Abstract tratto da www.darioflaccovio.it - Tutti i diritti riservati

GIS

Scheda sul sito ›

Luca Casagrande ~ Paolo Cavallini ~ Alessandro Frigeri
Alessandro Furieri ~ Ivan Marchesini ~ Markus Neteler

GIS Open Source

GRASS GIS, Quantum GIS e SpatiaLite

Elementi di software libero applicato al territorio

Dario Flaccovio Editore

L. Casagrande, P. Cavallini, A. Frigeri, A. Furieri, I. Marchesini, M. Neteler
GIS OPEN SOURCE

ISBN 978-88-579-0149-7

© 2012 by Dario Flaccovio Editore s.r.l. - tel. 0916700686
www.darioflaccovio.it info@darioflaccovio.it

Prima edizione: ottobre 2012

GIS open source, GRASS GIS, Quantum GIS e SpatiaLite : elementi di software libero applicato al territorio / Luca Casagrande...[et al.] - Palermo : D. Flaccovio, 2012.
ISBN 978-88-579-0149-7
1. Cartografia – Impiego [degli] Elaboratori. I, Casagrande, Luca <1978->.
526.0285 CDD-22 SBN Pal0247032

CIP - Biblioteca centrale della Regione siciliana "Alberto Bombace"

Stampa: Tipografia Priulla, Palermo, settembre 2012

RINGRAZIAMENTI

Si ringrazia Maurizio Napolitano (Fondazione Bruno Kessler, Trento) per le utili discussioni sui dati liberi e le loro licenze, Gisella Speranza (Roma) per la revisione del capitolo.
Armando Valle per la disponibilità nella verifica della procedura di installazione di GRASS su Microsoft Windows.
Gli autori ringraziano Mauro Rossi (CNR IRPI, Perugia) per aver fornito lo spunto per l'esempio sul contatto stratigrafico riportato nel capitolo 7.
Hanno collaborato alla stesura di questo libro: Pierluigi De Rosa, Luca Delucchi, Franca Iacobellis, Annalisa Minelli, Paola Napolitano, Duccio Rocchini

Tutte le figure contenute in questo libro sono liberamente scaricabili nel sito www.darioflaccovio.it nella pagina dedicata all'opera

Nomi e marchi citati sono generalmente depositati o registrati dalle rispettive case produttrici.
L'editore dichiara la propria disponibilità ad adempiere agli obblighi di legge nei confronti degli aventi diritto sulle opere riprodotte.
La fotocopiatura dei libri è un reato.
Le fotocopie per uso personale del lettore possono essere effettuate nei limiti del 15% di ciascun volume/fascicolo di periodico dietro pagamento alla SIAE del compenso previsto dall'art. 68, commi 4 e 5, della legge 22 aprile 1941 n. 633. Le riproduzioni effettuate per finalità di carattere professionale, economico o commerciale o comunque per uso diverso da quello personale possono essere effettuate solo a seguito di specifica autorizzazione rilasciata dagli aventi diritto/dall'editore.

INDICE

Premessa

1. Obiettivi e prerequisiti *di Ivan Marchesini*
1.1. Obiettivi ... pag. 11
1.2. Prerequisiti ... » 12

2. Software libero, licenze e dati geografici *di Alessandro Frigeri*
2.1. Il Software Free Open Source o Software Libero ... » 13
2.2. I dati geografici digitali e il software ... » 14
2.3. I formati dei dati GIS ... » 15
2.4. Le licenze sui dati e l'accessibilità ai dati ... » 16

3. GRASS GIS & Quantum GIS: caratteristiche e campi di applicazione *di Ivan Marchesini*
3.1. Cenni storici ... » 19
3.1.1. GRASS GIS ... » 19
3.1.2. Quantum GIS ... » 21
3.2. Differenze e potenzialità ... » 22
3.3. Bibliografia ... » 24

4. GIS, proiezioni e sistemi di riferimento *di Alessandro Frigeri*
4.1. Introduzione ... » 25
4.2. Geografia e cartografia ... » 25
4.2.1. Definizioni ... » 25
4.2.2. La forma della Terra, gli ellissoidi ... » 26
4.2.3. Proiezioni cartografiche ... » 27
4.2.4. I sistemi di coordinate ... » 27
4.2.5. I sistemi di riferimento delle coordinate ... » 28
4.2.6. I registri di parametri geodetici ... » 28
4.3. Tipi comuni di sistemi cartografici ... » 29
4.3.1. Sistemi cartografici usati in Italia ... » 29
4.3.2. Datum utilizzati in Italia ... » 29
4.3.3. Il sistema Gauss-Boaga ... » 30
4.3.4. I codici EPSG dei sistemi cartografici utilizzati in Italia ... » 30
4.3.5. La proiezione di Google o proiezione Web ... » 31

5. Quantum GIS *di Luca Casagrande e Paolo Cavallini (coautori: Pierluigi De Rosa, Paola Napolitano, Franca Iacobellis)*
5.1. Le prime operazioni per lavorare con Quantum GIS ... » 33
5.2. Installazione e personalizzazione dell'interfaccia ... » 33
5.2.1. Installazione nei sistemi Linux ... » 33
5.2.2. Installazione in ambiente Mac OS X ... » 34
5.2.3. Installazione in ambiente MS Windows ... » 35
5.2.4. L'interfaccia di Quantum GIS ... » 35
5.2.5. Localizzazione e personalizzazione della barra degli strumenti ... » 35
5.2.6. Personalizzazione delle barre degli strumenti ... » 36
5.2.7. Altre personalizzazioni ... » 36
5.3. Nuovo progetto e visualizzazione dei dati ... » 37
5.3.1. La gestione dei sistemi di riferimento ... » 37
5.3.2. Visualizzazione di dati vettoriali ... » 40
5.3.3. Visualizzazione di dati raster ... » 41
5.3.4. Visualizzazione di dati provenienti da WMS ... » 42
5.3.5. Visualizzazione di dati contenuti in database PostGIS ... » 43
5.3.6. Ottimizzare la visualizzazione dei dati caricati ... » 44
5.4. Gestione degli stili di rappresentazione ... » 45
5.4.1. Stili di dati vettoriali ... » 45
5.4.2. Stili di dati raster ... » 47

5.4.3. Un esempio di applicazione con i dati della Regione Piemonte ... » 47
5.5. Creazione e modifica di nuovi dati vettoriali ... » 49
5.5.1. Impostazioni ... » 49
5.5.2. Modifica di un dato esistente ... » 50
5.5.3. Creazione di un nuovo dato vettoriale ... » 51
5.6. Il sistema di stampa e la preparazione del layout ... » 53
5.6.1. Il gestore di stampa ... » 54
5.6.2. Come viene creato un layout di stampa ... » 54
5.6.3. Proprietà generali degli oggetti ... » 55
5.6.4. Gli elementi del layout ... » 55
5.6.4.1. Informazioni generali ... » 55
5.6.4.2. Mappa ... » 56
5.6.4.3. Legenda ... » 58
5.6.4.4. Barra di scala ... » 58
5.6.4.5. Testo e immagini ... » 59
5.6.5. Creazione dell'elaborato ... » 59
5.7. Funzionalità aggiuntive mediante plugin ... » 60
5.7.1. Installazione di nuovi plugin ... » 60
5.7.2. Plugin Aggiungi layer di testo delimitato ... » 60
5.7.3. Plugin Georeferenziatore ... » 62
5.7.4. Plugin Strumenti GPS ... » 65
5.7.5. Plugin Convertitore AutoCAD DXF ... » 67
5.7.6. Plugin Interpolatore ... » 67
5.7.7. Plugin – Utilizzo di cartografia proveniente dal servizio OpenStreetMap ... » 69

6. GRASS GIS *di Alessandro Frigeri e Markus Neteler (coautori: Luca Delucchi, Duccio Rocchini)*
6.1. Installazione di GRASS GIS ... » 71
6.1.1. Cenni introduttivi ... » 71
6.1.2. Installazione su sistemi GNU/Linux ... » 71
6.1.3. Installazione di pacchetti RPM ... » 72
6.1.4. Installazione su Red Hat Enterprise Linux, CentOS e Scientific Linux ... » 72
6.1.5. Installazione di GRASS GIS su Opensuse ... » 73
6.1.6. Installazione di GRASS GIS su Mandriva ... » 73
6.1.7. Installazione su Debian e Ubuntu ... » 74
6.1.8. Installazione di GRASS GIS su Mac OS X ... » 75
6.1.8.1. Scaricamento dei pacchetti dalla pagina web ... » 75
6.1.8.2. Installazione dei pacchetti ... » 76
6.1.9. Installazione di GRASS GIS su Microsoft Windows ... » 77
6.2. Creazione di un nuovo progetto: il mapset e la location ... » 79
6.2.1. Gisdbase, Location, Mapset ... » 79
6.2.2. Creare una nuova location ... » 80
6.2.3. Creare una nuova location attraverso interfaccia grafica ... » 81
6.2.3.1. Selezionare il sistema di coordinate da una lista ... » 81
6.2.3.2. Selezionare il codice EPSG del sistema di riferimento spaziale ... » 82
6.2.3.3. Leggere proiezione e datum da un file georeferenziato ... » 83
6.2.3.4. Leggere proiezione e datum da un file WKT o PRJ ... » 83
6.2.3.5. Specificare proiezione e datum utilizzando parametri propri di PROJ.4 ... » 84
6.2.3.6. Creare un sistema arbitrario di coordinate non terrestri (XY) ... » 84
6.2.4. Creare una nuova location attraverso interfaccia testuale ... » 84
6.2.5. Il concetto della regione computazionale ... » 86
6.3. Gestione di mappe raster ... » 87
6.3.1. L'importanza della regione computazionale ... » 87
6.3.2. Importare/linkare mappe raster ... » 88
6.3.3. Operazioni di base ... » 90
6.3.4. Interrogare mappe raster ... » 94
6.3.5. Mask ... » 95
6.3.6. Raster map algebra ... » 96

6.3.7. Analisi raster ... » 99
6.3.8. Visualizzazione 3D NVIZ ... » 104
6.3.9. Raster 3D ... » 104
6.3.10. Export di mappe raster ... » 105
6.3.11. Extra: modellizzatore grafico ... » 106
6.3.12. Bibliografia ... » 106
6.4. Gestione di dati vettoriali ... » 107
6.4.1. Principi di mappe vettoriali e attributi alfanumerici ... » 107
6.4.2. Importare/collegare mappe vettoriali ... » 108
6.4.3. Operazioni di base (mappe tematiche, topologia, tipologia) ... » 109
6.4.4. Interrogare livelli vettoriali ... » 112
6.4.5. Collegamento tra geometria e attributi ... » 113
6.4.6. Calcolo sulla geometria ... » 113
6.4.7. Vettori 2D e 3D ... » 115
6.4.8. Vector network analysis ... » 117
6.4.9. Gestione degli attributi ... » 118
6.4.10. Digitalizzazione ... » 119
6.4.11. Export di dati vettoriali ... » 122
6.4.12. Extra: compositore grafico ... » 123
6.5. Bibliografia ... » 125

7. Strumenti di analisi in GRASS GIS *di Ivan Marchesini (coautore: Annalisa Minelli)*
7.1. Cenni introduttivi ... » 127
7.2. Applicazioni geomorfologiche ... » 129
7.2.1. Elementi geomorfologici ... » 129
7.2.2. Creazione di profili topografici ... » 133
7.3. Applicazioni idrologiche ... » 136
7.3.1. I poligoni di Thiessen per dati pluviometrici e le isoiete ... » 136
7.3.2. Estrazione, a partire da un DEM, del bacino e del reticolo idrografico ... » 143
7.4. Applicazioni paesaggistiche ... » 150
7.4.1. Visibilità di un sito ... » 150
7.4.2. Calcolo dei volumi di un possibile sito estrattivo ... » 154
7.5. Applicazioni geologiche e idrogeologiche ... » 159
7.5.1. Tracciare un contato tra formazioni geologiche ... » 159
7.5.2. Pericolosità sismica e norme tecniche per le costruzioni ... » 162
7.5.3. Idrogeologia e simulazione di emungimenti ... » 166
7.6. Bibliografia ... » 170

8. Spatial DBMS: SQLite, SpatiaLite e RasterLite *di Alessandro Furieri*
8.1. DBMS e Spatial DBMS ... » 171
8.2. Struttura di un DBMS ... » 171
8.2.1. Sistemi di archiviazione basati su file ... » 171
8.2.2. Principi basilari dei DBMS ... » 172
8.2.3. Architettura client-server ... » 173
8.2.4. Architettura personal DBMS ... » 173
8.3. Il linguaggio SQLite ... » 174
8.3.1. Caratteristiche generali ... » 174
8.3.2. Assestamento: a che punto siamo ... » 177
8.4. Gli strumenti a supporto dei DBMS ... » 177
8.4.1. Strumenti di FRONT-END ... » 177
8.4.2. Connettori ... » 177
8.5. Gli Spatial DBMS e lo Spatial SQLite ... » 178
8.5.1. Principi fondamentali ... » 178
8.5.2. Lo standard OGC-SFS ... » 178
8.5.3. Il tipo dato geometry ... » 179
8.6. Spatial DBMS disponibili sul mercato ... » 182
8.6.1. Offerta complessiva ... » 182

8.6.2. MySQL Spatial ... » 182
8.6.3. PostgreSQL + PostGIS ... » 183
8.6.4. SQLite + SpatiaLite ... » 183
8.6.5. SQLite + SpatiaLite in dettaglio ... » 184
8.7. Prima esercitazione pratica: manipolazione di Shapefiles ... » 185
8.7.1. Descrizione del problema ... » 185
8.7.2. Importazione dello Shapefile ... » 185
8.7.3. Verifica dei dati importati da Shapefile ... » 187
8.7.4. Verifica dei dati importati da DBF ... » 189
8.7.5. Creazione della tavola definitiva ... » 190
8.7.6. Alimentazione della tavola definitiva ... » 192
8.7.7. Esportazione dello shapefile rielaborato ... » 193
8.7.8. Assestamento al termine dell'esercitazione ... » 194
8.8. Seconda esercitazione pratica: fusione di poligoni ... » 194
8.8.1. Descrizione del problema ... » 194
8.8.2. Verifica preliminare ... » 194
8.8.3. Creazione della tavola USL ... » 196
8.8.4. Verifica dei risultati ... » 197
8.8.5. Le tavole dei metadati spaziali ... » 198
8.9. Terza esercitazione pratica: analisi spaziale dei dati ... » 199
8.9.1. Descrizione del problema ... » 199
8.9.2. Stazioni ferroviarie e comuni ... » 199
8.9.3. Comuni attraversati da linee ferroviarie ... » 200
8.9.4. Conclusioni ... » 201
8.10. RasterLite e RasterLite2 ... » 201
8.10.1. Spatial DBMS e dati Raster ... » 201
8.10.2. RasterLite ... » 201
8.10.3. RasterLite2 ... » 202
8.10.4. Obiettivi di RasterLite/RasterLite2 ... » 202
8.11. SpatiaLite e Quantum GIS ... » 203
8.11.1. Connettere il database SpatiaLite ... » 203
8.11.2. Impostare il sistema di riferimento spaziale ... » 204
8.11.3. Interrogare gli elementi ... » 205
8.11.4. Consultare la tabella degli attributi ... » 206
8.11.5. I plugin di Quantum GIS per la gestione dei DBMS ... » 207
8.12. SQLite e GRASS GIS ... » 207

9. Ottenere aiuto e supporto *di Paolo Cavallini*
9.1. Cenni introduttivi ... » 209
9.2. Le comunità ... » 209
9.3. Come porre una domanda in mailing list ... » 210
9.4. Come guadagnarsi una buona reputazione ... » 212
9.5. Divulgazione con social media ... » 212
9.6. Segnalare malfunzionamenti e miglioramenti ... » 213
9.7. Supporto commerciale da aziende e professionisti ... » 213
9.8. Ulteriori considerazioni ... » 214

10. Per approfondire *di Ivan Marchesini*
10.1. Premessa ... » 215
10.2. Quantum GIS ... » 215
10.3. GRASS GIS ... » 218
10.4. Bibliografia ... » 220

