

Alma Mater Studiorum
Università di Bologna

SCUOLA DI INGEGNERIA E ARCHITETTURA

Corso di laurea specialistica in:
Ingegneria Civile – Curriculum “Idraulica e territorio”

Tesi di laurea in:
Water Resources Engineering Management

**Metodi per la stima spaziale della temperatura in territori ad
orografia complessa e parametrizzazione di un modello
idrologico semi-distribuito su alcuni bacini appenninici**

Relatore:
Prof.ssa Elena Toth

Presentata da:
Mattia Neri

III sessione
Anno accademico 2014/2015

Indice

Introduzione	Pag. 3
CAPITOLO 1 – Caso di studio e dati a disposizione	
1.1 Inquadramento dell'area di studio	Pag. 5
1.2 Dati	Pag. 7
CAPITOLO 2 – Delimitazione dei bacini idrografici	Pag. 11
CAPITOLO 3 – Stima spaziale della temperatura	
3.1 Premessa	Pag. 19
3.2 Metodi di interpolazione per la stima della temperatura	
3.2.1 Scelta dei metodi	Pag. 24
3.2.2 Calcolo dei gradienti termici	Pag. 25
3.2.3 Descrizione delle tecniche di interpolazione	Pag. 26
3.3 Confronto fra i metodi: ricampionamento jack-knife	Pag. 30
3.4 Risultati del ricampionamento e scelta del metodo	Pag. 34
3.5 Creazione delle mappe di temperatura	Pag. 41
CAPITOLO 4 – Calibrazione del modello afflussi – deflussi	
4.1 Premessa	Pag. 45
4.2 Il modello HBV	Pag. 45
4.3 Preparazione dei dati in ingresso	
4.3.1 Schematizzazione per utilizzo semi-distribuito e a scala oraria	Pag. 52
4.3.2 Aggregazione temporale e spaziale dei dati di input	Pag. 60
4.4 Calibrazione del modello	
4.4.1 L'algoritmo di ottimizzazione	Pag. 62
4.4.2 Valutazione del periodo di calibrazione	Pag. 64
4.4.3 Calibrazione e risultati	Pag. 65
4.5 Validazione del modello	Pag. 81
CAPITOLO 5 – Regionalizzazione del modello afflussi – deflussi	Pag. 93
Conclusioni	Pag.101
Bibliografia e Sitografia	Pag.103
Appendice 1	Pag.105
Appendice 2	Pag.127

Introduzione

I bacini idrografici appenninici romagnoli rappresentano una fonte idropotabile ed energetica di essenziale importanza nella regione Emilia-Romagna. Infatti “La principale fonte di approvvigionamento dell’Acquedotto della Romagna è rappresentata dall’invaso artificiale di Ridracoli, posto in provincia di Forlì-Cesena, da cui si ottiene il soddisfacimento di circa il 50% del fabbisogno idrico totale. La captazione della risorsa idrica avviene, oltre che dal bacino diretto intercettato dalla diga di Ridracoli, anche da altri corsi intercettati da una galleria di gronda e da essa allacciati all’invaso” (Toth & Brath, 2015). Per questo la consapevolezza delle caratteristiche dei corsi d’acqua di interesse, soprattutto per quanto riguarda il regime idrologico, è un importante oggetto di ricerca e studi. In particolare, la modellazione idrologica dei bacini è stata ed è una delle principali protagoniste.

In questo elaborato di tesi ci si propone l’applicazione di un modello afflussi-deflussi caratterizzato da un’esplicita rappresentazione di fenomeni nevosi, ritenuti particolarmente importanti nell’area di studio e spesso trascurati nelle analisi riferite ai bacini appenninici. Il modello è stato implementato in relazione ad alcuni dei bacini idrografici in questione, utilizzando due diverse schematizzazioni per rappresentare il bacino: la prima prevede un approccio di tipo concentrato (Lumped), con il quale si considera un’unica area responsabile della generazione del deflusso, mentre la seconda uno di tipo semi-distribuito, dove la superficie del bacino viene suddivisa in fasce altimetriche.

Presumendo che un approccio semi-distribuito porti vantaggi rispetto ad uno di tipo totalmente concentrato, ci si pone come obiettivo la valutazione dell’effettivo miglioramento dei risultati in relazione all’incremento di complessità del modello. Come illustrato nei capitoli che seguono, la complessità del modello è elevata e include un elevato numero di parametri; si porrà particolare attenzione ai risultati della calibrazione nelle due modalità.

Per una corretta simulazione sia dei fenomeni di evapotraspirazione sia di quelli nevosi, è molto importante la corretta interpolazione dei dati meteorologici in ingresso e in particolare la stima della temperatura: in una zona ad orografia complessa come quella in questione non risulta affatto immediata la scelta della metodologia di interpolazione ottimale; per questo motivo la modellazione idrologica è stata preceduta dal confronto tra diverse metodologie proposte in letteratura per la stima della temperatura in funzione dell'elevazione del terreno, ricercando l'approccio che permettesse le migliori prestazioni in relazione all'area di studio.

CAPITOLO 1

Caso di studio e dati a disposizione

1.1 Inquadramento dell'area di studio

L'invaso di Ridracoli è l'opera idraulica di maggiore importanza, come già detto, degli appennini romagnoli. Essa è situata nella parte alta della valle del Bidente da cui è alimentata direttamente. Inoltre in essa confluisce acqua indirettamente da una galleria di gronda posta in sinistra idrografica che capta i deflussi superficiale dai bacini del Rio Bacine, Bidente di Celle, bidente di Campigna e Fiumicello, un affluente del torrente Rabbi. In Fig.1.1 è rappresentato lo schema di captazione dell'invaso.

L'area di studio scelta per l'applicazione del modello idrologico comprende appunto i cinque bacini sopra elencati i cui deflussi vengono convogliati in ingresso all'invaso, tramite opere di captazione dei deflussi. Ad essi viene aggiunto un ulteriore bacino, quello del torrente Rabbi a monte della confluenza col Rio Fiumicello, sul quale si potrebbe ipotizzare il prolungamento della galleria di gronda su tre diverse sezioni. In questo caso, ai fini della calibrazione, la sezione di chiusura è stata posta in corrispondenza della stazione idrometrica presente sul corso d'acqua dagli anni '80. Come si vedrà a fine elaborato, una volta applicato e calibrato il modello, è stata effettuata una regionalizzazione dei parametri sulle altre due sezioni non strumentate.

I bacini si trovano al confine fra Emilia-Romagna e Toscana. Sono compresi quasi totalmente nella provincia di Forlì-Cesena e in parte nelle province di Firenze e Arezzo.

La regione di interesse si estende lungo la catena appenninica per una ventina di chilometri. In Fig.1.2 se ne riporta la collocazione sulla penisola.



Figura 1.1 – Schematizzazione della galleria di gronda che convoglia le portate al bacino di Ridracoli

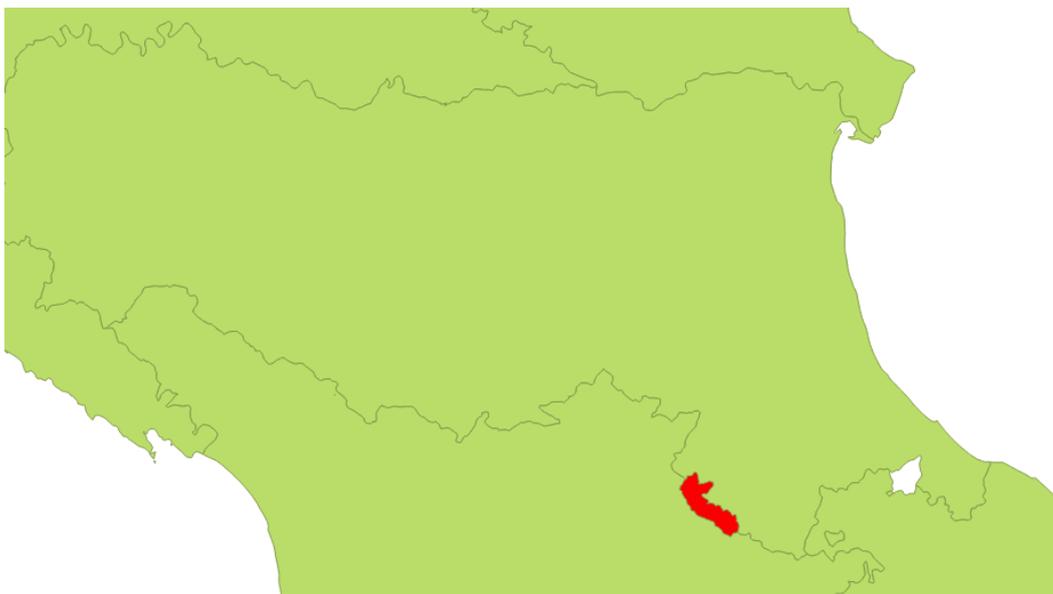


Figura 1.2 – Collocazione dell'area di studio

1.2 Dati

I dati necessari per le analisi da effettuare sono misure di temperatura e precipitazione sulle superfici dei bacini e di portata sulle sezioni di chiusura. Queste ultime, essendo misure di per sé puntuali, coincidono con quelle fornite dalle stazioni idrometriche, mentre temperatura e precipitazione devono essere interpolate sulle superfici dei bacini a partire dalle misure puntuali dei sensori delle stazioni termo/pluviometriche.

I dati a disposizione provengono da diverse reti di misura. Come periodo di osservazione si è scelto quello compreso fra il 1 Gennaio 1996 e il 31 Dicembre 2014, all'interno del quale, per tutte le tipologie di osservazioni richieste, si hanno dati sufficienti per poter effettuare una modellazione afflussi-deflussi.

Per quanto riguarda le misure di portata si sono individuate sei sezioni di chiusura: la diga di Ridracoli, che chiude il bacino diretto, le quattro sezioni in corrispondenza delle opere di presa dei bacini allacciati alla gronda e la stazione di misura sul Rabbi. Romagna Acque – Società delle Fonti Spa ha messo a disposizione dell'università i dati di portata giornalieri e orari. Di seguito si riporta una tabella contenente caratteristiche e posizione in coordinate UTM fuso 32 N delle sezioni di chiusura e in Fig.3 se ne riporta la collocazione sulla carta.

È da precisare come i valori di portata per la sezione *Diga* non sono frutto di misurazione diretta, ma sono stati ricavati dal bilancio idrico per l'invaso di Ridracoli e sono di conseguenza affetti da maggiore incertezza.

Stazione idrometrica	Est	Nord	Località
Diga (bacino diretto)	728106	4862008	---
Campigna	723383	4863700	Corniolo – Opera di presa Campigna
Bacine	726579	4862152	Ridracoli – Opera di presa Rio Bacine
Celle	722583	4864632	Corniolo – Opera di presa Celle
Fiumicello	720520	4869207	Fiumicello – Opera di presa
Rabbi	719972	4870180	Rabbi

Per quanto riguarda le misure di temperatura e precipitazione, si sono reperiti dati da dodici sensori, nove dei quali forniscono entrambi i dati, mentre i restanti solo misure pluviometriche. Per le simulazioni si è scelto di utilizzare quando possibile valori di temperatura media oraria e precipitazione cumulata oraria; infatti, potendo effettuare in questo modo simulazioni a risoluzione più spinta, è possibile cogliere fenomeni idrologici che, essendo la dimensione dei bacini molto ridotta, potrebbero evolvere velocemente e che sarebbe perciò impossibile cogliere a scala giornaliera. Le stazioni metereologiche sono gestite da tre differenti enti:

- Romagna Acque – Società delle Fonti Spa (indicato in tabella come R.A.)
- Arpa Emilia-Romagna, con dati a disposizione sulla piattaforma “Dexter” (interfaccia con la quale si è in grado visualizzare le stazioni di misura sulle mappe e scaricare i dati richiesti)
- SIR, Servizio Idrologico della Regione Toscana

Si riportano in tabella le caratteristiche delle stazioni (posizione in coordinate UTM 32N, gestore e tipologia di sensore) e in Fig.1.3 se ne riporta la collocazione nella regione di interesse.

Nome	Tipologia sensore	Est	Nord	Gestore	Località
BadiaPrataglia	Stazione pluviometrica	731446	4853038	SIR	Poppi
Braccina	Stazione termo/pluviometrica	722816.4	4867134	R.A.	Braccina – Fiumicello presso Braccina
Camaldoli	Stazione termo/pluviometrica	727035	4853040	SIR	Poppi
Campigna	Stazione termo/pluviometrica	720731.9	4861301	R.A.	Campigna
Campominacci	Stazione termo/pluviometrica	725649.8	4859971	R.A.	Campominacci - Camposega
Capaccio	Stazione termo/pluviometrica	732146.7	4867637	R.A.	Capaccio – impianto di potabilizzazione
Castellaccio	Stazione termo/pluviometrica	728726.1	4866038	R.A.	Biserno - Castellaccio
Corniolo	Stazione termo/pluviometrica	724238.8	4867254	ARPA	Corniolo
Diga	Stazione termo/pluviometrica	728086.7	4861361	R.A.	Ridracoli - Diga
Lama	Stazione termo/pluviometrica	728156.7	4857101	R.A.	Lama – Casa forestale
Lastra	Stazione pluviometrica	732976.6	4865256	ARPA	Bagno di Romagna
Premilcuore	Stazione pluviometrica	722976	4873541	ARPA	Premilcuore

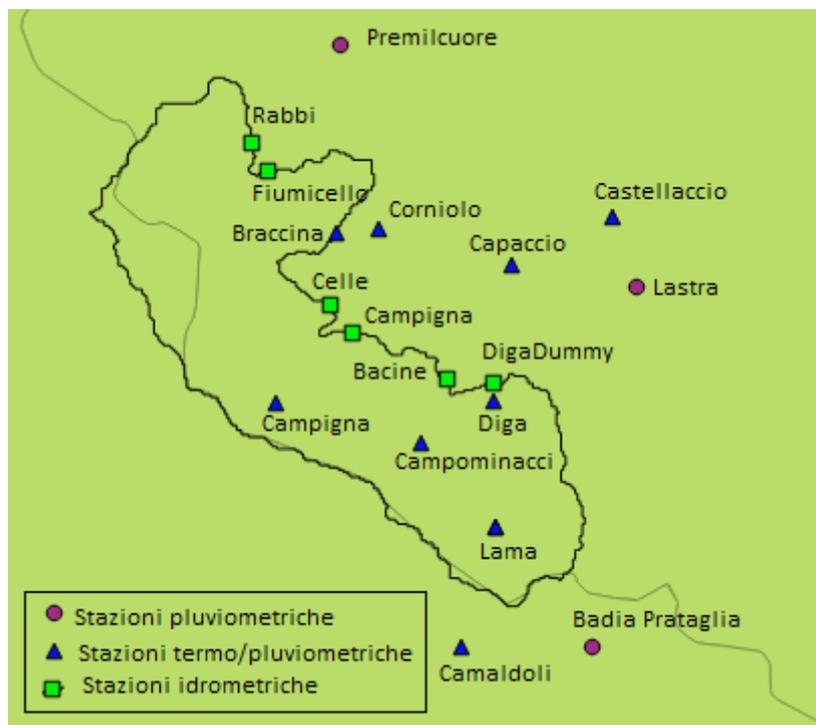


Figura 1.3 – Collocazione e tipologia dei sensori utilizzati

CAPITOLO 2

Delimitazione dei bacini idrografici

Per delimitare i confini dei bacini idrografici dei corsi d'acqua, è necessario individuare le sezioni di chiusura che si andranno a considerare per la calibrazione del modello idrologico. Queste corrisponderanno ovviamente con le stazioni idrometriche che forniscono misure di portata, individuate nel capitolo precedente, di cui se ne riporta una tabella riassuntiva:

Stazione	Est	Nord
Diga	728106	4862008
Campigna	723383	4863700
Bacine	726579	4862152
Celle	722583	4864632
Fiumicello	720520	4869207
Rabbi	719972	4870180

L'individuazione dei bacini idrografici passa attraverso alla conoscenza delle direzioni di drenaggio su tutta la superficie, in modo da poter determinare quale porzione della stessa faccia recapitare alla sezione di chiusura la precipitazione tramite scorrimento superficiale. Tale area può essere ricavata, trascurando eventuali stratificazioni sotterranee, conoscendo la distribuzione delle quote del terreno dell'area di interesse. A seconda del grado di discretizzazione della superficie, la determinazione dei confini dei bacini sarà più o meno accurata. Queste operazioni possono essere eseguite con l'utilizzo dei più comuni software GIS (Geographic Information System).

Una volta individuate le coordinate delle sezioni di chiusura, per la delimitazione dei confini di ogni relativo bacino è necessario perciò

possedere un modello di elevazione del terreno *DEM* sufficientemente accurato. Un *DEM* consiste in una mappa *raster*, quindi in formato matriciale, i cui elementi (pixel o celle elementari) assumono un valore che indica la quota del terreno. In questo caso si è utilizzato un modello di elevazione del centro-nord Italia di risoluzione 80mx80m, sufficiente per l'obbiettivo dell'analisi.

In Fig.2.1 e 2.2 si riporta la rappresentazione grafica del modello di elevazione sul quale è visualizzata anche la posizione delle sezioni di chiusura.

Tramite i software GIS QuantumGis e Grass è stato possibile ricavare facilmente il perimetro dei bacini. In poche parole, considerando il valore di quota di ogni cella, è possibile individuare in maniera automatica la direzione di drenaggio considerando la direzione di pendenza massima fra le 8 celle adiacenti, in Fig.2.3 se ne riporta un esempio; viene costruita una mappa sull'area interessata dove ad ogni cella corrisponde la direzione di drenaggio dell'acqua che ivi giunge. Di conseguenza il software è in grado di individuare il percorso dei corsi d'acqua e le celle di confine al di fuori delle quali il contributo liquido non raggiunge la sezione di chiusura (Fig.2.4 e in Fig.2.5).

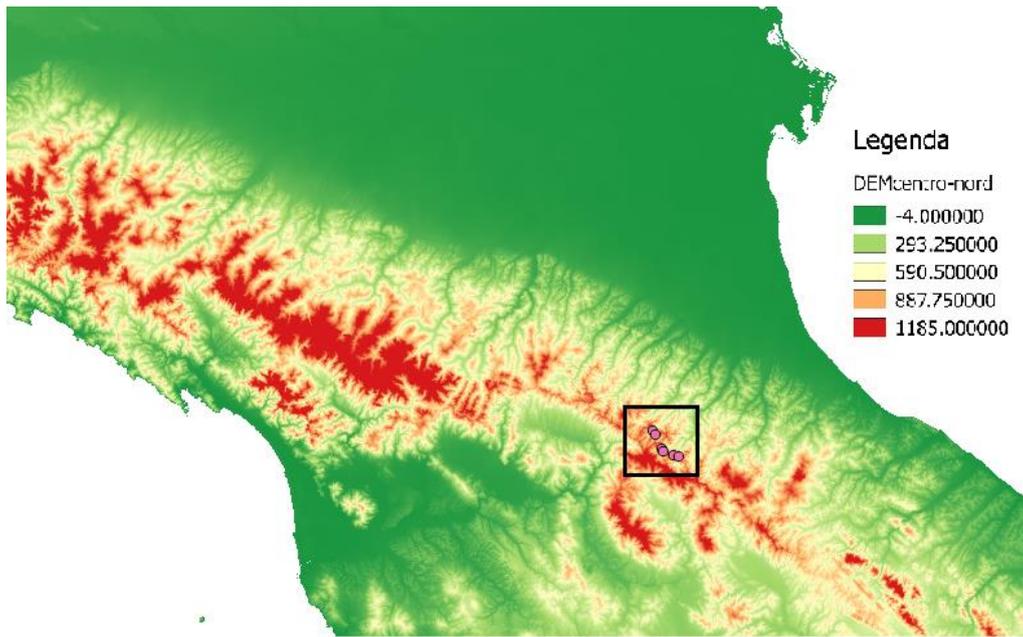


Figura 2.1 – DEM Italia Centro-Nord

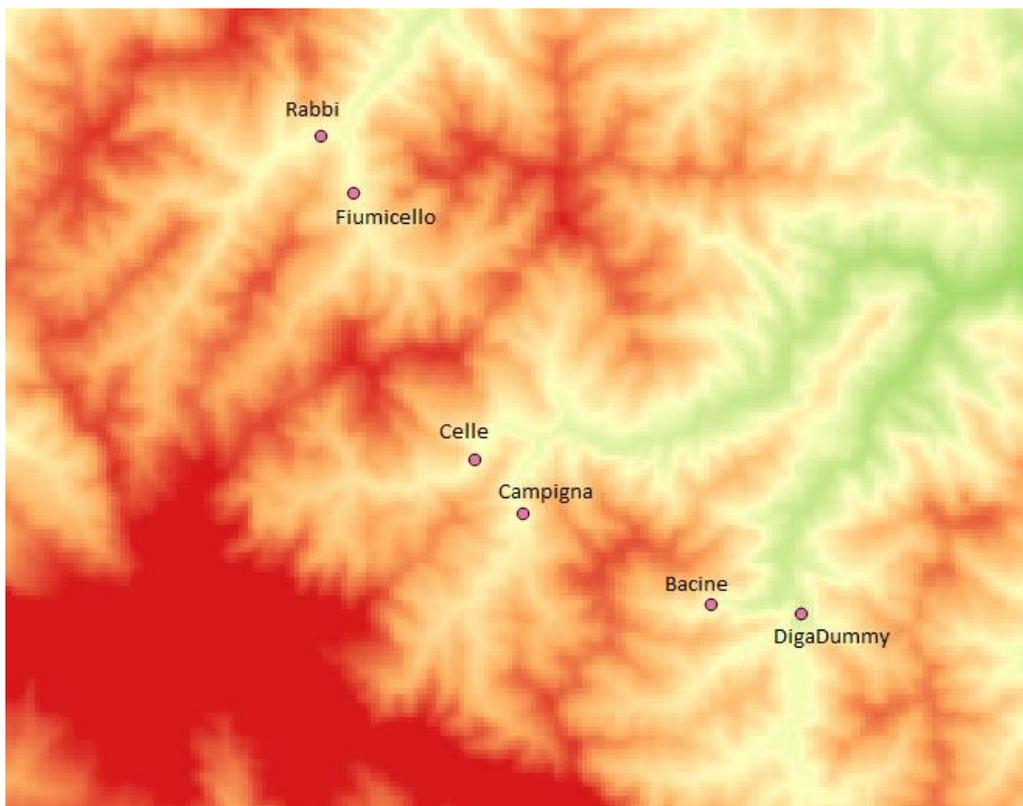


Figura 2.2 - Zoom DEM con coordinate sezioni di chiusura

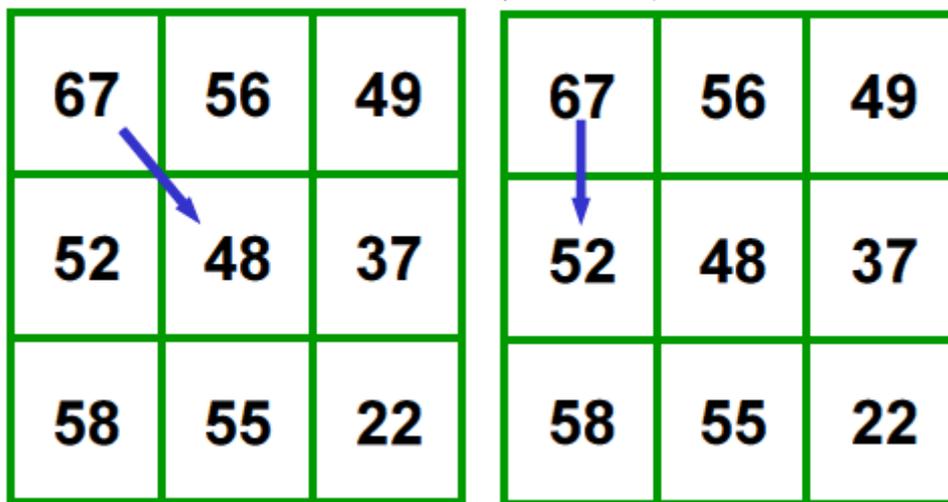


Figura 2.3 – Esempio: determinazione della direzione di drenaggio

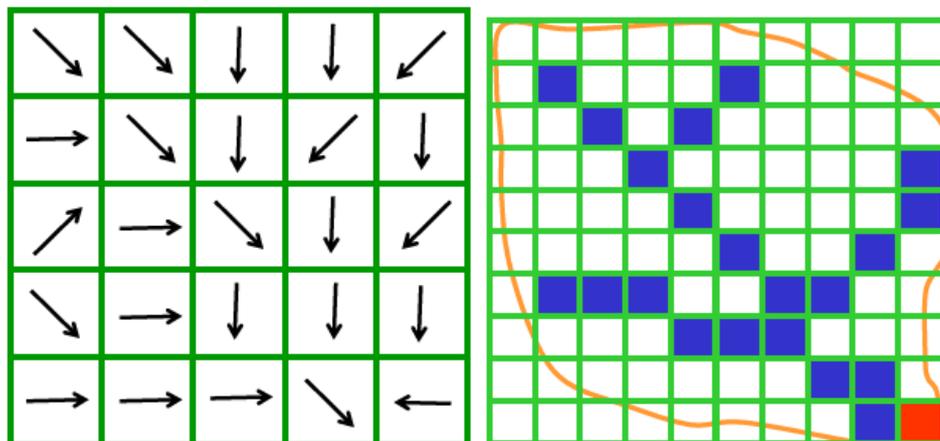


Figura 2.4 – Esempio: direzioni di drenaggio Figura 2.5 – Esempio: determinazione dei corsi d'acqua e dei confini del bacino

Si sono eseguiti i seguenti passi:

1. Tramite la funzione *r.watershed* di Grass, dando in input *DEM* e dimensione minima dei bacini, in questo caso posta pari a 1000 m², il programma genera quattro file *raster*:
 - Mappa di accumulazione: a ogni cella corrisponde il numero delle celle che drena la cella stessa
 - Mappa di drenaggio: ogni cella definisce la direzione di drenaggio (Fig.2.6)
 - Mappa dei corsi d'acqua, definiti come l'insieme delle celle che drenano un'area superiore alla soglia assegnata (Fig.2.7)



Figura 2.6 – Esempio: mappa di drenaggio

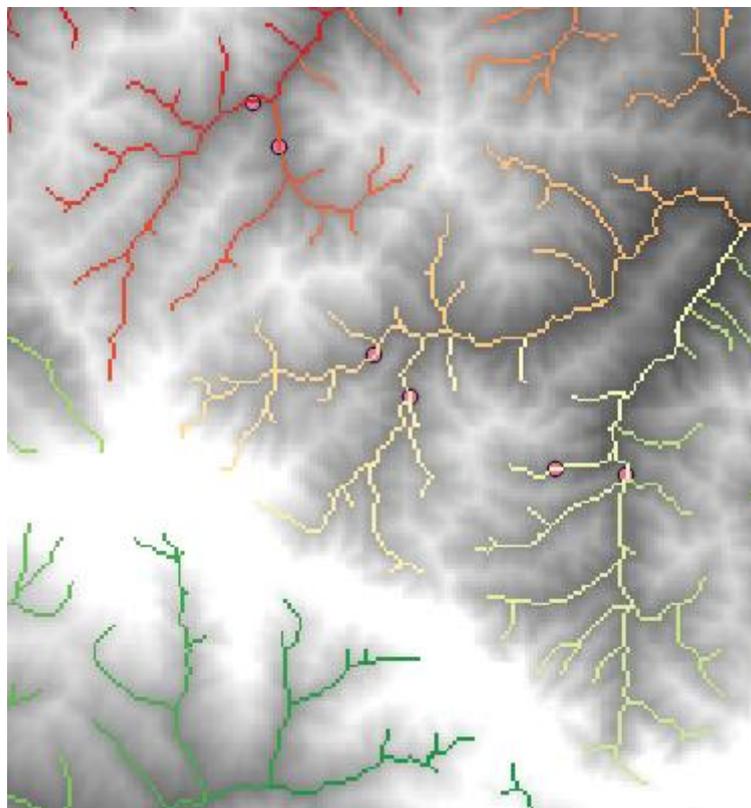


Figura 2.7 – Esempio: mappa dei corsi d'acqua

2. Tramite la funzione *r.water.outlet*, dando in input le coordinate della sezione di chiusura desiderata (che deve ricadere coerentemente su un corso d'acqua) e la mappa di drenaggio, si ottiene la definizione del bacino idrografico chiuso in corrispondenza della sezione stessa. Il file di output è di tipo *raster*, dove le celle comprese nel bacino possiedono valore unitario, mentre quelle esterne nullo; questo viene poi convertito in un file vettoriale (formato *shape*) che rappresenta il perimetro del bacino stesso. L'operazione viene effettuata per tutte le sezioni di chiusura. Si riporta il risultato in Fig.2.8.

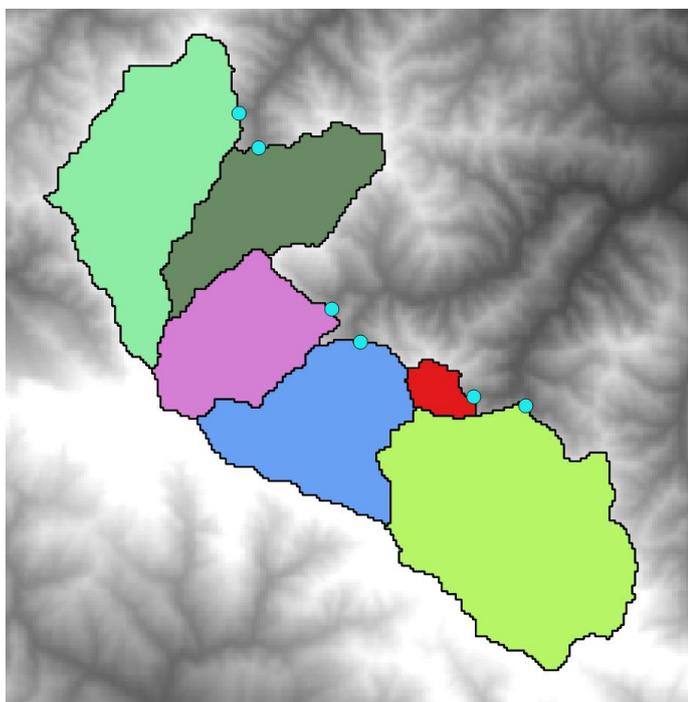


Figura 2.8 – Shape dei bacini idrografici ottenuti

3. Unendo tutti gli shape dei bacini ricavati, si ottiene la maschera vettoriale che comprende tutta l'area di studio (Fig.2.9). Grazie a quest'ultima sarà possibile estrarre dal *DEM* totale solo la porzione interessata, che verrà utilizzata per la creazione delle mappe di pioggia e di temperatura, come illustrato nei capitoli successivi.

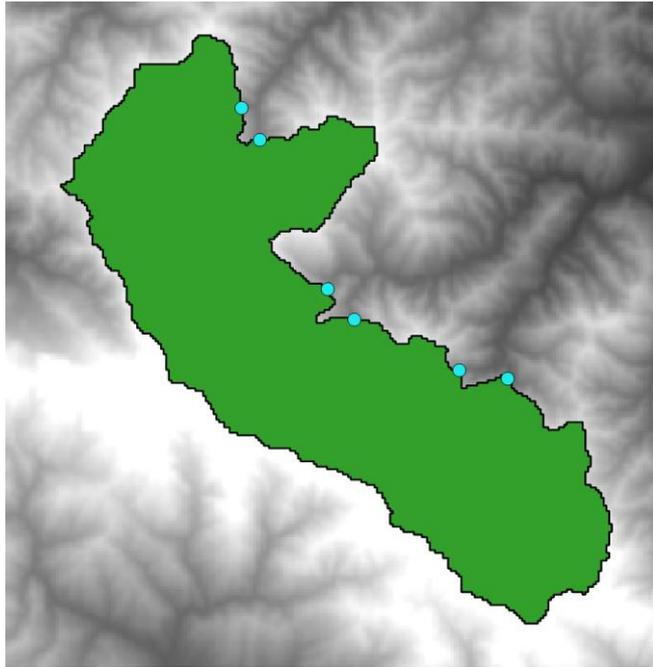


Figura 2.9 – Shape dei bacini uniti e sezioni di chiusura

CAPITOLO 3

Stima spaziale della temperatura

3.1 Premessa

La temperatura dell'aria è una delle variabili più importanti in ingresso ad un modello afflussi-deflussi che preveda la simulazione dell'evapotraspirato e/o un modulo per la gestione dell'accumulo e dello scioglimento nivale. Per questo, nasce la necessità di dover stimare la temperatura sull'intera superficie del bacino a partire da dati puntuali, ricavati da stazioni di misura posizionate irregolarmente sull'area d'interesse.

In regioni montane, con una topografia complessa, come l'area di studio in esame, l'operazione di stima si complica fortemente; infatti essa non dipenderà solo dalla posizione rispetto alle stazioni di misura di cui si posseggono i dati, ma sarà fortemente influenzata anche dall'elevazione del terreno.

In questa sezione dell'elaborato ci si pone come obiettivo l'individuazione della strategia di interpolazione che sia in grado di fornire la più accurata stima possibile sui bacini oggetto di studio.

In letteratura si possono trovare una gran quantità di tecniche, la cui affidabilità varia fortemente in relazione all'area a cui vengono applicate. Complessivamente alcuni fra quelli che senza dubbio trovano maggior impiego, sono i metodi basati sulla media pesata dei valori puntuali di temperatura noti; i pesi, nella maggioranza dei casi, contengono in parte o esclusivamente l'inversa distanza (singola o quadratica) dalle stazioni termometriche. Il concetto di inversa distanza nasce per fare in modo che i termometri posti nelle vicinanze del punto che si vuole interpolare abbiano giustamente un peso maggiore sulla media.

Come già accennato, in territori montuosi, o in generale ad orografia complessa, questo non è sufficiente per ottenere una stima accurata; è necessario perciò introdurre il concetto di gradiente termico, che indica la variazione di temperatura al variare della quota. Infatti i vari metodi si differenziano principalmente dal tipo di gradiente termico utilizzato e da come esso viene inserito all'interno della stima. Tipicamente esso viene utilizzato per riferire la temperatura dei sensori alla quota del punto da interpolare prima di effettuarne la media; questo si basa sull'idea di minimizzare l'influenza della differente elevazione. Le formulazioni saranno del tipo:

$$T_{x,i} = \frac{\sum_{j=1}^n [T_{j,i} - G(h_x - h_j)] w_{x,j}}{\sum_{j=1}^n w_{x,j}} \quad (3.1)$$

Dove:

$T_{j,i}$ è la temperatura misurata dalla stazione j al passo i [°C]

h_x è la quota del punto x dove si vuole stimare la temperatura [m]

h_j è la quota della stazione j [m]

$w_{x,j}$ è il peso della stazione j rispetto al punto x

G è il gradiente termico [°C/m]

Per rendere l'idea del significato del gradiente termico si riporta in Fig. 3.1, a titolo totalmente esemplificativo, l'andamento delle temperature medie annuali misurate dalle stazioni termometriche utilizzate per l'analisi in relazione alla quota delle stesse.

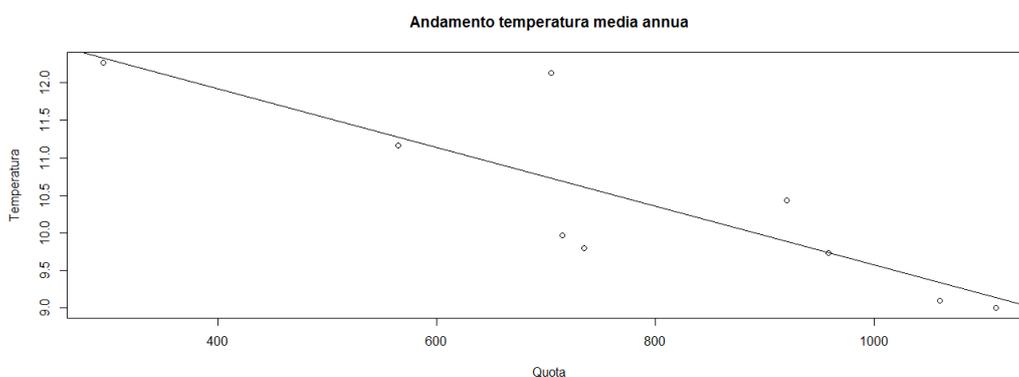


Figura 3.1 - Temperature medie annuali misurate dai sensori in relazione alla quota delle stesse e all'andamento medio del gradiente termico

Per quanto riguarda i valori assegnati al gradiente termico, si trovano in letteratura diversi approcci: in alcuni casi viene imposto un valore a priori che rappresenta l'andamento medio generale osservato in situazioni di questo tipo, pari a $-0.6/0.65$ °C ogni 100m (*Running et al., 1987; Dodson e Marks, 1997*); in altri casi si può utilizzare un valore fisso sempre costante per tutto l'anno, calcolato però appositamente per l'area di interesse, o in alternativa per tenere conto della stagionalità viene talvolta calcolato un gradiente variabile mensilmente (*Liston e Elder, 2006*).

Un'altra classe di metodi è rappresentata invece da quelli che stimano la temperatura utilizzando una regressione lineare sulla posizione. In base alla posizione delle stazioni e ai valori di temperatura assunti viene ricavata una relazione fra temperatura e posizione valida per tutta l'area di studio. All'interno di questa categoria può essere impiegata una regressione lineare singola, solo rispetto alla quota (*Garen e Marks, 2005*), o una regressione lineare multipla rispetto a quota e posizione planimetrica (*Bolstad et al., 1998; Xia et al., 2001*).

A seguito di una ricerca bibliografica, si sono individuati e paragonati sei diversi metodi, appartenenti a entrambe le categorie, esposti nei prossimi paragrafi. La bontà dei vari metodi è stata infine determinata tramite un ricampionamento di tipo jack-knife.

I dati utilizzati sono quelli relativi alle stazioni di misura della temperatura riportate nel capitolo 1. Di seguito se ne riporta di nuovo la collocazione spaziale (Fig.3.2), una tabella contenente, oltre alla posizione, anche la quota delle stesse e un'ulteriore tabella contenente le statistiche sui dati dei sensori.

