Dipartimento di Matematica e Informatica

Denise Cilia

X81000791

Progetto Data Mining – Classificatore di sottotipi e stadi di BC (*Breast Cancer*)

1. Introduzione

L'idea alla base del progetto assegnato è individuare, tra i classificatori studiati ovvero CART, randomForest, Naive Bayes e SVM, un classificatore in grado di distinguere, dato in input un certo dataset di pazienti, sottotipo e stadio del particolare caso di studio: il tumore al seno.

Lavoreremo con una serie di campioni tumorali riguardanti 1082 pazienti[1].

In particolare:

- un dataframe *brca_clinical* contenente nelle colonne id del paziente, sottotipo tumorale e stadio;
- una matrice di espressione *brca_expressions*, dove il generico elemento di indice (*i*,*j*) indica l'espressione numerica di un particolare gene per il paziente *j*;
- una particolare rete indicata come *biological pathway*[2] formata da un insieme di nodi, i quali contengono id, nome e alias dei vari geni, e un insieme di archi etichettati con funzioni di *inhibition* o *activation*, *expression*, *repression*, a seconda della connessione definita tra i nodi;
- infine l'algoritmo *MultiMotif* che, una volta costruite le reti per ciascuno dei pazienti presenti nel *brca_clinical*, conta tutti i possibili sottografi per ciascuna rete. Da questo risultato, verrà costruito un vettore di feature di cui potrà servirsi il classificatore.

Come primo step, i dati in input subiscono una fase di filtro. In particolare bisogna mantenere all'interno dei nodi della *pathway* solamente i geni che sono presenti anche nella matrice di espressione. Di conseguenza verranno mantenuti solo gli archi che coinvolgono tali nodi. Successivamente si passa alla costruzione delle reti per ciascun paziente presente nel *brca_clinical*. L'algoritmo effettua un ciclo che visualizza tutte le colonne con le varie espressioni dei geni, che verranno filtrati come *sovraespressi* se la loro espressione è maggiore o uguale a 1, *sottoespressi* se la loro espressione risulta minore o uguale a -1. Tutti i geni compresi nell'intervallo (-1,1) verranno indicati come *non significativamente espressi* e quindi rappresentati dal valore *NA*. Fissata la *j*-esima colonna, con un secondo ciclo sull'indice *i* si tiene conto del valore e dell'id del gene, informazione che verrà inserita all'interno di un insieme temporaneo da intersecare con *filtered_nodes* e l'insieme *edge*, in modo tale da estrarre per ogni paziente solo i nodi effettivamente presenti per quel paziente e le relative connessioni descritte dagli archi. Il risultato darà in output un file .txt

1 Fonte: https://www.cancer.gov/about-nci/organization/ccg/research/structural-genomics/tcga

2 Una **biological pathway** è descritta come una serie di interazioni tra le varie molecole presenti un una cella che può portare ad un certo prodotto o un cambiamento nella cella. Possono influenzare la creazione di nuove molecole e possono anche essere responsabili dell'attivazione o disattivazione di un gene. Le biological pathway più comuni sono coinvolte nel metabolismo, la regolazione dell'espressione di un gene, etc. (*Wikipedia*)

per ciascun paziente che rispetta il seguente formato: *idpaziente*.txt. (*I passi appena descritti vengono illustrati nella Figura 1.*).
L'output sarà un file che rispetterà la struttura richiesta in input dall'algoritmo