Premessa

L'informazione geografica, con il diffondersi delle tecnologie legate ai navigatori satellitari, dalla cartografia web agli smartphone, è ormai entrata a far parte della vita quotidiana di tutti.
Ingegneri, architetti, urbanisti, geologi, naturalisti, biologi, forestali si confrontano quotidianamente con la necessità di gestire e produrre cartografia numerica georiferita e/o di integrare i propri dati con quelli che possono ottenere da enti pubblici, privati, servizi web, cartografia partecipata.
Capire e conoscere gli strumenti utili per gestire informazioni territoriali è quindi diventata una necessità per i professionisti e tecnici che operano sul territorio, ma anche per gli studenti di materie ambientali, urbanistiche o comunque legate alla pianificazione territoriale e alla protezione civile.
Spesso però l'accesso a questi strumenti è ostacolato dai costi elevati delle licenze software e dalla scarsità di percorsi formativi adeguati alle esigenze di chi si avvicina per la prima volta a queste tematiche.
Il software geografico Libero ed Open Source e la sua comunità italiana e internazionale rappresentano la soluzione che consente all'utente di superare gli ostacoli sopra citati.
Il presente testo ha lo scopo di introdurre l'acquirente al mondo dei software GIS e in particolare all'uso di alcuni software geografici Liberi ed Open Source. Il testo non si limita a descrivere alcune funzionalità dei software ma fornisce nozioni di carattere generale sulla cartografia numerica e sui sistemi di riferimento.
Gli autori, tutti esponenti di rilevo della comunità nazionale e, in molti casi, anche internazionale dei GIS Open Source, svolgono attività di formazione, professionale e di ricerca scientifica e sono fortemente motivati dalla comune attività a sostegno del Sofware Libero. L'originalità del testo è quindi da ricercarsi nella possibilità, da parte dell'utente, di avvalersi immediatamente sia degli strumenti software che delle numerose risorse documentali accessibili via internet e descritte nel testo.
Il libro, composto di 10 capitoli, è caratterizzato da una prima parte in cui si descrivono i sistemi di riferimento e si assiste l'utente nell'installazione dei software su sistemi operativi Microsoft, Apple, GNU/Linux.
Successivamente sono introdotte le funzionalità più importanti di software ben noti come Quantum GIS o GRASS GIS e si descrive l'approccio alla gestione del dato geografico mediante motori di database relazionali (SpatiaLite).
Nei capitoli successivi vengono sia presentati alcuni esempi e applicazioni nel settore del paesaggio, della geomorfologia, dell'idrologia e della geologia sia descritte le risorse online, mediante il quale l'utente può ottenere gratuitamente aiuto e supporto.
Il libro si chiude con alcuni accenni a funzionalità avanzate.

Gli autori

1. Obiettivi e prerequisiti

di Ivan Marchesini

1.1. Obiettivi

Obiettivo principale di questo libro è quello di favorire e sostenere l'utilizzo dei GIS Open Source in Italia. Infatti, sebbene l'Italia svolga un ruolo fondamentale nella comunità mondiale del software GIS Open Source, l'adozione di questi strumenti da parte di privati e pubbliche amministrazioni è spesso ostacolato dalla carenza di testi di riferimento in lingua italiana.
Questo testo, tuttavia, pur avendo la finalità di introdurre il lettore al software geografico libero e Open Source, affronta alcune tematiche e nozioni di fondo (formati dei dati, le licenze d'uso e i sistemi di riferimento) valide anche laddove si utilizzi software proprietario.
Tra gli obiettivi del libro c'è anche la volontà, da parte degli autori, di dimostrare l'affidabilità, la robustezza e la qualità dei software Open Source per la gestione dei dati territoriali. La presente opera è infatti interamente realizzata da soci e fondatori dell'Associazione Italiana per l'Informazione Geografica Libera: GFOSS.it (www.gfoss.it) che ormai da alcuni anni si occupa di promuovere il software geografico Open Source oltre che di incentivare la pubblicazione e la diffusione dei dati geografici da parte delle pubbliche amministrazioni.
Il lettore, grazie a questo libro, sarà in grado di installare i software Quantum GIS (QGIS) e GRASS sul proprio computer (su sistemi operativi GNU-Linux, Mac OS X, e MS Windows).
Grazie al capitolo dedicato a come ottenere aiuto e supporto, il lettore potrà usufruire delle enormi risorse offerte dalla comunità mondiale del software geografico libero e mantenere aggiornato il software (il software Open Source ha frequenze di rilascio e di aggiornamento decisamente elevate).
Al termine della lettura del libro l'utente sarà inoltre in grado di:

- visualizzare, modificare e stampare diversi tipi di dato (anche forniti da servizi OGC – Open Geospatial Consortium) utilizzando QGIS;
- utilizzare alcuni particolari plugin di QGIS e capire come ottenerne e usarne altri;
- importare, editare e analizzare dati utilizzando GRASS;
- usare modelli digitali del terreno per ottenere mappe derivate e visualizzazioni tridimensionali con GRASS.

È il caso di specificare che non rientra tra gli obiettivi di questo lavoro l'approfondi-

mento di tematiche specifiche e/o l'analisi di tutte le potenzialità di questi due software, perché non sarebbe possibile farlo nell'ambito di una sola pubblicazione.
Non è tra le finalità del libro dimostrare che, nel mondo dei software geografici Open Source, QGIS e GRASS rappresentino la miglior soluzione per l'utente. Il mondo dei GFOSS (Geospatial Free and Open Source Software) è vasto e articolato e la scelta di questi due software nasce sia da ragioni di carattere storico (GRASS è il primo GIS Open Source a essersi affermato su scala mondiale) sia da valutazioni sulla robustezza dei progetti: infatti GRASS e QGIS sono gli unici due software GIS lato desktop pienamente riconosciuti da OSGeo (Open Source Geospatial Foundation, una fondazione internazionale che si occupa di "certificare" la qualità dei progetti di software Open Source per la gestione del dato territoriale, di promuoverne lo sviluppo collaborativo e la diffusione).
Numerosi altri progetti GIS Open Source sono però disponibili per gli utenti (ad esempio il progetto gvSIG, che si trova in fase di approvazione da parte di OSGeo) e si tratta spesso di soluzioni affidabili e potenti. In particolare, si è deciso di dedicare un capitolo del testo al progetto SpatiaLite, l'estensione spaziale per il DBMS (DataBase Management System) SQLite. Le ragioni di questa scelta sono molteplici:

- la semplicità e leggerezza di questo DBMS e della sua estensione spaziale;
- l'ottima sinergia esistente tra SpatiaLite e QGIS e tra SQLite e GRASS;
- il fatto che SpatiaLite è di fatto il prodotto del lavoro di sviluppo di un italiano: l'autore del capitolo 8 del libro, Alessandro Furieri.

Nel corso dell'opera non si affronta inoltre la tematica, enorme, delle soluzioni Open Source esistenti per la pubblicazione e distribuzione di dati e metadati secondo standard internazionali ed europei. Per questi argomenti saranno eventualmente necessarie pubblicazioni specifiche.

1.2. Prerequisiti

La struttura e i contenuti del libro sono pensati in modo tale da non richiedere particolari competenze pregresse da parte del lettore.
Il libro è indirizzato a attuali e potenziali fruitori di dati territoriali e cartografie numeriche in genere.
È abbastanza chiaro che, per coloro che hanno già esperienza di uso di GIS in ambiente proprietario (non Open Source), il testo risulterà molto più immediato e scorrevole ma, la particolare attenzione posta a descrivere argomenti di base (già citati nel paragrafo sugli obiettivi), rende il testo utile e particolarmente efficace soprattutto per coloro che non hanno particolari conoscenze GIS.
All'utente è richiesto di avere padronanza con l'utilizzo dei computer e avere la possibilità e le autorizzazioni per installare i software su un computer (portatile, casa, ufficio).
Il software descritto è sempre disponibile online ed è quindi necessario che l'utente abbia la possibilità e le autorizzazioni per avere accesso alla rete internet e scaricarlo. Nei capitoli relativi all'installazione saranno indicate le procedure per installare il software su diversi sistemi operativi.
Assieme al libro sono forniti dati di esempio che sono utilizzati per le procedure descritte nei vari capitoli: in questo modo l'utente è messo nelle condizioni di replicare quanto letto nel testo.

2. Software libero, licenze e dati geografici

di Alessandro Frigeri

2.1. Il Software Free Open Source o Software Libero

"Il software è la sostanza di cui è fatto il XXI secolo.
Quello che l'acciaio è stato per l'economia del XX secolo oggi è rappresentato dal software.
È il tassello cruciale, il componente con cui è fatto tutto il resto.
E quando parlo di tutto il resto intendo, ovviamente, la libertà."

Queste sono le parole con cui il prof. Eben Moglen della Columbia Law School di New York ha aperto il suo intervento alla conferenza degli sviluppatori di software libero (FOSDEM) nel 2011 a Bruxelles.
Il software libero è attualmente conosciuto in ambito internazionale come *Free Open Source Software* (FOSS) o *Free Libre Open Source Software* (FLOSS), ed è il risultato di 30 anni di sviluppo, conferenze e progetti portati avanti da singoli sviluppatori, imprese, compagnie e agenzie internazionali, università e istituti di ricerca.
Mentre nel mondo anglosassone il termine *Free Software* può significare sia software libero sia software gratuito, nella lingua italiana non ci sono dubbi: il software libero implica che il suo utilizzo, il suo sviluppo e la sua distribuzione sono sottoposti a regole diverse da quelle del software non libero o software proprietario.
Il concetto di software libero non è nuovo in ambito informatico, infatti i primi computer in circolazione avevano un costo elevato e solo centri di ricerca avanzati potevano affrontarne l'acquisto e la manutenzione. Questi sistemi venivano gestiti da software che veniva normalmente distribuito senza costo e condiviso tra sviluppatori ed utenti. Il codice scritto dagli utenti era a sua volta disponibile per altri utenti che lo avessero voluto usare o migliorare.
Con la diffusione dei computer all'esterno degli ambiti di ricerca e sviluppo, tra la seconda metà degli anni '70 e la prima metà degli anni '80, venne proposto un modello differente dal precedente, che prevedeva che il software, ovvero l'insieme di istruzioni impartite a un computer per eseguire operazioni, avesse un proprietario che potesse vendere copie agli utenti che sottoscrivono una licenza che definisce i termini di utilizzo del software. Negli accordi stipulati all'acquisto del software tra il proprietario del software e l'utente finale vengono di solito specificate il numero di installazioni possibili e obblighi sul tipo di utilizzo e di distribuzione del software.
Questo modello, che implica un accesso esclusivo al codice sorgente dei programmi, ha

portato alla diffusione della pratica dell'esternalizzazione, o *outsourcing*, dello sviluppo del software al di fuori degli ambienti dove veniva normalmente utilizzato ed ottimizzato.
In questo periodo, programmatori e ricercatori interni a istituzioni che avevano sempre avuto modo di migliorare il software in collaborazione con i fornitori si trovarono in difficoltà. In caso di problemi, ogni collaborazione che prevedesse l'acceso al codice sorgente di un programma doveva essere fatta nei termini di un contratto di esclusività (Non Disclosure Agreement, NDA) che impediva allo sviluppatore di utilizzare il proprio contributo in altri ambiti.
In questo contesto, un ricercatore del Massachusetts Institute of Technology (MIT) di Boston decise di interrompere la propria carriera per dedicarsi a tempo pieno a un progetto che permettesse a tutti di avere maggiori libertà in termini di sviluppo e di fruizione del software. Così Richard M. Stallman, nel 1984, fondò la Free Software Foundation con lo scopo di sviluppare e promuovere un sistema operativo corredato di strumenti di sviluppo che permettesse di generare altro software. Al fine di rendere il software accessibile e sviluppabile liberamente, la Free Software Foundation in collaborazione con un pool di esperti di legge ideò un sistema di licenze che utilizza il copyright per tutelare sia gli sviluppatori che gli utenti finali.
All'interno del testo delle licenze pubblicate dalla Free Software Foundation (FSF) si trova la definizione di cosa si intende per programma che sia identificabile come software libero. Per software libero si intende quindi un programma che soddisfa i seguenti quattro punti:

- il software è utilizzabile per qualsiasi scopo in tutte le sue funzionalità;
- il software può essere copiato e distribuito, senza limiti nel numero di installazioni possibili;
- il codice sorgente del software deve essere accessibile;
- si possono apportare migliorie al software.

Il termine Open Source è nato verso la fine degli anni '90 del secolo scorso, ad opera di Bruce Perens. Insieme ad Eric Raymond, i due sviluppatori fondarono la Open Source Initiative (OSI) finalizzata soprattutto alla promozione e certificazione del software libero in campo industriale, in un momento storico in cui il termine anglosassone *free software* poteva creare diffidenza. A partire dal 2007, la diffusione del termine Free Open Source Software ingloba il lavoro e le visioni della FSF e della OSI.

2.2. I dati geografici digitali e il software

In generale qualsiasi tipo di software gestisce, in qualche sua funzionalità, dei dati. Ad esempio, un elaboratore di testo riceve dei dati di ingresso (le lettere che formano le parole) e restituisce una versione formattata del testo stesso. In questo senso, il documento digitale generato rappresenta l'insieme di dati in uscita dal programma.
Questo può ovviamente essere esteso ai programmi espressamente sviluppati per il calcolo, ad esempio delle simulazioni numeriche che generano le previsioni del tempo: i dati meteorologici di ingresso sono elaborati dal software che genera dei risultati, ovvero delle simulazioni su come si pensa sarà la dinamica del tempo meteorologico.

Nel caso dei sistemi informativi geografici la stretta correlazione tra software e dati è ancora più evidente: cosa sarebbe un GIS se non si disponesse delle mappe digitali da elaborare? E cosa si potrebbe fare con delle mappe digitali senza un sistema per poterle leggere?
È chiaro quindi, in generale, che software e dati sono strettamente legati e nel caso dei GIS questo rapporto è particolarmente stretto.

2.3. I formati dei dati GIS

Con *formato digitale di dati* si intende l'ordine logico con cui le sequenze di bit che formano l'informazione digitale vengono immagazzinati in un file o trasmessi attraverso una rete informatica.
Nel caso dei GIS di solito si parla di formati di dati che permettono di archiviare cartografia digitale su disco oppure trasmetterla in rete attraverso servizi di vario tipo (*Web Services*).
In generale, ogni formato GIS prevede l'archiviazione o la trasmissione di almeno due tipi di informazione: i dati e i metadati. I dati geografici rappresentano le informazioni strettamente geografiche: un dato vettoriale può ad esempio rappresentare una rete stradale oppure un dato raster può descrivere le temperature superficiali di una data regione. I metadati sono letteralmente "dati sui dati", ovvero informazioni ausiliarie che permettono di gestire al meglio l'informazione contenuta nei dati. Senza i metadati, ad esempio, non sapremmo in che sistema di riferimento ci troviamo e quindi non potremmo importare o esportare la nostra mappa in altri sistemi senza commettere errori. Inoltre, i metadati permettono di sapere chi ha prodotto la mappa e quando lo ha fatto.
I formati GIS raster attualmente più diffusi sono:

- GeoTIFF;
- JPEG2000.

GeoTIFF

Basato sullo storico Tagged Image File Format, fa uso di speciali etichette, o tag, per definire i metadati geografici. È un formato molto versatile in quanto eredita dal formato TIFF utili caratteristiche come, ad esempio, la compressione dei dati. La dimensione massima di un file può essere 4Gb, che può essere estesa tramite una recente implementazione TIFF chiamata BigTIFF; questo pone un problema di interoperabilità per file di dimensioni maggiori di 4Gb.

JPEG2000

È un formato aperto derivato dal noto formato sviluppato dal Joint Photographic Experts Group (JPEG). È un formato che vanta la registrazione come standard ISO/IEC 15444. La principale caratteristica del JPEG2000 è che la compressione dei dati avviene attraverso algoritmi di compressione basati su wavelet, particolarmente efficienti nel caso di immagini di remote sensing. Questo standard permette di usare codec esterni per la compressione/decompressione, esattamente come succede per i formati video, e attualmente l'unico codec disponibile come software libero è JasPer. JPEG2000 suppor-

ta metadati in formato XML, e quindi è possibile includere metadati geografici come ad esempio la proiezione utilizzata o la copertura geografica dei dati, in formato GML.

