Università
Ca'Foscari
Venezia

Università degli Studi di Venezia
Dipartimento di Scienza Ambientali, Informatiche e Statistiche

Corso di Laurea in Informatica

Tesi di laurea

Implementazione del modello RUSLE attraverso plugin per QGis

Titolo provvisorio

Candidato:
Matteo Scarpa
Matricola 845087

Relatore:
Claudio Silvestri

Anno Accademico 2016–2017

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Dedicata a tutti quelli che mi sono stati vicini nelle giornate no.

Indice

Introduzione

In questa tesi viene spiegato come è stato realizzato un plugin per QGis per il calcolo del terreno eroso avendo dei dati geomorfici legati al territorio scelto. Questi dati comprendono la conformazione del terreno, le precipitazioni registrate nell'area, il tipo di copertura della vegetazione e il numero di anni per cui questa previsione viene eseguita.

Questo plugin è stato scritto in Python e implementa il plugin ipotizzato nella tesi [Zen13] che risulta essere indipendente e utilizzabile per il calcolo su una qualunque area geografica di cui si hanno i dati richiesti. In oltre il sistema risulta indipendente dalle dimensioni dell'area geografica selezionata. È infatti possibile inserire un'area geografica di una qualsiasi dimensione per procedere con l'elaborazione. Da questo punto di vista il programma può procedere con aree geografiche grandi quanto continenti ma non è detto che il modello matematico si comporti correttamente con aree sufficientemente grandi o piccole rispetto alla precisione dei dati acquisiti.

Capitolo 1

Descrizione dell'argomento

1.1 Problema da risolvere

Lo scopo di questa tesi è quello di calcolare e visualizzare in modo corretto l'erosione del territorio nell'area geografica scelta. Come già accennato nell'introduzione, si può scegliere qualunque area geografica superiore a un metro quadro[1].

Il modello viene definito col nome ***Rusle***[2] che altro non è che una serie di equazioni che permettono di calcolare la percentuale di suolo perso in un determinato periodo di tempo in base a dei parametri passati dall'utente.

Il problema può essere descritto nell'equazione

$$A = R * K * LS * C * P \tag{1.1}$$

in cui

- **A** corrispondente alla perdita di suolo annua
- **R** erosività delle precipitazioni
- **K** erodibilità del suolo
- **LS** rapporto tra la lunghezza e la pendenza
- **C** fattore di copertura del suolo

[1] Dimensione minima sotto la quale i file formato raster non possono andare per come sono strutturati e perché altrimenti l'errore del dato renderebbe inutilizzabile il dato stesso

[2] **R**evised **U**niversal **S**oil **L**oss **E**quation

- **P** i parametri delle misure di prevenzione dell'erosione

Tutti questi dati sono forniti sotto forma di file di tipo raster che vengono poi elaborati per ottenere il file raster corrispondente alla perdita di suolo annuo. Fa eccezione il parametro P, che puo' venire omesso e rimosso dall'equazione, visto che in caso non ci siano misure di prevenzione dell'erosione il valore corrispondente al punto della matrice corrisponde a 1 diventando ininfluente nell'equazione 1.1 che diventa

$$A = R * K * LS * C \tag{1.2}$$

1.1.1 R o erosività delle precipitazioni

Il parametro R rappresenta quanto le precipitazioni in una determinata area geografica. Viene fatto attraverso dati pluviometrici. Questi ultimi non sono dati con risoluzione temporale pari a un singolo evento meteorico ma da equazioni semplificate per ottenere il parametro da i dati di precipitazioni annue.

$$R = \sum_{i=1}^{n} \frac{p_i}{n} \tag{1.3}$$

In questa equazione

- n corrisponde al numero di anni in cui si ha lo storico pluviometrico
- p_i corrisponde alle precipitazioni dell'i-esimo anno.

Questo permette una approssimazione utile per l'applicazione del modello ***Rusle***.

1.1.2 K o erodibilità del suolo

Il parametro K rappresenta quanto il terreno puo' essere eroso. Questo dipende molto da come strutturato il terreno, dalla sua composizione geologica e da alcune caratteristiche fisiche e chimiche dello stesso.

Questo tipo di dato non è disponibile per aree geografiche di grosse dimensioni per cui è stata fatta una regresione matematica su dati sulla tessitura del terreno dell'area interessata. Questo ha permesso la creazione di un modello matematico descritto dall'equazione di Romkens che permette la creazione di una stima più precisa (per quanto riguarda i nostri scopi) dell'erodibilità del suolo.

$$K = 0,0034 + 0,0405 * \exp -0,5 \left(\frac{(\log(D_g) + 1,659)^2}{0,7101} \right) \tag{1.4}$$

In cui D_g è il diametro medio pesato rispetto alla classi tessiturali dominanti in mm. La formula è comunque rispettata anche se i dati sono ottenuti tramite regressione matematica. Questo permette di non avere differenze notabili tra un K misurato e un K ottenuto tramite regressione.

1.1.3 LS rapporto tra lunghezza e pendenza

LS rappresenta un rapporto tra lunghezza e pendenza a meno di due parametri $m = 0,4$ e $n = 1,3$, che vengono definiti per convenzione.

$$LS = \left(\frac{A}{22,13} \right)^m \left(\frac{\sin(\alpha)}{0,0896} \right)^n \tag{1.5}$$

L'equazione (1.5) viene modificata nella (1.6) per ottenere una equazione che meglio rappresenti superfici con elevata complessità topografica e/o alta variabilità di pendenze in modo da essere più coerente con la granularità dei dati ottenuti.

$$LS = \left(\frac{flowacc * cellsize}{22,13} \right)^{04} \left(-1,5 + \frac{17}{1 + e^{2,3-6,1*\sin\alpha}} \right) \tag{1.6}$$

In cui

- *flowacc* corrisponde all' UCA o numero di pixel tramite i quali l'acqua defluisce nell'area precisa di pixel
- *cellsize* dimensione dei pixel
- α pendenza dell'area

1.1.4 C o fattore di copertura del suolo

Il fattore C è un rapporto tra la perdita di suolo in precise condizioni di copertura e la perdita di suolo nel caso il terreno sia privo di qualunque protezione. Attualmente sono state fornite quarantaquattro classi di copertura di terreno con altrettanti valori del fattore C.