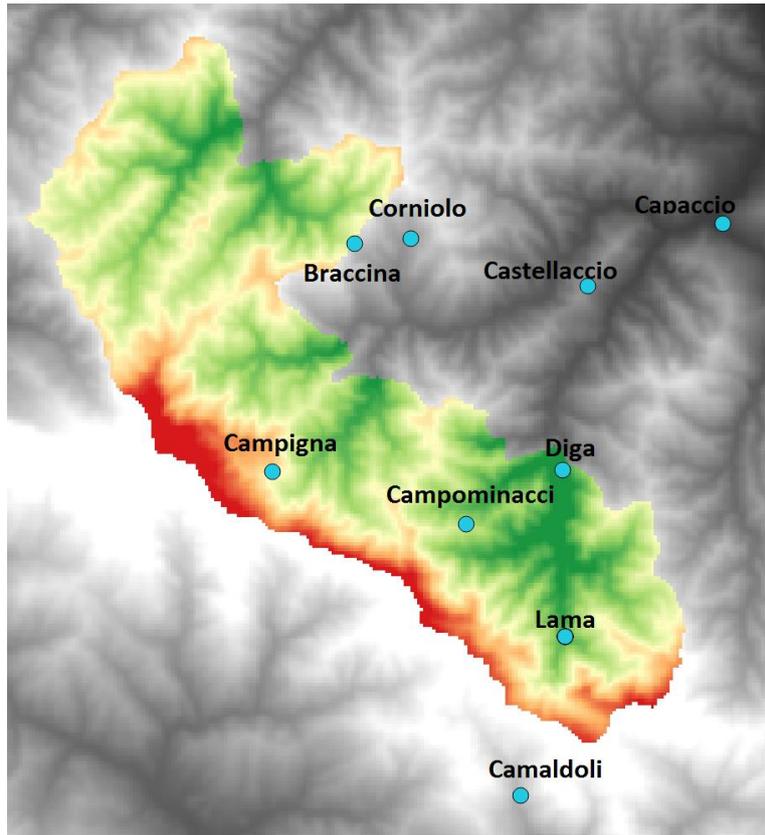


Figura 3.2 – Collocazione sull'area delle stazioni termometriche

Stazione	Est	Nord	Quota (msl)
Braccina	722816.4	4867134	958
Camaldoli	727035	4853040	1111
Campigna	720731.9	4861301	1060
Campominacci	725649.8	4859971	920
Capaccio	732146.7	4867637	295
Castellaccio	728726.1	4866038	705
Corniolo	724238.8	4867254	735
Diga	728086.7	4861361	565
Lama	728156.7	4857101	715

Stazione	Media [°C]	Dev.st. [°C]	Min [°C]	Max [°C]	% mancanti
Braccina	9.74	7.90	-12.91	34.00	1.3
Camaldoli	9.00	7.65	-12.60	32.20	23.3
Campigna	9.10	7.70	-14.00	30.50	2.5
Campominacci	10.44	7.99	-13.70	34.60	2.1
Capaccio	12.27	8.57	-13.30	38.20	2.6
Castellaccio	12.14	8.32	-28.22	38.10	1.5
Corniolo	9.80	7.87	-12.70	33.30	9.4
Diga	11.17	8.04	-13.22	35.60	1.1
Lama	9.97	7.95	-16.11	34.30	3.7

Come già accennato, sono state utilizzate le temperature medie orarie comprese nel periodo sul quale verrà applicato il modello afflussi-deflussi, fra il 1 gennaio 1996 e il 31 dicembre 2014. I dati forniti avevano già subito un primo controllo di qualità, nel quale erano stati eliminati gli “outliers”, ovvero i valori anomali, chiaramente distanti dalle altre osservazioni. Essendo il numero delle stazioni sufficientemente elevato, non si sono riscontrati eccessivi problemi dovuti alla mancanza di dati. Infatti con la metodologia identificata come più efficiente (vedi sez. 3.2 – 3.4), dopo specifica analisi, si è ricavato che solo nello 0,6% del periodo di analisi si hanno contemporaneamente meno di 4 stazioni attive (limite per avere un’interpolazione affidabile), con una lunghezza media degli intervalli mancanti pari a 1,8 ore e un massimo di 25 ore consecutive, verificatosi una sola volta sui diciotto anni di dati.

3.2 Metodi di interpolazione per la stima della temperatura

3.2.1 Scelta dei metodi

Come accennato in precedenza si sono analizzati diversi metodi per la stima della temperatura media oraria sulla superficie dei bacini. Per la scelta degli stessi, si è preso spunto da quelli esposti nel paragrafo precedente, citati e raccolti nell'articolo "Comparison of approaches for spatial interpolation of daily air temperature in a large region with complex topography and highly variable station density" (*Stahl et al., 2006*), dove viene esposto uno studio effettuato su un'area montuosa canadese di dimensioni di gran lunga maggiori dell'area di studio qui analizzata.

In particolare, fra i metodi basati sulla media pesata delle temperature misurate nelle varie stazioni di misura ad ogni intervallo temporale, si sono analizzati:

1) *InvD2 Grad06*

Media pesata rispetto alla distanza quadratica dalle stazioni, con unico gradiente termico fisso di $-0,6^{\circ}\text{C}$ ogni 100 metri di dislivello (*Running et al., 1987; Dodson e Marks, 1997*)

2) *InvD2 GradAnnual*

Media pesata rispetto alla distanza quadratica dalle stazioni, con unico gradiente termico annuale ricavato tramite regressione su tutto l'intervallo di dati disponibile per l'area di studio (vedi paragrafo 3.2.2)

3) *InvD2 GradMonthly*

Media pesata rispetto alla distanza quadratica dalle stazioni, con gradiente termico variabile mensilmente, ricavato tramite regressione su tutto l'intervallo di dati per ciascun mese (vedi paragrafo 3.2.2)

4) *InvD2+El GradMonthly*

Media pesata rispetto alla distanza quadratica dalle stazioni e rispetto alla quota delle stesse, con gradiente termico variabile mensilmente come al metodo 3)

Mentre fra i metodi che utilizzano una regressione lineare si è utilizzato:

5) *Linear Regression*

Regressione lineare singola rispetto alla quota ad ogni intervallo temporale

6) *Multiple Linear Regression*

Regressione lineare multipla rispetto a quota e posizione planimetrica ad ogni intervallo temporale

3.2.2 Calcolo dei gradienti termici

Per il calcolo del gradiente annuale e di quelli mensili per l'area di interesse si è proceduto ricavando la relazione che lega le quote delle stazioni termometriche e le temperature da loro registrate. Nella pratica questo viene fatto plottando tutte le temperature di tutte le stazioni in funzione della quota delle stesse, verificatesi nel periodo interessato; calcolando la retta di regressione, il gradiente termico sarà individuato dalla pendenza della stessa, che infatti rappresenta il coefficiente di diminuzione della temperatura all'aumentare della quota.

Il gradiente annuale viene ricavato in funzione delle misure simultaneamente registrate nelle stazioni della specifica area di studio, utilizzando le misure di tutte le stazioni e di tutti i mesi; mentre per i gradienti mensili, si è eseguita la stessa operazione suddividendo però le temperature relative ai vari mesi dell'anno, ricavando quindi dodici differenti valori.

Dalle analisi è risultato un gradiente annuale medio di -0.39 °C/100m, ottenuto con un coefficiente di correlazione R^2 pari a 0.014, mentre per quanto riguarda i gradienti mensili si riporta di seguito una tabella contenente i risultati.

Mese	G _m (°C/100m)	Mese	G _m (°C/100m)
Gennaio	- 0,25	Luglio	- 0,42
Febbraio	- 0,38	Agosto	- 0,38
Marzo	- 0,46	Settembre	- 0,39
Aprile	- 0,46	Ottobre	- 0,37
Maggio	- 0,46	Novembre	- 0,36
Giugno	- 0,43	Dicembre	- 0,28

3.2.3 Descrizione delle tecniche di interpolazione

1. InvD2 Grad06

La temperatura viene stimata tramite la media pesata delle temperature misurate dalle varie stazioni, riferite alla quota del punto in esame tramite un gradiente termico fissato a priori di -0,6°C ogni 100 metri di dislivello. La temperatura al passo temporale i del punto x è quindi definita come:

$$T_{x,i} = \frac{\sum_{j=1}^n [T_{j,i} - 0,006 * (h_x - h_j)] w_{x,j}}{\sum_{j=1}^n w_{x,j}} \quad (3.2)$$

Dove:

$T_{j,i}$ è la temperatura misurata dalla stazione j al passo i [°C]

h_x è la quota del punto x dove si vuole stimare la temperatura [m]

h_j è la quota della stazione j [m]

$w_{x,j}$ è il peso della stazione j rispetto al punto x definito come: [m⁻²]

$$w_{x,j} = \frac{1}{d_{x,j}^2} \quad (3.3)$$

Con $d_{x,j}$ distanza fra il punto x e la stazione j [m]

2. InvD2 GradAnnual

La temperatura viene stimata in maniera analoga al metodo 1, con l'unica differenza che il gradiente non viene posto fisso a priori, pari a -0,6°C ogni 100 metri, ma è stato invece stimato come illustrato nel paragrafo

precedente: come già detto, si è ricavato un gradiente medio pari a -0,39°C ogni 100 metri di dislivello.

3. InvD2 GradMonthly

Anche questo metodo, come i due precedenti, si basa sulla media pesata rispetto alle inverse distanze quadratiche, tuttavia il gradiente non è fisso, ma varia mensilmente. I gradienti utilizzati relativi ai vari mesi sono quelli riportati nel paragrafo precedente.

La temperatura al passo temporale i del punto x risulterà:

$$T_{x,i} = \frac{\sum_{j=1}^n [T_{j,i} + G_m * (h_x - h_j)] w_{x,j}}{\sum_{j=1}^n w_{x,j}} \quad (3.4)$$

Dove G_m è appunto il gradiente relativo al mese m a cui appartiene l'ora i .

4. InvD2+EI GradMonthly

In questo caso si tratta sempre di media pesata delle temperature misurate con l'utilizzo di un gradiente mensile come nel metodo precedente, ma con l'introduzione della differenza di quota fra punto e stazione oltre alla distanza planimetrica all'interno del calcolo delle distanze. Infatti l'espressione della temperatura rimane la seguente:

$$T_{x,i} = \frac{\sum_{j=1}^n [T_{j,i} + G_m * (h_x - h_j)] w_{x,j}}{\sum_{j=1}^n w_{x,j}} \quad (3.5)$$

Dove però i pesi non contengono solo l'inversa distanza quadratica, ma sono definiti come:

$$w_{x,j} = \frac{1}{d_{x,j}^2 + |F_e \Delta E_{x,j}|} \quad (3.6)$$

Dove:

F_e è un coefficiente di adattamento posto pari a 100 [m] come raccomandato dal report del National Weather River Forecasting System “Calibration on Conceptual Hydrologic Models for Use in River Forecasting” (Anderson, 2002)

$\Delta E_{x,j}$ è la differenza di quota fra il punto x e la stazione j [m]

In questo modo influiscono più significativamente nella stima della temperatura non solo le stazioni più vicine planimetricamente, ma anche quelle aventi quota più simile.

5. Linear Regression (con la sola quota)

Non viene considerata la posizione planimetrica delle stazioni. Ad ogni passo temporale, quindi ad ogni ora, viene ricavata la relazione fra temperatura e quota delle stazioni valida per tutta l'area di studio, tramite una regressione lineare ai minimi quadrati; si ricava una coppia di coefficienti (intercetta e pendenza), tramite i quali, conoscendo esclusivamente la quota del punto, è possibile ricavare la temperatura. Ne segue quindi:

$$T_{x,i} = C_{1,i} + Z_x * C_{2,i} \quad (3.7)$$

Dove:

$C_{1,i}$ è l'intercetta ricavata tramite regressione al passo i [°C]

Z_x è la quota del punto x [m]

$C_{2,i}$ è la pendenza della retta di regressione ricavata al passo i [°C*m⁻¹]

6. Multiple Linear Regression (con quota e coordinate planimetriche)

Ad ogni passo temporale viene effettuata una regressione multipla ai minimi quadrati fra temperatura e posizione delle stazioni nello spazio tridimensionale; si ricava la relazione che lega la temperatura alla quota e alla posizione planimetrica, tramite la determinazione di 4 coefficienti.

La temperatura in posizione x al passo i sarà data dalla seguente relazione:

$$T_{x,i} = C_{1,i} + Z_x * C_{2,i} + X_x * C_{3,i} + Y_x * C_{4,i} \quad (3.8)$$

Dove:

i coefficienti $C_{1,i}$ $C_{2,i}$ $C_{3,i}$ $C_{4,i}$ sono i coefficienti ricavati dalla regressione multipla ai minimi quadrati.

Z_x X_x Y_x rappresentano la posizione del punto x nello spazio, rispettivamente quota e coordinate UTM [m].

La regressione necessita però un minimo di quattro stazioni attive in ogni ora di calcolo per essere in grado di ricavare i coefficienti; nei rarissimi casi in cui questa condizione non fosse soddisfatta dai nostri dati, prima di eseguire la regressione, si è operato stimando preventivamente la temperatura delle stazioni mancanti tramite un approccio alle inverse distanze come al metodo 2. Sebbene una soluzione di questo tipo non sia del tutto corretta, si è valutato che non sarebbe andata ad alterare la bontà complessiva dei dati, avendo dovuto applicarla solo allo 0,6% del periodo.

È da notare che attraverso gli ultimi due metodi si tiene implicitamente conto anche dell'inversione termica, fenomeno che consiste nell'inversione del gradiente termico; in alcune ore della giornata, soprattutto in certi periodi dell'anno, è possibile avere gradienti termici positivi, ovvero temperatura crescente all'aumentare della quota. Eseguendo quindi regressioni a passi orari, è possibile individuare e considerare tale fenomeno, variabile nel tempo, all'interno della stima.

Tutte le operazioni descritte in precedenza (calcolo dei gradienti, stima dei coefficienti delle regressioni lineari ecc.) e quelle esposte nei prossimi paragrafi sono state effettuate sviluppando ex-novo diversi algoritmi nel linguaggio di programmazione R: sono stati assemblati gli script per l'analisi e l'elaborazione dei dati, utilizzando come dati in ingresso al problema solo i valori di temperatura misurati dai sensori, come esposto

precedentemente. Gli script con gli algoritmi utilizzati sono riportati in Appendice 1.

3.3 Confronto fra i metodi: ricampionamento jack-knife

Una volta scelte le metodologie di interpolazione, si è valutata la bontà di ognuna di esse, confrontando i risultati ottenuti da un ricampionamento di tipo jack-knife. Come già accennato tutte le operazioni di seguito descritte sono state implementate in algoritmi costruendo, come in precedenza, script di calcolo nel linguaggio R.

Il ricampionamento jack-knife è una procedura utilizzata in statistica per stimare l'errore quadratico medio. L'idea è quella di ricalcolare più volte la grandezza statistica lasciando fuori dal campione una osservazione alla volta; successivamente si confrontano le stime calcolate con le osservazioni reali, ricavandone l'errore che ne deriva.

In pratica, nel caso in esame, si sono svolte le seguenti operazioni per ognuno dei metodi proposti:

- Per ogni stazione di misura, si sono eliminate dalla stima le osservazioni fornite dalla stessa.
- Tramite il metodo analizzato si sono stimate le temperature, ad ogni passo temporale, nel punto di coordinate e quota corrispondenti alla stazione eliminata.
- Si sono paragonati le temperature osservate e stimate con la procedura jack-knife ricavandone l'errore quadratico medio *RMSE* (si è calcolato l'errore complessivo sul periodo di riferimento per ogni stazione, l'errore medio delle stazioni sui vari mesi dell'anno e l'errore complessivo mediato sulle stazioni e su tutto il periodo di riferimento).

In questo modo è possibile valutare, per l'area di studio, quale sia l'interpolazione che è in grado di fornire i dati più affidabili.

Gli script in R utilizzati per il ricampionamento vengono riportati in Appendice1, mentre di seguito si riportano le tabelle con gli errori risultanti divisi per mese e per stazione, per tutte le sei metodologie.

1. InvD2 Grad06 – RMSE [°C]

Mese / Stazione	Braccina	Camaldoli	Campigna	Camponinacci	Capaccio	Castellaccio	Corniolo	Diga	Lama
Gennaio	1.199	2.096	1.526	2.344	3.855	2.259	1.408	2.194	2.318
Febbraio	1.057	1.727	1.243	2.188	3.309	2.307	1.285	2.101	1.991
Marzo	1.092	1.558	1.351	1.983	2.894	2.144	1.329	1.917	1.834
Aprile	1.148	1.449	0.979	1.997	2.683	2.127	1.405	1.823	1.778
Maggio	1.214	1.540	1.047	2.168	2.687	2.222	1.488	1.962	2.013
Giugno	1.225	1.655	1.064	2.397	2.795	2.339	1.503	2.074	2.311
Luglio	1.297	1.826	1.150	2.557	2.943	2.565	1.538	2.119	2.632
Agosto	1.306	1.973	1.170	2.602	3.121	2.636	1.479	2.092	2.923
Settembre	1.213	1.816	1.076	2.227	2.911	2.415	1.427	1.898	2.427
Ottobre	1.126	1.921	1.197	1.978	2.913	2.163	1.341	1.875	2.045
Novembre	1.046	2.000	1.176	1.825	3.057	1.887	1.265	1.749	1.959
Dicembre	1.097	2.130	1.148	2.192	3.854	2.185	1.208	2.005	2.354

2. InvD2 GradAnnual – RMSE [°C]

Mese / Stazione	Braccina	Camaldoli	Campigna	Camponinacci	Capaccio	Castellaccio	Corniolo	Diga	Lama
Gennaio	0.874	1.839	1.317	2.130	3.376	2.166	1.120	2.004	2.217
Febbraio	0.698	1.565	1.220	2.012	2.945	2.210	0.953	1.951	1.903
Marzo	0.763	1.539	1.484	1.821	2.646	2.043	1.002	1.793	1.748
Aprile	0.854	1.405	1.134	1.826	2.429	2.022	1.095	1.674	1.688
Maggio	0.910	1.493	1.150	1.955	2.403	2.110	1.190	1.750	1.909
Giugno	0.941	1.540	1.169	2.166	2.462	2.221	1.198	1.844	2.189
Luglio	0.991	1.708	1.294	2.305	2.572	2.449	1.245	1.882	2.497
Agosto	1.014	1.759	1.275	2.337	2.665	2.520	1.187	1.854	2.777
Settembre	0.887	1.607	1.119	1.988	2.481	2.302	1.128	1.685	2.288
Ottobre	0.793	1.727	1.112	1.785	2.510	2.057	1.023	1.701	1.931
Novembre	0.687	1.831	1.103	1.666	2.708	1.794	0.930	1.632	1.874
Dicembre	0.751	1.882	1.051	1.992	3.370	2.085	0.879	1.847	2.250

3. InvD2 GradMonthly – RMSE [°C]

Mese / Stazione	Braccina	Camaldoli	Campigna	Camponinacci	Capaccio	Castellaccio	Corniolo	Diga	Lama
Gennaio	0.734	1.811	1.292	2.025	3.173	2.107	0.959	1.952	2.165
Febbraio	0.683	1.565	1.225	2.006	2.934	2.205	0.938	1.947	1.900
Marzo	0.861	1.509	1.418	1.866	2.695	2.076	1.108	1.817	1.773
Aprile	0.938	1.380	1.054	1.874	2.477	2.057	1.195	1.706	1.714
Maggio	0.999	1.471	1.087	2.018	2.462	2.147	1.285	1.805	1.941
Giugno	0.986	1.537	1.129	2.206	2.502	2.243	1.253	1.878	2.210
Luglio	1.029	1.707	1.260	2.338	2.608	2.466	1.285	1.908	2.515
Agosto	1.003	1.756	1.286	2.326	2.649	2.514	1.174	1.846	2.771
Settembre	0.887	1.607	1.119	1.988	2.481	2.302	1.128	1.685	2.288
Ottobre	0.768	1.723	1.118	1.771	2.486	2.047	0.994	1.693	1.921
Novembre	0.645	1.829	1.114	1.651	2.682	1.782	0.885	1.629	1.864
Dicembre	0.600	1.857	1.136	1.901	3.163	2.022	0.684	1.824	2.195

4. InvD2+EI GradMonthly – RMSE [°C]

Mese / Stazione	Braccina	Camaldoli	Campigna	Camponinacci	Capaccio	Castellaccio	Corniolo	Diga	Lama
Gennaio	0.733	1.810	1.292	2.024	3.173	2.106	0.960	1.951	2.164
Febbraio	0.682	1.565	1.225	2.005	2.933	2.205	0.940	1.946	1.899
Marzo	0.860	1.508	1.418	1.865	2.695	2.076	1.109	1.816	1.773
Aprile	0.937	1.379	1.054	1.873	2.477	2.056	1.196	1.705	1.714
Maggio	0.998	1.470	1.087	2.017	2.462	2.147	1.286	1.804	1.940
Giugno	0.984	1.536	1.129	2.205	2.502	2.243	1.254	1.876	2.210
Luglio	1.027	1.706	1.260	2.337	2.607	2.466	1.286	1.907	2.515
Agosto	1.002	1.755	1.286	2.325	2.649	2.514	1.175	1.844	2.771
Settembre	0.886	1.606	1.119	1.988	2.480	2.302	1.129	1.684	2.288
Ottobre	0.767	1.722	1.117	1.769	2.486	2.046	0.996	1.692	1.921
Novembre	0.644	1.829	1.114	1.650	2.681	1.781	0.886	1.628	1.864
Dicembre	0.600	1.856	1.136	1.900	3.162	2.021	0.685	1.823	2.195

5. Linear Regression – RMSE [°C]

Mese / Stazione	Braccina	Camaldoli	Campigna	Camponinacci	Capaccio	Castellaccio	Corniolo	Diga	Lama
Gennaio	0.743	1.203	0.878	1.033	1.117	1.774	1.449	1.025	1.771
Febbraio	0.755	1.069	0.859	1.049	0.896	1.849	1.420	0.952	1.517
Marzo	0.760	1.067	1.066	1.029	0.823	1.761	1.463	0.905	1.424
Aprile	0.698	0.973	0.774	0.999	0.683	1.756	1.481	0.840	1.350
Maggio	0.778	1.030	0.758	1.121	0.773	1.842	1.533	0.959	1.563
Giugno	0.813	1.067	0.782	1.264	0.850	1.942	1.507	1.076	1.777
Luglio	0.861	1.209	0.836	1.335	0.849	2.173	1.576	1.100	2.051
Agosto	0.852	1.161	0.875	1.275	0.824	2.237	1.482	1.009	2.297
Settembre	0.749	1.018	0.778	1.066	0.751	2.021	1.436	0.850	1.888
Ottobre	0.698	1.051	0.698	0.894	0.828	1.765	1.349	0.854	1.570
Novembre	0.644	1.122	0.689	0.832	1.024	1.510	1.327	0.878	1.512
Dicembre	0.663	1.126	0.837	0.956	1.192	1.709	1.293	1.015	1.794

6. Multiple Linear Regression – RMSE [°C]

Mese / Stazione	Braccina	Camaldoli	Campigna	Camponinacci	Capaccio	Castellaccio	Corniolo	Diga	Lama
Gennaio	0.928	0.872	0.883	1.060	1.089	1.140	1.063	0.970	1.071
Febbraio	0.859	0.771	0.814	1.021	0.913	1.061	0.968	0.826	0.878
Marzo	0.763	0.762	0.890	0.966	0.833	0.942	1.066	0.767	0.773
Aprile	0.742	0.726	0.726	0.960	0.707	0.890	1.007	0.666	0.688
Maggio	0.813	0.809	0.809	1.074	0.754	0.941	1.012	0.730	0.845
Giugno	0.902	0.827	0.828	1.228	0.800	0.993	0.990	0.849	0.967
Luglio	0.984	0.985	0.841	1.294	0.831	1.036	1.047	0.936	1.100
Agosto	1.006	0.829	0.799	1.257	0.824	1.041	0.962	0.969	1.188
Settembre	0.874	0.738	0.771	1.053	0.775	1.022	0.956	0.807	0.986
Ottobre	0.792	0.719	0.758	0.893	0.814	0.998	0.925	0.772	0.877
Novembre	0.706	0.795	0.745	0.845	0.988	1.011	0.906	0.834	0.956
Dicembre	0.781	0.803	0.782	0.977	1.157	1.094	0.899	1.033	1.097

3.4 Risultati del ricampionamento e scelta del metodo

Il ricampionamento jack-knife ha consentito la costruzione di curve di errore sui vari e mesi e sulle varie stazioni, che permettono la valutazione dell'efficacia delle varie metodologie d'interpolazione proposte.

Nei grafici che seguono si riporta l'andamento dell'errore quadratico medio *RMSE* dei metodi, facendone la media su tutte le stazioni, in corrispondenza di ogni mese dell'anno (Fig.3.3), su tutti i mesi in corrispondenza di ogni singola stazione (Fig.3.4) e infine su ogni mese in corrispondenza di ogni singola stazione (Fig.3.5-3.16)