Per quanto riguarda i formati vettoriali, i più diffusi sono attualmente:

- ESRI Shapefile;
- Geographic Markup Language (GML);
- Keyhole Markup Language (KML).

ESRI SHAPEFILE

È il formato di interscambio della multinazionale del software geografico *Environmental Systems Research Institute* (ESRI). Seppur molto diffuso, presenta notevoli limitazioni come, ad esempio, la mancanza di supporto per i valori nulli, il limite nel numero di caratteri per la definizione del nome di un campo, il limite di 256 caratteri per un campo testuale. Il vantaggio del formato Shapefile è che è un formato aperto supportato da tutti i software in circolazione e che la stragrande quantità di dati GIS archiviati dagli anni ’80 e ’90 è archiviata in questo formato.

GEOGRAPHIC MARKUP LANGUAGE (GML)

Utilizza l’Extensible Markup Language (XML), perciò rappresenta una soluzione molto interessante in quanto il suo contenuto può essere facilmente compreso sia da un umano che da un computer, e può essere esteso o incapsulato in un altro linguaggio basato su XML.

KEYHOLE MARKUP LANGUAGE (KML)

Basato su XML come il GML, è stato sviluppato dall’azienda Keyhole, nata nel 2001 e poi acquisita da Google nel 2005. Uno dei vantaggi maggiori del formato KML è la possibilità di inserire modelli tridimensionali all’interno dei file.

Una recente categoria di formati è costituita dai cosiddetti *microformati*. Basati su XML, permettono di estendere formati disponibili. Per quanto riguarda i dati geografici, il microformato più utile è sicuramente il *geo*, con cui, ad esempio, è possibile incapsulare una posizione geografica all’interno di una pagina web[1].

2.4. Le licenze sui dati e l’accessibilità ai dati

Negli ultimi anni la questione dell’accessibilità ai dati pubblici ha generato nuovi spunti per quella che sembra la nuova frontiera dell’accessibilità all’informazione. Similmente a quanto successo per il software libero negli anni ’80, istituzioni, professionisti e associazioni si stanno adoperando per rendere i dati digitali accessibili, attraverso progetti che hanno come oggetto i cosiddetti *Open Data* o *Dati Aperti*. A livello internazionale, la Open Knowledge Foundation (OKF) sta organizzando progetti di accessibilità ai dati e individuando possibili licenze d’uso che tutelino la fruibilità dei dati e i diritti dei produttori dei dati[2]. In generale, la Open Knowledge Foundation definisce dati aperti come:

[1] http://www.microformats.org/wiki/geo
[2] http://okfn.org

> "Un contenuto o un dato è aperto se chiunque ha la possibilità di usarlo, riutilizzarlo e ridistribuirlo liberamente – con l'eventuale vincolo di citare la fonte e/o ridistribuire allo stesso modo."

Questa definizione viene formalizzata in undici punti in cui viene fatto riferimento alla licenza d'uso dei dati. Come è successo per il software libero, anche per i dati, è necessario definire delle licenze d'uso che specifichino le modalità di uso e distribuzione dei dati.
Dal momento che le licenze Creative Commons sono soprattutto utili alle creazioni artistiche o ai contenuti multimediali, la Open Knowledge Foundation si è concentrata sulla proposta di una licenza specifica per i dati, ovvero contenuti oggettivi di osservazioni la cui qualità non dipende direttamente dalla creatività soggettiva di chi li produce.
L'attività della Open Knowledge Foundation ha portato alla stesura di una serie di Open Data Common Licenses, la più nota delle quali è la *Open Data Base License* (ObbL) che sta venendo presa in considerazione da progetti come OpenStreetMap o da istituzioni come l'amministrazione comunale di Parigi. Il concetto generale della OdbL è che i dati possono essere distribuiti e utilizzati per qualsiasi scopo, a patto che una copia dei dati rimanga disponibile, che la fonte venga citata e che le eventuali modifiche vengano rese disponibili. Mentre questa può costituire una soluzione ideale per un piccolo progetto, potrebbe rappresentare un problema nel caso di grandi archivi dati provenienti da istituzioni nazionali e internazionali, come ad esempio dati pubblicati da uffici meteorologici, istituti geofisici o ministeri. In questi casi, chi pubblica i dati sarà interessato al fatto che la sorgente dei dati sia citata correttamente, mentre l'utilizzatore sarà principalmente interessato all'accessibilità e attendibilità delle informazioni. Per questo tipo di situazioni possono essere utilizzate la Open Data Commons Attribution License che difende il diritto di citazione, e la Open Data Commons Public Domain Dedication and Licence che è la formalizzazione del *pubblico dominio*.
Un aspetto importante per le licenze sui dati è la compatibilità tra licenze. È importante che queste non contrastino con la possibilità di analisi di aggregati di vari tipi di dati. Infatti, se dal punto di vista tecnologico non esistono più ostacoli per l'aggregazione di dati liberi in formati aperti, il problema si sposta nel campo giuridico.
Attualmente, l'unica licenza che garantisce una compatibilità anche futura tra i dati è la Creative Common Zero, che non richiede nessun obbligo all'utilizzatore dei dati, come il pubblico dominio. Questa licenza è particolarmente adatta a dati la cui raccolta e pubblicazione viene finanziata da risorse economiche pubbliche, per cui l'unico ritorno necessario è la credibilità nei confronti del cittadino creata da una pubblica amministrazione efficiente.
Un esempio di dati pubblicati con modalità di pubblico dominio sono i dati degli enti governativi statunitensi. Il Freedom Of Information Act obbliga tutte le agenzie governative americane a rendere fruibile al cittadino il risultato del lavoro dei dipendenti pubblici e di investimenti di denaro pubblico in ricerca e sviluppo tecnologico. Per questo motivo, il Servizio Geologico degli Stati Uniti (USGS) o la NASA hanno da sempre allocato regolarmente risorse per pubblicare i dati geografici e renderli fruibili.
Il diverso contesto storico con cui si è sviluppata l'Europa ha fatto dei dati geografici una risorsa strategica in campo militare e l'accesso ai dati ha avuto sempre molte restrizioni.

In Europa, il tema degli Open Data geografici e non è un argomento di grande attualità e interesse.

Il tema delle licenze d'uso dei dati geografici e dell'accessibilità ai dati è in fase di rapido sviluppo e motivo di discussione tra utenti e realtà pubbliche e private italiane e internazionali. In generale, viene riconosciuto che la fruibilità dei dati rappresenta un valore aggiunto importante. Grazie a progetti come OpenStreetMap anche i non esperti del settore GIS stanno facendo esperienza della dinamicità dei dati geografici (e non solo). Errori, imprecisioni o mancanza di informazioni in un insieme di dati diventano quindi una fonte di nuovi sviluppi e progetti, a tutto vantaggio delle realtà private e pubbliche che gravitano attorno ai dati geografici. Le licenze d'uso sui dati sono quindi molto importanti per facilitare questo meccanismo di ri-uso e miglioramento.
Durante la scrittura di questo libro, la discussione in Italia è attualmente aperta e nuovi problemi, idee e soluzioni vengono proposti ogni giorno. L'associazione GFOSS.it, la fondazione OpenStreetMap, l'associazione per il Software Libero (ASSOLI) sono alcune delle realtà in cui la discussione su dati e software viene portata avanti quotidianamente. È possibile partecipare alle attività come soci oppure partecipando alle liste di discussione (*mailing list*) aperte ai non-soci. Molti dei dati a cui possiamo accedere oggi in Italia sono stati rilasciati proprio grazie a un confronto fra potenziali utenti e fornitori di dati e le singole esperienze vengono solitamente utilizzate per migliorare quelle future, in uno scenario particolarmente dinamico e vitale.

3. GRASS GIS & Quantum GIS: caratteristiche e campi di applicazione

di Ivan Marchesini

3.1. Cenni storici

Prima di entrare nel dettaglio dei software oggetto del presente libro, è bene cercare di documentarne la storia. La piena comprensione della genesi e dei processi che governano lo sviluppo dei software porterà infatti il lettore ad apprezzare ancora di più gli strumenti e le loro funzionalità.
Nel corso dell'opera si fa riferimento alle versioni dei software stabili durante la fase di scrittura (fine 2011-inizio 2012). Non si può escludere che, al momento dell'uscita del libro, le *release* in sviluppo siano diventate stabili. I mutamenti previsti con le nuove versioni non sono però tali da alterare, in alcun modo, la validità e l'applicabilità dei contenuti del volume. Nel testo inoltre, si fanno continui accenni alle funzionalità previste e integrate nelle versioni in procinto di essere rilasciate.

3.1.1. GRASS GIS

GRASS (Geographic Resources Analysis Support System) è il prodotto del lavoro di numerosi sviluppatori che si sono succeduti nel tempo[1].
GRASS trae origine dalla necessità di alcuni giovani ricercatori dell'Army Corps of Engineer's Construction Engineering Research Laboratory (CERL – Champaign/Urbana, Illinois – fine anni '70) di lavorare con i GIS. Dopo aver scartato, anche per ragioni economiche (il software e l'hardware necessario per farlo girare erano troppo costosi), la soluzione di adottare GIS da poco apparsi sul mercato del software proprietario/commerciale, decisero di iniziare lo sviluppo di un alcuni piccoli software per dimostrare l'utilità di strumenti capaci di gestire dati territoriali. Successivamente, in seguito alla nascita delle prime workstation con sistema operativo UNIX (era appena nata la SUN) e all'interesse di alcune compagnie nel distribuire questi circa 20 programmi (ancora chiamati IGIS – "Installation GIS") gli sviluppatori decisero di impacchettare tutto in un software chiamato appunto GRASS GIS.
Gli sviluppatori erano consci del fatto che non avrebbero potuto avere sufficienti fondi per far crescere GRASS internamente al CERL e decisero di dedicare gran parte del

[1] http://grass.osgeo.org/devel/grasshist.html

loro tempo nel rendere possibile, a chiunque fosse in grado di farlo, lo sviluppo di nuovi strumenti per GRASS. Alcune università e altre istituzioni militari si aggiunsero quindi alla lista degli sviluppatori di GRASS.
Nacque la necessità di fare incontri tra utenti e sviluppatori e, proprio in occasione del primo incontro annuale, fu rilasciato GRASS 1.0. Era il 1985, in Italia il Verona vinceva il campionato di calcio, in Giappone usciva per la prima volta Super Mario Bross, negli Stati Uniti a novembre usciva Windows 1.0 ma ad ottobre era appena nata la Free Software Foundation fondata da Richard Stallman e cuore del movimento per lo sviluppo del software libero.
Si decise la convenzione dei nomi: i numeri principali (1.0, 2.0, ecc.) avrebbero indicato cambiamenti sostanziali, i numeri secondari (1.1, 1.2), invece, aggiornamenti e correzioni di errori (*bug fixing*) nell'ambito della stessa versione.
Nel 1986 usciva la versione 1.1, ma nel 1987 si era già arrivati alla versione 2.0 e nel 1988 alla versione 3.0 che integrava il sistema di processamento delle immagini e la gestione dei dati vettoriali (GRASS nacque inizialmente come sistema orientato alla gestione di dati raster).
Nel 1987 viene realizzato un video promozionale di GRASS, tutt'ora disponibile in rete[2].
Nel 1991 esce la versione 4.0 che si occupa di uniformare i comandi (più di 170) già esistenti, oltre a fornire una guida utente di riferimento che spinge numerose università a organizzare corsi basati su GRASS.
GRASS inizia ad essere distribuito in rete e nasce una comunità di utenti (per lo più basata su enti di ricerca) che si occupa di mantenere, sviluppare e aggiornare il codice. Nel 1996, in seguito alle lamentele dei produttori di software GIS proprietario/commerciale, i quali non ritenevano corretto che il governo federale entrasse in competizione con esse (Westervelt, 2004), e alla diminuzione drastica del costo dei software e dell'hardware, il supporto a GRASS presso il CERL viene interrotto.
Si forma il primo GRASS Research Group presso la Baylor University, Texas, e prende il controllo dello sviluppo di GRASS (si deve ricordare che il codice del sofware era nel pubblico dominio). Contestualmente viene rilasciata la versione 4.2 di GRASS.
È il 1997, a Parigi muore lady Diana Spencer, Deep Blue della IBM batte Kasparov a scacchi, Dario Fo vince il premio nobel per la letteratura, ma soprattutto esce il libro "La Cattedrale e il Bazaar" di Eric Raymond, manifesto del software Open Source.
Dal 1998 iniziano ad uscire release di GRASS in cui Markus Neteler (un giovane studente dell'università di Hannover e co-autore di questo libro) contribuisce in modo più che sostanziale. Grazie al lavoro di Markus e altri contributori, nel 1999 viene pubblicata con licenza GPL la release 5.0 di GRASS (si rimanda al capitolo 2 per comprendere l'importanza di questo passaggio e si legga, su internet, l'annuncio ufficiale della pubblicazione[3]).
In poco tempo, grazie a questa licenza, il team di sviluppatori di GRASS cresce sensibilmente. Nel dicembre 1999 il codice sorgente viene pubblicato su CVS (Concurrent Versions System – un sistema, repositorio, per gestire il controllo delle versioni di file, tipicamente utilizzato nel processo di sviluppo software). In questo modo, il software

[2] http://grass.osgeo.org/grass_movie_CERL_1987
[3] http://grass.osgeo.org/announces/gnu-release.html

risultava aperto a tutti in lettura/download, mentre solo gli sviluppatori potevano operare modifiche. Si superava così il collo di bottiglia che si era venuto a creare quando molti sviluppatori chiedevano contemporaneamente a Markus di applicare modifiche al codice. Dopo alcuni anni ad Hannover, il sito web di GRASS per l'Europa raggiunge Trento (ente di ricerca ITC – oggi conosciuto come FBK, Fondazione Bruno Kessler) dove Markus si trovava a lavorare. Nel 2002 viene rilasciato GRASS 5.0 stabile e nel 2004 esce la release 5.7 che, pur essendo una versione di sviluppo, introduce numerose novità, soprattutto nel campo della gestione dei dati vettoriali.
Dal 2005, anno in cui esce la versione 6.0 stabile GRASS si sono succedute numerose altre realease come la 6.2 (2006) e la 6.4 (2008): si tratta di versioni interne alla release 6, caratterizzate dall'introduzione di numerose nuove funzionalità e interfacce grafiche, ma sempre compatibili, in termini di formato e struttura dei dati, con la release principale (6.0).
Ad oggi, lo sviluppo di GRASS si concentra sulla continua manutenzione e correzione del codice della versione stabile e sullo sviluppo di una nuova release (la versione 7) che però, poiché prevede grandi novità, necessita, per essere compatibile con la versione precedente, di convertitori di formato interni (peraltro già presenti). Poiché si tratta di software libero GRASS GIS 7 è ovviamente già disponibile online per gli utenti che vogliano provare a utilizzarlo in anteprima.
Il GRASS-PSC (Project Steering Committee, cioè un comitato direttivo), nominato da utenti e sviluppatori, si occupa di applicare meccanismi per il controllo della qualità del software e rappresentare il progetto in tutte le occasioni in cui sia necessario. Markus Neteler, uno degli autori di questo libro, è "chair man" di questo organismo.

3.1.2. Quantum GIS

La storia di Quantum GIS (o QGIS) è più breve.
L'avvio risale al 2002 quando Gary Sherman inizia a sviluppare un GIS per visualizzare dati di tipo PostGIS (estensione spaziale per PostgreSQL rilasciata l'anno precedente da Refraction Research, si veda il capitolo 8 per altre informazioni). Gary pubblica su SourceForge (una piattaforma web per sviluppare software Open Source in modo collaborativo) il suo progetto e da qui inizia la fortunata ascesa di QGIS.
Numerosi sviluppatori si sono aggiunti al progetto e le release di QGIS si sono succedute con notevole velocità, aumentando stabilità e potenzialità. Fino al 2009 ci sono stati circa 26 rilasci di nuove release di QGIS. Nel 2009 però QGIS raggiunge la maturità con una release, la 1.0.0, che sviluppa una API (*Application Programming Interface*) stabile e utilizzabile da utenti esperti per sviluppare propri strumenti con linguaggi come Python e C++.
QGIS ha una politica di rilascio di versioni stabili (1.0.0, 1.4.0, 2.0.0) per le quali prevede stabilità delle API per almeno 2 anni dalla uscita della release. Il "release team" prevede poi delle versioni intermedie per le quali si mantiene la compatibilità con le API precedenti, ma queste vengono modificate (estese e aggiornate). Attualmente, il "release team" sta lavorando alla versione 1.8.0 e ha rilasciato la versione 1.7.3, che risolve molti problemi ereditati dalle versioni precedenti e introduce molte nuove funzionalità, come ad esempio la riproiezione al volo dei dati di tipo raster.