È possibile anche utilizzare i dati provenienti da sensori di tipo AVHRR[3] con cui si calcola l'indice NDVI[4] che varia tra 1 e -1

[3] Advance Very High Resolution Radiometer
[4] Normalized Difference Vegetation Index

$$C = \exp\left(-\alpha \frac{NDVI}{\beta - NDVI}\right) \tag{1.7}$$

1.1.5 P o prevenzione dell'erosione

Fattore rappresentante misure di controllo quali sistemi di drenaggio, terrazzamenti e sistemi specifici di coltivazione. Questo viene rappresentato sotto forma di un coefficiente a dimensionale compreso tra 0 e 1.

1.2 Possibili risoluzioni

La tesi [Zen13] propone quattro sistemi per applicare e visualizzare l'applicazione del modello RUSTLE per l'area di terreno indicato.

Tutti questi sistemi per la visione e applicazione del modello RUSTLE sono stati vagliati e valutati secondo le nostre esigenze. Tutte questi sistemi hanno dei pregi e dei difetti e sono stati segnalati in base alle nostre necessità

- *Ripetibilità* il sistema scelto deve essere facilmente ripetibile.
- *Veloce* il sistema scelto deve eseguire le operazioni per l'elaborazione e la visualizzazione deve rimanere in tempi ristretti anche se la mole di dati risulta enorme.
- *Indipendente* il sistema scelto deve essere indipendente dal sistema operativo, in modo da poter essere utilizzato su qualunque dispositivo[5].
- *Semplice* il sistema non deve avere pre-requisiti di competenze o capacità specifici in modo da permettere a teoreticamente chiunque di applicare il modello ***Rusle***.
- *Automatizzabile* il sistema scelto deve essere automatizzabile in modo da poterlo lanciare attraverso un sistema automatizzato senza la presenza costante di intervento umano.
- *Visuabilizzabile* il sistema deve avere un output dati facilmente leggibile per l'essere umano attraverso il sistema stesso o un software specifico.

[5] Si considerano solo computer, i dispositivi mobili non sono *espressamente* richiesti

1.2.1 Software di grafic design

Attraverso i classici programmi di creazione di ambienti virtuali è possibile visualizzare l'ambiente e le modifiche dell'ambiente in base ai dati raccolti. Questo permette di creare una rappresentazione realistica e molto dettagliata dell'ambiente preso in esame ma richiede una grossa quantità di tempo e notevoli sforzi, senza contare che necessità di un esperto di modellazione 3D che si occupi del lavoro.

- *Visuabilizzabile* Molto facile da visualizzare e navigare. Questo risulta il migliore sistema esaminato per vedere gli effetti dell'erosione nel terreno preso in esame
- *Non ripetibile e complesso* Il sistema prevede che l'utente abbia una buona conoscenza di software di modellazione 3D e deve calcolare tutta l'erosione in modo autonomo.
- *Lento e non automatizzabile* Il sistema prevede la presenza di uno o più persone che sviluppino una mappa 3D. Anche l'utilizzo di una mappa 3D pre-esistente comunque porta il lavoro a tempi superiori a venti minuti e risulta infattibile senza la presenza della persona.

1.2.2 Programmazione

Attraverso alcuni linguaggi di programmazione è possibile calcolare il raster di output. Per esempio è possibile farlo con Mathlab, R o Python. La differenza tra i linguaggi è sostanzialmente la velocità di esecuzione del codice e quanto il codice è mantenibile.

- *Ripetibile e automatizzabile* Una volta realizzato il sistema è pronto per essere utilizzato in modo automatico e ripetibile un numero infinito di volte.
- *Non indipendente* In base al linguaggio scelto e a come è stato compilato il programma è possibile che questo richieda dei prerequisiti, sotto forma di librerie di terze parti o macchine virtuali per il loro funzionamento, creando così dipendenze esterne non desiderabili.
- *Forse visuabilizzabile* La parte più difficile di questo sistema. Ci sono due strade possibili: l'utilizzo di un visualizzatore esterno a cui affidiamo il compito di visualizzare i dati oppure sviluppiamo anche un sistema di visualizzazione

personalizzato cercando di tenere al minimo le dipendenze con librerie di terze parti.

Questo sistema riesce a essere quello più adatto alle nostre esigenze ma ci vincola ad un numero variabile di dipendenze a progetti esterni su cui non abbiamo nessun controllo aumentando in modo esponenziale il tempo di mantenimento del codice provocando un aumento dei costi di mantenimento.

1.2.3 Desktop QGis

E' un software che permette di visualizzare e elaborare raster e altri formati di file geomorfici. Molto usato in ambito industriale e accademico, supporta, nativamente o attraverso plugin installabili, la quasi totalità dei formati di file geomorfici esistenti. In oltre permette, attraverso codice Python, C e C++, di estendere le funzionalità del programma stesso.

I vantaggi di un plugin per QGis sono i seguenti:

- *Ripetibile* una volta scritto il plugin risulta riutilizzabile semplicemente cambiando i dati.

