Alma Mater Studiorum · Università di Bologna

Scuola di Scienze

Corso di Laurea in Fisica

Applicazione del metodo QDanet_PRO alla classificazione di dati omici

Relatore:
Prof. Daniel Remondini

Presentata da:
Elisa Berti

Correlatore:
Dott. Giuseppe Levi

Sessione II
Anno Accademico 2014/2015

Indice

Sommario

Il presente lavoro di tesi si pone nell'ambito dell'analisi dati attraverso un metodo (QDanet_PRO), elaborato dal Prof. Remondini in collaborazine coi Dott. Levi e Malagoli, basato sull'analisi discriminate a coppie e sulla Teoria dei Network, che ha come obiettivo la classificazione di dati contenuti in dataset dove il numero di campioni è molto ridotto rispetto al numero di variabili.

Attraverso questo studio si vogliono identificare delle *signature*, ovvero un'insieme ridotto di variabili che siano in grado di classificare correttamente i campioni in base al comportamento delle variabili stesse.

I dati che verranno analizzati sono di tipo omico (quantificazione dell'espressione di miRNA e proteine), e i campioni in questione sono individui classificati in due gruppi relativi alla sopravvivenza o meno dell'individuo.

Per poter testare il metodo vengono utilizzati gli stessi dati usati nello studio pubblicato su Nature dal titolo *Assessing the clinical utility of cancer genomic and proteomic data across tumor types* (Yuan Yuan et al., 2014), dove vengono trattate diverse tipologie di tumore analizzando i vari dataset e provando delle loro combinazioni.

Nel presente lavoro ci si limita allo studio dei dataset che si riferiscono ad un solo tipo di tumore, il glioblastoma (GBM), analizzando due diversi tipi di dati, separatamente e in congiunzione, per valutarne l'efficacia nella classificazione dei campioni.

L'elaborazione dei diversi dataset, che sono pubblicati sul sito *www.synapse.org*, avviene attraverso diverse fasi; si comincia con una un'analisi discriminante

a coppie per identificare le performance di ogni coppia di variabili per poi passare alla ricerca delle coppie più performanti attraverso un processo che combina la Teoria dei Network con la *Cross Validation*. Una volta ottenuta la *signature* si conclude l'elaborazione con una validazione per avere un'analisi quantitativa del successo o meno del metodo.

Capitolo 1

Metodi di classificazione

I dati omici, che si dividono in genomici e proteomici, sono caratterizzati da un elevato numero di variabili (fino a 10^4) ma da uno scarso numero di campioni; questo risulta essere un problema per analisi di tipo statistico in cui solitamente il numero di campioni è molto più elevato di quelle delle variabili. Per poter risolvere questa problematica i dati vengono analizzati tramite l'analisi discriminante e la *Cross Validation* che portano alla determinazione di una *signture* a bassa dimensionalità.

La bassa dimensionalità delle *signature* è di rilevante importanza per poter dare un reale supporto alle diagnosi mediche, in quanto l'obiettivo è quello di poter fare previsioni a seguito di semplici analisi preliminari. Oltre a questa caratteristica le *signature* individuate dovranno essere altamente performanti, in modo da garantire una corretta predizione.

Sull'efficacia di questi tipi di analisi predittive emergono principalmente due filoni: uno a sostegno dell'ipotesi che queste *signature* rappresentino un buon modello di classificazione dei dati, mentre l'altro afferma la limitatezza di questo metodo e quindi la sua insufficienza predittiva.

Considerando le signature come un buon modello predittivo esistono diversi tipi di apprendimento:

- *unsupervised learning*, dove la *machine learning* è "libera" di trovare una struttura all'interno del dataset, questo tipo di apprendimento

risulta utile per l'identificazione di nuove classi, che nel presente caso possono essere viste come diverse tipologie di tumori;

- *supervised learning*, la *machine learning* segue uno schema predefinito per la classificazione dei dati in input e conosce le classi di output che si possono presentare; questo tipo di *machine learning* viene usato nella classificazione di classi già note.

All'interno di questo capitolo verranno esposte le principali strutture usate nella classificazione a partire dalla Teoria dei Network, fino all'analisi discriminante.

1.1 Teoria dei Network

La Teoria dei Network si basa sulle interazione presenti tra gli elementi di un sistema. Un network viene graficato tramite nodi, che identificano gli elementi del sistema e linee (*link*), che rappresentano le connessioni tra i vari elementi. Un altro possibile metodo di rappresentazione di un network è una matrice di adiacenza $N \times N$, dove N è il numero di elementi e ogni suo elemento A_ij rappresenta il livello di connessione tra l'elemento i e l'elemento j.