5. Quantum GIS[1]

di Luca Casagrande e Paolo Cavallini
coautori: Pierluigi De Rosa, Paola Napolitano, Franca Iacobellis

Tutti i dati geografici utilizzati all'interno del presente capitolo sono liberamente scaricabili dalla pagina dedicata a questo libro nel sito www.darioflaccovio.it.

5.1. Le prime operazioni per lavorare con Quantum GIS

Il primo passo per poter utilizzare QGIS è ovviamente quello di installarlo nel proprio computer. Questo procedimento, che verrà illustrato per i principali sistemi operativi, non prevede operazioni particolarmente complesse ed è spesso limitato all'avvio del file di installazione (nel caso dei sistemi operativi MS Windows), oppure alla scelta di QGIS nei canali software già disponibili per la distribuzione utilizzata (caso dei sistemi operativi Linux). Solo nel caso di sistemi Mac OS X è necessario installare a parte alcune librerie prima di procedere con il pacchetto di QGIS.
Indipendentemente dal proprio sistema operativo, i pacchetti possono essere prelevati dalla pagina web del progetto[2].
Al primo avvio QGIS verrà caricato con delle impostazioni predefinite sia per quanto riguarda l'interfaccia (posizione delle icone, lingua, ecc.) sia per le impostazioni di progetto (come l'unità di misura o il sistema di riferimento adottato).

5.2. Installazione e personalizzazione dell'interfaccia

5.2.1. Installazione nei sistemi Linux

La procedura di installazione per i sistemi operativi GNU/Linux è molto più semplice di quanto si possa pensare; spesso, infatti, QGIS si trova già nei repositorio *software* che le distribuzioni usano per archiviare gli applicativi che possono venir installati. Può capitare tuttavia che la versione presente in questi canali sia più vecchia rispetto all'ultima disponibile; in questo caso, è necessario attivare dei repositori aggiuntivi, in genere realizzati dagli sviluppatori del progetto e non supportati ufficialmente dalla distribuzione. Oltre a semplificare notevolmente la procedura di installazione, con questo sistema si è sempre aggiornati in caso di nuove versioni.
La prima distribuzione GNU/Linux presa in esame è *Ubuntu*. Attualmente arrivata alla versione 11.04, nome in codice *Natty*, contiene nei propri repositori la versione 1.4.0 di QGIS, che è indietro di tre *release* rispetto all'ultima versione stabile. Per mettere a dispo-

[1] I dati vettoriali provengono dal dataset di geodati liberi relativo al territorio piemontese e limitrofo, mentre i dati raster provengono dal progetto Landsat.
[2] http://download.qgis.org

sizione degli utenti *Ubuntu* le ultime versioni dei principali applicativi geografici, è stato realizzato un repositorio ad hoc chiamato *UbuntuGIS.* A seconda della propria versione *Ubuntu*, si dovrà utilizzare la versione *stable* o quella *unstable* del repositorio. Nella prima non sono presenti le ultime versioni dei software che vengono rilasciati, cosa che invece accade nella versione *Unstable*. La versione più recente di *Ubuntu* attualmente utilizza il secondo. Per aggiungere questo nuovo repositorio, si può intervenire tramite interfaccia grafica oppure con la linea di comando: in questo caso si analizzerà la seconda opzione così da permettere all'utente di familiarizzare con questo comodo strumento.
Dal menu *Accessori* selezionare la voce *terminale* e digitare i seguenti comandi seguiti dal tasto invio. Tutte le volte che verrà richiesto l'inserimento di una password, utilizzare quella del proprio utente scelta durante la fase di installazione.

sudo add-apt-repository ppa:ubuntugis/ubuntugis-unstable
sudo apt-get update
sudo apt-get install qgis

Con la prima riga di comando si aggiunge il repositorio *UbuntuGIS unstable*, con la seconda si aggiornano tutti i repositori così da aggiungere all'elenco anche i pacchetti del repositorio appena aggiunto e con l'ultima si procede all'installazione di QGIS nel sistema. Al termine della procedura verrà aggiunta una nuova voce nel menu SCIENZA per avviare QGIS.

Altra distribuzione diffusa tra gli utenti di GNU/Linux è *OpenSuse*. Attualmente alla versione 11.4, implementa un sistema di installazione dei software non ufficialmente distribuiti chiamato *OneClick*, tramite il quale è sufficiente cliccare su un link per installare il pacchetto desiderato. A tal proposito, è disponibile un sito che contiene un archivio di tutti i pacchetti disponibili. Cercando il termine QGIS, verrà mostrato l'elenco dei pacchetti che si riferiscono al software; basterà scegliere la versione 1.7.0 e cliccare sul pulsante *OneClick*.

5.2.2. Installazione in ambiente Mac OS X

Va sottolineato che dalla versione 1.6.0 di QGIS non è più disponibile un unico pacchetto di installazione come avveniva per le precedenti versioni, ma attualmente è necessario installare alcuni pacchetti come prerequisiti e quindi procedere all'installazione di QGIS.
Il materiale necessario è disponibile presso il progetto *KyngChaos*[3] sempre attivo nella realizzazione di pacchetti per applicativi geografici *Open Source* da utilizzare in ambiente Mac OS X.
Per la versione 1.7.0 è necessario installare sia il *GDAL Framework 1.8.0* sia il *GNU Scientific Library* (*GSL*), anche essi disponibili nella pagina di QGIS del progetto KyngChaos. Facoltativa è invece l'installazione di GRASS GIS.
Dopo avere installato tutte le dipendenze, si potrà procedere con l'installazione vera e propria di QGIS secondo le normali modalità delle applicazioni per Mac OS X.

[3] http://www.kyngchaos.com/software

5.2.3. Installazione in ambiente MS Windows

L'installazione in ambiente MS Windows avviene tramite un singolo file di installazione come la maggior parte degli applicativi disponibili per questo sistema operativo. Dopo aver avviato il file con un doppio click, ci si troverà a rispondere a una serie di domande come il tipo di installazione o il percorso dove installare il software. Attenzione va posta alla richiesta relativa ai file di esempio: è infatti possibile installare alcuni file (vettoriali e raster) utilizzabili per provare QGIS. L'unica elemento da tenere in considerazione è che i dati vengono prelevati tramite la rete internet e quindi il processo può necessitare di una certa quantità di tempo.

5.2.4. L'interfaccia di Quantum GIS

Al primo avvio la schermata di QGIS sarà simile a quella mostrata in figura 5.1; vi potranno essere delle differenze riguardo alle icone presenti nella barra degli strumenti o nella lingua dell'interfaccia, ma si vedrà nel prosieguo del capitolo come modificare queste impostazioni di default.

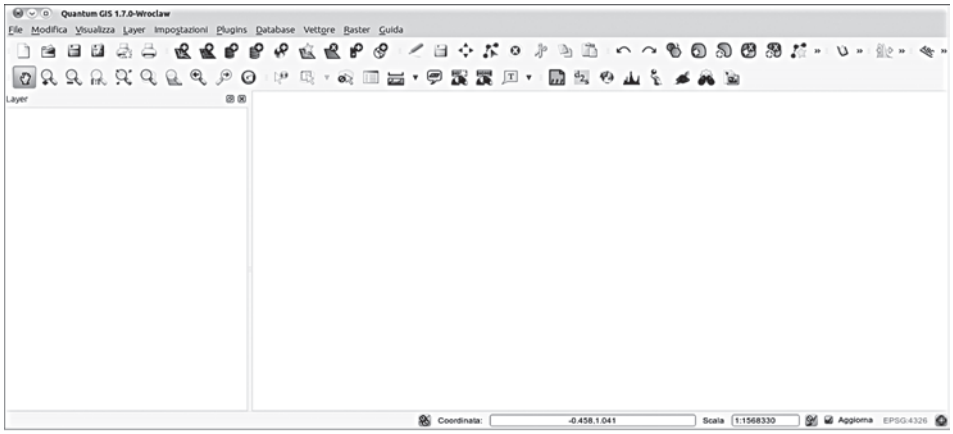

Figura 5.1. L'interfaccia di Quantum GIS al suo primo avvio

L'interfaccia di QGIS può venir suddivisa in tre aree: quella che contiene le barre degli strumenti, posizionata in alto, l'area per la visualizzazione dei dati caricati che occupa la parte principale della schermata, mentre la barra di stato, che contiene informazioni sull'area di studio come l'estensione geografica, è posizionata in fondo alla schermata. La parte destinata alla visualizzazione del dato può essere ridotta per aggiungere altri elementi come l'elenco dei layer caricati o la mappa panoramica. Come impostazione predefinita è attiva la visualizzazione dell'elenco dei layer.

5.2.5. Localizzazione e personalizzazione della barra degli strumenti

Può capitare che, in funzione della lingua del sistema operativo utilizzato, QGIS venga caricato con una lingua differente da quella che si desidera utilizzare.

Per modificare le impostazione sulla lingua, è sufficiente selezionare il menu *Impostazioni* (Settings) e qui scegliere la voce *Opzioni* (*Options*). Nella schermata visualizzata in seguito selezionare la voce *lingua* (locale) (nel caso non fosse visibile scorrere con la barra laterale), abilitare l'opzione "*sovrascrivi lingua in uso*" e scegliere la lingua desiderata.
Confermare la scelta cliccando su OK, chiudere e riavviare QGIS per rendere definitive la nuove impostazioni. Come nota a margine, si tiene a precisare che anche il lavoro di traduzione dell'interfaccia viene svolto principalmente da volontari e spesso viene lasciato per ultimo nella lista delle priorità per una data versione. Ecco perché potrà capitare che alcuni termini non siano stati tradotti e vengano caricati in inglese. Non essendo necessaria alcuna conoscenza informatica, il lavoro di traduzione dei progetti Open Source è solitamente il primo passo che viene fatto da coloro che intendono contribuire a un determinato progetto.

5.2.6. Personalizzazione delle barre degli strumenti

Ogni barra degli strumenti raggruppa una serie di pulsanti per svolgere operazioni dello stesso tipo; le barre possono essere spostate cliccando sull'elemento presente all'inizio di ciascuna barra e disposta in qualsiasi punto dello schermo. Si noti che tutta l'area intorno alla mappa può essere utilizzata per posizionare la barra. Può capitare che la dimensione di una barra sia troppo grande per la posizione scelta e che quindi alcuni pulsanti presenti vengano nascosti da altre barre o dal bordo della schermata. In questo caso QGIS visualizza un'icona che, quando selezionata, visualizzerà i pulsanti non visibili.
È anche possibile selezionare quale barra degli strumenti visualizzare: questa scelta viene fatta facendo click con il tasto destro del mouse sulla barra o su un'area vuota e scegliendo le barre da visualizzare tramite le apposite checkbox.
In questo stesso elenco, oltre alle barre degli strumenti, sono selezionabili anche la schermata layer e la panoramica. La barra di stato contiene strumenti che permettono la visualizzazione delle coordinate corrispondenti alla posizione del mouse. L'ampiezza dell'area visualizzata può essere modificata utilizzando l'apposito pulsante visualizzando l'estensione attuale della mappa.
Di seguito viene indicata la scala di rappresentazione attuale e l'opzione per attivare/disattivare l'aggiornamento della mappa. L'ultimo strumento della barra di stato ha una duplice funzionalità: con un click permette di accedere alla schermata delle proprietà del progetto (di cui si parlerà in dettaglio nei capitoli seguenti); oltre a questo, permette di verificare subito se è attiva la riproiezione al volo dei dati come è possibile vedere nella figura 5.2.

5.2.7. Altre personalizzazioni

Attraverso il menu OPZIONI, è possibile modificare alcune impostazioni utilizzate durante il lavoro con QGIS. Sebbene siano molte le opzioni modificabili, di seguito verranno prese in considerazione solo alcune, lasciando la descrizione di tutte le altre al manuale di riferimento del software.

5.3. Nuovo progetto e visualizzazione dei dati

Ogni volta che QGIS viene avviato, viene automaticamente creato un nuovo progetto. All'interno del progetto possono essere caricati uno o più livelli informativi, sia vettoriali che raster. Quando si salva il progetto, le informazioni vengono salvate in un file di progetto (*.qgs) e includono i layer inseriti, le proprietà dei layer e la proiezione usata per il progetto. Le opzioni del progetto possono essere configurate tramite il menu *Proprietà progetto*, come mostrato in figura 5.2.

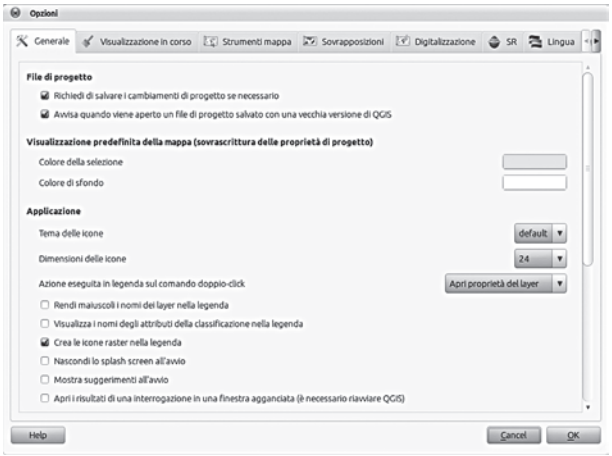

Figura 5.2. La schermata per le proprietà del progetto

Nella scheda *Generale* è possibile impostare, oltre alle scelte di visualizzazione (colore di visualizzazione e di sfondo, ecc.) e l'unità di misura, la modalità di salvataggio del percorso dei file: "assoluto" salva tutto il percorso, a partire dal nome del disco o dalla directory root, ed è quindi utile quando si voglia mantenere la possibilità di spostare il progetto all'interno del file system (o del disco), mantenendo la certezza che i file saranno trovati nella loro dislocazione; "relativo" invece consente di spostare una directory con il progetto e tutti i file relativi (ad esempio, trasferirla da un computer all'altro, tramite chiave USB o internet) mantenendo intatti tutti i riferimenti.