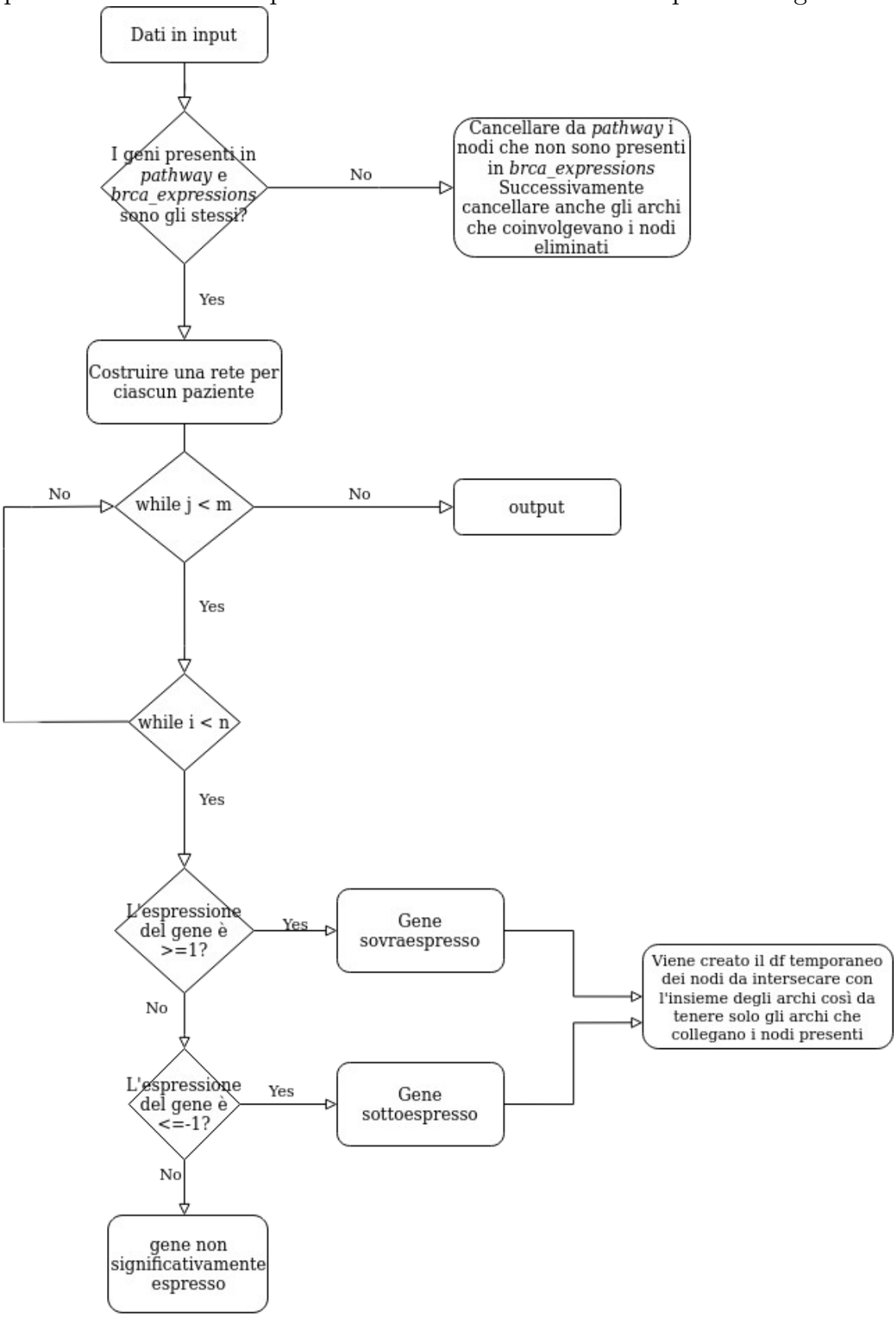

Figura 1. Diagramma di flusso

MultiMotif. Dunque, la prima riga definisce l'orientamento del grafo, se direzionato o no. La seconda riga contiene il numero *N* di nodi presenti all'interno del grafo i cui indici partono da *0* fino ad arrivare a *N*-1. Nelle successive *N* righe vengono espresse le etichette dei corrispettivi nodi e infine nelle righe successive troviamo la lista degli archi che collegano i nodi presenti nella rete. Il primo numero è l'indice del nodo sorgente, il secondo numero è l'indice del nodo di destinazione. Il terzo campo è una lista di etichette per archi. Un esempio è dato dalla *Figura 2.*

```
directed
3351
sottoespresso
sottoespresso
sottoespresso
sovraespresso
sottoespresso
sottoespresso
sottoespresso
sottoespresso
sovraespresso
sottoespresso
[...]
sottoespresso
sottoespresso
sovraespresso
sottoespresso
0 102 INHIBITION
0 181 ACTIVATION
0 182 ACTIVATION
0 1131 ACTIVATION
0 1571 ACTIVATION
```

Figura 2. Esempio di *TCGA-3C-AAAU.txt*

2. Applicazione *MultiMotif*

Generati i file .txt corrispondenti alle reti dei 1082 pazienti presenti nello studio, daremo in input ciascun file all'algoritmo *MultiMotif*. Quest'ultimo avrà il compito di calcolare tutti i conteggi relativi alle triple presenti nella rete del singolo paziente considerando le possibili etichettature degli archi che collegano i nodi.

