proportion has to exceed before a grid cell is counted as flowing to a downslope

grid cell, e.g. $P_{ik} > T$ ($=0.2$ say) to avoid dispersion to grid cells that get very little flow. T will be specified as a user input. If all upslope grid cells are to be used T may be input as 0.

Avalanche source sites are to be input as a short integer grid (name suffix **ass*, e.g. *demass*) comprised of positive values where avalanches may be triggered and 0 values elsewhere.

The following grids are output:

- *rz* — A runout zone indicator with value 0 to indicate that this grid cell is not in the runout zone and value > 0 to indicate that this grid cell is in the runout zone. Since there may be information in the angle to the associated source site, this variable will be assigned the angle to the source site (in degrees)
- *dm* — Along flow distance from the source site that has the highest angle to the point in question

Parameters

D-Infinity Flow Direction Grid [raster] A grid giving flow direction by the D-infinity method. Flow direction is measured in radians, counter clockwise from east. This can be created by the tool “**D-Infinity Flow Directions**”.

Pit Filled Elevation Grid [raster] This input is a grid of elevation values. As a general rule, it is recommended that you use a grid of elevation values that have had the pits removed for this input. Pits are generally taken to be artifacts that interfere with the analysis of flow across them. This grid can be obtained as the output of the “**Pit Remove**” tool, in which case it contains elevation values where the pits have been filled to the point where they just drain.

Avalanche Source Site Grid [raster] This is a grid of source areas for snow avalanches that are commonly identified manually using a mix of experience and visual interpretation of maps. Avalanche source sites are to be input as a short integer grid (name suffix **ass*, e.g. *demass*) comprised of positive values where avalanches may be triggered and 0 values elsewhere.

Proportion Threshold [number] This value is a threshold proportion that is used to limit the dispersion of flow caused by using the D-infinity multiple flow direction method for determining flow direction. The D-infinity multiple flow direction method often causes very small amounts of flow to be dispersed to some downslope cells that might overstate the affected area, so a threshold proportion can be set to avoid this excess dispersion.

Default: *0.2*

Alpha Angle Threshold [number] This value is the threshold angle, called the Alpha Angle, that is used to determine which of the cells downslope from the source cells are in the affected area. Only the cells downslope from each source area cell, up to the point where the slope from the source to the affected area is less than a threshold angle are in the affected area.

Default: *18*

Measure distance along flow path [boolean] This option selects the method used to measure the distance used to calculate the slope angle. If option is *True* then measure it along the flow path, where the *False* option causes the slope to be measure along the straight line distance from the source cell to the evaluation cell.

Default: *True*

Outputs

Runout Zone Grid [raster] This grid Identifies the avalanche’s runout zone (affected area) using a runout zone indicator with value 0 to indicate that this grid cell is not in the runout zone and value > 0 to indicate that this grid cell is in the runout zone. Since there may be information in the angle to the associated source site, this variable will be assigned the angle to the source site (in degrees).

Path Distance Grid [raster] This is a grid of the flow distance from the source site that has the highest angle to each cell.

Console usage

```
processing.runalg('taudem:dinfiniteavalancherunout', -ang, -fel, -ass, -thresh, -alpha, -direct, ...)
```

See also

D-Infinity Concentration Limited Accumulation

Description

This function applies to the situation where an unlimited supply of a substance is loaded into flow at a concentration or solubility threshold C_{sol} over a region indicated by an indicator grid (dg). It a grid of the concentration of a substance at each location in the domain, where the supply of substance from a supply area is loaded into the flow at a concentration or solubility threshold. The flow is first calculated as a D-infinity weighted contributing area of an input Effective Runoff Weight Grid (notionally excess precipitation). The concentration of substance over the supply area (indicator grid) is at the concentration threshold. As the substance moves downslope with the D-infinity flow field, it is subject to first order decay in moving from cell to cell as well as dilution due to changes in flow. The decay multiplier grid gives the fractional (first order) reduction in quantity in moving from grid cell x to the next downslope cell. If the outlets shapefile is used, the tool only evaluates the part of the domain that contributes flow to the locations given by the shapefile. This is useful for a tracking a contaminant or compound from an area with unlimited supply of that compound that is loaded into a flow at a concentration or solubility threshold over a zone and flow from the zone may be subject to decay or attenuation.

The indicator grid (dg) is used to delineate the area of the substance supply using the (0, 1) indicator function $i(x)$. $A[]$ denotes the weighted accumulation operator evaluated using the D-Infinity Contributing Area function. The Effective Runoff Weight Grid gives the supply to the flow (e.g. the excess rainfall if this is overland flow) denoted as $w(x)$. The specific discharge is then given by:

$$Q(x) = A[w(x)]$$

This weighted accumulation $Q(x)$ is output as the Overland Flow Specific Discharge Grid. Over the substance supply area concentration is at the threshold (the threshold is a saturation or solubility limit). If $i(x) = 1$, then

$$C(x) = C_{sol}, \text{ and } L(x) = C_{sol} Q(x),$$

where $L(x)$ denotes the load being carried by the flow. At remaining locations, the load is determined by load accumulation and the concentration by dilution:

Here $d(x) = d(i, j)$ is a decay multiplier giving the fractional (first order) reduction in mass in moving from grid cell x to the next downslope cell. If travel (or residence) times $t(x)$ associated with flow between cells are available $d(x)$ may be evaluated as $\exp(-k t(x))$ where k is a first order decay parameter. The Concentration grid output is $C(x)$. If the outlets shapefile is used, the tool only evaluates the part of the domain that contributes flow to the locations given by the shapefile.

Useful for a tracking a contaminant released or partitioned to flow at a fixed threshold concentration.

Parameters

D-Infinity Flow Direction Grid [raster] A grid giving flow direction by the D-infinity method. Flow direction is measured in radians, counter clockwise from east. This grid can be created by the function “D-Infinity Flow Directions”.

Disturbance Indicator Grid [raster] A grid that indicates the source zone of the area of substance supply and must be 1 inside the zone and 0 or NODATA over the rest of the domain.

Decay Multiplier Grid [raster] A grid giving the factor by which flow leaving each grid cell is multiplied before accumulation on downslope grid cells. This may be used to simulate the movement of an attenuating or decaying substance. If travel (or residence) times $t(x)$ associated with flow between cells are available $d(x)$ may be evaluated as $\exp(-k t(x))$ where k is a first order decay parameter.

Effective Runoff Weight Grid [raster] A grid giving the input quantity (notionally effective runoff or excess precipitation) to be used in the D-infinity weighted contributing area evaluation of Overland Flow Specific Discharge.

Outlets shapefile [vector: point] Optional.

This optional input is a point shapefile defining outlets of interest. If this file is used, the tool will only evaluate the area upslope of these outlets.

Concentration Threshold [number] The concentration or solubility threshold. Over the substance supply area, concentration is at this threshold.

Default: *1.0*

Check for edge contamination [boolean] This option determines whether the tool should check for edge contamination. Edge contamination is defined as the possibility that a value may be underestimated due to grid cells outside of the domain not being considered when determining contributing area.

Default: *True*

Outputs

Concentration Grid [raster] A grid giving the resulting concentration of the compound of interest in the flow.

Console usage

```
processing.runalg('taudem:dinfiniteconcentrationlimitedaccumulation', -ang, -dg, -dm, -q, -o, -cs
```

See also

D-Infinity Decaying Accumulation

Description

The D-Infinity Decaying Accumulation tool creates a grid of the accumulated quantity at each location in the domain where the quantity accumulates with the D-infinity flow field, but is subject to first order decay in moving from cell to cell. By default, the quantity contribution of each grid cell is the cell length to give a per unit width accumulation, but can optionally be expressed with a weight grid. The decay multiplier grid gives the fractional (first order) reduction in quantity in accumulating from grid cell x to the next downslope cell.

A decayed accumulation operator $DA[.]$ takes as input a mass loading field $m(x)$ expressed at each grid location as $m(i, j)$ that is assumed to move with the flow field but is subject to first order decay in moving from cell to cell. The output is the accumulated mass at each location $DA(x)$. The accumulation of m at each grid cell can be numerically evaluated.

Here $d(x) = d(i, j)$ is a decay multiplier giving the fractional (first order) reduction in mass in moving from grid cell x to the next downslope cell. If travel (or residence) times $t(x)$ associated with flow between cells are available $d(x)$ may be evaluated as $\exp(-k t(x))$ where k is a first order decay parameter. The weight grid is used to represent the mass loading $m(x)$. If not specified this is taken as 1. If the outlets shapefile is used the function is only evaluated on that part of the domain that contributes flow to the locations given by the shapefile.

Useful for a tracking contaminant or compound subject to decay or attenuation.

Parameters

D-Infinity Flow Direction Grid [raster] A grid giving flow direction by the D-infinity method. Flow direction is measured in radians, counter clockwise from east. This grid can be created by the function “**D-Infinity Flow Directions**”.

Decay Multiplier Grid [raster] A grid giving the factor by which flow leaving each grid cell is multiplied before accumulation on downslope grid cells. This may be used to simulate the movement of an attenuating substance.

Weight Grid [raster] Optional.

A grid giving weights (loadings) to be used in the accumulation. If this optional grid is not specified, weights are taken as the linear grid cell size to give a per unit width accumulation.

Outlets Shapefile [vector: point] Optional.

This optional input is a point shapefile defining outlets of interest. If this file is used, the tool will only evaluate the area upslope of these outlets.

Check for edge contamination [boolean] This option determines whether the tool should check for edge contamination. Edge contamination is defined as the possibility that a value may be underestimated due to grid cells outside of the domain not being considered when determining contributing area.

Default: *True*

Outputs

Decayed Specific Catchment Area Grid [raster] The D-Infinity Decaying Accumulation tool creates a grid of the accumulated mass at each location in the domain where mass moves with the D-infinity flow field, but is subject to first order decay in moving from cell to cell.

Console usage

```
processing.runalg('taudem:dinfiniteaccumulation', -ang, -dm, -wg, -o, -nc, -dsca)
```

See also

D-Infinity Distance Down

Description

Calculates the distance downslope to a stream using the D-infinity flow model. The D-infinity flow model is a multiple flow direction model, because the outflow from each grid cell is proportioned between up to 2 downslope grid cells. As such, the distance from any grid cell to a stream is not uniquely defined. Flow that originates at a particular grid cell may enter the stream at a number of different cells. The statistical method may be selected as the longest, shortest or weighted average of the flow path distance to the stream. Also one of several ways of measuring distance may be selected: the total straight line path (Pythagoras), the horizontal component of the straight line path, the vertical component of the straight line path, or the total surface flow path.

Parameters

D-Infinity Flow Direction Grid [raster] A grid giving flow direction by the D-infinity method. Flow direction is measured in radians, counter clockwise from east. This can be created by the tool “**D-Infinity Flow Directions**”.

Pit Filled Elevation Grid [raster] This input is a grid of elevation values. As a general rule, it is recommended that you use a grid of elevation values that have had the pits removed for this input. Pits are generally taken to be artifacts that interfere with the analysis of flow across them. This grid can be obtained as the output of the “**Pit Remove**” tool, in which case it contains elevation values where the pits have been filled to the point where they just drain.

Stream Raster Grid [raster] A grid indicating streams, by using a grid cell value of 1 on streams and 0 off streams. This is usually the output of one of the tools in the “**Stream Network Analysis**” toolset.

Weight Path Grid [raster] Optional.

A grid giving weights (loadings) to be used in the distance calculation. This might be used for example where only flow distance through a buffer is to be calculated. The weight is then 1 in the buffer and 0 outside it. Alternatively the weight may reflect some sort of cost function for travel over the surface, perhaps representing travel time or attenuation of a process. If this input file is not used, the loadings will assumed to be one for each grid cell.

Statistical Method [selection] Statistical method used to calculate the distance down to the stream. In the D-Infinity flow model, the outflow from each grid cell is proportioned between two downslope grid cells. Therefore, the distance from any grid cell to a stream is not uniquely defined. Flow that originates at a particular grid cell may enter the stream at a number of cells. The distance to the stream may be defined as the longest (maximum), shortest (minimum) or weighted average of the distance down to the stream.

Options:

- 0 — Minimum
- 1 — Maximum
- 2 — Average

Default: 2

Distance Method [selection] Distance method used to calculate the distance down to the stream. One of several ways of measuring distance may be selected: the total straight line path (Pythagoras), the horizontal component of the straight line path (horizontal), the vertical component of the straight line path (vertical), or the total surface flow path (surface).

Options:

- 0 — Pythagoras
- 1 — Horizontal
- 2 — Vertical
- 3 — Surface

Default: 1

Check for edge contamination [boolean] A flag that determines whether the tool should check for edge contamination. This is defined as the possibility that a value may be underestimated due to grid cells outside of the domain not being counted. In the context of Distance Down this occurs when part of a flow path traced downslope from a grid cell leaves the domain without reaching a stream grid cell. With edge contamination checking selected, the algorithm recognizes this and reports no data for the result. This is the desired effect and indicates that values for these grid cells is unknown due to it being dependent on terrain outside of the domain of data available. Edge contamination checking may be overridden in cases where you know this is not an issue or want to evaluate the distance using only the fraction of flow paths that terminate at a stream.

Default: *True*

Outputs

D-Infinity Drop to Stream Grid [raster] Grid containing the distance to stream calculated using the D-infinity flow model and the statistical and path methods chosen.

Console usage

```
processing.runalg('taudem:dinfinitydistancedown', dinf_flow_dir_grid, pit_filled_grid, stream_grid)
```

See also

D-Infinity Distance Up

Description

This tool calculates the distance from each grid cell up to the ridge cells along the reverse D-infinity flow directions. Ridge cells are defined to be grid cells that have no contribution from grid cells further upslope. Given the convergence of multiple flow paths at any grid cell, any given grid cell can have multiple upslope ridge cells. There are three statistical methods that this tool can use: maximum distance, minimum distance and waited flow average over these flow paths. A variant on the above is to consider only grid cells that contribute flow with a proportion greater than a user specified threshold (t) to be considered as upslope of any given grid cell. Setting t=0.5 would result in only one flow path from any grid cell and would give the result equivalent to a D8 flow model, rather than D-infinity flow model, where flow is proportioned between two downslope grid cells. Finally there are several different optional paths that can be measured: the total straight line path (Pythagoras), the horizontal component of the straight line path, the vertical component of the straight line path, or the total surface flow path.

Parameters

D-Infinity Flow Direction Grid [raster] A grid giving flow direction by the D-infinity method. Flow direction is measured in radians, counter clockwise from east. This can be created by the tool “**D-Infinity Flow Directions**”.

Pit Filled Elevation Grid [raster] This input is a grid of elevation values. As a general rule, it is recommended that you use a grid of elevation values that have had the pits removed for this input. Pits are generally taken to be artifacts that interfere with the analysis of flow across them. This grid can be obtained as the output of the “**Pit Remove**” tool, in which case it contains elevation values where the pits have been filled to the point where they just drain.

Slope Grid [raster] This input is a grid of slope values. This is measured as drop/distance and it is most often obtained as the output of the “**D-Infinity Flow Directions**” tool.

Statistical Method [selection] Statistical method used to calculate the distance down to the stream. In the D-Infinity flow model, the outflow from each grid cell is proportioned between two downslope grid cells. Therefore, the distance from any grid cell to a stream is not uniquely defined. Flow that originates at a particular grid cell may enter the stream at a number of cells. The distance to the stream may be defined as the longest (maximum), shortest (minimum) or weighted average of the distance down to the stream.

Options:

- 0 — Minimum
- 1 — Maximum
- 2 — Average

Default: 2

Distance Method [selection] Distance method used to calculate the distance down to the stream. One of several ways of measuring distance may be selected: the total straight line path (Pythagoras), the horizontal component of the straight line path (horizontal), the vertical component of the straight line path (vertical), or the total surface flow path (surface).

Options:

- 0 — Pythagoras

- 1 — Horizontal
- 2 — Vertical
- 3 — Surface

Default: *1*

Proportion Threshold [number] The proportion threshold parameter where only grid cells that contribute flow with a proportion greater than this user specified threshold (τ) is considered to be upslope of any given grid cell. Setting $\tau=0.5$ would result in only one flow path from any grid cell and would give the result equivalent to a D8 flow model, rather than D-Infinity flow model, where flow is proportioned between two downslope grid cells.

Default: *0.5*

Check for edge contamination [boolean] A flag that determines whether the tool should check for edge contamination. This is defined as the possibility that a value may be underestimated due to grid cells outside of the domain not being counted.

Default: *True*

Outputs

D-Infinity Distance Up [raster] Grid containing the distances up to the ridge calculated using the D-Infinity flow model and the statistical and path methods chosen.

Console usage

```
processing.runalg('taudem:dinfinitydistanceup', dinf_flow_dir_grid, pit_filled_grid, slope_grid, ...)
```

See also

D-Infinity Reverse Accumulation

Description

This works in a similar way to evaluation of weighted Contributing area, except that the accumulation is by propagating the weight loadings upslope along the reverse of the flow directions to accumulate the quantity of weight loading downslope from each grid cell. The function also reports the maximum value of the weight loading downslope from each grid cell in the Maximum Downslope grid.

This function is designed to evaluate and map the hazard due to activities that may have an effect downslope. The example is land management activities that increase runoff. Runoff is sometimes a trigger for landslides or debris flows, so the weight grid here could be taken as a terrain stability map. Then the reverse accumulation provides a measure of the amount of unstable terrain downslope from each grid cell, as an indicator of the danger of activities that may increase runoff, even though there may be no potential for any local impact.

Parameters

D-Infinity Flow Direction Grid [raster] A grid giving flow direction by the D-infinity method. Flow direction is measured in radians, counter clockwise from east. This can be created by the tool “D-Infinity Flow Directions”.

Weight Grid [raster] A grid giving weights (loadings) to be used in the accumulation.

Outputs

Reverse Accumulation Grid [raster] The grid giving the result of the “Reverse Accumulation” function. This works in a similar way to evaluation of weighted Contributing area, except that the accumulation is by propagating the weight loadings upslope along the reverse of the flow directions to accumulate the quantity of loading downslope from each grid cell.

Maximum Downslope Grid [raster] The grid giving the maximum of the weight loading grid downslope from each grid cell.

Console usage

```
processing.runalg('taudem:dinfinityreverseaccumulation', -ang, -wg, -racc, -dmax)
```

See also

D-Infinity Transport Limited Accumulation - 2

Description

This function is designed to calculate the transport and deposition of a substance (e.g. sediment) that may be limited by both supply and the capacity of the flow field to transport it. This function accumulates substance flux (e.g. sediment transport) subject to the rule that transport out of any grid cell is the minimum between supply and transport capacity, T_{cap} . The total supply at a grid cell is calculated as the sum of the transport in from upslope grid cells, T_{in} , plus the local supply contribution, E (e.g. erosion). This function also outputs deposition, D , calculated as total supply minus actual transport.

Here E is the supply. T_{out} at each grid cell becomes T_{in} for downslope grid cells and is reported as Transport limited accumulation (t_{la}). D is deposition (t_{dep}). The function provides the option to evaluate concentration of a compound (contaminant) adhered to the transported substance. This is evaluated as follows:

Where L_{in} is the total incoming compound loading and C_{in} and T_{in} refer to the Concentration and Transport entering from each upslope grid cell.

If

else

where C_s is the concentration supplied locally and the difference in the second term on the right represents the additional supply from the local grid cell. Then,

C_{out} at each grid cell comprises is the concentration grid output from this function.

If the outlets shapefile is used the tool only evaluates that part of the domain that contributes flow to the locations given by the shapefile.

Transport limited accumulation is useful for modeling erosion and sediment delivery, including the spatial dependence of sediment delivery ratio and contaminant that adheres to sediment.

Parameters

D-Infinity Flow Direction Grid [raster] A grid giving flow direction by the D-infinity method. Flow direction is measured in radians, counter clockwise from east. This can be created by the tool “**D-Infinity Flow Directions**”.

Supply Grid [raster] A grid giving the supply (loading) of material to a transport limited accumulation function. In the application to erosion, this grid would give the erosion detachment, or sediment supplied at each grid cell.

Transport Capacity Grid [raster] A grid giving the transport capacity at each grid cell for the transport limited accumulation function. In the application to erosion this grid would give the transport capacity of the carrying flow.

Input Concentration Grid [raster] A grid giving the concentration of a compound of interest in the supply to the transport limited accumulation function. In the application to erosion, this grid would give the concentration of say phosphorous adhered to the eroded sediment.

Outlets Shapefile [vector: point] Optional.

This optional input is a point shapefile defining outlets of interest. If this file is used, the tool will only evaluate the area upslope of these outlets.

Check for edge contamination [boolean] This option determines whether the tool should check for edge contamination. Edge contamination is defined as the possibility that a value may be underestimated due to grid cells outside of the domain not being considered when determining the result.

Default: *True*

Outputs

Transport Limited Accumulation Grid [raster] This grid is the weighted accumulation of supply accumulated respecting the limitations in transport capacity and reports the transport rate calculated by accumulating the substance flux subject to the rule that the transport out of any grid cell is the minimum of the total supply (local supply plus transport in) to that grid cell and the transport capacity.

Deposition Grid [raster] A grid giving the deposition resulting from the transport limited accumulation. This is the residual from the transport in to each grid cell minus the transport capacity out of the grid cell. The deposition grid is calculated as the transport in + the local supply - the transport out.

Output Concentration Grid [raster] If an input concentration in supply grid is given, then this grid is also output and gives the concentration of a compound (contaminant) adhered or bound to the transported substance (e.g. sediment) is calculated.

Console usage

```
processing.runalg('taudem:dinfiniteytransportlimitedaccumulation2', dinf_flow_dir_grid, supply_grid)
```

See also

D-Infinity Transport Limited Accumulation

Description

This function is designed to calculate the transport and deposition of a substance (e.g. sediment) that may be limited by both supply and the capacity of the flow field to transport it. This function accumulates substance flux (e.g. sediment transport) subject to the rule that transport out of any grid cell is the minimum between supply and transport capacity, T_{cap} . The total supply at a grid cell is calculated as the sum of the transport in from upslope grid cells, T_{in} , plus the local supply contribution, E (e.g. erosion). This function also outputs deposition, D , calculated as total supply minus actual transport.

Here E is the supply. T_{out} at each grid cell becomes T_{in} for downslope grid cells and is reported as Transport limited accumulation (t_{la}). D is deposition (t_{dep}). The function provides the option to evaluate concentration of a compound (contaminant) adhered to the transported substance. This is evaluated as follows:

Where L_{in} is the total incoming compound loading and C_{in} and T_{in} refer to the Concentration and Transport entering from each upslope grid cell.

If

else

where C_s is the concentration supplied locally and the difference in the second term on the right represents the additional supply from the local grid cell. Then,

C_{out} at each grid cell comprises is the concentration grid output from this function.

If the outlets shapefile is used the tool only evaluates that part of the domain that contributes flow to the locations given by the shapefile.

Transport limited accumulation is useful for modeling erosion and sediment delivery, including the spatial dependence of sediment delivery ratio and contaminant that adheres to sediment.

Parameters

D-Infinity Flow Direction Grid [raster] A grid giving flow direction by the D-infinity method. Flow direction is measured in radians, counter clockwise from east. This can be created by the tool “D-Infinity Flow Directions”.

Supply Grid [raster] A grid giving the supply (loading) of material to a transport limited accumulation function. In the application to erosion, this grid would give the erosion detachment, or sediment supplied at each grid cell.

Transport Capacity Grid [raster] A grid giving the transport capacity at each grid cell for the transport limited accumulation function. In the application to erosion this grid would give the transport capacity of the carrying flow.

Outlets Shapefile [vector: point] Optional.

This optional input is a point shapefile defining outlets of interest. If this file is used, the tool will only evaluate the area upslope of these outlets.

Check for edge contamination [boolean] This option determines whether the tool should check for edge contamination. Edge contamination is defined as the possibility that a value may be underestimated due to grid cells outside of the domain not being considered when determining the result.

Default: *True*

Outputs

Transport Limited Accumulation Grid [raster] This grid is the weighted accumulation of supply accumulated respecting the limitations in transport capacity and reports the transport rate calculated by accumulating the substance flux subject to the rule that the transport out of any grid cell is the minimum of the total supply (local supply plus transport in) to that grid cell and the transport capacity.

Deposition Grid [raster] A grid giving the deposition resulting from the transport limited accumulation. This is the residual from the transport in to each grid cell minus the transport capacity out of the grid cell. The deposition grid is calculated as the transport in + the local supply - the transport out.

Console usage

```
processing.runalg('taudem:dinfiniteyntransportlimitedaccumulation', dinf_flow_dir_grid, supply_grid,
```

See also

D-Infinity Upslope Dependence

Description

The D-Infinity Upslope Dependence tool quantifies the amount each grid cell in the domain contributes to a destination set of grid cells. D-Infinity flow directions proportion flow from each grid cell between multiple downslope grid cells. Following this flow field downslope the amount of flow originating at each grid cell that reaches the destination zone is defined. Upslope influence is evaluated using a downslope recursion, examining grid cells downslope from each grid cell, so that the map produced identifies the area upslope where flow through the destination zone originates, or the area it depends on, for its flow.

The figures below illustrate the amount each source point in the domain x (blue) contributes to the destination point or zone y (red). If the indicator weighted contributing area function is denoted $I(y; x)$ giving the weighted contribution using a unit value (1) from specific grid cells y to grid cells x , then the upslope dependence is: $D(x; y) = I(y; x)$.

This is useful for example to track where flow or a flow related substance or contaminant that enters a destination area may come from.

Parameters

D-Infinity Flow Direction Grid [raster] A grid giving flow direction by the D-Infinity method where the flow direction angle is determined as the direction of the steepest downward slope on the eight triangular facets formed in a 3x3 grid cell window centered on the grid cell of interest. This grid can be produced using the “**D-Infinity Flow Direction**” tool.

Destination Grid [raster] A grid that encodes the destination zone that may receive flow from upslope. This grid must be 1 inside the zone y and 0 over the rest of the domain.

Outputs

Output Upslope Dependence Grid [raster] A grid quantifying the amount each source point in the domain contributes to the zone defined by the destination grid.

Console usage

```
processing.runalg('taudem:dinfinityupslopedependence', -ang, -dg, -dep)
```

See also

Slope Average Down

Description

This tool computes slope in a D8 downslope direction averaged over a user selected distance. Distance should be specified in horizontal map units.

Parameters

D8 Flow Direction Grid [raster] This input is a grid of flow directions that are encoded using the D8 method where all flow from a cells goes to a single neighboring cell in the direction of steepest descent. This grid can be obtained as the output of the “**D8 Flow Directions**” tool.

Pit Filled Elevation Grid [raster] This input is a grid of elevation values. As a general rule, it is recommended that you use a grid of elevation values that have had the pits removed for this input. Pits are generally taken to be artifacts that interfere with the analysis of flow across them. This grid can be obtained as the output of the “**Pit Remove**” tool, in which case it contains elevation values where the pits have been filled to the point where they just drain.

Downslope Distance [number] Input parameter of downslope distance over which to calculate the slope (in horizontal map units).

Default: 50

Outputs

Slope Average Down Grid [raster] This output is a grid of slopes calculated in the D8 downslope direction, averaged over the selected distance.

Console usage

```
processing.runalg('taudem:slopeaveragedown', -p, -fel, -dn, -slpd)
```

See also

Slope Over Area Ratio

Description

Calculates the ratio of the slope to the specific catchment area (contributing area). This is algebraically related to the more common $\ln(a/\tan \beta)$ wetness index, but contributing area is in the denominator to avoid divide by 0 errors when slope is 0.

Parameters

Slope Grid [raster] A grid of slope. This grid can be generated using either the “**D8 Flow Directions**” tool or the “**D-Infinity Flow Directions**” tool.

Specific Catchment Area Grid [raster] A grid giving the contributing area value for each cell taken as its own contribution plus the contribution from upslope neighbors that drain in to it. Contributing area is counted in terms of the number of grid cells (or summation of weights). This grid can be generated using either the “**D8 Contributing Area**” tool or the “**D-Infinity Contributing Area**” tool.

Outputs

Slope Divided By Area Ratio Grid [raster] A grid of the ratio of slope to specific catchment area (contributing area). This is algebraically related to the more common $\ln(a/\tan \beta)$ wetness index, but contributing area is in the denominator to avoid divide by 0 errors when slope is 0.

Console usage

```
processing.runalg('taudem:slopeoverarearatio', -slp, -sca, -sar)
```

See also

18.8.3 Stream Network Analysis

D8 Extreme Upslope Value

Description

Evaluates the extreme (either maximum or minimum) upslope value from an input grid based on the D8 flow model. This is intended initially for use in stream raster generation to identify a threshold of the slope times area product that results in an optimum (according to drop analysis) stream network.

If the optional outlet point shapefile is used, only the outlet cells and the cells upslope (by the D8 flow model) of them are in the domain to be evaluated.

By default, the tool checks for edge contamination. This is defined as the possibility that a result may be underestimated due to grid cells outside of the domain not being counted. This occurs when drainage is inwards from the boundaries or areas with “no data” values for elevation. The algorithm recognizes this and reports “no data” for the result for these grid cells. It is common to see streaks of “no data” values extending inwards from boundaries along flow paths that enter the domain at a boundary. This is the desired effect and indicates that the result for these grid cells is unknown due to it being dependent on terrain outside of the domain of data available. Edge

contamination checking may be turned off in cases where you know this is not an issue or want to ignore these problems, if for example, the DEM has been clipped along a watershed outline.

Parameters

D8 Flow Directions Grid [raster] A grid of D8 flow directions which are defined, for each cell, as the direction of the one of its eight adjacent or diagonal neighbors with the steepest downward slope. This grid can be obtained as the output of the “**D8 Flow Directions**” tool.

Upslope Values Grid [raster] This is the grid of values of which the maximum or minimum upslope value is selected. The values most commonly used are the slope times area product needed when generating stream rasters according to drop analysis.

Outlets Shapefile [vector: point] Optional.

A point shape file defining outlets of interest. If this input file is used, only the area upslope of these outlets will be evaluated by the tool.

Check for edge contamination [boolean] A flag that indicates whether the tool should check for edge contamination.

Default: *True*

Use max upslope value [boolean] A flag to indicate whether the maximum or minimum upslope value is to be calculated.

Default: *True*

Outputs

Extreme Upslope Values Grid [raster] A grid of the maximum/minimum upslope values.

Console usage

```
processing.runalg('taudem:d8extremeupslopevalue', -p, -sa, -o, -nc, -min, -ssa)
```

See also

Length Area Stream Source

Description

Creates an indicator grid (1, 0) that evaluates $A \geq (M) (L^y)$ based on upslope path length, D8 contributing area grid inputs, and parameters M and y. This grid indicates likely stream source grid cells. This is an experimental method with theoretical basis in Hack’s law which states that for streams $L \sim A^{0.6}$. However for hillslopes with parallel flow $L \sim A$. So a transition from hillslopes to streams may be represented by $L \sim A^{0.8}$ suggesting identifying grid cells as stream cells if $A > M (L^{(1/0.8)})$.

Parameters

Length Grid [raster] A grid of the maximum upslope length for each cell. This is calculated as the length of the flow path from the furthest cell that drains to each cell. Length is measured between cell centers taking into account cell size and whether the direction is adjacent or diagonal. It is this length (L) that is used in the formula, $A > (M) (L^y)$, to determine which cells are considered stream cells. This grid can be obtained as an output from the “**Grid Network**” tool.

Contributing Area Grid [raster] A grid of contributing area values for each cell that were calculated using the D8 algorithm. The contributing area for a cell is the sum of its own contribution plus the contribution from all upslope neighbors that drain to it, measured as a number of cells. This grid is typically obtained as the output of the “**D8 Contributing Area**” tool. In this tool, it is the contributing area (A) that is compared in the formula $A > (M) (L^y)$ to determine the transition to a stream.

Threshold [number] The multiplier threshold (M) parameter which is used in the formula: $A > (M) (L^y)$, to identify the beginning of streams.

Default: *0.03*

Exponent [number] The exponent (y) parameter which is used in the formula: $A > (M) (L^y)$, to identify the beginning of streams. In branching systems, Hack’s law suggests that $L = 1/M A^{(1/y)}$ with $1/y = 0.6$ (or 0.56) (y about 1.7). In parallel flow systems L is proportional to A (y about 1). This method tries to identify the transition between these two paradigms by using an exponent y somewhere in between (y about 1.3).

Default: *1.3*

Outputs

Stream Source Grid [raster] An indicator grid (1,0) that evaluates $A \geq (M)(L^y)$, based on the maximum upslope path length, the D8 contributing area grid inputs, and parameters M and y . This grid indicates likely stream source grid cells.

Console usage

```
processing.runalg('taudem:lengthareastreamsource', length_grid, contrib_area_grid, threshold, exp
```

See also

Move Outlets To Streams

Description

Moves outlet points that are not aligned with a stream cell from a stream raster grid, downslope along the D8 flow direction until a stream raster cell is encountered, the “max_dist” number of grid cells are examined, or the flow path exits the domain (i.e. a “no data” value is encountered for the D8 flow direction). The output file is a new outlets shapefile where each point has been moved to coincide with the stream raster grid, if possible. A field “dist_moved” is added to the new outlets shapefile to indicate the changes made to each point. Points that are already on a stream cell are not moved and their “dist_moved” field is assigned a value 0. Points that are initially not on a stream cell are moved by sliding them downslope along the D8 flow direction until one of the following occurs: a) A stream raster grid cell is encountered before traversing the “max_dist” number of grid cells. In which case, the point is moved and the “dist_moved” field is assigned a value indicating how many grid cells the point was moved. b) More than the “max_number” of grid cells are traversed, or c) the traversal ends up going out of the domain (i.e., a “no data” D8 flow direction value is encountered). In which case, the point is not moved and the “dist_moved” field is assigned a value of -1.

Parameters

D8 Flow Direction Grid [raster] A grid of D8 flow directions which are defined, for each cell, as the direction of the one of its eight adjacent or diagonal neighbors with the steepest downward slope. This grid can be obtained as the output of the “**D8 Flow Directions**” tool.

Stream Raster Grid [raster] This output is an indicator grid (1, 0) that indicates the location of streams, with a value of 1 for each of the stream cells and 0 for the remainder of the cells. This file is produced by several different tools in the “**Stream Network Analysis**” toolset.

Outlets Shapefile [vector: point] A point shape file defining points of interest or outlets that should ideally be located on a stream, but may not be exactly on the stream due to the fact that the shapefile point locations may not have been accurately registered with respect to the stream raster grid.

Maximum Number of Grid Cells to traverse [number] This input parameter is the maximum number of grid cells that the points in the input outlet shapefile will be moved before they are saved to the output outlet shapefile.

Default: *50*

Outputs

Output Outlet Shapefile [vector] A point shape file defining points of interest or outlets. This file has one point in it for each point in the input outlet shapefile. If the original point was located on a stream, then the point was not moved. If the original point was not on a stream, the point was moved downslope according to the D8 flow direction until it reached a stream or the maximum distance had been reached. This file has an additional field “*dist_moved*” added to it which is the number of cells that the point was moved. This field is 0 if the cell was originally on a stream, -1 if it was not moved because there was not a stream within the maximum distance, or some positive value if it was moved.

Console usage

```
processing.runalg('taudem:moveoutletstostreams', -p, -src, -o, -md, -om)
```

See also

Peuker Douglas

Description

Creates an indicator grid (1, 0) of upward curved grid cells according to the Peuker and Douglas algorithm.

With this tool, the DEM is first smoothed by a kernel with weights at the center, sides, and diagonals. The Peuker and Douglas (1975) method (also explained in Band, 1986), is then used to identify upwardly curving grid cells. This technique flags the entire grid, then examines in a single pass each quadrant of 4 grid cells, and unflags the highest. The remaining flagged cells are deemed “upwardly curved”, and when viewed, resemble a channel network. This proto-channel network generally lacks connectivity and requires thinning, issues that were discussed in detail by Band (1986).

Parameters

Elevation Grid [raster] A grid of elevation values. This is usually the output of the “**Pit Remove**” tool, in which case it is elevations with pits removed.

Center Smoothing Weight [number] The center weight parameter used by a kernel to smooth the DEM before the tool identifies upwardly curved grid cells.

Default: *0.4*

Side Smoothing Weight [number] The side weight parameter used by a kernel to smooth the DEM before the tool identifies upwardly curved grid cells.

Default: *0.1*

Diagonal Smoothing Weight [number] The diagonal weight parameter used by a kernel to smooth the DEM before the tool identifies upwardly curved grid cells.

Default: *0.05*

Outputs

Stream Source Grid [raster] An indicator grid (1, 0) of upward curved grid cells according to the Peuker and Douglas algorithm, and if viewed, resembles a channel network. This proto-channel network generally lacks connectivity and requires thinning, issues that were discussed in detail by Band (1986).

Console usage

```
processing.runalg('taudem:peukerdouglas', elevation_grid, center_weight, side_weight, diagonal_we
```

See also

- Band, L. E., (1986), “Topographic partition of watersheds with digital elevation models”, *Water Resources Research*, 22(1): 15-24.
- Peuker, T. K. and D. H. Douglas, (1975), “Detection of surface-specific points by local parallel processing of discrete terrain elevation data”, *Comput. Graphics Image Process.*, 4: 375-387.

Slope Area Combination

Description

Creates a grid of slope-area values = $(S_m)^m (A_n)^n$ based on slope and specific catchment area grid inputs, and parameters m and n . This tool is intended for use as part of the slope-area stream raster delineation method.

Parameters

Slope Grid [raster] This input is a grid of slope values. This grid can be obtained from the “**D-Infinity Flow Directions**” tool.

Contributing Area Grid [raster] A grid giving the specific catchment area for each cell taken as its own contribution (grid cell length or summation of weights) plus the proportional contribution from upslope neighbors that drain in to it. This grid is typically obtained from the “**D-Infinity Contributing Area**” tool.

Slope Exponent [number] The slope exponent (m) parameter which will be used in the formula: $(S_m)^m (A_n)^n$, that is used to create the slope-area grid.

Default: *2*

Area Exponent [number] The area exponent (n) parameter which will be used in the formula: $(S_m)^m (A_n)^n$, that is used to create the slope-area grid.

Default: *1*

Outputs

Slope Area Grid [raster] A grid of slope-area values = $(S_m)^m (A_n)^n$ calculated from the slope grid, specific catchment area grid, m slope exponent parameter, and n area exponent parameter.

Console usage

```
processing.runalg('taudem:slopeareacombination', slope_grid, area_grid, slope_exponent, area_exponent)
```

See also

Stream Definition By Threshold

Description

Operates on any grid and outputs an indicator (1, 0) grid identifying cells with input values \geq the threshold value. The standard use is to use an accumulated source area grid to as the input grid to generate a stream raster grid as the output. If you use the optional input mask grid, it limits the domain being evaluated to cells with mask values \geq 0. When you use a D-infinity contributing area grid (**sca*) as the mask grid, it functions as an edge contamination mask. The threshold logic is:

```
src = ((ssa >= thresh) & (mask >= s0)) ? 1:0
```

Parameters

Accumulated Stream Source Grid [raster] This grid nominally accumulates some characteristic or combination of characteristics of the watershed. The exact characteristic(s) varies depending on the stream network raster algorithm being used. This grid needs to have the property that grid cell values are monotonically increasing downslope along D8 flow directions, so that the resulting stream network is continuous. While this grid is often from an accumulation, other sources such as a maximum upslope function will also produce a suitable grid.

Threshold [number] This parameter is compared to the value in the Accumulated Stream Source grid (**ssa*) to determine if the cell should be considered a stream cell. Streams are identified as grid cells for which ssa value is \geq this threshold.

Default: *100*

Mask Grid [raster] Optional.

This optional input is a grid that is used to mask the domain of interest and output is only provided where this grid is \geq 0. A common use of this input is to use a D-Infinity contributing area grid as the mask so that the delineated stream network is constrained to areas where D-infinity contributing area is available, replicating the functionality of an edge contamination mask.