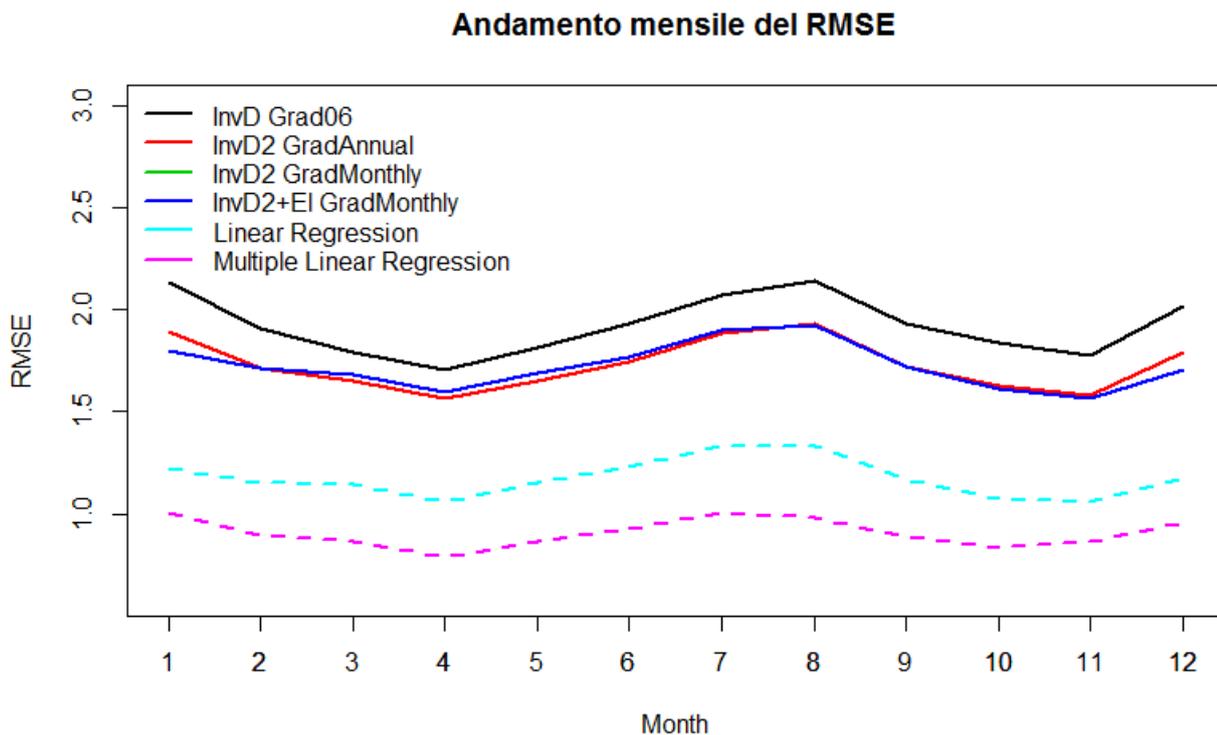


Figura 3.3 – Andamento dell'RMSE medio fra le stazioni lungo i mesi dell'anno

Andamento del RMSE per stazione

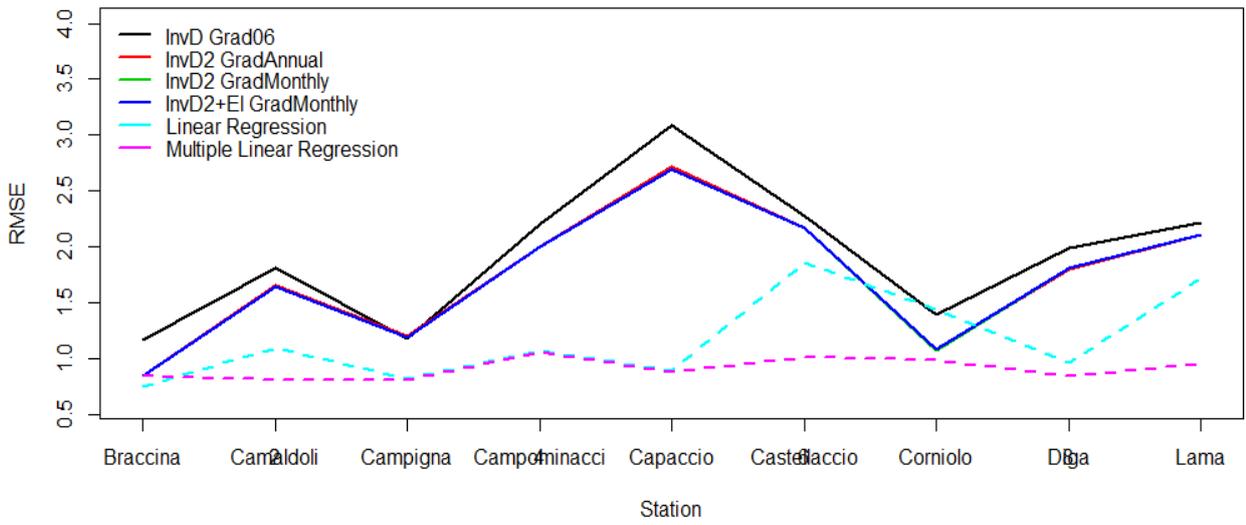


Figura 3.4 – Andamento dell'RMSE medio annuale sulle varie stazioni

Andamento del RMSE per stazione - Gennaio

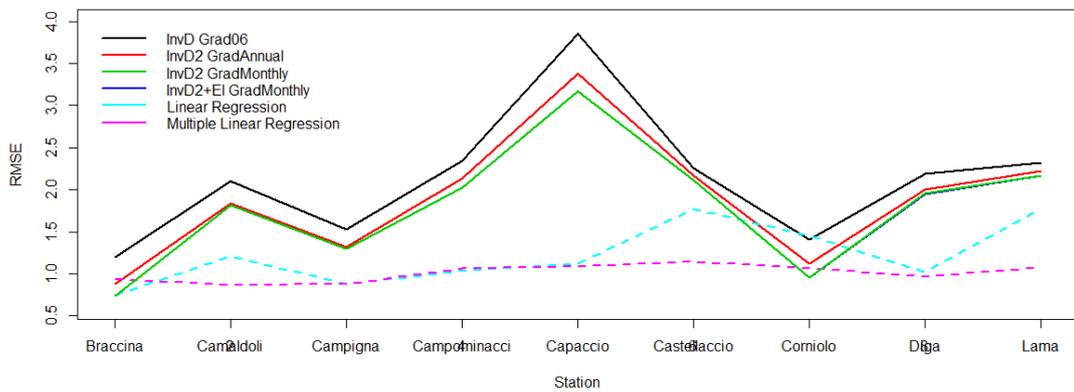


Figura 3.5 – Andamento dell'RMSE medio mensile sulle varie stazioni – gennaio

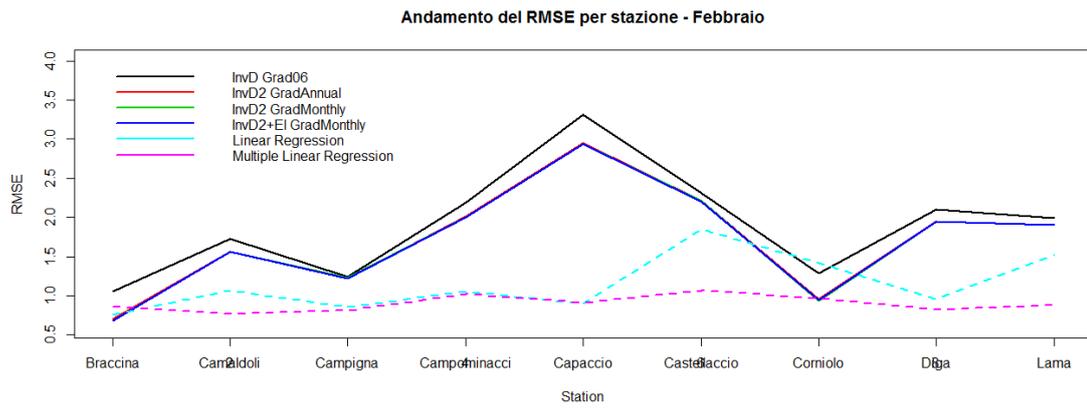


Figura 3.6 – Andamento dell'RMSE medio mensile sulle varie stazioni – febbraio

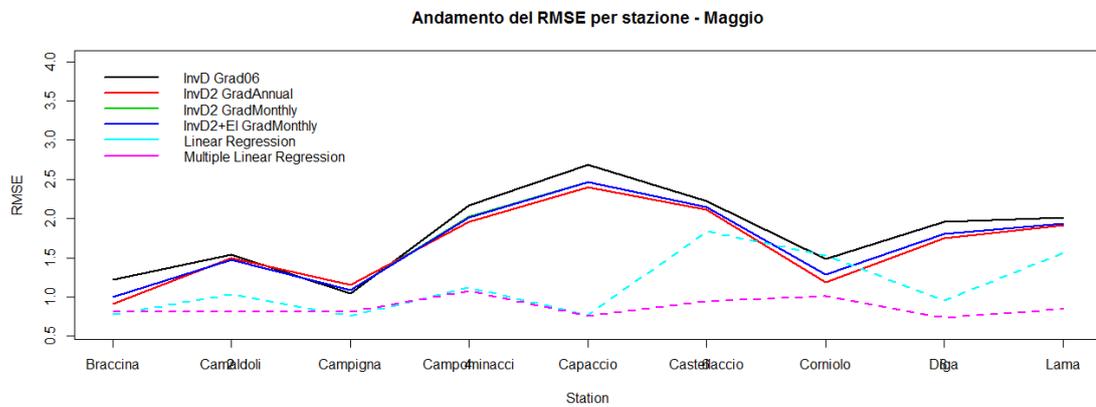


Figura 3.7 – Andamento dell'RMSE medio mensile sulle varie stazioni – marzo

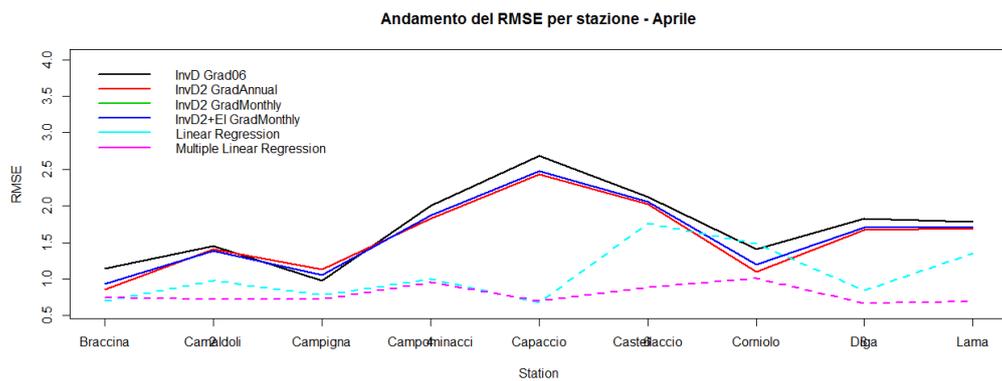


Figura 3.8 – Andamento dell'RMSE medio mensile sulle varie stazioni – aprile

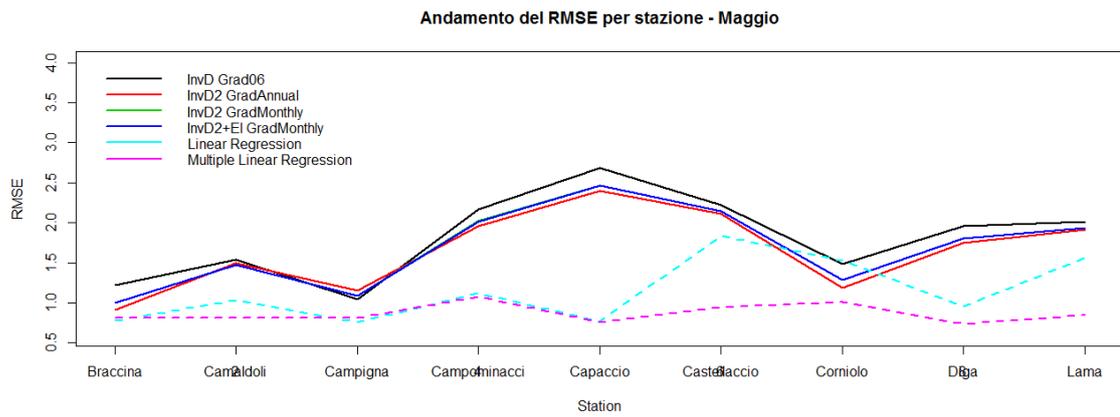


Figura 3.9 – Andamento dell'RMSE medio mensile sulle varie stazioni – maggio

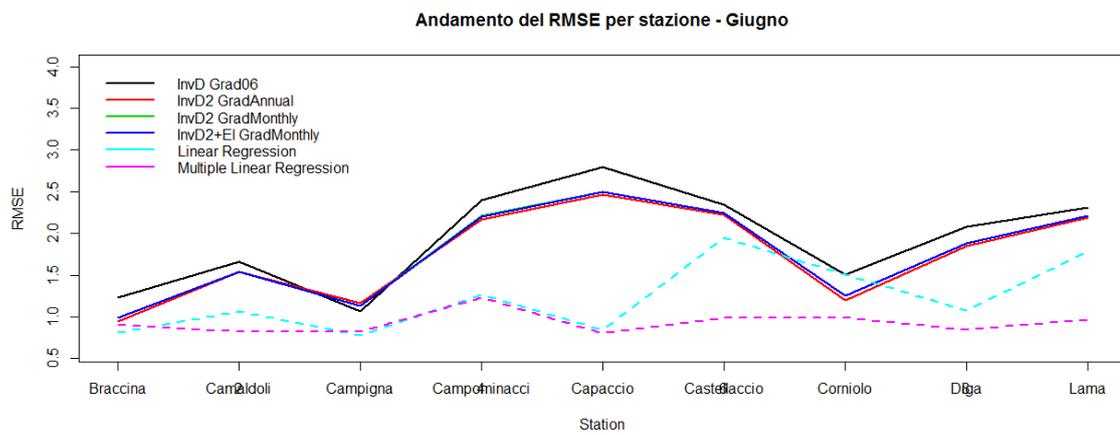


Figura 3.10 – Andamento dell'RMSE medio mensile sulle varie stazioni - giugno

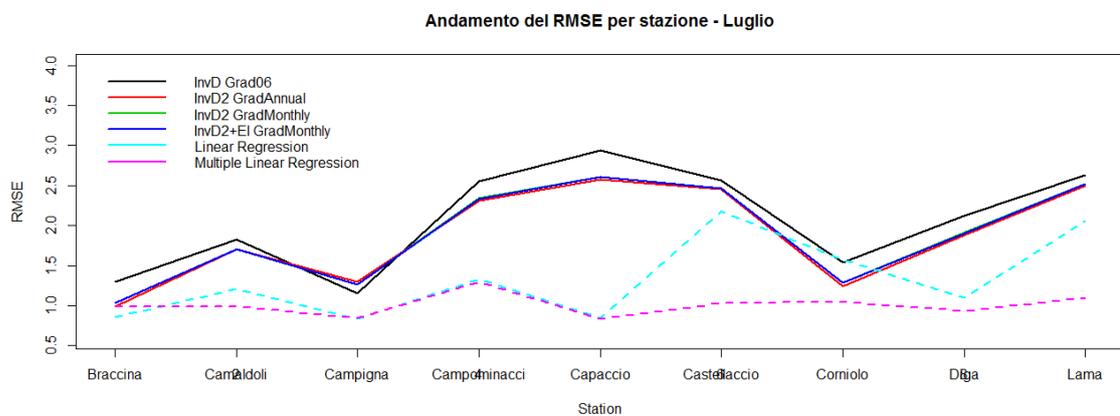


Figura 3.11 – Andamento dell'RMSE medio mensile sulle varie stazioni – luglio

Andamento del RMSE per stazione - Agosto

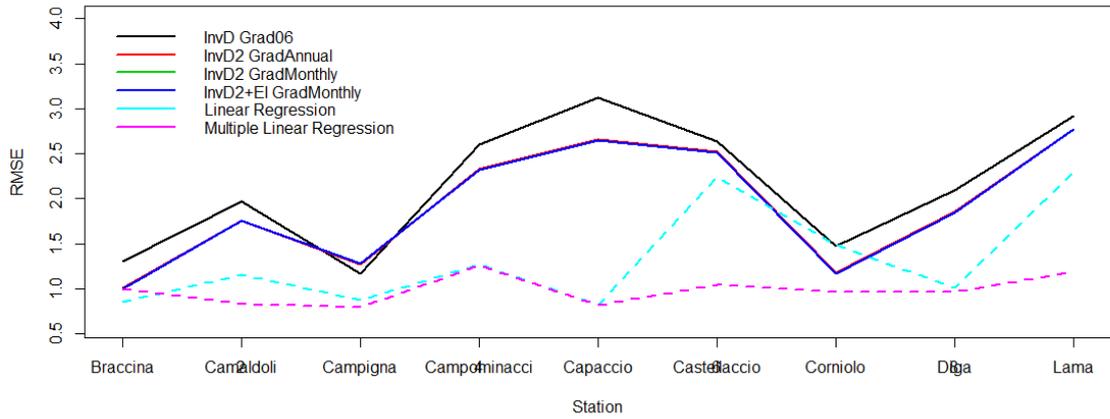


Figura 3.12 – Andamento dell'RMSE medio mensile sulle varie stazioni – agosto

Andamento del RMSE per stazione - Settembre

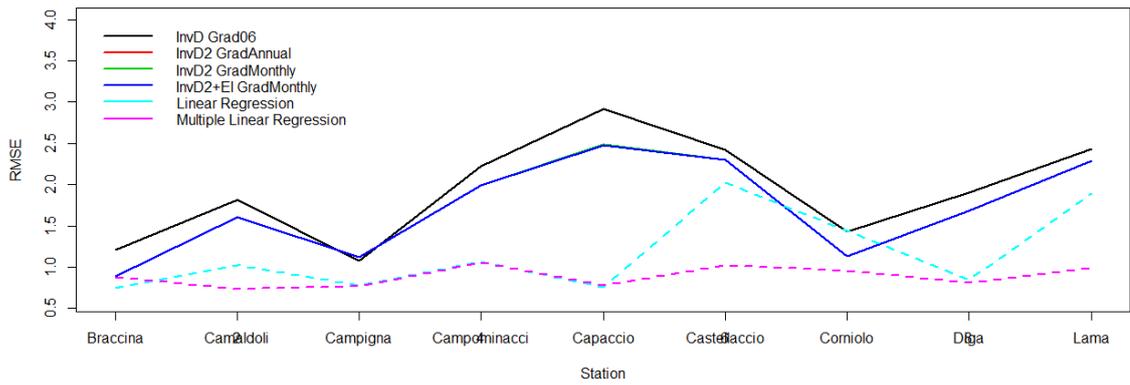


Figura 3.13 – Andamento dell'RMSE medio mensile sulle varie stazioni - settembre

Andamento del RMSE per stazione - Ottobre

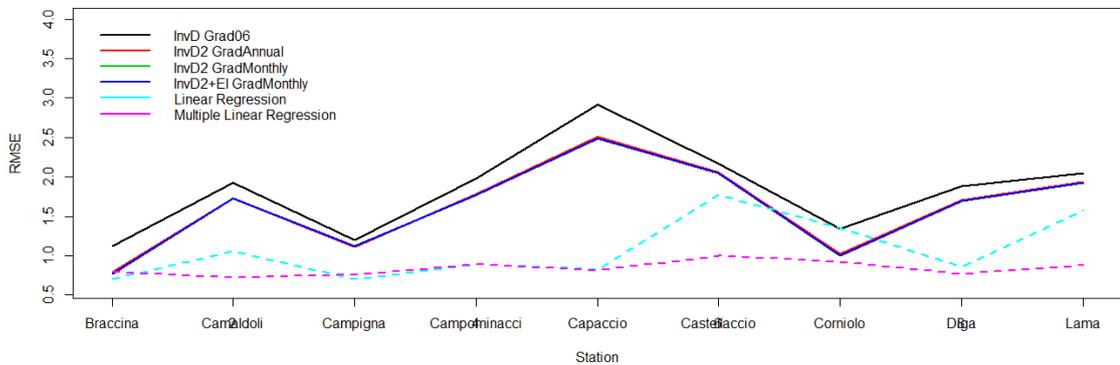


Figura 3.14 – Andamento dell'RMSE medio mensile sulle varie stazioni – ottobre

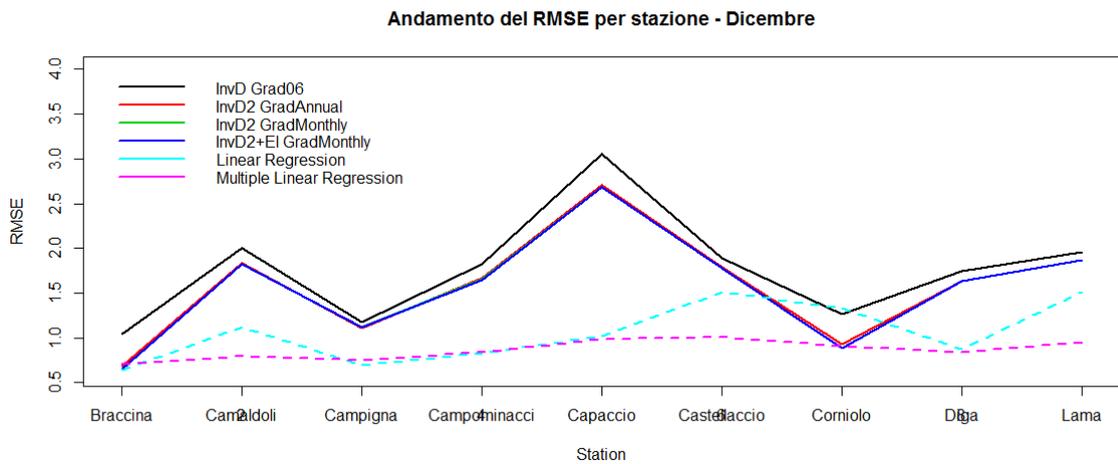


Figura 3.15 – Andamento dell'RMSE medio mensile sulle varie stazioni – novembre

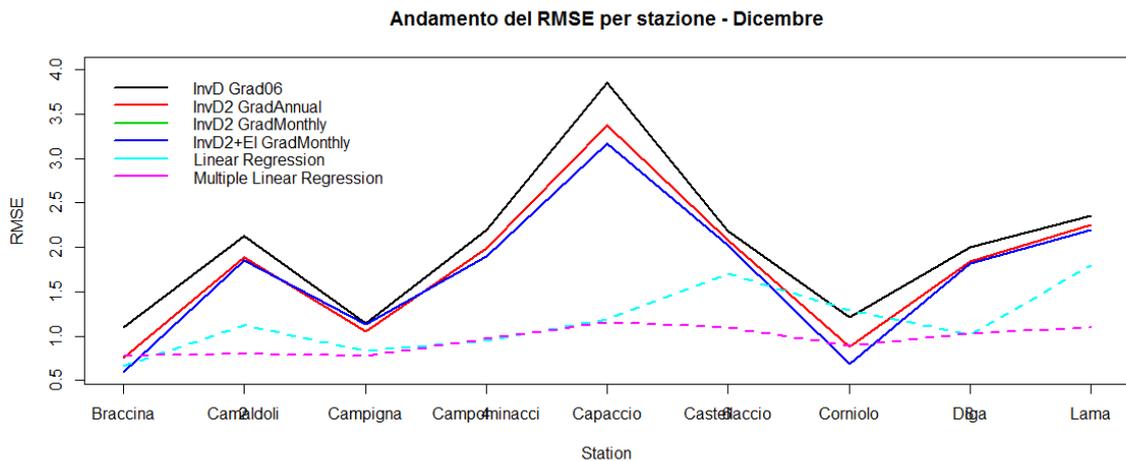


Figura 3.16 – Andamento dell'RMSE medio mensile sulle varie stazioni - dicembre

Come si può ben notare sia in Fig.3.3 che in Fig.3.4 l'interpolazione tramite la regressione multipla fornisce i risultati di gran lunga migliori, con un errore quadratico medio che, sia lungo la durata dell'anno e sia considerando le singole stazioni, non si spinge mai al di sopra del singolo grado circa. Segue in efficienza la regressione lineare sulla sola quota, con valori dell'RMSE di poco superiori (0,25°C circa) alla regressione multipla, considerando i valori medi mensili, ma più discontinui guardando l'andamento sulle varie stazioni. Si può quindi affermare con certezza che i metodi basati sulle regressioni lineari, utilizzando interpolazioni a passo

orario, hanno dato i migliori risultati sul territorio in esame, molto probabilmente per essere in grado di considerare le continue variazioni di gradiente termico a scala oraria, che evidentemente influenzano fortemente la stima.

Tutti gli altri metodi basati sulla media pesata, hanno dimostrato una peggiore efficienza, con un errore medio superiore di quasi un grado su tutto l'anno e fortemente discontinuo sulle varie stazioni; in particolare l'interpolazione tramite media pesata con gradiente fisso a priori di $-0,6^{\circ}\text{C}/100\text{m}$ è risultata particolarmente inadatta, probabilmente per il fatto che, come dimostrato anche dal risultato della stima del gradiente unico nel metodo 2, il valore standard di $-0,6^{\circ}\text{C}/100\text{m}$ non è rappresentativo della zona appenninica studiata. Si può notare come i tre metodi restanti, tutti basati su inversa distanza e gradiente termico valutato per la specifica area di studio, diano risultati molto simili, leggermente migliori rispetto al gradiente standard: coerentemente con i valori rappresentati in Fig.3.3, le differenze tra l'uso di gradiente annuale e mensile (metodo 2, rispetto al 3 e al 4), si risentono solo nei mesi di dicembre e gennaio, nei quali i valori mensili si distaccano maggiormente da quello annuale pari a $-0,39^{\circ}\text{C}/100\text{m}$; inoltre si può notare, come l'introduzione della differenza di quota nella valutazione dei pesi (metodo 4), non porta apprezzabili differenze con il metodo 3.

In generale si evince che l'errore risulta più consistente per tutti i metodi nei periodi dell'anno con temperature più estreme, dicembre – gennaio e luglio – agosto.

Nelle figure dalla 3.5 alla 3.16 è chiaramente visibile come per la stazione di Capaccio l'errore dei metodi basati su inversa distanza e gradiente termico sia quasi doppio rispetto a quello delle altre stazioni: questo probabilmente è dovuto al fatto che essa si trova molto decentrata e a quota sensibilmente inferiore rispetto all'area di studio (vedi in Fig.3.2); evidentemente questi metodi perdono affidabilità quando si va ad interpolare su punti con quota e posizione distanti dalla media di quelli in esame. Al contrario i metodi di regressione sembrano adattarsi bene su tutte le stazioni; tuttavia osservando le tabelle del paragrafo precedente, che riportano gli errori medi mensili per mese e per stazione, si vede come

per le stazioni di Braccina e Corniolo si hanno valori leggermente sopra la norma per la regressione multipla nei mesi invernali, più alti rispetto ai metodi che utilizzano un gradiente mensile; sull'area attorno a date stazioni (bacini dei torrenti Rabbi e Fiumicello) ci si aspetterà una imprecisione leggermente maggiore.

Dati i chiari risultati dell'analisi, si sceglie di interpolare la temperatura sui bacini con l'utilizzo della regressione lineare multipla su quota e posizione planimetrica.

3.5 Creazione delle mappe di temperatura

Una volta scelta la metodologia di interpolazione è necessario applicarla su tutta l'area di studio. Essa è discretizzata in celle elementari corrispondenti alle celle del modello di elevazione del terreno *DEM* utilizzato per la definizione dei bacini idrografici.

Tramite l'utilizzo dei software QuantumGIS e Grass viene estratta dal *DEM* del centro-nord Italia solo la porzione relativa all'estensione dei bacini dell'area di studio (Fig.3.17). Il file raster ricavato viene poi esportato in formato ESRI ARC-GRID, ottenendo in questo modo un file di testo leggibile da un software di calcolo come una matrice i cui elementi corrispondono proprio alle quote delle celle del modello di elevazione. Se una cella si trova al di fuori del perimetro dei bacini, le viene assegnato un valore "non numerico" *NA* per indicare che non appartiene all'area di interesse.

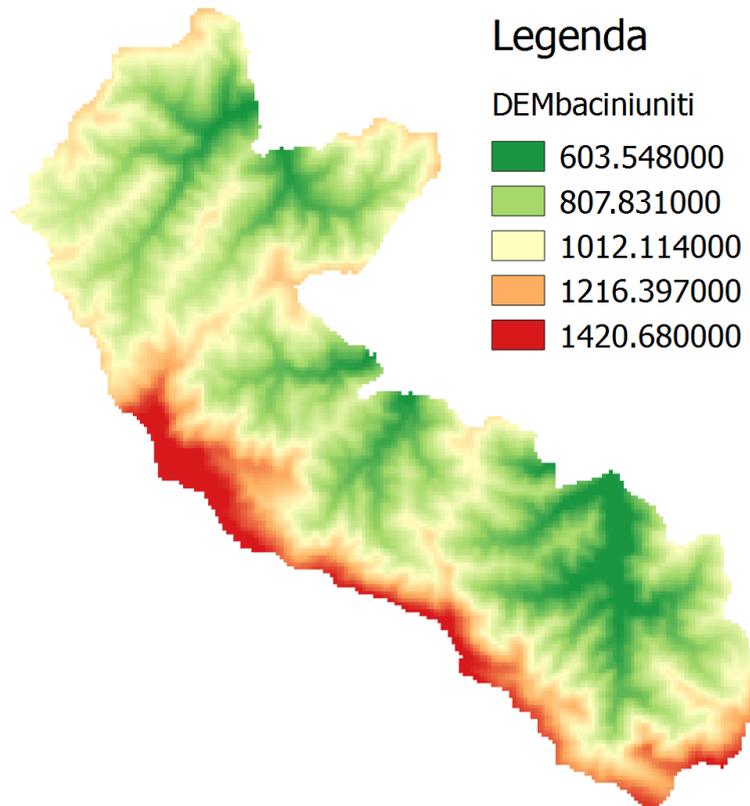


Figura 3.17 – DEM dell'area di studio

Nell'header del file sono affiancate le coordinate della prima cella in basso a sinistra (in questo caso in coordinate UTM zone 32 N) e la dimensione delle celle (in questo caso 80mx80m) tramite le quali è possibile costruire due matrici delle stesse dimensioni che riportano rispettivamente le coordinate X e Y (corrispondenti alle coordinate UTM Est e Nord) di ogni elemento della matrice.

A questo punto conoscendo coordinate X, Y e quota Z di ogni cella x, contenute nelle tre matrici costruite, è possibile ricavarne la temperatura ad ogni passo temporale tramite l'espressione 3.8 già nota:

$$T_{x,i} = C_{1,i} + Z_x * C_{2,i} + X_x * C_{3,i} + Y_x * C_{4,i}$$

Dove i coefficienti della regressione sono quelli stimati preventivamente ora per ora come riportato nella descrizione del metodo (paragrafo 3.2.3).

Anche queste operazioni sono state effettuate tramite l'implementazione di algoritmi appositamente costruiti nel linguaggio R. Gli cui script di calcolo sono riportati in Appendice1.

Il risultato dell'interpolazione consisterà in tante matrici di temperatura quanti sono i passi temporali, ovvero una mappa per ogni ora dal 1/1/1996 al 31/12/2014. In Fig.3.18 si riporta un esempio di mappa di temperatura visualizzata tramite QuantumGIS.

Queste matrici di temperatura possono rappresentare uno dei file di ingresso di un modello afflussi-deflussi distribuito. Nel caso in questione, in cui viene utilizzato un modello semi-distribuito o concentrato, sarà necessario delimitare le differenti zone di ogni bacino e mediane i valori di temperatura all'interno, operazione che verrà illustrata nel capitolo riguardante la preparazione dei dati in ingresso al modello.

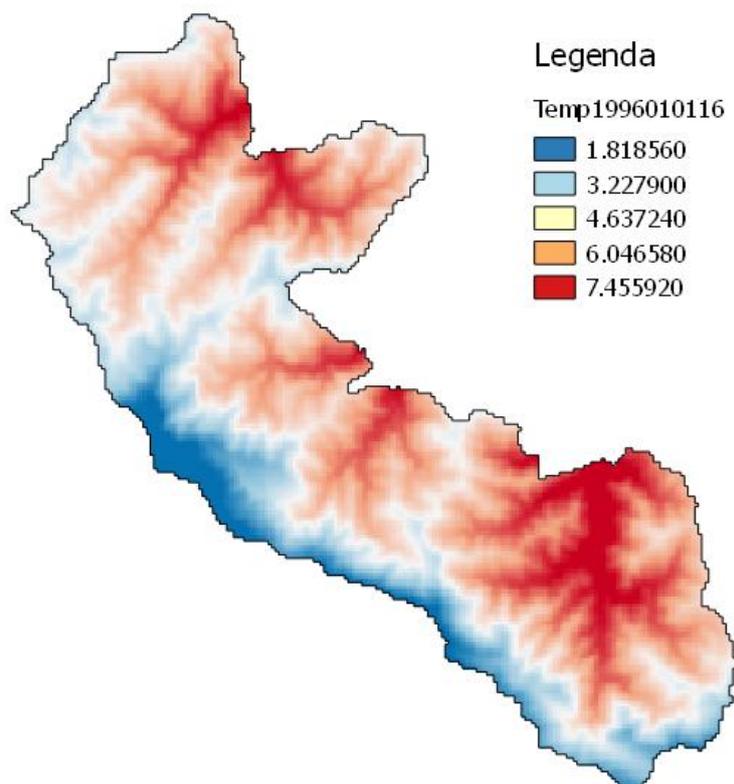


Figura 3.18 – Esempio di mappa temperatura creata con la regressione multipla rispetto a quota a posizione planimetrica

CAPITOLO 4

Calibrazione del modello afflussi - deflussi

4.1 Premessa

In questa sezione dell'elaborato si procede con l'applicazione di un modello afflussi-deflussi ai bacini compresi nell'area di studio in esame; questo verrà fatto sia in modalità concentrata che in modalità semi-distribuita, con l'obiettivo di confrontare i risultati della calibrazione derivanti dai due differenti approcci e valutarne i vantaggi in relazione alla differente complessità; le simulazioni saranno effettuate sia a passo orario che giornaliero.

La scelta del modello è ricaduta sul HBV che, come illustrato nel paragrafo che segue, è un modello concettuale che si adatta alle esigenze di questa applicazione.

L'applicazione in modalità semi-distribuita viene effettuata individuando, su ogni bacino, fasce di altitudine di circa 200/250 metri di dislivello e di conseguenza fornendo in input al modello distintamente i dati meteorologici relativi ad ogni fascia; l'idea alla base dello studio è quella di capire se l'aumento di dettaglio nella descrizione dei dati in ingresso, differenziando zone vallive e montane, porti un effettivo beneficio all'efficienza del modello, soprattutto per quanto riguarda l'influenza della temperatura su accumulo e scioglimento nivale alle differenti quote.

4.2 Il modello HBV

Il modello HBV è un modello afflussi-deflussi originariamente sviluppato presso l'Istituto di Meteorologia ed Idrologia Svedese (SMHI) a partire dal 1971. Negli anni è stato rielaborato sia dagli stessi autori che lo hanno

ideato, sia da altri studiosi che lo hanno adottato e adattato alle proprie necessità.

HBV nasce come modello concentrato ma può anche essere adottato in forma semi-distribuita, pensando di suddividere il bacino oggetto di studio in sotto-bacini, oppure in fasce altimetriche o ancora in aree che si differenziano perché caratterizzate dalla presenza di particolari elementi come laghi o una precisa tipologia di vegetazione. Nella maggioranza dei casi si adotta un passo temporale giornaliero ma esistono versioni di HBV, come quella utilizzata, che possono simulare a qualunque passo temporale. Questi rappresentano alcuni dei motivi per cui è stato scelto per l'analisi effettuata in questo elaborato, permettendo l'applicazione in fasce di altitudine e potendo simulare sia a scala giornaliera che oraria. Inoltre HBV è un modello di discreta complessità, infatti, come esposto in seguito, con i suoi 15 parametri si presta a una rappresentazione dettagliata di formazione e propagazione dei deflussi; nello specifico è implementato un accurato modulo neve descritto da un terzo dei parametri totali, aspetto essenziale che ne ha determinato la scelta.

Si è utilizzata la versione del modello HBV fornita dall'Università Tecnica di Vienna "TUWmodel" per ambiente R (A. Viglione e J. Parajka, <https://cran.r-project.org/web/packages/TUWmodel/index.html>), utilizzata e testata dagli autori stessi. Viene distribuito attraverso la piattaforma CRAN (Comprehensive R Archive Network).

Questa versione prevede, come l'originale, l'utilizzo sia in modalità concentrata che semi-distribuita, con la suddivisione di ciascun bacino in differenti fasce altimetriche all'interno delle quali si valuta sia il contenuto di umidità al suolo che l'accumulo e lo scioglimento nivale; a seconda della tipologia dei dati in input il modello è in grado di riconoscere automaticamente se si sta lavorando o meno a fasce; in questa trattazione lo si utilizzerà, come già detto, in entrambe le configurazioni, adottando un passo di calcolo sia giornaliero che orario e fornendo in ingresso, per ciascun bacino considerato, osservazioni di precipitazione cumulata oraria e giornaliera, temperatura ed evapotraspirazione potenziale media oraria e giornaliera.

Il modello è composto da tre moduli, descritti in seguito. In Fig.4.1 si riporta uno schema del modello HBV nella sua versione originale proposta dallo SMHI, dove sono però presenti alcune differenze importanti rispetto a quella utilizzata in questo studio, esposte durante la descrizione dei vari moduli.

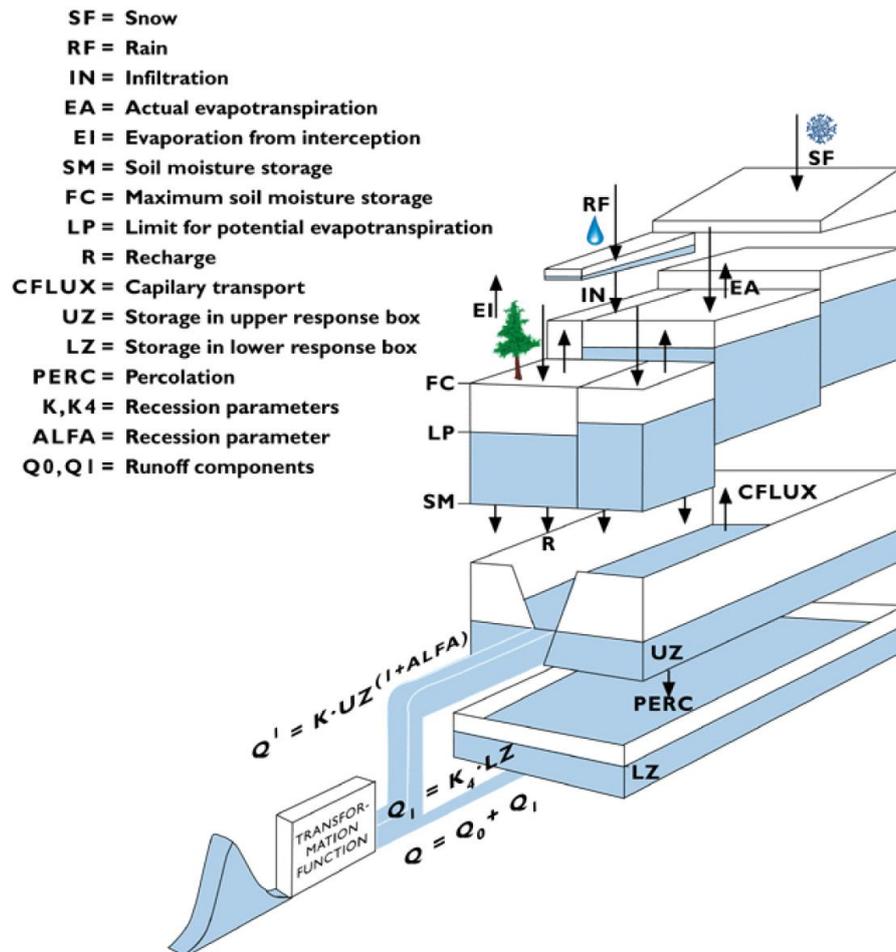


Figura 4.1 – Struttura generale del modello HBV nella versione dello SMHI e pensato come applicato ad un sotto-bacino.

1. Modulo Neve

Il modulo implementato in HBV per la rappresentazione dei processi di accumulo e scioglimento nivale segue un approccio gradi-giorno, che considera unicamente la temperatura come indicatore della presenza o meno di precipitazione in forma solida. Si prevede la distinzione della precipitazione in pioggia P_R o neve P_s , secondo le seguenti relazioni:

$$P_R = P \text{ se } T \geq T_R \quad (4.1)$$

$$P_R = P \frac{T - T_S}{T_R - T_S} \text{ se } T_S < T < T_R \quad (4.2)$$

$$P_R = 0 \text{ se } T_S \geq T \quad (4.3)$$

Dove T , T_R e T_S sono rispettivamente la temperatura dell'aria media giornaliera/oraria e le due temperature di soglia superiore e inferiore che delimitano i campi in cui la precipitazione si valuta come totalmente liquida, totalmente solida o parzialmente solida (si tratta di parametri del modello da calibrare in funzione della climatologia dell'area di studio). Lo scioglimento nivale M , è modellato considerando due ulteriori parametri del modello: DDF e T_M ; essi rappresentano il fattore gradi-giorno, cioè la quantità (in mm) di neve che si scioglie nell'intervallo giornaliero per ogni grado centigrado sopra la soglia, e la temperatura sopra alla quale si ha scioglimento. Si ha:

$$M = DDF * (T - T_M) \quad (4.4)$$

Inoltre si applica un'ulteriore coefficiente moltiplicativo SCF che tiene conto di eventuali errori di misura commessi nella rilevazione delle precipitazioni in forma solida, anch'esso parametro del modello. In questa versione del HBV, viene trascurato il fenomeno di intercettazione da parte della vegetazione; se ne tiene implicitamente conto calibrando il coefficiente SCF , che assorbe in parte gli errori dati da questa semplificazione.

2. Modulo "umidità del suolo"

Descrive la generazione di deflusso superficiale e contemporaneamente le variazioni di umidità, quindi di contenuto d'acqua, del suolo. L'approccio previsto dal modello HBV vuole semplificare i complicati processi di generazione del deflusso, ma al tempo stesso non vuole ricadere in

semplificazioni eccessive, ad esempio rappresentando il bacino come un serbatoio lineare. Per questo viene utilizzato un approccio probabilistico: si assume che le varie zone del bacino siano dotate di capacità di immagazzinamento differenti, rappresentabili attraverso una distribuzione statistica. Ci si aspetta che la pioggia o il contributo dello scioglimento nivale generino modesti contributi al deflusso quando il contenuto d'acqua presente nel suolo è modesto, mentre siano responsabili di un ingente deflusso quando il suolo ha un alto contenuto di umidità.

Il contenuto di umidità all'istante i -esimo $S_{SM,i}$ è espresso attraverso la relazione:

$$S_{SM,i} = S_{SM,i-1} + P_R + M - E_A \quad (4.5)$$

Dove $S_{SM,i-1}$ è il contenuto di umidità all'istante precedente e E_A indica l'evapotraspirazione effettiva. La porzione di pioggia e scioglimento nivale ΔS_{UZ} (R in figura) che contribuisce al deflusso superficiale è fornito dalla relazione:

$$\Delta S_{UZ} = \left(\frac{S_{SM}}{FC} \right)^{BETA} (P_R + M) \quad (4.6)$$

Dove FC e $BETA$ sono due ulteriori parametri del modello, da calibrare, che rappresentano rispettivamente la capacità di immagazzinamento del terreno e il parametro di non linearità.

L'evapotraspirazione effettiva E_A viene calcolata in funzione della evapotraspirazione potenziale E_P (dato in ingresso al modello) e del contenuto di umidità del suolo. Infatti fino a quando lo strato superficiale di suolo non si satura, l'evapotraspirazione sarà inferiore a quella potenziale:

$$E_A = E_P \frac{S_{SM}}{LP} \quad \text{se } S_{SM} < LP \quad (4.7)$$

$$E_A = E_P \quad \text{se } S_{SM} \geq LP \quad (4.8)$$

Dove LP è un ulteriore parametro che indica la capacità del suolo oltre la quale l'evapotraspirazione eguaglia quella potenziale.

3. Modulo di propagazione del deflusso

Descrive la propagazione del deflusso lungo le sponde del bacino ed è schematizzata con due serbatoi lineari sovrapposti in parallelo ma comunicanti fra loro, che rappresentano il comportamento di due differenti strati di suolo, risposta lenta e veloce del bacino. Le quantità di acqua immagazzinate, ai vari istanti temporali, nei due serbatoi vengono indicate come S_{UZ} ed S_{LZ} . La portata derivante dalla formazione del deflusso ΔS_{UZ} entra all'interno del serbatoio superiore (responsabile della risposta veloce del bacino), che descrive ciò che accade nello strato superficiale, e segue tre diversi percorsi a seconda delle condizioni dei serbatoi in ogni istante:

- Percolazione, verso il serbatoio inferiore che schematizza il comportamento della falda profonda, a una velocità costante C_{perc} e conseguente efflusso dallo stesso regolato dal parametro K_2 .
- Scarico al fondo dal serbatoio superiore, regolato dal parametro K_1 , quando la quantità liquida derivante dalla formazione del deflusso eccede quella percolata.
- Scarico tramite sfioro dal serbatoio superiore, modellato ricorrendo all'utilizzo di un terzo parametro K_3 , quando il contenuto d'acqua del serbatoio supera la capacità dello stesso L_{SUZ} .

Nella versione originale rappresentata in Fig.4.1 non è prevista la risposta "velocissima" del bacino tramite lo sfioro dal serbatoio superiore, infatti come si può notare, esso non prevede i parametri K_3 e L_{SUZ} , ma invece un ulteriore parametro ALFA in aggiunta alla costante K per la descrizione dell'efflusso dal serbatoio stesso.

La somma dei deflussi così generati dai serbatoi viene indicata con Q_G ; essa viene ulteriormente trasformata tramite una funzione di trasferimento triangolare che rappresenta la propagazione dei deflussi all'interno della rete:

$$B_Q = B_{max} - C_R * Q_G \quad \text{se} \quad (B_{max} - C_R * Q_G) \geq 1 \quad (4.9)$$

$$B_Q = 1 \quad \text{altrimenti} \quad (4.10)$$

Con B_{max} e C_R (*Croute*) ulteriori parametri del modello.

Riassumendo, il modello è governato da 15 parametri, 5 parametri per il modulo neve, 3 per il modulo di umidità del suolo e 7 per il modulo di propagazione del deflusso superficiale. Di seguito se ne riporta una tabella riassuntiva:

Parametro	Descrizione	Unità di misura
SCF	Coefficiente di correzione	[-]
DDF	Fattore gradi-giorno	[mm/(°C+imestep)]
Ts	T soglia per la neve	[°C]
Tr	T soglia per la pioggia	[°C]
Tm	T soglia di scioglimento	[°C]
LP	Limite umidità del suolo per evapotraspirazione	[mm]
FC	Capacità di immagazzinamento del suolo	[mm]
BETA	Parametro di non linearità	[-]
K0, K1, K2	Costanti di rilascio dei serbatoi	[imestep]
Cperc	Velocità di percolazione	[mm/imestep]
Lsuz	Capacità massima del serbatoio superiore	[mm]
Bmax	Base della funzione di trasferimento triangolare	[imestep]
Croute	Parametro di scala (per funzione triangolare)	[imestep^2/mm]

Anche nel caso di modellazione per fasce di altitudine, si utilizza ugualmente un unico set di parametri, che vengono quindi considerati invariati sui sottobacini. Questi parametri vengono calibrati con l'utilizzo di un algoritmo di ottimizzazione, argomento dei prossimi paragrafi.

Il modello fornisce in uscita, oltre che il valore della portata simulata totale q [mm/timestep], anche i seguenti oggetti:

- q_{zoned} portate simulate per le varie fasce [mm/timestep]
- q_0 efflusso rapido di superficie (serbatoio superiore) [mm/timestep]
- q_1 efflusso di superficie (serbatoio superiore) [mm/timestep]
- q_2 efflusso profondo di falda (serbatoio inferiore) [mm/timestep]
- r_{moist} umidità relativa del suolo [-]
- $rain$ precipitazione liquida
- $snow$ precipitazione solida
- $melt$ scioglimento nivale [mm/timestep]
- eta evapotraspirazione effettiva [mm/timestep]

Come si può notare, la simulazione avviene in termini di portate per unità di superficie, infatti l'area del bacino non entra all'interno dei calcoli e viene utilizzata soltanto per ricavare la portata effettiva sulla sezione di chiusura; nel caso semi-distribuito vengono utilizzate le percentuali di superficie di ogni fascia di altitudine per eseguire la media pesata dei contributi provenienti da ciascuna zona.

4.3 Preparazione dei dati in ingresso

Come già citato, il modello HBV, richiede in ingresso tre serie di dati: temperatura, evapotraspirazione potenziale e precipitazione. Avendo scelto di lavorare sia in modalità concentrata che semi-distribuita e sia a passo orario che giornaliero, sono stati inizialmente costruiti i dati in input per fasce di altitudine a scala oraria; successivamente, si sono aggregati a scala giornaliera e/o spazialmente, per eseguire gli altri tipi di simulazione.

4.3.1 Schematizzazione per utilizzo semi-distribuito e a scala oraria

1. Temperatura

È necessario ottenere un insieme di vettori per ogni bacino idrologico, ognuno contenente la temperatura media per una fascia di altitudine considerata, ad ogni intervallo temporale. Nel capitolo 3, è stato illustrato come, a seguito di un'indagine sulla migliore metodologia di interpolazione, sono state ricavate le mappe di temperatura per l'intera area di studio, tramite una regressione lineare multipla partendo dalle temperature delle stazioni di misura considerate. Per la creazione dei file di input al modello è quindi stato sufficiente, per ogni fascia di ogni bacino, mediare la temperatura delle celle che appartengono alla fascia stessa. Per comodità di calcolo, questo non è stato effettuato a partire dalle mappe di temperatura relative all'intera area di studio, comprendente tutti e 6 i bacini analizzati; si è proceduto invece secondo i seguenti passi per ogni bacino idrografico:

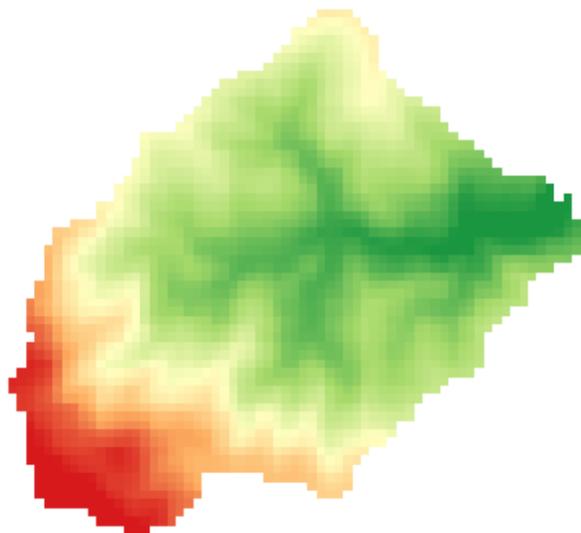


Figura 4.2 – Esempio: DEM Celle (estratto da figura 3.29)

- Tramite l'utilizzo dei software QuantumGIS e Grass, si è ritagliato dal DEM originale, solo la porzione relativa al bacino di interesse, diminuendo notevolmente le dimensioni del raster di calcolo (si riporta l'esempio del DEM di Celle in Fig.1).
- Per ogni bacino, si è individuato l'intervallo fra quota massima e minima e suddiviso in fasce di ampiezza fra i 200 e i 250 m circa. Successivamente, con l'utilizzo del "Calcolatore raster" dei due software, si sono raggruppate le celle della matrice nelle varie fasce e si sono create delle maschere raster, aventi valore unitario

nelle celle comprese nella fascia in questione, e NA (“non numerico”) nelle restanti. In tabella a fine paragrafo, si riporta la suddivisione in fasce dei vari bacini con quote e aree delle stesse, mentre nelle figure 4.3-8 ne viene riportata la visualizzazione grafica in ambiente GIS.

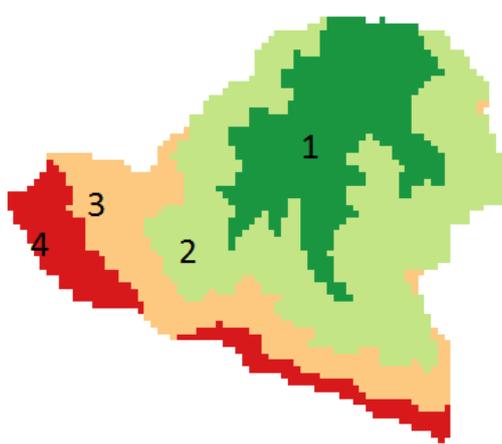


Figura 4.3 – Bacino Bidente di Campigna, suddivisione in fasce altimetriche

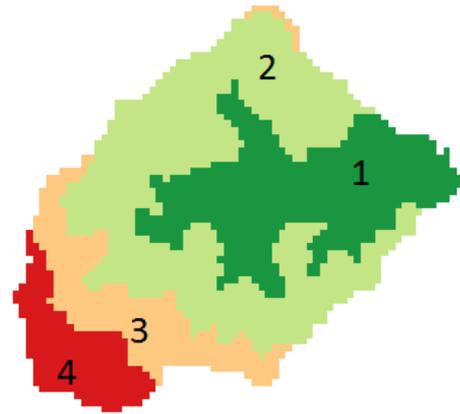


Figura 4.4 – Bacino Bidente di Celle, suddivisione in fasce altimetriche

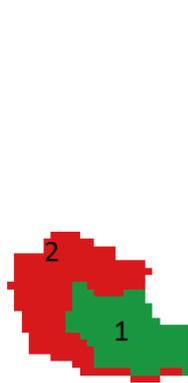


Figura 4.5 – Bacino Rio Bacine, suddivisione in fasce altimetriche

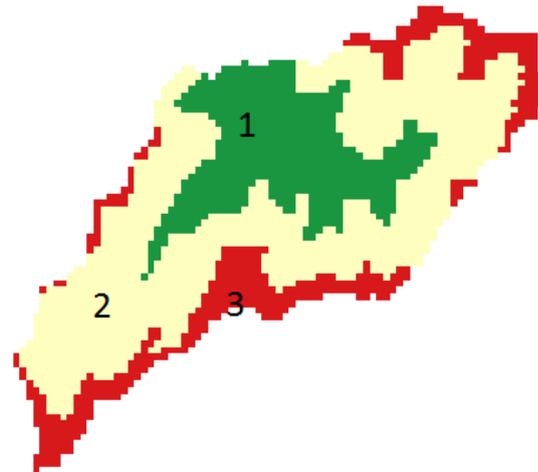


Figura 4.6 – Bacino Fiumicello, suddivisione in fasce altimetriche

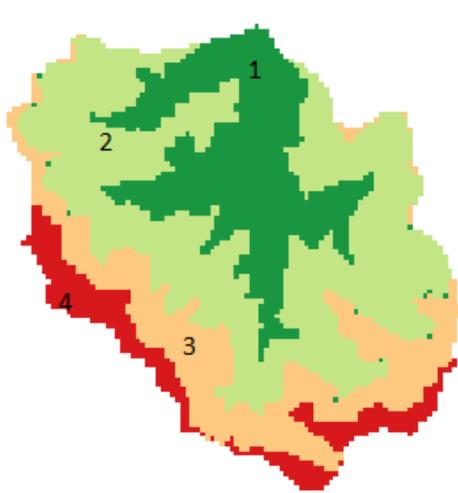


Figura 4.7 – Bacino Diretto (Ridracoli),
suddivisione in fasce altimetriche

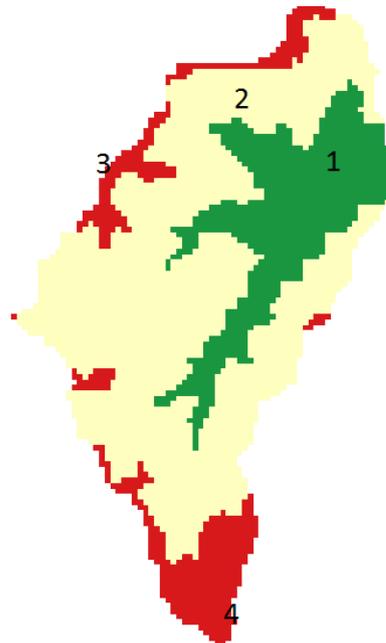


Figura 4.8 – Bacino Rabbi,
suddivisione in fasce altimetriche

- Con l'utilizzo del software R, dopo avere esportato i file raster in formato numerico, si sono create, come per le mappe del capitolo 3, le matrici contenenti le coordinate X e Y delle celle del DEM; successivamente, ad ogni passo temporale si è effettuata la regressione lineare multipla per ogni cella creando le mappe di temperatura. Si ricorda la formulazione della regressione lineare multipla:

$$T_{x,i} = C_{1,i} + Z_x * C_{2,i} + X_x * C_{3,i} + Y_x * C_{4,i} \quad (4.11)$$

Dove $C_{1,i}$ $C_{2,i}$ $C_{3,i}$ $C_{4,i}$ sono i coefficienti ricavati dalla regressione multipla ai minimi quadrati. Per evitare di dover salvare ogni mappa, ad ogni passo temporale si è moltiplicata direttamente la mappa di temperatura creata tramite la (4.11) per ognuna delle maschere relative alle varie fasce: si ottengono in questo modo tante matrici quante sono le fasce, ognuna delle quali contenente valori numerici solo sulla relativa porzione del bacino. Mediando queste ultime si sono ottenuti i valori di temperatura delle zone del bacino.