	Motif_Nodes	Motif_edges	Num_occ_input_graph
1	sovraespresso,sottoespresso,sottoespresso	(0,2,INHIBITION,ACTIVATION)(0,1,ACTIVATION)(1,0,I...	2
2	sovraespresso,sottoespresso,sottoespresso	(0,2,INHIBITION,ACTIVATION)(0,1,ACTIVATION)(1,0,I...	1
3	sovraespresso,sovraespresso,sottoespresso	(2,0,INHIBITION)(2,1,REPRESSION)	8
4	sovraespresso,sovraespresso,sottoespresso	(0,2,INHIBITION)(2,1,REPRESSION)	4
5	sovraespresso,sovraespresso,sottoespresso	(0,2,INHIBITION)(1,2,EXPRESSION,ACTIVATION)	2
6	sottoespresso,sottoespresso,sottoespresso	(0,2,INHIBITION,ACTIVATION)(1,2,INHIBITION,ACTIVA...	8
7	sottoespresso,sottoespresso,sottoespresso	(0,2,INHIBITION,ACTIVATION)(1,2,ACTIVATION)(2,0,I...	59
8	sottoespresso,sottoespresso,sottoespresso	(0,2,ACTIVATION)(1,2,ACTIVATION)(2,0,INHIBITION)	159
9	sottoespresso,sovraespresso,sottoespresso	(0,2,ACTIVATION)(0,1,INHIBITION,ACTIVATION)(1,0,I...	1
10	sottoespresso,sovraespresso,sottoespresso	(0,2,ACTIVATION)(0,1,INHIBITION,ACTIVATION)(1,0,A...	1
11	sottoespresso,sottoespresso,sottoespresso	(0,2,ACTIVATION)(1,2,INHIBITION,ACTIVATION)(2,0,I...	8
12	sottoespresso,sottoespresso,sottoespresso	(0,2,INHIBITION,ACTIVATION)(0,1,INHIBITION,ACTIVA...	2
13	sottoespresso,sottoespresso,sottoespresso	(0,2,INHIBITION,ACTIVATION)(0,1,ACTIVATION)(1,2,I...	8
14	sottoespresso,sottoespresso,sottoespresso	(0,2,ACTIVATION)(0,1,ACTIVATION)(1,2,INHIBITION)	168
15	sottoespresso,sottoespresso,sottoespresso	(0,2,ACTIVATION)(0,1,INHIBITION,ACTIVATION)(1,2,I...	3

Figura 3. Esempio di output dell'algoritmo MultiMotif su *TCGA-3C-AAAU.txt*

Terminata l'esecuzione avremo ottenuto in output tutti i conteggi sulla quale costruire il singolo vettore di features con cui potremo allenare i vari classificatori.