Grazie al metodo grafico viene ben evidenziato il *grado di connettività* K, ovvero il numero di *link* entranti o uscenti da ogni singolo nodo; nella rappresentazione matriciale il grado di connettività corrisponde alla somma del numero di righe o colonne della matrice di adiacenza.

Il grado di connettività rappresenta l'importanza di un certo nodo all'interno del network e corrisponde al numero di *link* che sarebbe necessario rimuovere per rendere isolato tale nodo.

Un'altra misura di centralità di un nodo è la *closeness* Cl, definita come l'inverso della distanza media di un nodo da tutti gli altri nodi

$$Cl_i = \frac{N-1}{\sum_j P_{ij}} \tag{1.1}$$

con P_{ij} distanza di un nodo dagli altri nodi. La *closeness* mette in evidenza come la vicinanza tra nodi renda questi più rilevanti.

All'interno di un network ci possono essere nodi che risultano non connessi, ovvero a distanza infinita, appartenenti quindi a sottografi, che prendono il nome di *componenti connesse del grafo*; all'interno di questi sottografi ogni nodo è connesso agli altri nodi del sottografo stesso ma mai con nodi appartenenti ad altre componenti connesse del grafo.

Le componenti connesse del grafo corrispondono esattamente alle *signature* che saranno ricercate durante l'applicazione del metodo QDanet_PRO; attraverso una successiva elaborazione verranno eliminati, se presenti, i nodi con grado di connettività 1, in modo da ridurre la dimensionalità della *signature*.

Infine un'ultima misura di centralità di un nodo è data dalla *betweenness centrality*, la quale è definita come il numero di shortest path passanti per il nodo i-esimo, quindi un nodo con bassa connettività può avere grande *betweenness centrality*, come nel caso di nodi che collegano due blocchi, dove il flusso di interazioni del network è elevato.

$$BC_i = \frac{\sum_{m,n} P_{m,n}(i)}{\sum_{m,n} P_{m,n}} \tag{1.2}$$

1.2 Machine Learning

Le *machine learning* sono macchine che aumentano il loro potere predittivo ad ogni iterazione. Due dei principali esempi di *machine learning* sono: le Reti Neurali e la *Support Vector Machine*.

1.2.1 Reti Neurali

La Rete Neurale è un modello di regressione o classificazione a due fasi attraverso metodi statistici non lineari; uno tra i più semplici e usati è il *Single layer perceptron*.

Il *Single layer perceptron* è un classificatore binario che si basa sulla combinazione lineare pesata delle variabili in ingresso comandata dalla fun-

zione di attivazione, tipicamente scelta tra la funzione sigmoidea e le funzioni gaussiane a base radiale.

Lo scopo di una Rete Neurale è quello di migliorare il proprio output attraverso un processo iterativo che porta alla determinazione dei pesi delle varie variabili, in modo da poter generalizzare i risultati. Il processo di apprendimento avviene tipicamente tramite un training set di variabili in ingresso, le quali vengono raggruppate per identificare le *signature* rappresentative dei dati stessi, attraverso l'utilizzo di metodi probabilistici. Utilizzando la *Single layer perceptron* si ottiene una soluzione solo se il problema è linearmente separabile.

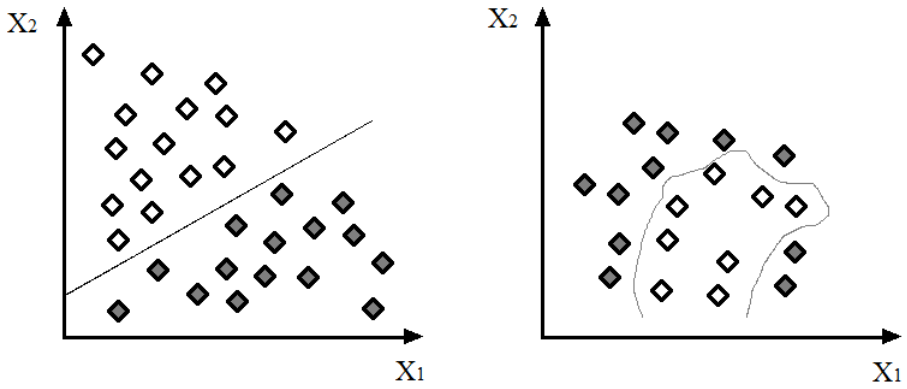

Figura 1.1: Classi linearmente separabili e classi non linearmente separabili

Nel *layer perceptron* multilivello sono presenti uno o più livelli intermedi nascosti (*hidden layer*) tra le variabili di input e l'output; questi passaggi intermedi fanno sì che gli output dei nodi di un livello siano gli input del livello successivo ed ogni nodo si comporta come un *Single layer perceptron*.