- *Automatizzabile* Se scritto correttamente il plugin può effettuare la task assegnata in modo automatico ricevendo i dati da un altro processo.

- *Visualizzabile* QGis dispone di un intero motore di renderig integrato che permette di visualizzare i dati in modo semplice e, con l'aggiunta di plugin, è anche possibile applicare i dati su una mappa, mostrando in modo più efficace i dati geo referenziati

- *Multi piattaforma* QGis è installabile sulla totalità dei desktop pc e su molti server. In oltre, se scritti correttamente, anche i plugin per QGis funzionano correttamente su tutte le piattaforme e alcuni tipi di server.

In oltre QGis risulta essere la prima scelta, come software, per l'analisi di dati geomorfici in ambito scientifico, per cui risulta naturale la scelta di sviluppare un plugin per questo applicativo.

1.3 QGis

QGis si definisce come "Un Sistema di Informazione Geografica Libero e Open Source"[Sitb] ovvero è un software che gestisce, elabora e visualizza dati geomorfici e georeferenziati, completamente gratuito e modificabile in ogni sua parte(è limitato dalla licenza utilizzata nello sviluppo del software [Sita]).

Questo permette uno sviluppo continuo e slegato da esigenze commerciali evitando così aggiornamenti fatti solo per motivi economici, obblighi d'uso di formati proprietari e abbandoni di sviluppo in caso di fallimento della ditta. Ogni individuo dotato di pc, connessione a internet e conoscenze di programmazione è in grado di aggiungere, rimuovere, migliorare funzionalità del programma e mandarle alla community. Quest'ultima dovrà quindi approvare le modifiche e applicarle al progetto e, così, dall'aggiornamento successivo, le modifiche saranno applicate su QGis.

In questo progetto sono state utilizzate delle componenti del motore di QGis in modo da poter dedicare maggior attenzione allo sviluppo dei componenti realmente importanti del plugin, in modo da poter garantire il corretto funzionamento e la compatibilità con QGis.

In particolare sono stati utilizzati:

- *Componenti di input e output* È il sistema che rende possibile interfacciare il plugin scritto con il normale sistema di input/output di QGis ottenendo così un'interfaccia unica e coerente indipendentemente dal sistema operativo utilizzato. In oltre permette di visualizzare sempre la finestra che si cerca di visualizzare in quanto utilizza librerie grafiche già installate sul dispositivo in uso in quanto già necessarie per il normale funzionamento di QGis.

- *Sistema di supporto di formati di file* Collegandosi al sistema di supporto dei formati di file è possibile utilizzare tutti i formati supportati dall'installazione di QGis. Questo, a sua volta, è estendibile con l'utilizzo di altri plugin che aggiungono formati di file utilizzabili in lettura e scrittura che, a loro volta, possono essere utilizzati da gli altri plugin installati, tra cui quello descritto in questa tesi.

- *Motore di calcolo* QGis comprende un pacchetto di operazioni standard sui file raster. Questo pacchetto risulta molto pratico in quanto ottimizzato per le

esigenze di QGis risulta particolarmente efficiente a elaborare i raster.

- *Sistema di rendering a video dei dati* Il sistema permette di visualizzare i file a schermo in versione grafica. Questo può essere fatto con l'output dei dati ottenuti dal plugin e sovrapposto a mappe fisiche, rendendo così più immediata la visualizzazione.

- *Api di supporto per i plugin* Il sistema con cui si collega a QGis. Permette l'utilizzo da parte del plugin delle funzionalità di QGis senza difficoltà e fissando delle convenzioni per il funzionamento e i sistemi di comunicazione. Questo porta alla possibilità di combinare diversi plugin per ottenere automatismi più complessi di quelli di partenza.

1.3.1 Descrizione dei tipi di dati supportati

Per applicare il modello ***Rusle*** è necessario passare al programma i dati necessari per l'applicazione del modello matematico. In linea generale esistono svariati sistemi per passare i dati a un programma che si dividono in tre categorie:

- *Input dell'utente* L'utente utilizzatore utilizza una interfaccia grafica o un terminale per inserire i dati uno a uno

- *File o database* Viene utilizzato un file che contiene i dati completi pronti per la elaborazione diretta

- *Flusso diretto* L'elaboratore riceve da un sensore o un dispositivo i dati mano a mano che questi risultano disponibili a quest'ultimo

L'input diretto di dati all' interno dell'interfaccia grafica o del terminale risulta fattibile nel caso i dati da inserire siano pochi. Questo non è possibile nel nostro caso. Infatti i dati che possono essere forniti possono essere delle matrici contenenti i dati geomorfici divisi per celle di dimensione un metro per un metro. Questo comporta che per un area di un kilometro quadrato ci siano un miglione di celle da inserire manualmente per ogni parametro del'equazione (1.1).

L'input attraverso flusso di dati in questo problema non è fattibile. Questo perchè non è possibile avere questi dati da sensori. Come spiegato nel paragrafo 1.1, i dati sono legati a un periodo di tempo o alla struttura del territorio ottenendo così solo dati "statistici" o "estrapolati".

Quindi per il nostro problema i dati vengono forniti sotto forma di file o di database in quanto necessitano di una pre elaborazione e di una formattazione in base allo standard scelto. Questo sistema è utile anche perchè permette di riutilizzare gli stessi dati per elaborazioni successive, cosa non prevista negli altri due casi.