Outputs

Stream Raster Grid [raster] This is an indicator grid (1, 0) that indicates the location of streams, with a value of 1 for each of the stream cells and 0 for the remainder of the cells.

Console usage

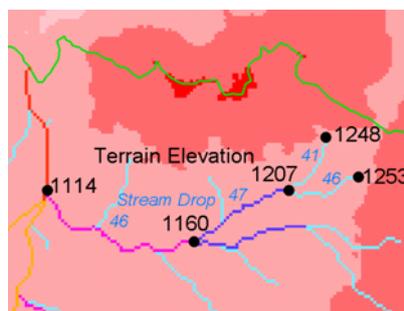
```
processing.runalg('taudem:streamdefinitionbythreshold', -ssa, -thresh, -mask, -src)
```

See also

Stream Drop Analysis

Description

Applies a series of thresholds (determined from the input parameters) to the input accumulated stream source grid (*ssa) grid and outputs the results in the *drp.txt file the stream drop statistics table. This function is designed to aid in the determination of a geomorphologically objective threshold to be used to delineate streams. Drop Analysis attempts to select the right threshold automatically by evaluating a stream network for a range of thresholds and examining the constant drop property of the resulting Strahler streams. Basically it asks the question: Is the mean stream drop for first order streams statistically different from the mean stream drop for higher order streams, using a T-test. Stream drop is the difference in elevation from the beginning to the end of a stream defined as the sequence of links of the same stream order. If the T-test shows a significant difference then the stream network does not obey this “law” so a larger threshold needs to be chosen. The smallest threshold for which the T-test does not show a significant difference gives the highest resolution stream network that obeys the constant stream drop “law” from geomorphology, and is the threshold chosen for the “objective” or automatic mapping of streams from the DEM. This function can be used in the development of stream network rasters, where the exact watershed characteristic(s) that were accumulated in the accumulated stream source grid vary based on the method being used to determine the stream network raster.



The constant stream drop “law” was identified by Broscocoe (1959). For the science behind using this to determine a stream delineation threshold, see Tarboton et al. (1991, 1992), Tarboton and Ames (2001).

Parameters

D8 Contributing Area Grid [raster] A grid of contributing area values for each cell that were calculated using the D8 algorithm. The contributing area for a cell is the sum of its own contribution plus the contribution from all upslope neighbors that drain to it, measured as a number of cells or the sum of weight loadings. This grid can be obtained as the output of the “**D8 Contributing Area**” tool. This grid is used in the evaluation of drainage density reported in the stream drop table.

D8 Flow Direction Grid [raster] A grid of D8 flow directions which are defined, for each cell, as the direction of the one of its eight adjacent or diagonal neighbors with the steepest downward slope. This grid can be obtained as the output of the “**D8 Flow Directions**” tool.

Pit Filled Elevation Grid [raster] A grid of elevation values. This is usually the output of the “**Pit Remove**” tool, in which case it is elevations with pits removed.

Accumulated Stream Source Grid [raster] This grid must be monotonically increasing along the downslope D8 flow directions. It is compared to a series of thresholds to determine the beginning of the streams. It is often generated by accumulating some characteristic or combination of characteristics of the watershed with the “**D8 Contributing Area**” tool, or using the maximum option of the “**D8 Flow Path Extreme**” tool. The exact method varies depending on the algorithm being used.

Outlets Shapefile [vector: point] A point shapefile defining the outlets upstream of which drop analysis is performed.

Minimum Threshold [number] This parameter is the lowest end of the range searched for possible threshold values using drop analysis. This technique looks for the smallest threshold in the range where the absolute value of the t-statistic is less than 2. For the science behind the drop analysis see Tarboton et al. (1991, 1992), Tarboton and Ames (2001).

Default: 5

Maximum Threshold [number] This parameter is the highest end of the range searched for possible threshold values using drop analysis. This technique looks for the smallest threshold in the range where the absolute value of the t-statistic is less than 2. For the science behind the drop analysis see Tarboton et al. (1991, 1992), Tarboton and Ames (2001).

Default: 500

Number of Threshold Values [number] The parameter is the number of steps to divide the search range into when looking for possible threshold values using drop analysis. This technique looks for the smallest threshold in the range where the absolute value of the t-statistic is less than 2. For the science behind the drop analysis see Tarboton et al. (1991, 1992), Tarboton and Ames (2001).

Default: 10

Spacing for Threshold Values [selection] This parameter indicates whether logarithmic or linear spacing should be used when looking for possible threshold values using drop analysis.

Options:

- 0 — Logarithmic
- 1 — Linear

Default: 0

Outputs

D-Infinity Drop to Stream Grid [file] This is a comma delimited text file with the following header line:

```
:: Threshold,DrainDen,NoFirstOrd,NoHighOrd,MeanDFirstOrd,MeanDHighOrd,StdDevFirstOrd,StdDevHighOrd,T
```

The file then contains one line of data for each threshold value examined, and then a summary line that indicates the optimum threshold value. This technique looks for the smallest threshold in the range where the absolute value of the t-statistic is less than 2. For the science behind the drop analysis, see Tarboton et al. (1991, 1992), Tarboton and Ames (2001).

Console usage

```
processing.runalg('taudem:streamdropanalysis', d8_contrib_area_grid, d8_flow_dir_grid, pit_filled,
```

See also

- Broscoc, A. J., (1959), “Quantitative analysis of longitudinal stream profiles of small watersheds”, Office of Naval Research, Project NR 389-042, Technical Report No. 18, Department of Geology, Columbia University, New York.
- Tarboton, D. G., R. L. Bras and I. Rodriguez-Iturbe, (1991), “On the Extraction of Channel Networks from Digital Elevation Data”, Hydrologic Processes, 5(1): 81-100.
- Tarboton, D. G., R. L. Bras and I. Rodriguez-Iturbe, (1992), “A Physical Basis for Drainage Density”, Geomorphology, 5(1/2): 59-76.
- Tarboton, D. G. and D. P. Ames, (2001), “Advances in the mapping of flow networks from digital elevation data”, World Water and Environmental Resources Congress, Orlando, Florida, May 20-24, ASCE, <http://www.engineering.usu.edu/dtarb/asce2001.pdf>.

Stream Reach and Watershed

Description

This tool produces a vector network and shapefile from the stream raster grid. The flow direction grid is used to connect flow paths along the stream raster. The Strahler order of each stream segment is computed. The subwatershed draining to each stream segment (reach) is also delineated and labeled with the value identifier that corresponds to the WSNO (watershed number) attribute in the Stream Reach Shapefile.

This tool orders the stream network according to the Strahler ordering system. Streams that don't have any other streams draining in to them are order 1. When two stream reaches of different order join the order of the downstream reach is the order of the highest incoming reach. When two reaches of equal order join the downstream reach order is increased by 1. When more than two reaches join the downstream reach order is calculated as the maximum of the highest incoming reach order or the second highest incoming reach order + 1. This generalizes the common definition to cases where more than two reaches join at a point. The network topological connectivity is stored in the Stream Network Tree file, and coordinates and attributes from each grid cell along the network are stored in the Network Coordinates file.

The stream raster grid is used as the source for the stream network, and the flow direction grid is used to trace connections within the stream network. Elevations and contributing area are used to determine the elevation and contributing area attributes in the network coordinate file. Points in the outlets shapefile are used to logically split stream reaches to facilitate representing watersheds upstream and downstream of monitoring points. The program uses the attribute field "id" in the outlets shapefile as identifiers in the Network Tree file. This tool then translates the text file vector network representation in the Network Tree and Coordinates files into a shapefile. Further attributes are also evaluated. The program has an option to delineate a single watershed by representing the entire area draining to the Stream Network as a single value in the output watershed grid.

Parameters

Pit Filled Elevation Grid [raster] A grid of elevation values. This is usually the output of the "**Pit Remove**" tool, in which case it is elevations with pits removed.

D8 Flow Direction Grid [raster] A grid of D8 flow directions which are defined, for each cell, as the direction of the one of its eight adjacent or diagonal neighbors with the steepest downward slope. This grid can be obtained as the output of the "**D8 Flow Directions**" tool.

D8 Drainage Area [raster] A grid giving the contributing area value in terms of the number of grid cells (or the summation of weights) for each cell taken as its own contribution plus the contribution from upslope neighbors that drain in to it using the D8 algorithm. This is usually the output of the "**D8 Contributing Area**" tool and is used to determine the contributing area attribute in the Network Coordinate file.

Stream Raster Grid [raster] An indicator grid indicating streams, by using a grid cell value of 1 on streams and 0 off streams. Several of the "**Stream Network Analysis**" tools produce this type of grid. The Stream Raster Grid is used as the source for the stream network.

Outlets Shapefile as Network Nodes [vector: point] Optional.

A point shape file defining points of interest. If this file is used, the tool will only delineate the stream network upstream of these outlets. Additionally, points in the Outlets Shapefile are used to logically split stream reaches to facilitate representing watersheds upstream and downstream of monitoring points. This tool **REQUIRES THAT THERE BE** an integer attribute field "id" in the Outlets Shapefile, because the "id" values are used as identifiers in the Network Tree file.

Delineate Single Watershed [boolean] This option causes the tool to delineate a single watershed by representing the entire area draining to the Stream Network as a single value in the output watershed grid. Otherwise a separate watershed is delineated for each stream reach. Default is *False* (separate watershed).

Default: *False*

Outputs

Stream Order Grid [raster] The Stream Order Grid has cells values of streams ordered according to the Strahler order system. The Strahler ordering system defines order 1 streams as stream reaches that don't have any other reaches draining in to them. When two stream reaches of different order join the order of the downstream reach is the order of the highest incoming reach. When two reaches of equal order join the downstream reach order is increased by 1. When more than two reaches join the downstream reach order is calculated as the maximum of the highest incoming reach order or the second highest incoming reach order + 1. This generalizes the common definition to cases where more than two flow paths reaches join at a point.

Watershed Grid [raster] This output grid identified each reach watershed with a unique ID number, or in the case where the delineate single watershed option was checked, the entire area draining to the stream network is identified with a single ID.

Stream Reach Shapefile [vector] This output is a polyline shapefile giving the links in a stream network. The columns in the attribute table are:

- LINKNO — Link Number. A unique number associated with each link (segment of channel between junctions). This is arbitrary and will vary depending on number of processes used
- DSLINKNO — Link Number of the downstream link. -1 indicates that this does not exist
- USLINKNO1 — Link Number of first upstream link. (-1 indicates no link upstream, i.e. for a source link)
- USLINKNO2 — Link Number of second upstream link. (-1 indicates no second link upstream, i.e. for a source link or an internal monitoring point where the reach is logically split but the network does not bifurcate)
- DSNODEID — Node identifier for node at downstream end of stream reach. This identifier corresponds to the "id" attribute from the Outlets shapefile used to designate nodes
- Order — Strahler Stream Order
- Length — Length of the link. The units are the horizontal map units of the underlying DEM grid
- Magnitude — Shreve Magnitude of the link. This is the total number of sources upstream
- DS_Cont_Ar — Drainage area at the downstream end of the link. Generally this is one grid cell upstream of the downstream end because the drainage area at the downstream end grid cell includes the area of the stream being joined
- Drop — Drop in elevation from the start to the end of the link
- Slope — Average slope of the link (computed as drop/length)
- Straight_L — Straight line distance from the start to the end of the link
- US_Cont_Ar — Drainage area at the upstream end of the link
- WSNO — Watershed number. Cross reference to the **w.shp* and **w* grid files giving the identification number of the watershed draining directly to the link
- DOUT_END — Distance to the eventual outlet (i.e. the most downstream point in the stream network) from the downstream end of the link
- DOUT_START — Distance to the eventual outlet from the upstream end of the link
- DOUT_MID — Distance to the eventual outlet from the midpoint of the link

Network Connectivity Tree [file] This output is a text file that details the network topological connectivity is stored in the Stream Network Tree file. Columns are as follows:

- Link Number (Arbitrary — will vary depending on number of processes used)
- Start Point Number in Network coordinates (**coord.dat*) file (Indexed from 0)
- End Point Number in Network coordinates (**coord.dat*) file (Indexed from 0)

- Next (Downstream) Link Number. Points to Link Number. -1 indicates no links downstream, i.e. a terminal link
- First Previous (Upstream) Link Number. Points to Link Number. -1 indicates no upstream links
- Second Previous (Upstream) Link Numbers. Points to Link Number. -1 indicates no upstream links. Where only one previous link is -1, it indicates an internal monitoring point where the reach is logically split, but the network does not bifurcate
- Strahler Order of Link
- Monitoring point identifier at downstream end of link. -1 indicates downstream end is not a monitoring point
- Network magnitude of the link, calculated as the number of upstream sources (following Shreve)

Network Coordinates [file] This output is a text file that contains the coordinates and attributes of points along the stream network. Columns are as follows:

- X coordinate
- Y Coordinate
- Distance along channels to the downstream end of a terminal link
- Elevation
- Contributing area

Console usage

```
processing.runalg('taudem:streamreachandwatershed', -fel, -p, -ad8, -src, -o, -sw, -ord, -w, -net,
```

See also

.

Compositore di stampe

Con il Compositore di Stampa puoi creare mappe belle e atlanti, che possono essere stampati o salvati in formato PDF, immagine o SVG. Questo è utile per condividere informazioni geografiche prodotte con QGIS che possono essere incluse in relazioni o pubblicate.

Il compositore di stampe ti permette di creare layout di stampa molto potenti. Ti permette di aggiungere elementi come la mappa che hai creato in QGIS, etichette di testo, immagini, legende, barre di scala, forme semplici, frecce, tabelle degli attributi e cornici HTML. Puoi scegliere la dimensione, raggruppare e allineare la posizione di ogni elemento e aggiustarne le proprietà. Puoi esportare la composizione che hai creato in vari formati, PostScript, PDF o SVG (l'esportazione SVG non funziona sempre correttamente a causa della recente versione di Qt4: devi provare e verificare di persona sul tuo computer). Puoi salvare i layout come modelli e caricarli in seguito in altre sessioni. Infine, puoi anche creare facilmente diverse mappe basate sullo stesso modello grazie al generatore atlante. Vedi l'elenco degli strumenti in [table_composer_1](#):

Icona	Azione	Icona	Azione
	Salva progetto		Nuova composizione
	Duplica composizione		Gestore di stampe
	Caraica da modello		Salva come modello
	Stampa o esporta come		Esporta come immagine
	Esporta come SVG		Esporta come PDF
	Annulla l'ultimo cambiamento		Rispristina l'ultimo cambiamento
	Vista ad estensione massima		Zoom al 100%
	Ingrandisci		Rimpicciolisci
	Aggiorna la vista		Zoom a una regione specifica
	Sposta		Sposta contenuto elemento
	Scegli/Sposta oggetto		Aggiungi immagine
	Aggiungi una nuova immagine dalla mappa di QGIS		Aggiungi nuova legenda vettoriale
	Aggiungi etichetta		Aggiungi forma base
	Aggiungi nuova barra di scala alla composizione di stampa		Aggiungi tabella attributi
	Aggiungi freccia		Rimuovi raggruppamento
	Aggiungi una cornice HTML		Sblocca tutti gli elementi
	Raggruppa oggetti		Muovi in basso
	Blocca gli oggetti selezionati		Porta in fondo
	Muovi in alto		Allinea a destra
	Porta in cima		Allinea su asse orizzontale
	Allinea a sinistra		Allinea in basso
	Allinea su asse verticale		Prima geometria
	Allinea in alto		Geometria successiva
	Anteprima atlante		Stampa atlante
	Geometria precedente		Impostazioni atlante
	Ultima geometria		
	Esporta atlante come immagini		

Strumenti del Compositore di Stampe

Tutti gli strumenti del compositore di stampe sono disponibili sia dai menu che dalle icone. Puoi disattivare la barra degli strumenti facendo click con il tasto destro del mouse sulla barra stessa.

19.1 Primi passi

19.1.1 Aprire un nuovo modello di stampa

Prima di iniziare a lavorare con il compositore di stampe, devi caricare qualche raster e vettore nella mappa di QGIS e aggiustarne qualche proprietà. Una volta che hai scelto la visualizzazione e la simbologia clicca sull'icona

 Nuova composizione di stampa oppure clicca su *File* → *Nuova composizione di stampa*. Ti verrà chiesto di inserire un titolo per la composizione.

19.1.2 Panoramica del compositore di stampe

All'apertura il compositore di stampa presenta una mappa bianca che rappresenta la superficie del foglio da utilizzare per la stampa. Trovi icone per aggiungere voci mappa compositore; l'attuale QGIS mappa, testi, immagini, leggende, barre di scala, forme base, frecce, tabelle di attributi e frame HTML. Negli strumenti trovi la barra degli strumenti per navigare, lo zoom e panoramica della vista, la barra degli strumenti per selezionare un elemento della mappa compositore e per spostare il contenuto della mappa.

Figure_composer_overview mostra la vista iniziale del compositore di stampa prima dell'aggiunta di elementi.

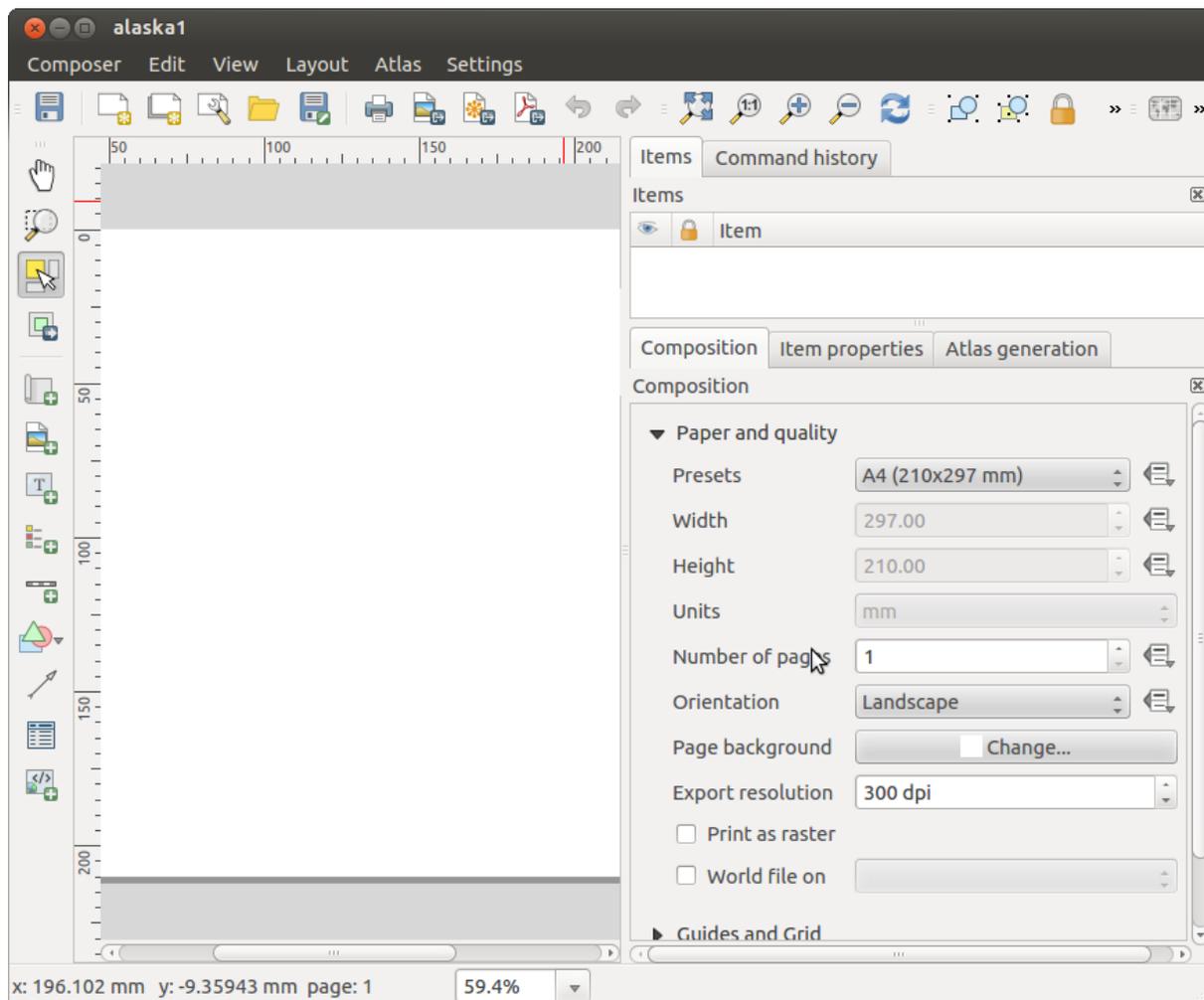


Figura 19.1: Compositore di stampe 

A destra della mappa accanto la tela si trovano due pannelli. Il pannello superiore contiene le schede: `guiabel: degli articoli e: guiabel: Command Storia dell'Attore e il pannello inferiore contiene le schede: guiabel: Composizione`; `guiabel: Proprietà oggetto`, `guiabel: Generazione atlante e: guiabel: Oggetti`.

- La scheda *Items* mostra una lista di tutti gli oggetti aggiunti al foglio.
- La scheda: `guiabel: Storico dei comandi` visualizza una cronologia di tutte le modifiche applicate al compositore. Con un clic del mouse, è possibile annullare e ripetere le operazioni avanti e indietro.
- La scheda `:guiabel: Composizione` consente di impostare il formato del foglio, l'orientamento, lo sfondo della pagina, il numero di pagine e qualità di stampa per il file di output in dpi. Inoltre, puoi attivare la

icasella :guilabel: *Stampa come raster*. Ciò significa che tutti gli articoli saranno convertiti in raster prima di stampare o salvare in PostScript o PDF. In questa scheda, puoi anche personalizzare le impostazioni per reticolo e guide.

- La scheda *Proprietà oggetto* mostra le proprietà dell'oggetto selezionato. Clic  *Seleziona/Sposta oggetto* icona per selezionare un elemento (ad esempio, la leggenda, barra di scala o etichetta) sul foglio. Quindi fare clic sulla scheda *Proprietà oggetto* e personalizza le impostazioni per l'elemento selezionato.
- La scheda *Generazione atlante* ti permette di abilitare la generazioni di un atlante per la composizione attuale avrai accesso a tutti i suoi parametri.
- Infine, puoi salvare la composizione di stampa con il bottone : sup: *Salva progetto*.

Nella parte inferiore della finestra del compositore di stampe, troverai una barra di stato con la posizione del mouse, il numero attuale della pagine e un menu a tendina per selezionare il livello di zoom.

Puoi aggiungere elementi multipli al compositore. Puoi anche visualizzare più di una mappa o leggenda o barra di scala nella vista del compositore su una o più pagine. Ogni elemento ha le sue proprietà e, nel caso della mappa, la sua estensione. Se vuoi rimuovere un qualsiasi elemento dalla vista selezionalo e premi il pulsante *Delete* o *Backspace*.

Strumenti per l'esplorazione del layout di stampa

Per navigare nella mappa, il compositore ti offre diversi strumenti:

-  *Ingrandisci*
-  *Rimpicciolisci*
-  *Zoom completo*
-  *Zoom al 100%*
-  *Aggiorna la vista*, che serve nel caso in cui la vista nel layout non rispecchi quanto presente nella vista mappa di QGIS
-  *Sposta compositore*
-  *Zoom alla selezione* (zoom a una specifica regione del compositore)

Puoi cambiare il livello di zoom usando la rotellina del mouse o il menu a tendina della barra di stato. Se devi cambiare alla modalità di spostamento mentre stai lavorando tieni premuta la Barra spaziatrice o premi la rotellina del mouse. Con la combinazione di tasti *Ctrl+Barra spaziatrice*, puoi temporaneamente cambiare alla modalità zoom alla selezione e con la combinazione *Ctrl+Shift+Barra spaziatrice* passi alla modalità di zoom out.

19.1.3 Esempio

Per creare una mappa segui le seguenti istruzioni.

1. Nella Barra degli strumenti *Oggetti del compositore*, selezionare il  *Aggiungi nuova mappa* e disegna un rettangolo sulla tela tenendo premuto il tasto sinistro del mouse. All'interno del rettangolo disegnato la visualizzazione della mappa QGIS alla tela.
2. Seleziona  *Aggiungi nuova barra di scala* e posiziona l'oggetto con il pulsante sinistro del mouse sul foglio del compositore di stampe. Una barra di scala sarà aggiunta.
3. Seleziona il bottone  *Aggiungi nuova leggenda* e disegna un rettangolo sul foglio tenendo premuto il pulsante sinistro del mouse. Nel rettangolo verrà disegnata una leggenda.

4. Seleziona  Seleziona/Muovi oggetto per selezionare la mappa sul foglio e spostarla.
5. Mentre l'oggetto mappa è selezionato puoi modificare le dimensioni della mappa. Fai clic tenendo premuto il tasto sinistro del mouse, in un piccolo rettangolo bianco in uno degli angoli della mappa e scegli una nuova posizione per modificare la sua dimensione.
6. Scegli la scheda *Proprietà oggetto* e imposta *Rotazione mappa* di '15.00°'. Vedrai cambiare l'orientamento del contenuto della mappa.
7. Infine, puoi salvare la composizione di stampa con il bottone : sup: *Salva progetto*.

19.1.4 Opzioni compositore di stampe

Dal menu *Settings* → *Opzioni compositore* puoi scegliere alcune opzioni che verranno usate in maniera predefinita durante il flusso di lavoro.

- *Opzioni predefinite del compositore* ti permette di scegliere il carattere da usare.
- Con l'opzione *Reticolo* puoi selezionare lo stile e il colore del reticolo.
- Da *Opzioni predefinite per il reticolo* puoi specificare la spaziatura, lo scostamento e la tolleranza del reticolo. Ci sono tre tipi di reticolo: **Punti**, **Pieno** e **Croci**.
- *Opzioni predefinite per le guide* ti permette di scegliere la tolleranza delle guide.

19.1.5 Scheda Composizione — Impostazioni generali

Nella scheda *Composizione* puoi scegliere le impostazioni generali della tua composizione di stampa.

- Puoi scegliere una delle opzioni *Preimpostazioni* per i tuoi fogli di carta oppure puoi inserire manualmente i valori di *larghezza* e *altezza*.
- Puoi ora dividere la composizione in diverse pagine. Per esempio, la prima pagina mostra una mappa, la seconda mostra la tabella degli attributi di un vettore e la terza mostra una cornice HTML della tua azienda. Imposta il *Numero di pagine* con il valore che preferisci. Puoi scegliere l'*Orientamento* e la *Risoluzione di esportazione*. Quando spuntata, la casella di controllo *stampa come raster* permette di trasformare in raster tutti gli elementi prima del salvataggio o esportazione in PostScript o PDF.
- *Reticolo* ti permette di personalizzare: *spaziatura*, *offset* e *tolleranza*.
- In *Tolleranza di snap* puoi cambiare la *Tolleranza* ovvero la distanza massima ai quali gli oggetti vengono agganciati alle guide.

Puoi attivare l'aggancio al reticolo e/o le guide intelligenti dal menu *Visualizza*. In questo menu puoi anche scegliere se mostrare o nascondere il reticolo e le guide intelligenti.

19.1.6 Composer items common options

Gli oggetti del compositore hanno proprietà comuni che trovi nella scheda *Proprietà oggetto*: *Posizione e dimensione*, *Rotazione*, *Cornice*, *Sfondo*, *ID oggetto* e *visualizzazione* (Vedi [figure_composer_common_1](#))

- La voce *Posizione e dimensione* ti permette di scegliere la posizione della cornice che contiene l'oggetto. Puoi anche scegliere quale deve essere il *Punto di riferimento* delle coordinate **X** e **Y**.
- Con *Rotazione* puoi impostare la rotazione dell'elemento (in gradi).
- La casella di controllo *Cornice* mostra o nasconde la cornice. Clicca sui pulsanti [**Colore**] e [**Spessore**] per specificarne i valori.
- La casella di controllo *Sfondo* ti permette di attivare/disattivare un colore di sfondo. Clicca sul pulsante [**Colore...**] per visualizzare una finestra di dialogo dove puoi cliccare su un colore o sceglierne uno fra quelli predefiniti. Puoi anche aggiustare la trasparenza attraverso il campo **alpha**.

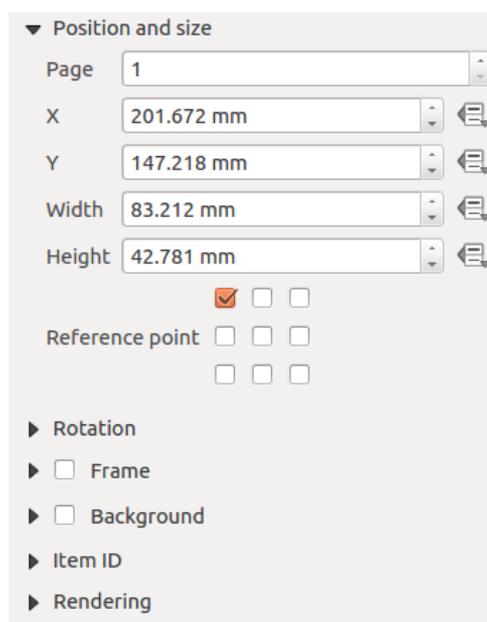


Figura 19.2: Finestra di dialogo delle proprietà dell'oggetto 🐧

- Usa *ID oggetto* per creare una relazione con altri oggetti del compositore. Questa opzione è usata con QGIS server e altri potenziali web client. Puoi impostare un ID di un oggetto (per esempio una mappa e un'etichetta) e poi il web client sarà in grado di impostare un corretto stile per i dati (esempio le etichette). Il comando `GetProjectSettings` elencherà gli oggetti e quali ID sono disponibili nel layout.
- Puoi selezionare la modalità *Visualizzazione* con la giusta opzione. Vedi [Rendering_Mode](#).

Nota:

- If you checked *Use live-updating color chooser dialogs* in the QGIS general options, the color button will update as soon as you choose a new color from **Color Dialog** windows. If not, you need to close the **Color Dialog**.
- The  *Data defined override* icon next to a field means that you can associate the field with data in the map item or use expressions. These are particularly helpful with atlas generation (See [atlas_data_defined_overrides](#)).

19.2 Modalità Visualizzazione

QGIS ti permette di effettuare visualizzazioni avanzate per ogni elemento del compositore proprio come per i vettori e per i raster.

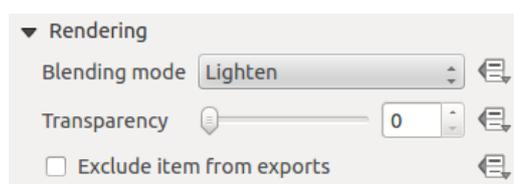


Figura 19.3: Modalità Visualizzazione 🐧

- *Trasparenza* : con questo strumento puoi decidere il grado di visibilità del vettore. Usa questo cursore per adattare la trasparenza del vettore. Puoi anche impostare un valore preciso di trasparenza nella casella presente a destra del cursore.

-  *Exclude item from exports*: You can decide to make an item not visible in all exports. After activating this checkbox, the item will not be included in PDF's, prints etc..
- *Modalità fusione*: grazie a questi strumenti usati soprattutto in ambito grafico, potrai creare effettivi visivi speciali. I pixel del vettore più in alto vengono mischiati con i pixel di tutti gli elementi sottostanti. Hai molte scelte diverse.
 - Normale: è la modalità fusione predefinita che usa il canale alpha del pixel più in alto fondendolo con quello sotto. I colori non sono quindi mescolati.
 - Schiarisci: seleziona il valore massimo di ogni componente dal basso verso l'alto. Il risultato può apparire frastagliato e duro.
 - Scolorisci: i pixel chiari provenienti dal vettore sorgente vengono dipinti sopra la destinazione, mentre i pixel più scuri no. Questa modalità è molto utile per mescolare le trame di un vettore con un altro (per esempio un raster di ombreggiatura con un altro layer).
 - Scherma: questa modalità schiarirà e saturerà i pixel sottostanti in base a quanto sono chiari i pixel di sopra. In questo modo, i pixel più chiari in cima aumenteranno la saturazione e schiariranno i pixel sottostanti. Otterrai il miglior risultato se i pixel in cima non sono troppo chiari, altrimenti l'effetto sarà troppo estremo.
 - Addizione: questa modalità addiziona semplicemente il valore dei pixel di un vettore con i valori dei layer sottostanti. Se i valori sono maggiori di uno (ovvero quando si lavora con bande RGB) verrà mostrato il bianco. Questa modalità è ottima per evidenziare dei particolari.
 - Scurisci: il pixel finale conserva il valore minore dei pixel del layer in cima e in fondo. Come la modalità schiarisci, il risultato tende a essere frastagliato e duro.
 - Moltiplica: qui, il valore di ogni pixel del layer in cima viene moltiplicato per il valore dei pixel corrispondenti di tutti i layer sottostanti. Il risultato tende quindi a essere piuttosto scuro.
 - Brucia: i colori più scuri del layer in cima scuriranno i layer sottostanti. Questa modalità è utile per aggiustare e colorare i layer sottostanti.
 - Sovrapponi: è una combinazione delle modalità moltiplica e scolorisci. Le parti chiare risulteranno ancora più chiare e quelle scure ancora più scure.
 - Luce diffusa: molto simile alla modalità sovrapponi, ma invece di combinare le modalità moltiplica/scolorisci, combina brucia/scherma. Il risultato è una luce chiara e luccicante su tutta l'immagine.
 - Luce intensa: anche questa modalità è simile alla modalità sovrapponi. Proietta una luce molto intensa su tutta l'immagine.
 - Differenza: vengono sottratti i pixel in cima da quelli sul fondo, oppure al contrario, ma in modo da ottenere sempre valori positivi. Questa modalità non ha effetti con il nero, perché la differenza di questo colore con tutti gli altri è sempre zero.
 - Sottrai: questa modalità sottrae semplicemente i valori di un pixel dagli altri pixel. Se il valore dovesse essere negativo verrà visualizzato il nero.

19.3 Oggetti del compositore

19.3.1 The Map item

Clicca sul pulsante  *Aggiungi mappa* presente nella barra degli strumenti del compositore per aggiungere una mappa. Ora tieni premuto il pulsante del mouse e trascina il rettangolo corrispondente per aggiungere la mappa. Per visualizzare la mappa attuale puoi scegliere fra tre differenti modalità accessibili dalla scheda *Proprietà oggetto*:

- **Rettangolo** visualizza un rettangolo vuoto con la scritta 'La mappa verrà stampata qui'.

- **Cache** disegna la mappa alla risoluzione corrente dello schermo. Se ingrandisci/rimpiccolisci la finestra del compositore, la mappa non viene ridisegnata, ma l'immagine viene scalata.
- **Visualizza** a differenza del metodo cache, in questo caso se ridimensioni la finestra del compositore, la mappa viene ridisegnata.

Cache è la modalità predefinita per ogni nuova composizione di stampa.

Puoi ridimensionare la mappa in un momento successivo cliccando sul pulsante  *Seleziona/Sposta oggetto*, selezionando un elemento e trascinando uno dei quadrati agli angoli della mappa. Una volta selezionata una mappa, puoi regolarne ulteriori proprietà nella scheda *Opzioni oggetto*.

To move layers within the map element, select the map element, click the  *Move item content* icon and move the layers within the map item frame with the left mouse button. After you have found the right place for an item, you can lock the item position within the Print Composer canvas. Select the map item and use the toolbar  *Lock Selected Items* or the *Items* tab to Lock the item. A locked item can only be selected using the *Items* tab. Once selected you can use the *Items* tab to unlock individual items. The  *Unlock All Items* icon will unlock all locked composer items.

Proprietà principali

The *Main properties* dialog of the map *Item Properties* tab provides the following functionalities (see [figure_composer_map_1](#)):

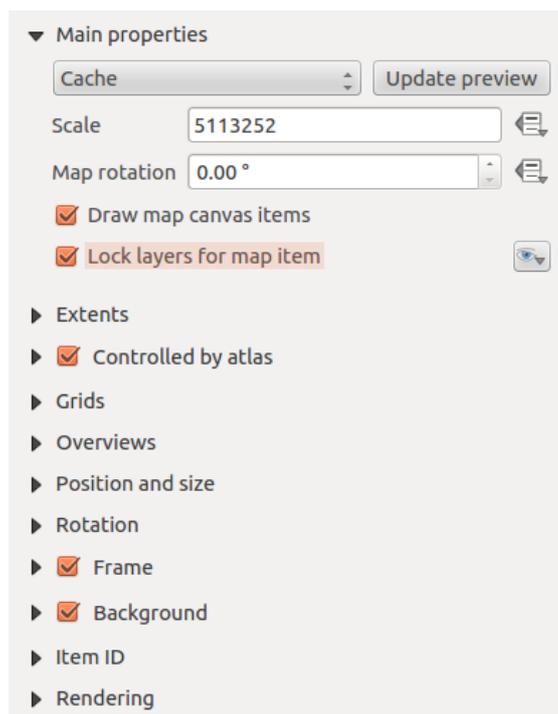


Figura 19.4: Scheda proprietà oggetti 

- L'area **Anteprima** ti permette di scegliere fra le modalità ' Rettangolo', 'Cache' e 'Visualizza' descritte sopra. Se cambi la vista della mappa in QGIS cambiando le proprietà dei vettori e dei raster, puoi aggiornare la vista del compositore selezionando l'elemento corrispondente e premendo il pulsante [**Aggiorna anteprima**].
- Il campo *Scala* ti permette di inserire una scala manuale.
- Il campo *Rotazione* ti permette di ruotare l'elemento della mappa in senso orario e in gradi. Puoi aggiungere la cornice delle coordinate solamente con il valore predefinito 0.

- *Disegna elementi sulla mappa* ti permette di visualizzare le note testuali che hai aggiunto sulla mappa di QGIS.
- You can choose to lock the layers shown on a map item. Check *Lock layers for map item*. After this is checked, any layer that would be displayed or hidden in the main QGIS window will not appear or be hidden in the map item of the Composer. But style and labels of a locked layer are still refreshed according to the main QGIS interface.
- The  button allows you to add quickly all the presets views you have prepared in QGIS. Clicking on the  button you will see the list of all the preset views: just select the preset you want to display. The map canvas will automatically lock the preset layers by enabling the *Lock layers for map item*: if you want to unselect the preset, just uncheck the and press on the  button. See *Legenda* to find out how to create presets views.

Estensione mappa

The *Extents* dialog of the map item tab provides the following functionalities (see [figure_composer_map_2](#)):

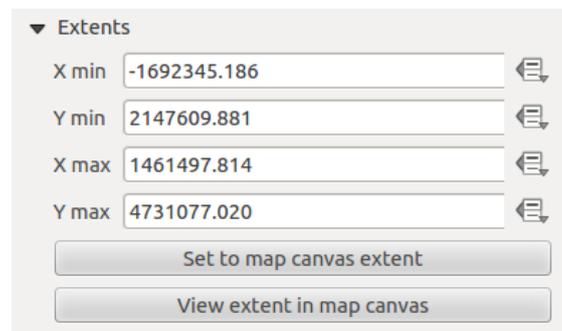


Figura 19.5: Finestra di dialogo estensione mappa 

- The **Map extents** area allows you to specify the map extent using X and Y min/max values and by clicking the [**Set to map canvas extent**] button. This button sets the map extent of the composer map item to the extent of the current map view in the main QGIS application. The button [**View extent in map canvas**] does exactly the opposite, it updates the extent of the map view in the QGIS application to the extent of the composer map item.

If you change the view on the QGIS map canvas by changing vector or raster properties, you can update the Print Composer view by selecting the map element in the Print Composer and clicking the [**Update preview**] button in the map *Item Properties* tab (see [figure_composer_map_1](#)).

Grids

The *Grids* dialog of the map *Item Properties* tab provides the the possibility to add several grids to a map item.

- With the plus and minus button you can add or remove a selected grid.
- With the up and down button you can move a grid in the list and set the drawing priority.

When you double click on the added grid you can give it another name.