5.3.1. La gestione dei sistemi di riferimento

Un problema molto frequente, e di solito inevitabile nei GIS, è quello di trovarsi a dover lavorare con informazioni, fornite da vari soggetti (ad esempio Comunità Europea, Ministeri, Regioni, Province, Comuni, ecc.) e registrate in sistemi di riferimento diversi. In questi casi, è in primo luogo necessario conoscere il sistema in cui i dati sono georeferenziati; nei casi migliori, questo è completamente automatico, in quanto i file stessi sono già dotati di informazioni sulla loro georeferenziazione. Ad esempio, per gli ESRI Shapefile esiste un file accessorio, in formato testo e di estensione .prj, al cui interno sono contenute le informazioni necessarie; ad esempio, il contenuto di un .prj per la proiezione Universale Trasversa di Mercatore, zona 32 è:

PROJCS["WGS_1984_UTM_Zone_32N",GEOGCS["GCS_WGS_1984",DATUM["D_WGS_1984",SPHEROID["WGS_1984",6378137,298.257223563]],PRIMEM["Greenwich",0],UNIT["Degree",0.0174532925199433295]],PROJECTION["Transverse_Mercator"],PARAMETER["latitude_of_origin",0],PARAMETER["central_meridian",9],PARAMETER["scale_factor",0.9996],PARAMETER["false_easting",500000],PARAMETER["false_northing",0],UNIT["Meter",1]]

Quando questo file (o analoghi, per altri formati di file) è correttamente presente, QGIS lo interpreta associando a quel layer quel sistema di riferimento. Se non esiste, l'utente è responsabile dell'attribuzione delle proiezione corretta. Questo si realizza facendo click con il tasto destro del mouse sul nome del layer, selezionando la voce *Proprietà layer* e, nella scheda *Generale*, l'opzione "Specifica SR", ovvero Sistema di Riferimento.
Le proiezioni riconosciute da QGIS sono inserite in un database interno, che include tutte quelle codificate dall'*European Petroleum Specialist Group*, oltre ad alcune definite dall'*Institut Géographique National* Francese; c'è inoltre la possibilità di definire proiezioni personalizzate, nel caso ad esempio siano disponibili fattori di correzione locali. I codici EPSG sono di particolare importanza, perché estremamente diffusi e univoci per ogni sistema. È quindi conveniente riferirsi ai sistemi più usati non tramite la loro descrizione verbale (ad esempio WGS84), che è sempre più o meno ambigua, ma tramite il loro codice EPSG, sempre univoco e definito con chiarezza; ad esempio il codice 4326 indica:

GEOGCS["WGS 84",
DATUM["WGS_1984",
SPHEROID["WGS 84",6378137,298.257223563,
AUTHORITY["EPSG","7030"]],
AUTHORITY["EPSG","6326"]],
PRIMEM["Greenwich",0,
AUTHORITY["EPSG","8901"]],
UNIT["degree",0.01745329251994328,
AUTHORITY["EPSG","9122"]],
AUTHORITY["EPSG","4326"]]

Una volta che ad ogni strato sia stato attribuito il sistema di riferimento corretto, in modo automatico o manuale, è sufficiente abilitare la riproiezione al volo, tramite il menu *Proprietà del progetto*, scheda *Sistema di Riferimento (SR)* per avere tutti gli strati allineati. È da notare come si possano riproiettare gli strati non solo in una delle proiezioni dei dati di origine, ma anche in un altro sistema. È anche possibile impostare, tramite le *Opzioni*, un sistema di riferimento che tutti i nuovi progetti adotteranno. Inoltre, si può definire, tramite lo stesso menu, come comportarsi nel caso che un layer caricato non abbia un file di proiezione associato; se si lavora con molti sistemi diversi, è più opportuno fare in modo che QGIS richieda ogni volta quale sistema attribuire; se invece la maggior parte dei layer, senza sistema di riferimento associato esplicitamente, hanno la stessa origine, si può impostare un SR predefinito. In casi particolari, in cui si carichino solo layer con SR identici ma senza file .prj, si può selezionare l'opzione *Usa il SR del progetto* e

impostando il progetto stesso con il SR voluto. Ovviamente, in questo caso non è utile abilitare la riproiezione al volo.
Riproiettare un layer impone un certo carico computazionale sul computer, in quanto va ricalcolata la posizione di ogni vertice (o ogni cella nel caso dei raster). Si tenga presente che questo, nel caso di layer molto grandi o complessi, e di computer di bassa potenza, può causare dei rallentamenti. Se si verifica questo, è consigliabile riproiettare tutti i layer in modo statico, salvando un nuovo file. Ciò può essere realizzato con la massima semplicità, tramite il menu *Salva come...*, attivabile cliccando col tasto destro del mouse sul nome del vettore, nella legenda. Tramite lo stesso menu si può contestualmente convertire in un altro formato.
Ogni riproiezione introduce un certo errore, che dipende dalle formule di conversione utilizzate (in generale pochi metri, fino a decine di metri nei casi peggiori); QGIS effettua queste operazioni tramite la libreria PROJ.4, la stessa utilizzata in molti altri programmi GIS, sia liberi che proprietari; perciò la precisione è la stessa nella maggior parte dei casi. Il sistema più appropriato per aumentare la precisione è quello di utilizzare il metodo dei grigliati, ora disponibile tramite un plugin addizionale di QGIS (*Transformation Manager*). Per utilizzare questo metodo sono necessari dei file di conversione, che per l'Italia non sono liberamente disponibili (vanno richiesti all'Istituto Geografico Militare o, in alcuni casi, alle Regioni); con questo metodo si possono realizzare riproiezioni, al volo o statiche, con un livello di precisione molto più alto (in generale nell'ordine dei decimetri o meno, in funzione della distanza fra i punti della griglia).
Nella schermata dello schedario *Proprietà Layer > Layer interrogabili* è inoltre possibile scegliere su quali layer deve agire lo strumento di interrogazione (recupero informazioni).

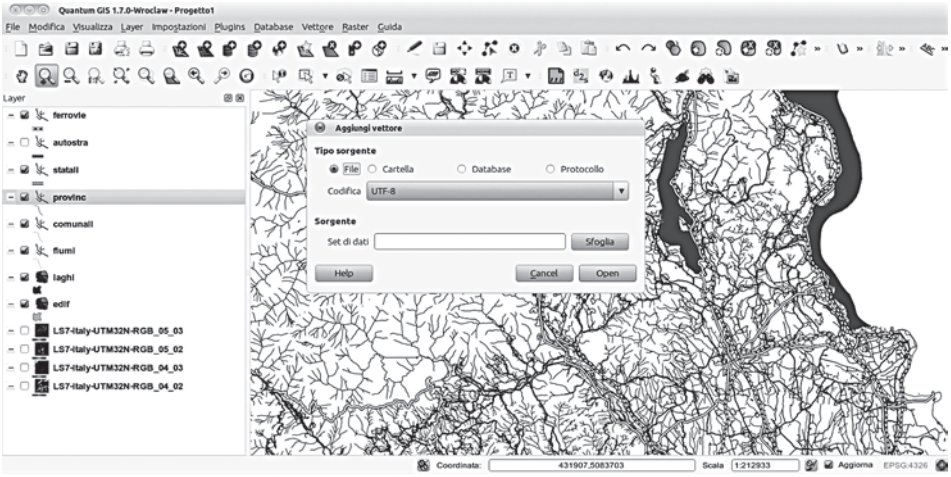

Figura 5.3. Schermata per aggiungere un nuovo file

Una volta caricati i layer, è possibile cambiarne le proprietà di visualizzazione nella scheda *Stile* presente nella finestra delle proprietà del layer, attivabile con doppio click

sul nome del layer oppure click destro su nome del layer e selezionare, nel menu contestuale, *Proprietà*.

5.3.2. Visualizzazione di dati vettoriali

Uno dei punti di forza di QGIS è la capacità di caricare molti tipi di dati territoriali: siano essi in formato file, come i diffusi ESRI ShapeFile o GeoTiff, oppure database o servizi web, come il Web Map Service adottato anche dal Portale Cartografico Nazionale[4].
Il caso più semplice è quello del caricamento di un dato vettoriale cliccando sul tasto *Aggiungi vettore* o selezionando il *Menu Layer > Aggiungi Vettore*.
Nella schermata visualizzata è possibile selezionare il file, la cartella o il database da caricare ed è possibile sfogliare la lista dei file, filtrando il tipo di interesse fra le decine di formati disponibili (ad esempio ESRI ShapeFile, MapInfo, Microstation DGN, GML, KML, ecc.). Al caricamento, ad ogni layer viene attribuito un colore casuale (a meno che il layer non abbia già uno stile associato, come si vedrà in seguito), in modo che ogni layer sia riconoscibile dagli altri. Man mano che si aggiungono altri layer, questi vengono sovrapposti, nella legenda e nell'area di mappa, per cui può capitare che alcuni layer caricati precedentemente vengano nascosti da quelli sovrastanti.
Una volta caricati i layer, è possibile cambiare le proprietà di visualizzazione tramite la finestra *Schedario Proprietà layer > Stile*, attivabile con doppio click sul nome del layer oppure click destro sul nome del layer e selezionare, nel menu contestuale, *Proprietà*.

Figura 5.4. Gestione dello stile

[4] http://www.pcn.minambiente.it/PCNDYN/catalogowms.jsp

6. GRASS GIS

di Alessandro Frigeri e Markus Neteler
coautori: Luca Delucchi, Duccio Rocchini

Tutti i dati geografici utilizzati all'interno del presente capitolo sono liberamente scaricabili dalla pagina dedicata a questo libro nel sito www.darioflaccovio.it.

6.1. Installazione del GRASS GIS

di Alessandro Frigeri

6.1.1. Cenni introduttivi

Il GRASS GIS[1] rappresenta il riferimento per chi usa il GIS per risolvere una vasta quantità di problemi spaziali, specie se si devono elaborare grandi quantità di dati in catene di elaborazione particolari. Proprio per l'elevata motivazione degli utenti, all'inizio GRASS era un software che veniva installato in computer di ricercatori, professionisti e studenti partendo dal codice sorgente. La nascita di pacchetti precompilati di GRASS ha permesso poi la sua diffusione al di fuori di istituti e enti specializzati, luoghi cioè in cui non è semplice trovare ambienti di sviluppo ed esperti in grado di compilare e mantenere un software a partire dal suo codice sorgente.
Negli ultimi 10 anni, l'installazione di GRASS è divenuta sempre più semplice e intuitiva. I sistemi GNU/Linux hanno pacchetti di GRASS che permettono di installare velocemente il software senza preoccuparsi di installare le numerose librerie software da cui dipende. Il porting nativo su Microsoft Windows ha esteso le possibilità di installazione, arrivando fino a installer grafici estremamente intuitivi. Anche l'installazione su OS X è particolarmente semplice, ma necessita di alcuni pacchetti software reperibili attraverso l'installer grafico del sistema operativo.

6.1.2. Installazione su sistemi GNU/Linux

L'installazione di software libero su un sistema operativo libero è l'esperienza che più entusiasma e sconvolge chi da sempre è stato abituato a utilizzare software proprietario. Infatti, praticamente ogni sistema operativo basato su software libero ha un sistema per la gestione automatica dei software installati. I due più diffusi sistemi di impacchettamento di software sono quelli sviluppati da Red Hat e Debian, a cui fanno riferimento le distribuzioni derivate da esse. Oltre ai pacchetti software e agli strumenti per installarli, sono disponibili alcuni strumenti per l'installazione e aggiornamento automatico non

[1] http://grass.osgeo.org

solo del software desiderato, ma anche di tutti i software a lui necessari per funzionare correttamente.
Essendo GNU/Linux un sistema multi utente, l'installazione dei pacchetti richiede di agire in veste di amministratore di sistema. Tutti i comandi di installazione di questa sezione dovranno essere eseguiti come amministratore di sistema, che laddove si utilizzi la riga di comando, solitamente si ottengono tramite (richiede password di root):

su -c "comando_da_eseguire_come_amministratore"

Oppure (richiede password dell'utente):

sudo "comando_da_eseguire_come_amministratore"

Laddove si usino le interfacce grafiche, una finestra richiederà l'immissione della password di amministratore.

6.1.3. Installazione di pacchetti RPM

Il Red Hat Package Manager (RPM) è un sistema di gestione di pacchetti a riga di comando con il quale è possibile installare, disinstallare, verificare, interrogare e aggiornare pacchetti software. RPM è software libero, rilasciato con licenza GPL ed è diventato il gestore di pacchetti, oltre che della distribuzione Red Hat, anche di Fedora Project, SUSE, openSUSE, CentOS, Mandriva, Scientific Linux. Nel caso delle distribuzioni derivate da Red Hat, esistono diversi strumenti di alto livello per la gestione automatica degli aggiornamenti: *yum* su Fedora, Red Hat Enterprise Linux, CentOS e Scientific Linux, *urmpi* su Mandriva, *zypper* per OpenSUSE.

6.1.4. Installazione su Red Hat Enterprise Linux, CentOS e Scientific Linux

Per installare GRASS su Red Hat Enterprise Linux (RHEL), CentOS e Scientific Linux è necessario aggiungere dei repositori di pacchetti alla distribuzione standard.
Si aggiunga innanzitutto il repositorio EPEL (Extra Packages for Enterprise Linux). Se il nostro sistema è basato su Red Hat 5 (RHEL 5, CentOS 5, Scientific Linux 5), lanciamo il comando da terminale:

rpm -Uvh http://download.fedora.redhat.com/pub/epel/5/i386/epel-release-5-4.noarch.rpm

Se invece il nostro sistema è basato su Red Hat 6 (RHEL 6, CentOS 6, Scientific Linux 6), si lanci il comando da terminale:

rpm -Uvh http://download.fedora.redhat.com/pub/epel/6/i386/epel-release-6-5.noarch.rpm

Nel caso specifico del sistema operativo RHEL 6, per permettere l'aggiunta di EPEL si abiliti il repositorio *Optional* attraverso il comando *rhn-channel* (facendo riferimento al manuale della distribuzione).

A questo punto aggiungiamo il repositorio ELGIS (Enterprise Linux GIS), che contiene una lunga lista di software GIS oltre a GRASS e può essere attivato lanciando il comando:

rpm -Uvh http://elgis.argeo.org/repos/6/elgis-release-6-6_0.noarch.rpm

È possibile controllare che i repositori siano stati inseriti correttamente impartendo il comando:

yum repolist

ELGIS dovrebbe essere elencato. Ora si verifichi che GRASS sia disponibile nel repositorio remoto:

yum list grass

si dovrebbe ottenere la versione di GRASS fornita da ELGIS. Se tutto è andato liscio, possiamo installare GRASS:

yum install grass

6.1.5. Installazione di GRASS su OpenSUSE

Per installare GRASS in OpenSUSE si aggiunga il repositorio Geo, scaricando dalla rete il file:

http://download.opensuse.org/repositories/Application:/Geo/openSUSE_XX.Y/Application:Geo.repo

Dove XX.Y è la versione di OpenSUSE che si sta usando (per esempio 11.2, 11.3, 11.4, 12.1). Si aggiunga il contenuto del file appena scaricato al file /etc/zypp/repo.d. e si aggiorni la lista di pacchetti con il comando:

zypper update

A questo punto installiamo GRASS*:*

zypper install grass

6.1.6. Installazione di GRASS GIS su Mandriva

La distribuzione Mandriva contiene già pacchetti aggiornati di GRASS, e l'installazione si limita al semplice comando:

urmpi grass

Oppure tramite l'interfaccia grafica di installazione e rimozione software che si trova nel menu applicazioni: *Installazione e Rimozione Software*.

6.1.7. Installazione su Debian e Ubuntu

Il sistema di gestione dei pacchetti della distribuzione Debian e delle sue derivate, come la popolare Ubuntu, si chiama *dpkg*. Come abbiamo visto con RPM, anche *dpkg* permette di installare, disinstallare, verificare, interrogare e aggiornare pacchetti software. In Debian e le distribuzioni da lei derivate, dal 1998 esiste APT, l'Advanced Package Tool che fornisce il programma a riga di comando *apt-get*, capace di scaricare e installare programmi da diverse sorgenti della rete. Una volta scaricato il software, *apt-get* chiama automaticamente le funzionalità di *dpkg*.
Le funzionalità di *apt* sono molto apprezzate da amministratori di sistema e utenti di tutto il mondo. Infatti APT permette di installare un software senza doversi preoccupare di scaricare anche i software da cui dipende, che vengono scaricati e installati secondo un ordine che permetterà al software desiderato di funzionare.
APT è un sistema sofisticato ed elegante, ed è importante "aiutarlo" a fare il proprio lavoro. La regola fondamentale affinché funzionino perfettamente i sistemi Debian è utilizzare i pacchetti forniti dalla distribuzione ed evitare assolutamente installazioni di pacchetti generati localmente o non rilasciati assieme alla distribuzione. In questo modo si otterrà un sistema che sarà funzionante e aggiornato per anni, e laddove si verifichi un problema, si potrà segnalarlo e probabilmente risolverlo con l'aiuto degli sviluppatori che curano i pacchetti software.
Installare GRASS con il sistema APT è molto semplice e, come la maggioranza delle operazioni in ambienti simili a UN*X, possiamo utilizzare un programma a interfaccia grafica oppure la riga di comando. Il front-end grafico più utilizzato di APT si chiama *Synaptic* e permette di cercare il software per categoria oppure attraverso una form di ricerca testuale. Per trovare GRASS in *synaptic* si dovrà andare nella categoria *Scienza*, cercare in ordine alfabetico il pacchetto e, cliccando con il tasto destro del mouse sulla riga contenente il pacchetto GRASS, selezionarlo per l'installazione. È bene installare anche la relativa documentazione: *grass-doc*. Si avrà così la possibilità di accedere al manuale già dal primo utilizzo.
Installare GRASS da riga di comando è ancora più semplice e veloce: usando la riga di comando, come superutente, digitare:

```
sudo apt-get install grass grass-doc
```

SUDO (*super user do*) è un sistema usato di default nei sistemi tipo Ubuntu per permettere all'utente di eseguire operazioni di amministrazione. Anteponendo il comando *sudo* a qualsiasi istruzione, verrà richiesta la password di amministratore e l'istruzione verrà eseguita in modalità *superuser*.
Nel caso in cui si volesse aggiornare GRASS a una versione più recente, basterà aggiornare gli archivi dei pacchetti in *synaptic* oppure eseguire, come superutente, il comando:

```
sudo apt-get update
sudo apt-get upgrade grass grass-doc
```

Con questi due comandi, lanciati in sequenza, si ordina di modificare gli indici con le versioni aggiornate dei pacchetti e di aggiornare i pacchetti *grass* e *grass-doc*.