I formati di dati possono essere divisi in tre macrocategorie in base alla codifica utilizzata:

- File Vettoriali
- File Raster
- DBRMS con estensione spaziale [6]

In base alla tipologia di dati puo' risultare più efficiente una operazione in vece che un altra. Quindi, per alcune operazioni, QGis converte i dati nell formato appropiato in modo che sia sempre possibile fare tutte le operazioni disponibili nel modo più efficiente possibile, tenendo al minimo le approssimazioni dei dati nella conversione. Sarebbe consigliato quindi di utilizzare direttamente il file nella corretta codifica in modo da limitare i possibili errori causati dalla conversione da e in formati divrsi.

L'utilizzo di un unico formato di file sia per l'input che per l'output limita quindi la propagazione degli errori in modo da ottenere dati approsimativamente più concordi col modello applicato evitando così di dover applicare fattori di correzione all'output dei dati.

1.3.2 La struttura di QGis

QGis è un programma avanzato con molte funzionalità che possono essere suddivise in alcune macro-categorie che sono indipendenti dal tipo di dato o file utilizzato ed è possibile utilizzarli in contemporanea.

Le funzionalità standard di QGis sono gestite attraverso dei moduli che costituiscono l'installazione di base del programma. Questi moduli sono forniti dalla community sotto forma di un unico pacchetto (QGis per l'appunto) che viene diviso in applicativo server e applicativo client. Per come è stato impostato il progetto è stato preso in considerazione solo l'installazione client e successivamente è stato modificato per poter aggiungere il supporto ad automatismi e applicativi server. I moduli sono scritti in C e C++ e vengono modificati nel repository ufficiale di QGis. L'utente comune e quello

[6] Database resource managements

esperto non hanno mai bisogno di modificare questi moduli, al massimo solo le loro configurazioni. Per questo motivo è stato creato un sistema di API per il supporto di implementazione di funzionalità aggiuntive.

Le funzionalità aggiuntive vengono implementate attraverso plugin dipendenti da questi moduli di base o altri plugin e script per poter funzionare correttamente. Questo permette l'aggiunta di qual si voglia funzionalità non standard senza andar a toccare i moduli pre compilati dell'installazione. Per questo motivo è stato neessario scegliere un linguaggio di programmazione che permetta un corretto interfacciamento con i moduli C e C++ e risulti di facile mantenimento. Sul sito ufficiale di QGis [Sitb] viene spiegato che l'utilizzo di python per i plugin di QGis è dato dalla diffusione del linguaggio e dal suo ampio uso nel campo di *Data Analisys*. In oltre python si interfaccia correttamente coi i file C e C++ senza portare nessuna modifica a questi ma facendoli semplicemente importare nello script interessato.

Di seguito presento quindi i moduli principali di QGis e i tipi di plugin più diffusi.

Rendering grafico dei dati

QGis ha la capacità di realizzare svariate visualizzazioni grafiche dei dati inseriti o calcolati. Questo permette la realizzazioni di immagini ad alta definizione dei dati inseriti e ne permetta anche l'esportazione nei formati più comuni di immagini. Risulta essere anche configurabile dal punto di vista cromatico e di qualità dell'immagine in output ma può essere di difficile lettura senza un riferimento o delle indicazioni in base alla posizione dei dati o alla conformazione geologica del territorio interessato dai dati.

Realizzazione mappe

In presenza di dati georeferenziati QGis permette di sovrapporre la rappresentazione grafica dei dati alla mappa corrispondente alle coordinate geografiche. Fondamentalmente utilizzato per correlare i dati alla mappa geografica corrispondente. Molto utilizzato per visualizzare il traffico diviso per aree di una città o i movimenti dei fronti temporaleschi.

Molto importante anche la possibilità di fare una serie di "mappe" in sequenza per mostrere la

Creazione, elaborazione e conversione dati

QGis presenta due moduli per l'elaborazioni di dati:

- **Class** modulo standard di QGis per i calcoli aritmetici elementari e alcuni calcoli statistici di base
- **Grass** modulo aggiuntivo di QGis per i calcoli avanzati e regressioni. Contiene anche modelli previsionali e probabilistici.

Analisi dati

Quando si lavora con un insieme di dati non è detto che questi siano tutti corretti o non abbiano un livello di approssimazione. All'interno di QGis sono disponibili, di base, un insieme di funzionalità atte a calcolare la bontà delle approssimazioni o dei dati stessi presi in esame dal programma. Questo viene fatto di base con algoritmi generici per la maggior parte dei formati ma se si usa solo un tipo specifico o si vuole una approssimazione o un modello preciso è possibile ottenere il risultato desiderato attraverso plugin di terze parti.

1.3.3 Sistema dei plugin

Per aumentare lel funzionalità di QGis e scriptare alcune operazioni sono stati creati i plugin. Sono dei programmi dipendenti da QGis che implementano funzionalità non presenti nel programma di base o estendono il supporto delle funzioni esistenti o i tipi di file supportati.

Plugin per formati di file

Vengono utilizzati per elaborare dati grezzi altrimenti difficilmente manipolabili. Un esempio possono essere i dati grezzi ottenuti da degli strumenti di misurazione con dati in output non standard. Normalmente questi file sono leggibili solo col programma abilitato alle modifiche ma, con l'utilizzo del plugin corretto, è possibile leggerlo e elaborarlo direttamente in QGis.

Pluing che alterano o aumentano le funzionalità

La reale forza di QGis. Permettono di ampliare le funzioni del programma semplificando o aumentando le funzionalità pre esisteni. Alcuni esempi sono i plugin della

famiglia GRASS che aggiungono operazioni matematiche o gli strumenti di creazione timelapse partendo da una serie di ruster temporizzati.s

Capitolo 2

Realizzazione

Il progetto è stato realizzato attraverso la divisione in sotto problemi. Questo permette la sostituzione di una componente con un'altra in modo rapido ed sicuro in quanto viene a essere modificata solo la parte del codice interessata dall'operazione e non tutte le altre componenti, che rimangono invariate. Questo permette di testare singolarmente le parti senza necessità di avere tutte le componenti già sviluppate e renderle funzionalmente indipendenti.