After you have added a grid, you can active the checkbox *Show grid* to overlay a grid onto the map element. Expand this option to provides a lot of configuration options, see [Figure_composer_map_4](#).

As grid type, you can specify to use a solid line or cross. Symbology of the grid can be chosen. See section [Rendering_Mode](#). Furthermore, you can define an interval in the X and Y directions, an X and Y offset, and the width used for the cross or line grid type.

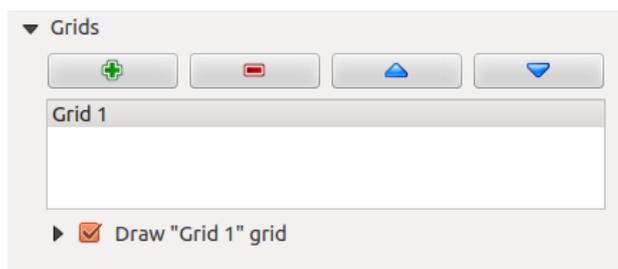


Figura 19.6: Map Grids Dialog 

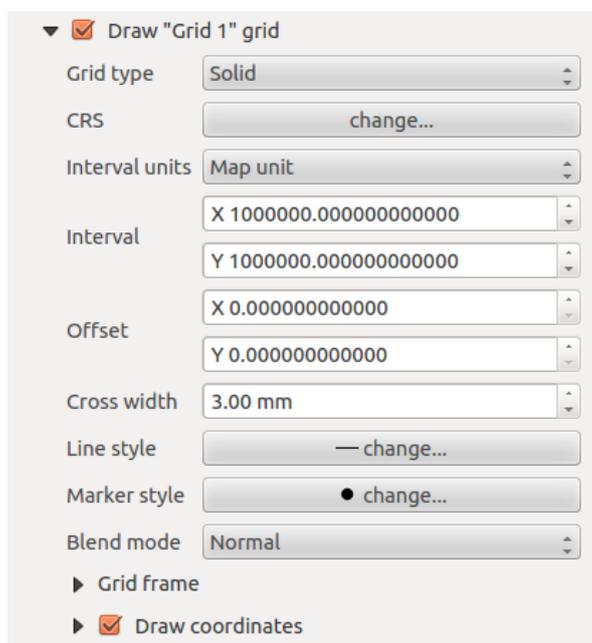


Figura 19.7: Draw Grid Dialog 

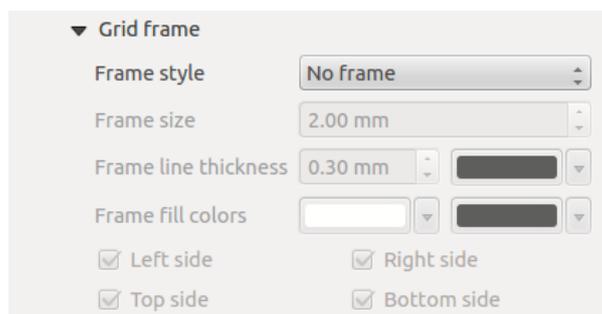


Figura 19.8: Grid Frame Dialog 

- There are different options to style the frame that holds the map. Following options are available: No Frame, Zebra, Interior ticks, Exterior ticks, Interior and Exterior ticks and Lineborder.
- Advanced rendering mode is also available for grids (see section [Rendering_mode](#)).
- La casella di controllo  *Scrivi coordinate* ti permette di aggiungere le coordinate alla mappa. Le coordinate possono essere scritte dentro o fuori la mappa e puoi scegliere di visualizzare orizzontali o verticali individualmente per ogni lato della mappa. Puoi scegliere le unità in metri o in gradi. Infine puoi anche scegliere il colore del reticolo, il carattere delle coordinate la distanza rispetto alla cornice e la loro precisione.

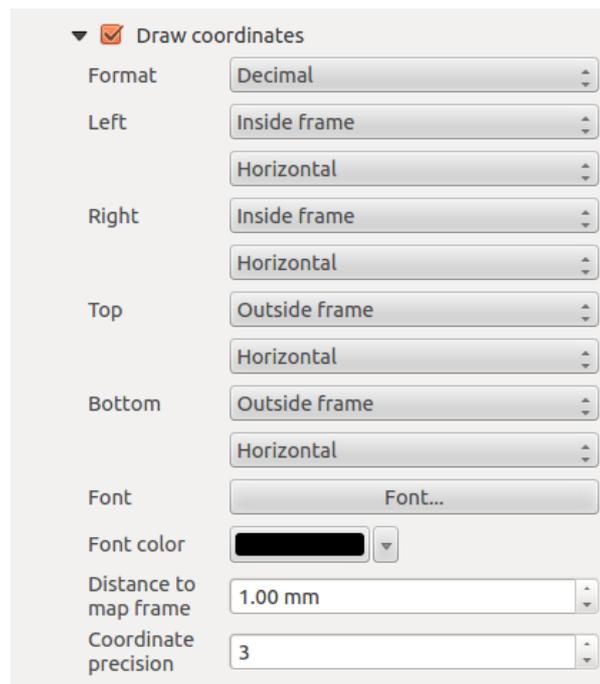


Figura 19.9: Grid Draw Coordinates dialog 

Overviews

The *Overviews* dialog of the map *Item Properties* tab provides the following functionalities:

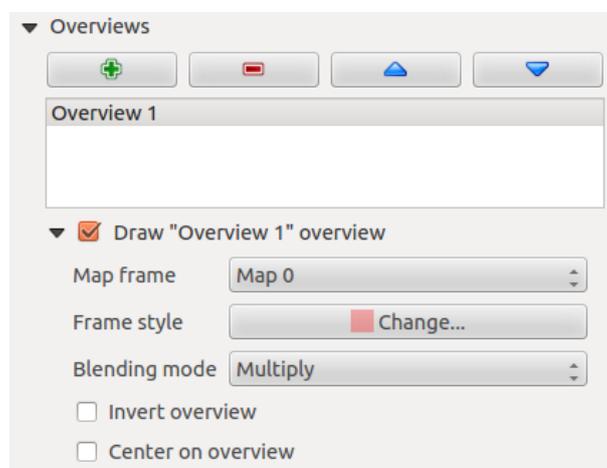


Figura 19.10: Map Overviews Dialog 

You can choose to create an overview map, which shows the extents of the other map(s) that are available in the composer. First you need to create the map(s) you want to include in the overview map. Next you create the map you want to use as the overview map, just like a normal map.

- With the plus and minus button you can add or remove an overview.
- With the up and down button you can move an overview in the list and set the drawing priority.

Open *Overviews* and press the green plus icon-button to add an overview. Initially this overview is named 'Overview 1' (see [Figure_composer_map_7](#)). You can change the name when you double-click on the overview item in the list named 'Overview 1' and change it to another name.

When you select the overview item in the list you can customize it.

- The *Draw* "<name_overview>" overview needs to be activated to draw the extent of selected map frame.
- The *Map frame* combo list can be used to select the map item whose extents will be drawn on the present map item.
- The *Frame Style* allows you to change the style of the overview frame.
- The *Blending mode* allows you to set different transparency blend modes. See [Rendering_Mode](#).
- The *Invert overview* creates a mask around the extents when activated: the referenced map extents are shown clearly, whereas everything else is blended with the frame color.
- The *Center on overview* puts the extent of the overview frame in the center of the overview map. You can only activate one overview item to center, when you have added several overviews.

19.3.2 The Label item

Per aggiungere un'etichetta, clicca sull'icona  *Aggiungi etichetta* e scegli dove inserirla cliccando sulla mappa con il mouse. Puoi personalizzare le etichette nella scheda *Proprietà oggetto*.

The *Item Properties* tab of a label item provides the following functionality for the label item (see [Figure_composer_label](#)):

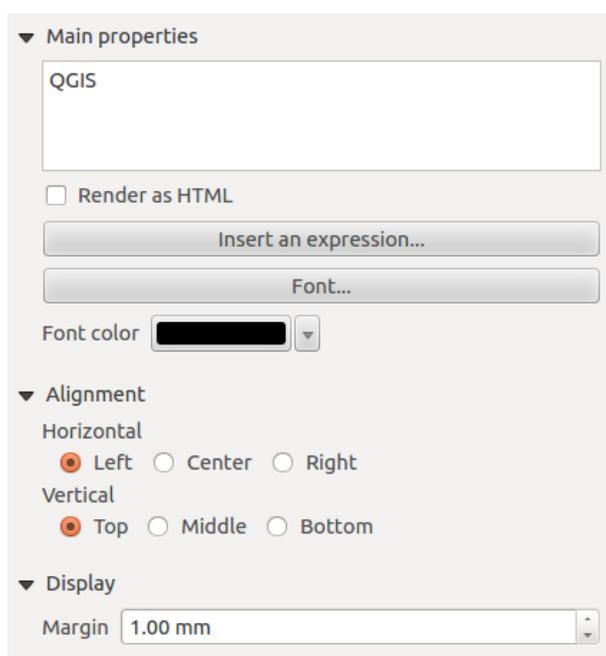


Figura 19.11: Scheda proprietà etichette 

Proprietà principali

- La voce proprietà principali è il posto dove devi inserire il testo (HTML oppure normale) o l'espressione che vuoi che compaia sull'etichetta.
- Puoi inserire del testo HTML per le etichette: spunta la casella di controllo *Visualizza come HTML*. Ora puoi inserire un URL, un'immagine collegata a un sito web oppure qualcosa di più complesso.
- You can also insert an expression. Click on **[Insert an expression]** to open a new dialog. Build an expression by clicking the functions available in the left side of the panel. Two special categories can be useful, particularly associated with the atlas functionality: geometry functions and records functions. At the bottom, a preview of the expression is shown.
- Define *Font* by clicking on the **[Font...]** button or a *Font color* selecting a color using the color selection tool.

Allineamento e visualizzazione

- La voce *Allineamento* ti permette di scegliere l'allineamento verticale o orizzontale.
- In the **Display** tag, you can define a margin in mm. This is the margin from the edge of the composer item.

19.3.3 The Image item

Per aggiungere un'immagine, clicca sull'icona  *Aggiungi immagine* e scegli dove inserirla cliccando sulla mappa con il mouse. Puoi personalizzare l'immagine nella scheda *Proprietà oggetto*.

The image *Item Properties* tab provides the following functionalities (see [figure_composer_image_1](#)):

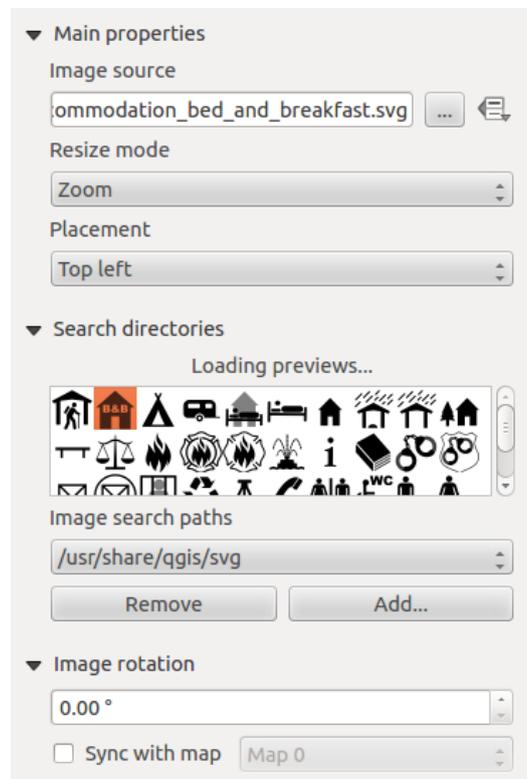


Figura 19.12: Scheda proprietà immagine 

You first have to select the image you want to display. There are several ways to set the *image source* in the **Main properties** area.

1. Use the browse button  of *image source* to select a file on your computer using the browse dialog. The browser will start in the SVG-libraries provided with QGIS. Besides SVG, you can also select other image formats like .png or .jpg.
2. You can enter the source directly in the *image source* text field. You can even provide a remote URL-address to an image.
3. From the **Search directories** area you can also select an image from *loading preview..* to set the image source.
4. Use the data defined button  to set the image source from a record or using a regular expression.

With the *Resize mode* option, you can set how the image is displayed when the frame is changed, or choose to resize the frame of the image item so it matches the original size of the image.

You can select one of the following modes:

- Zoom: Enlarges the image to the frame while maintaining aspect ratio of picture.
- Stretch: Stretches image to fit inside the frame, ignores aspect ratio.
- Clip: Use this mode for raster images only, it sets the size of the image to original image size without scaling and the frame is used to clip the image, so only the part of the image inside the frame is visible.
- Zoom and resize frame: Enlarges image to fit frame, then resizes frame to fit resultant image.
- Resize frame to image size: Sets size of frame to match original size of image without scaling.

Selected resize mode can disable the item options 'Placement' and 'Image rotation'. The *Image rotation* is active for the resize mode 'Zoom' and 'Clip'.

With *Placement* you can select the position of the image inside it's frame. The **Search directories** area allows you to add and remove directories with images in SVG format to the picture database. A preview of the pictures found in the selected directories is shown in a pane and can be used to select and set the image source.

Images can be rotated with the *Image rotation* field. Activating the  *Sync with map* checkbox synchronizes the rotation of a picture in the QGIS map canvas (i.e., a rotated north arrow) with the appropriate Print Composer image.

It is also possible to select a north arrow directly. If you first select a north arrow image from **Search directories** and then use the browse button  of the field *Image source*, you can now select one of the north arrow from the list as displayed in [figure_composer_image_2](#).

Nota: Many of the north arrows do not have an 'N' added in the north arrow, this is done on purpose for languages that do not use an 'N' for North, so they can use another letter.

19.3.4 The Legend item

Per aggiungere una legenda, clicca su  **Aggiungi legenda**, e scegli dove posizionarla con il tasto sinistro del mouse. Puoi personalizzare l'aspetto della legenda nella scheda *Proprietà oggetto*.

The *Item properties* of a legend item tab provides the following functionalities (see [figure_composer_legend_1](#)):

Proprietà principali

The *Main properties* dialog of the legend *Item Properties* tab provides the following functionalities (see [figure_composer_legend_2](#)):

In Main properties you can:

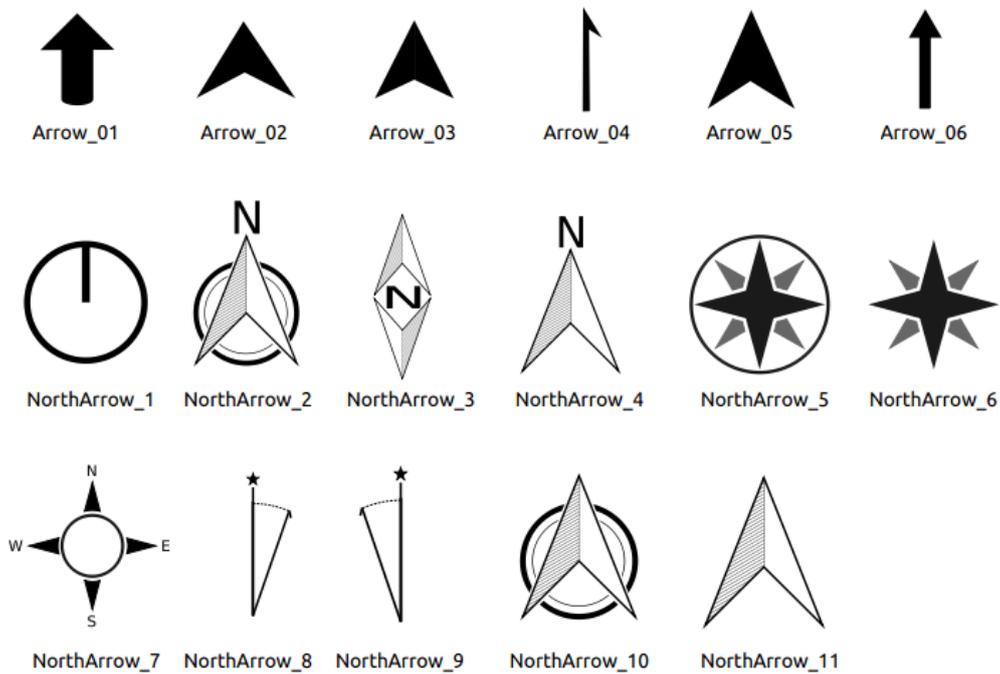


Figura 19.13: North arrows available for selection in provided SVG library

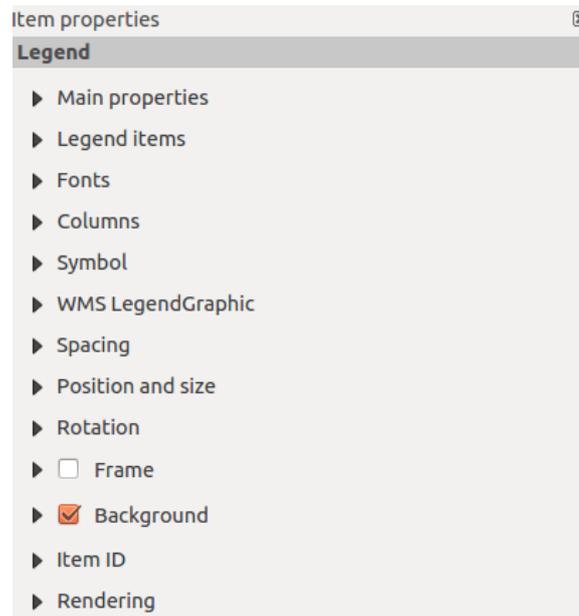


Figura 19.14: Scheda proprietà legenda 

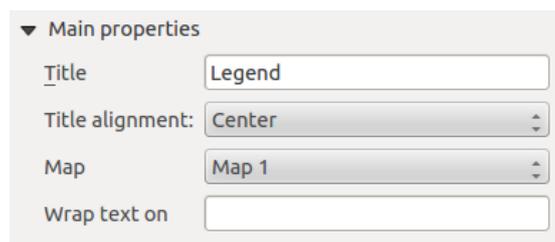


Figura 19.15: Finestra di dialogo proprietà principali della legenda 

- Change the title of the legend.
- Set the title alignment to Left, Center or Right.
- You can choose which *Map* item the current legend will refer to in the select list.
- You can wrap the text of the legend title on a given character.

Oggetti della legenda

The *Legend items* dialog of the legend *Item Properties* tab provides the following functionalities (see [figure_composer_legend_3](#)):

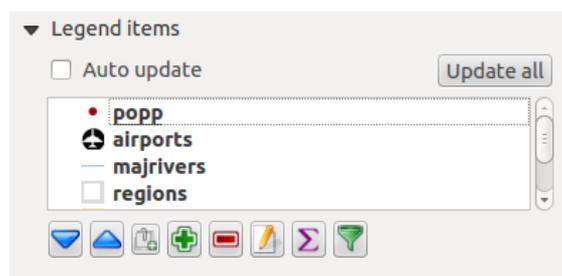


Figura 19.16: Finestra di dialogo oggetti legenda 🐧

- The legend will be updated automatically if *Auto-update* is checked. When *Auto-update* is unchecked this will give you more control over the legend items. The icons below the legend items list will be activated.
- The legend items window lists all legend items and allows you to change item order, group layers, remove and restore items in the list, edit layer names and add a filter.
 - The item order can be changed using the [**Up**] and [**Down**] buttons or with ‘drag-and-drop’ functionality. The order can not be changed for WMS legend graphics.
 - Use the [**Add group**] button to add a legend group.
 - Use the [**plus**] and [**minus**] button to add or remove layers.
 - The [**Edit**] button is used to edit the layer-, groupname or title, first you need to select the legend item.
 - The [**Sigma**] button adds a feature count for each vector layer.
 - Use the [**filter**] button the filter the legend by map content, only the legend items visible in the map will be listed in the legend.

After changing the symbology in the QGIS main window, you can click on [**Update**] to adapt the changes in the legend element of the Print Composer.

Fonts, Columns, Symbol

The *Fonts*, *Columns* and *Symbol* dialogs of the legend *Item Properties* tab provide the following functionalities (see [figure_composer_legend_4](#)):

- Puoi cambiare il carattere del titolo, gruppo, sotto-gruppo o elemento singolo della legenda. Clicca sul pulsante corrispondente per aprire la finestra **Selezione carattere**.
- You provide the labels with a **Color** using the advanced color picker, however the selected color will be given to all font items in the legen..
- Legend items can be arranged over several columns. Set the number of columns in the *Count* field.
 - La casella di controllo *Uguale larghezza delle colonne* imposta come le colonne della legenda devono essere gestite.

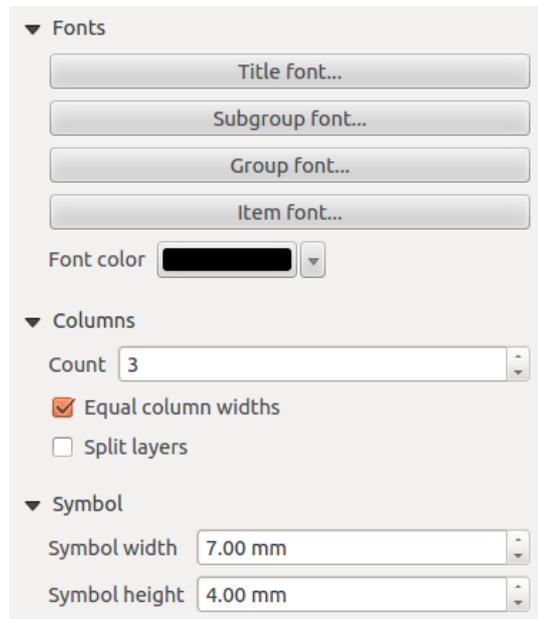


Figura 19.17: Finestra di dialogo caratteri, colonne, simboli e spaziatura legenda 

- La casella di controllo  *Dividi layers* ti permette una divisione dei layer in base alla loro categorizzazione.
- Puoi cambiare la larghezza e l'altezza di un simbolo della legenda.

WMS legendGraphic and Spacing

The *WMS legendGraphic* and *Spacing* dialogs of the legend *Item Properties* tab provide the following functionalities (see [figure_composer_legend_5](#)):

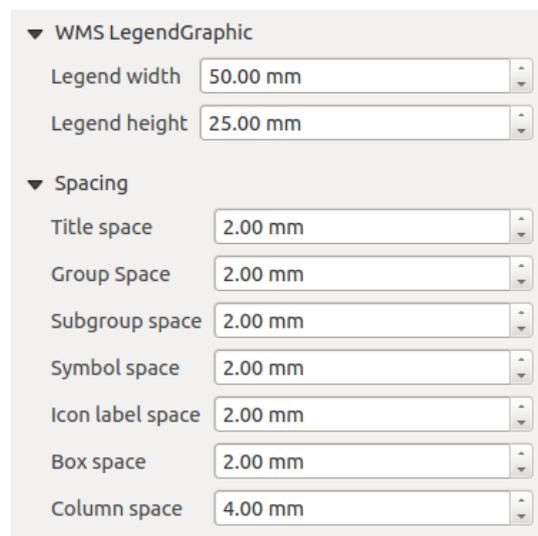


Figura 19.18: WMS legendGraphic Dialogs 

When you have added a WMS layer and you insert a legend composer item, a request will be send to the WMS server to provide a WMS legend, This Legend will only be shown if the WMS server provides the GetLegendGraphic capability. The WMS legend content will be provided as a raster image.

WMS legendGraphic is used to be able to adjust the *Legend width* and the *legend height* of the WMS legend raster image.

Spacing around title, group, subgroup, symbol, icon label, box space or column space can be customized through this dialog.

19.3.5 The Scale Bar item

Per aggiungere una barra di scala, clicca su  *Aggiungi nuova barra di scala*, scegli dove posizionarla e clicca con il tasto sinistro del mouse. Hai pieno accesso alla personalizzazione attraverso la scheda *Proprietà oggetto*.

The *Item properties* of a scale bar item tab provides the following functionalities (see [figure_composer_scalebar_1](#)):

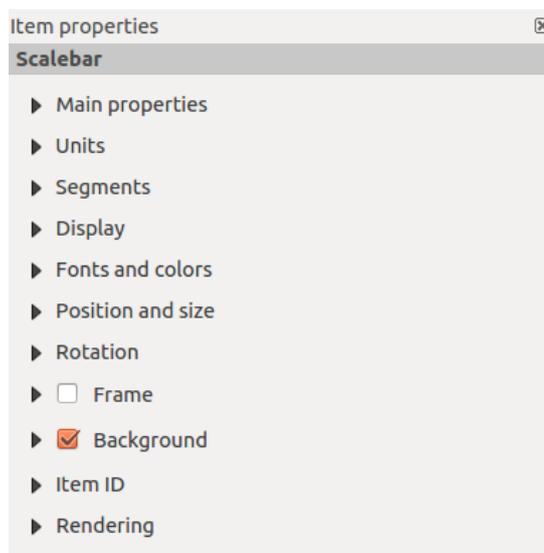


Figura 19.19: Proprietà barra di scala 

Proprietà principali

The *Main properties* dialog of the scale bar *Item Properties* tab provides the following functionalities (see [figure_composer_scalebar_2](#)):

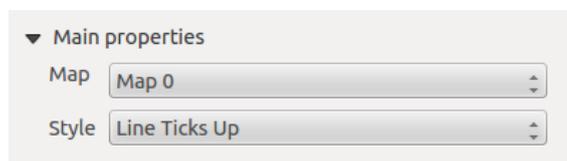


Figura 19.20: Proprietà principali della barra di scala 

- Prima di tutto scegli a quale mappa vuoi aggiungere la barra di scala.
- Poi scegli lo stile della barra di scale. Hai a disposizione sei stili:
 - **Riquadro singolo** e **Riquadro doppio** che contengono una o due linee con colori alternati.
 - **Linea con tacche al centro**, **Linea con tacche in basso** o **Linea con tacche in alto**.
 - **Numerico** che mostra semplicemente il rapporto di scala (per esempio

Unità e segmenti

The *Units* and *Segments* dialogs of the scale bar *Item Properties* tab provide the following functionalities (see [figure_composer_scalebar_3](#)):

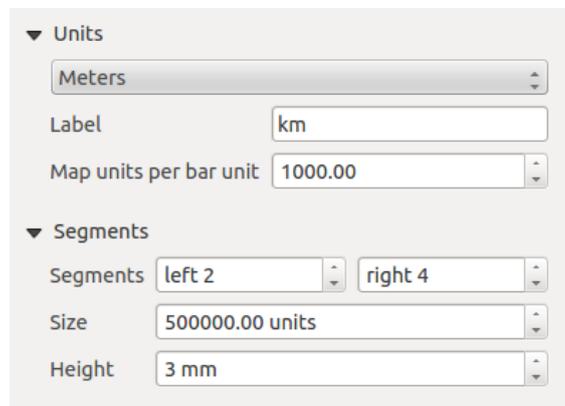


Figura 19.21: Unità e segmenti della barra di scala 🐧

In queste due sezioni puoi scegliere come deve essere rappresentata la barra di scala.

- Select the map units used. There are four possible choices: **Map Units** is the automated unit selection; **Meters**, **Feet** or **Nautical Miles** force unit conversions.
- Specifica nel campo *Etichetta* il testo che descriverà l'unità scelta.
- Il campo *Unità mappa per unità di barra* ti permette di fissare il rapporto fra le unità di mappa e la loro rappresentazione nella barra di scala.
- Puoi scegliere quanti *Segmenti* devono essere disegnati a sinistra e a destra della barra di scala e quanto ogni segmento deve essere lungo (*Dimensione*). Puoi anche scegliere l'*Altezza*.

Display

The *Display* dialog of the scale bar *Item Properties* tab provide the following functionalities (see [figure_composer_scalebar_4](#)):

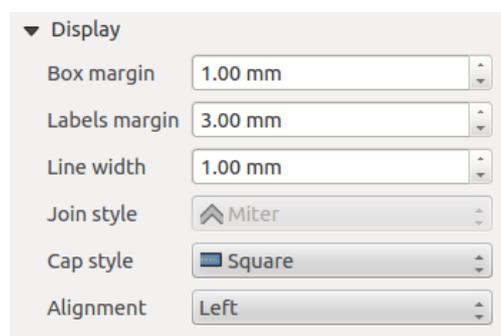


Figura 19.22: Scale Bar Display 🐧

You can define how the scale bar will be displayed in its frame.

- *Box margin* : space between text and frame borders
- *Labels margin* : space between text and scale bar drawing
- *Line width* : line width of the scale bar drawing

- *Join style* : Corners at the end of scalebar in style Bevel, Rounded or Square (only available for Scale bar style Single Box & Double Box)
- *Cap style* : End of all lines in style Square, Round or Flat (only available for Scale bar style Line Ticks Up, Down and Middle)
- *Alignment* : Puts text on the left, middle or right side of the frame (works only for Scale bar style Numeric)

Fonts and colors

The *Fonts and colors* dialog of the scale bar *Item Properties* tab provide the following functionalities (see figure_composer_scalebar_5):

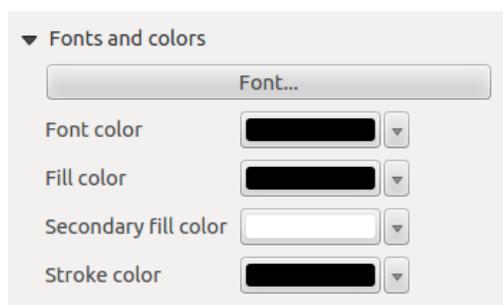


Figura 19.23: Scale Bar Fonts and colors Dialogs 

You can define the fonts and colors used for the scale bar.

- Use the [**F**ont] button to set the font
- *Font color*: set the font color
- *Fill color*: set the first fill color
- *Secondary fill color*: set the second fill color
- *Stroke color*: set the color of the lines of the Scale Bare

Fill colors are only used for scale box styles Single Box and Double Box. To select a color you can use the list option using the dropdown arrow to open a simple color selection option or the more advanced color selection option, that is started when you click in the colored box in the dialog.

19.3.6 The Basic Shape Items

To add a basic shape (ellipse, rectangle, triangle), click the  Add basic shape icon or the  Add Arrow icon, place the element holding down the left mouse. Customize the appearance in the *Item Properties* tab.

When you also hold down the `Shift` key while placing the basic shape you can create a perfect square, circle or triangle.

The *Shape* item properties tab allows you to select if you want to draw an ellipse, rectangle or triangle inside the given frame.

You can set the style of the shape using the advanced symbol style dialog with which you can define its outline and fill color, fill pattern, use markers etcetera.

For the rectangle shape, you can set the value of the corner radius to round of the corners.

Nota: Unlike other items, you can not style the frame or the background color of the frame.

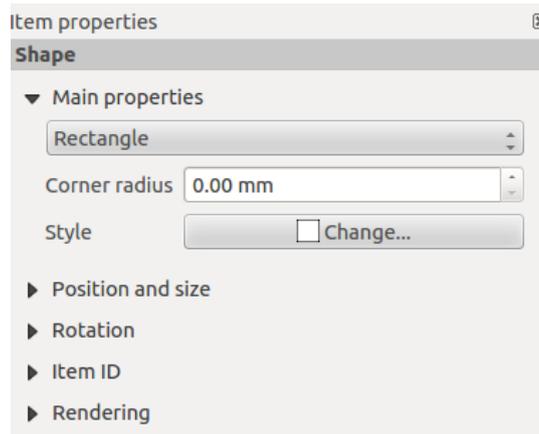


Figura 19.24: Scheda proprietà forma 

19.3.7 The Arrow item

To add an arrow, click the  **Add Arrow** icon, place the element holding down the left mouse button and drag a line to draw the arrow on the Print Composer canvas and position and customize the appearance in the scale bar *Item Properties* tab.

When you also hold down the `Shift` key while placing the arrow, it is placed in an angle of exactly 45° .

The arrow item can be used to add a line or a simple arrow that can be used, for example, to show the relation between other print composer items. To create a north arrow, the image item should be considered first. QGIS has a set of North arrows in SVG format. Furthermore you can connect an image item with a map so it can rotate automatically with the map (see [the_image_item](#)).

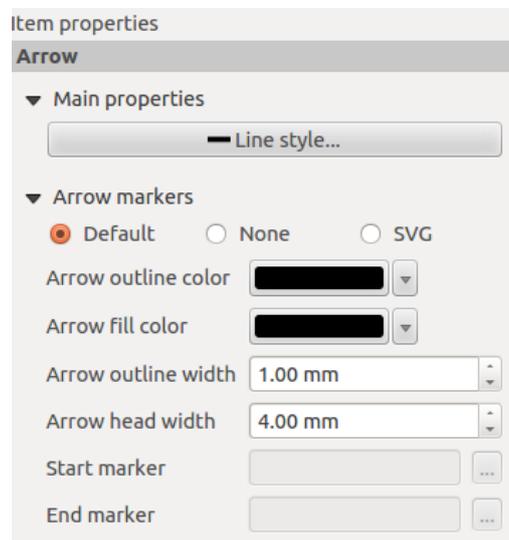


Figura 19.25: Scheda proprietà freccia 

Item Properties

The *Arrow* item properties tab allows you to configure an arrow item.

The [**Line style ...**] button can be used to set the line style using the line style symbol editor.

In *Arrows markers* you can select one of three radio buttons.

- *Default* : To draw a regular arrow, gives you options to style the arrow head
- *None* : To draw a line without arrow head
- *SVG Marker* : To draw a line with an *SVG Start marker* and/or *End marker*

For *Default* Arrow marker you can use following options to style the arrow head.

- *Arrow outline color* : Set the outline color of the arrow head
- *Arrow fill color* : Set the fill color of the arrow head
- *Arrow outline width* : Set the outline width of the arrow head
- *Arrow head width*: Set the size of the arrow head

For *SVG Marker* you can use following options.

- *Start marker* : Choose an SVG image to draw at the beginning of the line
- *End marker* : Choose an SVG image to draw at the end of the line
- *Arrow head width*: Sets the size of Start and/or headmarker

SVG images are automatically rotated with the line. The color of the SVG image can not be changed.

19.3.8 The Attribute Table item

Puoi aggiungere parti della tabella degli attributi di un vettore al compositore di stampe: clicca sull'icona  *Aggiungi tabella attributi*, scegli dove posizionarla e clicca con il tasto sinistro del mouse. In seguito potrai personalizzarne gli aspetti dalla scheda *Proprietà oggetto*.

The *Item properties* of an attribute table item tab provides the following functionalities (see [figure_composer_table_1](#)):

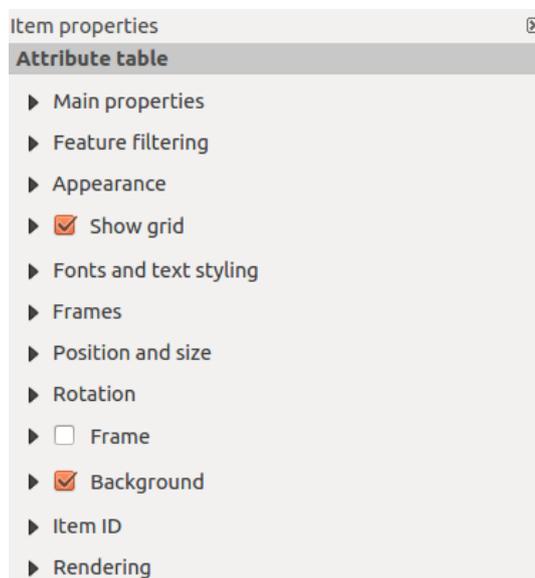


Figura 19.26: Attribute table Item properties Tab 

Proprietà principali

The *Main properties* dialogs of the attribute table *Item Properties* tab provide the following functionalities (see [figure_composer_table_2](#)):

- For *Source* you can normally select only 'Layer features'.

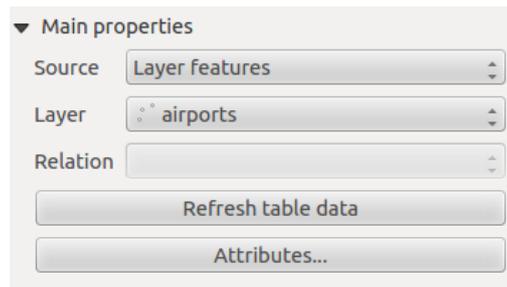


Figura 19.27: Attribute table Main properties Dialog 

- With *Layer* you can choose from the vector layers loaded in the project.
- The button [**Refresh table data**] can be used to refresh the table when the actual contents of the table has changed.
- The button [**Attributes...**] starts the *Select attributes* menu, see [figure_composer_table_3](#), that can be used to change the visible contents of the table. After making changes use the [**OK**] button to apply changes to the table.

In the *Columns* section you can:

- Remove an attribute, just select an attribute row by clicking anywhere in a row and press the minus button to remove the selected attribute.
- Add a new attribute use the plus button. At the end a new empty row appears and you can select empty cell of the column *Attribute*. You can select a field attribute from the list or you can select to build a new attribute using a regular expression.
- Use the up and down arrows to change the order of the attributes in the table.
- Select a cel in the Headings column to change the Heading, just type a new name.
- Select a cel in the Alignment column and you can choose between Left, Center or Right alignment.
- Select a cel in the Width column and you can change it from Automatic to a width in mm, just type a number. When you want to change it back to Automatic, use the cross.
- The [**Reset**] button can allways be used to restore it to the original attribute settings.

In the *Sorting* section you can:

- Add an attribute to sort the table with. Select an attribute and set the sorting order to 'Ascending' or 'Descending' and press the plus button. A new line is added to the sort order list.
- select a row in the list and use the up and down button to change the sort priority on attribute level.
- use the minus button to remove an attribute from the sort order list.

Feature filtering

The *Feature filtering* dialogs of the attribute table *Item Properties* tab provide the following functionalities (see [figure_composer_table_4](#)):

You can:

- Define the *Maximum rows* to be displayed.
- Activate *Remove duplicate rows from table* to show unique records only.
- Activate *Show only visible features within a map* and select the corresponding *Composer map* to display the attributes of features only visible on selected map.

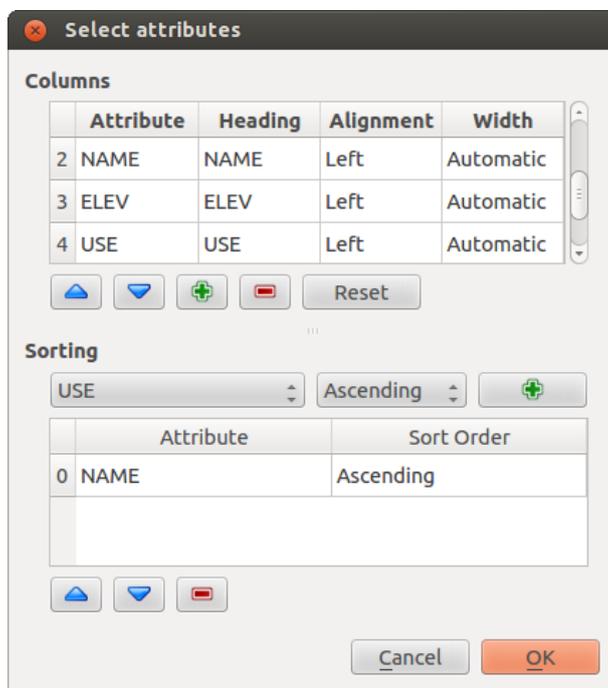


Figura 19.28: Finestra di dialogo scegli attributo 

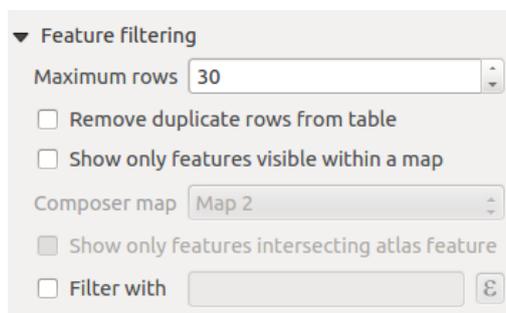


Figura 19.29: Attribute table Feature filtering Dialog 

- Activate *Show only features intersecting Atlas feature* is only available when *Generate an atlas* is activated. When activated it will show a table with only the features shown on the map of that particular page of the atlas.
- Activate *Filter with* and provide a filter by typing in the input line or insert a regular expressing use the given expression button. A few examples of filtering statements you can use when you have loaded the airports layer from the Sample dataset:
 - ELEV > 500
 - NAME = ' ANIAK'
 - NAME NOT LIKE 'AN%
 - regexp_match(attribute(\$currentfeature, 'USE') , '[i]')

The last regular expression will include only the arpoirts that have a letter 'i' in the attribute field 'USE'.