Gli *hidden layer* possono essere pensati come un'espansione degli input, partendo da queste espansioni come input il modello diventa lineare standard. Nel caso in cui la funzione di attivazione fosse la funzione identità si avrebbe un metodo lineare a partire già dalle variabili di input.

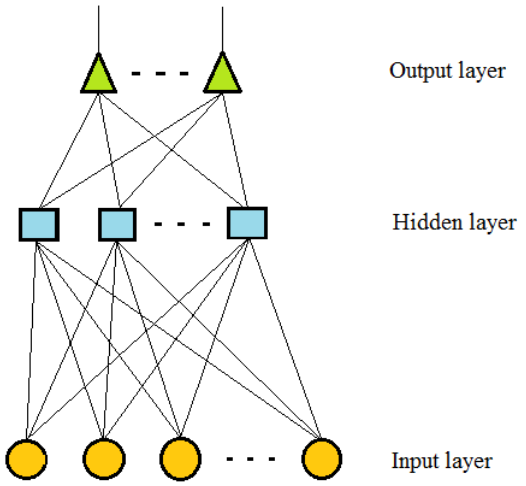

Figura 1.2: Rete Neurale a layer nascosto

Back Propagation

Il processo che porta alla determinazione dei pesi delle varie variabili non è privo di errore. Si cerca di minimizzare questo errore attraverso la discesa del gradiente ($Back - Propagation$), ovvero un aggiornamento continuo dei pesi iterazione dopo iterazione.

Il processo di $Back - Propagation$ è diviso in due fasi: la prima in cui i pesi ottenuti vengono usati per il calcolo degli output ($forward - phase$), passando dal livello $k-$esimo al livello $k+1-$esimo; ; la seconda in cui i pesi del livello $k+1-$esimo esimo vengono aggiornati prima di quelli del livello $k-$esimo, in questo modo gli errori al livello $k+1-$esimo vengono utilizzati per stimare gli errori al livello $k-$esimo ($backward - phase$).

I livelli su cui si applica questo algoritmo sono gli *hidden layer*, dei quali non si conosce il valore desiderato di uscita a differenza del valore di output.

1.2.2 Support Vector Machine

La *Support Vector Machine* è un classificatore discriminante che affronta il problema della separazione di classi differenti.

Un analisi duale del problema permette di generalizzare il procedimento per classi non linearmente separabili, i cui elementi si sovrappongono. Il problema viene risolto trasformando lo spazio in cui si trovano le classi in modo da costruire un confine (*boundary lineare*), ovvero dato un training set viene restituito l'iperpiano ottimale in grado si classificare i dati.

L'iperpiano ottimale che separa le coppie di dati del training set $(x_1, y_2) \ldots (x_N, y_N)$ con $x_i \in R^p$ e $y_i \in \{ -1, 1\}$ risulta essere il seguente:

$$\{ x : f(x) = x^T\beta + \beta_0 = 0\} \tag{1.3}$$

Tale iperpiano viene calcolato a partire della massimizzazione del margine $M = 1/||\beta||$ (con β vettore normalizzato unitario), dove con margine si intende la massima distanza tra i punti più vicini delle due classi. L'iperpiano ottimale risulta quindi essere quello con il più grande margine tra i punti del piano (*support vectors*) di classe -1 e quelli di classe 1.

Il problema di ottimizzazione dell'iperpiano corrisponde al problema di massimizzazione del margine, viene quindi posto come:

$$min_{\beta,\beta_0} M \tag{1.4}$$

$$y_i(x_i^T\beta + \beta_0) \geq 1, i = 1, \ldots, N \tag{1.5}$$

Considerando classi non linearmente separabili, dove i dati delle due classi si sovrappongono non è facile trovare il miglior iperpiano, quindi è necessario trovare un compromesso; a tal proposito viene definita la variabile di scarto (*slack variable*) ε che modifica il problema in:

$$y_i(x_i^T\beta + \beta_0) \geq M(\varepsilon_i) \tag{1.6}$$

Se $0 \leq \varepsilon \leq 1$ il punto si trova all'interno del margine, mentre per $\varepsilon \geq 1$ si ha una classificazione errata del punto.