Questo ha portato anche alla ricerca di alcune librerie per eseguire le operazioni necessarie alle funzioni scelte in modo da poter rispettare lo schema di sotto problemi deciso precedentemente.

2.1 Divisione in sotto problemi

Quindi il problema di calcolo del quantitativo di terreno annuo perso è stato diviso prima in quattro sotto problemi a cui è stata data una soluzione attraverso quattro componenti indipendenti che dialogano tra di loro ,preferendo una risoluzione modulare del problema, rendendo così le componenti meno soggette a errori e più facili da testare. Ne consegue che i sotto problemi esaminati sono:

- Come l'utente comunica con l'applicazione e come gli passa i comandi
- Come l'applicazione riceve i dati e i formati di dati compatibili e/o supportati
- Come vengono elaborati i dati in modo da seguire il modello ***Rusle***

- Output dei dati seguendo metodi che permettano un facile accesso ai risultati sia all'utente che alla macchina.

Questa strutturazione permette il riciclo del codice di altri progetti che hanno affrontato lo stesso problema o sotto problema e favoriscono l'aggiunta di funzionalità e l'aggiornamento delle parti.

2.1.1 Interfaccia grafica

Al momento della progettazione dell'applicazione è stato necessario pensare come realizzare l'inserimento dei dati da elaborare da parte dell'utente. Si è quindi pensato alla tipologia di utente che può utilizzare il nostro plugin facendo attenzione alle sue competenze e al suo background. Questo ha portato a ridurre la soluzione in due macro categorie:

- Interfaccia e plugin sviluppati per **terminale**. È la soluzione di più facile dal punto di vista dell'implementazione anche se non è quella più pratica per l'utente. Infatti, anche se questo sistema di input e output favorisce una modalità *verbosa*[1] di output. Di contro però obbliga l'utente a conoscere i comandi per il terminale, quali navigazione tra file e cartelle, e i comandi per eseguire operazioni direttamente col plugin senza lanciare direttamente QGis. Queste caratteristiche portano questa soluzione a essere considerata per utenti avanzati.

- Interfaccia e plugin sviluppati per **GUI**. Questo è la soluzione solitamente utilizzata per QGis. Risulta più naturale per l'utente e molto più intuitiva dell'altra. È anche la soluzione che ti obbliga a utilizzare il maggior numero di librerie esterne e, di conseguenza, aumenta le dipendenze da queste stesse. Nel nostro caso, però, non vengono aumentate le dipendenze in quanto è possibile utilizzare le stesse librerie grafiche che QGis utilizza e quindi rientrare nelle dipendenze già soddisfatte installando QGis. Con queste ultime librerie è quindi anche possibile avere una interfaccia unica indipendente dal sistema operativo

[1] Questa modalità permette di inserire nell'output un quantitativo maggiore di informazioni solitamente utilizzate per il debugging, la creazione di componenti avanzati o comunque task per utenti avanzati.

usato invece che, come invece accade nella soluzione via terminale, avere delle differenze[2].

Per come è strutturato QGis è possibile avere entrambi implementati nel plugin ma è stato preferito dare maggiore attenzione alla componente grafica anche se è comunque possibile aprire il terminale Python interno di QGis ed eseguire dei comandi per ottenere gli stessi risultati ma senza avviare l'interfaccia grafica.

In particolare in questa applicazione è stata implementata l'interfaccia grafica in quanto risulta molto più adatta all'uso effettivo dell'applicazione.

Infatti come si vede nell'immagine 2.1 per il progetto in esame è necessario soltanto una GUI di inserimento dati in cui, in base al tipo di file inserito e al tipo di output,li elabora correttamente. Per come è stato costruito il plugin se si vuole passare i file di input è necessario passarli anche la codifica del file per permettere di scrivere o leggere correttamente il file[3].

Qt4

Le librerie grafiche per Python[4] sono molte e varie ma la scelta è stata abbastanza facile: QT versione 4.

Questa scelta è stata dettata dal desiderio di ridurre al minimo le dipendenze esterne dal plugin. Questo comporta l'utilizzo preferenziale delle librerie utilizzate da QGis per semplificare l'installazione delle dipendenze. In particolare, con QGis è stato necessario scegliere anche la versione di QT con cui sviluppare il plugin in modo da non avere versioni differenti di QT rispetto a quella utilizzata QGis. La scelta è stata fatta su QT4 ovvero la funzione supportata dalla versione 2.0 in poi di QGis. Questo permette la completa compatibilità di tutte le versioni esistenti supportate di QGis[5].

La versione QT5 che verrà impiegata per le future versioni di QGis è già interamente supportata dal plugin. Questo è stato possibile attraverso un attento sviluppo attraverso

[2] Nella risoluzione del terminale possiamo vincolare solo alcune operazioni del plugin ai comandi da noi definiti. Il sistema operativo comunque obbliga gli altri comandi necessari per le altre operazioni a parole chiave spesso non concordanti tra sistemi differenti.