3. Risultati

Prima di concentrarci sullo studio del classificatore che fornisce la migliore accuratezza nel caso dell'etichetta *sottotipo tumorale* e dell'etichetta *stadio*, bisogna costruire il vettore di features, che conterrà al suo interno tutti i conteggi riferiti a specifici motivi, per un determinato paziente. Costruiamo, dunque, un primo dataframe che verrà denominato *motifset*. Esso conterrà tutti i possibili motivi, individuati e calcolati per ciascun paziente. È caratterizzato da tre colonne: *Motif_Nodes* indica i nodi, *Motif_edges* indica gli archi etichettati che collegano i nodi, e infine una colonna *Id* che servirà da identificativo univoco per riferirsi ai particolari motivi. Successivamente costruiamo il vettore delle features, denominato *breast_cancer_df*. Sarà formato da 1082 righe, una per ciascun paziente, e da tante colonne quanti sono i motivi trovati, per un totale di 16249 colonne, ciascuna caratterizzata dall'*Id* del motivo. In generale, l'elemento (*i*,*j*) all'interno del dataframe corrisponde al conteggio del particolare motivo *j* per il paziente *i*. È bene sottolineare che i pazienti presentano reti diverse, dunque un paziente potrebbe possedere un motivo e presentare un conteggio, mentre un altro paziente no. Basandoci sull'ultima osservazione, viene effettuata una pulizia del dataframe che sostituisce 0 a tutti i valori *NA*. Verranno aggiunte a *breast_cancer_df* due ulteriori colonne che rappresentano il sottotipo tumorale e lo stadio per il paziente *i*. Allo stesso modo descritto precedentemente, vengono sostituiti con 0 tutti i valori *NA* presenti nelle due colonne etichette. Nel particolare, per l'etichetta *STAGE* si procede con un'ulteriore pulizia che permette di raggruppare i vari stadi concentrandoci solo su *STAGE I, STAGE II*, ..., etc. anziché *STAGE IA, STAGE IB, STAGE IC, STAGE IIA*, ..., e così via.

	Motivo1	Motivo2	Motivo3	Motivo4	Motivo5	Motivo6	Motivo7	Motivo8	Motivo9
1	2	1	8	4	2	8	59	159	1
2	0	0	30	6	0	0	17	182	0
3	2	2	30	8	0	4	24	94	2
4	0	0	24	8	0	0	8	54	0
5	1	1	30	8	1	0	13	74	0
6	1	1	12	4	0	8	41	71	1
7	0	0	12	3	0	0	0	269	0
8	0	0	2	0	0	0	0	153	0
9	0	0	0	0	0	0	12	30	0
10	1	1	0	0	2	0	7	61	0
11	2	2	12	0	0	11	58	128	2
12	0	0	16	6	0	0	15	73	0
13	1	1	7	2	0	0	9	17	0
14	0	0	12	4	0	0	5	83	0
15	0	0	15	3	0	0	14	53	0

Figura 4. Prime righe e colonne di *breast_cancer_df*

Concludiamo con l'ultimo passaggio: suddiviamo *breast_cancer_df* in modo da avere una porzione pari al 75% di training set, e 25% test set. Tramite training set alleniamo i classificatori CART, randomForest, SVM e Naive Bayes sulle due etichette *AJCC_PATHOLOGIC* e *STAGE*. Utilizzeremo il framework *caret* che tramite metodo di *cross-validation* trova i parametri ottimali per i vali modelli di classificazione, in grado di massimizzare l'accuratezza. I risultati della predizione vengono forniti dalla matrice di confusione seguiti dai parametri statistici che individuano sensitività, specificità, etc. per ciascuna etichetta. Otteniamo nel caso dell'etichetta *AJCC_PATHOLOGIC* (*sottotipo tumorale*) un'accuratezza del 65% con randomForest, nel caso dell'etichetta *STAGE* otteniamo invece un'accuratezza del 57% con Naive Bayes.

	Sottotipo	Stage Non raggruppato	Stage Raggruppato
randomForest	**0,6519**	0,1815	0,5593
naiveBayes	0,4889	0,0333	**0,5704**
SVM	0,5704	0,2333	0,5593
CART	0,5556	0,237	0,5593

Figura 5. Risultati finali

4. Possibili miglioramenti

Alla luce dei risultati ottenuti, sarebbe possibile incrementare l'accuratezza rimuovendo dal caso di studio tutti i pazienti che non presentano valori nel campo *sottotipo tumorale* o nel campo *stage* così da ridurre il numero di campioni osservato. Quest'ultimo passaggio permetterebbe, inoltre, di eliminare il livello nullo sia nel caso di *AJCC_PATHOLOGIC* che nel caso di *STAGE*. Infine, per migliorare le performarce sulla predizione del campo *STAGE* potrebbe essere utile filtrare il dataframe *breast_cancer_df* eliminando le colonne che contengono un numero di valori "0" superiore ad una certa soglia fissata. Questa operazione, dovrebbe ridurre il rumore presente nel dataset, e quindi migliorare il risultato finale.