Appearance

The *Appearance* dialogs of the attribute table *Item Properties* tab provide the following functionalities (see figure_composer_table_5):

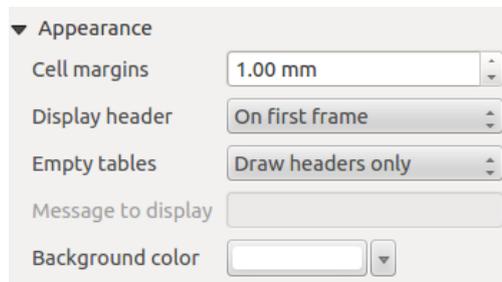


Figura 19.30: Attribute table appearance Dialog 

- With *Cell margins* you can define the margin around text in each cell of the table.
- With *Display header* you can select from a list one of 'On first frame', 'On all frames' default option, or 'No header'.
- The option *Empty table* controls what will be displayed when the result selection is empty.
 - **Draw headers only**, will only draw the header except if you have chosen 'No header' for *Display header*.
 - **Hide entire table**, will only draw the background of the table. You can activate *Don't draw background if frame is empty* in *Frames* to completely hide the table.
 - **Draw empty cells**, will fill the attribute table with empty cells, this option can also be used to provide additional empty cells when you have a result to show!
 - **Show set message**, will draw the header and adds a cell spanning all columns and display a message like 'No result' that can be provided in the option *Message to display*
- The option *Message to display* is only activated when you have selected **Show set message** for *Empty table*. The message provided will be shown in the table in the first row, when the result is an empty table.
- With *Background color* you can set the background color of the table.

Show grid

The *Show grid* dialog of the attribute table *Item Properties* tab provide the following functionalities (see figure_composer_table_6):

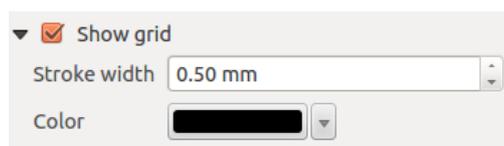


Figura 19.31: Attribute table Show grid Dialog 

- Activate *Show grid* when you want to display the grid, the outlines of the table cells.
- With *Stroke width* you can set the thickness of the lines used in the grid.
- The *Color* of the grid can be set using the color selection dialog.

Fonts and text styling

The *Fonts and text styling* dialog of the attribute table *Item Properties* tab provide the following functionalities (see [figure_composer_table_7](#)):

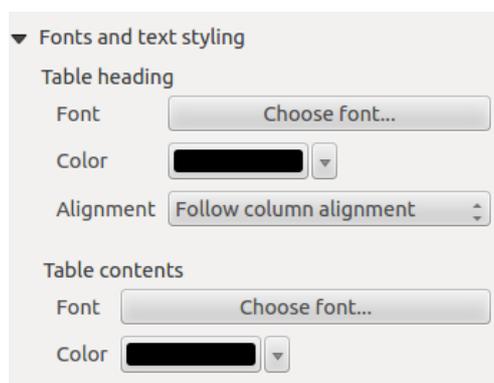


Figura 19.32: Attribute table Fonts and text styling Dialog 

- You can define *Font* and *Color* for *Table heading* and *Table contents*.
- For *Table heading* you can additionally set the *Alignment* and choose from *Follow column alignment*, *Left*, *Center* or *Right*. The column alignment is set using the *Select Attributes* dialog (see [Figure_composer_table_3](#)).

Frames

The *Frames* dialog of the attribute table *Item Properties* tab provide the following functionalities (see [figure_composer_table_8](#)):

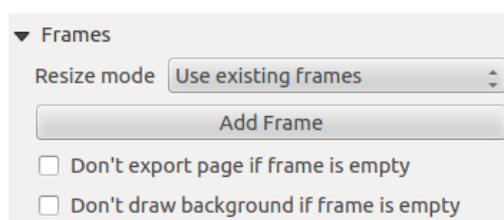


Figura 19.33: Attribute table Frames Dialog 

- With *Resize mode* you can select how to render the attribute table contents:
 - *Use existing frames* displays the result in the first frame and added frames only.
 - *Extent to next page* will create as many frames (and corresponding pages) as necessary to display the full selection of attribute table. Each frame can be moved around on the layout. If you resize a frame, the resulting table will be divided up between the other frames. The last frame will be trimmed to fit the table.
 - *Repeat until finished* will also create as many frames as the *Extend to next page* option, except all frames will have the same size.
- Use the **[Add Frame]** button to add another frame with the same size as selected frame. The result of the table that will not fit in the first frame will continue in the next frame when you use the *Resize mode Use existing frames*.
- Activate *Don't export page if frame is empty* prevents the page to be exported when the table frame has no contents. This means all other composer items, maps, scalebars, legends etc. will not be visible in the result.
- Activate *Don't draw background if frame is empty* prevents the background to be drawn when the table frame has no contents.

19.3.9 The HTML frame item

It is possible to add a frame that displays the contents of a website or even create and style your own HTML page and display it!

Click the  **Add HTML frame** icon, place the element by dragging a rectangle holding down the left mouse button on the Print Composer canvas and position and customize the appearance in the *Item Properties* tab (see figure_composer_html_1).

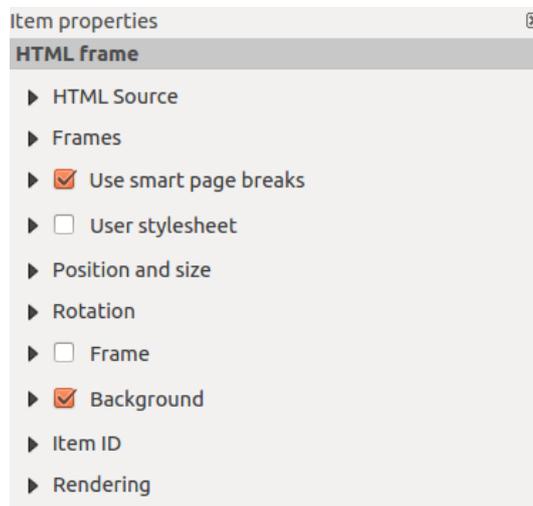


Figura 19.34: HTML frame, the item properties Tab 

HTML Source

As an HTML source, you can either set a URL and activate the URL radiobutton or enter the HTML source directly in the textbox provided and activate the Source radiobutton.

The *HTML Source* dialog of the HTML frame *Item Properties* tab provides the following functionalities (see figure_composer_html_2):

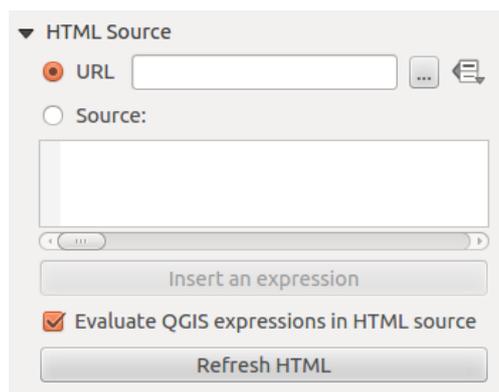


Figura 19.35: HTML frame, the HTML Source properties 🐧

- In *URL* you can enter the URL of a webpage you copied from your internet browser or select an HTML file using the browse button . There is also the option to use the Data defined override button, to provide an URL from the contents of an attribute field of a table or using a regular expression.
- In *Source* you can enter text in the textbox with some HTML tags or provide a full HTML page.
- The **[insert an expression]** button can be used to insert an expression like [%Year(\$now)%] in the Source textbox to display the current year. This button is only activated when radiobutton *Source* is selected. After inserting the expression click somewhere in the textbox before refreshing the HTML frame, otherwise you will lose the expression.
- Activate *Evaluate QGIS expressions in HTML code* to see the result of the expression you have included, otherwise you will see the expression instead.
- Use the **[Refresh HTML]** button to refresh the HTML frame(s) to see the result of changes.

Frames

The *Frames* dialog of the HTML frame *Item Properties* tab provides the following functionalities (see [figure_composer_html_3](#)):

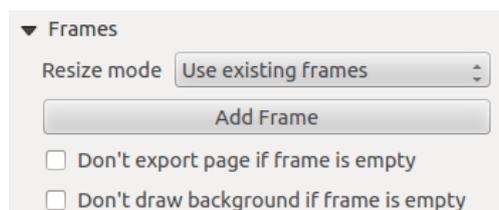


Figura 19.36: HTML frame, the Frames properties 🐧

- With *Resize mode* you can select how to render the HTML contents:
 - *Use existing frames* displays the result in the first frame and added frames only.
 - *Extent to next page* will create as many frames (and corresponding pages) as necessary to render the height of the web page. Each frame can be moved around on the layout. If you resize a frame, the webpage will be divided up between the other frames. The last frame will be trimmed to fit the web page.
 - *Repeat on every page* will repeat the upper left of the web page on every page in frames of the same size.
 - *Repeat until finished* will also create as many frames as the *Extend to next page* option, except all frames will have the same size.

- Use the **[Add Frame]** button to add another frame with the same size as selected frame. If the HTML page that will not fit in the first frame it will continue in the next frame when you use *Resize mode* or *Use existing frames*.
- Activate *Don't export page if frame is empty* prevents the map layout from being exported when the frame has no HTML contents. This means all other composer items, maps, scalebars, legends etc. will not be visible in the result.
- Activate *Don't draw background if frame is empty* prevents the HTML frame being drawn if the frame is empty.

Use smart page breaks and User style sheet

The *Use smart page breaks* dialog and *Use style sheet* dialog of the HTML frame *Item Properties* tab provides the following functionalities (see [figure_composer_html_4](#)):

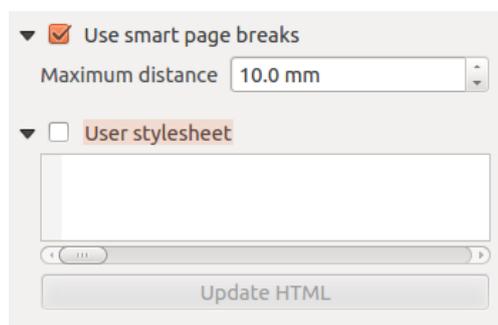


Figura 19.37: HTML frame, Use smart page breaks and User stylesheet properties 

- Activate *Use smart page breaks* to prevent the html frame contents from breaking mid-way a line of text so it continues nice and smooth in the next frame.
- Set the *Maximum distance* allowed when calculating where to place page breaks in the html. This distance is the maximum amount of empty space allowed at the bottom of a frame after calculating the optimum break location. Setting a larger value will result in better choice of page break location, but more wasted space at the bottom of frames. This is only used when *Use smart page breaks* is activated.
- Activate *User stylesheet* to apply HTML styles that often is provided in cascading style sheets. An example of style code is provide below to set the color of `<h1>` header tag to green and set the font and fontsize of text included in paragraph tags `<p>`.

```
h1 {color: #00ff00;
}
p {font-family: "Times New Roman", Times, serif;
font-size: 20px;
}
```

- Use the **[Update HTML]** button to see the result of the stylesheet settings.

19.4 Gestisci elementi

19.4.1 Dimensione e posizione

Puoi spostare/ridimensionare ogni elemento del compositore per creare un layout di stampa perfetto. Per entrambe le operazioni devi prima attivare cliccare sull'icona  *Scegli/Sposta oggetto* e poi cliccare sull'oggetto corrispondente; ora puoi muoverlo con il pulsante sinistro del mouse. Se vuoi forzare lo spostamento lungo un asse

orizzontale o verticale, tieni premuto il tasto `Shift`. Se hai bisogno di una maggiore precisione usa le `Frecce direzionali` della tastiera; se il movimento è troppo lento lo puoi velocizzare tenendo premuto il tasto `Shift`.

A selected item will show squares on its boundaries; moving one of them with the mouse, will resize the item in the corresponding direction. While resizing, holding `Shift` will maintain the aspect ratio. Holding `Alt` will resize from the item center.

The correct position for an item can be obtained using snapping to grid or smart guides. Guides are set by clicking and dragging in the rulers. Guide are moved by clicking in the ruler, level with the guide and dragging to a new place. To delete a guide move it off the canvas. If you need to disable the snap on the fly just hold `Ctrl` while moving the mouse.

Puoi usare lo strumento  `Sposta/Muovi oggetto` su più oggetti contemporaneamente. Tieni premuto il tasto `Shift` e clicca su tutti gli oggetti che vuoi selezionare. Ora li puoi spostare o ridimensionare tutti in un colpo solo.

Once you have found the correct position for an item, you can lock it by using the items on the toolbar or ticking the box next to the item in the *Items* tab. Locked items are **not** selectable on the canvas.

Locked items can be unlocked by selecting the item in the *Items* tab and unchecking the tickbox or you can use the icons on the toolbar.

Per deselezionare un oggetto, cliccaci sopra tenendo premuto il tasto `Shift`.

Nel menu *Modifica*, puoi trovare diversi strumenti che ti permettono di selezionare tutti gli elementi, di pulire o invertire la selezione attuale.

19.4.2 Allineamento

Raising or lowering functionalities for elements are inside the  `Raise selected items` pull-down menu. Choose an element on the Print Composer canvas and select the matching functionality to raise or lower the selected element compared to the other elements (see [table_composer_1](#)). This order is shown in the *Items* tab. You can also raise or lower objects in the *Items* tab by clicking and dragging an object's label in this list.

Ci sono diverse funzionalità disponibili nel menu a tendina  `Allinea gli oggetti selezionati` (vedi [table_composer_1](#)). Per usare una di queste funzionalità, seleziona prima alcuni oggetti e poi clicca sull'icona. In questo modo tutti gli elementi verranno allineati all'interno della loro cornice. Quando muovi gli elementi nel compositore, appaiono delle linee ai margini del foglio che ti aiutano ad allineare gli oggetti.

19.4.3 Copia/Taglia e Incolla oggetti

Il compositore di stampe include tutte le azioni più comuni di copia/taglia/incolla per gli oggetti del layout. Prima di tutto seleziona l'oggetto: nel menu *Modifica* potrai trovare le azioni che potrai effettuare. Mentre usi l'azione incolla, gli oggetti verranno incollati nella posizione attuale del mouse.

Nota: HTML items can not be copied in this way. As a workaround, use the **[Add Frame]** button in the *Item Properties* tab.

19.5 Strumenti Annulla e Ripristina

Durante la creazione di un layout di stampa puoi annullare e ripristinare le azioni. Lo puoi fare semplicemente usando lo strumento annulla/ripristina:

-  Annulla l'ultimo cambiamento
-  Ripristina l'ultimo cambiamento

This can also be done by mouse click within the *Command history* tab (see [figure_composer_29](#)).

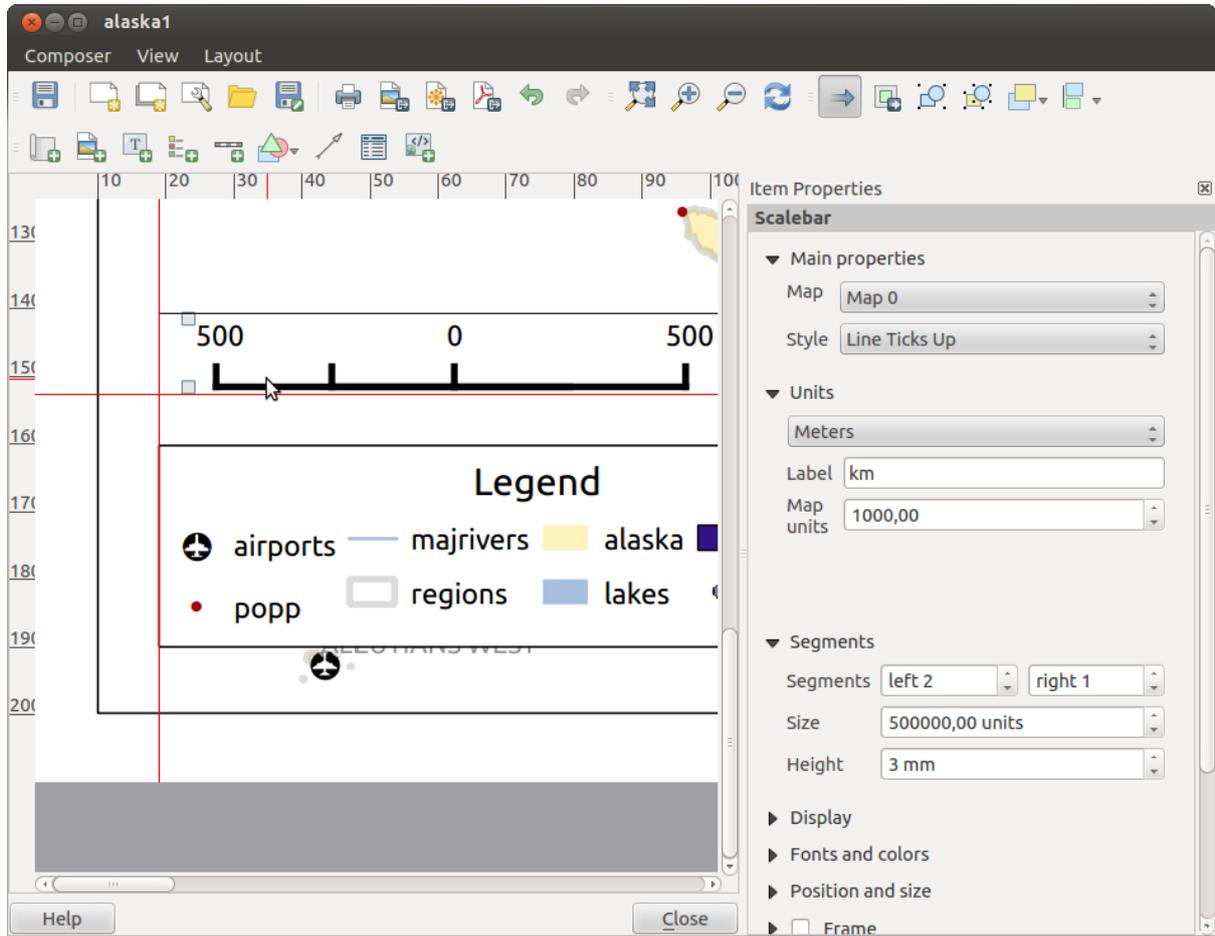


Figura 19.38: Linee guida di allineamento del compositore di stampe 🐧



Figura 19.39: Storico dei comandi del compositore di stampe 🐧

19.6 Generazione atlante

Il compositore di stampe include anche una funzionalità che ti permette di creare automaticamente degli atlanti. Il concetto è quello di usare un vettore di copertura che contiene campi e geometrie. Per ogni geometria di questo vettore potrai creare un nuovo output: ogni nuovo output sarà centrato sulla geometria corrispondente. Puoi usare i campi delle geometrie come etichette per gli output.

Every page will be generated with each feature. To enable the generation of an atlas and access generation parameters, refer to the *Atlas generation* tab. This tab contains the following widgets (see [Figure_composer_atlas](#)):

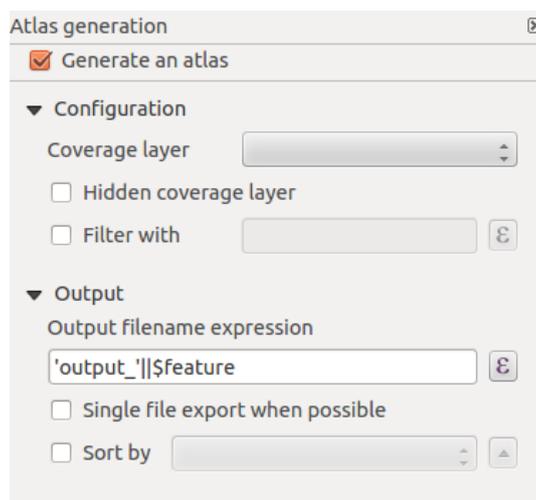


Figura 19.40: Scheda generazione atlante 

- *Genera un atlante*, che attiva o disattiva la generazione dell'atlante.
- Un menu a tendina *Layer di copertura*  che ti permette di scegliere il layer (vettore) contenente le geometrie che verranno iterate.
- Una casella di controllo opzionale *Layer copertura nascosto* che, se attivata, nasconderà solamente il layer di copertura durante la generazione dell'atlante.
- Una casella opzionale *Filtra con* che ti permette di specificare un'espressione per filtrare le geometrie del layer di copertura. Se l'espressione non è vuota, solamente le geometrie valutate come `True` verranno selezionate. Premendo il pulsante sulla destra puoi aprire il costruttore di espressioni.
- Usa la casella di testo *Espressione nome file di output* per generare un nome per ogni geometria. Si basa su un'espressione. Questo campo ha senso solamente per la creazione di file multipli.
- Una casella di controllo *Esporta file singolo se possibile* che ti permette di forzare la generazione di un singolo file se è compatibile con il formato di output scelto (per esempio, PDF). Se spuntata, il valore inserito in *Espressione del nome di file in output* non ha senso.
- Una casella di controllo opzionale *Ordina per* che ti permette di ordinare le geometrie del layer di copertura. Il menu a tendina associato ti permette di scegliere quale colonna dovrà essere usate per l'ordinamento. Puoi impostare l'ordine (crescente o decrescente) grazie al pulsante a destra del menu a tendina.

Puoi usare oggetti multipli con la generazione dell'atlante: ogni mappa verrà visualizzata secondo le geometrie del layer di copertura. Per impostare la generazione dell'atlante per un oggetto specifico, spunta la casella di controllo

Controllato dall'atlante nella scheda delle proprietà dell'oggetto. Una volta spuntata potrai impostare:

- Un *Margine attorno all'elemento* che ti permette di selezionare lo spazio attorno ad ogni geometria nella mappa corrispondente. Questo valore ha senso solo usando la modalità scala automatica.

- Una casella di controllo  *Scala fissa* che ti permette di passare dalla modalità a scala automatica a quella di scala fissa. In quest'ultima modalità, la mappa verrà generata solamente per le geometrie che possono essere centrate. Nella modalità scala automatica, le estensioni della mappa sono calcolate in modo che ogni geometria appaia in tutta la sua interezza.

19.6.1 Etichette

In order to adapt labels to the feature the atlas plugin iterates over, you can include expressions. For example, for a city layer with fields CITY_NAME and ZIPCODE, you could insert this:

```
The area of [% upper(CITY_NAME) || ', ' || ZIPCODE || ' is ' format_number($area/1000000,2) %] km2
```

The information [% upper(CITY_NAME) || ', ' || ZIPCODE || ' is ' format_number(\$area/1000000,2) %] is an expression used inside the label. That would result in the generated atlas as:

The area of PARIS,75001 is 1.94 km2

19.6.2 Data Defined Override Buttons

There are several places where you can use a  Data Defined Override button to override the selected setting. These options are particularly usefull with Atlas Generation.

For the following examples the *Regions* layer of the QGIS sample dataset is used and selected for Atlas Generation. We also assume the paper format *A4 (210X297)* is selected in the *Composite* tab for field *Presets*.

With a *Data Defined Override* button you can dynamically set the paper orientation. When the height (north-south) of the extents of a region is greater than it's width (east-west), you rather want to use *portrait* instead of *landscape* orientation to optimize the use of paper.

In the *Composition* you can set the field *Orientation* and select *Landscape* or *Portrait*. We want to set the orientation dynamically using an expression depending on the region geometry. press the  button of field *Orientation*, select *Edit ...* so the *Expression string builder* dialog opens. Give following expression:

```
CASE WHEN bounds_width($atlasgeometry) > bounds_height($atlasgeometry) THEN 'Landscape' ELSE 'Portrait'
```

Now the paper orients itself automatically for each Region you need to reposition the location of the composer item as well. For the map item you can use the  button of field *Width* to set it dynamically using following expression:

```
(CASE WHEN bounds_width($atlasgeometry) > bounds_height($atlasgeometry) THEN 297 ELSE 210 END) - 10
```

Use the  button of field *Height* to provide following expression:

```
(CASE WHEN bounds_width($atlasgeometry) > bounds_height($atlasgeometry) THEN 210 ELSE 297 END) - 10
```

When you want to give a title above map in the center of the page, insert a label item above the map. First use the item properties of the label item to set the horizontal alignment to  *Center*. Next activate from *Reference point* the upper middle checkbox. You can provide following expression for field *X* :

```
(CASE WHEN bounds_width($atlasgeometry) > bounds_height($atlasgeometry) THEN 297 ELSE 210 END) / 2
```

For all other composer items you can set the position in a similar way so they are correctly positioned when page is automatically rotated in portrait or landscape.

Information provided is derived from the excellent blog (in english and portugese) on the Data Defined Override options [Multiple_format_map_series_using_QGIS_2.6](#) .

This is just one example of how you can use Data Defined Overrides.

19.6.3 Anteprima

Una volta che hai configurato le impostazioni dell'atlante e che hai selezionato gli oggetti della mappa, puoi creare un'anteprima di tutte le pagine cliccando sul pulsante *Atlante* → *Anteprima atlante* e usare le frecce dello stesso menu per navigare fra le varie geometrie.

19.6.4 Generazione

Puoi effettuare la generazione dell'atlante in modi diversi. Per esempio tramite il menu *Atlante* → *Stampa atlante*. Puoi anche creare un PDF cliccando su *Atlante* → *Esporta atlante come PDF*: ti verrà chiesto un percorso in cui salvare i PDF (eccetto se hai spuntato la casella di controllo *Esporta file singolo se possibile*). Se devi stampare solamente una pagina dell'atlante, fai partire l'anteprima, fermati alla pagina che vuoi e clicca su *Compositore* → *Stampa* (o esporta come PDF).

19.7 Creazione del file in output

[Figure_composer_output](#) shows the Print Composer with an example print layout, including each type of map item described in the sections above.

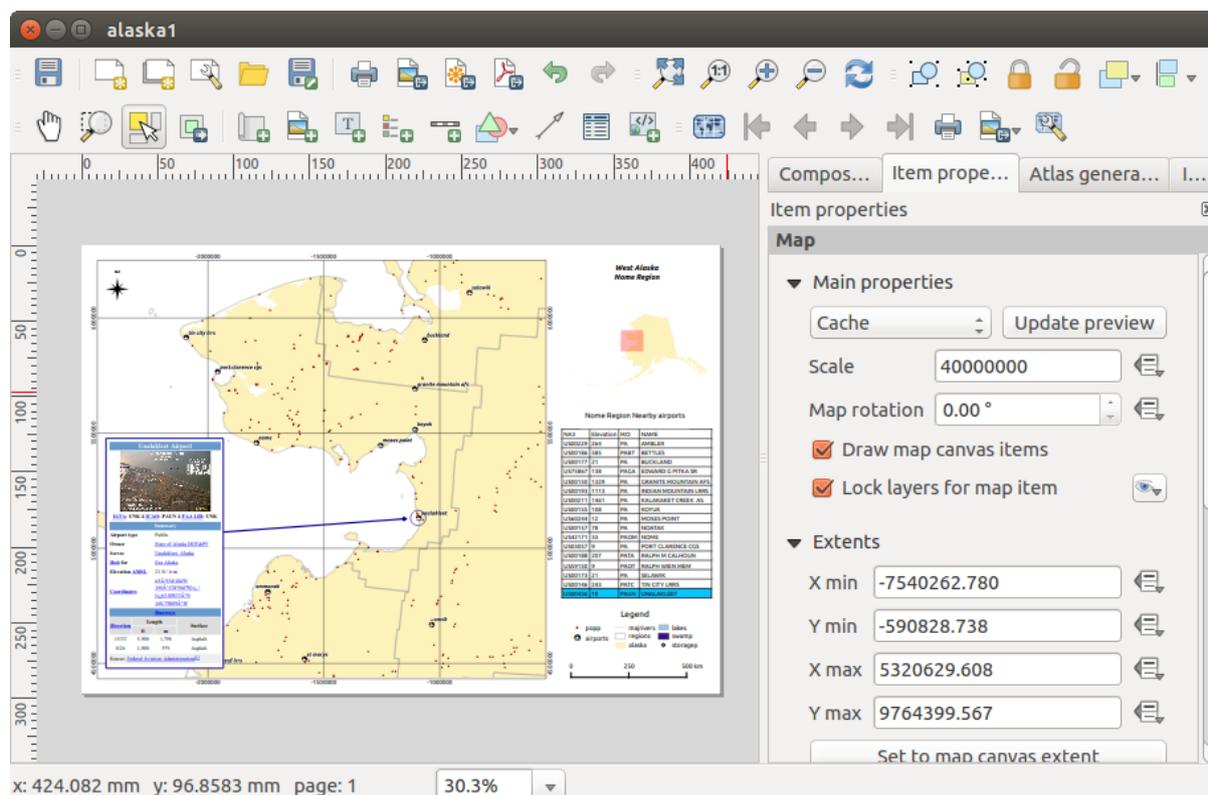


Figura 19.41: Compositore con mappa, legenda, immagine, barra di scala, coordinate, testo e cornice HTML 

Il compositore di stampe ti permette di creare diversi formati di output e puoi scegliere la risoluzione (qualità di stampa) e il formato pagina:

- L'icona  *Stampa* ti permette di stampare il layout su una stampante collegata o su un file PDF o Postscript.
- L'icona  *Esporta come immagine* esporta il layout in diversi formati immagine come PNG, BPM, TIF, JPG,...
- L'icona  *Esporta come PDF* esporta il layout in formato PDF.

- L'icona  **Esporta come SVG** salva il layout di stampa in formato SVG (Scalable Vector Graphic).

If you need to export your layout as a **georeferenced image** (i.e., to load back inside QGIS), you need to enable this feature under the Composition tab. Check  **World file on** and choose the map item to use. With this option, the 'Export as image' action will also create a world file.

Nota:

- Currently, the SVG output is very basic. This is not a QGIS problem, but a problem with the underlying Qt library. This will hopefully be sorted out in future versions.
- Exporting big rasters can sometimes fail, even if there seems to be enough memory. This is also a problem with the underlying Qt management of rasters.

19.8 Gestisci le composizioni di stampa

With the  **Save as template** and  **Add items from template** icons, you can save the current state of a Print Composer session as a `.qpt` template and load the template again in another session.

Il pulsante  **Gestore composizioni** nella barra di QGIS nel menu *Compositore* → *Gestore composizioni* ti permette di aggiungere un nuovo modello di stampa basato su un modello salvato o di gestire modelli già esistenti.

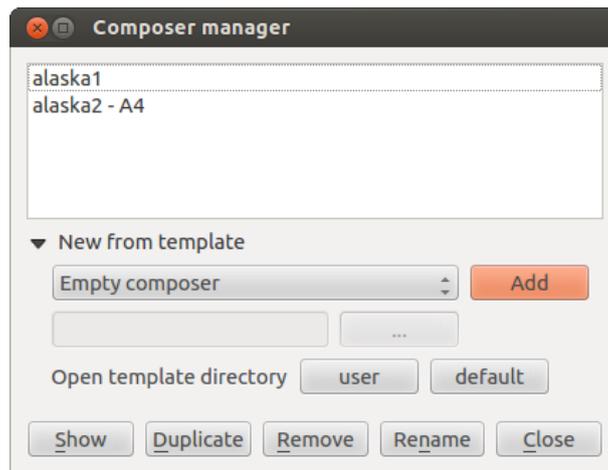


Figura 19.42: Gestore di stampe 

In modo predefinito, il gestore composizioni cerca i modelli nella cartella

I pulsanti  **Nuova composizione** e  **Duplica composizione** nella barra degli strumenti di QGIS e nel menu *Compositore* → *Nuova compositore* e *Compositore* → *Duplica compositore* ti permettono di aprire una nuova finestra di dialogo in cui creare un nuovo layout oppure di duplicare un layout esistente.

Infine, puoi salvare le tue composizioni con il pulsante  **Salva progetto**. È lo stesso pulsante presente nella finestra principale di QGIS. Tutti i cambiamenti e le modifiche verranno quindi salvate su un file di progetto di QGIS.

Plugin di QGIS

20.1 Plugin di QGIS

QGIS è stato progettato con un'architettura plugin. Ciò permette l'aggiunta di numerosi nuovi elementi e funzioni. Molte delle funzioni di QGIS sono attualmente implementate come plugin.

Puoi gestire i tuoi Plugin nella finestra di dialogo che apri con *Plugins > Gestisci e installa plugins....*

When a plugin needs to be updated, and if plugins settings have been set up accordingly, QGIS main interface could display a blue link in the status bar to tell you that there are some plugins updating waiting to be applied.

20.1.1 La finestra di dialogo Plugins

The menus in the Plugins dialog allow the user to install, uninstall and upgrade plugins in different ways. Each plugin have some metadatas displayed in the right panel:

- information if the plugin is experimental
- description
- rating vote(s) (you can vote for your preferred plugin!)
- tags
- some useful links as the home page, tracker and code repository
- author(s)
- version available

You can use the filter to find a specific plugin.



Here, all the available plugins are listed, including both core and external plugins. Use **[Upgrade all]** to look for new versions of the plugins. Furthermore, you can use **[Install plugin]**, if a plugin is listed but not installed, and **[Uninstall plugin]** as well as **[Reinstall plugin]**, if a plugin is installed. If a plugin is installed, it can be de/activated using the checkbox.



In this menu, you can find only the installed plugins. The external plugins can be uninstalled and reinstalled using the **[Uninstall plugin]** and **[Reinstall plugin]** buttons. You can **[Upgrade all]** here as well.



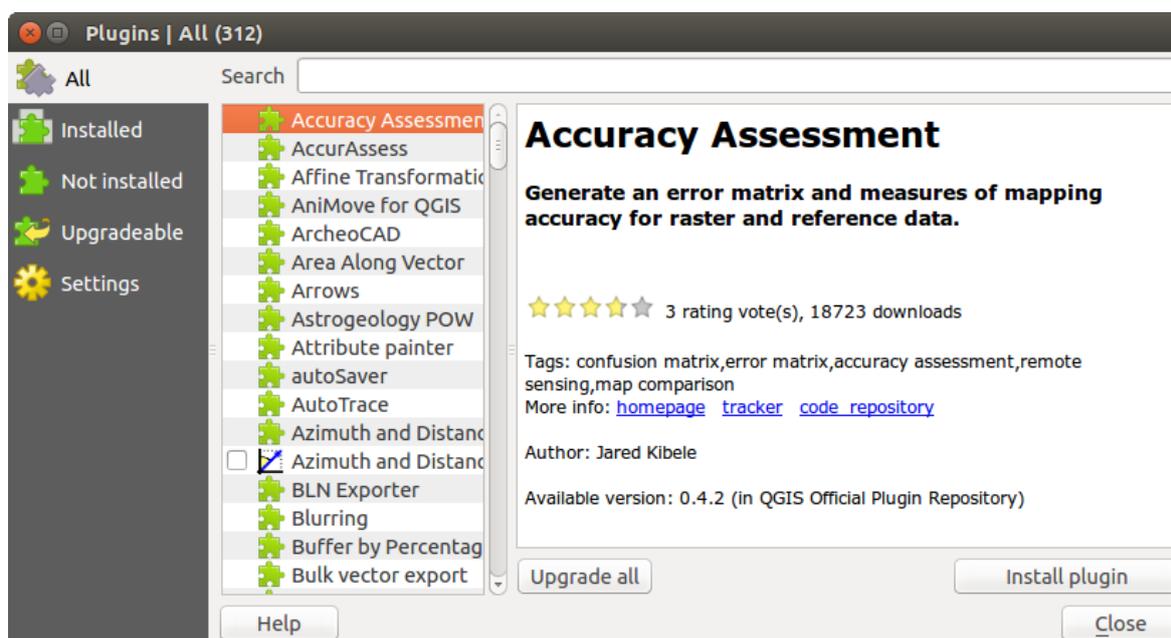


Figura 20.1: The  All menu 

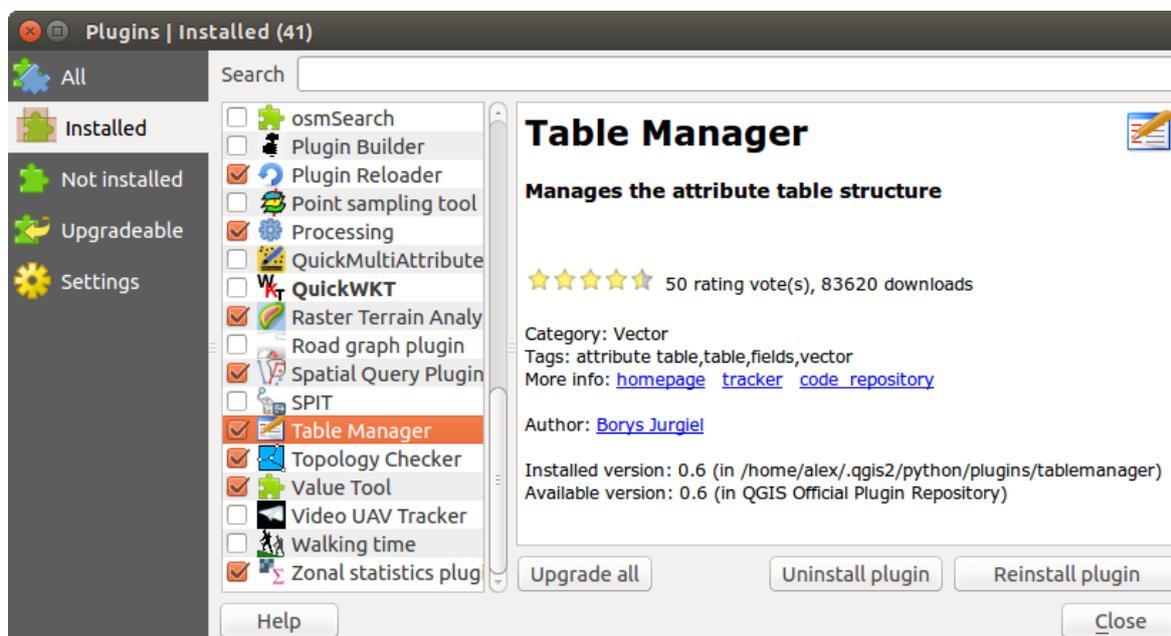


Figura 20.2: The  Installed menu 

This menu lists all plugins available that are not installed. You can use the **[Install plugin]** button to implement a plugin into QGIS.



Figura 20.3: The  *Not installed* menu 

Upgradeable

If you activated *Show also experimental plugins* in the  *Settings* menu, you can use this menu to look for more recent plugin versions. This can be done with the **[Upgrade plugin]** or **[Upgrade all]** buttons.

Settings

In this menu, you can use the following options:

- *Check for updates on startup*. Whenever a new plugin or a plugin update is available, QGIS will inform you ‘every time QGIS starts’, ‘once a day’, ‘every 3 days’, ‘every week’, ‘every 2 weeks’ or ‘every month’.
- *Show also experimental plugins*. QGIS will show you plugins in early stages of development, which are generally unsuitable for production use.
- *Show also deprecated plugins*. These plugins are deprecated and generally unsuitable for production use.

To add external author repositories, click **[Add...]** in the *Plugin repositories* section. If you do not want one or more of the added repositories, they can be disabled via the **[Edit...]** button, or completely removed with the **[Delete]** button.

The *Search* function is available in nearly every menu (except  *Settings*). Here, you can look for specific plugins.