[3] Questo viene fatto automaticamente dall'interfaccia grafica selezionando il file desiderato utilizzando la finestra 2.1

[4] Il linguaggio in cui è stato scritto il plugin

[5] Il plugin non è compatibile con QGis versioni antecedenti alla 2.x in quanto quelle versioni non sono più considerate stabili e non vengono più aggiornate. Questo perché quelle versioni utilizzano versioni precedenti di QT che porta ad renderizzare la grafica in modo corretto

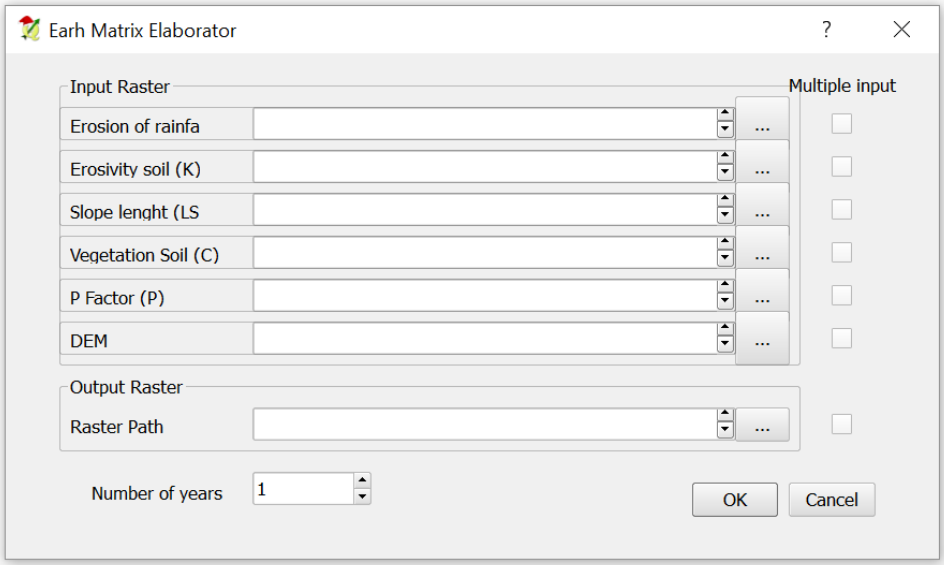

Figura 2.1: La finestre di configurazione dell'input e output del plugin. Permette di indicare il path dei file in modo grafico utilizzando la finestra standard di selezione del file del sistema operativo che si sta utilizzando.

funzioni non deprecate e controllo continuo delle nuove funzionalità di QT5 e delle modifiche che questa fa nel funzionamento complessivo in modo di essere già compatibile quando sarà necessario passare a QT5 in quanto QT4 non sarà più prerequisito di QGis venendo sostituito da QT5.

Ovviamente si poteva avere lo stesso risultato utilizzando altre librerie grafiche esterne o quelle di Python ma è stato scelto QT anche per avere una coerenza grafica tra le varie finestre che vanno ad aprirsi. Infatti se venisse utilizzata un altra libreria grafica sarebbe necessario una rielaborazione della grafica standard delle librerie in modo che si comportino coerentemente rispetto a QT e QGis per ogni sistema operativo, il che porterebbe a una complicazione del codice non necessaria[6].

2.1.2 Leggere e scrivere i dati

Il plugin prende in ingresso i file contenenti i dati necessari per l'applicazione al modello matematico descritto dall'equazione 1.1. Quindi, una volta eseguite le

[6] È considerata non necessarie in quanto non porta vantaggi al sistema e ne aumenta la quantità rendendolo più complesso da mantenere

operazioni produce l'output corrispondente all'erosione per il numero di anni dati in input.

Le operazioni di lettura e scrittura sono le operazioni più sensibili in questo applicativo in quanto non c'è modo di controllare la correttezza dei dati in input dal punto di vista fisico/geologico, solo dal punto di vista informatico di buona formattazione. Per questo motivo assumiamo da questo punto in poi che i dati passati in input siano corretti dal punto di vista geologico ma non obbligatoriamente corretto dal punto di vista della formattazione[7].

Input

Per una questione di coerenza rispetto a QGis è stato deciso di passare i dati in input seguendo gli standard internazionali, in modo che possano essere facilmente condivisi e possano interagire più facilmente con QGis stesso. Per fare ciò è stato necessario fare una ricerca sui tipi di file utilizzati per i dati georeferenziati.

Si è quindi iniziato recuperando l'elenco dei formati supportati nativamente da QGis di tipo raster[8] e l'elenco dei formati che è possibile supportare attraverso plugin esterni in modo da avere un elenco di tutti i possibili formati da supportare.

È stata quindi usata una delle funzioni del core di QGis per abbinare il formato di file ricevuto in input con quelli in memoria di QGis in modo da poter utilizzare il driver corretto per l'elaborazione. L'elenco che viene esaminato per l'ottenimento del driver si riferisce solo ai driver che permettono la lettura del file. Se per qualche ragione ci fossero dei driver di sola scrittura per qualche formato di file allora questi non apparirebbero tra i driver utilizzabili[9].

Grazie a questo sistema si è potuto evitare di dover creare un archivio dei formati disponibili e dei driver esistenti che avremmo dovuto cercare all'interno del computer in cui si esegue il plugin. Invece viene consultato il catalogo interno di QGis che viene interamente gestito dall'installazione di QGis e da i plugin che aggiungono supporto

[7] Il controllo del formato del file viene fatto in modo automatico quando il plugin cerca di caricare i file passati per indirizzo

[8] Il tipo di dati che il plugin accetta in quanto è l'unica struttura dati adatta a contenere i dati che ci servono per l'elaborazione

[9] Da tenere in considerazione quando si eseguono i comandi direttamente da terminale, in quanto lo stesso comando, in base al parametro passato, ritorna tutti i driver o solo quelli di lettura o solo quelli di scrittura

per altri formati, ottenendo così un elenco in tempo reale e senza ulteriore sforzo da parte del plugin.

In oltre, il codice di gestione per i file in input è integrato nell'interfaccia grafica in modo da rendere semplice l'inserimento di dati all'interno del programma stesso. Supporta comunque l'utilizzo attraverso il terminale di QGis anche se ne è sconsigliato l'uso ai meno esperti e a chi si approccia per la prima volta a questo plugin.