- Si è assemblata una matrice per ogni bacino, file di input al modello, avente tante righe quanti passi temporali e tante colonne quante fasce altimetriche.

Gli script costruiti sono riportati in Appendice2.

Bacino	n° fasce	Intervallo di quote [msl]	Area fascia [kmq]
Rio	2	577 – 832	0,92
Bacine		832 - 1086	1,28
Bidente di Campigna	4	589-829	4,64
		829 - 1069	7,93
		1069 - 1309	3,64
		1309 - 1548	2,05
Bidente di Celle	4	596 - 858	4,19
		858 - 1120	6,78
		1120 - 1382	2,20
		1382 - 1643	1,44
Bacino Diretto	4	479 - 733	9,32
		733 - 987	18,38
		987 - 1241	7,88
		1241 - 1494	3,35
Fiumicello	3	604 - 808	3,72
		808 - 1012	8,93
		1012 - 1217	3,08
Rabbi	4	576 - 812	5,58
		812 - 1048	16,74
		1048 - 1284	3,16
		1284 - 1522	0,40

2. Precipitazioni

Per le precipitazioni si è lavorato in maniera analoga a quanto eseguito con le temperature. In tabella a fine paragrafo sono riportate le caratteristiche delle stazioni pluviometriche (posizione in coordinate UTM fuso 32N, quota e alcune statistiche) che hanno fornito i dati utilizzati nell'analisi. Le mappe di precipitazione sono state in questo caso elaborate da un altro studente laureando, Gianluca Colliva, che ha lavorato sulla stessa area di studio, utilizzando però un modello pienamente distribuito. Analogamente a quanto effettuato in questo studio per le temperature, con una procedura di tipo jack-knife ha testato vari metodi di interpolazione delle precipitazioni sulla superficie del bacino. Il metodo più preciso e quindi più idoneo all'analisi è risultato essere l'interpolazione tramite media pesata dalle inverse distanze cubiche. In un generico punto x sul bacino, all'istante i -esimo si avrà:

$$P_{x,i} = \frac{\sum_{j=1}^n P_{j,i} w_{x,j}}{\sum_{j=1}^n w_{x,j}} \quad (4.12)$$

Dove:

$P_{j,i}$ è l'altezza di pioggia misurata dalla stazione j al passo i [°C]

$w_{x,j}$ è il peso della stazione j rispetto al punto x definito come: [m⁻³]

$$w_{x,j} = \frac{1}{d_{x,j}^3} \quad (4.13)$$

Con $d_{x,j}$ distanza fra il punto x e la stazione j [m]

Una volta in possesso delle mappe di pioggia distribuite su tutta la superficie di interesse, si sono create, analogamente che con le temperature, i vettori di dati di pioggia corrispondenti alle fasce ce costituiranno l'ingresso al modello semi-distribuito e a scala oraria: per ogni bacino e per ogni passo temporale, si sono sovrapposte le mappe di pioggia con le maschere delle varie fasce di ogni bacino, mediando poi successivamente i valori ottenuti all'interno delle stesse. In questo modo anche per le piogge si è ottenuta una matrice per ogni bacino avente tante

colonne quante fasce di altitudine e tante righe quanti passi temporali. Si riportano gli script utilizzati in Appendice2.

Nome Pluviometro	Est	Nord	Quota (mslm)	Media annua (mm/h)	Max orario (mm)
BadiaPrataglia	731446	4853038	845	0.2	44.4
Braccina	722816.4	4867134	958	0.1	39.0
Camaldoli	727035	4853040	1111	0.2	32.9
Campigna	720731.9	4861301	1060	0.2	48.4
Campominacci	725649.8	4859971	920	0.2	39.0
Capaccio	732146.7	4867637	295	0.1	63.2
Castellaccio	728726.1	4866038	705	0.1	33.8
Corniolo	724238.8	4867254	735	0.1	35.2
Diga	728086.7	4861361	565	0.2	40.6
Lama	728156.7	4857101	715	0.2	49.2
Lastra	732976.6	4865256	600	0.1	53.6
Premilcuore	722976	4873541	452	0.1	49.6

3. Evapotraspirazione potenziale

L'evapotraspirazione non è altro che l'altezza d'acqua che evapora dal terreno e dalla superficie vegetale. Essa comprende anche la traspirazione che consiste nell'estrazione da parte delle piante dell'acqua presente nel terreno. Si può discernere fra:

- *Evapotraspirazione effettiva*: indica la quantità liquida effettivamente riportata in atmosfera, funzione dell'umidità del suolo, dalle condizioni atmosferiche e dalle caratteristiche della superficie. Essa viene calcolata dal modello in funzione dell'evapotraspirazione potenziale.
- *Evapotraspirazione potenziale (ETP)*: è la quantità d'acqua che evaporerebbe da un terreno completamente ricoperto da vegetazione con approvvigionamento d'acqua limitato. Dipende solo da fattori climatici e limitatamente dal tipo di coltura. Per questo permette una sua stima matematica.

Per il calcolo dell'ETP si è utilizzato un metodo semplificato proposto da Blaney e Criddle (1920-1930), basato unicamente sulle misure di temperatura dell'area.

Con questo metodo si ricava l'evapotraspirato potenziale mensile ETP_{0d} come funzione lineare della temperatura:

$$ETP_{0d} = a + b * N(i) * W_{ta}(i) * T_m(i) \quad \left[\frac{mm}{mese} \right] \quad (4.14)$$

Dove:

$T_m(i)$ è la temperatura media dell'area in esame relativa al mese i-esimo [°C], calcolata a partire dalle matrici di temperatura dei vari bacini.

a e b sono i parametri caratteristici del modello da stimare

$N(i)$ esprime il valore mensile del numero massimo di ore di sole giornaliere tabulato in funzione della latitudine, in questo caso si è scelto, essendo l'area di studio molto limitata e posta proprio in corrispondenza della latitudine 44°Nord, di considerare un valore fisso tabellato proprio per quella stessa latitudine:

Mese	Gen	Feb	Mar	Apr	Mag	Giu	Lug	Ago	Set	Ott	Nov	Dic
N(i)	9.3	10.5	11.9	13.4	14.7	15.4	15.2	14	12.6	11	9.7	8.9

$W_{ta}(i)$ è un fattore correttivo dipendente dalla temperatura media mensile e dalla quota media del bacino o, nella presente applicazione semi-distribuita, dalla quota media della sottozona del bacino corrispondente alla fascia altimetrica. L'espressione di W_{ta} può essere ottenuta direttamente da opportune tabelle oppure attraverso l'interpolazione di una parabola di forma:

$$W_{ta}(i) = A * T_m^2(i) + B * T_m(i) + C \quad (4.15)$$

Dove A, B e C sono coefficienti da stimare.

Essendo, come già accennato i valori del fattore correttivo anche tabulati, funzione di temperatura e quota (500, 1000, 2000 m), per eseguire le regressioni e ricavare i parametri A, B e C si sono utilizzate tali tabelle,

interpolando linearmente i valori tabellati per le varie temperature alle quote note per le quote medie di ogni fascia di ogni bacino. In questo modo si sono ottenuti valori del coefficiente $W_{ta}(i)$ per varie temperature (quelle tabulate) e successivamente, interpolando, si sono ricavati i tre coefficienti necessari per la stima di $W_{ta}(i)$ per i vari mesi.

Avendo quindi tutto a disposizione, è stato possibile ricavare i coefficienti a e b per ogni fascia di ogni bacino e quindi, a partire dalla matrice di input delle temperature, stimare l'evapotraspirato medio orario. Sebbene con questo metodo si ricavi il valore di evapotraspirazione potenziale media mensile, è stato sufficiente dividere il risultato per il numero di intervalli temporali in un mese (orari in questo caso: $24 \cdot \text{numero giorni del mese}$). Si sono così costruite tabelle di dati di input, analoghe a quelle realizzate per pioggia e temperatura.

Nella tabella a fine paragrafo si riportano i valori di temperatura media mensile e dei coefficienti A, B, C, a e b ricavati per l'analisi per ogni fascia altimetrica di ogni bacino, mentre si riportano gli script utilizzati in Appendice2.

4.3.2 Aggregazione temporale e spaziale dei dati di input

Per l'utilizzo dei dati a scala giornaliera si è effettuata una semplice aggregazione temporale: per i dati di temperatura si è effettuata la media giornaliera, mentre per precipitazione e evapotraspirato si sono sommati i valori medi orari sull'intera giornata per ricavarne la cumulata giornaliera. Per effettuare le simulazioni in modalità concentrata (Lumped) è stato necessario aggregare spazialmente i valori di temperatura, evapotraspirato e precipitazione fra le varie fasce di ogni bacino idrografico considerato. Questo è stato fatto implementando direttamente all'interno dello script per la calibrazione del modello Lumped, una breve funzione che, prima dell'utilizzo dei dati, effettuasse la media pesata, rispetto alle aree delle zone del bacino, dei valori di input relativi a ciascuna di esse, ad ogni passo temporale (ovvero la media pesata di ogni riga delle matrici di input).

Il TUWmodel è infatti in grado di riconoscere automaticamente se si sta lavorando in modalità concentrata o semi-distribuita, a seconda del numero di vettori di dati in ingresso che vengono forniti (se è uno solo, si lavora in modalità Lumped).

Bacino	Fascia	Quota media	A	B	C	a	b	Temperatura media mensile											
								G	F	M	A	M	G	L	A	S	O	N	D
Bacine	1	704	-0.0002	0.0182	0.4155	1.868302	0.597728	1.78	1.86	5.24	8.68	13.41	17.41	19.65	19.66	14.84	11.1	6.35	2.58
	2	1918	-0.0002	0.0170	0.4542	2.051852	0.580399	1.79	1.74	4.99	8.43	13.35	17.5	19.87	20.04	14.98	11	6.02	2.56
Bidente di Campigna	1	709	-0.0002	0.0182	0.4156	3.0356	0.6154	1.2	1.08	4.35	7.77	12.4	16.3	18.44	18.48	13.82	10.3	5.74	1.94
	2	949	-0.0002	0.0190	0.4185	3.182848	0.597763	1.22	1	4.16	7.57	12.37	16.4	18.64	18.79	13.94	10.23	5.48	1.94
	3	1189	-0.0002	0.0187	0.4263	3.371921	0.602547	1	0.57	3.54	6.93	11.9	16.05	18.4	18.71	13.68	9.81	4.87	1.68
	4	1428	-0.0002	0.0181	0.4355	3.190238	0.604643	0.95	0.38	3.21	6.6	11.75	16.03	18.51	18.96	13.72	9.63	4.49	1.61
Bidente di Celle	1	877	-0.0002	0.0187	0.4176	3.38976	0.621673	0.96	0.73	3.97	7.37	12.01	15.89	18.04	18.12	13.47	9.97	5.45	1.67
	2	989	-0.0002	0.0191	0.4189	3.622419	0.606959	0.88	0.52	3.62	7.02	11.82	15.82	18.09	18.31	13.47	9.78	5.06	1.57
	3	1251	-0.0002	0.0186	0.4287	3.607862	0.611796	0.72	0.15	3.06	6.45	11.44	15.58	17.96	18.34	13.3	9.42	4.5	1.37
	4	1512	-0.0002	0.0179	0.4387	3.424661	0.615247	0.65	-0.1	2.64	6.01	11.22	15.52	18.05	18.6	13.31	9.18	4.01	1.27
Bacino Diretto	1	606	-0.0001	0.0179	0.4143	1.512221	0.601614	1.93	2.13	5.56	9.01	13.66	17.62	19.77	19.68	14.96	11.33	6.66	2.74
	2	860	-0.0002	0.0187	0.4174	1.638651	0.583472	1.98	2.08	5.4	8.84	13.67	17.78	20.05	20.1	15.15	11.29	6.4	2.79
	3	1114	-0.0002	0.0189	0.4234	1.773686	0.575755	1.94	1.89	5.05	8.48	13.5	17.74	20.12	20.3	15.15	11.1	5.99	2.72
	4	1367	-0.0002	0.0183	0.4331	1.661372	0.578906	1.86	1.66	4.67	8.09	13.29	17.68	20.18	20.52	15.15	10.89	5.56	2.62
Fiurmicello	1	706	-0.0002	0.0182	0.4155	2.811198	0.611181	1.25	1.1	4.43	7.85	12.56	16.5	18.76	18.88	14.1	10.4	5.76	1.99
	2	910	-0.0002	0.0189	0.4180	2.968356	0.596245	1.25	1	4.22	7.64	12.5	16.55	18.92	19.15	14.19	10.31	5.48	1.97
	3	1114	-0.0002	0.0189	0.4234	2.768001	0.588178	1.33	1.02	4.15	7.57	12.58	16.76	19.23	19.57	14.42	10.35	5.32	2.05
Rabbi	1	694	-0.0002	0.0181	0.4154	3.411668	0.628222	0.84	0.55	3.82	7.22	11.86	15.73	17.94	18.08	13.4	9.85	5.33	1.53
	2	930	-0.0002	0.0189	0.4182	3.810946	0.621139	0.61	0.14	3.24	6.63	11.4	15.35	17.64	17.9	13.1	9.44	4.8	1.27
	3	1166	-0.0002	0.0188	0.4254	3.742745	0.616292	0.59	0.01	3	6.39	11.3	15.35	17.72	18.09	13.13	9.32	4.51	1.23
	4	1403	-0.0002	0.0182	0.4345	3.645146	0.619505	0.53	-0.2	2.59	5.96	11.06	15.28	17.74	18.25	13.08	9.08	4.05	1.15

4.4 Calibrazione del modello

4.4.1 L'algoritmo di ottimizzazione

Qualunque modello idrologico di tipo concettuale, come quello qui utilizzato, deve essere calibrato. Calibrare il modello significa stimare i parametri che regolano la simulazione del deflusso (in questo caso quelli elencati nel paragrafo 4.2). Come suggerisce il nome, questi parametri sono rappresentazioni “concettuali” di caratteristiche, nella maggior parte dei casi, sostanzialmente astratte del bacino; per questo la calibrazione avviene tramite un processo iterativo, cercando di diminuire quanto più possibile lo scarto tra idrogramma simulato e osservato. In passato questo veniva fatto tramite taratura manuale, necessitando dell'esperienza di un esperto idrologo.

Oggi, per la calibrazione si utilizzano algoritmi automatici che sono in grado di ricercare la migliore configurazione di parametri. Questo viene fatto seguendo uno schema di ricerca predefinito: l'algoritmo viene assemblato con l'obiettivo di minimizzare una particolare funzione, che prende il nome di *funzione obiettivo*.

La funzione obiettivo è tipicamente una funzione che rappresenta appunto lo scarto fra i valori osservati e i valori in uscita del modello. Nel caso in questione si è scelto di utilizzare la media degli scarti al quadrato; all'interno della funzione obiettivo viene effettuata la simulazione con il set di parametri proposto dall'algoritmo ottimizzatore; in seguito viene fornito il risultato in uscita dalla funzione nella forma:

$$F.O. = \text{mean}(\text{simu} - \text{obse})^2 \quad (4.16)$$

Dove *obse* rappresenta il vettore delle portate osservate, mentre *simu* quello delle portate simulate dal modello.

Esistono numerosi algoritmi di ottimizzazione utilizzabili per la procedura di calibrazione, che però si suddividono in due classi principali:

- Metodi ricerca locale: metodi indicati per la ricerca del minimo di funzioni monotone; il problema di questi metodi sta nel fatto che, data la non linearità della maggior parte dei problemi di ottimizzazione, la funzione spesso presenta punti nello spazio dei parametri che non rappresentano minimi assoluti, bensì locali, avendo quindi efficienza limitata. Infatti il risultato dipenderà fortemente dal punto di inizio della ricerca.
- Metodi di ricerca globale: metodi che superano le difficoltà dei precedenti, utilizzando un approccio *multi-start*, ovvero sviluppando delle ricerche a partire da punti diversi, scelti casualmente, nello spazio dei parametri. In questo modo la probabilità di errore diminuisce fortemente.

Si è utilizzato l'algoritmo di ottimizzazione a ricerca globale chiamato DEoptim (D. Adria, K. Mullen, B. Peterson, J. Ulrich, K. Boudt), appartenente alla seconda classe, disponibile anch'esso sulla piattaforma CRAN (<https://cran.r-project.org/web/packages/DEoptim/index.html>), implementabile quindi nel linguaggio R. Esso è in grado di eseguire l'ottimizzazione di un'unica funzione obiettivo, richiedendo in ingresso la funzione stessa (definita precedentemente) e due vettori valori limite inferiore e superiore del campo di ricerca di ogni parametro. Per il TUWmodel utilizzato, si sono utilizzati i valori limite consigliati dagli autori del modello, che si riportano in seguito:

Parametro	L inf	L sup	Parametro	L inf	L sup
SCF	0.9	1.5	K0	0	2
DDF	0	5	K1	2	30
Tr	1	3	K2	30	250
Ts	-3	1	Lsuz	1	100
Tm	-2	2	Cperc	0	8
LP	0	1	Bmax	0	30
FC	0	600	Croute	0	50
BETA	0	20			

4.4.2 Valutazione del periodo di calibrazione

Come già discusso del paragrafo precedente, un modello idrologico di tipo concettuale come l'HBV deve essere calibrato, garantendo al tempo stesso che i parametri ottenuti possano riprodurre correttamente anche condizioni diverse da quelle del periodo di taratura. A questo scopo è necessario effettuare una rigorosa procedura di cross-validazione: si suddivide l'intervallo di simulazione (dove si posseggono i dati) in due sotto-intervalli, uno dei quali sarà utilizzato appunto per la calibrazione del modello. Il secondo invece verrà impiegato, come spiegato nei prossimi paragrafi, per validare la calibrazione effettuata e valutarne l'efficienza. Per la scelta del periodo più idoneo, si è effettuata una calibrazione a dati giornalieri in modalità concentrata (essendo la configurazione con minori tempi di calcolo) su tutto il periodo di dati, per tutti i bacini. Successivamente, con i parametri risultanti si è validato il modello sempre sull'intero periodo, effettuandone la simulazione. Si sono calcolate le efficienze (RMSE ed Efficienza di Nash, la cui formulazione è riportata nel paragrafo che segue) per ciascun anno. Come si può ben notare dal grafico che segue, l'efficienza è tendenzialmente inferiore sulla prima parte dell'intervallo di dati; per questo si è scelto di dividere circa a metà il periodo, utilizzando per la calibrazione la porzione con efficienza media inferiore: in particolare si sono scelti i dati dal 1 gennaio 1996 al 31 agosto 2005 (chiusura dell'anno idrologico).

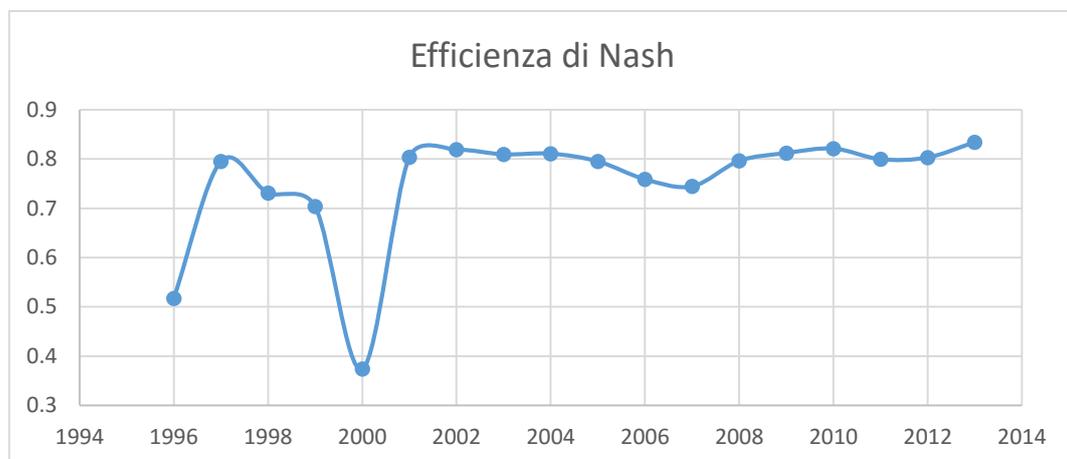


Figura 4.9 – Efficienza di Nash media fra i bacini sui vari anni per la valutazione del periodo di calibrazione

4.4.3 Calibrazione e risultati

4.4.3.1 Calibrazione dei bacini con misure di portata sia giornaliera che oraria

Rispettando quanto detto in precedenza è stato calibrato il modello per tutti i sei bacini oggetto di studio, sia in configurazione concentrata (*Lumped* in tabella) che semi-distribuita per dati orari e giornalieri. Si è utilizzato un periodo di *warm-up* di un anno (coincidente quindi con l'intero anno 1996), per entrambe le serie di dati. Il *warm-up* è la porzione iniziale della serie di dati utilizzati per la calibrazione che viene esclusa dal calcolo della F.O. per evitare l'influenza delle condizioni iniziali del bacino (rappresentato dai valori delle variabili di stato) sulla simulazione, che porterebbero a una stima poco efficiente dei parametri; consiste in un vero e proprio "riscaldamento" che si lascia fare al modello, infatti con l'avanzare della simulazione diminuisce l'effetto dell'errore dovuto alla mancata conoscenza delle condizioni iniziali.

Nelle prossime pagine si riportano i valori dei parametri calibrati per tutti i bacini idrografici e per tutte le simulazioni, ad eccezione dei bacini del Rabbi e del Fiumicello per i quali si è effettuata una calibrazione differente, esposta in seguito.

Parametri	Bacino Diretto (Ridracoli)					
	Dati giornalieri		Dati orari		Limiti	
	Lumped	Fasce	Lumped	Fasce	Inf	Sup
SCF	1.005	1.012	1.116	0.973	0.9	1.5
DDF	3.24E-05	5.01E-04	9.24E-07	8.57E-06	0	5
Tr	2.977	2.989	2.998	2.903	1	3
Ts	-2.995	-2.996	-0.639	-0.056	-3	1
Tm	1.580	1.865	1.373	1.957	-2	2
LP	0.969	0.926	0.636	0.797	0	1
FC	520.037	432.695	348.884	453.387	0	600
BETA	10.166	8.587	4.841	6.585	0	20
K0	1.055	0.996	0.238	0.249	0	2
K1	2.655	2.727	29.693	29.710	2	30
K2	237.773	248.002	214.280	193.492	30	250
Lsuz	31.037	32.083	20.242	21.075	1	100
Cperc	3.794	3.492	0.544	0.515	0	8
Bmax	28.486	27.643	1.757	3.340	0	30
Croute	23.854	25.139	16.302	20.020	0	50

Parametri	Rio Bacine					
	Dati giornalieri		Dati orari		Limiti	
	Lumped	Fasce	Lumped	Fasce	Inf	Sup
SCF	0.900	0.900	0.901	0.900	0.9	1.5
DDF	1.465	1.596	0.029	0.046	0	5
Tr	1.963	1.520	1.655	1.082	1	3
Ts	-2.993	-2.986	-2.306	-1.056	-3	1
Tm	-1.558	-1.097	-1.582	-1.865	-2	2
LP	0.015	0.021	3.09E-04	0.003	0	1
FC	208.015	207.282	250.615	248.018	0	600
BETA	1.445	1.441	3.275	3.169	0	20
K0	1.002	1.002	0.319	0.316	0	2
K1	3.717	3.575	29.970	29.966	2	30
K2	249.271	237.072	116.845	120.977	30	250
Lsuz	23.795	24.774	24.467	24.697	1	100
Cperc	1.343	1.312	0.407	0.414	0	8
Bmax	29.630	29.718	0.792	1.715	0	30
Croute	40.143	40.552	29.248	29.427	0	50

Bidente di Campigna						
Parametri	Dati giornalieri		Dati orari		Limiti	
	Lumped	Fasce	Lumped	Fasce	Inf	Sup
SCF	0.900	0.919	0.900	0.901	0.9	1.5
DDF	1.751	1.745	0.089	0.093	0	5
Tr	1.318	1.219	1.110	1.018	1	3
Ts	-2.996	-2.993	-1.983	-1.720	-3	1
Tm	-1.014	-1.099	-0.056	-0.316	-2	2
LP	0.841	0.867	7.71E-01	0.809	0	1
FC	343.376	367.186	349.928	389.035	0	600
BETA	5.504	6.057	6.172	8.056	0	20
K0	1.691	1.607	0.241	0.368	0	2
K1	4.257	4.308	29.964	23.212	2	30
K2	30.977	30.313	134.483	120.580	30	250
Lsuz	25.037	24.356	21.050	38.257	1	100
Cperc	2.463	2.456	0.483	0.548	0	8
Bmax	22.260	18.598	1.820	8.895	0	30
Croute	24.988	20.765	47.557	4.793	0	50

Bidente di Celle						
Parametri	Dati giornalieri		Dati orari		Limiti	
	Lumped	Fasce	Lumped	Fasce	Inf	Sup
SCF	0.900	0.900	0.900	0.908	0.9	1.5
DDF	2.059	2.472	0.135	0.115	0	5
Tr	1.010	1.001	1.848	1.211	1	3
Ts	-2.999	-2.999	-2.986	-2.347	-3	1
Tm	-1.224	-1.595	-1.184	-1.559	-2	2
LP	0.919	0.926	0.876	0.857	0	1
FC	394.199	409.480	382.352	378.969	0	600
BETA	9.784	10.404	9.892	10.314	0	20
K0	1.007	1.022	0.341	0.344	0	2
K1	2.046	2.212	28.503	28.581	2	30
K2	30.013	30.064	140.649	135.675	30	250
Lsuz	23.757	21.994	15.072	15.459	1	100
Cperc	2.378	2.229	0.369	0.323	0	8
Bmax	18.382	14.914	1.424	2.603	0	30
Croute	49.697	48.033	32.426	36.345	0	50

		Valore Funzione obbiettivo (mm/tstep)			
Scala	Modalità di simulazione	Diretto	Bacine	Campigna	Celle
Giornaliera	Lumped	3.84	4.765	3.92	5.117
	Fasce	3.81	4.757	3.877	5.103
Oraria	Lumped	0.011	0.014	0.009	0.013
	Fasce	0.016	0.014	0.009	0.013

In generale si può notare la tendenza dei parametri a stabilirsi intorno agli stessi valori sia nel caso di calibrazione semi-distribuita che lumped, mentre passando da scala oraria a giornaliera la configurazione subisce una drastica variazione; questo è un risultato che ci si aspettava, in quanto, essendo il modello di tipo concettuale, il cambio del passo temporale di simulazione porta a uno sconvolgimento delle modalità di invaso e trasformazione dei deflussi governate da parametri non fisicamente basati.

Sono stati evidenziati in colore giallo i valori assunti dai parametri “schiacciati” sulla soglia inferiore o superiore imposta dall’algoritmo. Come si può notare questo accade frequentemente per le temperature di soglia T_r e T_s , quasi sempre per le calibrazioni a scala giornaliera, mentre molto meno per le calibrazioni orarie, spesso per il coefficiente correttivo SCF schiacciato sulla soglia inferiore, mentre più saltuariamente per le costanti K_1 e K_2 dei serbatoi. Per capire se questo accadesse per difficoltà del modello a simulare il comportamento di questi bacini o semplicemente per un valore non adatto della soglia, si è tentato, per ognuno di essi, abbassando o alzando le soglie stesse, di farli stabilizzare. Questo non ha portato a risultati soddisfacenti, in quanto, in tutte le configurazioni di calibrazione effettuate, essi rimanevano schiacciati sul nuovo valore di soglia. Inoltre, essendo i parametri spesso dipendenti l’uno dall’altro (soprattutto per quanto riguarda le due costanti dei serbatoi) andando a variare le soglie si rischiava di diminuire non indifferentemente l’efficienza. Si è quindi deciso inizialmente di lasciare invariati i limiti inferiori e superiori consigliati dagli autori.

4.4.3.2 Calibrazioni in base a sole portate osservate a scala giornaliera

Per i bacini afferenti alle stazioni di Fiumicello e Rabbi si possiedono solo dati a scala giornaliera; per le calibrazioni a scala oraria, si è perciò tentato l'utilizzo di una funzione obiettivo differente, dove il vettore *simu* sia anch'esso un valore giornaliero (e quindi confrontabile con le osservazioni disponibili), che viene costruito mediando i valori orari di portata sull'intera giornata. Questo tipo di calibrazione ha portato a risultati piuttosto insoddisfacenti in entrambe le modalità di simulazione a scala oraria sia lumped che semi-distribuita: sebbene l'algoritmo di ottimizzazione fosse in grado di minimizzare bene la funzione obiettivo, ottenendo dei valori di errore medio in linea con quello degli altri bacini calibrati a scala oraria (viene confrontata la portata media oraria in mm/timestep), la portata simulata presentava delle anomalie, con fluttuazioni continue e innaturali, come riportato nell'esempio in Fig.4.10. Il comportamento risulta simile a quello osservato per la simulazione a scala oraria con approccio semi-distribuito per il bacino del Bidente di Campigna: essa subiva delle oscillazioni per nulla realistiche, specialmente in corrispondenza dei rami di esaurimento dei picchi di piena, se ne riporta un esempio in Fig.4.11.

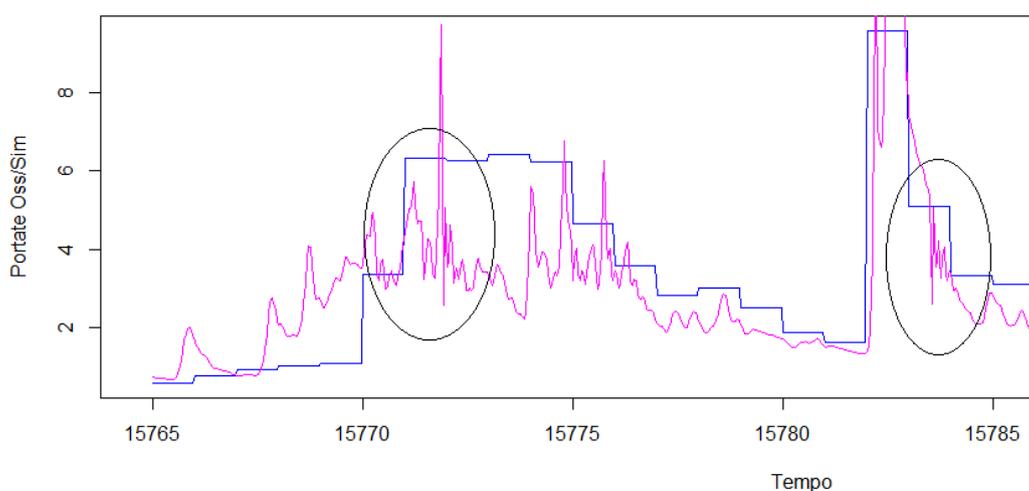


Figura 4.10 – Esempio dell'andamento anomalo della portata simulata del torrente Rabbi (rosa) con fluttuazioni innaturali rispetto alla portata osservata (blu)

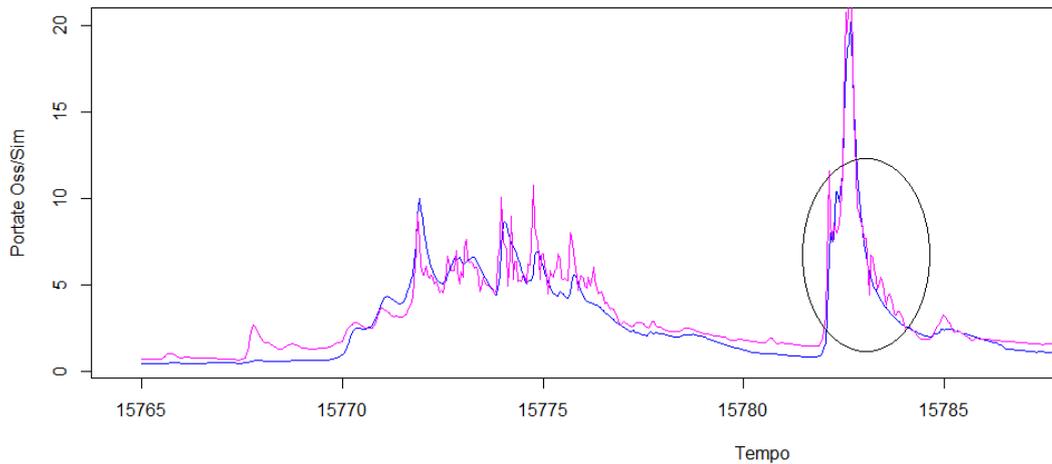


Figura 4.11 – Esempio anomalia sulla portata simulata (rosa) del Bidente di Campigna a scala oraria in modalità semi-distribuita

Si riportano nelle tabelle seguenti le configurazioni di parametri inizialmente ottenuti per i due bacini, ricavati tramite la calibrazione sopra descritta.

Parametri	Fiumicello					
	Dati giornalieri		Dati orari (calib. giorn.)		Limiti	
	Lumped	Fasce	Lumped	Fasce	Inf	Sup
SCF	0.901	0.901	0.904	0.901	0.9	1.5
DDF	3.898	3.847	0.315	0.286	0	5
Tr	1.003	1.007	1.002	1.004	1	3
Ts	-2.441	-2.921	-2.636	-2.670	-3	1
Tm	0.594	0.589	1.968	1.822	-2	2
LP	0.842	0.840	0.781	0.776	0	1
FC	235.136	233.939	253.535	255.185	0	600
BETA	4.545	4.473	4.939	5.051	0	20
K0	1.016	1.029	0.221	0.235	0	2
K1	2.799	2.712	28.000	25.174	2	30
K2	30.126	30.060	126.713	130.186	30	250
Lsuz	28.303	27.718	15.950	17.858	1	100
Cperc	1.725	1.742	0.403	0.411	0	8
Bmax	16.035	13.343	10.401	10.279	0	30
Croute	46.137	34.116	12.956	12.168	0	50

Parametri	Rabbi					
	Dati giornalieri		Dati orari (calib. giorn.)		Limiti	
	Lumped	Fasce	Lumped	Fasce	Inf	Sup
SCF	1.001	1.028	1.029	1.122	0.9	1.5
DDF	4.328	4.422	0.199	0.148	0	5
Tr	1.005	1.000	1.000	1.000	1	3
Ts	-3.000	-2.998	-2.998	-2.999	-3	1
Tm	-0.303	-0.361	0.529	-0.524	-2	2
LP	0.874	0.844	0.749	0.767	0	1
FC	296.499	285.970	300.788	308.781	0	600
BETA	7.142	7.068	5.923	6.100	0	20
K0	1.007	1.026	0.464	1.792	0	2
K1	2.842	2.862	29.934	29.495	2	30
K2	30.035	30.163	118.730	130.772	30	250
Lsuz	34.295	33.480	25.940	26.430	1	100
Cperc	1.599	1.593	0.317	0.304	0	8
Bmax	11.149	12.679	15.054	18.379	0	30
Croute	44.626	47.367	17.725	23.288	0	50

Valore Funzione obbiettivo (mm/tstep)			
Scala	Modalità di simulazione	Fiumicello	Rabbi
Giornaliera	Lumped	2.262	2.863
	Fasce	2.265	2.863
Oraria	Lumped	0.003	0.004
	Fasce	0.003	0.004

4.4.3.3 Nuove calibrazioni per risolvere problemi identificati nelle simulazioni a scala oraria

Per cercare di individuare più facilmente quali parametri potessero essere maggiormente responsabili delle anomalie riscontrate, si è plottato l'andamento degli stessi, normalizzati rispetto ai loro limiti, per le simulazioni di tutti i bacini nelle due modalità a scala oraria (Fig.4.12 e 4.13). Ci si è accorti come proprio per le simulazioni che presentavano problemi (Campigna semi-distribuito e Rabbi e Fiumicello per entrambe le modalità) i parametri B_{max} e $Croute$ avessero un andamento opposto rispetto agli altri bacini, visibile anche dalle tabelle (evidenziati in colore rosso); in aggiunta, per Campigna, si nota stranamente una decisa differenza fra i parametri calibrati in approccio concentrato e semi-distribuito, contrariamente a quanto osservato per i restanti bacini.