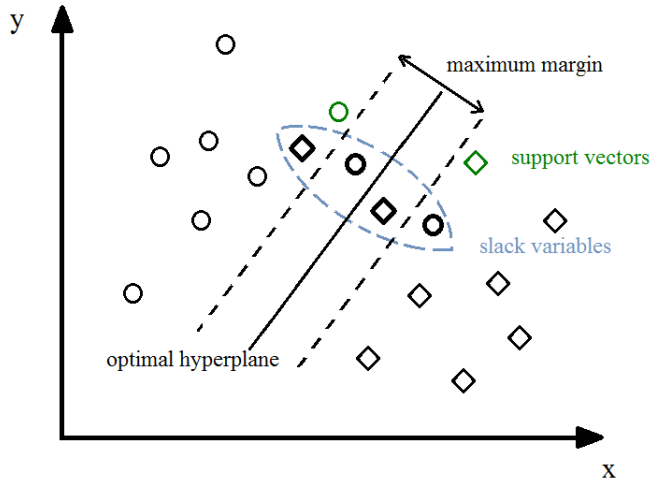

Figura 1.3: *Support Vector Machine* nel caso di due classi non separabili

Il problema da risolvere risulta un problema quadratico con condizioni di disuguaglianza lineare, ovvero un problema di ottimizzazione convessa che grazie al formalismo duale e ai moltiplicatori di Lagrange dà il seguente risultato:

$$\hat{\beta} = \sum_{i=1}^{N} \hat{\alpha}_i y_i x_i \tag{1.7}$$

con $\hat{\alpha}_i$ *vettori di supporto*. Dopo aver ricavato le soluzioni $\hat{\beta}_0$ e $\hat{\beta}$ la funzione di decisione dell'iperpiano risulta essere:

$$\hat{G}(x) = sing[x^T \hat{\beta} + \hat{\beta}_0] \tag{1.8}$$

Partendo da questo metodo, che restituisce un iperpiano lineare, è possibile allargare lo spazio degli oggetti con espansioni polinomiali; lavorando con spazi allargati, di dimensione molto grande o anche infinita, la separazione tra le classi risulta migliorata e il *boundary* che risulta lineare nello spazio allargato non lo sarà più nello spazio di partenza.

All'interno di questi spazi allargati è necessario definire le funzioni base $h_m(x)$, $m = 1, \ldots, M$ che permettono l'adattamento delle variabili in input $h(x_i) = (h_1(x_i), h_2(x_i), \ldots, h_M(x_i))$ con $i = 1, \ldots, N$. Questa trasformazione

porta alla definizione di una funzione non lineare per l'iperpiano:$\hat{f}(x) = h(x)^T\hat{\beta} + \hat{\beta}_0$ e dà come classificatore $\hat{G}(x) = sing(\hat{f}(x))$.

Il risultato appena ottenuto può essere esteso a molte classi, ripetendo il procedimento per due classi alla volta, arrivando ad ottenere un unico classificatore finale, il più importante.

1.3 Analisi Discriminante

Lo scopo di questa analisi è quello di trovare un discriminante che permetta di classificare i vari campioni in base al comportamento delle varie variabili.

Nel presente caso di dati ad alta dimensionalità, siamo in presenza di matrici di dati X di dimensione $N \times P$, con $N \gg P$ e $N = numero\ di\ variabili$ e $P = numero\ di\ campioni$, ovvero dal punto di vista dell'analisi discriminante siamo in presenza di P campioni con osservazioni x di dimensionalità N che vengono combinati linearmente $y = w^T x$ tramite un vettore generico w e suddivisi in classi.

L'obiettivo si riduce quindi alla determinazione di w che massimizza la distanza tra le varie classi.

Come prima operazione viene identificata la mediana per ogni classe:

$$m_i = \frac{1}{n_i} i \sum_{x \in D} x \tag{1.9}$$

con D insieme dei sample. La proiezione di ogni classe risulta quindi:

$$\tilde{m}_i = \frac{1}{n_i} \sum_{y \in Y} y = \frac{1}{n_i} \sum_{y \in Y} w^T x = w^t m_i \tag{1.10}$$

con Y insieme delle classi.