Suggerimento: Core and external plugins

QGIS plugins are implemented either as **Core Plugins** or **External Plugins**. **Core Plugins** are maintained by the QGIS Development Team and are automatically part of every QGIS distribution. They are written in one of two languages: C++ or Python. **External Plugins** are currently all written in Python. They are stored in external repositories and are maintained by the individual authors.

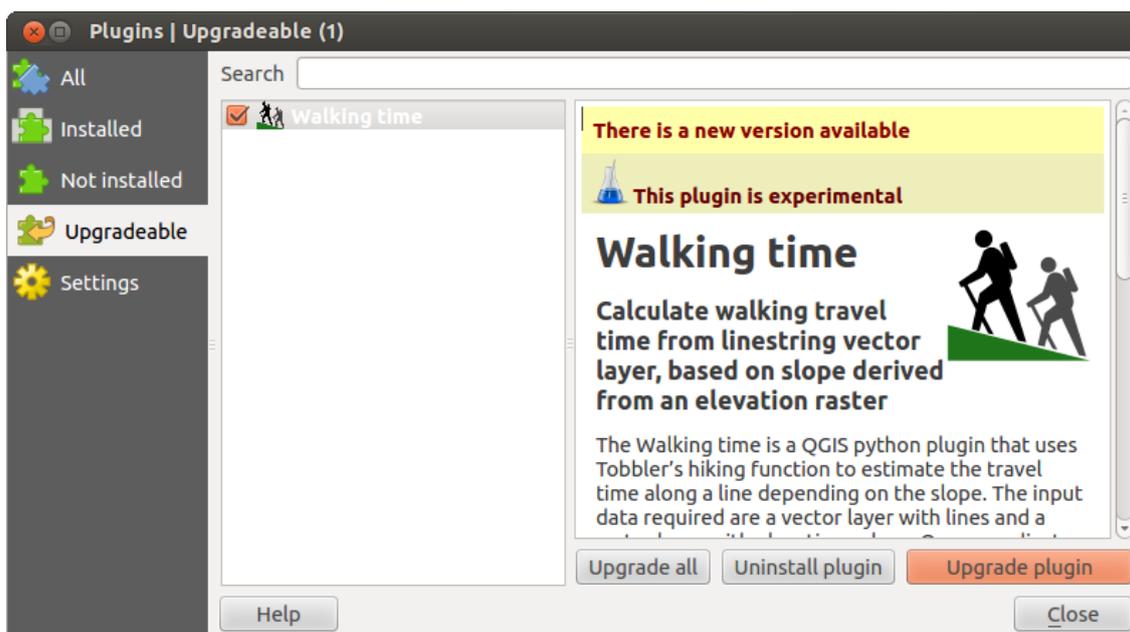


Figura 20.4: The  Upgradeable menu 

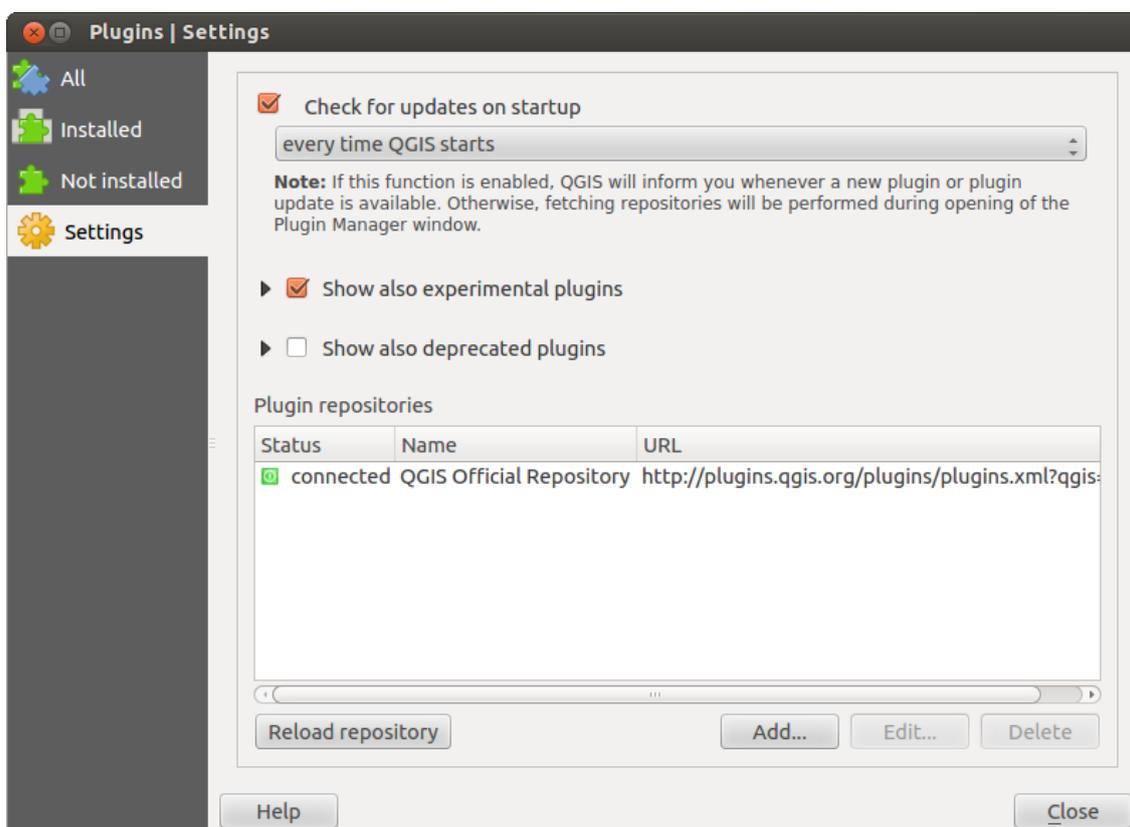


Figura 20.5: The  Settings menu 

Detailed documentation about the usage, minimum QGIS version, home page, authors, and other important information are provided for the 'Official' QGIS Repository at <http://plugins.qgis.org/plugins/>. For other external repositories, documentation might be available with the external plugins themselves. In general, it is not included in this manual.

20.2 Uso dei plugin di base di QGIS

Icona	Plugin	Descrizione	Riferimento guida
	Cattura Coordinate	Cattura le coordinate del mouse usando un SR diverso	<i>Plugin Cattura coordinate</i>
	DB Manager	Gestisce i database all'interno di QGIS	<i>Plugin DB Manager</i>
	Convertitore DXF2Shape	Converte da dxf a shp	<i>Plugin Convertitore DXF2Shape</i>
	eVis	Uno strumento di visualizzazione di eventi. Visualizza immagini associate agli elementi di un vettore	<i>Plugin eVis</i>
	fTools	Strumenti per l'analisi e la gestione di dati vettoriali	<i>Plugin fTools</i>
	Strumenti GPS	Strumenti per caricare e importare dati GPS	<i>Plugin GPS</i>
	GRASS	Attiva i potenti strumenti di GRASS	<i>Integrazione con GRASS GIS</i>
	Strumenti GDAL	Strumenti raster: interfaccia grafica semplificata per l'utilizzo dei programmi GDAL più comuni	<i>Plugin GDALTools</i>
	Georeferenziatore raster (GDAL)	Georeferenziare i raster con GDAL	<i>Plugin Georeferenziatore</i>
	Mappa di concentrazione	Crea una mappa raster di concentrazione a partire da un vettore di punti	<i>Plugin Mappa di concentrazione</i>
	Plugin di interpolazione	Un plugin per l'interpolazione basata sui vertici di un vettore	<i>Plugin Interpolazione</i>
	Offline Editing	Consente l'editing offline e la sincronizzazione con il database	<i>Plugin Offline Editing</i>
	Oracle Spatial Georaster	Accede a Oracle Spatial GeoRasters	<i>Oracle Spatial GeoRaster Plugin</i>
	Gestore plugin	Gestisci i plugin di base e quelli esterni	<i>La finestra di dialogo Plugins</i>
	Plugin per l'analisi geomorfologica	Un plugin per l'analisi geomorfologica basata su raster	<i>Plugin Analisi geomorfologica</i>
	Plugin grafo stradale	Trova il percorso più breve	<i>Plugin grafo strade</i>
	Plugin SQL Anywhere	Salva vettori in un database SQL anywhere	<i>Plugin SQL Anywhere</i>
	Plugin di interrogazione spaziale	Un plugin per effettuare interrogazioni spaziali su dati vettoriali	<i>Plugin Spatial Query</i>
	SPIT	Strumento per importare shapefile in PostGIS	<i>Plugin SPIT</i>
	Statistiche zonali	Calcola statistiche raster per ogni poligono di un vettore	<i>Plugin Statistica zonale</i>
	Metamotore	Interagisce con Catalog Service for the Web (CSW)	<i>Client Catalogo MetaSearch</i>

20.3 Plugin Cattura coordinate

Il plugin Cattura Coordinate è facile da usare e ti permette di mostrare sulla mappa coordinate in due sistemi di riferimento distinti.

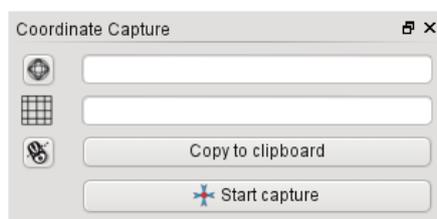


Figura 20.6: Plugin Cattura coordinate 

1. Avvia QGIS, apri proprietà del progetto  *Proprietà progetto* nel menu *Impostazioni* (KDE, Windows) o *File* (Gnome, OSX) e scegli la scheda *Sistema di riferimento (SR)*. In alternativa, clicca sull'icona  Stato SR nell'angolo in basso a destra della barra di stato.
2. Attivare *Abilita la riproiezione al volo* e selezionare un sistema di coordinate proiettate a scelta (sezione *Lavorare con le proiezioni*).
3. Attiva il plugin cattura coordinate nel Gestore Plugin (vedi :ref: *finestra di dialogo plugins*) e assicurarti che la finestra è visibile all'indirizzo :MenuSelection:' Vista -> Panels' e verifica che **lacasella** *cattura coordinate* sia abilitata. La finestra di cattura di coordinate appare come mostrato in Figura [figure_coordinate_capture_1](#). In alternativa, si può anche andare a *Vettore-> Cattura coordinate* verificando che la **lacasella**: *guilabel:' Cattura coordinate'* sia abilitata.
4. Cliccare su  Clicca per selezionare il SR da usare durante la visualizzazione delle coordinate e selezionare un SR diverso da quello selezionato precedentemente.
5. Cliccare su **[Start capture]** per iniziare la cattura delle coordinate. Cliccare un punto nella mappa e il plugin mostrerà le coordinate espresse nei due SR selezionati.
6. Per abilitare la tracciatura via mouse delle coordinate seleziona l'icona  Clicca per abilitare la tracciatura mouse...
7. Le coordinate selezionate possono essere copiate negli appunti.

20.4 Plugin DB Manager

The DB Manager Plugin is officially part of the QGIS core and is intended to replace the SPIT Plugin and, additionally, to integrate all other database formats supported by QGIS in one user interface. The  DB Manager Plugin provides several features. You can drag layers from the QGIS Browser into the DB Manager, and it will import your layer into your spatial database. You can drag and drop tables between spatial databases and they will get imported. You can also use the DB Manager to execute SQL queries against your spatial database and then view the spatial output for queries by adding the results to QGIS as a query layer.

The *Database* menu allows you to connect to an existing database, to start the SQL window and to exit the DB Manager Plugin. Once you are connected to an existing database, the menus *Schema* and *Table* additionally appear.

The *Schema* menu includes tools to create and delete (empty) schemas and, if topology is available (e.g., PostGIS 2), to start a *TopoViewer*.

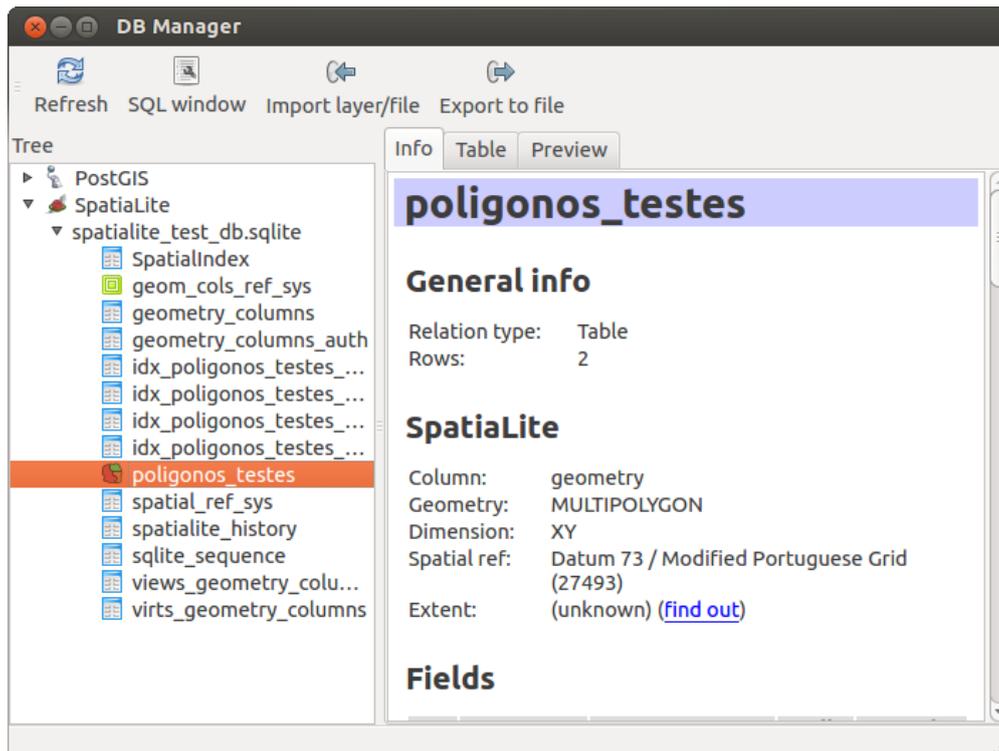


Figura 20.7: Finestra di dialogo DB Manager 

The *Table* menu allows you to create and edit tables and to delete tables and views. It is also possible to empty tables and to move tables from one schema to another. As further functionality, you can perform a VACUUM and then an ANALYZE for each selected table. Plain VACUUM simply reclaims space and makes it available for reuse. ANALYZE updates statistics to determine the most efficient way to execute a query. Finally, you can import layers/files, if they are loaded in QGIS or exist in the file system. And you can export database tables to shape with the Export File feature.

The *Tree* window lists all existing databases supported by QGIS. With a double-click, you can connect to the database. With the right mouse button, you can rename and delete existing schemas and tables. Tables can also be added to the QGIS canvas with the context menu.

If connected to a database, the **main** window of the DB Manager offers three tabs. The *Info* tab provides information about the table and its geometry, as well as about existing fields, constraints and indexes. It also allows you to run Vacuum Analyze and to create a spatial index on a selected table, if not already done. The *Table* tab shows all attributes, and the *Preview* tab renders the geometries as preview.

20.5 Plugin Convertitore DXF2Shape

The dxf2shape converter plugin can be used to convert vector data from DXF to shapefile format. It requires the following parameters to be specified before running:

- **Input DXF file:** Enter the path to the DXF file to be converted.
- **Output Shp file:** Enter desired name of the shapefile to be created.
- **Output file type:** Specify the geometry type of the output shapefile. Currently supported types are polyline, polygon, and point.
- **Export text labels:** When this checkbox is enabled, an additional shapefile point layer will be created, and the associated DBF table will contain information about the “TEXT” fields found in the DXF file, and the text strings themselves.

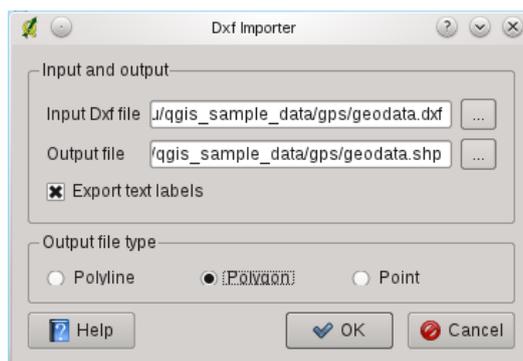


Figura 20.8: Plugin Convertitore DXF2Shape

20.5.1 Utilizzo del plugin

1. Start QGIS, load the Dxf2Shape plugin in the Plugin Manager (see *La finestra di dialogo Plugins*) and click on the  icon, which appears in the QGIS toolbar menu. The Dxf2Shape plugin dialog appears, as shown in *Figure_dxf2shape_1*.
2. Enter the input DXF file, a name for the output shapefile and the shapefile type.
3. Abilitare la casella di controllo *Esporta le etichette di testo*, se si vuole creare un layer aggiuntivo di punti con le etichette.
4. Cliccare su [OK].

20.6 Plugin eVis

(This section is derived from Horning, N., K. Koy, P. Ersts. 2009. eVis (v1.1.0) User's Guide. American Museum of Natural History, Center for Biodiversity and Conservation. Available from <http://biodiversityinformatics.amnh.org/>, and released under the GNU FDL.)

The Biodiversity Informatics Facility at the American Museum of Natural History's (AMNH) Center for Biodiversity and Conservation (CBC) has developed the Event Visualization Tool (eVis), another software tool to add to the suite of conservation monitoring and decision support tools for guiding protected area and landscape planning. This plugin enables users to easily link geocoded (i.e., referenced with latitude and longitude or X and Y coordinates) photographs, and other supporting documents, to vector data in QGIS.

eVis is now automatically installed and enabled in new versions of QGIS, and as with all plugins, it can be disabled and enabled using the Plugin Manager (see *La finestra di dialogo Plugins*).

Il plugin consta di tre moduli, Connessione Database, ID evento, Browser evento che permettono di collegare a vettori in QGIS foto ed altri documenti geocodificati (es. con coordinate X,Y o lat/long).

20.6.1 Browser evento

The Event Browser module provides the functionality to display geocoded photographs that are linked to vector features displayed in the QGIS map window. Point data, for example, can be from a vector file that can be input using QGIS or it can be from the result of a database query. The vector feature must have attribute information associated with it to describe the location and name of the file containing the photograph and, optionally, the compass direction the camera was pointed when the image was acquired. Your vector layer must be loaded into QGIS before running the Event Browser.

Aprire il modulo Browser evento

To launch the Event Browser module, click on *Database* → *eVis* → *eVis Event Browser*. This will open the *Generic Event Browser* window.

The *Event Browser* window has three tabs displayed at the top of the window. The *Display* tab is used to view the photograph and its associated attribute data. The *Options* tab provides a number of settings that can be adjusted to control the behavior of the eVis plugin. Lastly, the *Configure External Applications* tab is used to maintain a table of file extensions and their associated application to allow eVis to display documents other than images.

Scheda Visualizza

To see the *Display* window, click on the *Display* tab in the *Event Browser* window. The *Display* window is used to view geocoded photographs and their associated attribute data.

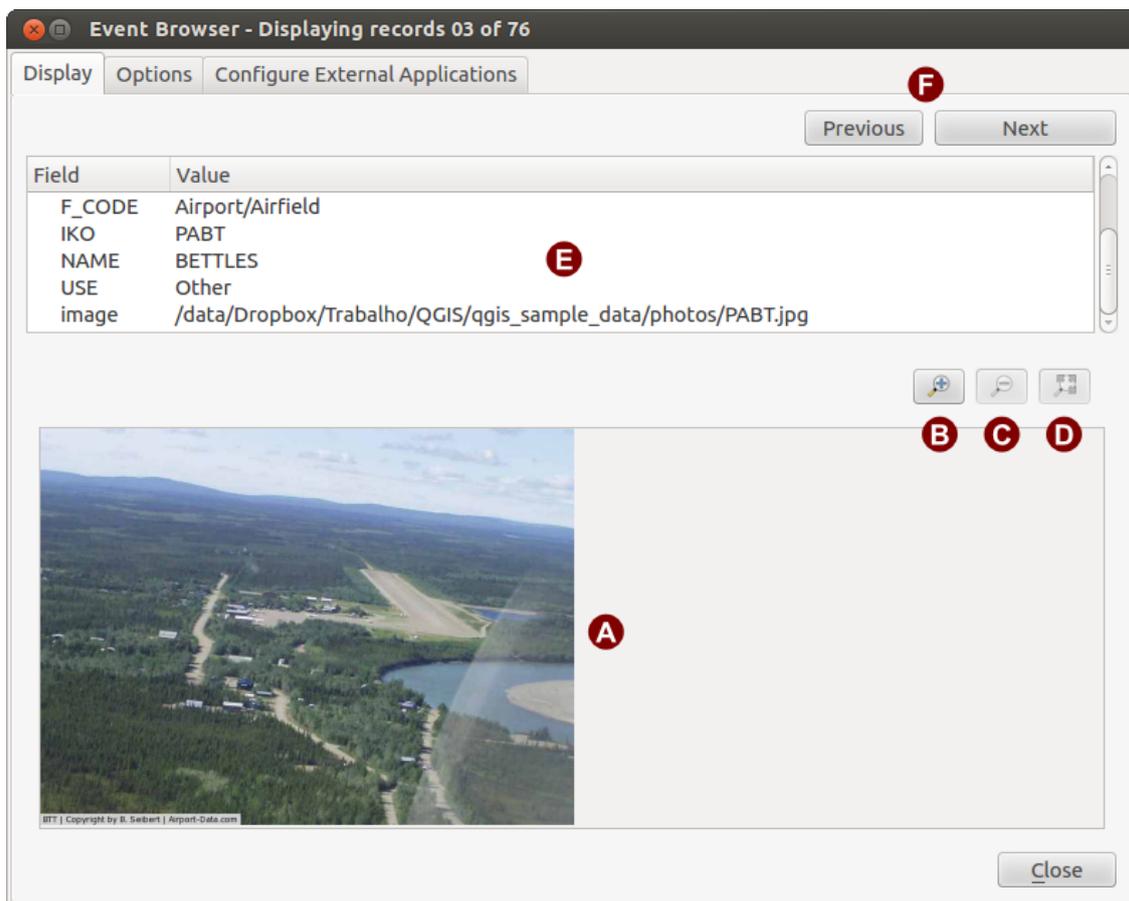


Figura 20.9: The eVis display window

1. **Area di visualizzazione dell'immagine:** è il riquadro inferiore della scheda.
2. **Ingrandisci:** ingrandisce l'immagine per avere più dettagli. Se l'immagine è troppo grande per l'area di visualizzazione, compaiono delle barre di scorrimento.
3. **Rimpicciolisci:** rimpicciolisce l'immagine.
4. **Zoom completo:** visualizza tutta l'immagine.
5. **Attribute information window:** All of the attribute information for the point associated with the photograph being viewed is displayed here. If the file type being referenced in the displayed record is not an image but is of a file type defined in the *Configure External Applications* tab, then when you double-click on the value of the field containing the path to the file, the application to open the file will be launched to

view or hear the contents of the file. If the file extension is recognized, the attribute data will be displayed in green.

6. **Pulsanti per la navigazione:** usare i pulsanti Precedente all'altro.

Scheda Opzioni

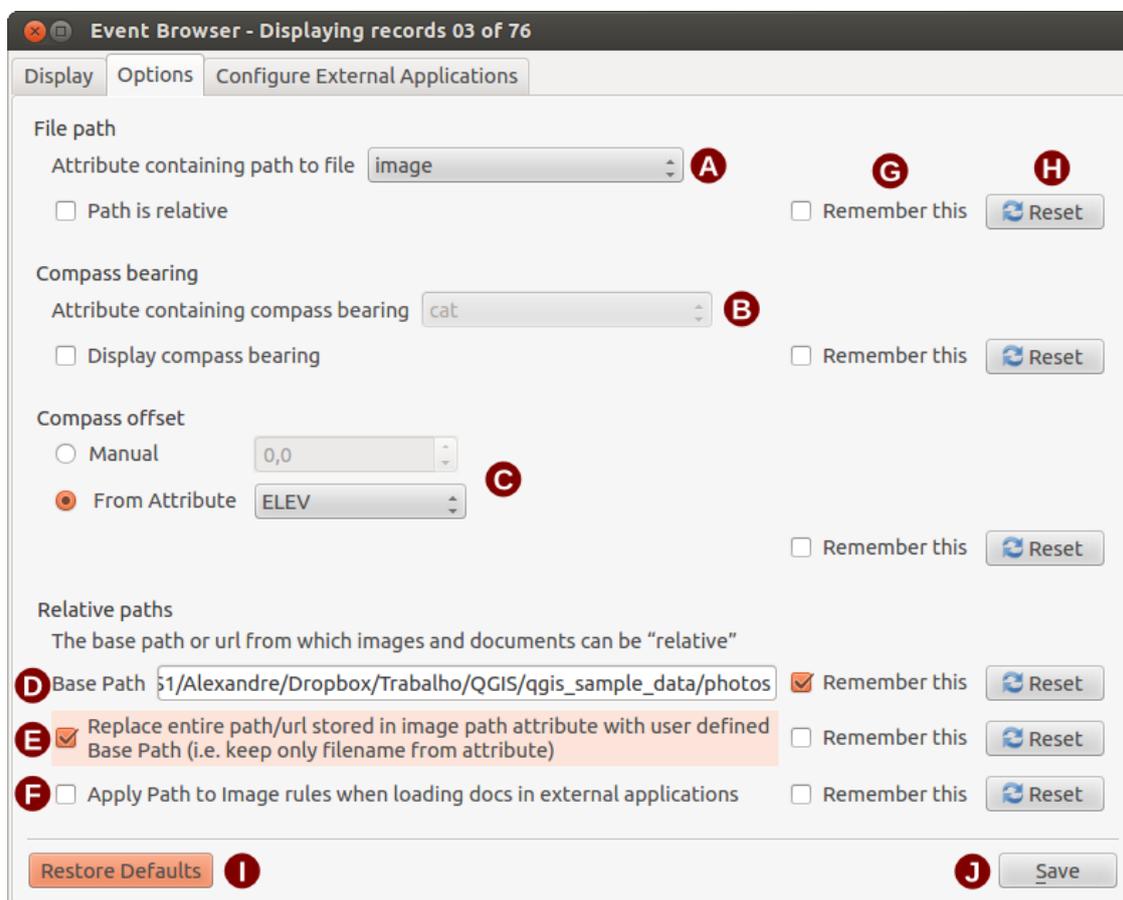


Figura 20.10: The *eVis* Options window

1. **File path:** A drop-down list to specify the attribute field that contains the directory path or URL for the photographs or other documents being displayed. If the location is a relative path, then the checkbox must be clicked. The base path for a relative path can be entered in the *Base Path* text box below. Information about the different options for specifying the file location are noted in the section *Specificare la localizzazione ed il nome di una foto* below.
2. **Compass bearing:** A drop-down list to specify the attribute field that contains the compass bearing associated with the photograph being displayed. If compass bearing information is available, it is necessary to click the checkbox below the drop-down menu title.
3. **Compass offset:** Compass offsets can be used to compensate for declination (to adjust bearings collected using magnetic bearings to true north bearings). Click the *Manual* radio button to enter the offset in the text box or click the *From Attribute* radio button to select the attribute field containing the offsets. For both of these options, east declinations should be entered using positive values, and west declinations should use negative values.
4. **Percorso base:** il percorso di base utilizzato dal percorso relativo definito in Figura [Figure_eVis_2](#) (A).
5. **Replace path:** If this checkbox is checked, only the file name from A will be appended to the base path.

6. **Apply rule to all documents:** If checked, the same path rules that are defined for photographs will be used for non-image documents such as movies, text documents, and sound files. If not checked, the path rules will only apply to photographs, and other documents will ignore the base path parameter.
7. **Remember settings:** If the checkbox is checked, the values for the associated parameters will be saved for the next session when the window is closed or when the **[Save]** button below is pressed.
8. **Ripristina:** reimposta il campo al valore predefinito.
9. **Restore defaults:** This will reset all of the fields to their default settings. It has the same effect as clicking all of the **[Reset]** buttons.
10. **Salva:** salva le impostazioni senza chiudere la scheda *Opzioni*.

Configura applicazioni esterne

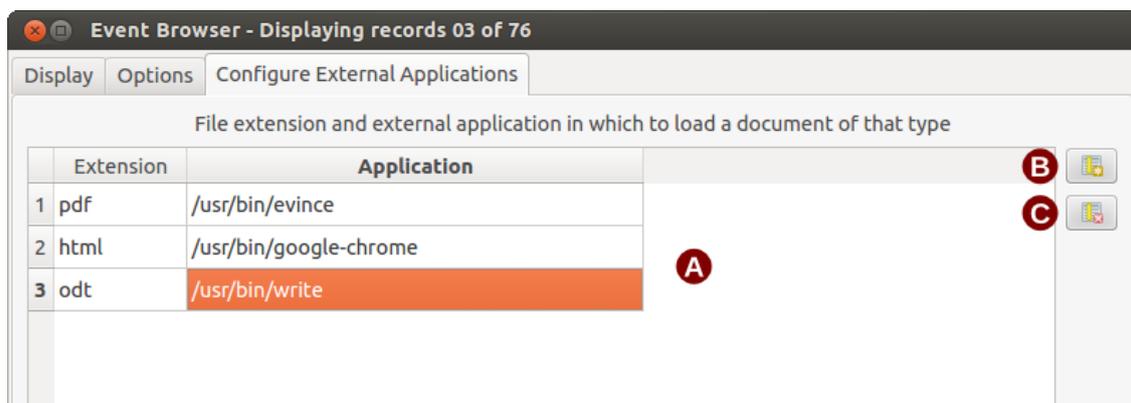


Figura 20.11: The *eVis* External Applications window

1. **Tabella riferimento file:** una tabella contenente i vari tipi di file utilizzati da *eVis*. Ogni tipo file necessita di un'estensione e di un percorso all'applicazione in grado di gestirlo. Ciò permette di aprire diversi tipi di file come filmati, suoni e documenti testuali, oltre che solo immagini.
2. **Aggiungi nuovo tipo file:** aggiunge un nuovo tipo di file (estensione ed applicazione).
3. **Elimina riga corrente:** elimina il tipo di file selezionato in tabella.

20.6.2 Specificare la localizzazione ed il nome di una foto

The location and name of the photograph can be stored using an absolute or relative path, or a URL if the photograph is available on a web server. Examples of the different approaches are listed in Table [evis_examples](#).

X	Y	FILE	BEARING
780596	1784017	C:\Workshop\eVis_Data\groundphotos\DSC_0168.JPG	275
780596	1784017	/groundphotos/DSC_0169.JPG	80
780819	1784015	http://biodiversityinformatics.amnh.org/\ evis_testdata/DSC_0170.JPG	10
780596	1784017	pdf:http://www.testsite.com/attachments.php?\ attachment_id=12	76

20.6.3 Specificare la localizzazione ed il nome di altri documenti

Supporting documents such as text documents, videos, and sound clips can also be displayed or played by *eVis*. To do this, it is necessary to add an entry in the file reference table that can be accessed from the *Configure External Applications* window in the *Generic Event Browser* that matches the file extension to an application that can be

used to open the file. It is also necessary to have the path or URL to the file in the attribute table for the vector layer. One additional rule that can be used for URLs that don't contain a file extension for the document you want to open is to specify the file extension before the URL. The format is — `file extension:URL`. The URL is preceded by the file extension and a colon; this is particularly useful for accessing documents from wikis and other web sites that use a database to manage the web pages (see Table [evis_examples](#)).

20.6.4 Using the Event Browser

When the *Event Browser* window opens, a photograph will appear in the display window if the document referenced in the vector file attribute table is an image and if the file location information in the *Options* window is properly set. If a photograph is expected and it does not appear, it will be necessary to adjust the parameters in the *Options* window.

If a supporting document (or an image that does not have a file extension recognized by eVis) is referenced in the attribute table, the field containing the file path will be highlighted in green in the attribute information window if that file extension is defined in the file reference table located in the *Configure External Applications* window. To open the document, double-click on the green-highlighted line in the attribute information window. If a supporting document is referenced in the attribute information window and the file path is not highlighted in green, then it will be necessary to add an entry for the file's filename extension in the *Configure External Applications* window. If the file path is highlighted in green but does not open when double-clicked, it will be necessary to adjust the parameters in the *Options* window so the file can be located by eVis.

If no compass bearing is provided in the *Options* window, a red asterisk will be displayed on top of the vector feature that is associated with the photograph being displayed. If a compass bearing is provided, then an arrow will appear pointing in the direction indicated by the value in the compass bearing display field in the *Event Browser* window. The arrow will be centered over the point that is associated with the photograph or other document.

To close the *Event Browser* window, click on the **[Close]** button from the *Display* window.

20.6.5 Strumento ID evento

The 'Event ID' module allows you to display a photograph by clicking on a feature displayed in the QGIS map window. The vector feature must have attribute information associated with it to describe the location and name of the file containing the photograph and, optionally, the compass direction the camera was pointed when the image was acquired. This layer must be loaded into QGIS before running the 'Event ID' tool.

Aprire ID Evento

To launch the 'Event ID' module, either click on the  Event ID icon or click on *Database* → *eVis* → *Event ID Tool*. This will cause the cursor to change to an arrow with an 'i' on top of it signifying that the ID tool is active.

To view the photographs linked to vector features in the active vector layer displayed in the QGIS map window, move the Event ID cursor over the feature and then click the mouse. After clicking on the feature, the *Event Browser* window is opened and the photographs on or near the clicked locality are available for display in the browser. If more than one photograph is available, you can cycle through the different features using the **[Previous]** and **[Next]** buttons. The other controls are described in the [ref:evis_browser](#) section of this guide.

20.6.6 Connessione database eVis

Il modulo Connessione Database permette di connettersi ed interrogare un database o altre risorse ODBC, es. un foglio di calcolo.

eVis can directly connect to the following types of databases: PostgreSQL, MySQL, and SQLite; it can also read from ODBC connections (e.g., MS Access). When reading from an ODBC database (such as an Excel spreadsheet), it is necessary to configure your ODBC driver for the operating system you are using.

Aprire Connessione Database

To launch the 'Database Connection' module, either click on the appropriate icon  or click on *Database* → *eVis* → *Database Connection*. This will launch the *Database Connection* window. The window has three tabs: *Predefined Queries*, *Database Connection*, and *SQL Query*. The *Output Console* window at the bottom of the window displays the status of actions initiated by the different sections of this module.

Connessione Database

Click on the *Database Connection* tab to open the database connection interface. Next, use the *Database Type*  combo box to select the type of database that you want to connect to. If a password or username is required, that information can be entered in the *Username* and *Password* textboxes.

Enter the database host in the *Database Host* textbox. This option is not available if you selected 'MS Access' as the database type. If the database resides on your desktop, you should enter "localhost".

Enter the name of the database in the *Database Name* textbox. If you selected 'ODBC' as the database type, you need to enter the data source name.

When all of the parameters are filled in, click on the [**Connect**] button. If the connection is successful, a message will be written in the *Output Console* window stating that the connection was established. If a connection was not established, you will need to check that the correct parameters were entered above.

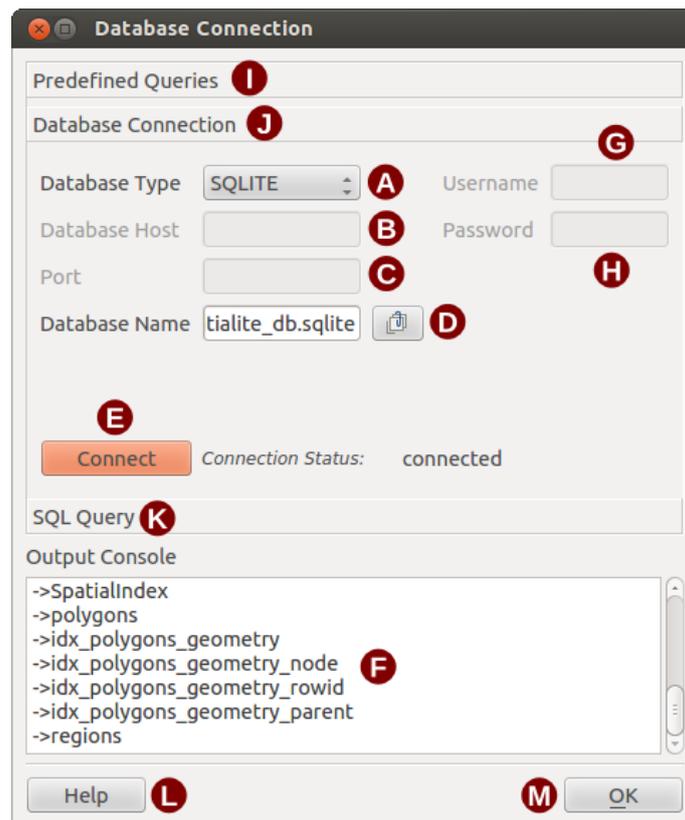


Figura 20.12: The eVis Database connection window

1. **Database Type:** A drop-down list to specify the type of database that will be used.
2. **Host Database:** nome host del database.
3. **Port:** The port number if a MySQL or PostgreSQL database type is selected.
4. **Database Name:** The name of the database.

5. **Connect:** A button to connect to the database using the parameters defined above.
6. **Output Console:** The console window where messages related to processing are displayed.
7. **Nome utente:** nome utente in caso di database protetto.
8. **Password:** password in caso di database protetto.
9. **Query Predefinite:** scheda “Query Predefinite”.
10. **Connessione Database:** scheda “Connessione Database”.
11. **Query SQL:** scheda “Query SQL”.
12. **Help:** Displays the online help.
13. **OK:** chiude Connessione Database .

Eeguire query SQL

SQL queries are used to extract information from a database or ODBC resource. In eVis, the output from these queries is a vector layer added to the QGIS map window. Click on the *SQL Query* tab to display the SQL query interface. SQL commands can be entered in this text window. A helpful tutorial on SQL commands is available at <http://www.w3schools.com/sql>. For example, to extract all of the data from a worksheet in an Excel file, `select * from [sheet1$] where sheet1` is the name of the worksheet.

Click on the **[Run Query]** button to execute the command. If the query is successful, a *Database File Selection* window will be displayed. If the query is not successful, an error message will appear in the *Output Console* window.

Nella finestra *Scegli file Database* assegnare un nome al nuovo layer che sarà creato dai risultati della query.

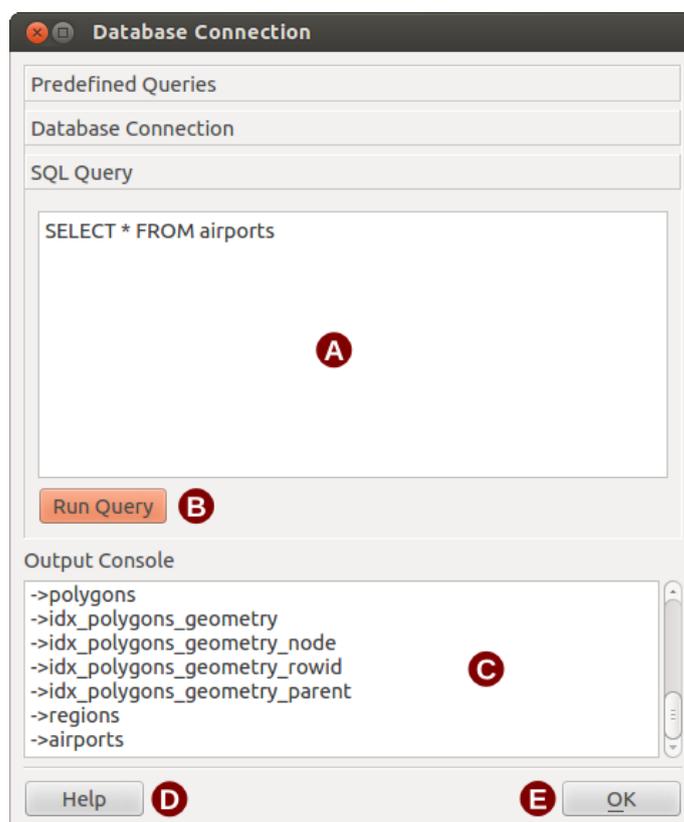


Figura 20.13: The eVis SQL query tab

1. **Query SQL:** è il riquadro per inserire le query SQL.