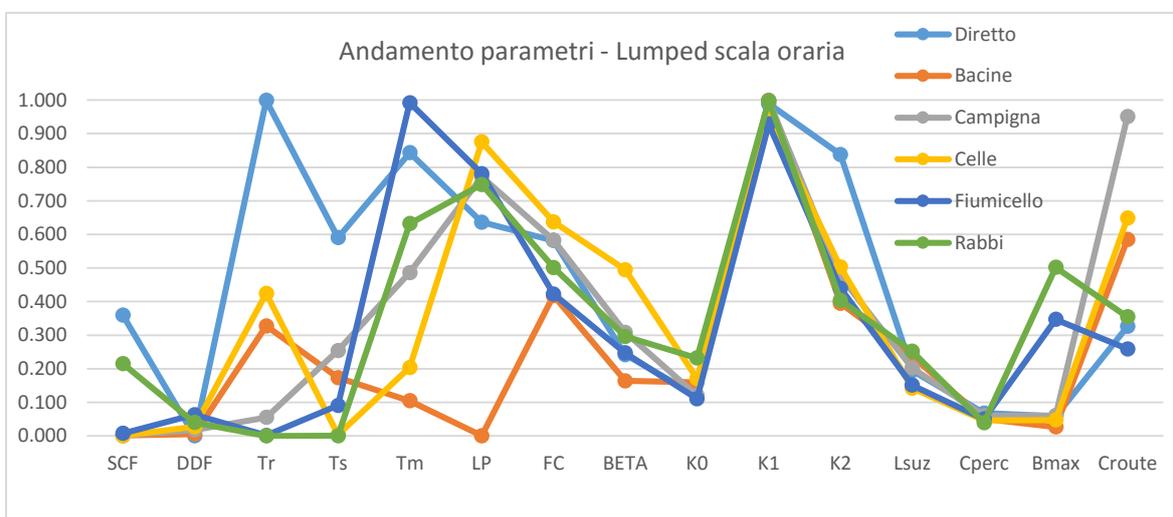


Figura 4.12 – Andamento parametri normalizzati per calibrazione oraria lumped

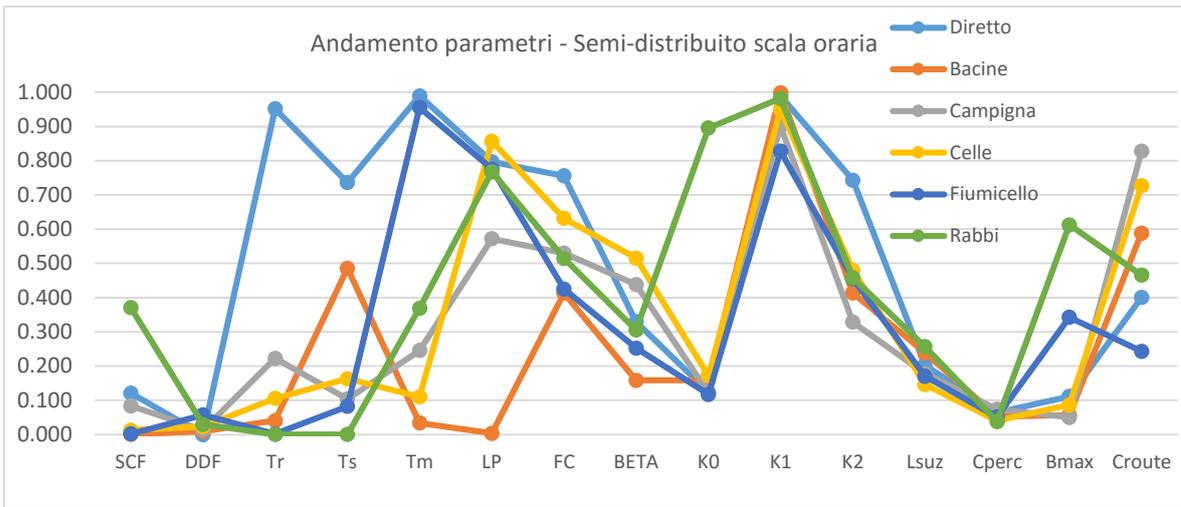


Figura 4.13 – Andamento parametri normalizzati per calibrazione oraria semi-distribuita

Si è deciso di modificare il campo di ricerca per la calibrazione di questi due parametri: si è alzata la soglia minima di *Croute* a 25 e si è abbassata quella massima di *Bmax* a 5, cercando in questo modo di mantenere la stessa tendenza dei parametri di tutte le altre simulazioni a scala oraria; successivamente si sono effettuate nuovamente le calibrazioni in questione. Segue il nuovo set dei limiti dei parametri.

Parametri	Vecchi limiti		Nuovi limiti	
	Inf	Sup	Inf	Sup
SCF	0.9	1.5	0.9	1.5
DDF	0	5	0	5
Tr	1	3	1	3
Ts	-3	1	-3	1
Tm	-2	2	-2	2
LP	0	1	0	1
FC	0	600	0	600
BETA	0	20	0	20
K0	0	2	0	2
K1	2	30	2	30
K2	30	250	30	250
Lsuz	1	100	1	100
Cperc	0	8	0	8
Bmax	0	30	0	5
Croute	0	50	25	50

Bidente di Campigna (cambio limiti parametri)				
Parametri	Dati orari		Nuovi limiti	
	Fasce (vecchi)	Fasce(nuovi)	Inf	Sup
SCF	0.901	0.950	0.9	1.5
DDF	0.093	0.066	0	5
Tr	1.018	1.446	1	3
Ts	-1.720	-2.585	-3	1
Tm	-0.316	-1.018	-2	2
LP	0.809	0.572	0	1
FC	389.035	318.003	0	600
BETA	8.056	8.766	0	20
K0	0.368	0.242	0	2
K1	23.212	27.235	2	30
K2	120.580	102.271	30	250
Lsuz	38.257	18.601	1	100
Cperc	0.548	0.591	0	8
Bmax	8.895	1.511	0	5
Croute	4.793	41.380	25	50

I risultati ottenuti per Campigna sono ora molto buoni, non si presentano le oscillazioni e aumenta notevolmente l'efficienza generale (Fig.4.14); i parametri tornano ad essere in linea con quelli derivanti dalla calibrazione in modalità concentrata.

Per una calibrazione futura si potrebbe quindi valutare la possibilità di modificare, per tutti i bacini calibrati a scala oraria, il set dei limiti dei parametri su questi valori. Di fatto le soglie indicate dagli autori sono tali da adattarsi ad una grande varietà di bacini e sono stati messi a punto per bacini appartenenti a contesti idroclimatici piuttosto diversi da quelli appenninici, ed è perciò ragionevole identificare valori di soglia specifici per quest'area di studio; con queste impostazioni sarà meno probabile ottenere simulazioni non realistiche come accaduto per il bacino in questione.

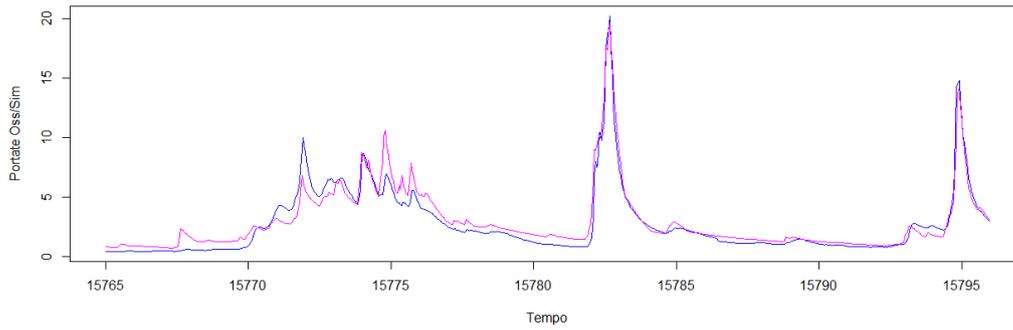


Figura 4.14 – Esempio di simulazione oraria e in modalità semi-distribuita per il Bidente di Campagna dopo la correzione delle soglie dei parametri

Per quanto riguarda Rabbi e Fiumicello, ci si è focalizzati sul miglioramento della calibrazione oraria per la sola modalità lumped, a costo computazionale decisamente inferiore. Il cambio delle soglie dei due parametri (*Croute* e *Bmax*) porta a una modifica dei parametri e un miglioramento delle simulazioni anche per questi bacini diminuendo leggermente le oscillazioni (Fig.4.15); tuttavia il risultato non è dei migliori, persistono molte micro oscillazioni ed un comportamento a spezzata delle curve di esaurimento.

Parametri	Fiumicello (cambio limiti parametri)	
	Dati giornalieri	Dati orari (calib. giorn.)
	Lumped (vecchie soglie)	Lumped (nuove soglie)
SCF	0.904	0.908
DDF	0.315	0.312
Tr	1.002	1.071
Ts	-2.636	-2.732
Tm	1.968	1.964
LP	0.781	0.546
FC	253.535	218.185
BETA	4.939	4.221
K0	0.221	0.163
K1	28.000	18.251
K2	126.713	101.398
Lsuz	15.950	14.892
Cperc	0.403	0.538
Bmax	10.401	0.026
Croute	12.956	44.830

Parametri	Rabbi (cambio limiti parametri)	
	Dati giornalieri	Dati orari (calib. giorn.)
	Lumped (vecchie soglie)	Lumped (nuove soglie)
SCF	1.029	1.075
DDF	0.199	0.195
Tr	1.000	1.081
Ts	-2.998	-2.958
Tm	0.529	0.220
LP	0.749	0.700
FC	300.788	285.022
BETA	5.923	5.604
K0	0.464	0.433
K1	29.934	29.367
K2	118.730	113.861
Lsuz	25.940	22.611
Cperc	0.317	0.370
Bmax	15.054	2.013
Croute	17.725	28.861

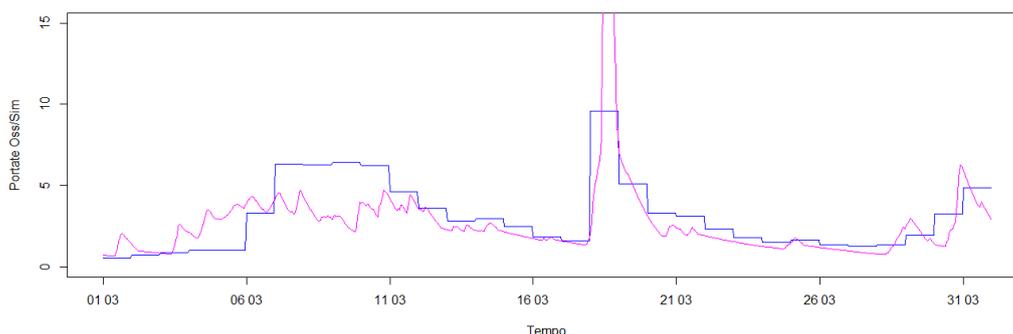


Figura 4.15 – Esempio di simulazione oraria in modalità lumped per il Torrente Rabbi dopo la correzione delle soglie dei parametri

Rabbi e Fiumicello, essendo calibrati rispetto a dati giornalieri, sono di per sé più soggetti a oscillazioni della portata; infatti, dato che l'algoritmo ricerca la configurazione di parametri che minimizza lo scarto fra media giornaliera delle portate simulate e portata media giornaliera osservata, non si tiene conto delle oscillazioni orarie.

Per questo si è ipotizzata una soluzione alternativa al cambio delle soglie dei parametri: l'idea è di minimizzare le fluttuazioni della portata, imponendo il minor numero possibile di picchi negativi o positivi; si è

operato andando a modificare la funzione obiettivo, rendendola una somma pesata di due componenti da minimizzare.

La prima componente è sempre lo scarto quadratico, anche se si è scelto di utilizzare, piuttosto che la media dei residui la loro somma; la seconda componente da minimizzare ha lo scopo di ridurre le fluttuazioni e cioè il numero di volte in cui la portata simulata passa da una crescita a un calo: per far ciò si è calcolato il numero di cambi di “direzione” della portata (crescente o decrescente).

L’espressione della somma dei residui assume la forma:

$$\sum(simu - obse)^2 \quad (4.17)$$

Dove *obse* rappresenta il vettore delle portate osservate, mentre *simu* quello delle portate simulate dal modello.

Per ricavare invece il numero di cambi di direzione si sono seguiti i seguenti passaggi:

- Si è creato un vettore *D* la cui componente *i*-esima venisse ricavata sottraendo a ogni termine *i*-esimo del vettore *simu* il termine precedente in posizione (*i*-1):

$$D(i) = simu(i) - simu(i - 1) \quad (4.18)$$

- Si è creato un vettore *S* che indicasse il segno di ogni elemento del vettore *D*:

$$S(i) = segno(D(i)) \quad (4.19)$$

- A questo punto si è eseguita la stessa operazione applicata al vettore *simu* su *S*, ottenendo un nuovo vettore *C*:

$$C(i) = S(i) - S(i - 1) \quad (4.20)$$

- Il numero di elementi non nulli di *C*, che si è chiamato *Nc* rappresenta il numero di cambi di “direzione” della portata.

Per il implementare il calcolo di tale funzione si sono utilizzate le funzioni “sign” e lagged difference” del linguaggio R, come si può osservare in Appendice 2.

Essendo l’algoritmo DEoptim un ottimizzatore mono-obiettivo, per poter minimizzare entrambe le componenti, si è dovuta eseguire una media

pesata dei loro valori. I valori in uscita si sono normalizzati rispetto al loro massimo stimato (rispettivamente 400 mm/tstep per la somma dei residui e 20000 per il numero di cambi di direzione) ed i pesi dei due termini sono stati individuati iterativamente (0.8 per la somma dei residui e 0.2 per il numero di cambi di direzione). A seguito di numerosi tentativi si è riuscito a migliorare discretamente i risultati attraverso la seguente espressione:

$$\frac{\text{sum}(\text{simu} - \text{obse})^2}{400} * 0.8 + \frac{Nc}{20000} * 0.2 \quad (4.21)$$

Gli script elaborati con le nuove funzioni obiettivo sono riportati in Appendice2.

Tale modifica risulta più efficace ed il risultato ottenuto si avvicina maggiormente a quello sperato per entrambi i bacini: lo smorzamento delle oscillazioni è più consistente e l'idrogramma simulato è ora accettabile (esempio per torrente Rabbi in Fig. 4.16 e 4.17).

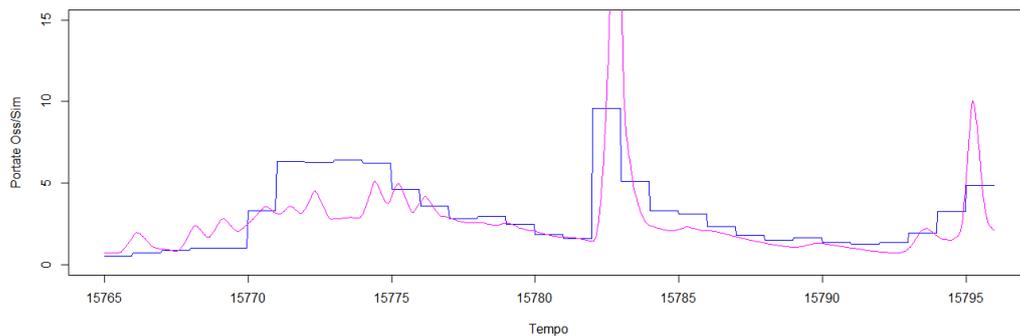


Figura 4.16 – Esempio di simulazione a scala oraria in modalità lumped per il torrente Rabbi dopo la modifica della funzione obiettivo (plot per 1 mese)

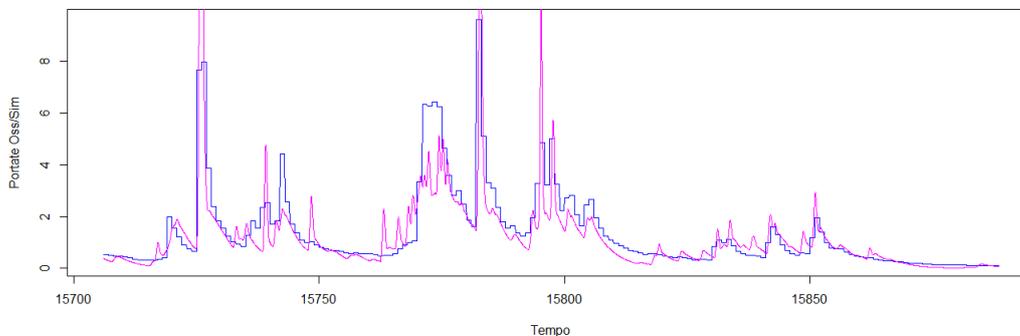


Figura 4.17 – Esempio di simulazione a scala oraria in modalità lumped per il torrente Rabbi dopo la modifica della funzione obiettivo (plot per 6 mesi)

Si è deciso per tanto di mantenere i set di parametri ricavati per Fiumicello e Rabbi con la modifica della funzione obiettivo.

Parametri	Rabbi (cambio F.O.)	
	Dati orari (calib. giorn.)	Dati orari (calib. giorn.)
	Lumped (vecchia F.O.)	Lumped (nuova F.O.)
SCF	1.029	1.080
DDF	0.199	0.163
Tr	1.000	1.004
Ts	-2.998	-2.890
Tm	0.529	-0.268
LP	0.749	0.721
FC	300.788	323.526
BETA	5.923	6.675
K0	0.464	1.205
K1	29.934	4.076
K2	118.730	94.087
Lsuz	25.940	50.059
Cperc	0.317	0.773
Bmax	15.054	29.610
Croute	17.725	4.860

Parametri	Fiumicello (cambio F.O.)	
	Dati orari (calib. giorn.)	Dati orari (calib. giorn.)
	Lumped (vecchia F.O.)	Lumped (nuova F.O.)
SCF	0.904	0.960
DDF	0.315	0.238
Tr	1.002	1.037
Ts	-2.636	-2.279
Tm	1.968	1.011
LP	0.781	0.524
FC	253.535	219.623
BETA	4.939	18.603
K0	0.221	0.735
K1	28.000	9.073
K2	126.713	177.574
Lsuz	15.950	66.997
Cperc	0.403	0.716
Bmax	10.401	29.781
Croute	12.956	9.073

4.4.3.4 Utilizzo portate orarie dal 2009 al 2014 per Fiumicello

Per il bacino di Fiumicello sono disponibili anche le portate orarie comprese fra il 1 maggio 2009 e il 31 dicembre 2014. Al fine di rispettare la procedura di cross-validazione definita al paragrafo 4.4.2 non sarebbe corretto calibrare solo questo bacino su un periodo differente. Tuttavia è stata eseguita comunque la calibrazione su tale intervallo temporale (in modalità concentrata) per possedere un ulteriore termine di paragone fra i parametri ottenuti. Seguono le tabelle con i nuovi parametri corretti per Bidente di Campigna, Rabbi e Fiumicello, quest'ultimo calibrato sia rispetto ai dati giornalieri che rispetto ai dati orari disponibili.

Parametri	Fiumicello
	Dati orari (calibrazione su periodo 1/5/2009 - 31/12/2014)
	Lumped
SCF	1.320
DDF	0.044
Tr	1.007
Ts	-2.580
Tm	-0.575
LP	0.093
FC	264.488
BETA	2.673
K0	0.282
K1	29.969
K2	134.171
Lsuz	10.720
Cperc	0.261
Bmax	0.198
Croute	33.422

4.5 Validazione del modello

La validazione di un modello idrologico consiste nella simulazione su un intervallo temporale i cui dati di input non siano stati utilizzati per la calibrazione dei parametr. In questo caso, come già accennato, per tutti i bacini idrografici si è utilizzato il periodo compreso fra l'1 settembre 2005 e il 31 dicembre 2014.

La validazione permette di stimare le prestazioni del modello idrologico, quindi la sua affidabilità nella previsione delle portate sulla sezione di chiusura, su dati indipendenti da quelli utilizzati per la taratura dei parametri. L'analisi delle prestazioni si avvale del calcolo di diversi indici prestazionali in funzione delle differenze fra portata osservata e simulata, che vengono riportati in seguito:

- 1) *RMSE*, scarto quadratico medio fra portata osservata e simulata:

$$RMSE = \sqrt{\frac{\sum_{t=1,N}[Q_{obs}(t) - Q_{sim}(t)]^2}{N}} \quad (4.17)$$

Dove $Q_{obs}(t)$ e $Q_{sim}(t)$ sono rispettivamente portate osservate e misurate e N il numero totale dei dati di portata.

- 2) *NSEff*, efficienza di Nash:

$$NSEff = 1 - \frac{\sum_{t=1,N}[Q_{obs}(t) - Q_{sim}(t)]^2}{\sum_{t=1,N}[Q_{obs}(t) - \mu_{obs}]^2} \quad (4.18)$$

Dove μ_{obs} rappresenta la media delle portate osservate.

- 3) *MAE*, errore medio assoluto:

$$MAE = \frac{\sum_{t=1,N}|Q_{obs}(t) - Q_{sim}(t)|}{N} \quad (4.19)$$

- 4) *Bias*, media degli errori con i relativi segni, per valutare la tendenza alla sovra/sottostima delle portate:

$$Bias = \frac{\sum_{t=1,N}[Q_{sim}(t) - Q_{obs}(t)]}{N} \quad (4.20)$$

Fra questi l'indice ampiamente più utilizzato per valutare la bontà di una simulazione rispetto ad un'altra è l'efficienza di Nash, con la quale si possono confrontare anche risultati provenienti da differenti bacini e per la quale il valore 1 corrisponde a simulazioni esattamente coincidenti con le osservazioni (simulazione 'perfetta').

Di seguito si riportano le tabelle delle efficienze sui periodi di calibrazione e validazione, per tutte le modalità di simulazione, dei quattro bacini di cui si posseggono sia portate osservate giornaliere che orarie su tutto il periodo di riferimento.

BACINO DIRETTO (Ridracoli)				
Validazione				
Indici di Efficienza	Dati giornalieri		Dati orari	
	Lumped	Fasce	Lumped	Fasce
RMSE (mm/tstep)	2.710	2.713	0.128	0.128
NSEff (-)	0.712	0.711	0.776	0.776
Bias (mm/tstep)	0.601	0.593	0.012	0.012
MAE (mm/tstep)	1.381	1.371	0.055	0.055
Calibrazione				
Indici di Efficienza	Dati giornalieri		Dati orari	
	Lumped	Fasce	Lumped	Fasce
RMSE (mm/tstep)	1.959	1.952	0.105	0.127
NSEff (-)	0.766	0.768	0.698	0.778
Bias (mm/tstep)	0.633	0.619	0.012	0.010
MAE (mm/tstep)	1.183	1.170	0.049	0.054

RIO BACINE				
Validazione				
Indici di Efficienza	Dati giornalieri		Dati orari	
	Lumped	Fasce	Lumped	Fasce
RMSE (mm/tstep)	4.296	4.294	0.256	0.255
NSEff (-)	0.689	0.689	0.625	0.628
Bias (mm/tstep)	-0.236	-0.229	-0.015	-0.015
MAE (mm/tstep)	1.610	1.605	0.063	0.063
Calibrazione				
Indici di Efficienza	Dati giornalieri		Dati orari	
	Lumped	Fasce	Lumped	Fasce
RMSE (mm/tstep)	2.183	2.181	0.118	0.119
NSEff (-)	0.745	0.743	0.679	0.680
Bias (mm/tstep)	0.395	0.401	0.011	0.011
MAE (mm/tstep)	1.120	1.118	0.045	0.045

BIDENTE DI CAMPIGNA				
Validazione				
Indici di Efficienza	Dati giornalieri		Dati orari	
	Lumped	Fasce	Lumped	Fasce
RMSE (mm/tstep)	2.252	2.176	0.105	0.103
NSEff (-)	0.845	0.855	0.847	0.851
Bias (mm/tstep)	0.276	0.279	0.010	0.007
MAE (mm/tstep)	1.075	1.038	0.045	0.046
Calibrazione				
Indici di Efficienza	Dati giornalieri		Dati orari	
	Lumped	Fasce	Lumped	Fasce
RMSE (mm/tstep)	1.980	1.969	0.095	0.097
NSEff (-)	0.799	0.801	0.792	0.783
Bias (mm/tstep)	0.126	0.128	0.003	2.7E-05
MAE (mm/tstep)	0.969	0.955	0.041	0.043

BIDENTE DI CELLE				
Validazione				
Indici di Efficienza	Dati giornalieri		Dati orari	
	Lumped	Fasce	Lumped	Fasce
RMSE (mm/tstep)	2.821	2.817	0.132	0.132
NSEff (-)	0.827	0.828	0.853	0.853
Bias (mm/tstep)	0.030	0.022	5.00E-04	3.00E-05
MAE (mm/tstep)	1.080	1.072	0.043	0.042
Calibrazione				
Indici di Efficienza	Dati giornalieri		Dati orari	
	Lumped	Fasce	Lumped	Fasce
RMSE (mm/tstep)	2.262	2.259	0.113	0.114
NSEff (-)	0.760	0.761	0.770	0.767
Bias (mm/tstep)	-0.019	-0.028	-0.002	-0.003
MAE (mm/tstep)	0.962	0.957	0.039	0.039

Come si può notare il risultato generale della validazione è abbastanza buono, infatti si sono ottenute delle efficienze di Nash tutte intorno allo 0.8, tranne che per il Rio Bacine, dove però ce lo si aspettava; infatti, per prima cosa la dimensione molto ridotta del bacino porta a un minor effetto “filtro” sugli errori di stima dei dati di input, inoltre ha una posizione e un’orientazione particolari, tali da conferirgli un’esposizione solare molto più bassa della media in questa regione; infine è piuttosto scoperto per quanto riguarda i sensori di misura di precipitazione e temperatura, come si può notare dalle mappe nel capitolo 1.

È evidente inoltre come l’approccio semi-distribuito non porti significativi miglioramenti in termini di bontà delle prestazioni, sia a scala di simulazione giornaliera che oraria. A conferma di questo, nelle Fig.4.18-21 vengono riportati, per esempio, i grafici sia a scala oraria che giornaliera che paragonano le portate osservate e simulate coi due approcci (lumped e semi-distribuito) sulle sezioni di chiusura dei bacini Bidente di Campigna e Bidente di Celle nel primo semestre dell’anno 2013. Si nota come esse, a entrambe le scale, siano praticamente coincidenti per la maggior parte dell’anno, salvo per alcuni intervalli molto brevi nei quali si discostano leggermente.

Validazione scala oraria Bidente di Campigna 1/1/2013 - 30/6/2013

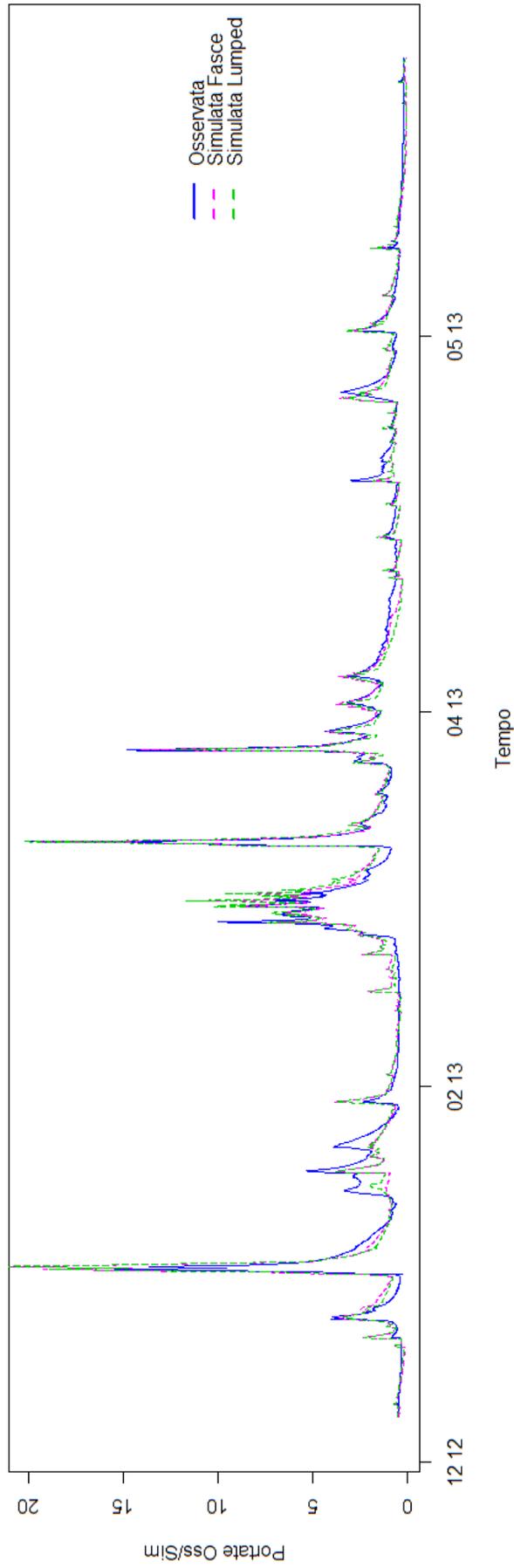


Figura 4.18 – Esempio validazione portate a scala oraria Bidente di Campigna e paragone fra portate simulata Lumped e a fasce di altitudine

Validazione scala giornaliera Bidente di Campigna 1/1/2013 - 30/6/2013

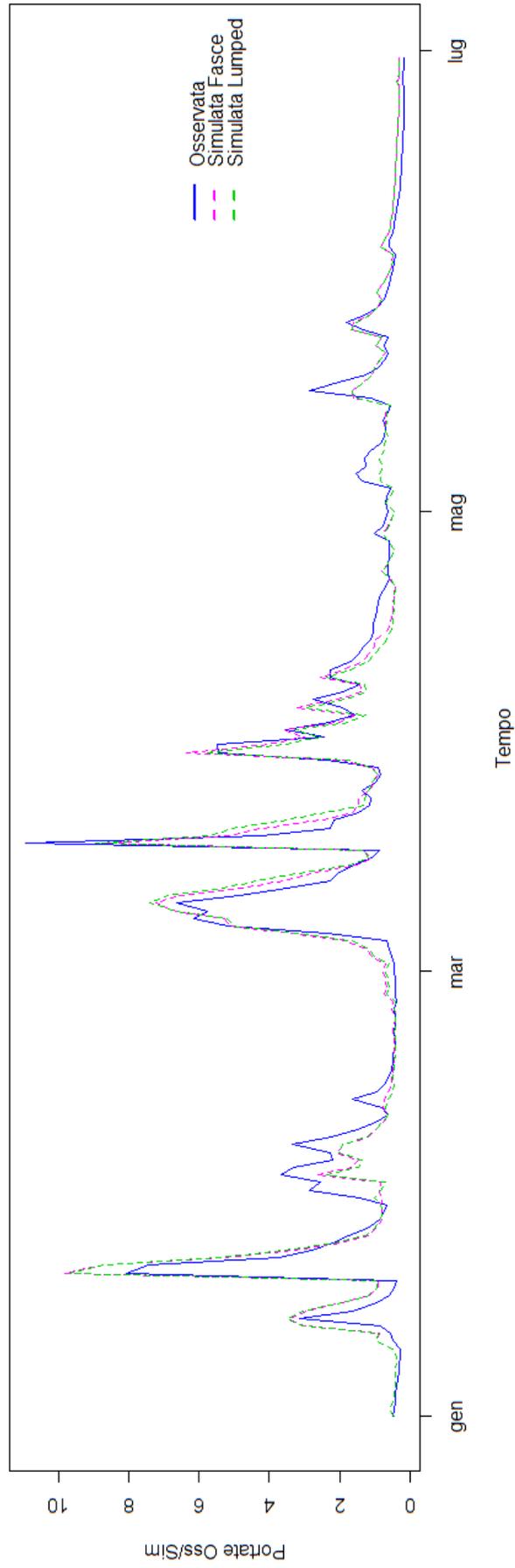


Figura 4.19 – Esempio validazione portate a scala giornaliera Bidente di Campigna e paragone fra portate simulata Lumped e a fasce di altitudine

Validazione scala oraria Bidente di Cella 1/1/2013 - 30/6/2013

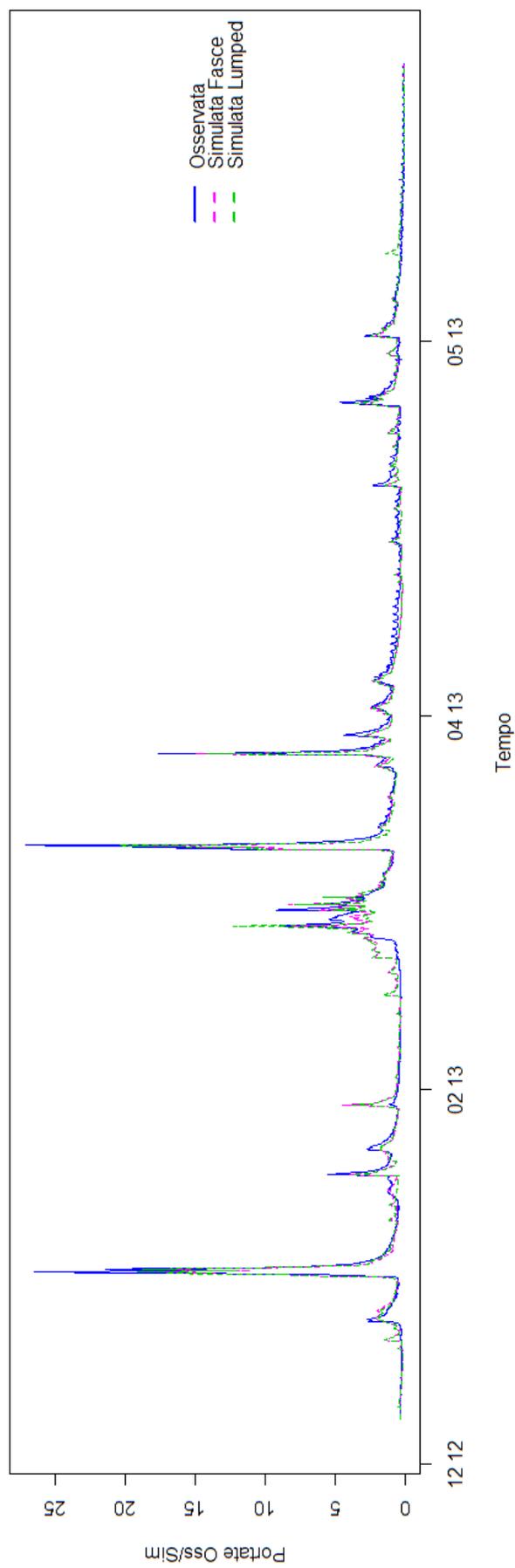


Figura 4.20 – Esempio validazione portate a scala oraria Bidente di Cella e paragone fra portate simulata Lumped e a fasce di altitudine

Validazione scala giornaliera Bidente di Cella 1/1/2013 - 30/6/2013

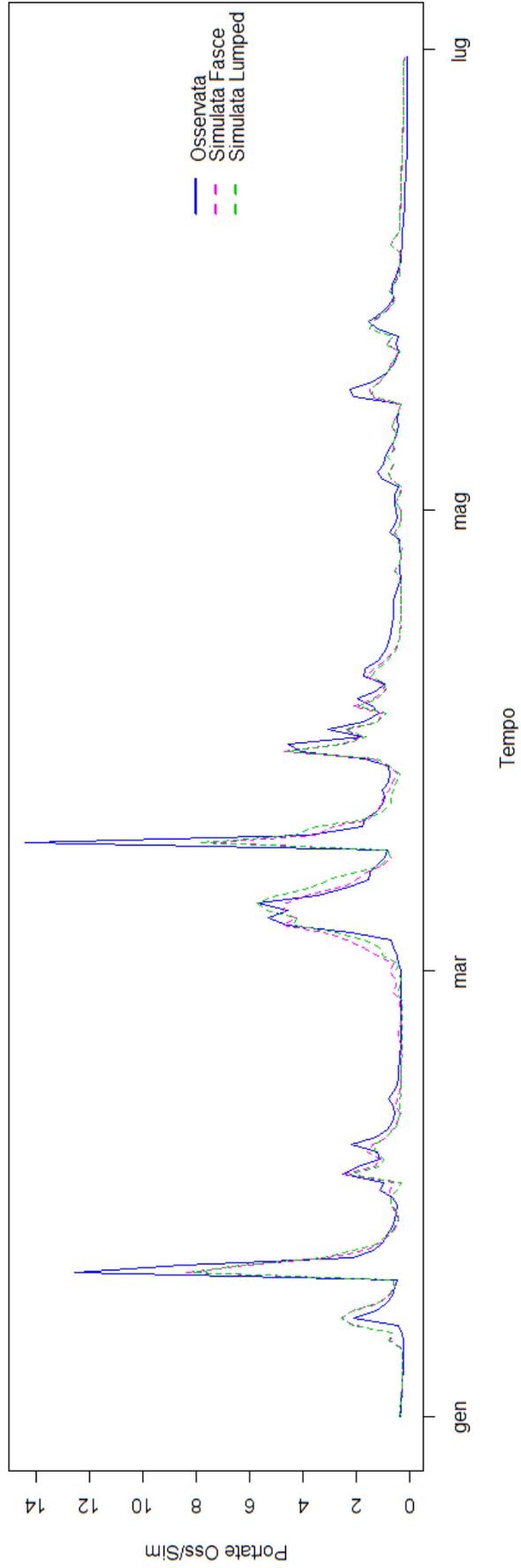


Figura 4.21 – Esempio validazione portate a scala giornaliera Bidente di Cella e paragone fra portate simulata Lumped e a fasce di altitudine

Per quanto riguarda i bacini del torrente Rabbi e di Fiumicello, dove le uniche portate disponibili sono quelle a scala giornaliera, si è operato analogamente a quanto eseguito in calibrazione. Per la validazione del modello a scala oraria vengono calcolati gli indici di efficienza rispetto alla media giornaliera delle portate simulate. Nelle tabelle che seguono se ne riportano i risultati. Sono ben visibili le differenze di efficienza rispetto alle calibrazioni svolte su portate osservate a scala oraria, soprattutto per quando riguarda Fiumicello. Di quest'ultimo, essendo in possesso anche delle portate orarie a partire dal 1/5/2009, si è effettuata, come riportato in tabella, un'ulteriore validazione a partire da suddetta data confrontando direttamente le portate orarie come per i bacini precedenti. Anche qui si notano gli effetti negativi della calibrazione eseguita rispetto ai dati giornalieri.

In Fig.4.22 viene riportata, sempre a titolo di esempio, la validazione a scala oraria per il bacino del Fiumicello (sempre relativa ai primi sei mesi del 2013), nella quale vengono plottate portate orarie simulate in modalità concentrata insieme alle portate giornaliere medie osservate estese sulle 24 ore.

Nelle tabelle che seguono le validazioni orarie sono riferite ai parametri definitivi utilizzati, riportati a pag. 79.

RABBI			
Validazione			
Indici di Efficienza	Dati giornalieri		Dati orari (valid. giorn)
	Lumped	Fasce	Lumped
RMSE (mm/tstep)	1.704	1.705	0.076
NSEff (-)	0.809	0.809	0.780
Bias (mm/tstep)	0.106	0.108	0.002
MAE (mm/tstep)	0.786	0.774	0.027
Calibrazione			
Indici di Efficienza	Dati giornalieri		Dati orari (valid. giorn)
	Lumped	Fasce	Lumped
RMSE (mm/tstep)	1.692	1.692	0.066
NSEff (-)	0.763	0.763	0.794
Bias (mm/tstep)	-0.082	-0.080	-0.006
MAE (mm/tstep)	0.763	0.758	0.031

FIUMICELLO			
Validazione			
Indici di Efficienza	Dati giornalieri		Dati orari (valid. giorn)
	Lumped	Fasce	Lumped
RMSE (mm/tstep)	1.728	1.718	0.086
NSEff (-)	0.779	0.782	0.685
Bias (mm/tstep)	0.044	0.048	-0.004
MAE (mm/tstep)	0.746	0.745	0.032
Calibrazione			
Indici di Efficienza	Dati giornalieri		Dati orari (valid. giorn)
	Lumped	Fasce	Lumped
RMSE (mm/tstep)	1.504	1.505	0.064
NSEff (-)	0.748	0.747	0.735
Bias (mm/tstep)	-0.048	-0.044	-0.008
MAE (mm/tstep)	0.650	0.650	0.029

FIUMICELLO	
Indici di Efficienza	1/5/2009-31/12/2014
	Dati orari
	Lumped
RMSE (mm/tstep)	0.136
NSEff (-)	0.499
Bias (mm/tstep)	-0.003
MAE (mm/tstep)	0.410

Si riportano infine le efficienze relative all'ulteriore calibrazione eseguita sul Fiumicello con i dati orari a partire dal maggio 2009, sono riferite ai parametri utilizzati, riportati a pag. 80.