In seguito si stima la distanza tra le classi attraverso la differenza delle medie:

$$|\tilde{m}_1 - \tilde{m}_2| = |w^t(m_1 - m_2)| \tag{1.11}$$

Massimizzare la distanza tra le classi significa quindi massimizzare $|w^t(m_1 - m_2)|$. Questa massimizzazione è necessario farla rispetto alla varianza che viene stiamata come:

$$\tilde{s}_i^2 = \sum_{y \in Y} (y - \tilde{m}_i)^2 \tag{1.12}$$

$$\frac{1}{n}(\tilde{s}_1^2 + \tilde{s}_i^2) \tag{1.13}$$

L'analisi del discrimiante si traduce quindi nella determinazione del massimo della funzione $J(w)$:

$$J(w) = \frac{|\tilde{m}_1 - \tilde{m}_2|^2}{\tilde{s}_1^2 + \tilde{s}_i^2} \tag{1.14}$$

La dipendenza di $J(w)$ da w viene esplicitata introducendo la matrice di varianza tra le classi S_B (*between*) e la matrice di varianza interna alle classi S_W (*within*):

$$S_B = (m_1 - m_2)(m_1 - m_2)^t \tag{1.15}$$

$$S_W = S_1 + S_2 \tag{1.16}$$

$$S_i = \sum_{x \in D} (x - m_i)(x - m_i)^t \tag{1.17}$$

A seguito di queste definizioni la funzione $J(w)$ prende la forma di:

$$J(w) = \frac{w^t S_B w}{w^t S_W w} \tag{1.18}$$

quindi per avere la massimizzazione di tale funzione il vettore w deve soddisfare la condizione di $S_W^{-1} S_B w = \lambda w$, dove λ risulta essere l'autovalore e w il rispettivo autovettore.

La variabile del discriminante che massimizza la distanza tra le varie classi è definita a partire dall'autovettore w_{max} che corrisponde al massimo autovalore λ_{max}:

$$y_{max} = w_{max} x \tag{1.19}$$

L'autovalore λ_{max} prende il nome di potere discriminante di y e ne indica la capacità di separare i campioni.

Si osserva che è possibile determinare tante funzioni discriminanti quanti sono gli autovalori non nulli della matrice $S_W^{-1} S_B$.

Grazie a questa analisi la classe di appartenenza di una data osservazione è quella con il vettore del valor medio più vicino all'osservazione, nello spazio delle variabili del discriminante.

1.4 K-fold Cross Validation

I metodi di *Cross Validation* hanno l'obiettivo di stimare i possibili errori che si possono commettere nel processo di classificazione tramite algoritmi di apprendimento. Uno tra questi metodi è il $K - fold\ Cross\ Validation$ che prevede la divisione dei dati in k partizioni dove $k-1$ di queste partizioni vengono usate per impostare il modello e la restante partizione per testarlo; questo comporta che a seguito dei vari cicli viene usata la totalità dei dati sia per creare il modello che per terstarlo.

Viene definita una funzione per identificare la partizione che si sta considerando e una funzione fitted $\hat{f}^{(-k)}(x)$ calcolata rimuovendo la k-esima parte dei dati.

La stima dell'errore sulla predizione fatta risulta essere:

$$CV(\hat{f}) = 1/N \sum_{i=1}^{N} L(y_i, \hat{f}^{-\sigma(i)}(x_i)) \tag{1.20}$$

dove $L(y_i, \hat{f}^{-\sigma(i)}(x_i)$ è detta funzione di perdita per la misurazione degli errori tra le variabile y_i e $f^{-\sigma(i)}(x_i)$. L'errore complessivo è la media tra le diverse stime:

$$E = 1/k \sum_{i=1}^{N} E_i \tag{1.21}$$

Considerando invece un set di modelli indicizzati dal parametro alfa $f(x, \alpha)$ viene definita la funzione $\hat{f}^{(-k)}(x, \alpha)$ che porta alla definizione di:

$$CV(\hat{f}, \alpha) = 1/N \sum_{i=1}^{N} L(y_i, \hat{f}^{-\sigma(i)}(x_i, \alpha)) \tag{1.22}$$

Grazie a questa funzione è possibile determinare il parametro $\hat{\alpha}$ che minimizza la stima della curva d'errore. Il modello finale sarà quindi $f(x, \hat{\alpha})$.

1.5 Metodo QDanet_PRO

Il metodo QDanet_PRO che viene utilizzato per l'identificazione delle *signature* è stato pensato per l'elaborazione di dataset con un elevato numero di variabili, quindi sfrutta tipologie di classificazione quali l'analisi discriminante e la $Cross\ Validation$, per concludere con una validazione del modello di classificazione creato.