Come vedete dall'immagine il plugin apre una finestra di sistema differente in base al sistema operativo utilizzato, in quanto la gestione della scelta del file è delagato a esso. Una volta avvenuta la scelta questa finestra si chiuderà passando la stringa corrispondente al path del file aperto al campo adiacente presente nella gui principale 2.1. Se l'utente lo desidera puo' editare o direttamente inserire il path al file scelto direttamente nel campo senza aprire la finestra per la scelta della cartella.

Il file raster ricevuto in input viene quindi separato nelle sue componenti. Il file raster infatti è composto da due componenti:

- una matrice che contiene i dati di interesse di quel raster
- una enupla legata ai dati della matrice

Questo meccanismo permette di avere dei dati che permettono di definire meglio i dati rappresentati nella matrice. Questi dati sono :

- Numero di righe e colonne della matrice. Serve per avere la dimensione della matrice.
- Coordinate spaziali. Danno la posizione georeferenziata per la costruzione della matrice.
- Dimensioni della cella della matrice. Misura rappresentante la lunghezza e la larghezza dell'area di terreno i cui dati sono rappresentati dalla cella. Assieme alle coordinate spaziali serve a identificare l'area rappresentata dal file raster.
- Tipo di dato. Contiene delle informazioni rispetto al tipo di dato contenuto nelle celle. Questo è uniformo per tutto il file raster.

Questi dati non sono sempre presenti all'interno del file raster in quanto alcuni formati hanno per definizione uno di questi campi fissati a un valore, per cui sta al

driver di qgis leggere i file in modo corretto e ritornare un enupla contenente i dati necessari.

Anche se presenti sempre non siamo interessati ai campi che specificano il software utilizzato per la creazione dei file, il copyright con cui vengono distribuiti e la data di creazione del file per cui non li leggiamo.

Una volta letti i dati di un raster vengono immagazzinati in una enupla, di più facile gestione, che viene elaborata dal plugin.

Output

Come per l'input, l'output può essere gestito in due modi: attraverso il terminale python o via interfaccia grafica.

In entrambi i casi è possibile indicare il path e il nome del file di output e il formato in cui lo si vuole salvare. Il formato è scelto dall'utente da un elenco di formati disponibili nativamente da QGis con l'aggiunta dei formati disponibili da i plugin installati[10] con driver per la scrittura. Per come è costruito QGis è possibile che alcuni formati non siano disponibili per la scrittura in quanto i driver sono di solo lettura. In questo caso, indipendentemente dalla scelta di utilizzare il terminale o una gui il plugin lancerà una finestra con l'errore corrispondente all'errore lanciato dal driver. Questo viene fatto in quanto è possibile che i formati in input non possono essere correttamente codificati nel formato di output scelto.

In ogni caso si consiglia che il formato di input e output dei file sia lo stesso per avere il minor problemi dati dalla decodifica e codifica degli stessi.

Nel casos i driver non vengono registrati all'interno dell'elenco dei driver per l'output di QGis seguendo le api sarà necessario implementarle sequendole per poterle utilizzare all'interno del plugin.

2.1.3 Elaborazione dati

Una volta ricevuti i dati in input si pensa a elaborarli. Per farlo è stato necessario generare una struttura dati idonea ai formati in input. È stato quindi deciso di implementare l'elaborazione attraverso la struttura dati raster. A sua volta i dati

[10] Solo se questi registrano il formato nell'elenco nativamente presente in QGis come indicato dalle API per sviluppatori.

geomorfici del raster sono all'interno di una matrice, che è l'oggetto con cui il plugin esegue le operazioni.

In oltre è stato necessario trovare una o più librerie che implementassero le funzionalità richieste per l'elaborazione dei dati raster mantenendo al minimo le dipendenze esterne. In caso di assenza di librerie si sarebbe dovuto implementare tutto da zero. Per nostra fortuna sono disponibili alcune librerie che eseguono calcoli in raster.

Grass

Plugin standard che da accesso a tutte le funzionalità avanzate di GRASS GIS. Dal punto di vista tecnico non è una libreria ma un insieme di librerie che gestiscono raster e elaborazioni avanzate dei dati. Essendo di facile installazione e, in alcune versioni, distribuito direttamente con QGis è un ottimo sistema per avere facilmente una libreria per l'elaborazione dei raster.

Il plugin descritto in questa tesi richiede l'istallazione di Grass per il suo corretto funzionamento anche se non usa direttamente la libreria di Grass.

La libreria di Grass non è stata usata direttamente ma solo una delle sue componenti per avere maggiore libertà nei formati disponibili di file. Infatti, a meno di un ridotto numero di plugin, non è possibile aumentare i tipi di file supportati da Grass in quanto si limitano solo ai formati forniti con l'installazione standard di QGis.

Osgeo

Una delle librerie utilizzate da Grass. Formalmente è una libreria per il supporto di operazioni su raster, in pratica è un porting di Numpy per i formati di file contenenti dati georeferenziati.

Appendice A

Estendere le funzionalità del plugin

Questa appendice spiega come è strutturato a livello di codice il plugin. In particolare si vuole spiegare dove bisogna mettere mano al codice per modificarlo ed aggiungere funzionalità nuove.

A.1 Struttura del plugin

Il plugin, per poter funzionare, deve seguire la struttura standard indicata sulle api ufficiali [Com]:

- **File di configurazione**: necessari per il corretto funzionamento del plugin, contengono le informazioni per far funzionare correttamente il plugin, i codici delle versioni di QGis supportate e gli indirizzi delle risorse grafiche.