Fiumicello (calibrazione dati orari 1/5/2009-31/12/2014)	
Indici di Efficienza	Lumped
	Calibrazione
RMSE (mm/tstep)	0.080
NSEff (-)	0.825
Bias (mm/tstep)	-0.001
MAE (mm/tstep)	0.031

Validazione scala oraria Fiumicello 1/1/2013 - 30/6/2013

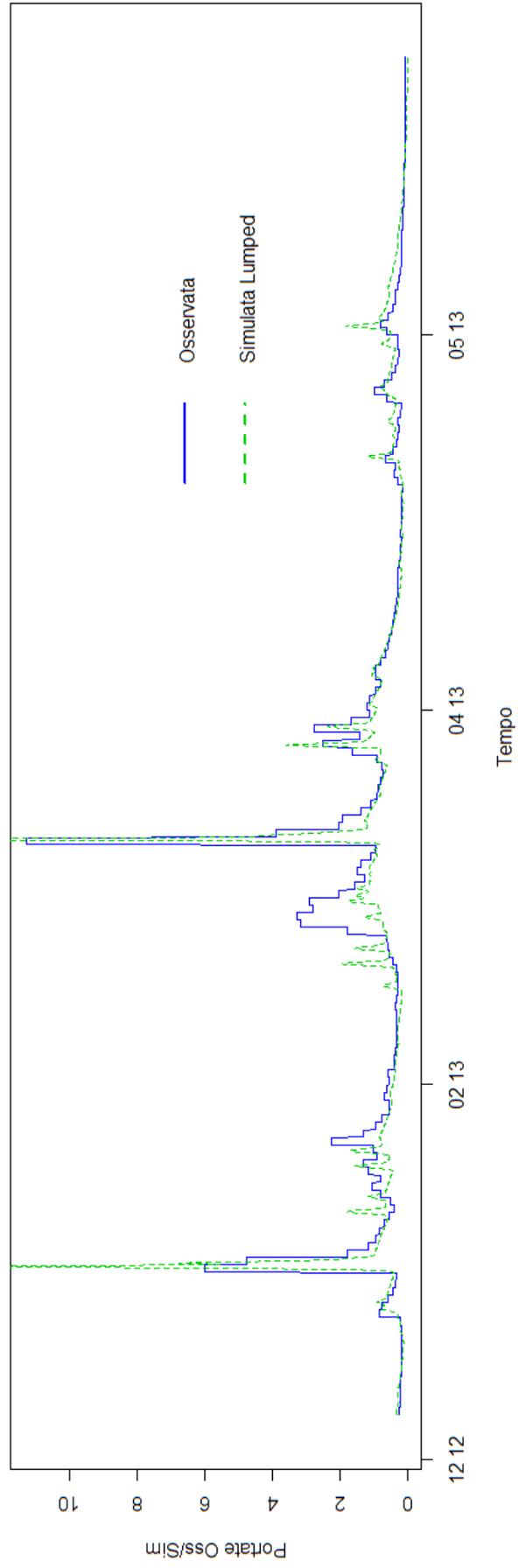


Figura 4.22 – Esempio validazione sulla media giornaliera di portate a scala oraria Fiumicello e paragone fra portate simulata Lumped e a fasce di altitudine

CAPITOLO 5

Regionalizzazione del modello afflussi-deflussi

Nella presentazione del caso di studio, si è spiegato come il bacino del torrente Rabbi sia stato incluso nell'analisi per valutare un eventuale futuro prolungamento della galleria di gronda fino a una sezione del fiume Rabbi, sul ramo che precede la confluenza col Fiumicello (vedasi in Fig.5.1 e 5.2). Fino ad ora si è considerata come sezione di chiusura del bacino quella corrispondente alla posizione della stazione idrometrica omonima, in modo da poter effettuare una normale calibrazione del modello, avendo a disposizione delle osservazioni storiche di portata seppure a

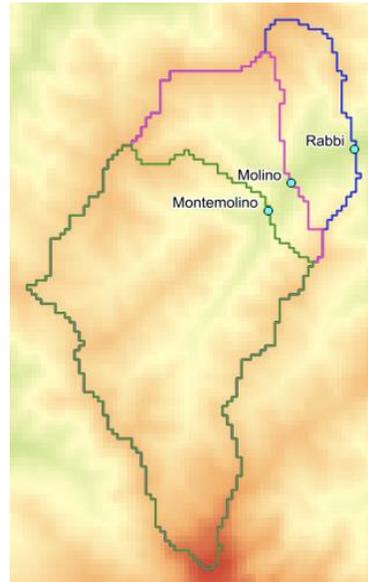


Figura 5.1 – Eventuale collocazione delle sezioni di chiusura dove si ipotizza l'opera di presa

scala giornaliera. Il prolungamento in questione in realtà non avverrebbe su data sezione di chiusura, essendo situata infatti poco a monte del punto di confluenza del torrente con il rio Fiumicello, quindi a quota inferiore dell'opera di presa sullo stesso, ma su una sezione più a monte, che permetta un dislivello sufficiente a trasferire la portata derivata fino all'attuale posizione terminale della gronda. A questo scopo si sono individuati due possibili posizioni dell'opera di presa, di coordinate UTM fuso 32N:

1. 718912E 4869590N
2. 718518E 4869120N

Fra queste due sezioni è presente un molino, che utilizza già parte della portata. La sistemazione dell'opera di presa sulla sezione di valle non andrebbe a creare conflitto con il molino e inoltre si intercetterebbero le

portate provenienti dall'area del Fosso dell'Alpe (in sinistra idrografica); tuttavia la quota per la derivazione potrebbe non essere sufficiente per la derivazione delle portate fino all'allacciamento con il Fiumicello. Per questo si è ipotizzata la seconda sezione più a monte, a quota maggiore; in questo caso si avrebbe però il rischio di creare conflitti con il molino ed inoltre si avrebbe a disposizione minore portata.

I bacini idrografici afferenti alle sezioni in date posizioni verranno d'ora in poi chiamati rispettivamente Valle molino (1) e Monte molino (2).

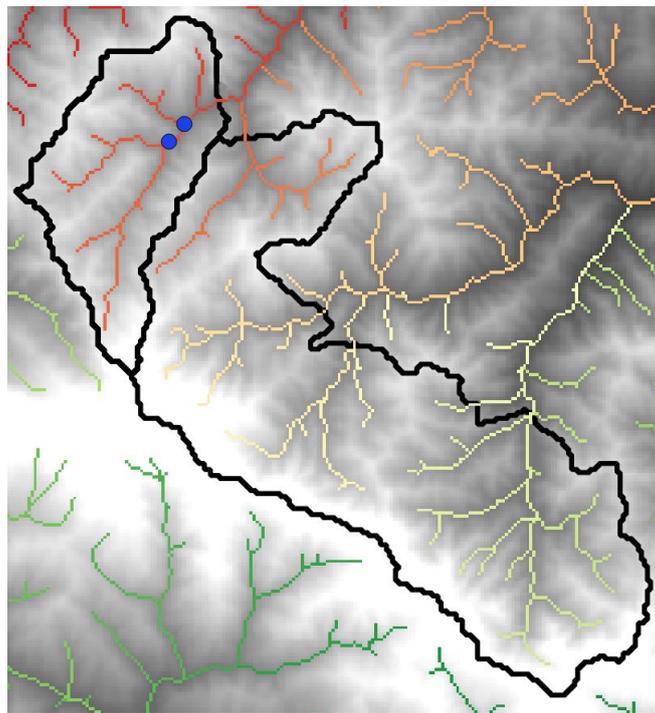


Figura 5.2 – Rete idrografica e posizione delle sezioni Valle molino e Monte molino

Date sezioni non sono però strumentate; essendo impossibile effettuare una calibrazione, si è effettuata un'operazione di regionalizzazione per ricavare la portata defluente dalle stesse.

Per la preparazione dei dati di input si è operato in maniera completamente analoga a quanto eseguito per gli altri bacini su cui si è calibrato il modello (capitolo 2 e 4.3):

- Si è definito il contorno dei bacini (Fig.4.2).
- Si è estratto il DEM relativo alle aree degli stessi.

- Si è diviso ogni bacino in 4 fasce altimetriche e si sono definite aree e perimetri delle stesse (vedi tabella seguente con caratteristiche delle fasce).
- Si sono create le mappe di pioggia e di temperatura con i metodi di interpolazione esposti e con l'utilizzo delle maschere delle varie fasce si sono creati i vettori di input di ogni fascia per entrambi i bacini; tramite le temperature si sono ricavati anche gli input per l'evapotraspirato potenziale.

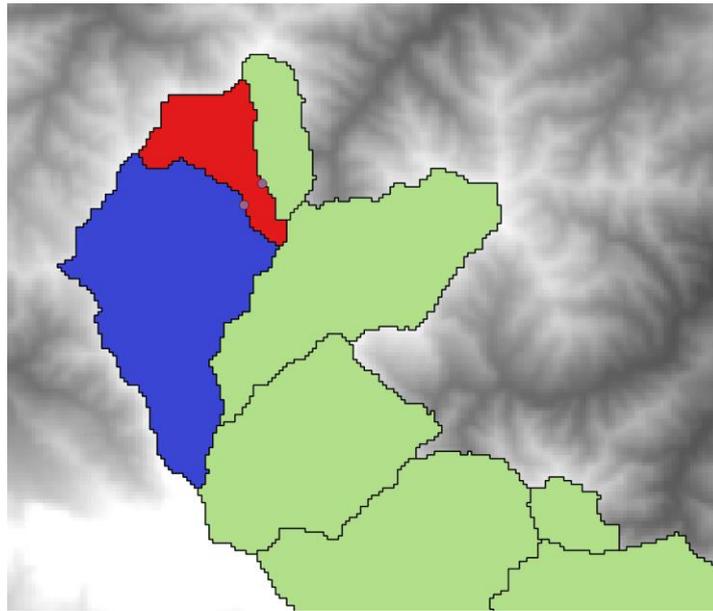


Figura 5.3 – Bacini Valle molino e Monte molino costruiti a partire dalle coordinate delle stazioni di chiusura

Bacino	n° fasce	Intervallo di quote [msl]	Area fascia [kmq]
Valle molino	4	598 – 829	4,64
		829 – 1060	7,93
		1060 – 1291	3,64
		1291 - 1522	2,05
Monte molino	4	638 – 859	4,19
		859 - 1080	6,78
		1080 – 1301	2,20
		1301 – 1522	1,44

Operare una regionalizzazione, significa stimare i parametri del modello afflussi-deflussi per i bacini non strumentati in funzione di quelli di un gruppo di bacini “donatori” su cui il modello è invece stato calibrato. In questo caso i bacini donatori sono rappresentati dai sei bacini oggetto dell’analisi precedente.

Inizialmente in letteratura sono state proposte tecniche di regionalizzazione che stimassero i parametri del modello relativo ai bacini non strumentati ricercando una relazione fra i parametri ottenuti tramite calibrazione sulle sezioni strumentate e le caratteristiche geomorfoclimatiche dei bacini donatori. In funzione delle caratteristiche geomorfoclimatiche era quindi possibile stabilire i valori da assegnare a rispettivi parametri.

Negli ultimi anni, in ragione dei limiti di tali tecniche, che comportano diversi errori dovuti alla dipendenza reciproca dei parametri, si sono affermate metodologie che comportano la modellazione con l’intero set dei parametri di ciascuno dei bacini donatori (Parajka et al., 2005; Oudin et al., 2008; Chiew, 2010): il modello viene applicato tante volte quanti sono i bacini strumentati, utilizzando ogni volta – con gli stessi dati di input relativi al bacino non strumentato – un differente set di parametri (ottenuto appunto dalla calibrazione sui bacini donatori); successivamente la portata regionalizzata finale viene determinata mediando le simulazioni ottenute.

Nel presente elaborato si è perciò eseguita la procedura qui descritta, utilizzata anche da Toth (2010): si è simulato il modello sui due bacini non strumentati (Valle molino e Monte molino) utilizzando i set di parametri calibrati a scala oraria per i bacini donatori, successivamente si sono mediate le portate così ottenute per ricavare le simulazioni regionalizzate. In particolare si è deciso di:

- utilizzare un approccio di tipo concentrato in modo da poter includere nella regionalizzazione anche i bacini del Rabbi e di Fiumicello aventi regime idrologico sicuramente più simile a Molino e Montemolino, tenendo conto anche delle minime differenze ottenute rispetto all’approccio semi-distribuito.

- Escludere dall'insieme dei bacini donatori il bacino diretto, in quanto presenta caratteristiche differenti e dimensioni molto maggiori
- utilizzare i parametri del Fiumicello derivanti dalla calibrazione sui dati orari dal maggio 2009 data la loro maggiore affidabilità, nonostante siano relativi a un diverso intervallo temporale.

In Fig.5.3 e 5.4 viene plottato a titolo d'esempio la simulazione regionalizzata ottenuta, su un intervallo di un mese, insieme alle portate simulate con i set di parametri provenienti dai cinque bacini donatori.

Regionalizzazione Bacino Valle molino 1/3/2013 - 31/3/2013

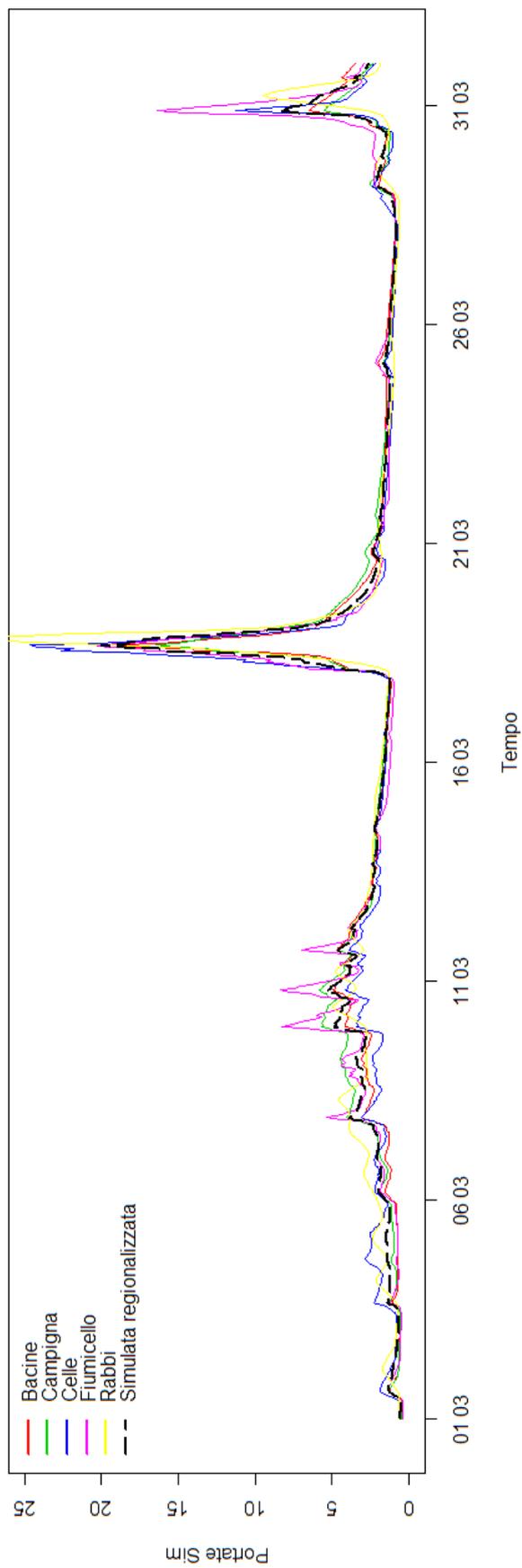


Figura 5.3 – Esempio regionalizzazione bacino Molino relativo al marzo 2013: portate simulate con i set dei parametri provenienti dai bacini donatori e simulazione regionalizzata finale

Regionalizzazione Bacino Monte molino 1/3/2013 - 31/3/2013

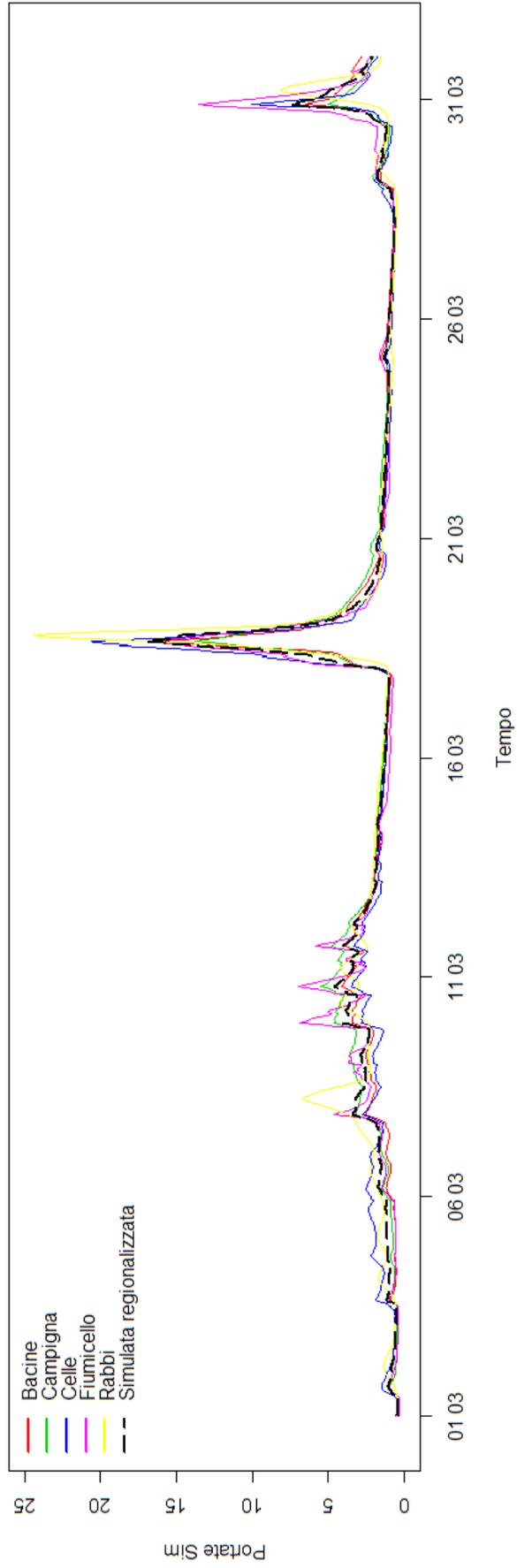


Figura 5.4 – Esempio regionalizzazione bacino Montemolino relativo al marzo 2013: portate simulate con i set dei parametri provenienti dai bacini donatori e simulazione regionalizzata finale

Conclusioni

Con il presente elaborato si è implementato un modello completo per lo studio e la valutazione del regime idrologico dell'area appenninica responsabile dei volumi idrici in ingresso all'invaso di Ridracoli, affrontando sia degli aspetti relativi alla miglior spazializzazione dei dati meteorologici in ingresso al modello (in particolare in relazione alla stima della temperatura, fondamentale per la rappresentazione dei processi di evapotraspirazione e dei fenomeni di accumulo e scioglimento nevoso), sia gli aspetti di calibrazione dei parametri, confrontando le simulazioni ottenute a partire da diverse configurazioni del modello in termini di rappresentazione spaziale e temporale dei fenomeni. Inoltre si è eseguita una regionalizzazione del modello su due sezioni fluviali che sono al momento oggetto di analisi idrologiche, fornendo supporto alla valutazione della possibilità di realizzazione di una nuova opera di presa. A partire dai dati puntuali dei sensori termometrici disponibili si è quindi innanzitutto ricercata la migliore tecnica di interpolazione della temperatura sull'area di studio, ottenendo una discreta precisione, con un errore in procedura di ricampionamento jack-knife raramente superiore al grado centigrado. Utilizzando quindi tale tecnica, basata su una regressione della quota e delle coordinate altimetriche, si sono ottenuti campi spaziali di temperatura e di evapotraspirazione su tutta l'area. Tramite la creazione delle maschere per l'aggregazione spaziale, per fasce altimetriche, dei dati contenuti in tali mappe (ed in quelle che rappresentano il campo di precipitazione, sempre a scala oraria) si sono poi predisposti i dati in ingresso al modello, a scala spaziale sia semi-distribuita che concentrata.

La calibrazione del modello TUWien in approccio semi-distribuito, sia quando applicato a scala oraria sia quando applicato a scala giornaliera, ha portato a buoni risultati. La complessità del modello, dovuta in gran parte alla presenza di uno specifico modulo per la gestione di accumulo e scioglimento nivale, è stata ripagata dalle buone prestazioni delle

simulazioni ottenute, sia sul periodo di calibrazione sia su quello di validazione. Tuttavia, in disaccordo con le aspettative, si è osservato che l'approccio semi-distribuito, non ha portato benefici sostanziali rispetto a uno di tipo totalmente concentrato, soprattutto in relazione al forte aumento di costo computazionale. Un eventuale approfondimento futuro potrebbe concentrarsi sulla modifica di alcune dinamiche interne al modello, riguardanti soprattutto la funzione di trasferimento, per migliorare l'interazione fra i contributi provenienti dalle varie fasce altimetriche, valorizzando la disponibilità di dati in ingresso spazialmente distribuiti. Di pari passo potrebbe essere approfondita maggiormente la ricerca di un set di soglie del dominio dei parametri ammissibili ad hoc per l'area di interesse, in modo da facilitare la calibrazione e aumentare l'affidabilità dell'utilizzo sulla regione appenninica.

I risultati della simulazione a scala oraria sui bacini per cui sono disponibili le sole portate giornaliere a disposizione hanno evidenziato la presenza di oscillazioni non realistiche della simulazione; il problema è stato in parte ridotto con la proposta di una nuova funzione obiettivo, mirata proprio alla riduzione di tali fluttuazioni subgiornaliere.

L'utilizzo di algoritmi multi-obiettivo e la ricerca di un migliore set di soglie per la calibrazione dei parametri potrebbe portare probabilmente a risultati ancor migliori che per di più rafforzerebbero l'affidabilità della stima delle simulazioni regionalizzate in corrispondenza delle due sezioni non strumentate.

Bibliografia

E.Toth, A.Brath, Relazione conclusiva primo anno di attività – Convenzione quinquennale tra Romagna Acque Società delle Fonti Spa e Dipartimento di Ingegneria Civile, Chimica, Ambientale e dei Materiali dell'Università di Bologna, 2015

K.Stahl, R.D.Moore, J.A.Floyer, M.G.Asplin, I.G.McKendry, Comparison of approaches for spatial interpolation of daily air temperature in a large region with complex topography and highly variable station density, 2006

R.Dodson, D.Marks, Daily air temperature interpolated at high spatial resolution over a large mountainous region, 1997

E.Anderson, Calibration on Conceptual Hydrologic Models for Use in River Forecasting (NWSRFS), 2002

D.C.Garen, D.Marks, Spatially distributed energy balance snowmelt modelling in a mountainous river basin: estimation of meteorological inputs and verification of model results, 2005

Laura Lombardi, Calibrazione di modelli idrologici con algoritmi multiobiettivo, 2011

Elena Montosi, Variabilità spaziale dei parametri di modelli afflussi-deflussi, 2012

J.Parajka, R.Merz, G.Blöschl, A comparison of regionalisation methods for catchment model parameters, 2005

Elena Toth, Tecniche di analisi delle serie temporali per il miglioramento delle previsioni di piena in tempo reale, 1999

E.Toth, A.Brath, Modellazione afflussi - deflussi con regionalizzazione dei parametri per i bacini afferenti all'invaso di Ridracoli, 2010

G.E.Liston, K.Elder, A meteorological distribution system for high-resolution terrestrial modeling (MicroMet), 2006

C.J.Willmott, S.M.Roberson, Climatologically aided interpolation (CAI) of terrestrial air temperature, 1995

D.B.Kettel, T.Yao, K.Yang, L.Tian, G.Yang, Temperature lapse rate in complex terrain of the southern slope of central Himalayas, 2013

L. Oudin, V. Andréassian, C. Perrin, C. Michel, Spatial proximity, physical similarity, regression and ungaged catchments: A comparison of regionalization approaches based on 913 French catchments, 2008

Sitografia

Wikipedia – www.wikipedia.org

Romagna Acque | Società delle Fonti – www.romagnacque.it

SIR Toscana – www.sir.toscana.it

ARPA Emilia-Romagna – www.arpa.emr.it

Geoportale regione Emilia-Romagna – www.geoportale.regione.emilia-romagna.it

Materiale informativo Grass Università degli studi di Trento - http://www.ing.unitn.it/~grass/docs/tutorial_62/htdocs/introduzione/intro.html

The Comprehensive R Archive Network – www.cran.r-project.org

RStudio - www.rstudio.com/products/rstudio

Progetto Quantum GIS – www.qgis.org

GRASS project – www.grass.osgeo.org

Appendice 1

Script in R costruiti per la generica interpolazione della temperatura

1. InvD2 Grad06

```
rm(list=ls())

setwd("C:\\Users\\Mattia\\Desktop\\NeriLM_RegionalizzazioneParametri\\R\\Interp_T_t
emporale\\ID2_grad_0.006")

#Lettura dati
T_data=read.table("TempTOT_0.txt", sep="", header=FALSE,
colClasses=rep("numeric",13))
coord_data=read.table("coordinate_staz.txt",
sep="",header=FALSE,colClasses=rep("numeric",4))

#Creazione file
braccina=c(as.numeric(paste(T_data[,5])))
camaldoli=c(as.numeric(paste(T_data[,6])))
campigna=c(as.numeric(paste(T_data[,7])))
campominacci=c(as.numeric(paste(T_data[,8])))
capaccio=c(as.numeric(paste(T_data[,9])))
castellaccio=c(as.numeric(paste(T_data[,10])))
corniolo=c(as.numeric(paste(T_data[,11])))
diga=c(as.numeric(paste(T_data[,12])))
lama=c(as.numeric(paste(T_data[,13])))
X=c(as.numeric(paste(coord_data[,2])))
Y=c(as.numeric(paste(coord_data[,3])))
Ttot=matrix(c(braccina,camaldoli,campigna,campominacci,capaccio,castellaccio,cornio
lo,diga,lama),nrow=nrow(T_data),ncol=nrow(coord_data),byrow=FALSE)
coord=matrix(c(X,Y),nrow=nrow(coord_data),ncol=2,byrow=FALSE)
Z=c(as.numeric(paste(coord_data[,4])))

#Calcolo distanze
n_staz=nrow(coord)
dist=matrix(nrow=n_staz, ncol=n_staz)

for (i in 1:n_staz)
{
  for (j in 1:n_staz)
  {
    dist[i,j]=sqrt((coord[i,1]-coord[j,1])^2+(coord[i,2]-coord[j,2])^2)
  }
}

#Ricavo posizione elementi mancanti
ind=matrix(nrow=nrow(Ttot),ncol=ncol(Ttot))
Ttot_fill=Ttot

for (i in 1:n_staz)
{
  for (j in 1:nrow(Ttot))
  {
    if (Ttot[j,i]<(-90))
    {
      ind[j,i]=j
    }
  }
}
```

```

    }
  }
}

#Interpolazione
for (i in 1:n_staz)
{
  x=na.omit(c(ind[,i]))
  for (j in 1:length(x))
  {
    u=rep(NA,times=n_staz)
    for (k in 1:n_staz)
    {
      if (Ttot[x[j],k]>(-70) & dist[i,k]<15000) #non vengono contate le stazione più
      lontane di 15km
      {
        u[k]=k
      }
    }
    m=na.omit(u)
    d=dist[i,m]
    if (length(m)>0)
    {
      invdsq=(1/d)^2
      p=invdsq/sum(invdsq)
      Ttot_grad=Ttot[x[j],m]-(Z[i]-Z[m])*0.006
      Ttot_fill[x[j],i]=sum(Ttot_grad*p)
    }
  }
}

#Stampa risultato
write.table(Ttot_fill,file="TempTOT_fill")

```

2. InvD2 GradAnnuale

Calcolo Gradiente Annuale:

```
rm(list=ls())
```

```
setwd("C:\\Users\\Mattia\\Desktop\\NeriLM_RegionalizzazioneParametri\\R\\Interp_T_t
emporale\\ID2_grad_annual")
```

#Lettura dati

```
T_dataNA=read.table("TempTOT_0NA.txt", sep="", header=FALSE,
colClasses=rep("numeric",13))
coord_data=read.table("coordinate_staz.txt",
sep="",header=FALSE,colClasses=rep("numeric",4))
```

#Creazione file

```
braccina=c(as.numeric(paste(T_dataNA[,5])))
camaldoli=c(as.numeric(paste(T_dataNA[,6])))
campigna=c(as.numeric(paste(T_dataNA[,7])))
campominacci=c(as.numeric(paste(T_dataNA[,8])))
capaccio=c(as.numeric(paste(T_dataNA[,9])))
castellaccio=c(as.numeric(paste(T_dataNA[,10])))
```

```

corniolo=c(as.numeric(paste(T_dataNA[,11])))
diga=c(as.numeric(paste(T_dataNA[,12])))
lama=c(as.numeric(paste(T_dataNA[,13])))
TtotNA=matrix(c(braccina,camaldoli,campigna,campominacci,capaccio,castellaccio,cornio
lo,diga,lama),nrow=nrow(T_dataNA),ncol=9,byrow=FALSE)
Z=c(as.numeric(paste(coord_data[,4])))

#Calcolo Gradienti mensili
GM=c(NA,NA)

T=c(TtotNA[,1],TtotNA[,2],TtotNA[,3],TtotNA[,4],TtotNA[,5],TtotNA[,6],TtotNA[,7],TtotN
A[,8],TtotNA[,9])
z=c(rep(Z[1],nrow(TtotNA)),rep(Z[2],nrow(TtotNA)),rep(Z[3],nrow(TtotNA)),rep(Z[4],nro
w(TtotNA)),rep(Z[5],nrow(TtotNA)),rep(Z[6],nrow(TtotNA)),rep(Z[7],nrow(TtotNA)),rep(Z
[8],nrow(TtotNA)),rep(Z[9],nrow(TtotNA)))
reg=lm(T~z)
GM[1]=reg$coef[1]
GM[2]=reg$coef[2]

gradM=GM[2]
write.csv((gradM),file="gradM")

Interpolazione:
rm(list=ls())

setwd("C:\\Users\\Mattia\\Desktop\\NeriLM_RegionalizzazioneParametri\\R\\Interp_T_t
emporale\\ID2_grad_annual")

#Lettura dati
T_data=read.table("TempTOT_0.txt", sep=" ", header=FALSE,
colClasses=rep("numeric",13))
coord_data=read.table("coordinate_staz.txt",
sep=" ",header=FALSE,colClasses=rep("numeric",4))

#Creazione file
braccina=c(as.numeric(paste(T_data[,5])))
camaldoli=c(as.numeric(paste(T_data[,6])))
campigna=c(as.numeric(paste(T_data[,7])))
campominacci=c(as.numeric(paste(T_data[,8])))
capaccio=c(as.numeric(paste(T_data[,9])))
castellaccio=c(as.numeric(paste(T_data[,10])))
corniolo=c(as.numeric(paste(T_data[,11])))
diga=c(as.numeric(paste(T_data[,12])))
lama=c(as.numeric(paste(T_data[,13])))
X=c(as.numeric(paste(coord_data[,2])))
Y=c(as.numeric(paste(coord_data[,3])))
Ttot=matrix(c(braccina,camaldoli,campigna,campominacci,capaccio,castellaccio,cornio
lo,diga,lama),nrow=nrow(T_data),ncol=nrow(coord_data),byrow=FALSE)
coord=matrix(c(X,Y),nrow=nrow(coord_data),ncol=2,byrow=FALSE)
Z=c(as.numeric(paste(coord_data[,4])))

#Calcolo distanze
n_staz=nrow(coord)
dist=matrix(nrow=n_staz, ncol=n_staz)

for (i in 1:n_staz)
{
  for (j in 1:n_staz)

```

```

{
  dist[i,j]=sqrt((coord[i,1]-coord[j,1])^2+(coord[i,2]-coord[j,2])^2)
}
}

#Ricavo posizione elementi mancanti
ind=matrix(nrow=nrow(Ttot),ncol=ncol(Ttot))
Ttot_fill=Ttot

for (i in 1:n_staz)
{
  for (j in 1:nrow(Ttot))
  {
    if (Ttot[j,i]<(-90))
    {
      ind[j,i]=j
    }
  }
}

#Interpolazione
for (i in 1:n_staz)
{
  x=na.omit(c(ind[,i]))
  for (j in 1:length(x))
  {
    u=rep(NA,times=n_staz)
    for (k in 1:n_staz)
    {
      if (Ttot[x[j],k]>(-70) & dist[i,k]<15000) #non vengono contate le stazione più
      lontane di 15km
      {
        u[k]=k
      }
    }
    m=na.omit(u)
    d=dist[i,m]
    if (length(m)>0)
    {
      invdsq=(1/d)^2
      p=invdsq/sum(invdsq)
      Ttot_grad=Ttot[x[j],m]-(Z[i]-Z[m])*0.0039
      Ttot_fill[x[j],i]=sum(Ttot_grad*p)
    }
  }
}

#Stampa risultato
write.table(Ttot_fill,file="TempTOT_fillGradAnnual")

```

3. *InvD2 GradMonthly*

Calcolo Gradienti Mensili

```
rm(list=ls())
```

```
setwd("C:\\Users\\Mattia\\Desktop\\NeriLM_RegionalizzazioneParametri\\R\\Interp_T_t  
temporale\\ID2_grad_monthly")
```

```
#Lettura dati
```

```
T_dataNA=read.table("TempTOT_0NA.txt", sep="", header=FALSE,  
colClasses=rep("numeric",13))
```

```
coord_data=read.table("coordinate_staz.txt",  
sep="",header=FALSE,colClasses=rep("numeric",4))
```

```
#Creazione file
```

```
mesi=c(as.numeric(paste(T_dataNA[,2])))
```

```
braccina=c(as.numeric(paste(T_dataNA[,5])))
```

```
camaldoli=c(as.numeric(paste(T_dataNA[,6])))
```

```
campigna=c(as.numeric(paste(T_dataNA[,7])))
```

```
campominacci=c(as.numeric(paste(T_dataNA[,8])))
```

```
capaccio=c(as.numeric(paste(T_dataNA[,9])))
```

```
castellaccio=c(as.numeric(paste(T_dataNA[,10])))
```

```
corniolo=c(as.numeric(paste(T_dataNA[,11])))
```

```
diga=c(as.numeric(paste(T_dataNA[,12])))
```

```
lama=c(as.numeric(paste(T_dataNA[,13])))
```

```
TtotNA=matrix(c(mesi,braccina,camaldoli,campigna,campominacci,capaccio,castellacc  
io,corniolo,diga,lama),nrow=nrow(T_dataNA),ncol=10,byrow=FALSE)
```

```
Z=c(as.numeric(paste(coord_data[,4])))
```

```
#Calcolo Gradienti mensili
```

```
GM=matrix(nrow=12,ncol=2)
```

```
for (i in 1:12)
```

```
{
```

```
  Ttotg=TtotNA[TtotNA[,1]==i,]
```

```
T=c(Ttotg[,2],Ttotg[,3],Ttotg[,4],Ttotg[,5],Ttotg[,6],Ttotg[,7],Ttotg[,8],Ttotg[,9],Ttotg[,10])
```

```
z=c(rep(Z[1],nrow(Ttotg)),rep(Z[2],nrow(Ttotg)),rep(Z[3],nrow(Ttotg)),rep(Z[4],nrow(Ttot  
g)),rep(Z[5],nrow(Ttotg)),rep(Z[6],nrow(Ttotg)),rep(Z[7],nrow(Ttotg)),rep(Z[8],nrow(Ttot  
g)),rep(Z[9],nrow(Ttotg)))
```

```
  reg=lm(T~z)
```

```
  GM[i,1]=reg$coef[1]
```

```
  GM[i,2]=reg$coef[2]
```

```
}
```

```
gradM=c(GM[,2])
```

```
write.csv((gradM),file="gradM")
```

```
Interpolazione:
```

```
rm(list=ls())
```

```
setwd("C:\\Users\\Mattia\\Dropbox\\IO\\Tesi\\NeriLM_Tesi\\R\\Interp_T_temporale\\ID2  
_grad_monthly")
```

```
#Lettura dati
```

```
T_data=read.table("TempTOT_0.txt", sep="", header=FALSE,  
colClasses=rep("numeric",13))
```

```
coord_data=read.table("coordinate_staz.txt",  
sep="",header=FALSE,colClasses=rep("numeric",4))
```

```

#Creazione file
mesi=c(as.numeric(paste(T_data[,2])))
braccina=c(as.numeric(paste(T_data[,5])))
camaldoli=c(as.numeric(paste(T_data[,6])))
campigna=c(as.numeric(paste(T_data[,7])))
campominacci=c(as.numeric(paste(T_data[,8])))
capaccio=c(as.numeric(paste(T_data[,9])))
castellaccio=c(as.numeric(paste(T_data[,10])))
corniolo=c(as.numeric(paste(T_data[,11])))
diga=c(as.numeric(paste(T_data[,12])))
lama=c(as.numeric(paste(T_data[,13])))
X=c(as.numeric(paste(coord_data[,2])))
Y=c(as.numeric(paste(coord_data[,3])))
Ttot=matrix(c(braccina,camaldoli,campigna,campominacci,capaccio,castellaccio,corniolo,diga,lama),nrow=nrow(T_data),ncol=9,byrow=FALSE)
coord=matrix(c(X,Y),nrow=nrow(coord_data),ncol=2,byrow=FALSE)
Z=c(as.numeric(paste(coord_data[,4])))

#Calcolo distanze e differenze di quote
n_staz=nrow(coord)
dist=matrix(nrow=n_staz, ncol=n_staz)
dZ=matrix(nrow=n_staz, ncol=n_staz)

for (i in 1:n_staz)
{
  for (j in 1:n_staz)
  {
    dist[i,j]=sqrt((coord[i,1]-coord[j,1])^2+(coord[i,2]-coord[j,2])^2)
  }
}

#Ricavo posizione elementi mancanti
ind=matrix(nrow=nrow(Ttot),ncol=ncol(Ttot))
Ttot_fillM=Ttot

for (i in 1:n_staz)
{
  for (j in 1:nrow(Ttot))
  {
    if (Ttot[j,i]<(-90))
    {
      ind[j,i]=j
    }
  }
}

#Gradienti mensili
GM=c(-0.0025,-0.0038,-0.0046,-0.0046,-0.0046,-0.0043,-0.0042,-0.0038,-0.0039,-0.0037,-0.0036,-0.0025)

#Interpolazione
for (i in 1:n_staz)
{
  x=na.omit(c(ind[,i]))
  for (j in 1:length(x))
  {
    u=rep(NA,times=n_staz)
    for (k in 1:n_staz)

```

```

    {
      if (Ttot[x[j],k]>(-70) & dist[i,k]<15000) #non vengono contate le stazione più
      lontane di 15km
      {
        u[k]=k
      }
    }
    m=na.omit(u)
    d=dist[i,m]
    if (length(m)>0)
    {
      invdsq=(1/d)^2
      p=invdsq/sum(invdsq)
      Ttot_grad=Ttot[x[j],m]+(Z[i]-Z[m])*GM[mesi[x[j]]]
      Ttot_fillM[x[j],i]=sum(Ttot_grad*p)
    }
  }
}

#Stampa risultato
write.table(Ttot_fillM,file="TempTOT_fillMonthlyGrad")