6.1.8. Installazione di GRASS su Mac OS X

Con Mac OS X si entra nel campo dei sistemi operativi, e quindi anche dei gestori di pacchetti, proprietari. Di solito i pacchetti software per Mac OS X vengono distribuiti sotto forma di file con estensione *.dmg*, che indica un'Apple Disk Image, un formato di immagine disco proprietaria che supporta la compressione e quindi adatto alla distribuzione di pacchetti software in rete. A differenza di quanto accade per i sistemi operativi Open Source, non esiste un gestore di pacchetti che si preoccupi di scaricare e installare delle componenti software per far funzionare l'applicazione. Per avere un'installazione aggiornata di GRASS su Mac OS X si dovrà quindi installare i diversi pacchetti software necessari all'esecuzione di GRASS nelle versioni e nell'ordine giusti. L'utente di GRASS William Kyngesburye fornisce sul suo sito[2] tutti i binari necessari al funzionamento del GRASS in Mac OS X Leopard o nell'ultima versione, Snow Leopard.
Il primo passo per poter scaricare la versione corretta dei pacchetti è individuare la versione di Mac OS X su cui si vuole installare GRASS. Per sapere su quale versione di Mac OS X si sta lavorando è possibile procedere in due diversi modi. Il primo è attraverso l'interfaccia grafica, andando a selezionare la prima voce in alto dal menu Apple (la mela in alto a sinistra del pannello di Mac OS X): "Informazioni su questo Mac". Nella finestra di informazioni che appare, sotto la scritta Mac OS X, è riportata la versione nella forma di tre numeri separati da un punto. Il primo numero indica che si sta lavorando su un Mac OS X. Dopo il primo numero si ha il codice che identifica la versione dei rilasci più rilevanti di Mac OS X. I codici numerici dei rilasci maggiori di Mac OS X sono comunemente associati ad animali, in particolare felini selvatici (nota: tradizione nata dall'iniziale competizione con MS Windows Vista che era codificata con 'Longhorn', un bovino da carne). Quindi quando si parla di Mac OS X Jaguar, Panther, Tiger, Leopard, Snow Leopard si sta rispettivamente parlando delle versioni 10.2, 10.3, 10.4., 10.5. e 10.6. Nota la versione di Mac OS X, è possibile procedere all'acquisizione e installazione dei diversi pacchetti. Si divida l'installazione di GRASS in due fasi: l'acquisizione dei pacchetti *dmg*, l'installazione dei Framework e di GRASS nell'ordine giusto.

6.1.8.1. Scaricamento dei pacchetti dalla pagina web

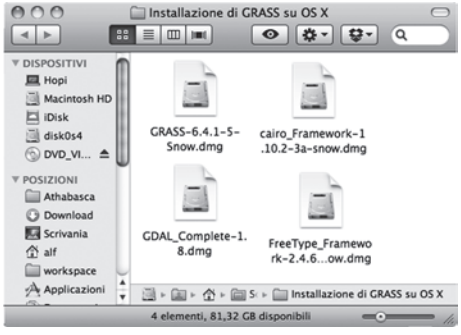

Figura 6.1. I file necessari all'installazione di GRASS in Mac OS X

[2] http://www.kyngchaos.com/software/grass

Dal sito *www.kyngchaos.com/software/frameworks* si scarichino i pacchetti FreeType GDAL-complete e Cairo facendo attenzione a scaricare la versione giusta per la versione di Mac OS X desiderata. Si scarichi quindi il *dmg* GRASS dalla pagina *www.kyngchaos.com/software/grass*. A questo punto si avranno a disposizione gli elementi necessari all'installazione di GRASS.

6.1.8.2. Installazione dei pacchetti

Per ogni pacchetto la procedura di installazione è la stessa. Visto che è necessario seguire un ordine nell'installazione dei pacchetti, si inizi con il pacchetto FreeType facendo doppio click sull'icona del *dmg* FreeType. Si aprirà una finestra che contiene diversi file, tra cui la licenza d'uso e un file *pkg*. I file *pkg* è il vero e proprio pacchetto software che viene gestito dall'applicazione installer di Mac OS X. Un doppio click sul file *pkg* avvierà automaticamente l'installer e una procedura guidata accompagnerà l'utente attraverso i passi di accettazione della licenza d'uso, le informazioni sullo spazio disco che verrà occupato, l'inserimento della password di amministratore e l'installazione del pacchetto. L'ultima schermata dell'installer, insieme a un segnale acustico, comunica che l'installazione è andata a buon fine.
Si segua la stessa procedura per GDAL-complete (che include anche UnixImageIO e Proj4) e in seguito per il pacchetto Cairo.
Se si vuole interfacciare GRASS con il software di statistica R, a questo punto è necessario installarlo seguendo le indicazioni fornite nel sito web ufficiale di R: www.r-project.org/. Si evidenzia che questo passo è opzionale.
Dopo aver installato i *framework* è ora possibile installare GRASS. Aprendo il *dmg* di GRASS si nota che ci sono due pacchetti: GRASS e GRASS-GDAL Plugin. Si installi prima GRASS con la stessa procedura usata per i pacchetti precedenti. Per concludere l'installazione di GRASS, si installi il GRASS-GDAL Plugin dal pacchetto che si trova nella finestra aperta precedentemente dal *dmg* di GRASS.

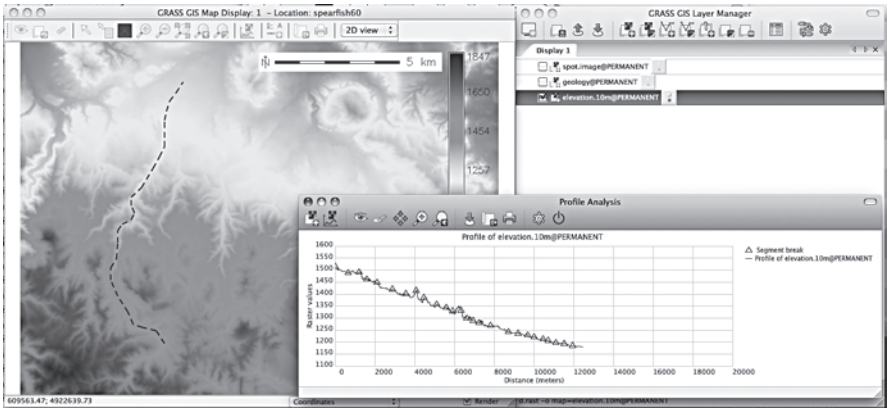

Figura 6.2. GRASS GIS in azione su Mac OS X, con la visualizzazione del modello digitale di terreno dei dati di esempio Spearfish location

Per avviare il programma si cerchi l'icona di GRASS nella cartella *Applicazioni* oppure si vada in Spotlight (la lente di ingrandimento a destra della barra nella parte superiore del desktop) e si digiti "GRASS". Spotlight troverà l'applicazione. Al primo avvio occorrerà attendere circa un minuto per permettere a GRASS di trovare i componenti necessari, ma dal secondo avvio in poi GRASS partirà molto più velocemente.

6.1.9. Installazione di GRASS *su Microsoft Windows*

Per installare GRASS sui sistemi operativi Microsoft Windows si dispone di tre possibilità. La soluzione più semplice per chi vuole usare GRASS e avere i dati di esempio scaricati nel processo di installazione è quella dell'installer standalone mantenuto da Nielsen e Kudrnovsky. Per installare GRASS con questo metodo bisogna scaricare l'installer dalla pagina:

grass.osgeo.org/grass64/binary/mswindows/native/

dove è presente anche una buona documentazione in inglese. Una volta scaricato il file di installazione (che avrà un nome tipo *WinGRASS-X.X.X-Y-Setup.exe* dove X ed Y sono i numeri delle versioni del software e dell'installatore) è sufficiente cliccare due volte sull'icona del file e seguire le istruzioni comunicate nei diversi passaggi dall'installatore. Durante il processo di installazione è possibile scaricare i dati di esempio della Carolina del Nord e del Sud Dakota, che verranno sistemati nella directory 'GIS DataBase' presente nella cartella Documenti dell'utente. Queste due Location sono le stesse utilizzate nel libro di GRASS di Markus Neteler ed Helena Mitasova (2008). Questo tipo di installazione è chiaramente indicata per chi desidera imparare ad usare GRASS, o per corsi di GIS in laboratori informatici attrezzati con Microsoft Windows.
La seconda opzione per installare GRASS su MS Windows è rappresentata dall'installer OSGeo4W, sviluppata dalla fondazione OSGeo (www.osgeo.org). L'installer è scaricabile a questo indirizzo:

download.osgeo.org/osgeo4w/osgeo4w-setup.exe

Una volta scaricato l'eseguibile, fare doppio click sull'icona del file con la bussola, e all'avviso di sicurezza, permettere l'esecuzione del software. Questo avvierà il processo di installazione. Selezionare la prima opzione *Express Desktop Install*. Tra l'elenco dei software disponibili, assicurarsi che la checkbox di GRASS sia selezionata e quindi cliccare *Avanti*. A questo punto più barre di progressione indicheranno lo stato dello scaricamento dei singoli pacchetti necessari, lo stato di avanzamento totale dell'installazione e lo spazio disco rimanente. L'installazione necessita di una ventina di minuti, variabili a seconda della velocità della rete e del numero di pacchetti selezionati.
Al termine dell'installazione verrà chiesto di creare le icone sul desktop, selezionare l'opzione desiderata e cliccare su *Fine*.
Per selezionare le applicazioni di GRASS basterà aprire il menù a tendina che si trova in *Avvio* (o icona di MS Windows in basso a sinistra sul desktop) < *Tutti i programmi* < *OSGeo4W*.

trovi nel punto corretto. Il comando viene mostrato, mentre lo si compone, nella parte bassa dell'interfaccia; questo permette anche il salvataggio di comandi e il caricamento di quelli precedentemente salvati.

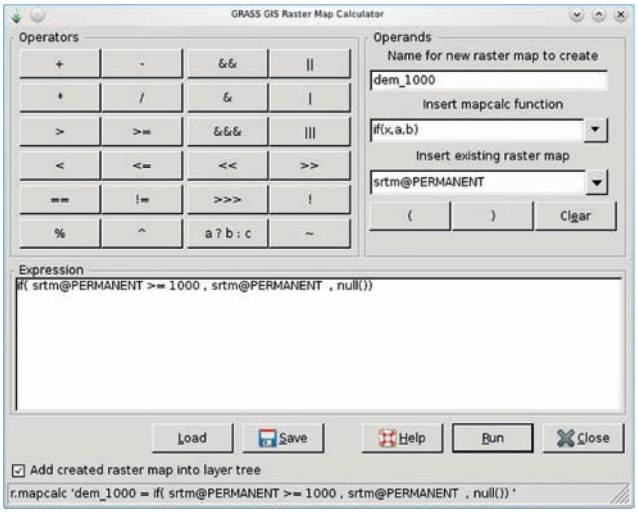

Figura 6.22. L'interfaccia grafica a *r.mapcalc*

6.3.7. Analisi raster

GRASS offre un'ampia gamma di moduli per l'analisi spaziale sui raster: si va dall'overlay al buffer, analisi di serie, del terreno, dei bacini, di "neighborhood", del percorso più corto e molto altro. Di seguito verranno elencati alcuni comandi e descritti rapidi esempi e applicazioni.

Un'analisi delle fasce di rispetto è banalmente realizzabile utilizzando le classiche funzioni di buffer implementate nel modulo *r.buffer* (*Raster* < *Buffer raster*):

r.mapcalc "urbano = if(uso_suolo2007 <=11, 1, null())"
r.buffer input=urbano output=urbano_buf100 buffer=100.0

La prima istruzione seleziona solo le zone urbane, la seconda costruisce un buffer di 100 metri attorno alle stesse.

In altri casi si vuole ottenere un unico buffer contenente anche la superficie originale nella stessa categoria (cioè le aree urbane). In questo caso si ha a disposizione *r.grow* (*Raster* < *Caratteristiche di trasformazione* < *Aumenta*) che permette di utilizzare tre diverse metriche (euclidea, manhattan o taxi e massima, si veda il manuale del comando per capire la differenza tra le tre metriche):

r.grow input=urbano output=urbano_grow100 radius=100.0

Il comando *r.neighbors* (*Raster* < *Analisi vicinanze* < *Muovi finestra*), implementa le funzioni di *neighborhood*. Mette a disposizione diverse operazioni da effettuare sui pixel contigui a quello preso in esame: media, diversità (numero di valori differenti nelle celle considerate), interspersione (percentuale delle celle con valore differente da quello del pixel centrale, più uno), massimo, minimo, mediano, moda, deviazione standard, somma, varianza. In questo esempio si calcola il range utilizzando una finestra mobile di cinque celle:

r.neighbors input=srtm output=srtm_range5 method=range size=5

Il valore delle celle del layer risultante è pari al range di valori esistente in un intorno di 2 celle a destra, a sinistra, sopra e sotto di ciascuna cella del layer in input.

L'analisi topografica[7] permette di calcolare alcuni parametri che descrivono le proprietà della superficie studiata (Hofierka et al., 2007), tra i quali:

- profili;
- elementi descrittivi dei singoli punti della superficie (pendenza, esposizione, curvature);
- analisi sui flussi e bacini idrografici;
- insolazione;
- linea di vista;
- parametri legati alla forma del terreno, ad esempio l'erosione potenziale.