2. **Esegui Query:** pulsante per mandare in esecuzione una query.
3. **Console di Output:** mostra i messaggi relativi all'esecuzione delle query.
4. **Help:** Displays the online help.
5. **OK:** chiude *Connessione Database*.

Use the *X Coordinate*  and *Y Coordinate*  combo boxes to select the fields from the database that stores the X (or longitude) and Y (or latitude) coordinates. Clicking on the [OK] button causes the vector layer created from the SQL query to be displayed in the QGIS map window.

To save this vector file for future use, you can use the QGIS 'Save as...' command that is accessed by right-clicking on the layer name in the QGIS map legend and then selecting 'Save as...'

Suggerimento: Creare un layer vettoriale da un foglio di lavoro Microsoft Excel

When creating a vector layer from a Microsoft Excel Worksheet, you might see that unwanted zeros ("0") have been inserted in the attribute table rows beneath valid data. This can be caused by deleting the values for these cells in Excel using the `Backspace` key. To correct this problem, you need to open the Excel file (you'll need to close QGIS if you are connected to the file, to allow you to edit the file) and then use *Edit* → *Delete* to remove the blank rows from the file. To avoid this problem, you can simply delete several rows in the Excel Worksheet using *Edit* → *Delete* before saving the file.

Eseguire query predefinite

With predefined queries, you can select previously written queries stored in XML format in a file. This is particularly helpful if you are not familiar with SQL commands. Click on the *Predefined Queries* tab to display the predefined query interface.

To load a set of predefined queries, click on the  *Open File* icon. This opens the *Open File* window, which is used to locate the file containing the SQL queries. When the queries are loaded, their titles as defined in the XML file will appear in the drop-down menu located just below the  *Open File* icon. The full description of the query is displayed in the text window under the drop-down menu.

Select the query you want to run from the drop-down menu and then click on the *SQL Query* tab to see that the query has been loaded into the query window. If it is the first time you are running a predefined query or are switching databases, you need to be sure to connect to the database.

Click on the [Run Query] button in the *SQL Query* tab to execute the command. If the query is successful, a *Database File Selection* window will be displayed. If the query is not successful, an error message will appear in the *Output Console* window.

1. **Open File:** Launches the "Open File" file browser to search for the XML file holding the predefined queries.
2. **Predefined Queries:** A drop-down list with all of the queries defined by the predefined queries XML file.
3. **Descrizione query:** breve descrizione della query derivata dal file XML.
4. **Console di Output:** mostra i messaggi relativi all'esecuzione delle query.
5. **Help:** Displays the online help.
6. **OK:** chiude *Connessione Database* .

Formato XML per le query predefinite di eVis

Tag XML letti da eVis

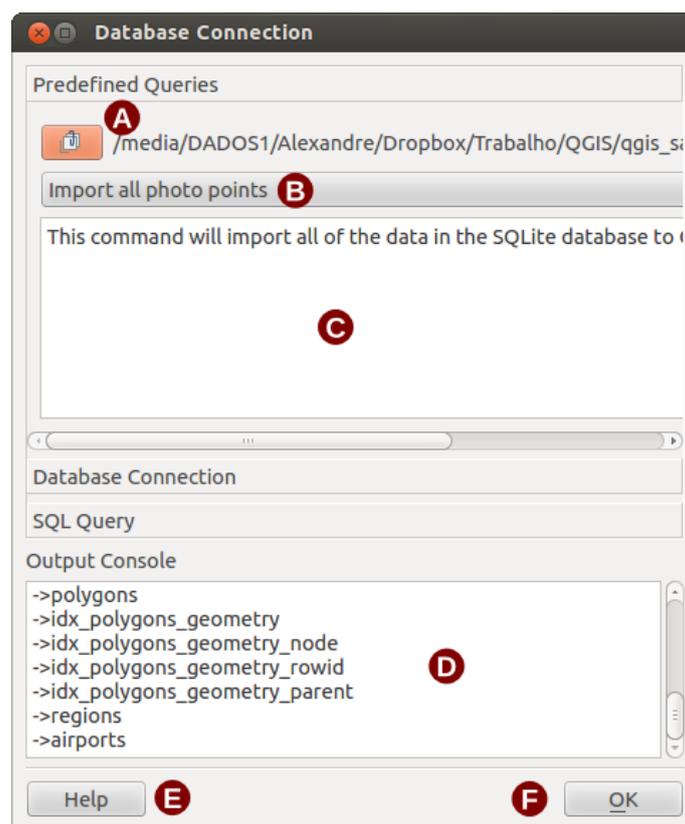


Figura 20.14: The *eVis* Predefined Queries tab

Tag	Descrizione
query	Definisce l'inizio e la fine di una istruzione di query.
shortdescription	A short description of the query that appears in the <i>eVis</i> drop-down menu.
description	Descrizione più dettagliata che viene mostrata nella casella 'Descrizione query' di <i>eVis</i> .
database-type	The database type, defined in the Database Type drop-down menu in the Database Connection tab.
database-port	The port as defined in the Port text box in the Database Connection tab.
database-name	The database name as defined in the Database Name text box in the Database Connection tab.
databaseusername	The database username as defined in the Username text box in the Database Connection tab.
databasepassword	The database password as defined in the Password text box in the Database Connection tab.
sqlstatement	Il comando SQL.
autoconnect	A flag ("true" or "false") to specify if the above tags should be used to automatically connect to the database without running the database connection routine in the Database Connection tab.

Segue un esempio completo di file XML contenente tre query:

```
<?xml version="1.0"?>
<doc>
  <query>
    <shortdescription>Import all photograph points</shortdescription>
    <description>This command will import all of the data in the SQLite database to QGIS
      </description>
    <databasetype>SQLITE</databasetype>
```

```

<databasehost />
<databaseport />
<database>C:\textbackslash Workshop\textbackslash
eVis\_Data\textbackslash PhotoPoints.db</database>
<databaseusername />
<databasepassword />
<sqlstatement>SELECT Attributes.*, Points.x, Points.y FROM Attributes LEFT JOIN
    Points ON Points.rec_id=Attributes.point_ID</sqlstatement>
<autoconnect>>false</autoconnect>
</query>
<query>
<shortdescription>Import photograph points "looking across Valley"</shortdescription>
<description>This command will import only points that have photographs "looking across
    a valley" to QGIS</description>
<database>SQLITE</database>
<databasehost />
<databaseport />
<database>C:\Workshop\eVis_Data\PhotoPoints.db</database>
<databaseusername />
<databasepassword />
<sqlstatement>SELECT Attributes.*, Points.x, Points.y FROM Attributes LEFT JOIN
    Points ON Points.rec_id=Attributes.point_ID where COMMENTS='Looking across
    valley'</sqlstatement>
<autoconnect>>false</autoconnect>
</query>
<query>
<shortdescription>Import photograph points that mention "limestone"</shortdescription>
<description>This command will import only points that have photographs that mention
    "limestone" to QGIS</description>
<database>SQLITE</database>
<databasehost />
<databaseport />
<database>C:\Workshop\eVis_Data\PhotoPoints.db</database>
<databaseusername />
<databasepassword />
<sqlstatement>SELECT Attributes.*, Points.x, Points.y FROM Attributes LEFT JOIN
    Points ON Points.rec_id=Attributes.point_ID where COMMENTS like '%limestone%'
</sqlstatement>
<autoconnect>>false</autoconnect>
</query>
</doc>

```

20.7 Plugin fTools

Il plugin fTools fornisce una risorsa comprensiva delle più comuni operazioni GIS basate su vettori, senza la necessità di software addizionale, librerie e soluzioni complesse: il plugin mette a disposizione una suite di funzioni di analisi veloci e funzionali.

fTools è installato di default nelle nuove versioni di QGIS e, come tutti gli altri plugin, può essere disabilitato nel gestore dei plugin (sezione *La finestra di dialogo Plugins*). Se abilitato, fTools aggiunge il nuovo menu *Vettore* all'interfaccia di QGIS: questo nuovo menu offre funzioni di ricerca, analisi, geoprocessing, gestione dati.

20.7.1 Strumenti di Analisi

Icona	Strumento	Azione
	Matrice di distanza	Misura le distanze tra due layer di punti e fornisce il risultato come a) Matrice di distanza lineare, b) Matrice di distanza standard, c) Sintesi matrice di distanza. Può limitare i calcoli ai 'k' punti più vicini.
	Somma lunghezze linee	Calcola la somma della lunghezza di tutte le linee per ogni poligono di un layer di poligoni.
	Punti nel poligono	Calcola il numero di punti che ricadono all'interno di ogni poligono di un layer di poligoni.
	Lista valori unici	Elenca i valori unici di un campo di un layer vettoriale.
	Statistiche di base	Calcola statistiche di base, es. media, deviazione standard, somma, di un campo di un layer vettoriale.
	Analisi del vicino più prossimo	Calcola le statistiche per valutare il livello di clustering in un layer vettoriale di punti.
	Media coordinata(e)	Calcola il centro medio (media normale o pesata) di un layer vettoriale o di un'insieme di elementi ed in funzione di un campo con ID unico.
	Intersezioni linee	Calcola l'intersezione tra linee e restituisce il risultato in uno shapefile di punti. Utile per localizzare intersezioni fra strade e punti; ignora le intersezioni con una lunghezza > 0.

fTools - Strumenti di Analisi

20.7.2 Strumenti di Ricerca

Icona	Strumento	Azione
	Selezione casuale	Seleziona in maniera casuale un numero intero "n" o percentuale "n%" di elementi.
	Selezione casuale con un sottoinsieme	Selezione casuale in un sottoinsieme tramite campo ID unico.
	Punti casuali	Genera punti pseudo-random.
	Punti regolari	Genera una griglia regolare di punti su un'area specifica e la esporta come shapefile di punti.
	Reticolo vettoriale	Genera una griglia di linee o di poligoni con spaziatura definita dall'utente.
	Selezione per posizione	Seleziona elementi in base alla loro posizione relativa ad un altro layer: crea una nuova selezione oppure aggiunge/sottrae alla selezione corrente.
	Poligono dall'estensione del layer	Crea un poligono rettangolare dall'estensione di un layer raster o vettoriale.

fTools - Strumenti di Ricerca

20.7.3 Strumenti di Geoprocessing

Icona	Strumento	Azione
	Poligono/i convesso/i	Crea il poligono minimo convesso di un layer vettoriale o poligoni minimi convessi sulla base di un campo in input.
	Buffer	Crea buffer intorno ad un elemento in funzione di una distanza impostata o di un campo in input.
	Intersezione	Sovrappone due layer e ne restituisce uno nuovo contenente la superficie di intersezione dei layer di input.
	Unione	Sovrappone due layer e ne restituisce uno nuovo contenente la superficie totale dei layer di input.
	Differenza simmetrica	Sovrappone due layer e ne restituisce uno nuovo contenente la superficie dei layer di input tranne la loro intersezione.
	Clip	Sovrappone due layer e ne restituisce uno nuovo contenente la superficie che interseca il clip layer.
	Differenza	Sovrappone due layer e ne restituisce uno nuovo contenente la superficie che non interseca il clip layer.
	Dissolvenza	Unisce elementi sulla base di un campo in input: gli elementi con lo stesso valore sono combinati in un elemento unico.
	Elimina poligoni frammentati	Unisce le geometrie selezionate con il poligono vicino con la maggiore area o con il più ampio confine comune.

fTools - Strumenti di Geoprocessing

20.7.4 Strumenti di Geometria

Icona	Strumento	Azione
	Verifica la validità della geometria	Controlla i poligoni per verificare la presenza di intersezioni e buchi chiusi e sistemare l'ordine dei nodi.
	Estrai/Aggiungi colonne geometriche	Aggiunge informazioni sulla geometria a layer di punti (XCOORD, YCOORD), di linee (LENGTH), di poligoni (AREA, PERIMETER).
	Centroidi di poligoni	Calcola i centroidi per ogni poligono di un layer di input.
	Triangolazione di Delaunay	Calcola la triangolazione di Delaunay su un layer di punti in input.
	Poligoni di Voronoi	Calcola i poligoni di Voronoi su un layer di punti in ingresso.
	Semplifica geometrie	Generalizza linee e/o poligoni con un algoritmo modificato di Douglas-Peucker.
	Infittisci geometrie	Infittisce linee o poligoni aggiungendo dei vertici
	Da parti multiple a parti singole	Converte elementi multi-parte in più elementi semplici. Crea linee e poligoni semplici.
	Da parti singole a parti multiple	Unisce più elementi in un elemento multi-parte sulla base di un campo in input.
	Da poligoni a linee	Converte poligoni in linee, poligoni multi-parte in linee semplici.
	Da linee a poligoni	Converte linee in poligoni, linee multi-parte in poligoni semplici.
	Estrai vertici	Estrae vertici da layer di linee e poligoni e restituisce un nuovo layer di punti.

fTools - Strumenti di Geometria

Nota: Lo strumento *Semplifica geometrie* può essere usato per rimuovere vertici doppi in layer di linee o di poligoni. Il trucco è impostare il valore 0 per il parametro *Tolleranza di semplificazione*.

20.7.5 Strumenti di Gestione Dati

Icona	Strumento	Azione
	Definisce la proiezione cartografica corrente	Specifica il SR per gli shapefile senza SR associato.
	Unisci attributi per posizione	Aggiunge attributi ad un layer vettoriale sulla base di relazioni spaziali. Attributi di un layer vengono aggiunti alla tabella attributi di un altro layer: il risultato è salvato come nuovo shapefile.
	Dividi vettore	Divide il layer di input in più layer separati sulla base di un campo in input.
	Unisci shapefile	Unisce più shapefile in un unico shapefile sulla base del tipo di layer (punti, linee, poligoni).
	Crea indice spaziale	Crea un indice spaziale per i formati supportati da OGR.

fTools - Strumenti di Gestione Dati

20.8 Plugin GDALTools

20.8.1 Cosa sono gli strumenti GDAL?

Il plugin GDALTools fa da interfaccia grafica ad una collezione di strumenti GDAL (Geospatial Data Abstraction Library), <http://gdal.osgeo.org>: es. strumenti per interrogare, riproiettare, unire raster in vari formati. Sono inclusi strumenti per derivare da un DEM dati tipo curve di livello, pendenze, ombreggiature o anche per collegare in un raster virtuale diversi file raster (Virtual Raster Tile in XML). Tutti questi strumenti sono disponibili quando il plugin è installato e attivato.

La libreria GDAL

La libreria GDAL consiste di un insieme di programmi da linea di comando, ognuno con una vasta lista di opzioni. Gli utenti che si trovano a loro agio lavorando con un terminale preferiranno usare gli strumenti GDAL da riga di comando ed avranno così accesso a tutte le opzioni. Il plugin GDALTools offre anche una semplice interfaccia per utilizzare tali programmi, visualizzando solo le opzioni più utilizzate.

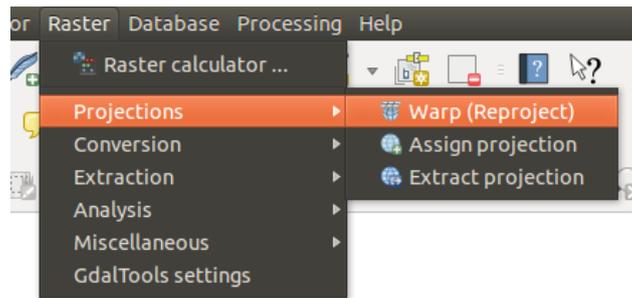


Figura 20.15: La lista degli *Strumenti GDAL*

20.8.2 Lista degli strumenti GDAL

Proiezioni

 <p><i>Riproiezione</i></p>	<p>This utility is an image mosaicing, reprojection and warping utility. The program can reproject to any supported projection, and can also apply GCPs stored with the image if the image is “raw” with control information. For more information, you can read on the GDAL website http://www.gdal.org/gdalwarp.html.</p>
 <p><i>Assegna proiezione</i></p>	<p>This tool allows you to assign projection to rasters that are already georeferenced but miss projection information. Also with its help, it is possible to alter existing projection definitions. Both single file and batch mode are supported. For more information, please visit the utility page at the GDAL site, http://www.gdal.org/gdalwarp.html.</p>
 <p><i>Estrai proiezione</i></p>	<p>This utility helps you to extract projection information from an input file. If you want to extract projection information from a whole directory, you can use the batch mode. It creates both <code>.prj</code> and <code>.wld</code> files.</p>

Conversione

 <p><i>Rasterizzazione</i></p>	<p>This program burns vector geometries (points, lines and polygons) into the raster band(s) of a raster image. Vectors are read from OGR-supported vector formats. Note that the vector data must in the same coordinate system as the raster data; on the fly reprojection is not provided. For more information see http://www.gdal.org/gdal_rasterize.html.</p>
 <p><i>Poligonizzazione</i></p>	<p>This utility creates vector polygons for all connected regions of pixels in the raster sharing a common pixel value. Each polygon is created with an attribute indicating the pixel value of that polygon. The utility will create the output vector datasource if it does not already exist, defaulting to ESRI shapefile format. See also http://www.gdal.org/gdal_polygonize.html.</p>
 <p><i>Conversione formati</i></p>	<p>This utility can be used to convert raster data between different formats, potentially performing some operations like subsetting, resampling, and rescaling pixels in the process. For more information you can read on http://www.gdal.org/gdal_translate.html.</p>
 <p><i>Da RGB a PCT</i></p>	<p>This utility will compute an optimal pseudocolor table for a given RGB image using a median cut algorithm on a downsampled RGB histogram. Then it converts the image into a pseudocolored image using the color table. This conversion utilizes Floyd-Steinberg dithering (error diffusion) to maximize output image visual quality. The utility is also described at http://www.gdal.org/rgb2pct.html.</p>
 <p><i>Da PCT a RGB</i></p>	<p>This utility will convert a pseudocolor band on the input file into an output RGB file of the desired format. For more information, see http://www.gdal.org/pct2rgb.html.</p>

Estrazione

 <p><i>Curve di livello</i></p>	<p>This program generates a vector contour file from the input raster elevation model (DEM). On http://www.gdal.org/gdal_contour.html, you can find more information.</p>
 <p><i>Clipper</i></p>	<p>This utility allows you to clip (extract subset) rasters using selected extent or based on mask layer bounds. More information can be found at http://www.gdal.org/gdal_translate.html.</p>

Analisi

 <p><i>Filtro</i></p>	<p>This utility removes raster polygons smaller than a provided threshold size (in pixels) and replaces them with the pixel value of the largest neighbor polygon. The result can be written back to the existing raster band, or copied into a new file. For more information, see http://www.gdal.org/gdal_sieve.html.</p>
 <p><i>Sposta al nero/bianco</i></p>	<p>This utility will scan an image and try to set all pixels that are nearly black (or nearly white) around the edge to exactly black (or white). This is often used to “fix up” lossy compressed aerial photos so that color pixels can be treated as transparent when mosaicing. See also http://www.gdal.org/nearblack.html.</p>
 <p><i>Riempimento nullo</i></p>	<p>This utility fills selected raster regions (usually nodata areas) by interpolation from valid pixels around the edges of the areas. On http://www.gdal.org/gdal_fillnodata.html, you can find more information.</p>
 <p><i>Prossimità</i></p>	<p>This utility generates a raster proximity map indicating the distance from the center of each pixel to the center of the nearest pixel identified as a target pixel. Target pixels are those in the source raster for which the raster pixel value is in the set of target pixel values. For more information see http://www.gdal.org/gdal_proximity.html.</p>
 <p><i>Interpolazione</i></p>	<p>This utility creates a regular grid (raster) from the scattered data read from the OGR datasource. Input data will be interpolated to fill grid nodes with values, and you can choose from various interpolation methods. The utility is also described on the GDAL website, http://www.gdal.org/gdal_grid.html.</p>
 <p><i>DEM (analisi geomorfologica)</i></p>	<p>Tools to analyze and visualize DEMs. It can create a shaded relief, a slope, an aspect, a color relief, a Terrain Ruggedness Index, a Topographic Position Index and a roughness map from any GDAL-supported elevation raster. For more information, see http://www.gdal.org/gdaldem.html.</p>

Miscellanea

 <i>Crea raster virtuale (catalogo)</i>	<p>This program builds a VRT (Virtual Dataset) that is a mosaic of the list of input GDAL datasets. See also http://www.gdal.org/gdalbuildvrt.html.</p>
 <i>Unione</i>	<p>This utility will automatically mosaic a set of images. All the images must be in the same coordinate system and have a matching number of bands, but they may be overlapping, and at different resolutions. In areas of overlap, the last image will be copied over earlier ones. The utility is also described at http://www.gdal.org/gdal_merge.html.</p>
 <i>Informazioni</i>	<p>This utility lists various information about a GDAL-supported raster dataset. On http://www.gdal.org/gdalinfo.html, you can find more information.</p>
 <i>Crea panoramica</i>	<p>The gdaladdo utility can be used to build or rebuild overview images for most supported file formats with one of several downsampling algorithms. For more information, see http://www.gdal.org/gdaladdo.html.</p>
 <i>Indice delle tile</i>	<p>This utility builds a shapefile with a record for each input raster file, an attribute containing the filename, and a polygon geometry outlining the raster. See also http://www.gdal.org/gdaltindex.html.</p>

GDAL Tools Settings

Use this dialog to embed your GDAL variables.

20.9 Plugin Georeferenziatore

Il Plugin Georeferenziatore è uno strumento per generare file di georeferenziazione (world file) per i raster. Permette di georeferenziare raster in sistemi di coordinate geografiche o proiettate, creando un nuovo GeoTiff oppure associandogli un world file. L'approccio di base del plugin è quello di individuare punti del raster per i quali puoi determinare accuratamente le coordinate.

Features

Icona	Azione	Icona	Azione
	Carica un raster		Avvia la georeferenziazione
	Genera uno script GDAL		Carica punti GCP (Ground Control Point)
	Salva punti GCP		Imposta la trasformazione
	Aggiunge un nuovo punto		Elimina un punto
	Sposta un punto GCP		Sposta la vista
	Ingrandisce la vista		Rimpicciolisce la vista
	Zoom sul layer		Zoom precedente
	Zoom successivo		Collega il georeferenziatore a QGIS
	Collega QGIS al georeferenziatore		Stiramento completo dell'istogramma
	Stiramento locale dell'istogramma		

Strumenti del georeferenziatore

20.9.1 Utilizzo del plugin

Per le coordinate X e Y (espresse in gradi, primi e secondi DMS (dd mm ss.ss), in gradi decimali (DD (dd.dd) o le coordinate proiettate (mmmm.mm) espresse in metri), che corrispondono ai punti selezionati sull'immagine, puoi usare due procedure alternative:

- Alcune volte nei raster sono presenti punti con le coordinate scritte sull'immagine. In questo caso puoi inserire manualmente le coordinate.
- Usare un layer già georiferito (vettore o raster) contenente le stesse entità/oggetti del raster da georiferire. In questo caso puoi inserire le coordinate cliccando sul layer di riferimento nella vista mappa.

Una procedura meno usuale consiste nel selezionare più punti del raster, specificarne le coordinate e scegliere un metodo di trasformazione. Sulla base dei parametri inseriti, il plugin calcola i parametri del world file. Più coordinate vengono fornite, migliore sarà il risultato.

Avvia QGIS, attiva il plugin di georeferenziazione (sezione *load_core_plugin*) e clicca sull'icona  Georeferenziatore che compare nella barra degli strumenti. La finestra di dialogo del plugin Georeferenziatore appare come mostrato in figura *figure_georeferencer_1*.

Come esempio si può provare a georiferire la carta topografica del South Dakota scaricabile da: http://grass.osgeo.org/sampleddata/spearfish_toposheet.tar.gz. Sarà possibile visualizzare la carta anche con i dati di GRASS della location *spearfish60*.

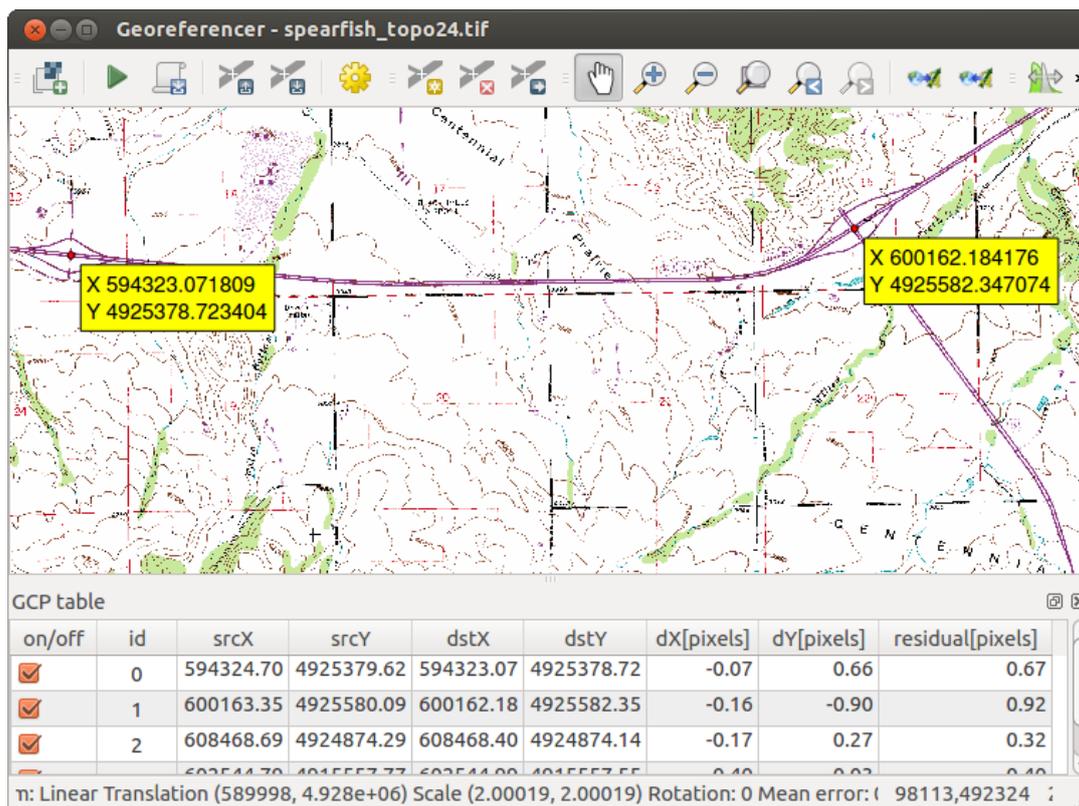


Figura 20.16: Finestra di dialogo del plugin Georeferenziatore 

Aggiungere punti GCP

1. Per iniziare a georeferire un raster non georeferenziato, devi caricarlo utilizzando il pulsante . Il raster verrà mostrato nell'area di lavoro principale dell'interfaccia. Una volta che il raster è stato caricato, puoi procedere con l'inserimento dei punti di riferimento.
2. Utilizzando il pulsante  **Aggiungi punto**, è possibile aggiungere punti all'interno dell'area di lavoro principale e inserire le rispettive coordinate (si veda la Figura [figure_georeferencer_2](#)). Per effettuare questa operazione, esistono tre diverse opzioni:
 - Cliccare su un punto del raster ed inserire le coordinate X/Y manualmente.
 - Clicca su un punto del raster ed usa il pulsante  **Dalla mappa** per inserire le coordinate X/Y con l'aiuto di layer già georeferito caricato nella vista mappa di QGIS.
 - Utilizzando il pulsante , è possibile spostare i punti di controllo in entrambe le finestre, nel caso in cui essi siano posizionati in maniera errata.
3. Continua a inserire punti. Dovresti inserire almeno 4 GCP: più punti vengono inseriti, migliore sarà il risultato. Ci sono strumenti del plugin per spostarsi nell'area di lavoro.

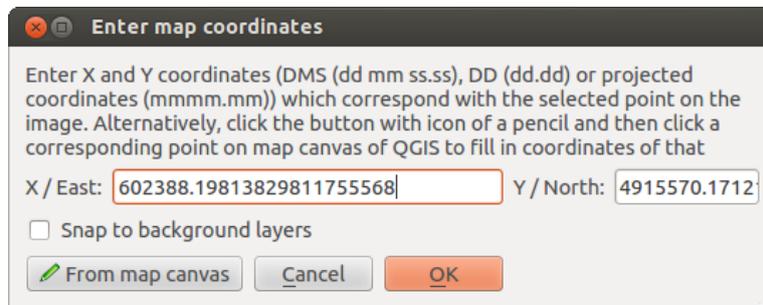


Figura 20.17: Aggiungi punti all'immagine raster 

I punti che sono stati aggiunti all'immagine verranno salvati in un file di testo separato (`[filename].points`), in genere nella stessa cartella in cui è presente l'immagine raster. Questo consente di aprire in un secondo momento il plugin Georeferenziatore e aggiungere nuovi punti oppure cancellarne alcuni, in maniera da migliorare il risultato finale. Il file di punti contiene i valori nel formato: `mapX`, `mapY`, `pixelX`, `pixelY`. La gestione dei file può essere fatta attraverso i pulsanti  **Carica i punti di controllo GCP** e  **Salva i punti di controllo GCP come**.

Impostare una trasformazione

Una volta aggiunti i GCP, è necessario definire le impostazioni di trasformazione del processo di georeferenziazione.

Algoritmi di trasformazione disponibili

Sono disponibili diversi algoritmi di trasformazione: la scelta dipende dal numero di GCP a disposizione, dal tipo e dalla qualità dei dati di input e dall'entità di distorsione geometrica accettabile nel risultato finale.

Sono disponibili le seguenti *tipologie di trasformazione*:

- L'algoritmo di trasformazione **lineare** è utilizzato per generare un file di georeferenziazione o world file, ed è differente rispetto agli altri algoritmi, in quanto attualmente non adatta deformandolo il raster originale. Questo algoritmo in genere non è adatto nel caso in cui si lavori con immagini scannerizzate.
- La trasformazione di **Helmert** esegue una rototraslazione del raster con variazione di scala.

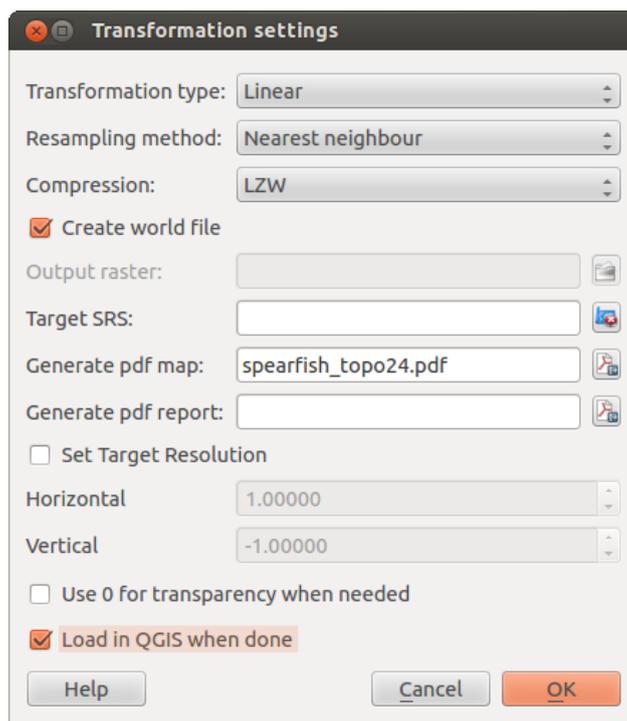


Figura 20.18: Impostare i parametri della georeferenziazione 🐧

- Gli algoritmi **Polinomiali** di grado 1-3 sono i più utilizzati per garantire la maggior coerenza tra i punti inseriti e quelli risultanti dopo la trasformazione. L'algoritmo polinomiale più utilizzato è la trasformazione attraverso un polinomio di secondo grado, che permette alcune curvature nell'immagine. La trasformazione polinomiale di primo grado (affine) permette di conservare la collinearità dei punti, e consente solamente di scalare, traslare e ruotare l'immagine.
- L'algoritmo di trasformazione **Thin Plate Spline (TPS)** è un metodo di georeferenziazione recente, che permette di introdurre delle deformazioni localizzate all'interno dei dati. Questo algoritmo risulta particolarmente efficace quando si vogliono georeferenziare immagini di scarsa qualità.
- La trasformazione **Proiettiva** consiste in una rotazione ed in una traslazione delle coordinate.

Metodo di ricampionamento

La scelta del metodo di ricampionamento dipende dai dati in input e da alcuni requisiti utente. Se, ad esempio, non si accettano modifiche alle statistiche dell'immagine, allora il metodo del vicino più prossimo sarà più adatto. Se, invece, si richiede un risultato più 'liscio' (smoothed) si utilizzerà il metodo cubico.

Puoi scegliere tra cinque diversi metodi di ricampionamento:

1. Vicino più prossimo
2. Lineare
3. Cubico
4. Spline cubica
5. Lanczos

Altre impostazioni di trasformazione

Bisogna definire varie altre opzioni per l'output.

- La casella di controllo *Crea il file di georeferenziazione* è attiva solo se scegli la trasformazione lineare, quando il raster non viene fisicamente deformato. In tal caso, quindi, la casella Raster in output non è attiva perché viene creato solo un nuovo file world.
- Per tutti gli altri tipi di trasformazione devi definire un *Raster in output*. Come modalità predefinita, viene creato un nuovo file ([nomefile]_modificato) nella stessa cartella del raster di partenza.
- Devi poi scegliere il *SR* (Sistema di riferimento) (sezione *Lavorare con le proiezioni*) per il Raster georiferito (vedi *Lavorare con le proiezioni*).
- Puoi creare delle **mappe pdf** e anche dei **report pdf**. Il report fornisce informazioni sui parametri di informazione utilizzati, un'immagine dei residui e una lista con tutti i punti GCP con i loro errori standard.
- Puoi attivare la casella di controllo *imposta risoluzione finale* che permette di definire la risoluzione del raster di output. Il valore predefinito è 1.
- Puoi attivare la casella di controllo *Utilizzare 0 per la trasparenza dove necessario*, i pixel con valore 0 saranno trasparenti.
- La casella di controllo *Carica in QGIS una volta eseguito*, carica l'output nella vista mappa di QGIS a trasformazione terminata.

Proprietà del raster

Cliccando su *Proprietà raster* nel menu *Preferenze*, si apre la finestra di dialogo Proprietà del layer - Raster.

Configurare il georeferenziatore

- Definire se visualizzare coordinate e/o ID dei GCP.
- Imposta le unità dei residui, pixel e unità di mappa.
- Per i report PDF puoi definire margini e dimensione pagina
- Puoi attivare la casella di controllo *Mostra la finestra del georeferenziatore agganciata*.

Eeguire la trasformazione

Una volta acquisiti i GCP necessari ed impostati i vari parametri della trasformazione, clicca su 
 Inizia georeferenziazione

20.10 Plugin Interpolazione

The Interpolation plugin can be used to generate a TIN or IDW interpolation of a point vector layer. It is very simple to handle and provides an intuitive graphical user interface for creating interpolated raster layers (see [Figure_interpolation_1](#)). The plugin requires the following parameters to be specified before running:

- Input **Vector layers**: Specify the input point vector layer(s) from a list of loaded point layers. If several layers are specified, then data from all layers is used for interpolation. Note: It is possible to insert lines or polygons as constraints for the triangulation, by specifying either “points”, “structure lines” or “break lines” in the *Type* combo box.
- **Interpolation attribute**: Select the attribute column to be used for interpolation or enable the *Use Z-Coordinate* checkbox to use the layer's stored Z values.

- **Interpolation Method:** Select the interpolation method. This can be either ‘Triangulated Irregular Network (TIN)’ or ‘Inverse Distance Weighted (IDW)’.
- **Number of columns/rows:** Specify the number of rows and columns for the output raster file.
- **File di output:** nome del raster di output.
- *Aggiungi il risultato al progetto* per caricare il risultato sulla mappa.

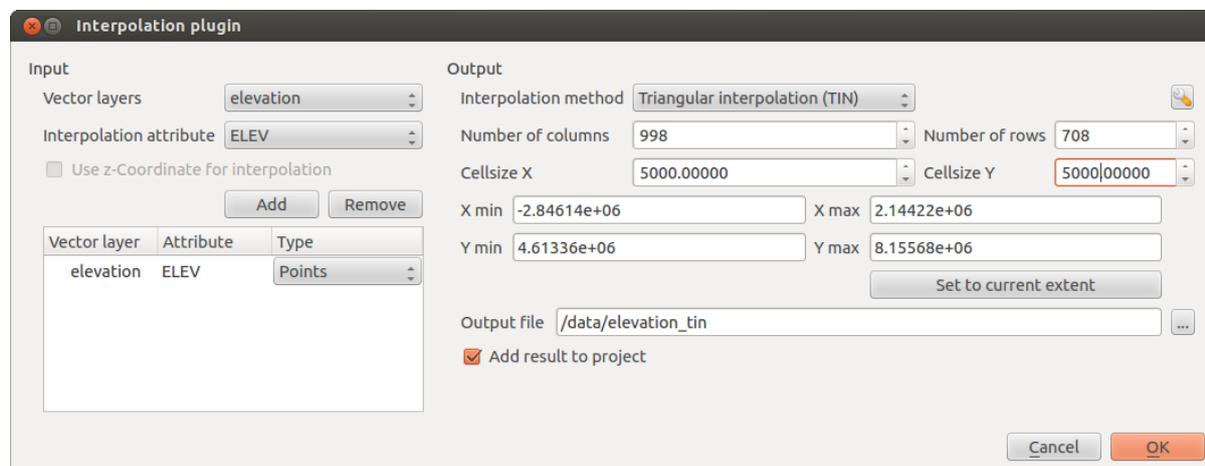


Figura 20.19: Plugin Interpolazione 

20.10.1 Utilizzo del plugin

1. Avvia QGIS e carica un vettore di punti (per esempio, `elevp.csv`).
2. Load the Interpolation plugin in the Plugin Manager (see *La finestra di dialogo Plugins*) and click on the *Raster* → *Interpolation* →  *Interpolation*, which appears in the QGIS menu bar. The Interpolation plugin dialog appears as shown in [Figure_interpolation_1](#).
3. Selezionare un layer in `input` (es. `elevp` ) e una colonna per l'interpolazione (e.g., `ELEV`).
4. Select an interpolation method (e.g., ‘Triangulated Irregular Network (TIN)’), and specify a cell size of 5000 as well as the raster output filename (e.g., `elevation_tin`).
5. Cliccare su **[OK]**.

20.11 Plugin Offline Editing

For data collection, it is a common situation to work with a laptop or a cell phone offline in the field. Upon returning to the network, the changes need to be synchronized with the master datasource (e.g., a PostGIS database). If several persons are working simultaneously on the same datasets, it is difficult to merge the edits by hand, even if people don't change the same features.

Il plugin  `Offline Editing` permette di automatizzare l'attività di sincronizzazione, copiando il contenuto della banca dati principale (solitamente un database PostGIS o un WFS-T) in un database Spatialite e memorizzando le modifiche non in linea in tabelle dedicate: le modifiche, poi, vengono sincronizzate una volta riconnessi alla rete.

20.11.1 Utilizzo del plugin

- Open some vector layers (e.g., from a PostGIS or WFS-T datasource).

- Save it as a project.
- Go to *Database* → *Offline Editing* →  *Convert to offline project* and select the layers to save. The content of the layers is saved to SpatiaLite tables.
- Modificare il layer in modalità non in linea.
- After being connected again, upload the changes using *Database* → *Offline Editing* →  *Synchronize*.