```

4. InvD2+EI GradMonthly

```

rm(list=ls())

setwd("C:\\Users\\Mattia\\Dropbox\\IO\\Tesi\\NeriLM_Tesi\\R\\Interp_T_temporale\\ID2
EL_grad_monthly")

#Lettura dati
T_data=read.table("TempTOT_0.txt",          sep=" ",          header=FALSE,
colClasses=rep("numeric",13))
coord_data=read.table("coordinate_staz.txt",
sep=" ",header=FALSE,colClasses=rep("numeric",4))

#Creazione file
mesi=c(as.numeric(paste(T_data[,2])))
braccina=c(as.numeric(paste(T_data[,5])))
camaldoli=c(as.numeric(paste(T_data[,6])))
campigna=c(as.numeric(paste(T_data[,7])))
campominacci=c(as.numeric(paste(T_data[,8])))
capaccio=c(as.numeric(paste(T_data[,9])))
castellaccio=c(as.numeric(paste(T_data[,10])))
corniolo=c(as.numeric(paste(T_data[,11])))
diga=c(as.numeric(paste(T_data[,12])))
lama=c(as.numeric(paste(T_data[,13])))
X=c(as.numeric(paste(coord_data[,2])))
Y=c(as.numeric(paste(coord_data[,3])))
Ttot=matrix(c(braccina,camaldoli,campigna,campominacci,capaccio,castellaccio,corniolo,diga,lama),nrow=nrow(T_data),ncol=9,byrow=FALSE)
coord=matrix(c(X,Y),nrow=nrow(coord_data),ncol=2,byrow=FALSE)
Z=c(as.numeric(paste(coord_data[,4])))

#Calcolo distanze e differenze di quote
n_staz=nrow(coord)

```

```

dist=matrix(nrow=n_staz, ncol=n_staz)
dZ=matrix(nrow=n_staz, ncol=n_staz)

for (i in 1:n_staz)
{
  for (j in 1:n_staz)
  {
    dist[i,j]=sqrt((coord[i,1]-coord[j,1])^2+(coord[i,2]-coord[j,2])^2)
    dZ[i,j]=Z[i]-Z[j]
  }
}

#Ricavo posizione elementi mancanti
ind=matrix(nrow=nrow(Ttot),ncol=ncol(Ttot))
Ttot_fillM=Ttot

for (i in 1:n_staz)
{
  for (j in 1:nrow(Ttot))
  {
    if (Ttot[j,i]<(-90))
    {
      ind[j,i]=j
    }
  }
}

#Gradienti mensili
GM=c(-0.0025,-0.0038,-0.0046,-0.0046,-0.0046,-0.0043,-0.0042,-0.0038,-0.0039,-
0.0037,-0.0036,-0.0025)

#Interpolazione
Fe=100
for (i in 1:n_staz)
{
  x=na.omit(c(ind[,i]))
  for (j in 1:length(x))
  {
    u=rep(NA,times=n_staz)
    for (k in 1:n_staz)
    {
      if (Ttot[x[j],k]>(-70) & dist[i,k]<15000) #non vengono contate le stazione più lontane
di 15km
      {
        u[k]=k
      }
    }
    m=na.omit(u)
    d=dist[i,m]
    dz=dZ[i,m]
    if (length(m)>0)
    {
      invdsq=1/(d^2+abs(Fe*dz))
      p=invdsq/sum(invdsq)
      Ttot_grad=Ttot[x[j],m]+(Z[i]-Z[m])*GM[mesi[x[j]]]
      Ttot_fillM[x[j],i]=sum(Ttot_grad*p)
    }
  }
}

```

```

}

#Stampa risultato
write.table(Ttot_fillM,file="TempTOT_fillMonthlyGradID2EL")

```

5. Linear Regression

```

rm(list=ls())

setwd("C:\\Users\\Mattia\\Desktop\\NeriLM_RegionalizzazioneParametri\\R\\Interp_T_t
emporale\\Reg_hourly")

#Lettura dati
T_data=read.table("TempTOT_0.txt",          sep="",          header=FALSE,
colClasses=rep("numeric",13))
T_dataNA=read.table("TempTOT_0NA.txt",      sep="",          header=FALSE,
colClasses=rep("numeric",13))
coord_data=read.table("coordinate_staz.txt",
sep="",header=FALSE,colClasses=rep("numeric",4))

#Creazione file
braccina=c(as.numeric(paste(T_data[,5])))
camaldoli=c(as.numeric(paste(T_data[,6])))
campigna=c(as.numeric(paste(T_data[,7])))
campominacci=c(as.numeric(paste(T_data[,8])))
capaccio=c(as.numeric(paste(T_data[,9])))
castellaccio=c(as.numeric(paste(T_data[,10])))
corniolo=c(as.numeric(paste(T_data[,11])))
diga=c(as.numeric(paste(T_data[,12])))
lama=c(as.numeric(paste(T_data[,13])))
X=c(as.numeric(paste(coord_data[,2])))
Y=c(as.numeric(paste(coord_data[,3])))
Ttot=matrix(c(braccina,camaldoli,campigna,campominacci,capaccio,castellaccio,cornio
lo,diga,lama),nrow=nrow(T_data),ncol=9,byrow=FALSE)
coord=matrix(c(X,Y),nrow=nrow(coord_data),ncol=2,byrow=FALSE)
Z=c(as.numeric(paste(coord_data[,4])))

braccina=c(as.numeric(paste(T_dataNA[,5])))
camaldoli=c(as.numeric(paste(T_dataNA[,6])))
campigna=c(as.numeric(paste(T_dataNA[,7])))
campominacci=c(as.numeric(paste(T_dataNA[,8])))
capaccio=c(as.numeric(paste(T_dataNA[,9])))
castellaccio=c(as.numeric(paste(T_dataNA[,10])))
corniolo=c(as.numeric(paste(T_dataNA[,11])))
diga=c(as.numeric(paste(T_dataNA[,12])))
lama=c(as.numeric(paste(T_dataNA[,13])))
TtotNA=matrix(c(braccina,camaldoli,campigna,campominacci,capaccio,castellaccio,cor
niolo,diga,lama),nrow=nrow(T_dataNA),ncol=9,byrow=FALSE)

n_staz=nrow(coord)

#Ricavo posizione elementi mancanti
ind=matrix(nrow=nrow(Ttot),ncol=ncol(Ttot))
Ttot_fillH=Ttot

```

```

for (i in 1:n_staz)
{
  for (j in 1:nrow(Ttot))
  {
    if (Ttot[j,i]<(-90))
    {
      ind[j,i]=j
    }
  }
}

#Gradienti orari
GH=matrix(rep(-999,2*nrow(Ttot)),nrow=nrow(Ttot),ncol=2)
for (i in 1:nrow(Ttot))
{
  T=c(TtotNA[i,])
  t=na.omit(T)
  if (length(t)>3)
  {
    reg=lm(T~Z)
    GH[i,1]=reg$coef[1]
    GH[i,2]=reg$coef[2]
  }
}
for (i in 2:nrow(Ttot))
{
  if (GH[i,1]==(-999))
  {
    GH[i,1]=GH[i-1,1]
    GH[i,2]=GH[i-1,2]
  }
}

#Interpolazione
for (i in 1:n_staz)
{
  x=na.omit(c(ind[,i]))
  for (j in 1:length(x))
  {
    Ttot_fillH[x[j],i]=GH[x[j],1]+Z[i]*GH[x[j],2]
  }
}

#Stampa risultato
write.table(Ttot_fillH,file="TempTOT_fillRegHourly")
write.table(GH,file="GH")

```

6. Multiple Linear Regression

```

rm(list=ls())

setwd("C:\\Users\\Mattia\\Dropbox\\IO\\Tesi\\NeriLM_Tesi\\R\\Interp_T_temporale\\MultipleReg_hourly")

#Lettura dati

```

```

T_data=read.table("TempTOT_0.txt",          sep="",          header=FALSE,
colClasses=rep("numeric",13))
T_dataNA=read.table("TempTOT_0NA.txt",      sep="",          header=FALSE,
colClasses=rep("numeric",13))
coord_data=read.table("coordinate_staz.txt",
sep="",header=FALSE,colClasses=rep("numeric",4))

#Creazione file
braccina=c(as.numeric(paste(T_data[,5])))
camaldoli=c(as.numeric(paste(T_data[,6])))
campigna=c(as.numeric(paste(T_data[,7])))
campominacci=c(as.numeric(paste(T_data[,8])))
capaccio=c(as.numeric(paste(T_data[,9])))
castellaccio=c(as.numeric(paste(T_data[,10])))
corniolo=c(as.numeric(paste(T_data[,11])))
diga=c(as.numeric(paste(T_data[,12])))
lama=c(as.numeric(paste(T_data[,13])))
X=c(as.numeric(paste(coord_data[,2])))
Y=c(as.numeric(paste(coord_data[,3])))
Ttot=matrix(c(braccina,camaldoli,campigna,campominacci,capaccio,castellaccio,corniolo,diga,lama),nrow=nrow(T_data),ncol=9,byrow=FALSE)
coord=matrix(c(X,Y),nrow=nrow(coord_data),ncol=2,byrow=FALSE)
Z=c(as.numeric(paste(coord_data[,4])))

braccina=c(as.numeric(paste(T_dataNA[,5])))
camaldoli=c(as.numeric(paste(T_dataNA[,6])))
campigna=c(as.numeric(paste(T_dataNA[,7])))
campominacci=c(as.numeric(paste(T_dataNA[,8])))
capaccio=c(as.numeric(paste(T_dataNA[,9])))
castellaccio=c(as.numeric(paste(T_dataNA[,10])))
corniolo=c(as.numeric(paste(T_dataNA[,11])))
diga=c(as.numeric(paste(T_dataNA[,12])))
lama=c(as.numeric(paste(T_dataNA[,13])))
TtotNA=matrix(c(braccina,camaldoli,campigna,campominacci,capaccio,castellaccio,corniolo,diga,lama),nrow=nrow(T_dataNA),ncol=9,byrow=FALSE)

n_staz=nrow(coord)

#Ricavo posizione elementi mancanti
ind=matrix(nrow=nrow(Ttot),ncol=ncol(Ttot))
Ttot_fillH=Ttot

for (i in 1:n_staz)
{
  for (j in 1:nrow(Ttot))
  {
    if (Ttot[j,i]<(-90))
    {
      ind[j,i]=j
    }
  }
}

#Coefficienti della regressione multipla oraria
GH=matrix(rep(NA,4*nrow(Ttot)),nrow=nrow(Ttot),ncol=4)
for (i in 1:nrow(Ttot))
{
  T=c(TtotNA[i,])

```

```

t=na.omit(T)
if (length(t)>3)
{
  reg=lm(T~Z+X+Y)
  GH[i,1]=reg$coef[1]
  GH[i,2]=reg$coef[2]
  GH[i,3]=reg$coef[3]
  GH[i,4]=reg$coef[4]
}
}
write.table(GH,file="GHMna",row.names=FALSE,col.names=FALSE)
#Interpolazione
for (i in 1:n_staz)
{
  x=na.omit(c(ind[,i]))
  for (j in 1:length(x))
  {
    Ttot_fillH[x[j],i]=GH[x[j],1]+Z[i]*GH[x[j],2]+X[i]*GH[x[j],3]+Y[i]*GH[x[j],4]
  }
}

#Stampa risultato
write.table(Ttot_fillH,file="TempTOT_fillMRegHourly_GHMna",row.names=FALSE,col.
names=FALSE)
#write.table(GH,file="GHM")

```

Script in R costruiti per ricampionamento Jack-Knife della temperatura (esempio per stazione di Braccina)

1. InvD2 Grad06

```

rm(list=ls())

setwd("C:\\Users\\Mattia\\Desktop\\NeriLM_RegionalizzazioneParametri\\R\\Interp_T_t
emporale\\ID2_grad_0.006")

#Lettura dati
T_data=read.table("TempTOT_0.txt", sep="", header=FALSE,
colClasses=rep("numeric",13))
coord_data=read.table("coordinate_staz.txt",
sep="",header=FALSE,colClasses=rep("numeric",4))

#Creazione file
braccina=c(as.numeric(paste(T_data[,5])))
camaldoli=c(as.numeric(paste(T_data[,6])))
campigna=c(as.numeric(paste(T_data[,7])))
campominacci=c(as.numeric(paste(T_data[,8])))
capaccio=c(as.numeric(paste(T_data[,9])))
castellaccio=c(as.numeric(paste(T_data[,10])))
corniolo=c(as.numeric(paste(T_data[,11])))
diga=c(as.numeric(paste(T_data[,12])))
lama=c(as.numeric(paste(T_data[,13])))
X=c(as.numeric(paste(coord_data[,2])))
Y=c(as.numeric(paste(coord_data[,3])))
Ttot=matrix(c(braccina,camaldoli,campigna,campominacci,capaccio,castellaccio,corniolo,diga,lama),nrow=nrow(T_data),ncol=9,byrow=FALSE)

```

```

coord=matrix(c(X,Y),nrow=nrow(coord_data),ncol=2,byrow=FALSE)
Z=c(as.numeric(paste(coord_data[,4])))

#Calcolo distanze e differenze di quote
n_staz=nrow(coord)
dist=matrix(nrow=n_staz, ncol=n_staz)
dZ=matrix(nrow=n_staz, ncol=n_staz)

for (i in 1:n_staz)
{
  for (j in 1:n_staz)
  {
    dist[i,j]=sqrt((coord[i,1]-coord[j,1])^2+(coord[i,2]-coord[j,2])^2)
  }
}

JK=rep(NA,nrow(Ttot))

x=1:nrow(Ttot)
for (j in 1:length(x))
{
  u=rep(NA,times=n_staz)
  h=c(2:9)
  for (k in h)
  {
    if (Ttot[x[j],k]>(-70) & dist[1,k]<15000) #non vengono contate le stazione più
    lontane di 15km
    {
      u[k]=k
    }
  }
  m=na.omit(u)
  d=dist[1,m]
  if (length(m)>0)
  {
    invdsq=(1/d)^2
    p=invdsq/sum(invdsq)
    Ttot_grad=Ttot[x[j],m]-(Z[1]-Z[m])*0.006
    JK[j]=sum(Ttot_grad*p)
  }
}

#Stampa risultato
write.table(JK,file="JK")

```

2. *InvD2 GrandAnnual*

```

rm(list=ls())

setwd("C:\\Users\\Mattia\\Desktop\\NeriLM_RegionalizzazioneParametri\\R\\Interp_T_t
emporale\\ID2_grad_annual")

#Lettura dati

```

```

T_data=read.table("TempTOT_0.txt",          sep="",          header=FALSE,
colClasses=rep("numeric",13))
coord_data=read.table("coordinate_staz.txt",
sep="",header=FALSE,colClasses=rep("numeric",4))

#Creazione file
braccina=c(as.numeric(paste(T_data[,5])))
camaldoli=c(as.numeric(paste(T_data[,6])))
campigna=c(as.numeric(paste(T_data[,7])))
campominacci=c(as.numeric(paste(T_data[,8])))
capaccio=c(as.numeric(paste(T_data[,9])))
castellaccio=c(as.numeric(paste(T_data[,10])))
corniolo=c(as.numeric(paste(T_data[,11])))
diga=c(as.numeric(paste(T_data[,12])))
lama=c(as.numeric(paste(T_data[,13])))
X=c(as.numeric(paste(coord_data[,2])))
Y=c(as.numeric(paste(coord_data[,3])))
Ttot=matrix(c(braccina,camaldoli,campigna,campominacci,capaccio,castellaccio,corniolo,diga,lama),nrow=nrow(T_data),ncol=9,byrow=FALSE)
coord=matrix(c(X,Y),nrow=nrow(coord_data),ncol=2,byrow=FALSE)
Z=c(as.numeric(paste(coord_data[,4])))

#Calcolo distanze e differenze di quote
n_staz=nrow(coord)
dist=matrix(nrow=n_staz, ncol=n_staz)
dZ=matrix(nrow=n_staz, ncol=n_staz)

for (i in 1:n_staz)
{
  for (j in 1:n_staz)
  {
    dist[i,j]=sqrt((coord[i,1]-coord[j,1])^2+(coord[i,2]-coord[j,2])^2)
  }
}

JK=rep(NA,nrow(Ttot))

x=1:nrow(Ttot)
for (j in 1:length(x))
{
  u=rep(NA,times=n_staz)
  h=c(2:9)
  for (k in h)
  {
    if (Ttot[x[j],k]>(-70) & dist[1,k]<15000) #non vengono contate le stazione più lontane di 15km
    {
      u[k]=k
    }
  }
  m=na.omit(u)
  d=dist[1,m]
  if (length(m)>0)
  {
    invdsq=(1/d)^2
    p=invdsq/sum(invdsq)
    Ttot_grad=Ttot[x[j],m]-(Z[1]-Z[m])*0.0039
    JK[j]=sum(Ttot_grad*p)
  }
}

```

```
}  
}
```

```
#Stampa risultato  
write.table(JK,file="JK")
```

3. InvD2 GradMonthly

```
rm(list=ls())
```

```
setwd("C:\\Users\\Mattia\\Desktop\\NeriLM_RegionalizzazioneParametri\\R\\Interp_T_t  
emporale\\ID2_grad_monthly")
```

```
#Lettura dati
```

```
T_data=read.table("TempTOT_0.txt", sep="", header=FALSE,  
colClasses=rep("numeric",13))  
coord_data=read.table("coordinate_staz.txt",  
sep="",header=FALSE,colClasses=rep("numeric",4))
```

```
#Creazione file
```

```
mesi=c(as.numeric(paste(T_data[,2])))  
braccina=c(as.numeric(paste(T_data[,5])))  
camaldoli=c(as.numeric(paste(T_data[,6])))  
campigna=c(as.numeric(paste(T_data[,7])))  
campominacci=c(as.numeric(paste(T_data[,8])))  
capaccio=c(as.numeric(paste(T_data[,9])))  
castellaccio=c(as.numeric(paste(T_data[,10])))  
corniolo=c(as.numeric(paste(T_data[,11])))  
diga=c(as.numeric(paste(T_data[,12])))  
lama=c(as.numeric(paste(T_data[,13])))  
X=c(as.numeric(paste(coord_data[,2])))  
Y=c(as.numeric(paste(coord_data[,3])))  
Ttot=matrix(c(braccina,camaldoli,campigna,campominacci,capaccio,castellaccio,cornio  
lo,diga,lama),nrow=nrow(T_data),ncol=9,byrow=FALSE)  
coord=matrix(c(X,Y),nrow=nrow(coord_data),ncol=2,byrow=FALSE)  
Z=c(as.numeric(paste(coord_data[,4])))
```

```
#Calcolo distanze e differenze di quote
```

```
n_staz=nrow(coord)  
dist=matrix(nrow=n_staz, ncol=n_staz)  
dZ=matrix(nrow=n_staz, ncol=n_staz)
```

```
for (i in 1:n_staz)  
{  
  for (j in 1:n_staz)  
  {  
    dist[i,j]=sqrt((coord[i,1]-coord[j,1])^2+(coord[i,2]-coord[j,2])^2)  
  }  
}
```

```
JK=rep(NA,nrow(Ttot))
```

```
#Gradienti mensili
```

```
GM=c(-0.0025,-0.0038,-0.0046,-0.0046,-0.0046,-0.0043,-0.0042,-0.0038,-0.0039,-  
0.0037,-0.0036,-0.0025)
```

```

x=1:nrow(Ttot)
for (j in 1:length(x))
{
  u=rep(NA, times=n_staz)
  h=c(2:9)
  for (k in h)
  {
    if (Ttot[x[j],k]>(-70) & dist[1,k]<15000) #non vengono contate le stazione più lontane
    di 15km
    {
      u[k]=k
    }
  }
  m=na.omit(u)
  d=dist[1,m]
  if (length(m)>0)
  {
    invdsq=(1/d)^2
    p=invdsq/sum(invdsq)
    Ttot_grad=Ttot[x[j],m]+(Z[1]-Z[m])*GM[mesi[x[j]]]
    JK[j]=sum(Ttot_grad*p)
  }
}

#Stampa risultato
write.table(JK, file="JK")

```

4. InvD2+EI GradMonthly

```

rm(list=ls())

setwd("C:\\Users\\Mattia\\Dropbox\\IO\\Tesi\\NeriLM_Tesi\\R\\Interp_T_temporale\\ID2
EL_grad_monthly")

#Lettura dati
T_data=read.table("TempTOT_0.txt", sep=" ", header=FALSE,
colClasses=rep("numeric",13))
coord_data=read.table("coordinate_staz.txt",
sep=" ", header=FALSE, colClasses=rep("numeric",4))

#Creazione file
mesi=c(as.numeric(paste(T_data[,2])))
braccina=c(as.numeric(paste(T_data[,5])))
camaldoli=c(as.numeric(paste(T_data[,6])))
campigna=c(as.numeric(paste(T_data[,7])))
campominacci=c(as.numeric(paste(T_data[,8])))
capaccio=c(as.numeric(paste(T_data[,9])))
castellaccio=c(as.numeric(paste(T_data[,10])))
corniolo=c(as.numeric(paste(T_data[,11])))
diga=c(as.numeric(paste(T_data[,12])))
lama=c(as.numeric(paste(T_data[,13])))
X=c(as.numeric(paste(coord_data[,2])))
Y=c(as.numeric(paste(coord_data[,3])))

```

```

Ttot=matrix(c(braccina,camaldoli,campigna,campominacci,capaccio,castellaccio,cornio
lo,diga,lama),nrow=nrow(T_data),ncol=9,byrow=FALSE)
coord=matrix(c(X,Y),nrow=nrow(coord_data),ncol=2,byrow=FALSE)
Z=c(as.numeric(paste(coord_data[,4])))

#Calcolo distanze e differenze di quote
n_staz=nrow(coord)
dist=matrix(nrow=n_staz, ncol=n_staz)
dZ=matrix(nrow=n_staz, ncol=n_staz)

for (i in 1:n_staz)
{
  for (j in 1:n_staz)
  {
    dist[i,j]=sqrt((coord[i,1]-coord[j,1])^2+(coord[i,2]-coord[j,2])^2)
    dZ[i,j]=Z[i]-Z[j]
  }
}

JK=rep(NA,nrow(Ttot))

#Gradienti mensili
GM=c(-0.0025,-0.0038,-0.0046,-0.0046,-0.0046,-0.0043,-0.0042,-0.0038,-0.0039,-
0.0037,-0.0036,-0.0025)
Fe=100

x=1:nrow(Ttot)
for (j in 1:length(x))
{
  u=rep(NA, times=n_staz)
  h=c(2:9)
  for (k in h)
  {
    if (Ttot[x[j],k]>(-70) & dist[1,k]<15000) #non vengono contate le stazione più lontane
di 15km
    {
      u[k]=k
    }
  }
  m=na.omit(u)
  d=dist[1,m]
  dz=dZ[1,m]
  if (length(m)>0)
  {
    invdsq=1/(d^2+abs(Fe*dz))
    p=invdsq/sum(invdsq)
    Ttot_grad=Ttot[x[j],m]+(Z[1]-Z[m])*GM[mesi[x[j]]]
    JK[j]=sum(Ttot_grad*p)
  }
}

#Stampa risultato
write.table(JK,file="JK")

```

5. Linear Regression

```
rm(list=ls())

setwd("C:\\Users\\Mattia\\Desktop\\NeriLM_RegionalizzazioneParametri\\R\\Interp_T_t
emporale\\Reg_hourly")

#Lettura dati
T_data=read.table("TempTOT_0.txt",          sep="",          header=FALSE,
colClasses=rep("numeric",13))
T_dataNA=read.table("TempTOT_0NA.txt",      sep="",          header=FALSE,
colClasses=rep("numeric",13))
coord_data=read.table("coordinate_staz.txt",
sep="",header=FALSE,colClasses=rep("numeric",4))

#Creazione file
braccina=c(as.numeric(paste(T_data[,5])))
camaldoli=c(as.numeric(paste(T_data[,6])))
campigna=c(as.numeric(paste(T_data[,7])))
campominacci=c(as.numeric(paste(T_data[,8])))
capaccio=c(as.numeric(paste(T_data[,9])))
castellaccio=c(as.numeric(paste(T_data[,10])))
corniolo=c(as.numeric(paste(T_data[,11])))
diga=c(as.numeric(paste(T_data[,12])))
lama=c(as.numeric(paste(T_data[,13])))
X=c(as.numeric(paste(coord_data[,2])))
Y=c(as.numeric(paste(coord_data[,3])))
Ttot=matrix(c(braccina,camaldoli,campigna,campominacci,capaccio,castellaccio,corni
lo,diga,lama),nrow=nrow(T_data),ncol=9,byrow=FALSE)
coord=matrix(c(X,Y),nrow=nrow(coord_data),ncol=2,byrow=FALSE)
Z=c(as.numeric(paste(coord_data[,4])))

braccina=c(as.numeric(paste(T_dataNA[,5])))
camaldoli=c(as.numeric(paste(T_dataNA[,6])))
campigna=c(as.numeric(paste(T_dataNA[,7])))
campominacci=c(as.numeric(paste(T_dataNA[,8])))
capaccio=c(as.numeric(paste(T_dataNA[,9])))
castellaccio=c(as.numeric(paste(T_dataNA[,10])))
corniolo=c(as.numeric(paste(T_dataNA[,11])))
diga=c(as.numeric(paste(T_dataNA[,12])))
lama=c(as.numeric(paste(T_dataNA[,13])))
TtotNA=matrix(c(braccina,camaldoli,campigna,campominacci,capaccio,castellaccio,cor
niolo,diga,lama),nrow=nrow(T_dataNA),ncol=9,byrow=FALSE)

n_staz=nrow(coord)

#Gradienti orari
GH=matrix(rep(-999,2*nrow(Ttot)),nrow=nrow(Ttot),ncol=2)
for (i in 1:nrow(Ttot))
{
  T=c(TtotNA[i,])
  t=na.omit(T)
  if (length(t)>3)
  {
    reg=lm(T~Z)
    GH[i,1]=reg$coef[1]
    GH[i,2]=reg$coef[2]
  }
}
```

```

}
for (i in 2:nrow(Ttot))
{
  if (GH[i,1]==(-999))
  {
    GH[i,1]=GH[i-1,1]
    GH[i,2]=GH[i-1,2]
  }
}

JK=rep(NA,nrow(Ttot))

#Interpolazione
for (i in 1:nrow(Ttot))
{
  JK[i]=GH[i,1]+Z[1]*GH[i,2]
}

#Stampa risultato
write.table(JK,file="JK")

```

6. Multiple Linear Regression

```

rm(list=ls())

setwd("C:\\Users\\Mattia\\Desktop\\NeriLM_RegionalizzazioneParametri\\R\\Interp_T_t
emporale\\MultipleReg_hourly")

#Lettura dati
T_data=read.table("TempTOT_0.txt",          sep="",          header=FALSE,
colClasses=rep("numeric",13))
T_dataNA=read.table("TempTOT_0NA.txt",      sep="",          header=FALSE,
colClasses=rep("numeric",13))
coord_data=read.table("coordinate_staz.txt",
sep="",header=FALSE,colClasses=rep("numeric",4))

#Creazione file
braccina=c(as.numeric(paste(T_data[,5])))
camaldoli=c(as.numeric(paste(T_data[,6])))
campigna=c(as.numeric(paste(T_data[,7])))
campominacci=c(as.numeric(paste(T_data[,8])))
capaccio=c(as.numeric(paste(T_data[,9])))
castellaccio=c(as.numeric(paste(T_data[,10])))
corniolo=c(as.numeric(paste(T_data[,11])))
diga=c(as.numeric(paste(T_data[,12])))
lama=c(as.numeric(paste(T_data[,13])))
X=c(as.numeric(paste(coord_data[,2])))
Y=c(as.numeric(paste(coord_data[,3])))
Ttot=matrix(c(braccina,camaldoli,campigna,campominacci,capaccio,castellaccio,corni
lo,diga,lama),nrow=nrow(T_data),ncol=9,byrow=FALSE)
coord=matrix(c(X,Y),nrow=nrow(coord_data),ncol=2,byrow=FALSE)
Z=c(as.numeric(paste(coord_data[,4])))

braccina=c(as.numeric(paste(T_dataNA[,5])))
camaldoli=c(as.numeric(paste(T_dataNA[,6])))
campigna=c(as.numeric(paste(T_dataNA[,7])))

```

```

campominacci=c(as.numeric(paste(T_dataNA[,8])))
capaccio=c(as.numeric(paste(T_dataNA[,9])))
castellaccio=c(as.numeric(paste(T_dataNA[,10])))
corniolo=c(as.numeric(paste(T_dataNA[,11])))
diga=c(as.numeric(paste(T_dataNA[,12])))
lama=c(as.numeric(paste(T_dataNA[,13])))
TtotNA=matrix(c(braccina,camaldoli,campigna,campominacci,capaccio,castellaccio,cor
niolo,diga,lama),nrow=nrow(T_dataNA),ncol=9,byrow=FALSE)

n_staz=nrow(coord)

#Coefficienti della regressione multipla oraria
GH=matrix(rep(-999,2*nrow(Ttot)),nrow=nrow(Ttot),ncol=4)
for (i in 1:nrow(Ttot))
{
  T=c(TtotNA[i,])
  t=na.omit(T)
  if (length(t)>3)
  {
    reg=lm(T~Z+X+Y)
    GH[i,1]=reg$coef[1]
    GH[i,2]=reg$coef[2]
    GH[i,3]=reg$coef[3]
    GH[i,4]=reg$coef[4]
  }
}
for (i in 2:nrow(Ttot))
{
  if (GH[i,1]==(-999))
  {
    GH[i,1]=GH[i-1,1]
    GH[i,2]=GH[i-1,2]
    GH[i,3]=GH[i-1,3]
    GH[i,4]=GH[i-1,4]
  }
}

JK=rep(NA,nrow(Ttot))

#Interpolazione
for (i in 1:nrow(Ttot))
{
  JK[i]=GH[i,1]+Z[1]*GH[i,2]+X[1]*GH[i,3]+Y[1]*GH[i,4]
}

#Stampa risultato
write.table(JK,file="JK")

```

Script in R costruito per la creazione delle mappe di temperatura ad ogni passo temporale

```

rm(list=ls())

setwd("C:\\Users\\Mattia\\Dropbox\\IO\\NeriLM_Tesi\\R\\Interp_T_temporale\\BaciniUnit
i")

#Assemblaggio dati di input

```

```

GH=read.table("GHM",sep="",header=FALSE,colClasses=rep("numeric",4))
#coefficienti di regressione
DEM=read.table("DEMBaciniuniti", sep="",header=FALSE,
colClasses=rep("numeric",209))
X0=714480
Y0=4872320
tstep=nrow(GH)

#Creazione vettore dei nomi delle mappe di temperatura
tempi_data=read.table("tempi.txt", sep="", header=FALSE)
tempi=c(as.character(paste(tempi_data[,1])))

#Creazione mappe delle coordinate X e Y
X=DEM
for (i in 1:nrow(DEM))
{
  for (j in 1:ncol(DEM))
  {
    X[i,j]=X0+(j-1)*80
  }
}

Y=DEM
for (i in 1:nrow(DEM))
{
  for (j in 1:ncol(DEM))
  {
    Y[i,j]=Y0-(i-1)*80
  }
}

#Interpolazione
l=matrix(1,nrow=nrow(DEM),ncol=ncol(DEM))
for (k in 35065:43848)
{
  T=l*GH[k,1]+DEM*GH[k,2]+X*GH[k,3]+Y*GH[k,4]
  T=replace(T,is.na(T),-9999)
  T=round(T,digits=2)
}

#Salvataggio matrice
write.table(T,file=paste("Temp",tempi[k],".txt",sep=""),row.names=FALSE,col.names=FALSE)
}

```


Appendice 2

Script costruiti per la creazione dei file di input delle temperature per le varie fasce (esempio bacino Campigna)

Creazione File dati orari:

```
rm(list=ls())

setwd("C:\\Users\\Mattia\\Dropbox\\IO\\Tesi\\NeriLM_Tesi\\R\\Interp_T_temporale\\Ca
mpigna")

#Carico i dati
#Coefficienti regressione
GH=read.table("GHM", sep="", header=FALSE, colClasses=rep("numeric",4))
#DEM e maschere zone
DEM=read.table("DEMcampigna", sep="", header=FALSE,
colClasses=rep("numeric",77))
Z1=read.table("Campigna1", sep="", header=FALSE, colClasses=rep("numeric",77))
Z2=read.table("Campigna2", sep="", header=FALSE, colClasses=rep("numeric",77))
Z3=read.table("Campigna3", sep="", header=FALSE, colClasses=rep("numeric",77))
Z4=read.table("Campigna4", sep="", header=FALSE, colClasses=rep("numeric",77))

X0=718800 #coordinate estremo
Y0=4863680
tstep=nrow(GH)
nzone=4

#Creazione mappe delle coordinate
X=DEM
for (i in 1:nrow(DEM))
{
  for (j in 1:ncol(DEM))
  {
    X[i,j]=X0+(j-1)*80
  }
}

Y=DEM
for (i in 1:nrow(DEM))
{
  for (j in 1:ncol(DEM))
  {
    Y[i,j]=Y0-(i-1)*80
  }
}

#Interpolazione
I=matrix(1,nrow=nrow(DEM),ncol=ncol(DEM))
Temp_Campigna=matrix(nrow=tstep,ncol=nzone) #Inizializzo File di output
for (k in 1:tstep)
{
  T=I*GH[k,1]+DEM*GH[k,2]+X*GH[k,3]+Y*GH[k,4] #Interpolazione
  #Eseguo media sulle varie zone
  Temp_Campigna[k,1]=mean(as.matrix(T*Z1),na.rm=TRUE)
```

```

Temp_Campigna[k,2]=mean(as.matrix(T*Z2),na.rm=TRUE)
Temp_Campigna[k,3]=mean(as.matrix(T*Z3),na.rm=TRUE)
Temp_Campigna[k,4]=mean(as.matrix(T*Z4),na.rm=TRUE)
}
Temp_Campigna=round(Temp_Campigna,digits=2) #Arrotondamento

#Salvataggio File di output
write.table(Temp_Campigna,file="Temp_CampignaH.txt",row.names=FALSE,col.names=FALSE)

Aggregazione a scala giornaliera:

rm(list=ls())

setwd("C:\\Users\\Mattia\\Dropbox\\IO\\Tesi\\NeriLM_Tesi\\R\\Interp_T_temporale\\Campigna")

nzones=4
#Carico dati orari
dataH=read.table("Temp_CampignaH.txt", header=FALSE, sep="",
colClasses=rep("numeric",nzones))

dataD=matrix(nrow=nrow(dataH)/24,ncol=nzones) #File di output

#Aggregazione a scala giornaliera
for (i in 1:nrow(dataD))
{
dataD[i,1]=mean(dataH[((i-1)*24+1):(i*24),1])
dataD[i,2]=mean(dataH[((i-1)*24+1):(i*24),2])
dataD[i,3]=mean(dataH[((i-1)*24+1):(i*24),3])
dataD[i,4]=mean(dataH[((i-1)*24+1):(i*24),4])
}
dataD=round(dataD,digits=2) #Arrotondamento

#Salvataggio File di output
write.table(dataD,file="Temp_CampignaD.txt", row.names=FALSE, col.names=FALSE)

```

Script costruiti per la creazione dei file di input delle precipitazioni per le varie fasce (esempio bacino Campigna)

Creazione File dei dati orari:

```

rm(list=ls())

setwd("D:/Mattia/Mia/Uni/Tesi/Piogge/prova_inv_dist1")

#Carico maschere zone
Z1=read.table("Campigna1", sep="", header=FALSE, colClasses=rep("numeric",209))
Z2=read.table("Campigna2", sep="", header=FALSE, colClasses=rep("numeric",209))
Z3=read.table("Campigna3", sep="", header=FALSE, colClasses=rep("numeric",209))
Z4=read.table("Campigna4", sep="", header=FALSE, colClasses=rep("numeric",209))

#Definisco vettore per caricare le mappe di pioggia
tempi_data=read.table("tempi.txt", sep="", header=FALSE)

```

```

tempi=c(as.character(paste(tempi_data[,1])))

nzone=4
tstep=157800

Prec_Campigna=matrix(nrow=tstep,ncol=nzone) #Inizializzo File di output

start=Sys.time()

#Creo File di output
for (k in 1:tstep)
{
  #Carico mappa k-esima
  P=read.table(paste(tempi[k], ".txt"),sep=" ",header=FALSE,skip=6,
colClasses=rep("numeric",209))
  #Eseguo media sulle zone
  Prec_Campigna[k,1]=mean(as.matrix(P*Z1),na.rm=TRUE)
  Prec_Campigna[k,2]=mean(as.matrix(P*Z2),na.rm=TRUE)
  Prec_Campigna[k,3]=mean(as.matrix(P*Z3),na.rm=TRUE)
  Prec_Campigna[k,4]=mean(as.matrix(P*Z4),na.rm=TRUE)
  if (k%%10000==0) {print(k)}
}

end=Sys.time()
end-start

Prec_Campigna=round(Prec_Campigna,digits=2) #Arrotondamento

#Salvataggio File di output
write.table(Prec_Campigna,file="Prec_CampignaH.txt",row.names=FALSE,col.names=
FALSE)

#Aggregazione a scala giornaliera:

rm(list=ls())

setwd("C:\\Users\\Mattia\\Dropbox\\IO\\Tesi\\NeriLM_Tesi\\R\\Precipitazioni\\ZoneP\\Ca
mpigna")

nzones=4
#Carico file dati orari
dataH=read.table("Prec_CampignaH.txt", header=FALSE, sep=" ",
colClasses=rep("numeric",nzones))

dataD=matrix(nrow=nrow(dataH)/24,ncol=nzones) #Inizializzo File di output

#Aggregazione a scala giornaliera
for (i in 1:nrow(dataD))
{
  dataD[i,1]=sum(dataH[((i-1)*24+1):(i*24),1])
  dataD[i,2]=sum(dataH[((i-1)*24+1):(i*24),2])
  dataD[i,3]=sum(dataH[((i-1)*24+1):(i*24),3])
  dataD[i,4]=sum(dataH[((i-1)*24+1):(i*24),4])
}
dataD=round(dataD,digits=2) #Arrotondamento

#Salvataggio File di output

```

```
write.table(dataD,file="Prec_CampignaD.txt", row.names=FALSE, col.names=FALSE)
```