- **Risorse grafiche**: file che contengono il codice che, compilato, produce il codice che disegna le interfacce grafiche del plugin. Questi file sono strati costruiti attraverso la libreria grafica QT e vengono poi compilati in python.

- **Codice**: il codice che viene effettivamente eseguito dal plugin. Diviso in più file per facilità di manutenzione, deve contenere un file _init_.py da cui si avvia interamente il plugin.

A.1.1 Elementi di base

RUSLECalculator segue alla lettere la stuttura indicata da [Com]. Questo comporta la presenza di alcuni file che segnalano a QGis la presenza del plugin e permettano a QGis di caricarlo correttamente. In particolare è necessaria la presenza di un file _init_.py in cui viene lanciato il plugin e un file metadata.txt in cui ci sono i dati richiesti sul plugin. Se questi due file sono assenti o configurati in modo sbagliato o ci sono errori all'interno QGis si rifiuterà di caricarli e, di conseguenza, apparirà nell'elenco dei plugin difettosi (sistema fatto affinché lo sviluppatore possa avere riscontro con QGis su quale sia l'errore incontrato). In oltre, la corretta e completa compilazione del file metadata.txt permette l'invio del plugin ai server ufficiali di QGis per l'inserimento del plugin all'interno del catalogo ufficiale, che viene visualizzato all'interno del menu Plugin quando si vede quali sono i plugin attivi, disattivati, installati e non installati in QGis.

A.1.2 metadata.txt

File contenente i dati sul plugin. Tutte le informazioni che QGis carica attraverso il catalogo plugin provengono da questo file. Questi dati vengono divisi in due macrocategorie:

- **Mandatory items**: Elementi senza i quali QGis non riconosce il plugin. Senza di questi elementi non è nemmeno possibile caricare il plugin nel catalogo online

  name Nome del plugin che appare nel catalogo e che viene visualizzato nel menu dei plugin

  qgisMinimumVersion Versione minima richiesta di QGis per far funzionare il plugin. Utile per vedere quanto è retrocompatibile il plugin

  description Breve descrizione del plugin in modo che sia facilmente comprensibile la sua funzione

  version Numero di versione del plugin. Essenziale in quanto permette a QGis di capire se è l'ultima versione o c'è una versione successiva che va installata

  author Nome dell'autore che ha realizzato il plugin

  email Email di riferimento per il plugin

- **Recommended** Elementi non obbligatori che aggiungono informazioni extra sul materiale. L'elenco che segue sono quelli utilizzati per il plugin

  tags Elenco di parole chiave separate da virgola che descrivono il plugin. Servono per poter apparire nei risultati del cerca del catalogo dei plugin

  homepage Pagina del plugin, in cui trovare ulteriori informazioni su di esso

  tracker Pagina web dove si segnalano tutti i problemi legati al funzionamento del plugin. Solitamente è un servizio che viene offerto dai siti che hostano il repository del plugin

  repository Link al repository dove è hostato il sorgente del plugin. Il link è alla pagina del progetto corrispondente

  category Stringa che identifica la tipologia del plugin sviluppato. Utile per essere correttamente assegnato alle categorie del catalogo dei plugin

  icon Definisce il path relativo dalla cartella del progetto alla immagine utilizzata come icona dal plugin

  experimental Indica se la versione del plugin è stabile o una versione sperimentale

- **deprecated** Voce per segnalare che l'intero plugin è deprecato. Fare attenzione perchè se segnato come True viene applicato a tutte le versioni del plugin

A.1.3 GdalTools_utils.py

Script dal pacchetto GdalTools. Permette di aprire una finestra di salvataggio del file. Utilizzandola siamo obbligati a utilizzare la licenza GNU GPU v2 o superiore. È stato scelto di utilizzare questo script all'interno del progetto in modo da evitare dipendenze dal pacchetto GdalTools che risulta essere incompatibile con alcune installazioni di python e alcuni plugin. Per questo motivo, seguendo le indicazioni della licenza e condizioni d'uso di GdalTools, è stato copiato in modo integrale il file *GdalTools_utils.py* contenente la funzione necessaria per la visualizzazione della finestre e dell'elenco dei formati disponibili.

Bibliografia

[CB08] Stefano Olivieri Claudio Bosco. "Climate Change, Impacts and Adaptation Strategies in the Alpine Space". In: *Strategic INTERREG III B Project CLIMCHALP* (29 gen. 2008).

[Com] QGis Community. *QGis Api for Developer*. URL: http://qgis.org/api/.

[Lab] Clark Labs. *Manuali utente di IDRISI GIS Analysis*.

[Lut11] Mark Lutz. *Programming Python*. A cura di O'Reilly. O'Reilly, 2011.

[Pan09] Lorenzo Pantieri. *L'arte di gestire la bibliografia con LaTeX*. 2009. URL: http://www.lorenzopantieri.net/LaTeX_files/Bibliografia.pdf.

[PG11] Lorenzo Pantieri e Tommaso Gordini. *L'arte di scrivere con LaTeX*. 2011. URL: http://www.lorenzopantieri.net/LaTeX_files/ArteLaTeX.pdf.

[Sita] *Creative Commons - Attribution-ShareAlike 3.0 Unported - CC BY-SA 3.0*. URL: http://creativecommons.org/licenses/by-sa/3.0/.

[Sitb] *Pagina del progetto di QGis*. URL: http://www.qgis.org/it/.

[Zen13] Michele Zen. "Metodi e strumenti per la costruzione di territori virtuali per l'applicazione di modelli e di scenari ambientali". Tesi di laurea mag. Università Ca Foscari Venezia, 2013/2014.