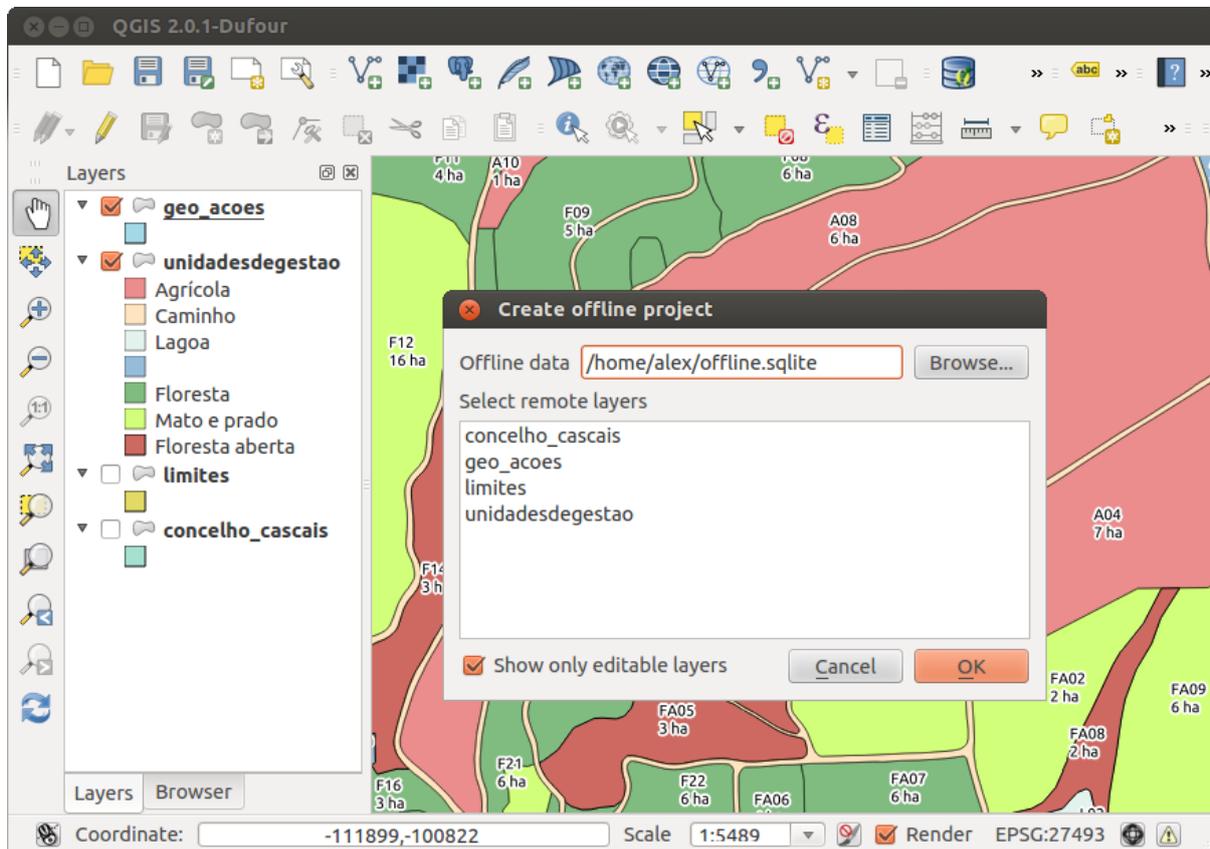


Figura 20.20: Crea un progetto offline da PostGis o layer WFS

20.12 Oracle Spatial GeoRaster Plugin

In Oracle databases, raster data can be stored in SDO_GEORASTER objects available with the Oracle Spatial extension. In QGIS, the  Oracle Spatial GeoRaster plugin is supported by GDAL and depends on Oracle's database product being installed and working on your machine. While Oracle is proprietary software, they provide their software free for development and testing purposes. Here is one simple example of how to load raster images to GeoRaster:

```
$ gdal_translate -of georaster input_file.tif geor:scott/tiger@orcl
```

carica un raster nella tabella predefinita GDAL_IMPORT in una colonna con nome RASTER.

20.12.1 Gestire le connessioni

Firstly, the Oracle GeoRaster Plugin must be enabled using the Plugin Manager (see *La finestra di dialogo Plugins*). The first time you load a GeoRaster in QGIS, you must create a connection to the Oracle database that

contains the data. To do this, begin by clicking on the  toolbar button – this will open the *Select Oracle Spatial GeoRaster* dialog window. Click on **[New]** to open the dialog window, and specify the connection parameters (See [Figure_oracle_raster_1](#)):

- **Name:** Enter a name for the database connection.
- **Database instance:** Enter the name of the database that you will connect to.
- **Username:** Specify your own username that you will use to access the database.
- **Password:** Provide the password associated with your username that is required to access the database.

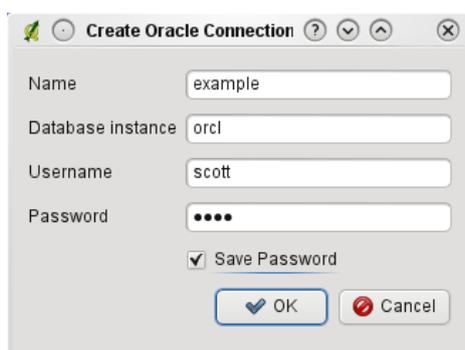


Figura 20.21: Finestra di dialogo di creazione connessione Oracle

Cliccando su **[OK]** parametri della connessione vengono salvati e si ritorna nella finestra di dialogo per la scelta del georaster (figura [Figure_oracle_raster_2](#)), Selezionare la connessione appena impostata e cliccare su **[Connetti]** per modificare la connessione cliccare su **[Modifica]**, per rimuoverla cliccare su **[Elimina]**.

20.12.2 Selezionare un GeoRaster

Once a connection has been established, the subdatasets window will show the names of all the tables that contain GeoRaster columns in that database in the format of a GDAL subdataset name.

Selezionare una tabella con il mouse e cliccare su **[Seleziona]**: apparirà un nuovo elenco con i nomi delle colonne GeoRaster della tabella selezionata.

Click on one of the listed subdatasets and then click on **[Select]** to choose one of the table/column combinations. The dialog will now show all the rows that contain GeoRaster objects. Note that the subdataset list will now show the Raster Data Table and Raster Id pairs.

At any time, the selection entry can be edited in order to go directly to a known GeoRaster or to go back to the beginning and select another table name.

The selection data entry can also be used to enter a `WHERE` clause at the end of the identification string (e.g., `geor:scott/tiger@orcl,gdal_import,raster,geoid=`). See http://www.gdal.org/frmt_georaster.html for more information.

20.12.3 Visualizzare un GeoRaster

Finally, by selecting a GeoRaster from the list of Raster Data Tables and Raster Ids, the raster image will be loaded into QGIS.

The *Select Oracle Spatial GeoRaster* dialog can be closed now and the next time it opens, it will keep the same connection and will show the same previous list of subdatasets, making it very easy to open up another image from the same context.

Nota: GeoRasters that contain pyramids will display much faster, but the pyramids need to be generated outside of QGIS using Oracle PL/SQL or `gdaladdo`.

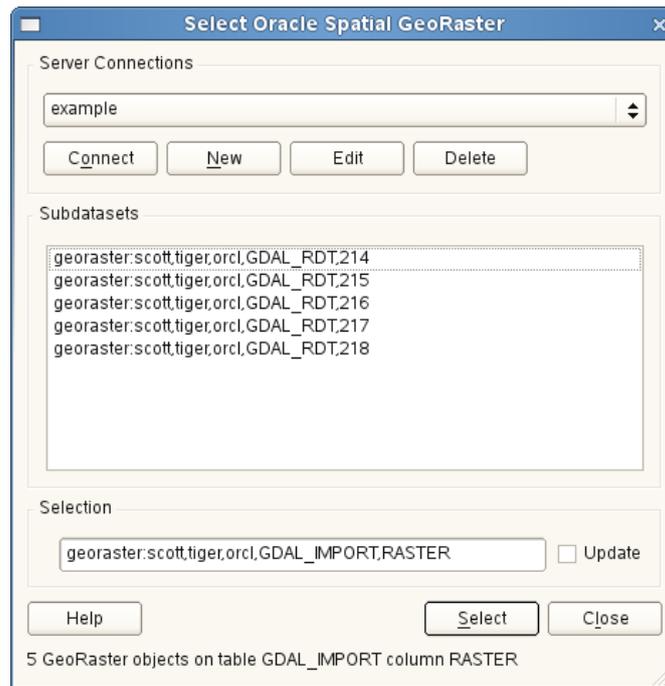


Figura 20.22: Finestra di dialogo di selezione GeoRaster Oracle

The following is an example using `gdaladdo`:

```
gdaladdo georaster:scott/tiger@orcl,georaster\_table,georaster,georid=6 -r
nearest 2 4 6 8 16 32
```

Questo è, invece, un esempio con PL/SQL:

```
$ sqlplus scott/tiger
SQL> DECLARE
  gr sdo_georaster;
BEGIN
  SELECT image INTO gr FROM cities WHERE id = 1 FOR UPDATE;
  sdo_geor.generatePyramid(gr, 'rLevel=5, resampling=NN');
  UPDATE cities SET image = gr WHERE id = 1;
  COMMIT;
END;
```

20.13 Plugin Analisi geomorfologica



The Raster Terrain Analysis Plugin can be used to calculate the slope, aspect, hillshade, ruggedness index and relief for digital elevation models (DEM). It is very simple to handle and provides an intuitive graphical user interface for creating new raster layers (see [Figure_raster_terrain_1](#)).

Descrizione delle analisi:

- **Slope:** Calculates the slope angle for each cell in degrees (based on first- order derivative estimation).
- **Esposizione:** 0 gradi per nord e continuando in senso orario.
- **Hillshade:** Creates a shaded map using light and shadow to provide a more three-dimensional appearance for a shaded relief map.

- **Ruggedness Index:** A quantitative measurement of terrain heterogeneity as described by Riley et al. (1999). It is calculated for every location by summarizing the change in elevation within the 3x3 pixel grid.
- **Relief:** Creates a shaded relief map from digital elevation data. Implemented is a method to choose the elevation colors by analysing the frequency distribution.

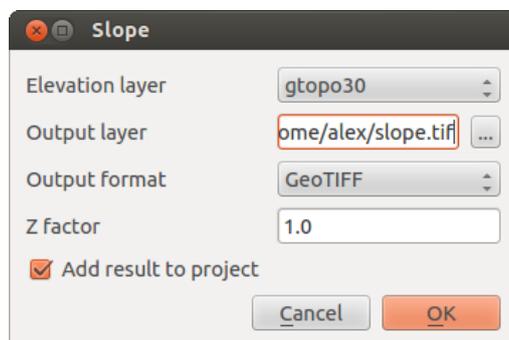


Figura 20.23: Plugin Analisi geomorfologica (calcolo pendenza)

20.13.1 Usare il plugin

1. Avviare QGIS e caricare il file raster `gtopo30` dalla location di GRASS.
2. Load the Raster Terrain Analysis plugin in the Plugin Manager (see *La finestra di dialogo Plugins*).
3. Select an analysis method from the menu (e.g., *Raster* → *Terrain Analysis* → *Slope*). The *Slope* dialog appears as shown in *Figure_raster_terrain_1*.
4. Specificare nome, percorso e formato del file di output.
5. Cliccare su [OK].

20.14 Plugin Mappa di concentrazione

The *Heatmap* plugin uses Kernel Density Estimation to create a density (heatmap) raster of an input point vector layer. The density is calculated based on the number of points in a location, with larger numbers of clustered points resulting in larger values. Heatmaps allow easy identification of “hotspots” and clustering of points.

20.14.1 Attivare il plugin Mappa di concentrazione

First this core plugin needs to be activated using the Plugin Manager (see *La finestra di dialogo Plugins*). After activation, the heatmap icon  can be found in the Raster Toolbar, and under the *Raster* → *Heatmap* menu.

Select the menu *View* → *Toolbars* → *Raster* to show the Raster Toolbar if it is not visible.

20.14.2 Utilizzo del plugin

Clicking the  *Heatmap* tool button opens the Heatmap plugin dialog (see *figure_heatmap_2*).

La finestra di dialogo ha le seguenti opzioni:

- **Input point layer:** Lists all the vector point layers in the current project and is used to select the layer to be analysed.

- **Output raster:** Allows you to use the  button to select the folder and filename for the output raster the Heatmap plugin generates. A file extension is not required.
- **Output format:** Selects the output format. Although all formats supported by GDAL can be chosen, in most cases GeoTIFF is the best format to choose.
- **Radius:** Is used to specify the heatmap search radius (or kernel bandwidth) in meters or map units. The radius specifies the distance around a point at which the influence of the point will be felt. Larger values result in greater smoothing, but smaller values may show finer details and variation in point density.

When the  *Advanced* checkbox is checked, additional options will be available:

- **Rows and Columns:** Used to change the dimensions of the output raster. These values are also linked to the **Cell size X** and **Cell size Y** values. Increasing the number of rows or columns will decrease the cell size and increase the file size of the output file. The values in Rows and Columns are also linked, so doubling the number of rows will automatically double the number of columns and the cell sizes will also be halved. The geographical area of the output raster will remain the same!
- **Cell size X and Cell size Y:** Control the geographic size of each pixel in the output raster. Changing these values will also change the number of Rows and Columns in the output raster.
- **Kernel shape:** The kernel shape controls the rate at which the influence of a point decreases as the distance from the point increases. Different kernels decay at different rates, so a triweight kernel gives features greater weight for distances closer to the point than the Epanechnikov kernel does. Consequently, triweight results in “sharper” hotspots, and Epanechnikov results in “smoother” hotspots. A number of standard kernel functions are available in QGIS, which are described and illustrated on [Wikipedia](#).
- **Decay ratio:** Can be used with Triangular kernels to further control how heat from a feature decreases with distance from the feature.
 - A value of 0 (=minimum) indicates that the heat will be concentrated in the centre of the given radius and completely extinguished at the edge.
 - A value of 0.5 indicates that pixels at the edge of the radius will be given half the heat as pixels at the centre of the search radius.
 - A value of 1 means the heat is spread evenly over the whole search radius circle. (This is equivalent to the ‘Uniform’ kernel.)
 - A value greater than 1 indicates that the heat is higher towards the edge of the search radius than at the centre.

The input point layer may also have attribute fields which can affect how they influence the heatmap:

- **Use radius from field:** Sets the search radius for each feature from an attribute field in the input layer.
- **Use weight from field:** Allows input features to be weighted by an attribute field. This can be used to increase the influence certain features have on the resultant heatmap.

When an output raster file name is specified, the [OK] button can be used to create the heatmap.

20.14.3 Tutorial: Creating a Heatmap

For the following example, we will use the `airports` vector point layer from the QGIS sample dataset (see *Dati campione*). Another excellent QGIS tutorial on making heatmaps can be found at <http://qgis.spatialthoughts.com>.

In [Figure_Heatmap_1](#), the airports of Alaska are shown.

1. Select the  *Heatmap* tool button to open the Heatmap dialog (see [Figure_Heatmap_2](#)).
2. In the *Input point layer*  field, select `airports` from the list of point layers loaded in the current project.
3. Specify an output filename by clicking the  button next to the *Output raster* field. Enter the filename `heatmap_airports` (no file extension is necessary).

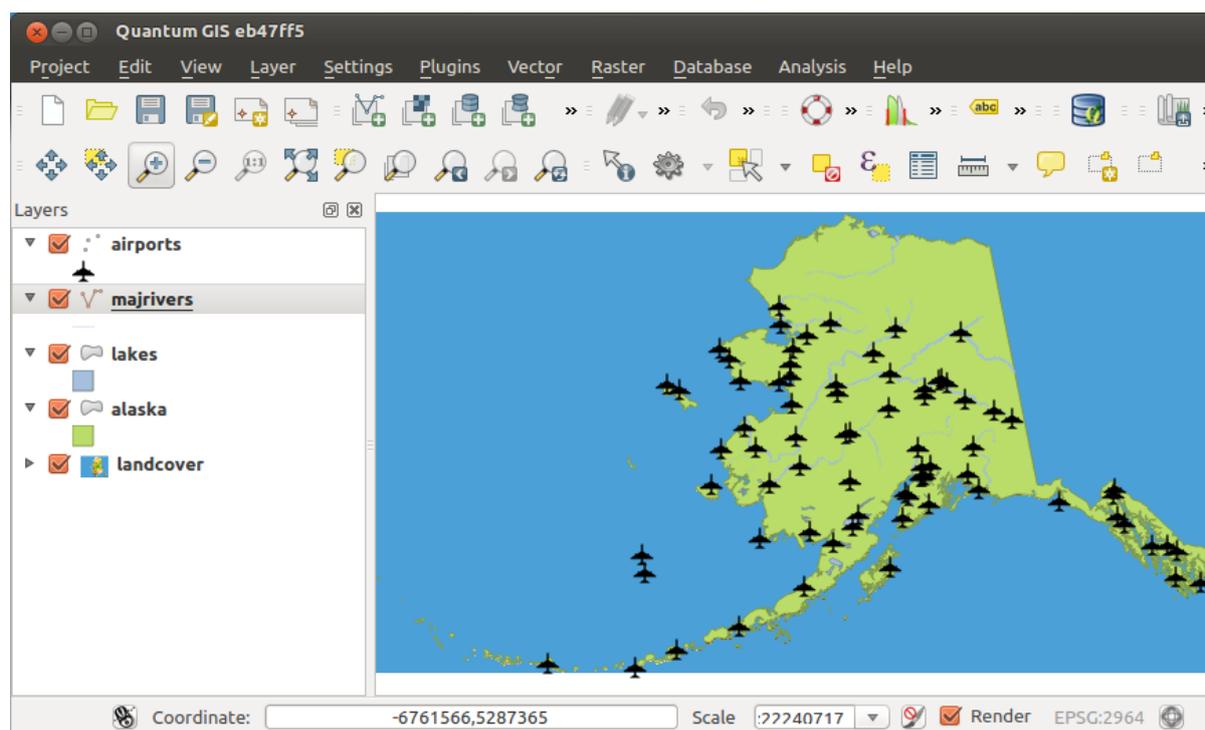


Figura 20.24: Airports of Alaska 🐧

4. Leave the *Output format* as the default format, GeoTIFF.
5. Change the *Radius* to 1000000 meters.
6. Click on [OK] to create and load the airports heatmap (see [Figure_Heatmap_3](#)).

QGIS will generate the heatmap and add the results to your map window. By default, the heatmap is shaded in greyscale, with lighter areas showing higher concentrations of airports. The heatmap can now be styled in QGIS to improve its appearance.

1. Open the properties dialog of the `heatmap_airports` layer (select the layer `heatmap_airports`, open the context menu with the right mouse button and select *Properties*).
2. Select the *Style* tab.
3. Change the *Render type* to 'Singleband pseudocolor'.
4. Select a suitable *Color map* , for instance YlOrRed.
5. Click the [Load] button to fetch the minimum and maximum values from the raster, then click the [Classify] button.
6. Press [OK] to update the layer.

Il risultato finale è mostrato nella figura [Figure_Heatmap_4](#).

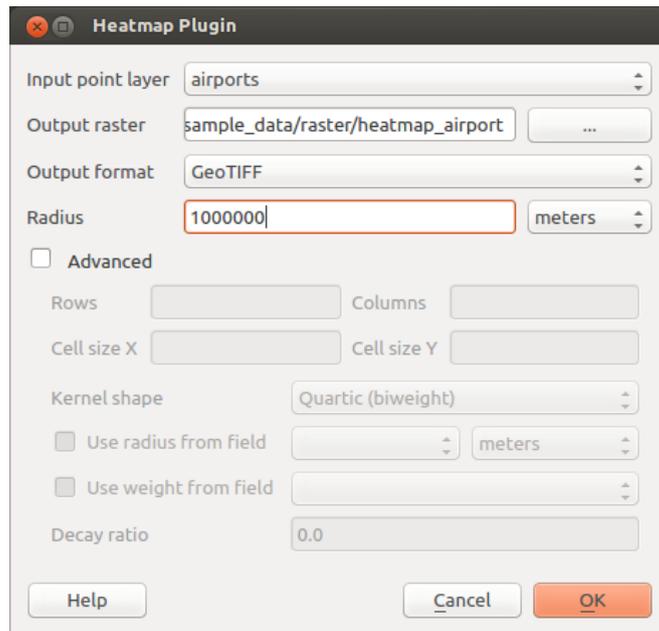


Figura 20.25: The Heatmap Dialog 

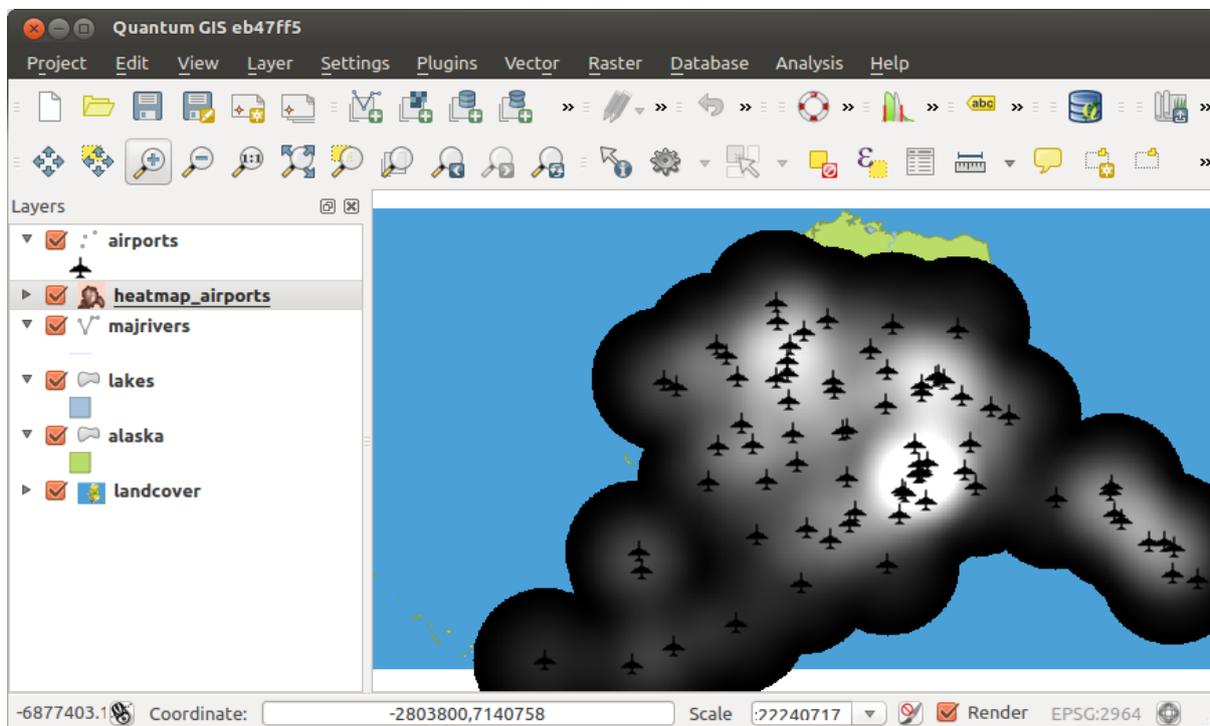


Figura 20.26: The heatmap after loading looks like a grey surface 

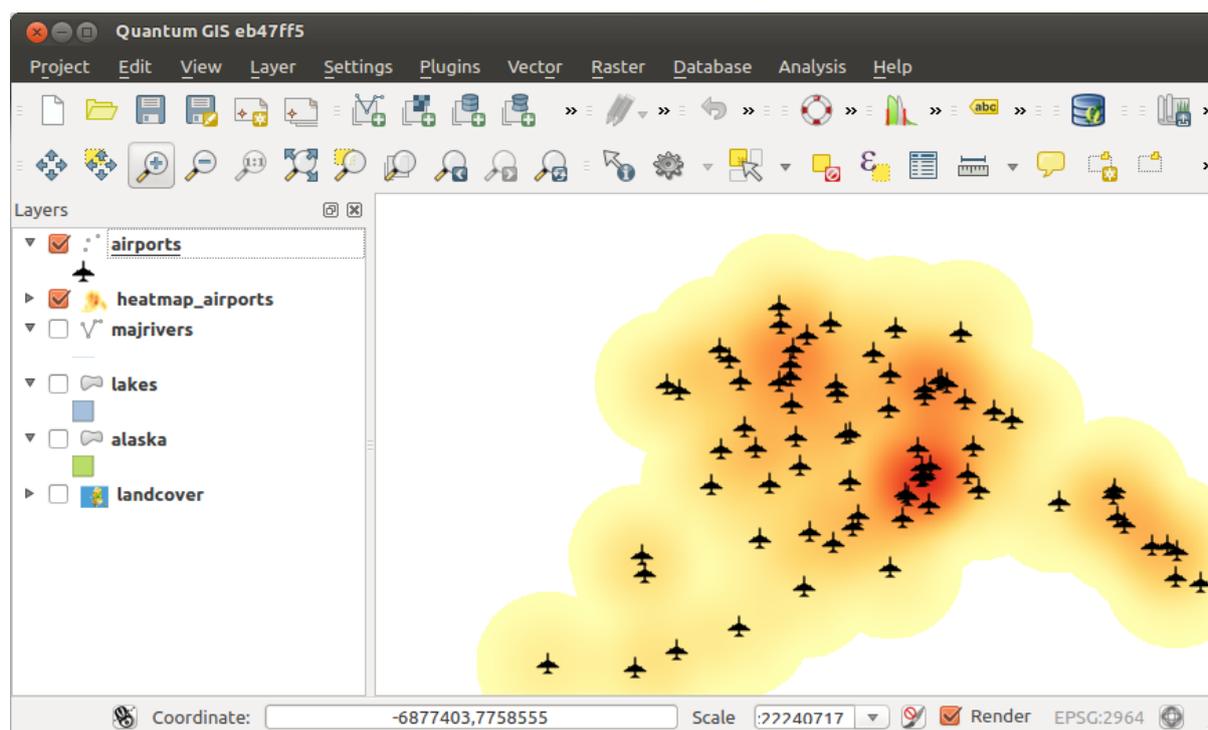
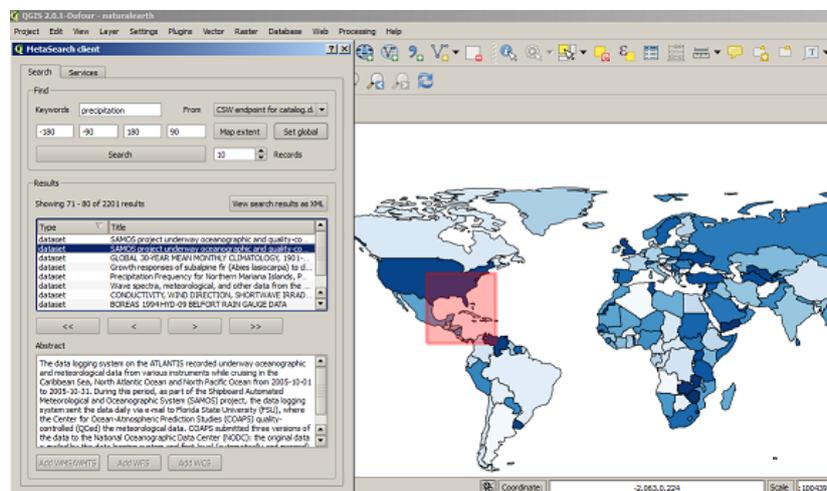


Figura 20.27: Styled heatmap of airports of Alaska 🐧

20.15 Client Catalogo MetaSearch



20.15.1 Introduzione

MetaSearch è un plugin di QGIS per interfacciarsi con i servizi di catalogazione metadati, con il supporto dello standard Catalogue Service for the Web (CSW) dell'OGC.

MetaSearch offre un approccio semplice ed intuitivo con un'interfaccia user-friendly per effettuare ricerche in cataloghi di metadati all'interno di QGIS.

20.15.2 Installazione

MetaSearch è incluso di default con QGIS 2.0 e superiori. Tutte le dipendenze sono inclusi con MetaSearch.

Installa MetaSearch dal gestore di plugin di QGIS o manualmente da <http://plugins.qgis.org/plugins/MetaSearch>.

20.15.3 Lavorare con i cataloghi di metadati in QGIS

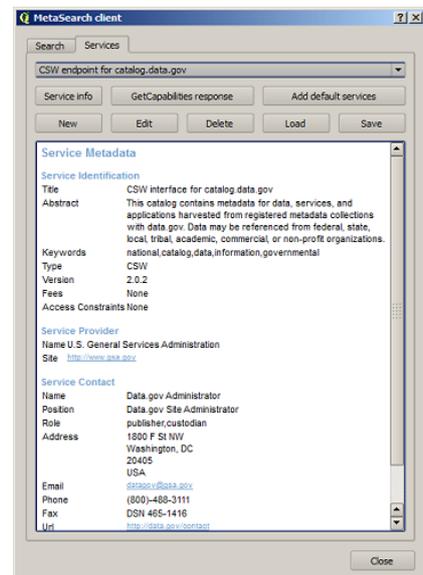
CSW (Catalogue Service for the Web)

CSW (Catalogue Service for the Web) è una specifica del OGC (Open Geospatial Consortium), che definisce un'interfaccia comune per scoprire, esplorare ed interrogare metadati associati a dati, servizi ed altre possibili risorse.

Avvio

Per avviare MetaSearch, fare click sull'icona MetaSearch oppure selezionare il menù Web / MetaSearch / MetaSearch nella barra dei menu di QGIS. L'interfaccia di MetaSearch apparirà. La schermata principale è composta da due sezioni: 'Servizi' e 'Ricerca'.

Gestione servizi catalogazione



La sezione 'Servizi' permette all'utente di gestire tutti i servizi di catalogazione disponibili. MetaSearch offre una lista di servizi predefiniti, che possono essere aggiunti premendo il pulsante 'Aggiungi servizi predefiniti'.

Per l'elenco di tutti i servizi di catalogazione, fare click sul menu a scomparsa.

Per aggiungere un servizio di catalogazione, fare click sul bottone 'Nuovo' ed inserire un nome per il servizio ed il relativo indirizzo Web. Nota che è necessario solamente la base dell'indirizzo Web (non l'URL completo per il GetCapabilities). Facendo click su ok il servizio sarà aggiunto all'elenco.

Per modificare un servizio di catalogazione esistente, sceglierlo dall'elenco e premere il pulsante 'Modifica'. Modificare Nome ed indirizzo Web e premere ok.

Per cancellare un servizio di catalogazione esistente, sceglierlo e premere il pulsante 'Cancella'. Verrà richiesta ulteriore conferma alla cancellazione.

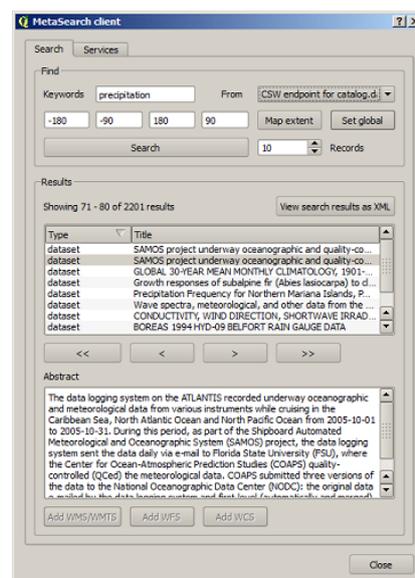
MetaSearch permette di caricare e salvare le connessioni usando un file XML. Questo è utile quando si devono condividere le informazioni tra più applicazioni. Segue un esempio del formato XML per questo tipo di impostazioni.

```
<?xml version="1.0" encoding="UTF-8"?>
<qgsCSWConnections version="1.0">
  <csw name="Data.gov CSW" url="http://catalog.data.gov/csw-all"/>
  <csw name="Geonorge - National CSW service for Norway" url="http://www.geonorge.no/geonetwork/1.0.0/ows?service=CSW&version=1.0.0">
  <csw name="Geoportale Nazionale - Servizio di ricerca Italiano" url="http://www.pcn.minambiente.it/geonetwork/1.0.0/ows?service=CSW&version=1.0.0">
  <csw name="LINZ Data Service" url="http://data.linz.govt.nz/feeds/csw"/>
  <csw name="Nationaal Georegister (Nederland)" url="http://www.nationaalgeoregister.nl/geonetwork/1.0.0/ows?service=CSW&version=1.0.0">
  <csw name="RNDT - Repertorio Nazionale dei Dati Territoriali - Servizio di ricerca" url="http://www.rndt.it/geonetwork/1.0.0/ows?service=CSW&version=1.0.0">
  <csw name="UK Location Catalogue Publishing Service" url="http://csw.data.gov.uk/geonetwork/1.0.0/ows?service=CSW&version=1.0.0">
  <csw name="UNEP/GRID-Geneva Metadata Catalog" url="http://metadata.grid.unep.ch:8080/geonetwork/1.0.0/ows?service=CSW&version=1.0.0">
</qgsCSWConnections>
```

Per caricare un elenco di connessioni, fare click sul pulsante ‘Carica’. Una nuova finestra verrà mostrata; fare click sul pulsante ‘Esplora’ per ricercare la posizione del file da aprire e premere ‘Apri’. L’elenco degli elementi presenti verrà mostrato. Scegli gli elementi da aggiungere e premi ‘Carica’.

Il pulsante ‘Informazioni sul servizio’ mostra informazioni sul servizio di catalogazione selezionato come identificativo, fornitore del servizio ed informazioni di contatto. Se vuoi vedere questa informazione in formato XML, fare click sul pulsante ‘Risposta GetCapabilities’. Una nuova finestra mostrerà il file XML contenente le Capabilities.

Ricerca servizi di catalogazione



La sezione ‘Ricerca’ permette all’utente di interrogare servizi di catalogazione per dati e servizi, impostare parametri di ricerca e visualizzare i risultati.

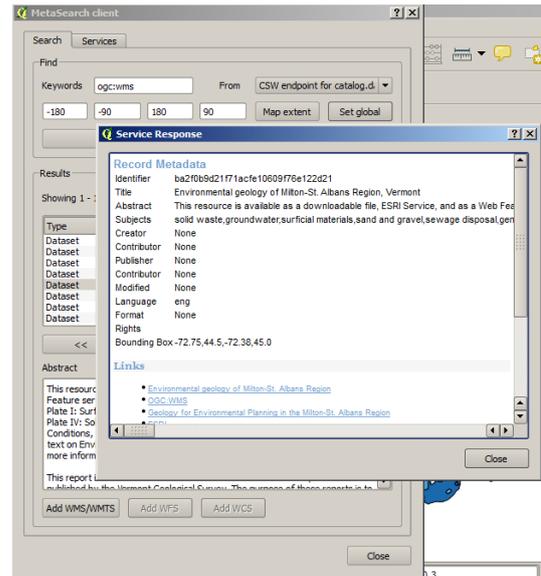
Sono disponibili i seguenti parametri di ricerca:

- **Parole chiave:** ricerca testuale sulle parole chiave
- **Da:** il servizio di catalogazione da interrogare
- **Estensione geografica:** l’estensione da utilizzare come filtro per la ricerca. L’impostazione di default corrisponde a quella visualizzata nella mappa. Selezionando ‘Imposta globale’ per effettuare una ricerca senza limiti, oppure inserisci valori arbitrari.
- **Record:** il numero di record da mostrare quando si esegue una ricerca. Il valore di default è 10.

Cliccando il pulsante ‘Ricerca’ verrà effettuata una ricerca nel catalogo selezionato. I risultati sono mostrati come lista e possono essere ordinati cliccando sul nome della colonna. Puoi esplorare i risultati della ricerca utilizzando i pulsanti sotto la finestra dei risultati. Cliccando ‘Visualizza i risultati di ricerca come XML’ si aprirà una finestra con la risposta del servizio in formato XML.

Selezionando un risultato ne verrà mostrato il riassunto nella finestra ‘Riassunto’ e saranno disponibili le seguenti opzioni:

- se il metadato ha un’estensione geografica associata, ne verranno mostrati i limiti nella mappa
- doppio click su un record ne mostra i metadati con qualsiasi collegamento di accesso associato. Cliccando su un link questo verrà aperto all’interno del browser
- se il record è un web service dell’OGC (WMS/WMTS, WFS, WCS), il relativo pulsante ‘Aggiungi WMS/WMTS/WFS/WCS’ verrà abilitato per permettere all’utente di aggiungerlo in QGIS. Quando si fa click su questo pulsante, MetaSearch verificherà che sia un valido servizio OWS. A questo punto il servizio verrà aggiunto nel relativo elenco di connessioni in QGIS e verrà mostrata la schermata di connessione a WMS/WMTS/WFS/WCS.



Impostazioni

Puoi ottimizzare MetaSearch con le seguenti impostazioni:

- **Risultati per pagina:** il numero di risultati da visualizzare per pagina durante la ricerca nei cataloghi di metadati
- **** Tempo massimo **:** il numero di secondi per bloccare il tentativo di connessione durante la ricerca cataloghi di metadati. Il valore predefinito è 10

20.16 Plugin grafo strade

The Road Graph Plugin is a C++ plugin for QGIS that calculates the shortest path between two points on any polyline layer and plots this path over the road network.

Caratteristiche principali:

- Calculates path, as well as length and travel time.
- Optimizes by length or by travel time.
- Exports path to a vector layer.
- Highlights roads directions (this is slow and used mainly for debug purposes and for the settings testing).

As a roads layer, you can use any polyline vector layer in any QGIS-supported format. Two lines with a common point are considered connected. Please note, it is required to use layer CRS as project CRS while editing a roads

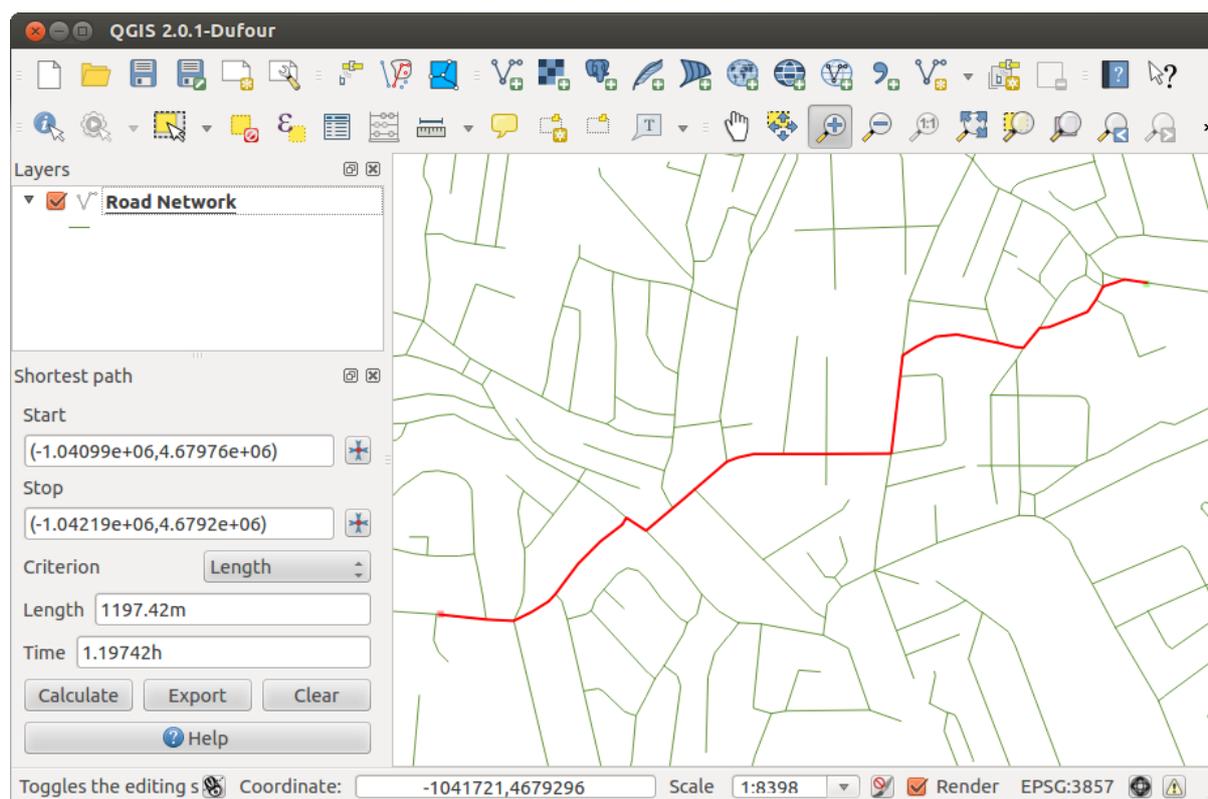


Figura 20.28: Plugin grafo strade 🐧

layer. This is due to the fact that recalculation of the coordinates between different CRSs introduces some errors that can result in discontinuities, even when ‘snapping’ is used.