Per creare il vettoriale delle curve di livello a partire da 0 metri sul livello mare con equidistanza di 100 m, in GRASS si può usare *r.contour*:

r.contour -s input=srtm output=srtm_curve_100 step=100 minlevel=0.0

Per quanto riguarda eventuali layer derivati da un modello digitale del terreno (DEM), il modulo *r.slope.aspect* (*Raster* < *Analisi terreno* < *Pendenza e esposizione*) ha come output gli indici più utilizzati quali pendenza, esposizione (calcolata in gradi partendo da est in senso antiorario 90° è il nord, 180° ovest, 270° sud e 360° est), diverse tipologie di curvature (profilo e tangenziale; per calcolare questi layer è bene che il DEM di input sia rappresentato da celle che assumono valori con virgola mobile) e derivate parziali di primo e secondo ordine. Ad esempio:

r.slope.aspect srtm slope=pendenza aspect=esposizione

Il modulo *r.shaded.relief* (*Raster* < *Analisi del terreno* < *Mappa rilievo ombreggiato*) tiene conto dell'altitudine e dell'azimut della posizione del sole, in modo tale da calcolare un layer dell'ombreggiatura dei rilievi (nel caso in cui si lavori in una location con coordinate geografiche, cioè latitudine e longitudine espresse in gradi, è necessario applicare un fattore di scala):

r.shaded.relief srtm

[7] http://www.geomorphometry.org/book

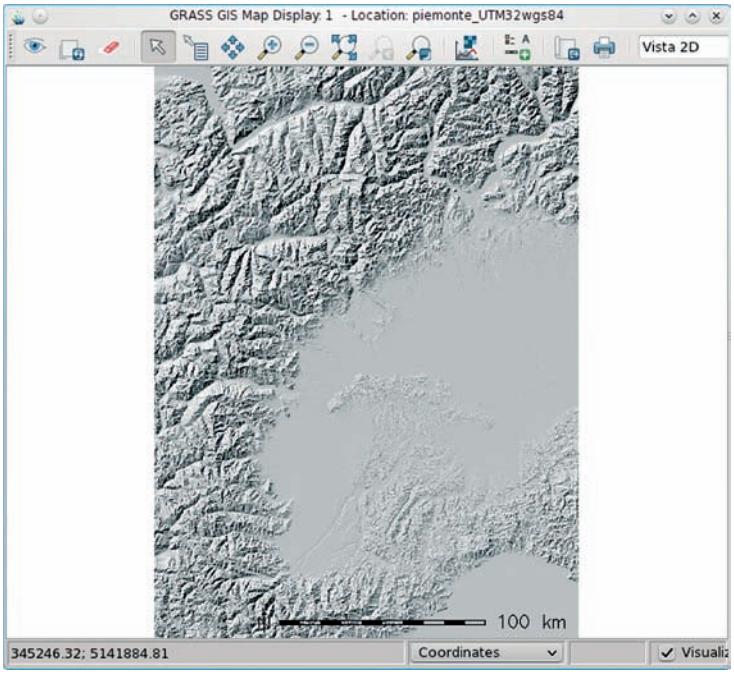

Figura 6.23. Mappa del rilievo ombreggiato (basata su SRTM)

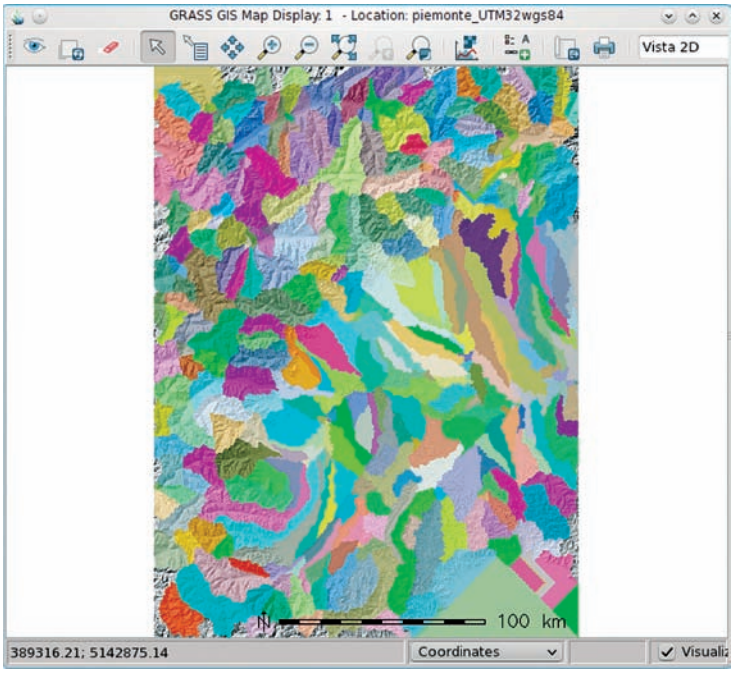

Figura 6.24. Bacini idrografici con rilievi ombreggiati come sfondo

L'analisi di flussi, reticoli e bacini idrografici è un argomento molto vasto, che non può essere esaurito in un volume come questo. In questo paragrafo verranno velocemente descritti i diversi moduli utili a questo tipo di analisi ma si rimanda anche al capitolo 7 per ulteriori esempi.

Il modulo *r.watershed* (*Raster* < *Modellazione idrologica* < *Analisi dei bacini idrografici*) elabora diverse tipologie di output. Come layer in input bisogna utilizzare un DEM e i risultati possono essere:

- il layer delle direzioni di drenaggio;
- il layer degli accumuli;
- il layer dei corsi d'acqua (che viene estratto dal modulo solo se viene fornito anche un valore di soglia).

Il modulo inoltre può anche calcolare i bacini in cui è divisa l'area in esame:

g.*region rast=srtm –p*
r.watershed srtm basin=bacini threshold=10000

Il modulo *r.flow* (*Raster* < *Modellazione idrologica* < *Linee di flusso*) serve per calcolare i layer di direzioni di drenaggio e densità di drenaggio utilizzando un diverso algoritmo rispetto a *r.watershed* (si veda il manuale del comando per i dettagli e i riferimenti bibliografici):

r.flow srtm flout=flowlines dsout=flowdensity

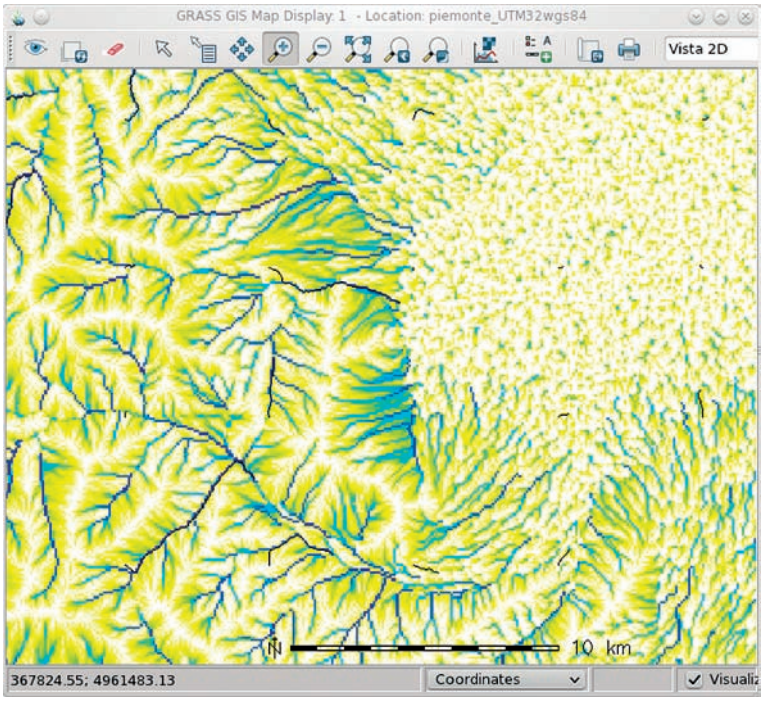

Figura 6.25. Direzioni di drenaggio e densità di drenaggio (sottoregione)

Il comando *r.sunmask* (*Raster* < *Radianza solare e ombreggiatura* < *Mappa delle ombre*) calcola la posizione del sole e la mappa delle ombreggiature. Sempre per quanto riguarda l'aspetto dell'energia solare, GRASS dispone del modulo *r.sun* (*Raster* < *Radianza solare e ombreggiatura* < *Irraggiamento e irradiazione solare*) che effettua analisi molto complesse restituendo come output diversi parametri quali: radiazione diretta, diffusa e riflessa, globale, ore di luce giornaliere e mappa della ombreggiatura. Il modulo ha un'opzione che tiene in considerazione (ove rappresentati dal modello digitale del terreno) gli elementi che ostacolano la radiazione, quali montagne e palazzi. Un esempio di calcolo per il 130-esimo giorno dell'anno è il seguente:

r.sun -s srtm beam_rad=rad_diretta insol_time=insolazione day=130

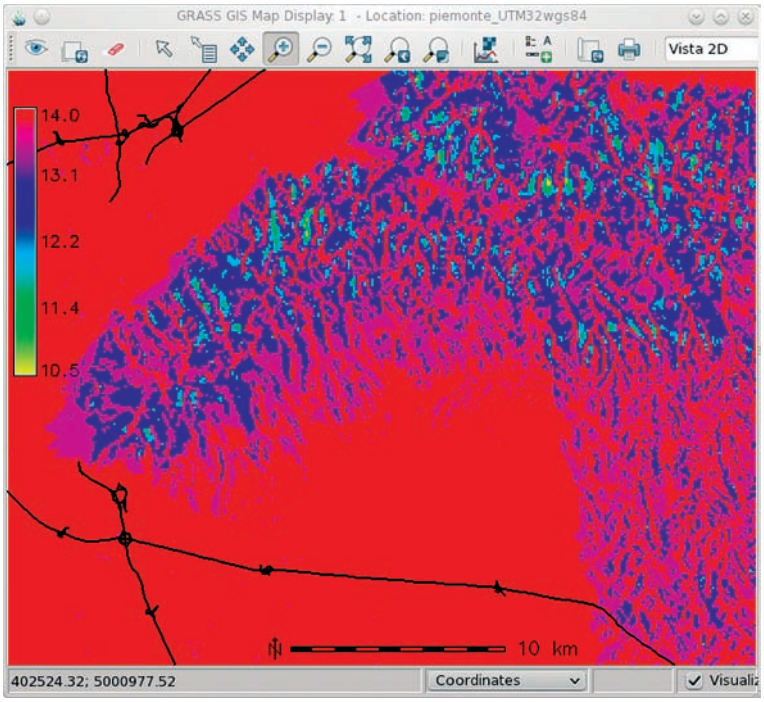

Figura 6.26. Insolazione (in ore) a sud-est di Torino

GRASS permette di lavorare con serie di dati attraverso il comando *r.series* (*Raster* < *Sovrapponi raster* < *Serie di raster*). L'input è costituito da una lista di dati raster separati da una virgola e può calcolare diversi output in funzione dell'operazione scelta. Ad esempio, media, conteggio dei valori non NULL, mediana, moda, minimo, massimo, range, deviazione standard, somma, varianza, diversità, e pendenza:

r.series
in=modis_lst_2010_01,modis_lst_2010_02,modis_lst_2010_03,modis_lst_2010_04,
modis_lst_2010_05,modis_lst_2010_06,modis_lst_2010_07,modis_lst_2010_08,
modis_lst_2010_09,modis_lst_2010_10,modis_lst_2010_11,modis_lst_2010_12
out=modis_lst_2010_min,modis_lst_2010_max method=min_raster,max_raster

L'esempio appena mostrato serve per ottenere le mappe che riportano per ogni pixel il numero del mese con valore minore ("min_raster") e maggiore ("max_raster") della temperatura della superficie della terra misurata dal sensore satellitare MODIS.

6.3.8. Visualizzazione 3D NVIZ

GRASS permette la visualizzazione dei dati, raster e vettoriali, in uno spazio 3D. NVIZ, il modulo dedicato, ha la possibilità di:

- visualizzare raster 2D come più superfici in 3D;
- alterare il posizionamento, gestire la scala delle altezze, effettuare zoom interattivi;
- modificare la posizione della luce per esaltare le ombre;
- visualizzare più di un layer vettoriale sovrapposto ad un dato raster;
- visualizzare raster tridimensionali (vedi paragrafo successivo) e volumi;
- realizzare animazioni (filmati);
- interrogare i raster;
- realizzare sezioni trasversali.

Si può ottenere un visualizzazione come quella nella figura 6.27 eseguendo dalla console il comando:

nviz elevation=elevation vector=fiumi,laghi

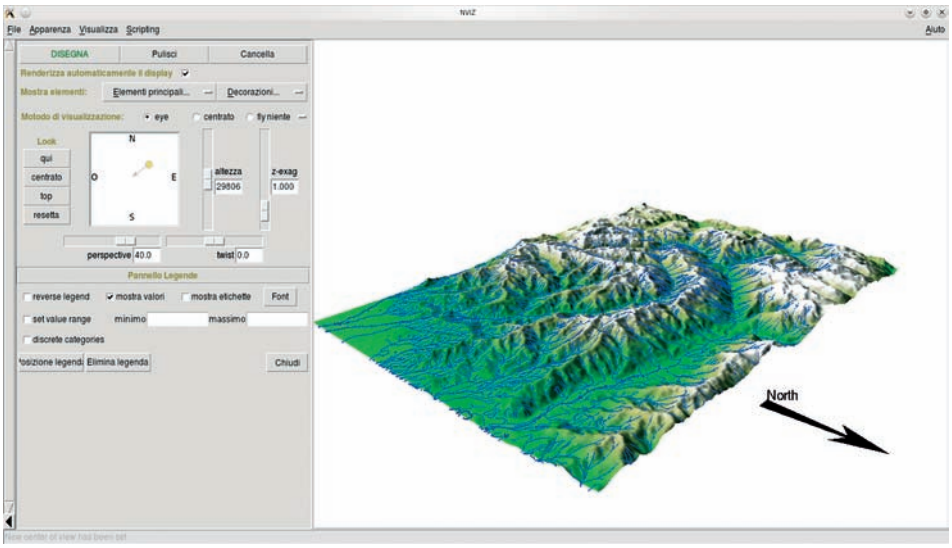

Figura 6.27. L'interfaccia grafica 3D NVIZ

6.3.9. Raster 3D

GRASS permette di lavorare anche con mappe raster tridimensionali, che sono chiamate solitamente "raster 3D" ma possono avere anche altre denominazioni tipo "voxel" e

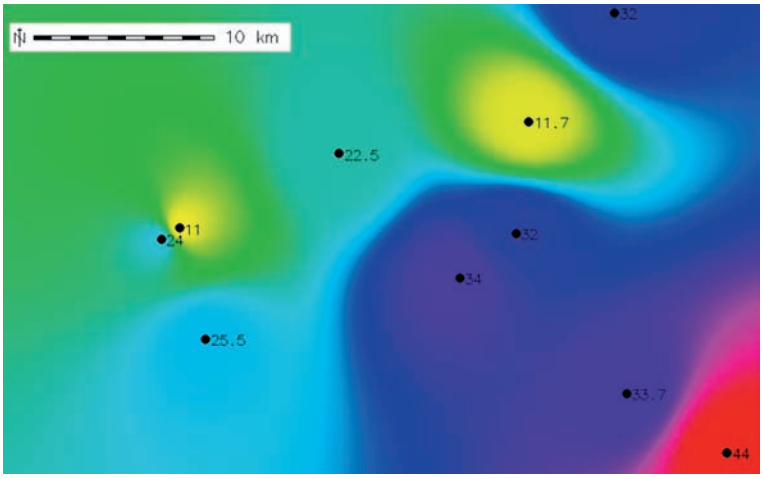

Figura 7.21. Il livello di pioggia interpolata secondo il metodo dell'inverso delle distanze pesate

Nel caso in esame:

r.contour input=pioggia_idw output=isoiete_contour step=5

Le curve isoiete, con questa sintassi, sono disegnate con passo di 5 mm di pioggia.
In figura 7.22 sono riportati l'interfaccia grafica del modulo e in figura 7.23 il layer in output, sovrapposto al livello prodotto dall'interpolazione con *v.surf.idw.*

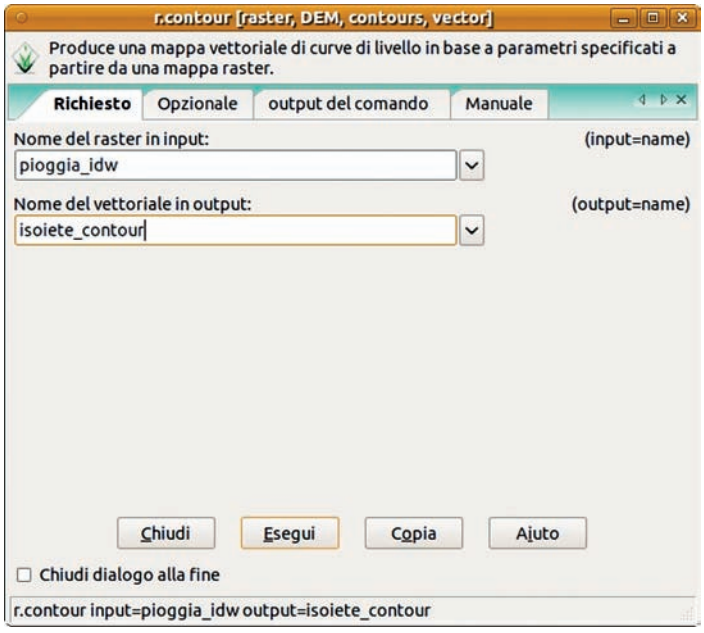

Figura 7.22. Interfaccia grafica del comando *r.contour*

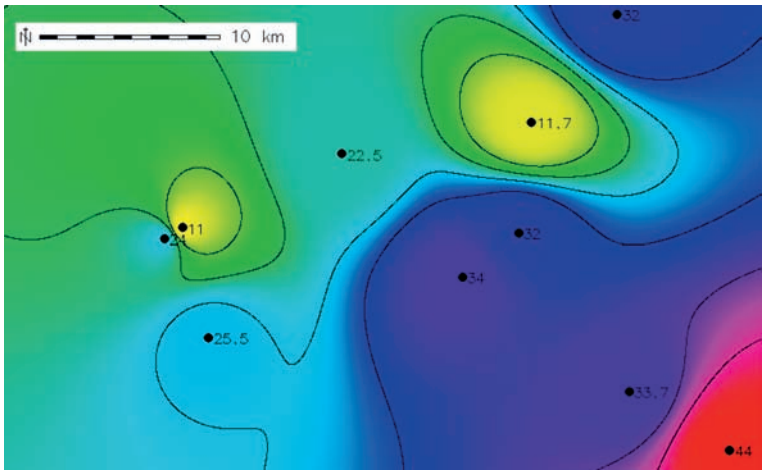

Figura 7.23. I layers prodotti in output da *r.contour*

7.3.2. Estrazione, a partire da un DEM, del bacino e del reticolo idrografico

Per l'analisi di bacino GRASS offre tutta una serie di strumenti che si possono trovare nella sezione *Raster* < *modellazione idrologica*. Esistono moduli capaci di compiere numerose operazioni: dalla simulazione di un "lago" su DEM, alla valutazione dell'erosione/sedimentazione dei suoli.