Script costruiti per la creazione dei file di input dell'evapotraspirato per le varie fasce (esempio bacino Campigna)

```
rm(list=ls())

setwd("C:\\Users\\Mattia\\Dropbox\\IO\\Tesi\\NeriLM_Tesi\\R\\ETP\\Campigna")

nzones=4
#Carico File di input delle temperature
dataH=read.table("Temp_CampignaH.txt", header=FALSE, sep="",
colClasses=rep("numeric",nzones))

#Creo File T medie mensili per ricavare i coefficienti W(i), a e b per le varie fasce
mesi_data=read.table("mesi.txt", header=FALSE, sep="", colClasses="numeric")
mesi=c(as.numeric(paste(mesi_data[,1])))

AM=matrix(nrow=12,ncol=nzones)

for (i in 1:12)
{
  AM[i,1]=mean(dataH[mesi==i,1])
  AM[i,2]=mean(dataH[mesi==i,2])
  AM[i,3]=mean(dataH[mesi==i,3])
  AM[i,4]=mean(dataH[mesi==i,4])
}
AM=round(AM,digits=2)

write.table(AM,file="TmediaM_CampignaM.txt", row.names=FALSE,
col.names=FALSE)

#####
###

#Carico file di input
N=read.table("N.txt", header=FALSE, sep="", colClasses=rep("numeric",2))
data_ETP=read.table("WtaAB.txt", header=FALSE, sep="",
colClasses=rep("numeric",nzones))
Wta=data_ETP[1:12,]
a=data_ETP[13,]
b=data_ETP[14,]

#Creo File di output evapotraspirato
ETP=dataH #Inizializzo File di output
start=Sys.time()
for (j in 1:nzones)
{
  for (i in 1:nrow(ETP))
  {
    ETP[i,j]=(a[j]+b[j]*N[mesi[i],1]*Wta[mesi[i],j]*dataH[i,j])/(24*N[mesi[i],2])
  }
}
ETP[ETP<0]=0
end=Sys.time()
end-start
```

```

ETP=round(ETP,digits=4) #Arrotondamento

#Salvataggio File dati orari
write.table(ETP,file="ETP_CampignaH.txt",row.names=FALSE,col.names=FALSE)

#Aggregazione giornaliera
ETPD=matrix(nrow=nrow(ETP)/24,ncol=nzones)

for (i in 1:nrow(ETPD))
{
  ETPD[i,1]=sum(ETP[((i-1)*24+1):(i*24),1])
  ETPD[i,2]=sum(ETP[((i-1)*24+1):(i*24),2])
  ETPD[i,3]=sum(ETP[((i-1)*24+1):(i*24),3])
  ETPD[i,4]=sum(ETP[((i-1)*24+1):(i*24),4])
}

#Arrotondamento e salvataggio File dati giornalieri
ETPD=round(ETPD,digits=4)
write.table(ETPD,file="ETP_CampignaD.txt", row.names=FALSE, col.names=FALSE)

```

Script costruiti per la calibrazione e la validazione a scala giornaliera in modalità concentrata (esempio bacino Celle)

```

rm(list=ls())

setwd("C:\\Users\\Mattia\\Dropbox\\IO\\NeriLM_Tesi\\R\\TUWmodel_Daily_Neri\\Lumped\\TUW_Celle_Daily_Lumped")

# Richiamo i pacchetti utili per l'elaborazione
library(zoo)
library(TUWmodel)
library(DEoptim)

Start=Sys.time()

# Leggo i dati relativi al bacino di interesse
data=read.table("CelleD.txt",sep=" ",header=FALSE,colClasses=rep("numeric",16))
parea=c(0.287, 0.464, 0.150, 0.099)
area=14.60 #(kmq)

# Creo le serie temporali
days=as.Date(strptime(paste(data[,1], data[,2], data[,3]), format="%d %m %Y"))
days1=days[1:3531]
days2=days[3532:6940]

T=zoo(data[,4:7],order.by=days) # temperature medie giornaliere (0C)
EP=zoo(data[,8:11],order.by=days) # evapotraspirato potenziale (mm/giorno)
P=zoo(data[,12:15],order.by=days) # precipitazioni (mm/giorno)
Q=zoo(data[,16],order.by=days) # portate medie giornaliere osservate (mc/s)

# Suddivido le serie temporali nei due periodi di uguale lunghezza
P1=window(P,end=as.Date(strptime(paste("31 8 2005"),format="%d %m %Y")))

```

```

T1=window(T,end=as.Date(strptime(paste("31 8 2005"),format="%d %m %Y")))
EP1=window(EP,end=as.Date(strptime(paste("31 8 2005"),format="%d %m %Y")))
Q1=window(Q,end=as.Date(strptime(paste("31 8 2005"),format="%d %m %Y")))

P2=window(P,start=as.Date(strptime(paste("1 9 2005"),format="%d %m %Y")))
T2=window(T,start=as.Date(strptime(paste("1 9 2005"),format="%d %m %Y")))
EP2=window(EP,start=as.Date(strptime(paste("1 9 2005"),format="%d %m %Y")))
Q2=window(Q,start=as.Date(strptime(paste("1 9 2005"),format="%d %m %Y")))

# Aggrego Spazialmente i dati per utilizzo LUMPED
P1=apply(P1, 1, weighted.mean, w=parea)
T1=apply(T1, 1, weighted.mean, w=parea)
EP1=apply(EP1, 1, weighted.mean, w=parea)

P2=apply(P2, 1, weighted.mean, w=parea)
T2=apply(T2, 1, weighted.mean, w=parea)
EP2=apply(EP2, 1, weighted.mean, w=parea)

# Definisco il periodo di warmup
warm_up=365

# Definisco la funzione obiettivo
MSE_optim=function(param,precip,temp,potevap,runoff,area)

{simu=TUWmodel(prec=as.numeric(precip),airt=as.numeric(temp),ep=as.numeric(pote
vap),area=area,param)$q
## CONTROLLARE ISTRUZIONE NaN
simu[is.nan(simu)] <- -999
simu=simu[-c(1:warm_up)] # rimuovo dalla serie il periodo di warmup
obse=3.6*24*as.numeric(runoff)/area
obse=obse[-c(1:warm_up)] # rimuovo dalla serie il periodo di warmup
mean((simu-obse)^2,na.rm=TRUE) # inserisco la funzione obiettivo: mean square
error
}

# Calibrazione sul periodo 1
rcal=list()
bestval1=1000
for (i in 1:5)
{
  calib_MSE_1=DEoptim(fn=MSE_optim,
                    lower=c(0.9,0.0,1.0,-3.0,-2.0,0.0,0.0,0.0,0.0,2.0,30.0,1.0,0.0,0.0,0.0),
                    upper=c(1.5,5.0,3.0,1.0,2.0,1.0,600.0,20.0,2.0,30.0,250.0,100.0,8.0,30.0,50.0),
                    control=DEoptim.control(NP=NA, itermax=600, reltol=1e-4, steptol=50,
                    trace=50,

parallelType=1,packages="TUWmodel",parVar=c("warm_up")),
                    precip=P1,temp=T1,potevap=EP1,runoff=Q1,area=area)

  rcal[[i]]=calib_MSE_1

  if(calib_MSE_1$optim$bestval<bestval1)
  { bestmem1=calib_MSE_1$optim$bestmem
    print(bestmem1)
    bestval1=calib_MSE_1$optim$bestval
    print(bestval1)
  }
}

```

```

names(bestmem1)=c("SCF","DDF","Tr","Ts","Tm","LPrat","FC","BETA","k0","k1","k2","I
suz","cperc","bmax","croute")
}

# Salvo le variabili ottenute
save.image("WORKSPACEcelleDlumped.RData")

End=Sys.time()
End-Start

# Validazioni
cal=TUWmodel(prec=P1,airt=T1,ep=EP1,area=area,param=bestmem1)
val=TUWmodel(prec=P2,airt=T2,ep=EP2,area=area,param=bestmem1)

# Definisco la funzione per il calcolo degli indici di efficienza
MSE=function (simu,obse)
{simu=simu[-c(1:warm_up)]
obse=obse[-c(1:warm_up)]

# RMSE = root mean square error (mm/d)
RMSE=sqrt(mean((simu-obse)^2,na.rm=TRUE))
# NE = Nash efficiency ()
mobse=mean(obse,na.rm=TRUE)
NE=1-sum((simu-obse)^2,na.rm=TRUE)/sum((obse-mobse)^2,na.rm=TRUE)
# INE = log Nash efficiency ()
mlobse=mean(log(obse),na.rm=TRUE)
INE=1-sum((log(simu)-log(obse))^2,na.rm=TRUE)/sum((log(obse)-
mlobse)^2,na.rm=TRUE)
# B = bias (mm/d)
B=mean(simu-obse,na.rm=TRUE)
# MAE = mean absolute error (mm/d)
MAE=mean(abs(simu-obse),na.rm=TRUE)
# MAIE = mean absolute log error (mm/d)
MAIE=exp(mean(abs(log(simu)-log(obse)),na.rm=TRUE))
# VE = volume error (%/%)
VE=(sum(simu[!is.na(obse)])-sum(obse,na.rm=TRUE))/sum(obse,na.rm=TRUE)

output=c(RMSE,NE,INE,B,MAE,MAIE,VE)
names(output)=c("RMSE (mm/d)","Nash efficiency ()","Log Nash efficiency ()","Bias
(mm/d)",
"Mean Absolute Error (mm/d)","Mean Absolute Log Error (mm/d)","Volume
error (%/%)")
return(output)
}

# Calcolo le efficienze
E_cal=MSE(cal$q,3.6*24*Q1/area)
E_val=MSE(val$q,3.6*24*Q2/area)

# Creo i file di output: parametri ed efficienze
write.table(bestmem1,file="Parametri_calibrazione.txt")
write.table(t(cal$q),file="Calibrazione.txt",row.names=FALSE,col.names=FALSE)
write.table(t(val$q),file="Validazione.txt",row.names=FALSE,col.names=FALSE)
write.table(E_cal,file="Efficienze_calibrazione.txt")
write.table(E_val,file="Efficienze_validazione.txt")

```

```
# Salvo le variabili ottenute
save.image("WORKSPACEcelleDlumped.RData")
```

Script costruiti per la calibrazione e la validazione a scala giornaliera in modalità semi - distribuita (esempio bacino Celle)

```
rm(list=ls())

setwd("C:\\Users\\nemat\\Dropbox\\IO\\NeriLM_Tesi\\R\\TUWmodel_Daily_Neri\\TUW_
Celle_Daily")

# Richiamo i pacchetti utili per l'elaborazione
library(zoo)
library(TUWmodel)
library(DEoptim)

Start=Sys.time()

# Leggo i dati relativi al bacino di interesse
data=read.table("CelleD.txt",sep=" ",header=FALSE,colClasses=rep("numeric",16))
area=c(4.19, 6.78, 2.20, 1.44)
areatot=14.60 #(kmq)

# Creo le serie temporali
days=as.Date(strptime(paste(data[,1], data[,2], data[,3]), format="%d %m %Y"))
days1=days[1:3531]
days2=days[3532:6940]

T=zoo(data[,4:7],order.by=days) # temperature medie giornaliere (°C)
EP=zoo(data[,8:11],order.by=days) # evapotraspirato potenziale (mm/giorno)
P=zoo(data[,12:15],order.by=days) # precipitazioni (mm/giorno)
Q=zoo(data[,16],order.by=days) # portate medie giornaliere osservate (mc/s)

# Suddivido le serie temporali nei due periodi di uguale lunghezza
P1=window(P,end=as.Date(strptime(paste("31 8 2005"),format="%d %m %Y")))
T1=window(T,end=as.Date(strptime(paste("31 8 2005"),format="%d %m %Y")))
EP1=window(EP,end=as.Date(strptime(paste("31 8 2005"),format="%d %m %Y")))
Q1=window(Q,end=as.Date(strptime(paste("31 8 2005"),format="%d %m %Y")))

P2=window(P,start=as.Date(strptime(paste("1 9 2005"),format="%d %m %Y")))
T2=window(T,start=as.Date(strptime(paste("1 9 2005"),format="%d %m %Y")))
EP2=window(EP,start=as.Date(strptime(paste("1 9 2005"),format="%d %m %Y")))
Q2=window(Q,start=as.Date(strptime(paste("1 9 2005"),format="%d %m %Y")))

# Definisco il periodo di warmup
warm_up=365

# Definisco la funzione obiettivo
MSE_optim=function(param,precip,temp,potevap,runoff,area)
{simu=TUWmodel(prec=precip,airt=temp,ep=potevap,area=area,param)$q
## CONTROLLARE ISTRUZIONE NaN
simu[is.nan(simu)] <- -999
simu=simu[-c(1:warm_up)] # rimuovo dalla serie il periodo di warmup
obse=3.6*24*runoff/14.60
obse=obse[-c(1:warm_up)] # rimuovo dalla serie il periodo di warmup
mean((simu-obse)^2,na.rm=TRUE) # inserisco la funzione obiettivo: mean square
error
}
```

```

# Calibrazione sul periodo 1
rcal=list()
bestval1=1000
for (i in 1:5)
{
  calib_MSE_1=DEoptim(fn=MSE_optim,
                    lower=c(0.9,0.0,1.0,-3.0,-2.0,0.0,0.0,0.0,0.0,2.0,30.0,1.0,0.0,0.0,0.0),
                    upper=c(1.5,5.0,3.0,1.0,2.0,1.0,600.0,20.0,2.0,30.0,250.0,100.0,8.0,30.0,50.0),
                    control=DEoptim.control(NP=NA, itermax=600, reftol=1e-4, steptol=50,
                    trace=50,

parallelType=1,packages="TUWmodel",parVar=c("warm_up")),
                    precip=P1,temp=T1,potevap=EP1,runoff=Q1,area=area)

  rcal[[i]]=calib_MSE_1

  if(calib_MSE_1$optim$bestval<bestval1)
  { bestmem1=calib_MSE_1$optim$bestmem
    print(bestmem1)
    bestval1=calib_MSE_1$optim$bestval
    print(bestval1)
  }

names(bestmem1)=c("SCF","DDF","Tr","Ts","Tm","LPrat","FC","BETA","k0","k1","k2","I
suz","cperc","bmax","croute")
}

# Salvo le variabili ottenute
save.image("WORKSPACEcelleD.RData")

End=Sys.time()
End-Start

# Validazioni
cal=TUWmodel(prec=P1,airt=T1,ep=EP1,area=area,param=bestmem1)
val=TUWmodel(prec=P2,airt=T2,ep=EP2,area=area,param=bestmem1)

# Definisco la funzione per il calcolo degli indici di efficienza
MSE=function (simu,obse)
{simu=simu[-c(1:warm_up)]
obse=obse[-c(1:warm_up)]

# RMSE = root mean square error (mm/d)
RMSE=sqrt(mean((simu-obse)^2,na.rm=TRUE))
# NE = Nash efficiency ()
mobse=mean(obse,na.rm=TRUE)
NE=1-sum((simu-obse)^2,na.rm=TRUE)/sum((obse-mobse)^2,na.rm=TRUE)
# INE = log Nash efficiency ()
mlobse=mean(log(obse),na.rm=TRUE)
INE=1-sum((log(simu)-log(obse))^2,na.rm=TRUE)/sum((log(obse)-
mlobse)^2,na.rm=TRUE)
# B = bias (mm/d)
B=mean(simu-obse,na.rm=TRUE)
# MAE = mean absolute error (mm/d)
MAE=mean(abs(simu-obse),na.rm=TRUE)

```

```

# MAIE = mean absolute log error (mm/d)
MAIE=exp(mean(abs(log(simu)-log(obse)),na.rm=TRUE))
# VE = volume error (%/%)
VE=(sum(simu[!is.na(obse)])-sum(obse,na.rm=TRUE))/sum(obse,na.rm=TRUE)

output=c(RMSE,NE,INE,B,MAE,MAIE,VE)
names(output)=c("RMSE (mm/d)","Nash efficiency ()","Log Nash efficiency ()","Bias
(mm/d)",
               "Mean Absolute Error (mm/d)","Mean Absolute Log Error (mm/d)","Volume
error (%/%)")
return(output)
}

# Calcolo le efficienze
E_cal=MSE(cal$q,3.6*24*Q1/areatot)
E_val=MSE(val$q,3.6*24*Q2/areatot)

# Creo i file di output: parametri ed efficienze
write.table(bestmem1,file="Parametri_calibrazione.txt")
write.table(cal$q,file="Calibrazione.txt",row.names=FALSE,col.names=FALSE)
write.table(val$q,file="Validazione.txt",row.names=FALSE,col.names=FALSE)
write.table(E_cal,file="Efficienze_calibrazione.txt")
write.table(E_val,file="Efficienze_validazione.txt")

# Salvo le variabili ottenute
save.image("WORKSPACEcelleD.RData")

```

Script costruiti per la calibrazione e la validazione a scala oraria in modalità concentrata (esempio bacino Celle)

```

rm(list=ls())

setwd("C:\\Users\\nemat\\Dropbox\\IO\\NeriLM_Tesi\\R\\TUWmodel_Hourly_Neri\\Lumped\\TUW_Celle_Hourly_Lumped")

# Richiamo i pacchetti utili per l'elaborazione
library(zoo)
library(TUWmodel)
library(DEoptim)
library(chron)

Start=Sys.time()

# Leggo i dati relativi al bacino di interesse
data=read.table("CelleH.txt",sep=" ",header=FALSE,colClasses=rep("numeric",19))
parea=c(4.19, 6.78, 2.20, 1.44)
area=14.60 #(kmq)

# Creo le serie temporali
hours=chron(dates=(paste(data[,4], data[,5],
data[,6])),times=(paste(data[,1],data[,2],data[,3])), format=c("d m y", "h m s"))
hours1=hours[1:84744]
hours2=hours[84745:166560]

T=zoo(data[,7:10],order.by=hours) # temperature medie orarie (°C)
EP=zoo(data[,11:14],order.by=hours) # evapotraspirato potenziale (mm/ora)

```

```

P=zoo(data[,15:18],order.by=hours) # precipitazioni (mm/ora)
Q=zoo(data[,19],order.by=hours) # portate medie orarie osservate (mc/s)

# Suddivido le serie temporali nei due periodi di uguale lunghezza
P1=window(P,end=chron(dates=(paste("31 8 2005")),times=(paste("23 00 00")),
format=c("d m y", "h m s")))
T1=window(T,end=chron(dates=(paste("31 8 2005")),times=(paste("23 00 00")),
format=c("d m y", "h m s")))
EP1=window(EP,end=chron(dates=(paste("31 8 2005")),times=(paste("23 00 00")),
format=c("d m y", "h m s")))
Q1=window(Q,end=chron(dates=(paste("31 8 2005")),times=(paste("23 00 00")),
format=c("d m y", "h m s")))

P2=window(P,start=chron(dates=(paste("1 9 2005")),times=(paste("00 00 00")),
format=c("d m y", "h m s")))
T2=window(T,start=chron(dates=(paste("1 9 2005")),times=(paste("00 00 00")),
format=c("d m y", "h m s")))
EP2=window(EP,start=chron(dates=(paste("1 9 2005")),times=(paste("00 00 00")),
format=c("d m y", "h m s")))
Q2=window(Q,start=chron(dates=(paste("1 9 2005")),times=(paste("00 00 00")),
format=c("d m y", "h m s")))

# Aggrego Spazialmente i dati per utilizzo LUMPED
P1=apply(P1, 1, weighted.mean, w=parea)
T1=apply(T1, 1, weighted.mean, w=parea)
EP1=apply(EP1, 1, weighted.mean, w=parea)

P2=apply(P2, 1, weighted.mean, w=parea)
T2=apply(T2, 1, weighted.mean, w=parea)
EP2=apply(EP2, 1, weighted.mean, w=parea)

# Definisco il periodo di warmup
warm_up=365*24

# Definisco la funzione obiettivo
MSE_optim=function(param,precip,temp,potevap,runoff,area)

{simu=TUWmodel(prec=as.numeric(precip),airt=as.numeric(temp),ep=as.numeric(pote
vap),area=area,param)$q
simu[is.nan(simu)] <- -999 ## CONTROLLARE ISTRUZIONE NaN
simu=simu[-c(1:warm_up)] # rimuovo dalla serie il periodo di warmup
obse=3.6*as.numeric(runoff)/area
obse=obse[-c(1:warm_up)] # rimuovo dalla serie il periodo di warmup
mean((simu-obse)^2,na.rm=TRUE) # inserisco la funzione obiettivo: mean square
error
}

rcal=list()
# Calibrazione sul periodo 1
bestval1=1000
for (i in 1:3)
{
calib_MSE_1=DEoptim(fn=MSE_optim,
lower=c(0.9,0.0,1.0,-3.0,-2.0,0.0,0.0,0.0,0.0,2.0,30.0,1.0,0.0,0.0,0.0),
upper=c(1.5,5.0,3.0,1.0,2.0,1.0,600.0,20.0,2.0,30.0,250.0,100.0,8.0,30.0,50.0),
control=DEoptim.control(NP=NA, itermax=600, reltol=1e-4, steptol=50,
trace=50,

```

```

parallelType=1,packages="TUWmodel",parVar=c("warm_up")),
      precip=P1,temp=T1,potevap=EP1,runoff=Q1,area=area)

rcal[[i]]=calib_MSE_1

if(calib_MSE_1$optim$bestval<bestval1)
{ bestmem1=calib_MSE_1$optim$bestmem
print(bestmem1)
bestval1=calib_MSE_1$optim$bestval
print(bestval1)
}

names(bestmem1)=c("SCF","DDF","Tr","Ts","Tm","LPrat","FC","BETA","k0","k1","k2","I
suz","cperc","bmax","crouete")
}

# Salvo le variabili ottenute
save.image("WORKSPACEcelleHlumped.RData")

End=Sys.time()
End-Start

# Validazioni
cal=TUWmodel(prec=P1,airt=T1,ep=EP1,area=area,param=bestmem1)
val=TUWmodel(prec=P2,airt=T2,ep=EP2,area=area,param=bestmem1)

# Definisco la funzione per il calcolo degli indici di efficienza
MSE=function (simu,obse)
{simu=simu[-c(1:warm_up)]
obse=obse[-c(1:warm_up)]

# RMSE = root mean square error (mm/d)
RMSE=sqrt(mean((simu-obse)^2,na.rm=TRUE))
# NE = Nash efficiency ()
mobse=mean(obse,na.rm=TRUE)
NE=1-sum((simu-obse)^2,na.rm=TRUE)/sum((obse-mobse)^2,na.rm=TRUE)
# INE = log Nash efficiency ()
mlobse=mean(log(obse),na.rm=TRUE)
INE=1-sum((log(simu)-log(obse))^2,na.rm=TRUE)/sum((log(obse)-
mlobse)^2,na.rm=TRUE)
# B = bias (mm/d)
B=mean(simu-obse,na.rm=TRUE)
# MAE = mean absolute error (mm/d)
MAE=mean(abs(simu-obse),na.rm=TRUE)
# MAIE = mean absolute log error (mm/d)
MAIE=exp(mean(abs(log(simu)-log(obse)),na.rm=TRUE))
# VE = volume error (%/%)
VE=(sum(simu[!is.na(obse)])-sum(obse,na.rm=TRUE))/sum(obse,na.rm=TRUE)

output=c(RMSE,NE,INE,B,MAE,MAIE,VE)
names(output)=c("RMSE (mm/d)","Nash efficiency ()","Log Nash efficiency ()","Bias
(mm/d)",
      "Mean Absolute Error (mm/d)","Mean Absolute Log Error (mm/d)","Volume
error (%/%)")
return(output)
}

```

```

# Calcolo le efficienze
E_cal=MSE(cal$q,3.6*Q1/area)
E_val=MSE(val$q,3.6*Q2/area)

# Creo i file di output: parametri ed efficienze
write.table(bestmem1,file="Parametri_calibrazione.txt")
write.table(t(cal$q),file="Calibrazione.txt",row.names=FALSE,col.names=FALSE)
write.table(t(val$q),file="Validazione.txt",row.names=FALSE,col.names=FALSE)
write.table(E_cal,file="Efficienze_calibrazione.txt")
write.table(E_val,file="Efficienze_validazione.txt")

# Salvo le variabili ottenute
save.image("WORKSPACEcelleHlumped.RData")

```

Script costruiti per la calibrazione e la validazione a scala oraria in modalità semi - distribuita (esempio bacino Celle)

```

rm(list=ls())

setwd("C:\\Users\\nemat\\Dropbox\\IO\\NeriLM_Tesi\\R\\TUWmodel_Hourly_Neri\\TUW_Celle_Hourly")

# Richiamo i pacchetti utili per l'elaborazione
library(zoo)
library(TUWmodel)
library(DEoptim)
library(chron)

Start=Sys.time()

# Leggo i dati relativi al bacino di interesse
data=read.table("CelleH.txt",sep=" ",header=FALSE,colClasses=rep("numeric",19))
area=c(4.19, 6.78, 2.20, 1.44)
areatot=14.60 #(kmq)

# Creo le serie temporali
hours=chron(dates=(paste(data[,4], data[,5],
data[,6])),times=(paste(data[,1],data[,2],data[,3])), format=c("d m y", "h m s"))
hours1=hours[1:84744]
hours2=hours[84745:166560]

T=zoo(data[,7:10],order.by=hours) # temperature medie orarie (°C)
EP=zoo(data[,11:14],order.by=hours) # evapotraspirato potenziale (mm/ora)
P=zoo(data[,15:18],order.by=hours) # precipitazioni (mm/ora)
Q=zoo(data[,19],order.by=hours) # portate medie orarie osservate (mc/s)

# Suddivido le serie temporali nei due periodi di uguale lunghezza
P1=window(P,end=chron(dates=(paste("31 8 2005")),times=(paste("23 00 00")),
format=c("d m y", "h m s")))
T1=window(T,end=chron(dates=(paste("31 8 2005")),times=(paste("23 00 00")),
format=c("d m y", "h m s")))
EP1=window(EP,end=chron(dates=(paste("31 8 2005")),times=(paste("23 00 00")),
format=c("d m y", "h m s")))
Q1=window(Q,end=chron(dates=(paste("31 8 2005")),times=(paste("23 00 00")),
format=c("d m y", "h m s")))

```

```

P2=window(P,start=chron(dates=(paste("1 9 2005")),times=(paste("00 00 00")),
format=c("d m y", "h m s")))
T2=window(T,start=chron(dates=(paste("1 9 2005")),times=(paste("00 00 00")),
format=c("d m y", "h m s")))
EP2=window(EP,start=chron(dates=(paste("1 9 2005")),times=(paste("00 00 00")),
format=c("d m y", "h m s")))
Q2=window(Q,start=chron(dates=(paste("1 9 2005")),times=(paste("00 00 00")),
format=c("d m y", "h m s")))

# Definisco il periodo di warmup
warm_up=365*24

# Definisco la funzione obiettivo
MSE_optim=function(param,precip,temp,potevap,runoff,area)
{simu=TUWmodel(prec=precip,airt=temp,ep=potevap,area=area,param)$q
simu[is.nan(simu)] <- -999 ## CONTROLLARE ISTRUZIONE NaN
simu=simu[-c(1:warm_up)] # rimuovo dalla serie il periodo di warmup
obse=3.6*as.numeric(runoff)/14.60
obse=obse[-c(1:warm_up)] # rimuovo dalla serie il periodo di warmup

mean((simu-obse)^2,na.rm=TRUE) # inserisco la funzione obiettivo: mean square
error
#mobs <- mean(obse,na.rm=TRUE)
}

rcal=list()
# Calibrazione sul periodo 1
bestval1=1000
for (i in 1:1)
{
  calib_MSE_1=DEoptim(fn=MSE_optim,
                    lower=c(0.9,0.0,1.0,-3.0,-2.0,0.0,0.0,0.0,0.0,2.0,30.0,1.0,0.0,0.0,0.0),
                    upper=c(1.5,5.0,3.0,1.0,2.0,1.0,600.0,20.0,2.0,30.0,250.0,100.0,8.0,30.0,50.0),
                    control=DEoptim.control(NP=NA, itermax=600, reltol=1e-4, steptol=50),
                    trace=50,

                    parallelType=1,packages="TUWmodel",parVar=c("warm_up")),
                    precip=P1,temp=T1,potevap=EP1,runoff=Q1,area=area)

  rcal[[i]]=calib_MSE_1

  if(calib_MSE_1$optim$bestval<bestval1)
  { bestmem1=calib_MSE_1$optim$bestmem
    print(bestmem1)
    bestval1=calib_MSE_1$optim$bestval
    print(bestval1)
  }

names(bestmem1)=c("SCF","DDF","Tr","Ts","Tm","LPrat","FC","BETA","k0","k1","k2","I
suz","cperc","bmax","croute")
}

# Salvo le variabili ottenute
save.image("WORKSPACEcelleH.RData")

End=Sys.time()

```

End-Start

```
# Validazioni
```

```
cal=TUWmodel(prec=P1,airt=T1,ep=EP1,area=area,param=bestmem1)  
val=TUWmodel(prec=P2,airt=T2,ep=EP2,area=area,param=bestmem1)
```

```
# Definisco la funzione per il calcolo degli indici di efficienza
```

```
MSE=function (simu,obse)  
{simu=simu[-c(1:warm_up)]  
obse=obse[-c(1:warm_up)]
```

```
# RMSE = root mean square error (mm/d)
```

```
RMSE=sqrt(mean((simu-obse)^2,na.rm=TRUE))
```

```
# NE = Nash efficiency ()
```

```
mobse=mean(obse,na.rm=TRUE)
```

```
NE=1-sum((simu-obse)^2,na.rm=TRUE)/sum((obse-mobse)^2,na.rm=TRUE)
```

```
# INE = log Nash efficiency ()
```

```
mlobse=mean(log(obse),na.rm=TRUE)
```

```
INE=1-sum((log(simu)-log(obse))^2,na.rm=TRUE)/sum((log(obse)-  
mlobse)^2,na.rm=TRUE)
```

```
# B = bias (mm/d)
```

```
B=mean(simu-obse,na.rm=TRUE)
```

```
# MAE = mean absolute error (mm/d)
```

```
MAE=mean(abs(simu-obse),na.rm=TRUE)
```

```
# MAIE = mean absolute log error (mm/d)
```

```
MAIE=exp(mean(abs(log(simu)-log(obse)),na.rm=TRUE))
```

```
# VE = volume error (%/%)
```

```
VE=(sum(simu[!is.na(obse)])-sum(obse,na.rm=TRUE))/sum(obse,na.rm=TRUE)
```

```
output=c(RMSE,NE,INE,B,MAE,MAIE,VE)
```

```
names(output)=c("RMSE (mm/d)","Nash efficiency ()","Log Nash efficiency ()","Bias  
(mm/d)",
```

```
"Mean Absolute Error (mm/d)","Mean Absolute Log Error (mm/d)","Volume  
error (%/%)")
```

```
return(output)
```

```
}
```

```
# Calcolo le efficienze
```

```
E_cal=MSE(cal$q,3.6*Q1/areatot)
```

```
E_val=MSE(val$q,3.6*Q2/areatot)
```

```
# Creo i file di output: parametri ed efficienze
```

```
write.table(bestmem1,file="Parametri_calibrazione.txt")
```

```
write.table(cal$q,file="Calibrazione.txt",row.names=FALSE,col.names=FALSE)
```

```
write.table(val$q,file="Validazione.txt",row.names=FALSE,col.names=FALSE)
```

```
write.table(E_cal,file="Efficienze_calibrazione.txt")
```

```
write.table(E_val,file="Efficienze_validazione.txt")
```

```
# Salvo le variabili ottenute
```

```
save.image("WORKSPACEcalleH.RData")
```

Script costruiti per la calibrazione e la validazione a scala oraria con portate giornaliere osservate in modalità Lumped (esempio bacino Rabbi)

```
rm(list=ls())
```

```
setwd("C:\\Users\\nemat\\Dropbox\\IO\\NeriLM_Tesi\\R\\TUWmodel_Hourly_Neri\\Lumped\\TUW_Vallemolino_Hourly_Lumped")
```

```

# Richiamo i pacchetti utili per l'elaborazione
library(zoo)
library(TUWmodel)
library(DEoptim)
library(chron)

Start=Sys.time()

# Leggo i dati relativi al bacino di interesse
data=read.table("VallemolinoH.txt",sep=" ",header=FALSE,colClasses=rep("numeric",1
8))
dataQD=read.table("VallemolinoQD.txt",sep=" ",header=FALSE,colClasses=rep("numer
ic",4))
parea=c(5.58, 16.74, 3.16, 0.40)
area=25.88 #(kmq)

# Creo le serie temporali
hours=chron(dates=(paste(data[,4], data[,5],
data[,6])),times=(paste(data[,1],data[,2],data[,3])), format=c("d m y", "h m s"))
hours1=hours[1:84744]
hours2=hours[84745:166560]

days=as.Date(strptime(paste(dataQD[,1], dataQD[,2], dataQD[,3]), format="%d %m
%Y"))

T=zoo(data[,7:10],order.by=hours) # temperature medie orarie (°C)
EP=zoo(data[,11:14],order.by=hours) # evapotraspirato potenziale (mm/ora)
P=zoo(data[,15:18],order.by=hours) # precipitazioni (mm/ora)
Q=zoo(dataQD[,4],order.by=days) # portate medie orarie osservate (mc/s)

# Suddivido le serie temporali nei due periodi di uguale lunghezza
P1=window(P,end=chron(dates=(paste("31 8 2005")),times=(paste("23 00 00")),
format=c("d m y", "h m s")))
T1=window(T,end=chron(dates=(paste("31 8 2005")),times=(paste("23 00 00")),
format=c("d m y", "h m s")))
EP1=window(EP,end=chron(dates=(paste("31 8 2005")),times=(paste("23 00 00")),
format=c("d m y", "h m s")))
Q1=window(Q,end=as.Date(strptime(paste("31 8 2005"),format="%d %m %Y")))

P2=window(P,start=chron(dates=(paste("1 9 2005")),times=(paste("00 00 00")),
format=c("d m y", "h m s")))
T2=window(T,start=chron(dates=(paste("1 9 2005")),times=(paste("00 00 00")),
format=c("d m y", "h m s")))
EP2=window(EP,start=chron(dates=(paste("1 9 2005")),times=(paste("00 00 00")),
format=c("d m y", "h m s")))
Q2=window(Q,start=as.Date(strptime(paste("1 9 2005"),format="%d %m %Y")))

# Aggreco Spazialmente i dati per utilizzo LUMPED
P1=apply(P1, 1, weighted.mean, w=parea)
T1=apply(T1, 1, weighted.mean, w=parea)
EP1=apply(EP1, 1, weighted.mean, w=parea)

P2=apply(P2, 1, weighted.mean, w=parea)
T2=apply(T2, 1, weighted.mean, w=parea)
EP2=apply(EP2, 1, weighted.mean, w=parea)

# Creo vettore portata per grafici

```

```

Qg=rep(NA,length(hours))
for (i in 1:length(Q))
{for (j in ((i-1)*24+1):(i*24))
{Qg[j]=Q[i]
}
}
Qg1=Qg[1:84744]
Qg2=Qg[84745:166560]

# Definisco il periodo di warmup
warm_up=365

# Definisco la funzione obiettivo
MSE_optim=function(param,precip,temp,potevap,runoff,area)
{simu=TUWmodel(prec=as.numeric(precip),airt=as.numeric(temp),ep=as.numeric(potevap),area=area,param)$q
simug=simu
dim(simug) <- c(24, length(runoff))
simug<-apply(simug,2,function(x) (mean(x,na.rm=TRUE)))
simug[is.nan(simug)] <- -999 ## CONTROLLARE ISTRUZIONE NaN
simug=simug[-c(1:warm_up)] # rimuovo dalla serie il periodo di warmup
obse=3.6*as.numeric(runoff)/area
obse=obse[-c(1:warm_up)] # rimuovo dalla serie il periodo di warmup
#sum((simug-obse)^2,na.rm=TRUE)
S=sum((simug-obse)^2,na.rm=TRUE)/500 # inserisco la funzione obiettivo: mean square error
ds=diff(sign(diff(t(simu))))
C=length(which(ds!=0))/100000
C+S
}

rcal=list()
# Calibrazione sul periodo 1
bestval1=1000
for (i in 1:1)
{
  calib_MSE_1=DEoptim(fn=MSE_optim,
                    lower=c(0.9,0.0,1.0,-3.0,-2.0,0.0,0.0,0.0,0.0,2.0,30.0,1.0,0.0,0.0,0.0),
                    upper=c(1.5,5.0,3.0,1.0,2.0,1.0,600.0,20.0,2.0,30.0,250.0,100.0,8.0,30.0,50.0),
                    control=DEoptim.control(NP=NA, itermax=200, reltol=1e-4, steptol=50,
                    trace=50,

parallelType=1,packages="TUWmodel",parVar=c("warm_up")),
                    precip=P1,temp=T1,potevap=EP1,runoff=Q1,area=area)

  rcal[[i]]=calib_MSE_1

  if(calib_MSE_1$optim$bestval<bestval1)
  { bestmem1=calib_MSE_1$optim$bestmem
    print(bestmem1)
    bestval1=calib_MSE_1$optim$bestval
    print(bestval1)
  }

names(bestmem1)=c("SCF","DDF","Tr","Ts","Tm","LPrat","FC","BETA","k0","k1","k2","I
suz","cperc","bmax","croute")

```

```

}

# Salvo le variabili ottenute
save.image("WORKSPACEvallemolinoHlumped5.RData")

End=Sys.time()
End-Start

# Validazioni
cal=TUWmodel(prec=P1,airt=T1,ep=EP1,area=area,param=bestmem1)
val=TUWmodel(prec=P2,airt=T2,ep=EP2,area=area,param=bestmem1)

calM=cal$q
dim(calM) <- c(24, length(Q1))
calM<-apply(calM,2,function(x) (mean(x,na.rm=TRUE)))

valM=val$q
dim(valM) <- c(24, length(Q2))
valM<-apply(valM,2,function(x) (mean(x,na.rm=TRUE)))

# Definisco la funzione per il calcolo degli indici di efficienza
MSE=function (simu,obse)
{simu=simu[-c(1:warm_up)]
obse=obse[-c(1:warm_up)]

# RMSE = root mean square error (mm/h)
RMSE=sqrt(mean((simu-obse)^2,na.rm=TRUE))
# NE = Nash efficiency ()
mobse=mean(obse,na.rm=TRUE)
NE=1-sum((simu-obse)^2,na.rm=TRUE)/sum((obse-mobse)^2,na.rm=TRUE)
# INE = log Nash efficiency ()
mlobse=mean(log(obse),na.rm=TRUE)
INE=1-sum((log(simu)-log(obse))^2,na.rm=TRUE)/sum((log(obse)-
mlobse)^2,na.rm=TRUE)
# B = bias (mm/h)
B=mean(simu-obse,na.rm=TRUE)
# MAE = mean absolute error (mm/h)
MAE=mean(abs(simu-obse),na.rm=TRUE)
# MAIE = mean absolute log error (mm/h)
MAIE=exp(mean(abs(log(simu)-log(obse)),na.rm=TRUE))
# VE = volume error (%/%)
VE=(sum(simu[!is.na(obse)])-sum(obse,na.rm=TRUE))/sum(obse,na.rm=TRUE)

output=c(RMSE,NE,INE,B,MAE,MAIE,VE)
names(output)=c("RMSE (mm/h)","Nash efficiency ()","Log Nash efficiency ()","Bias
(mm/h)",
"Mean Absolute Error (mm/h)","Mean Absolute Log Error
(mm/h)","Volume error (%/%)")
return(output)
}

# Calcolo le efficienze
E_cal=MSE(calM,3.6*Q1/area)
E_val=MSE(valM,3.6*Q2/area)

# Creo i file di output: parametri ed efficienze
write.table(bestmem1,file="Parametri_calibrazione.txt")
write.table(t(calM),file="Calibrazione.txt",row.names=FALSE,col.names=FALSE)

```

```
write.table(t(val$q),file="Validazione.txt",row.names=FALSE,col.names=FALSE)
write.table(E_cal,file="Efficienze_calibrazione.txt")
write.table(E_val,file="Efficienze_validazione.txt")
```

```
# Salvo le variabili ottenute
save.image("WORKSPACEvallemolinoHlumped4.RData")
```

Ringraziamenti

Desidero porgere un particolare ringraziamento alla prof. Elena Toth che mi ha seguito per la stesura di questo elaborato, fornendo continua disponibilità, supporto e incoraggiamento e generando in me ancor più interesse nella materia grazie alla sua precisione e professionalità.

Ringrazio inoltre i miei genitori, mio fratello, la mia ragazza e tutti miei amici per il supporto che direttamente e indirettamente mi danno tutti i giorni.