In the layer attribute table, the following fields can be used:

- Speed on road section (numeric field).
- Direction (any type that can be cast to string). Forward and reverse directions correspond to a one-way road, both directions indicate a two-way road.

If some fields don’t have any value or do not exist, default values are used. You can change defaults and some plugin settings in the plugin settings dialog.

20.16.1 Utilizzo del plugin

After plugin activation, you will see an additional panel on the left side of the main QGIS window. Now, enter some parameters into the *Road graph plugin settings* dialog in the *Vector* → *Road Graph* menu (see [figure_road_graph_2](#)).

After setting the *Time unit*, *Distance unit* and *Topology tolerance*, you can choose the vector layer in the *Transportation layer* tab. Here you can also choose the *Direction field* and *Speed field*. In the *Default settings* tab, you can set the *Direction* for the calculation.

Finally, in the *Shortest Path* panel, select a *Start* and a *Stop* point in the road network layer and click on **[Calculate]**.

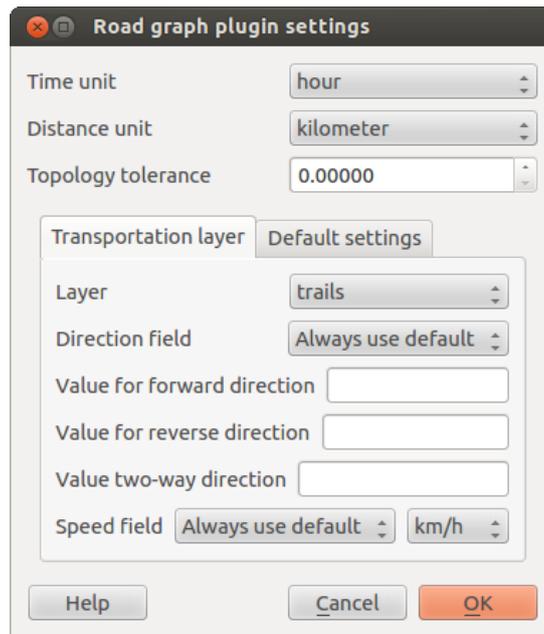


Figura 20.29: Road graph plugin settings 

20.17 Plugin Spatial Query

The  Spatial Query Plugin ti consente di effettuare una interrogazione spaziale (p.e. selezionando attributi particolari) in un layer riferendoti a elementi di un altro livello. Il funzionamento si basa sulla libreria GEOS e dipende dagli attributi selezionati dal layer di riferimento.

Gli operatori spaziali sono:

- Contiene
- E' uguale a
- Sovrappone
- Attraversa
- Interseca
- E' disgiunto
- Tocca
- E' contenuto

20.17.1 Come usare il plugin

Come esempio: trova le regioni dell'Alaska che contengono aeroporti. Sono necessari i seguenti passaggi:

1. Avviare QGIS e caricare i layer vettoriali `regions.shp` e `airports.shp`.
2. Carica il plugin Spatial Query nel Plugin Manager (vedie: ref: *managing_plugins*) e clicca su  Spatial Query icon, che appare in QGIS toolbar menu. Verrà visualizzata la finestra di dialogo del plugin.
3. Seleziona il layer `regions` come layer principale `airports` come layer di riferimento.
4. Seleziona l'operatore 'Contains' e clicca [**Apply**].

Ora avrai una lista di risultato IDs dall'interrogazione, come mostra la [figure_spatial_query_1](#)

- Clicca su  Create layer with list of items .
- Seleziona un ID dalla lista e clicca  Create layer with selected .
- Seleziona 'Rimuovi dalla selezione' nel campo *And use the result to*  .
- Inoltre puoi ingrandire *Zoom to item* o visualizzare *Log messages*.

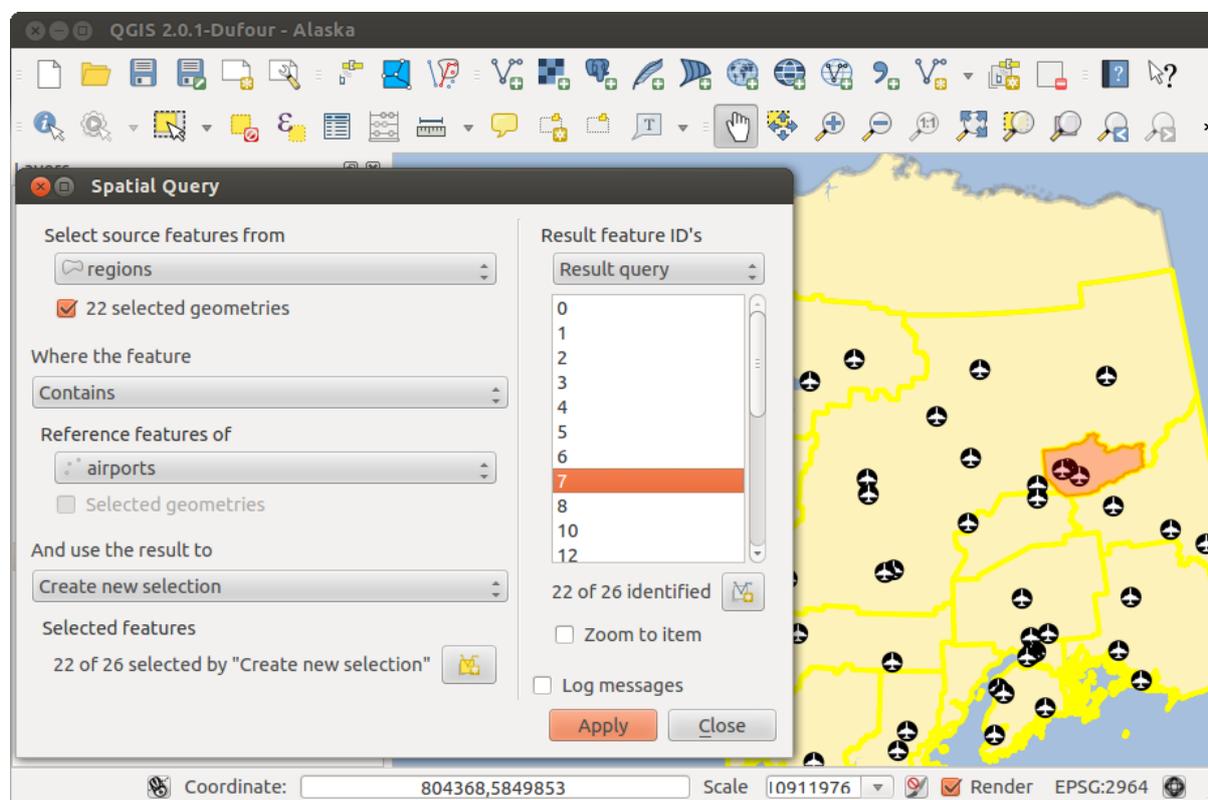


Figura 20.30: Analisi Spaziale - regioni che contengono aeroporti 

20.18 Plugin SPIT

QGIS comes with a plugin named SPIT (Shapefile to PostGIS Import Tool). SPIT can be used to load multiple shapefiles at one time and includes support for schemas. To use SPIT, open the Plugin Manager from the *Plugins* menu, in the  *Installed* menu check the box next to the *SPIT* and click [OK].

To import a shapefile, use *Database* → *Spit* → *Import Shapefiles to PostgreSQL* from the menu bar to open the *SPIT - Shapefile to PostGIS Import Tool* dialog. Select the PostGIS database you want to connect to and click on [Connect]. If you want, you can define or change some import options. Now you can add one or more files to the queue by clicking on the [Add] button. To process the files, click on the [OK] button. The progress of the import as well as any errors/warnings will be displayed as each shapefile is processed.

20.19 Plugin SQL Anywhere

SQL Anywhere is a proprietary relational database management system (RDBMS) from Sybase. SQL Anywhere provides spatial support, including OGC, shapefiles and built-in functions to export to KML, GML and SVG formats.

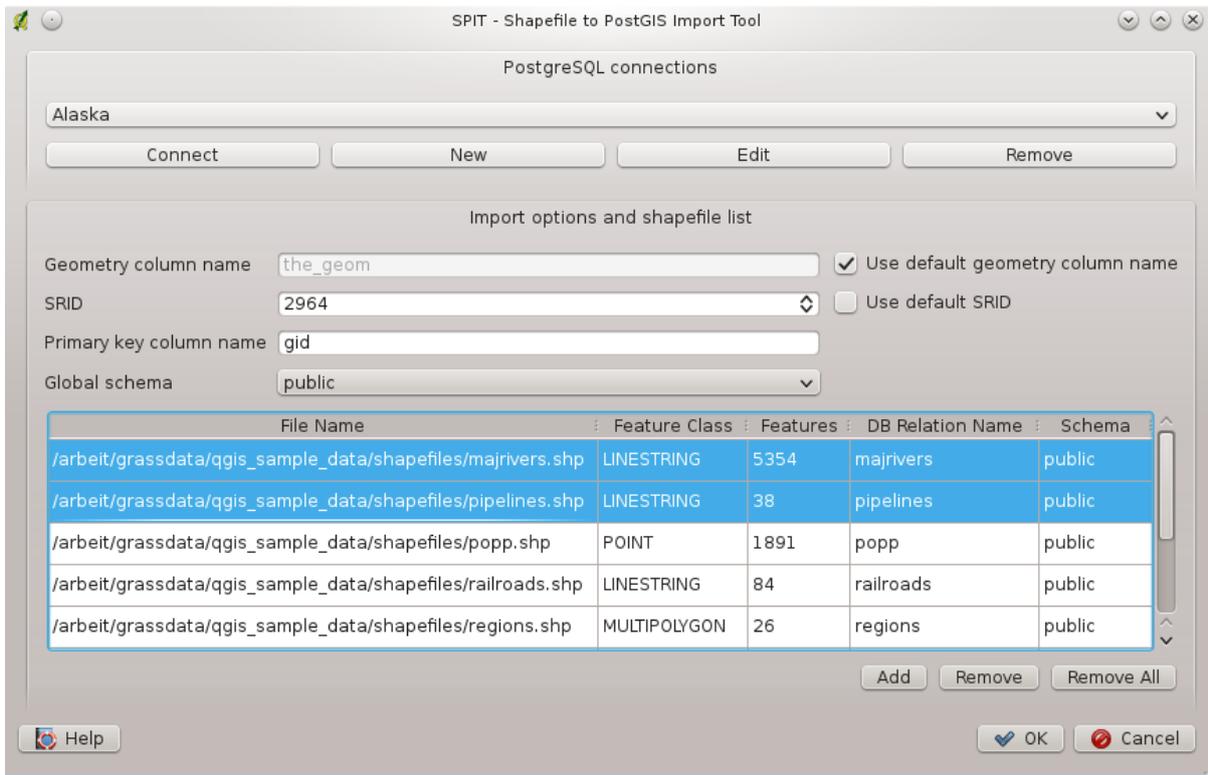


Figura 20.31: Using SPIT Plugin to import Shape files to PostGIS 🐧

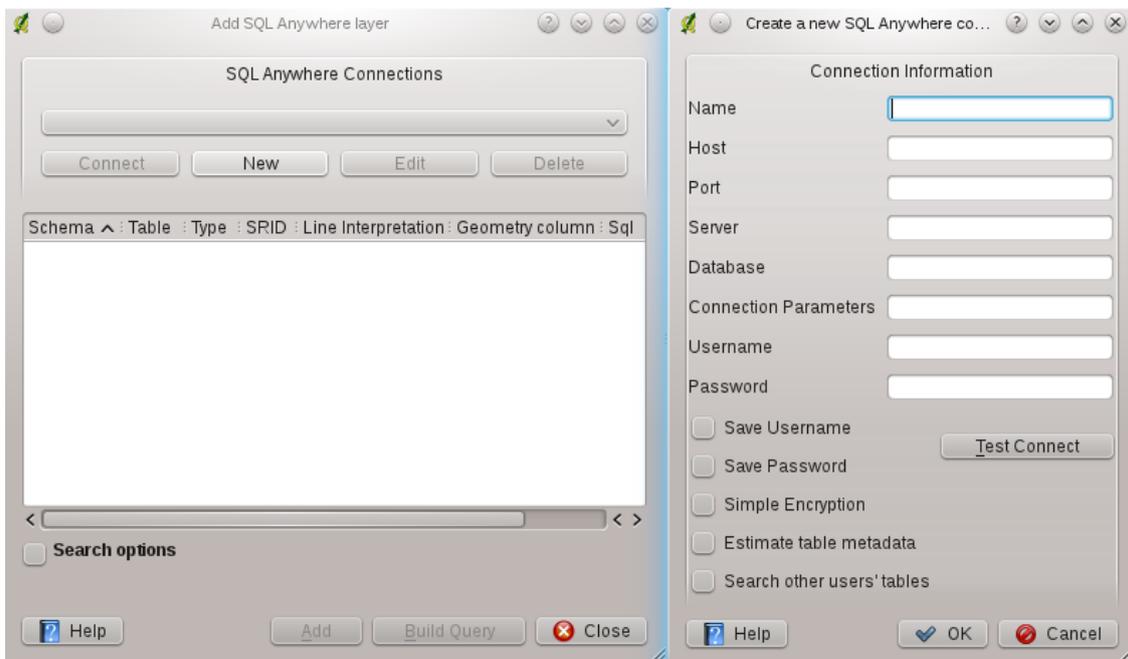


Figura 20.32: SQL Anywhere dialog (KDE) 🐧

 SQL Anywhere allows you to connect to spatially enabled SQL Anywhere databases. The *Add SQL Anywhere layer* dialog is similar in functionality to the dialogs for PostGIS and SpatialLite.

20.20 Topology Checker Plugin

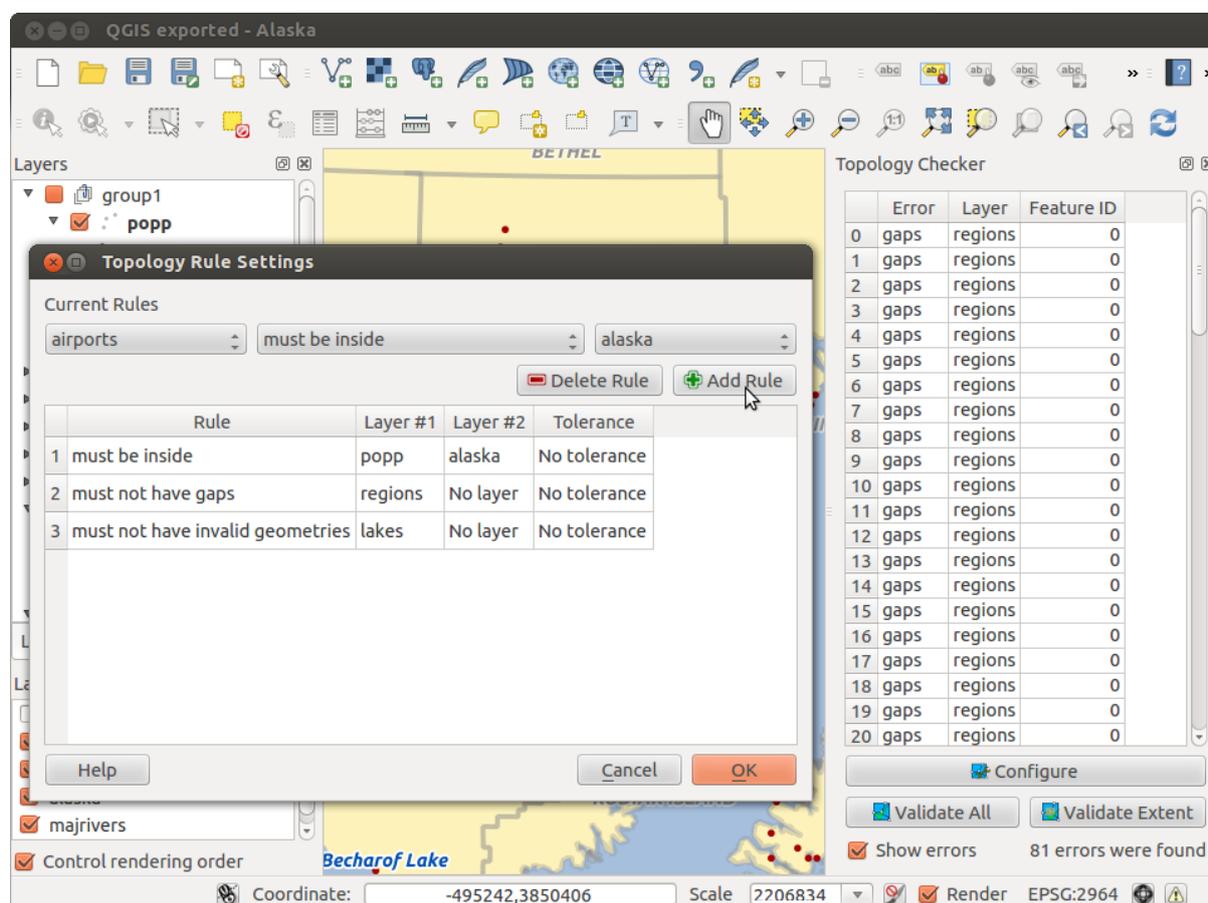


Figura 20.33: The Topology Checker Plugin

Topology describes the relationships between points, lines and polygons that represent the features of a geographic region. With the Topology Checker plugin, you can look over your vector files and check the topology with several topology rules. These rules check with spatial relations whether your features 'Equal', 'Contain', 'Cover', are 'CoveredBy', 'Cross', are 'Disjoint', 'Intersect', 'Overlap', 'Touch' or are 'Within' each other. It depends on your individual questions which topology rules you apply to your vector data (e.g., normally you won't accept overshoots in line layers, but if they depict dead-end streets you won't remove them from your vector layer).

QGIS has a built-in topological editing feature, which is great for creating new features without errors. But existing data errors and user-induced errors are hard to find. This plugin helps you find such errors through a list of rules.

It is very simple to create topology rules with the Topology Checker plugin.

On **point layers** the following rules are available:

- **Must be covered by:** Here you can choose a vector layer from your project. Points that aren't covered by the given vector layer occur in the 'Error' field.
- **Must be covered by endpoints of:** Here you can choose a line layer from your project.
- **Must be inside:** Here you can choose a polygon layer from your project. The points must be inside a polygon. Otherwise, QGIS writes an 'Error' for the point.

- **Must not have duplicates:** Whenever a point is represented twice or more, it will occur in the ‘Error’ field.
- **Must not have invalid geometries:** Checks whether the geometries are valid.
- **Must not have multi-part-geometries:** All multi-part points are written into the ‘Error’ field.

On **line layers**, the following rules are available:

- **End points must be covered by:** Here you can select a point layer from your project.
- **Must not have dangles:** This will show the overshoots in the line layer.
- **Must not have duplicates:** Whenever a line feature is represented twice or more, it will occur in the ‘Error’ field.
- **Must not have invalid geometries:** Checks whether the geometries are valid.
- **Must not have multi-part geometries:** Sometimes, a geometry is actually a collection of simple (single-part) geometries. Such a geometry is called multi-part geometry. If it contains just one type of simple geometry, we call it multi-point, multi-linestring or multi-polygon. All multi-part lines are written into the ‘Error’ field.
- **Must not have pseudos:** A line geometry’s endpoint should be connected to the endpoints of two other geometries. If the endpoint is connected to only one other geometry’s endpoint, the endpoint is called a pseudo node.

On **polygon layers**, the following rules are available:

- **Must contain:** Polygon layer must contain at least one point geometry from the second layer.
- **Must not have duplicates:** Polygons from the same layer must not have identical geometries. Whenever a polygon feature is represented twice or more it will occur in the ‘Error’ field.
- **Must not have gaps:** Adjacent polygons should not form gaps between them. Administrative boundaries could be mentioned as an example (US state polygons do not have any gaps between them...).
- **Must not have invalid geometries:** Checks whether the geometries are valid. Some of the rules that define a valid geometry are:
 - Polygon rings must close.
 - Rings that define holes should be inside rings that define exterior boundaries.
 - Rings may not self-intersect (they may neither touch nor cross one another).
 - Rings may not touch other rings, except at a point.
- **Must not have multi-part geometries:** Sometimes, a geometry is actually a collection of simple (single-part) geometries. Such a geometry is called multi-part geometry. If it contains just one type of simple geometry, we call it multi-point, multi-linestring or multi-polygon. For example, a country consisting of multiple islands can be represented as a multi-polygon.
- **Must not overlap:** Adjacent polygons should not share common area.
- **Must not overlap with:** Adjacent polygons from one layer should not share common area with polygons from another layer.

20.21 Plugin Statistica zonale

With the  *Zonal statistics* plugin, you can analyze the results of a thematic classification. It allows you to calculate several values of the pixels of a raster layer with the help of a polygonal vector layer (see [figure_zonal_statistics](#)). You can calculate the sum, the mean value and the total count of the pixels that are within a polygon. The plugin generates output columns in the vector layer with a user-defined prefix.

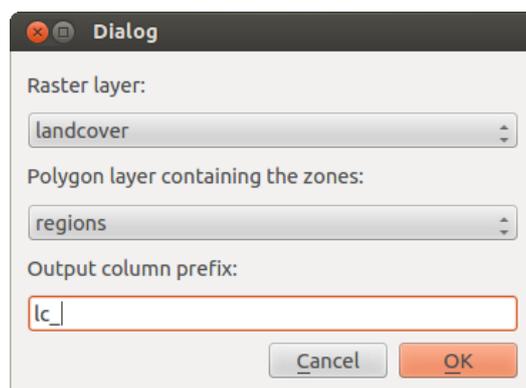


Figura 20.34: Finestra di dialogo Statistica zonale (KDE) 

Aiuto e supporto

21.1 Le Mailing list

QGIS is under active development and as such it won't always work like you expect it to. The preferred way to get help is by joining the qgis-users mailing list. Your questions will reach a broader audience and answers will benefit others.

21.1.1 qgis-users

This mailing list is used for discussion of QGIS in general, as well as specific questions regarding its installation and use. You can subscribe to the qgis-users mailing list by visiting the following URL: <http://lists.osgeo.org/mailman/listinfo/qgis-user>

21.1.2 fossgis-talk-liste

For the German-speaking audience, the German FOSSGIS e.V. provides the fossgis-talk-liste mailing list. This mailing list is used for discussion of open-source GIS in general, including QGIS. You can subscribe to the fossgis-talk-liste mailing list by visiting the following URL: <https://lists.fossgis.de/mailman/listinfo/fossgis-talk-liste>

21.1.3 qgis-developer

Gli sviluppatori con problemi di natura più tecnica possono unirsi alla mailing list qgis-developer andando qui: <http://lists.osgeo.org/mailman/listinfo/qgis-developer>

21.1.4 qgis-commit

Each time a commit is made to the QGIS code repository, an email is posted to this list. If you want to be up-to-date with every change to the current code base, you can subscribe to this list at: <http://lists.osgeo.org/mailman/listinfo/qgis-commit>

21.1.5 qgis-trac

Questa mailing list notifica email collegate alla gestione del progetto, inclusi i bug report, i compiti assegnati e le richieste di nuove caratteristiche. Potete sottoscriverla all'URL: <http://lists.osgeo.org/mailman/listinfo/qgis-trac>

21.1.6 qgis-community-team

This list deals with topics like documentation, context help, user guide, web sites, blog, mailing lists, forums, and translation efforts. If you would like to work on the user guide as well, this list is a good starting point to ask your questions. You can subscribe to this list at: <http://lists.osgeo.org/mailman/listinfo/qgis-community-team>

21.1.7 qgis-release-team

This list deals with topics like the release process, packaging binaries for various OSs and announcing new releases to the world at large. You can subscribe to this list at: <http://lists.osgeo.org/mailman/listinfo/qgis-release-team>

21.1.8 qgis-tr

Questa lista si occupa delle traduzioni. Chi desidera lavorare alla traduzione dei manuali o dell'interfaccia grafica (GUI) trova in questa lista un buon punto di partenza per le proprie domande. La lista può essere sottoscritta all'URL <http://lists.osgeo.org/mailman/listinfo/qgis-tr>

21.1.9 qgis-edu

This list deals with QGIS education efforts. If you would like to work on QGIS education materials, this list is a good starting point to ask your questions. You can subscribe to this list at: <http://lists.osgeo.org/mailman/listinfo/qgis-edu>

21.1.10 qgis-psc

This list is used to discuss Steering Committee issues related to overall management and direction of QGIS. You can subscribe to this list at: <http://lists.osgeo.org/mailman/listinfo/qgis-psc>

You are welcome to subscribe to any of the lists. Please remember to contribute to the list by answering questions and sharing your experiences. Note that the qgis-commit and qgis-trac lists are designed for notification only and are not meant for user postings.

21.2 IRC

We also maintain a presence on IRC - visit us by joining the #qgis channel on irc.freenode.net. Please wait for a response to your question, as many folks on the channel are doing other things and it may take a while for them to notice your question. If you missed a discussion on IRC, not a problem! We log all discussion, so you can easily catch up. Just go to <http://qgis.org/irclogs> and read the IRC-logs.

Commercial support for QGIS is also available. Check the website <http://qgis.org/en/commercial-support.html> for more information.

21.3 BugTracker

While the qgis-users mailing list is useful for general 'How do I do XYZ in QGIS?'-type questions, you may wish to notify us about bugs in QGIS. You can submit bug reports using the QGIS bug tracker at <http://hub.qgis.org/projects/quantum-gis/issues>. When creating a new ticket for a bug, please provide an email address where we can contact you for additional information.

Please bear in mind that your bug may not always enjoy the priority you might hope for (depending on its severity). Some bugs may require significant developer effort to remedy, and the manpower is not always available for this.

Le richieste per nuove caratteristiche possono essere sottoposte tramite lo stesso sistema di segnalazioni usato per i bug. Assicuratevi di aver prima selezionato il tipo `Feature`.

If you have found a bug and fixed it yourself, you can submit this patch also. Again, the lovely redmine ticketsystem at <http://hub.qgis.org/wiki/quantum-gis/issues> has this type as well. Check the `Patch supplied` checkbox and attach your patch before submitting your bug. One of the developers will review it and apply it to QGIS. Please don't be alarmed if your patch is not applied straight away – developers may be tied up with other commitments.

21.4 Blog

The QGIS community also runs a weblog at <http://planet.qgis.org/planet/>, which has some interesting articles for users and developers as well provided by other blogs in the community. You are invited to contribute your own QGIS blog!

21.5 Plugins

The website <http://plugins.qgis.org> provides the official QGIS plugins web portal. Here, you find a list of all stable and experimental QGIS plugins available via the 'Official QGIS Plugin Repository'.

21.6 Wiki

Lastly, we maintain a WIKI web site at <http://hub.qgis.org/projects/quantum-gis/wiki> where you can find a variety of useful information relating to QGIS development, release plans, links to download sites, message-translation hints and more. Check it out, there are some goodies inside!

22.1 GNU General Public License

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc. 59 Temple Place - Suite 330, Boston, MA 02111-1307, USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software—to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation’s software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author’s protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors’ reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone’s free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow. **TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION**

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The “Program”, below, refers to

any such program or work, and a “work based on the Program” means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term “modification”.) Each licensee is addressed as “you”.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program’s source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:
 - (a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.
 - (b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.
 - (c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:
 - (a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
 - (b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

- (c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.
5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.
6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.
7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.
9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and “any later version”, you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.
12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

QGIS Qt exception for GPL

In addition, as a special exception, the QGIS Development Team gives permission to link the code of this program with the Qt library, including but not limited to the following versions (both free and commercial): Qt/Non-commercial Windows, Qt/Windows, Qt/X11, Qt/Mac, and Qt/Embedded (or with modified versions of Qt that use the same license as Qt), and distribute linked combinations including the two. You must obey the GNU General Public License in all respects for all of the code used other than Qt. If you modify this file, you may extend this exception to your version of the file, but you are not obligated to do so. If you do not wish to do so, delete this exception statement from your version.

22.2 GNU Free Documentation License

Version 1.3, 3 November 2008

Copyright 2000, 2001, 2002, 2007, 2008 Free Software Foundation, Inc

<<http://fsf.org/>>

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The purpose of this License is to make a manual, textbook, or other functional and useful document “free” in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of “copyleft”, which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The **Document**, below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as “**you**”. You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A “**Modified Version**” of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A “**Secondary Section**” is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document’s overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The “**Invariant Sections**” are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The “**Cover Texts**” are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A “**Transparent**” copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not “Transparent” is called **Opaque**.

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The “**Title Page**” means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, “Title Page” means the text near the most prominent appearance of the work’s title, preceding the beginning of the body of the text.

The “**publisher**” means any person or entity that distributes copies of the Document to the public.

A section “**Entitled XYZ**” means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as “**Acknowledgements**”, “**Dedications**”, “**Endorsements**”, or “**History**”.)

To “**Preserve the Title**” of such a section when you modify the Document means that it remains a section “Entitled XYZ” according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document’s license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

1. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
2. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.
3. State on the Title page the name of the publisher of the Modified Version, as the publisher.
4. Preserve all the copyright notices of the Document.
5. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
6. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.

7. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
8. Include an unaltered copy of this License.
9. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
10. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
11. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
12. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
13. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.
14. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.
15. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements".

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an “aggregate” if the copyright resulting from the compilation is not used to limit the legal rights of the compilation’s users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document’s Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled “Acknowledgements”, “Dedications”, or “History”, the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, or distribute it is void, and will automatically terminate your rights under this License.

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, receipt of a copy of some or all of the same material does not give you any rights to use it.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <http://www.gnu.org/copyleft/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License “or any later version” applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation. If the Document specifies that a

proxy can decide which future versions of this License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Document.

11. RELICENSING

“Massive Multiauthor Collaboration Site” (or “MMC Site”) means any World Wide Web server that publishes copyrightable works and also provides prominent facilities for anybody to edit those works. A public wiki that anybody can edit is an example of such a server. A “Massive Multiauthor Collaboration” (or “MMC”) contained in the site means any set of copyrightable works thus published on the MMC site.

“CC-BY-SA” means the Creative Commons Attribution-Share Alike 3.0 license published by Creative Commons Corporation, a not-for-profit corporation with a principal place of business in San Francisco, California, as well as future copyleft versions of that license published by that same organization.

“Incorporate” means to publish or republish a Document, in whole or in part, as part of another Document.

An MMC is “eligible for relicensing” if it is licensed under this License, and if all works that were first published under this License somewhere other than this MMC, and subsequently incorporated in whole or in part into the MMC, (1) had no cover texts or invariant sections, and (2) were thus incorporated prior to November 1, 2008.

The operator of an MMC Site may republish an MMC contained in the site under CC-BY-SA on the same site at any time before August 1, 2009, provided the MMC is eligible for relicensing.

ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

Copyright © YEAR YOUR NAME. Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled “GNU Free Documentation License”.

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the “with ... Texts.” line with this:

with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.

Letteratura e riferimenti web

GDAL-SOFTWARE-SUITE. Geospatial data abstraction library. <http://www.gdal.org>, 2013.

GRASS-PROJECT. Geographic resource analysis support system. <http://grass.osgeo.org> , 2013.

NETELER, M., AND MITASOVA, H. Open source gis: A grass gis approach, 2008.

OGR-SOFTWARE-SUITE. Geospatial data abstraction library. <http://www.gdal.org/ogr> , 2013.

OPEN-GEOSPATIAL-CONSORTIUM. Web map service (1.1.1) implementation specification. <http://portal.opengeospatial.org>, 2002.

OPEN-GEOSPATIAL-CONSORTIUM. Web map service (1.3.0) implementation specification. <http://portal.opengeospatial.org>, 2004.

POSTGIS-PROJECT. Spatial support for postgresql. <http://postgis.refrations.net/> , 2013.

-
- %%, 102
- a cavallo della linea di longitudine 180 degrees, 74
- Aggiornamento della visualizzazione durante il disegno, 35
- apache, 156
- apache2, 156
- Arc/Info_ASCII_Grid, 135
- Arc/Info_Binary_Grid, 135
- ArcInfo_Binary_Coverage, 68
- Arresto della visualizzazione, 34
- Atlas_Generation, 658
- attribute table, 126
- Attribute_Actions, 102
- Attribute_Table, 650
- Attribute_Table_Selection, 126
- Avoid_Intersections_Of_Polygons, 116
- Azioni, 102
- Barra degli strumenti, 28
- Browse_Maps, 63
- calcolatore della scala, 32
- Calculator_Field, 132
- caricare uno shapefile, 66
- CAT, 147
- Categorized_Renderer, 83
- CGI, 156
- Colliding_labels, 91
- Color_interpolation, 140
- color_Ramp, 79
- colorBrewer, 79
- Common_Gateway_Interface, 156
- Compose_Maps, 629
- Composer_Manager, 663
- Composer_Template, 630
- compositore di stampa stampa veloce, 20
- Contrast_enhancement, 138
- Coordinate_Reference_System, 57, 151
- Create_Maps, 629
- Create_New_Layers, 124
- CRS, 151
- CSV, 68, 119
- Current_Edits, 117
- Custom_color_Ramp, 79
- Custom_CRS, 60
- Datum_transformation, 61
- DB_Manager, 75
- Debian_Squeeze, 156
- default_CRS, 57
- definire un'azione, 102
- Derived_Fields, 132
- Digitalizzazione, 116
- Discreto, 140
- Displacement_plugin, 86
- documentazione, 7
- editing, 114
- Elements_Alignment, 657
- EPSG, 57
- Equal_Interval, 84
- Erdas_Imagine, 135
- esempi di azioni, 103
- ESRI, 65
- European_Petroleum_Search_Group, 57
- Export_as_image, 662
- Export_as_PDF, 662
- Export_as_SVG, 662
- Expressions, 108
- FastCGI, 156
- Field_Calculator, 132
- Field_Calculator_Functions, 109
- finestra principale, 21
- GDAL, 135
- GeoTIFF, 135
- GeoTiff, 135
- GiST (indice dell'albero generalizzato di ricerca, Generalized Search Tree), 73
- GML, 147
- GNU General Public License, 717
- Gradient_color_Ramp, 79
- Graduated_Renderer, 84
- GRASS, 173, *vedi* Creating new vectors;editing;creating a new layer
- attribute linkage, 178
- attribute storage, 178
- category settings, 179
- digitizing tools, 178
- display results, 185
-

- region, 182
- region display, 182
- region editing, 182
- snapping tolerance, 180
- symbology settings, 180
- table editing, 180
- toolbox, 186
- GRASS toolbox, 182
 - Browser, 188
 - customize, 189
- GRASS vector data model, 177
- Guide contestuali, 33
- HTML_Frame, 655
- IGNF, 57
- Import_Maps, 63
- Informazione elementi, 37
- Institut_Geographique_National_de_France, 57
- InteProxy, 154
- Inverted_Polygon_Renderer, 86
- Istogramma, 142
- join, 105
- Layout_Maps, 629
- legenda, 28
- license document, 717
- loading_raster, 135
- Map_Legend, 642
- Map_Navigation, 115
- Map_Template, 630
- MapInfo, 68
- merge attributes of features, 123
- Merge_Attributes_of_Selected_Features, 123
- Merge_Selected_Features, 123
- Metadati, 143
- misurazioni, 35
 - angoli, 35
 - aree, 35
 - lunghezza linee, 35
- Modifica topologica, 116
- MSSQL Spatial, 75
- Multi_Band_Raster, 137
- multipolygon, 122
- Natural_Breaks_(Jenks), 84
- New_GPX_Layer, 124, 125
- New_Shapefile_Layer, 124
- New_SpatialLite_Layer, 124
- New_Spatiallite_Layer, 124
- Node_Tool, 118
- Nodes, 118
- Non_Spatial_Attribute_Tables, 128
- note, 40
- OGC, 147
- OGR, 65
- OGR Simple Feature Library, 65
- ogr2ogr, 73
- Open_Geospatial_Consortium, 147
- OpenStreetMap, 70
- opzioni linea di comando, 17
- Oracle Spatial, 75
- OSM, 70
- Output salva come immagine, 20
- Panoramica mappa, 46
- pgsql2shp, 72
- Picture_database, 641
- Piramidi, 142
- plugin, 665
- Point_Displacement_Renderer, 86
- PostGIS, 70
- PostGIS spatial index, 73
- PostgreSQL, 70
- Pretty_Breaks, 84
- print_composer
 - tools, 629
- Printing
 - Export_Map, 662
- progetti nidificati, 43
- Proiezioni, 57
- Proj.4, 60
- Proj4, 59
- Proj4_text, 59
- Proxy, 149
- proxy-server, 149
- QGIS_mapserver, 155
- QGIS_Server, 156
- QSpatialLite, 75
- qualità di visualizzazione, 35
- Quantile, 84
- Query_Builder, 130
- Raster, 135
- Raster_Calculator, 144
- Relations, 128
- Renderer_Categorized, 83
- Renderer_Graduated, 84
- Renderer_Point_Displacement, 86
- Renderer_Single_Symbol, 83
- Rendering_Mode, 634
- Rendering_Rule-based, 86
- Reticolo
 - Grids
 - Map_Grid, 637
- Revert_Layout_Actions, 658
- ring polygons, 121
- Rotate_Point_symbols, 123
- Rotated_North_Arrow, 641
- Rule-based_Rendering, 86
- Scala, 34
- Scalebar
 - Map_Scalebar, 646
- Scheda Mappa colore, 140

Scorciatoie da tastiera, 33
 Search_Radius, 115
 Secured_OGC_Authentication, 154
 segnalibri, 42
 segnalibri geospaziali
 visualizzare i segnalibri, 42
 Seleziona usando un'espressione, 132
 SFS, 147
 Shapefile, 65
 Shapefile_to_Postgis_Import_Tool, 708
 Shared_Polygon_Boundaries, 116
 shp2pgsql, 72
 Simbologia, 90, 137
 Single_Band_Raster, 137
 Single_Symbol_Renderer, 83
 SLD, 156
 SLD/SE, 156
 Snapping, 114
 Snapping_On_Intersections, 116
 Snapping_Tolerance, 114
 Spatialite, 74
 Spatialite_Manager, 75
 SPIT, 708
 Split_Features, 123
 Sposta, 115
 spostamento con i tasti direzionali, 31
 SQLite, 74
 SR, 57
 SRS, 151
 ST_Shift_Longitude, 74
 Strumenti di Analisi, 683
 Strumenti di Ricerca, 684
 Strumento per la georeferenziazione, 689

 Testo delimitato, 68
 Three_Band_Color_Raster, 137
 Tiger_Format, 68
 Toggle Editing, 117
 Trasparenza, 141

 UK_National_Transfer_Format, 68
 US_Census_Bureau, 68

 Vertex, 118
 Vertices, 118
 visibilità layer, 29
 Visualizzazione, 33
 Visualizzazione dipendente dalla scala, 34
 Voce di menu, 22

 WCS, 147, 155
 Web Coverage Service, 155
 WFS, 147, 155
 WFS-T, 155
 WFS_Transactional, 155
 WKT, 57, 119
 WMS, 147
 WMS-C, 152
 WMS_1.3.0, 155

 WMS_client, 147
 WMS_identify, 152
 WMS_layer_transparency, 151
 WMS_metadata, 153
 WMS_properties, 153
 WMS_tiles, 152
 WMTS, 152
 WMTS_client, 147
 Work_with_Attribute_Table, 126

 zoom con la rotellina del mouse, 31
 Zoom_In Zoom_Out, 115