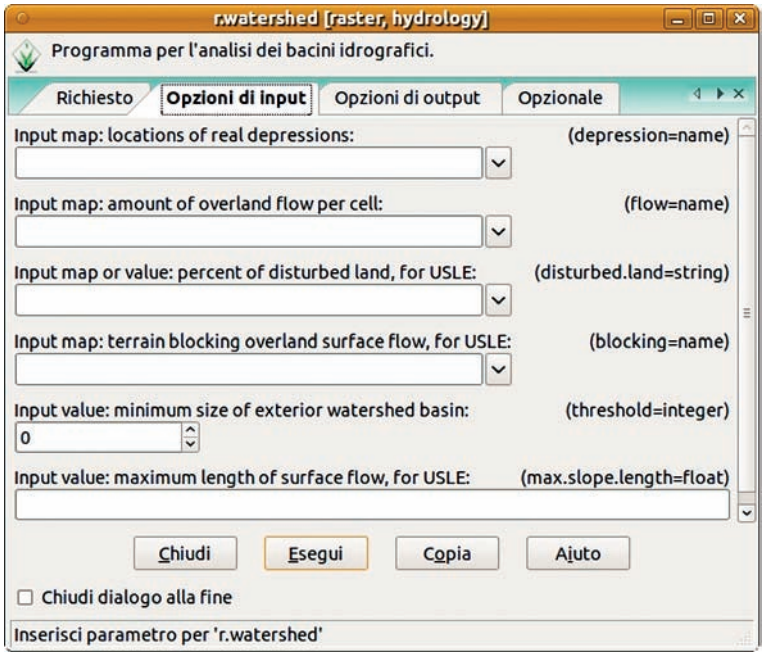

Figura 7.24. L'interfaccia grafica di *r.watershed*

8. Spatial DBMS: SQLite, SpatiaLite e RasterLite

di Alessandro Furieri

Tutti i dati geografici utilizzati all'interno del presente capitolo sono liberamente scaricabili dalla pagina dedicata a questo libro nel sito www.darioflaccovio.it.

8.1. DBMS e Spatial DBMS

L'acronimo DBMS significa *Data Base Management System* e sta appunto a indicare che si tratta di un sistema integrato per la gestione universale di qualsiasi base dati, indipendentemente dall'ambito applicativo specifico nel quale viene effettivamente utilizzato. Assai spesso, nell'uso comune, vengono chiamati semplicemente *database*.
I primi veri e propri DBMS sono apparsi sul mercato fin dagli inizi degli anni '70: si tratta quindi di una delle branche dell'informatica più antiche, di più ampia diffusione in tutti i contesti e quindi ben collaudata nel tempo.
Tutti i DBMS disponibili sul mercato si assomigliano fortemente, almeno per grandi linee, dato che tutti indistintamente derivano dal medesimo modello concettuale: cioè dal database relazionale, formalmente definito come una vera e propria teoria logico-matematica dal dr. Edgar Codd nel 1970.
Inoltre qualsiasi DBMS deve necessariamente essere (più o meno strettamente) conforme rispetto a precisi standard definiti a livello internazionale.
Come vedremo meglio in seguito, uno Spatial DBMS è semplicemente un normale DBMS che però supporta alcune estensioni particolari finalizzate alla gestione dei dati geografici e territoriali.
In altre parole, non tutti i DBMS sono Spatial DBMS: ma qualsiasi Spatial DBMS deve necessariamente essere un DBMS pienamente conforme agli standard di base comuni a qualsiasi altro DBMS di tipo meno specializzato.

8.2. Struttura di un DBMS

8.2.1. Sistemi di archiviazione basati su file

Fin dagli albori dell'informatica, i programmi e le applicazioni hanno fatto ricorso all'archiviazione dei dati su file. Seguendo l'approccio più diretto e meno sofisticato che sia possibile immaginare, ciascun singolo programma utilizza uno o più file opportunamente strutturati.
Questo approccio è naturale e diretto, ma implica un limite assai spiacevole: quel deter-

minato formato potrà essere letto e scritto solo utilizzando quel determinato programma, per cui le informazioni contenute in quel file saranno del tutto inutilizzabili e prive di senso per programmi differenti da quello originale.
Non solo: nel caso in cui occorra implementare qualche nuova funzionalità nel programma, facilmente occorrerà modificare anche il formato del file, con tutti i problemi di compatibilità facilmente immaginabili.
Infine, esiste un ultimo limite, particolarmente severo: non è possibile recuperare i dati (elaborarli, visualizzarli stamparli, ecc.), se non nei modi e nelle forme eventualmente previste dal programma corrispondente.
Se si desidera elaborare i dati in un modo non supportato (anche solo leggermente differente), occorre modificare il codice del programma: operazione evidentemente lunga, complessa e soprattutto inutilmente onerosa e costosa.

8.2.2. Principi basilari dei DBMS

I DBMS sono nati proprio per risolvere ed eliminare alla radice tutti i problemi sopra elencati. Un DBMS è uno strumento in grado di archiviare in modo sicuro e affidabile un'enorme mole di dati complessi e altamente strutturati: ed è anche in grado di selezionare ed elaborare i dati in modo molto efficiente e rapido.
Non solo: un DBMS è in grado di gestire contemporaneamente centinaia o anche migliaia di tavole differenti, ciascuna caratterizzata dalle proprie regole di formato indipendenti: molto grossolanamente, ciascuna tavola può essere considerata approssimativamente come se fosse un file distinto e separato.
Ma un DBMS è anche in grado di incrociare i dati contenuti in più tavole differenti, purché esista una qualche regola logica (per esempio, una codifica comune) che consenta di associare le diverse informazioni.
Dal punto di vista specifico del DBMS non fa alcuna differenza che si tratti di dati contabili oppure di dati sanitari o anche di dati geografici.
Per il DBMS si tratta in ogni caso di informazioni prive di qualsiasi significato intrinseco, ma formalmente ben definite e rigorosamente qualificate, che vanno gestite conformemente ad alcune regole logico-matematiche di carattere assolutamente generale valide in tutti i casi e in tutti i contesti possibili.
Sicuramente, questo approccio così astrattamente formale può apparire a prima vista assai poco naturale e per nulla intuitivo: ma quarant'anni di successi consolidati garantiscono sull'assoluta bontà delle fondamenta teoriche che stanno alla base del successo delle tecnologie DBMS.
Un DBMS non è semplicemente un sistema di archiviazione (come se fosse un insieme di normali file, per intendersi in modo poco rigoroso).
Viceversa, un DBMS è un prodotto software complesso e articolato, che non solo provvede alla memorizzazione dei dati, ma che si fa anche carico della loro elaborazione: eventualmente anche in modo totalmente autonomo e indipendente rispetto al normale software applicativo.
Alla luce di tutto quanto sopra, uno Spatial DBMS è quindi un sistema che non solo è in grado di archiviare e selezionare le informazioni geografiche, ma è anche in grado di svolgere autonomamente sofisticate elaborazioni spaziali.

8.2.3. Architettura client-SERVER

I primi DBMS apparsi sul mercato durante gli anni ’70 e ’80 avevano un’architettura tutto sommato semplice. Un unico computer centrale (di tipo mainframe o minicomputer) ospitava sia il DBMS che il restante software applicativo: quindi tutte le elaborazioni venivano elaborate dal medesimo hardware e sulla medesima piattaforma (di fatto, l’unica disponibile).
A partire dagli anni ’90 lo scenario è drammaticamente cambiato: con l’introduzione delle reti locali (LAN) e successivamente delle reti globali (Internet) moltissimi computer, magari anche di tipi e modelli diversi, hanno iniziato a operare in modo efficacemente integrato come se fossero un tutt’uno (informatica distribuita).
I DBMS si sono rapidamente adattati al nuovo scenario adottando la cosiddetta *architettura client-server*: in questo nuovo scenario il DBMS vero e proprio risiede su un’apposita macchina di adeguate dimensioni e potenza (server).
Dopodiché un numero anche molto elevato di postazioni periferiche (client) può estrarre le informazioni dal DBMS per elaborarle localmente in modo flessibile e del tutto autonomo e indipendente. La rete provvede a trasferire i dati nelle due direzioni tra server e client a seconda delle esigenze.
Questo tipo di soluzioni sofisticate copre più che adeguatamente tutte le esigenze delle medie, delle grandi e delle grandissime organizzazioni: ma richiede comunque un approccio specialistico, dato che evidentemente si tratta di sistemi abbastanza complessi per loro stessa natura.
In qualche modo questo giustifica la diffidenza che molti utenti nutrono nei confronti dei DBMS, che vengono percepiti come complicati, difficili, rigidi, poco amichevoli e poco flessibili.

8.2.4. Architettura personal DBMS

Negli ultimi anni stiamo assistiamo a una significativa inversione di tendenza: di fatto un PC qualsiasi degli anni 2000 offre molta più potenza di calcolo e capacità di archiviazione su disco di quella disponibile all’epoca sul più poderoso mainframe degli anni ’70.
Ecco quindi che ha iniziato a nascere tutta una nuova generazione di DBMS per uso personale, non basati su architettura client-server.
Si noti bene: si tratta comunque di veri e propri DBMS assolutamente standard, strettamente analoghi ai ben più sofisticati fratelli maggiori, ma assai meno complessi in termini strutturali e quindi di utilizzo più semplice e immediato.
Naturalmente alcuni di questi personal DBMS di nuova generazione supportano adeguatamente anche i dati geografici: sono finalmente arrivati sul mercato i personal/spatial DBMS. In questo modo, l’utente è oggi in grado di scegliere lo strumento ottimale in relazione al compito specifico.
Quando è necessario interagire strettamente con altri utenti, magari disseminati in più sedi distanti tra di loro, è chiaramente consigliabile l’uso di prodotti basati su sofisticare architetture client-server.
Ma in tutti i numerosi casi in cui invece sia richiesto uno strumento potente, ma assai semplice da utilizzare e quando le interazioni con altri utenti sono modeste e/o abbastan-

za lasche e indirette, allora l'uso di un prodotto più semplice con architettura personal diventa ovviamente vantaggioso.

8.3. Il linguaggio SQL

8.3.1. Caratteristiche generali

Tutti i DBMS funzionano in base a un preciso linguaggio formale denominato SQL – *Structured Query Language*.
In teoria un DBMS dovrebbe supportare esattamente il medesimo linguaggio SQL utilizzato da qualsiasi altro DBMS, dato che si tratta di uno standard definito a livello internazionale. Purtroppo, nella pratica si verifica che ciascun prodotto adotta una propria specifica variante dialettale del linguaggio: resta comunque il fatto che indiscutabilmente tutti i diversi DBMS tendono ad assomigliarsi fortemente, almeno per grandi linee generali. E questo va ovviamente a tutto vantaggio della compatibilità reciproca.
Probabilmente alcuni lettori avranno già una qualche conoscenza relativa a linguaggi di programmazione come Python, Java, PHP, VisualBasic, ecc.: tutti questi sono esempi di linguaggi cosiddetti *imperativi*.
SQL al contrario è un esempio di linguaggio *dichiarativo*.
Non è mai possibile specificare minuziosamente "come" deve essere eseguito un determinato compito, occorre invece indicare chiaramente "cosa" si intende ottenere. Sarà poi compito del DBMS stesso identificare la strada migliore e più efficiente per ottenere quel determinato risultato.
Sostanzialmente SQL è molto più semplice dei linguaggi di programmazione veri e propri; ma ovviamente imparare a usare SQL in modo efficace richiede pur sempre un discreto sforzo di apprendimento.
Naturalmente pretendere di spiegare SQL in poche pagine è un'impresa impossibile, ci limiteremo quindi ad esaminare velocemente le principali caratteristiche in modo tale da fornire al lettore alcune semplici nozioni di base, giusto per permettergli di afferrare i concetti principali e di svolgere in seguito alcuni semplici esercizi fondamentali.

Definire la struttura dei dati

Si consideri un semplice esempio assolutamente banale e cioè di volere creare una rubrica telefonica:

```
CREATE TABLE rubrica(nome VARCHAR(60) NOT NULL,
 cognome VARCHAR(60) NOT NULL, telefono VARCHAR(20));
```

Questa semplice dichiarazione consente di creare una nuova tavola all'interno del DBMS: la tavola conterrà tre colonne di tipo alfanumerico (VARCHAR) con la rispettiva lunghezza massima ammessa. Le due colonne "nome" e "cognome" sono obbligatorie, cioè devono necessariamente contenere un valore, dato che è stato specificata una clausola NOT NULL.
Invece la colonna "telefono" è opzionale e può anche non avere alcun valore assegnato. Dopo tutto è comunque desiderabile potere inserire un nominativo anche se ancora non è noto il numero telefonico, che potrebbe eventualmente essere inserito in un secondo momento.

Si noti come ciascuna singola istruzione SQL debba essere correttamente terminata indicando il segno punto e virgola [;].

Inserire i dati

```
INSERT INTO rubrica (nome, cognome, telefono)
 VALUES ('Angelo', 'Rossi', '06 12345678');
INSERT INTO rubrica (nome, cognome, telefono)
 VALUES ('Luigi', 'Bianchi', '02 87654321');
INSERT INTO rubrica (nome, cognome, telefono)
 VALUES ('Maria', 'Verdi', NULL);
INSERT INTO rubrica (nome, cognome)
 VALUES ('Francesca', 'Dall''Olio');
```

A questo punto, si può procedere al popolamento della tavola appena creata. Si noti come qualsiasi stringa alfanumerica deve essere racchiusa tra apici ['']; nel caso in cui il carattere apice appaia all'interno della stringa stessa, occorrerà ripeterlo due volte [Dall''Olio].
Per la colonna "telefono" la dichiarazione di formato consente che possa anche rimanere vuota. Questo risultato si può ottenere in due modi diversi: specificando il valore convenzionale NULL oppure evitando completamente di fare riferimento alla colonna corrispondente.

Modificare i dati

```
UPDATE rubrica SET telefono = '051 44556677'
 WHERE nome = 'Francesca' AND cognome = 'Dall''Olio';
UPDATE rubrica SET telefono = '055 767676'
 WHERE nome = 'Luigi' AND cognome = 'Bianchi';
```

Come era stato preannunciato, i numeri telefonici mancanti possono essere aggiunti in un secondo momento, così come è possibile correggere o modificare qualsiasi informazione quando è necessario.

Eliminare i dati

```
DELETE FROM rubrica WHERE nome = 'Luigi'
 AND cognome = 'Bianchi';
```

Ed infine è anche possibile eliminare definitivamente qualsiasi informazione che non sia più effettivamente utile.

Modificare la struttura dei dati

```
ALTER TABLE rubrica ADD COLUMN data_nascita DATE;
```

Qualora si rendesse indispensabile, è addirittura consentito modificare in un secondo momento persino la struttura stessa della tavola (per esempio, in caso di ripensamenti e/o di nuove esigenze inizialmente non previste).