Proprio per poter permettere l'operazione di validazione il dataset analizzato viene diviso in due parti:

- *training set* (70% dei campioni), utilizzato per la creazione del modello di classificazione;
- *test set* (30% dei campioni), utilizzato per la validazione del modello.

Come prima operazione il dataset, sotto forma di matrice $N \times M$, contenente le *label* nella prima riga, i campioni nelle colonne e le variabili nelle righe, viene elaborato attraverso una prima parte del metodo, scritta in C++, che svolge le seguenti funzioni:

- lettura dei dati dal file di testo in input;
- elaborazione dei dati: determinazione della posizione media dei gruppi di *label* e la classificazione delle coppie di variabili in base alla loro posizione, il tutto all'interno del piano determinato dall'espressione genica;
- creazione del file di output.

Dopo l'avvenuta lettura dei dati si passa alla loro elaborazione, con il calcolo del discriminante, ovvero viene calcolata la distanza della coppia da una classe e dall'altra. A seguito di questa operazione viene aumentato lo *score* della classe a cui la coppia risulta più vicina. L'aumento dello *score* rappresenta la performance di classificazione della coppia di variabili in questione. Questa risuta essere la parte più onerosa dell'algoritmo in quando il

procedimento appena descritto viene ripetuto per un totale di volte pari a: $(numero\ variabili) + (numero\ variabili) \times (numero\ variabili - 1)/2$.

I risultati così ottenuti vengono poi trascritti sul file di output che risulta essere anch'esso un file di testo.

Per ottenere le *signature* più performanti è necessaria un'ulteriore analisi che viene fatta grazie agli algoritmi $read\_QDanetPro$ e qda_res_comp compilabili sul software MATLAB.

1.5.1 Analisi del network con MATLAB

La seconda parte dell'analisi dei dati viene svolta in MATLAB perché permette una programmazione snella e l'utilizzo di algoritmi utili già implementati all'interno del software; inoltre i calcoli ancora da svolgere sono di dimensioni molto ridotte rispetto a quello del discriminante.

Lo scopo dell'analisi in MATLAB è quello di identificare le coppie di geni più performanti in modo da formare un network in grado di mantenere le prestazioni di classificazione. Come prima operazione viene determinata la "linea di taglio", ovvero la soglia per la selezione delle coppie che verranno utilizzate per la realizzazione del network.

Dopo la formazione del network avviene una nuova analisi discriminante su ogni componente del network e vengono riportate sulla command window le performance delle varie *signature*; in base ai dati riportati l'utente può decidere quale sia la migior *signature*. A seguito della scelta fatta dall'utente l'algoritmo prosegue con la formazione di un nuovo network a partire dai dati della componente più performante del network precedente e attraverso la tecnica statistica $K - fold\ Cross\ Validation$ si ottiene una classificazione ancora più specifica.

Infine è possibile ridurre il numero di *probes* che compongono la signature in modi differenti:

- rimuovendo i nodi a connettività 1;
- rimuovendo possibili nodi pendenti;

- rimuovendo manualmente alcuni nodi;

Dopo la possibile riduzione del numero di nodi viene rieseguita l'analisi del network in modo da ricavare delle performance che ci si aspetta essere migliorate.

L'ultima operazione mancante è la validazione, la quale viene svolta sempre su MATLAB grazie alla funzione $classify$. Questa funzione realizza un modello tramite i campioni contenuti nel training set e alle rispettive label; tale modello viene poi usato per classificare i campioni del test set.

La classificazione fatta dalla funzione $classify$ viene confrontata con le *label*, note a priori, dei campioni del test set in modo da avere una percentuale di performance della *signature* in questione.

Capitolo 2

Tipologie di dati e una loro analisi preliminare

I dati omici che saranno utilizzati per la determinazione delle *signature* sono i medesimi utilizzati nello studio pubblicato nell'articolo *Assessing the clinical utility of cancer genomic and proteomic data across tumor types* (Yuan Yuan et al., 2014) e reperibili on-line sul sito *www.synapse.org*.

Nell'articolo sopra citato vengono trattate diverse tipologie tumorali, ad ognuna delle quali corrispondono diversi dataset; nel presente studio ci si limita all'analisi dei dataset genomici e proteomici che si riferscono al glioblastoma (GBM). I dataset sono così divisi in base alla tipologia delle informazioni che contengono.

Dato che questi dati vengono scaricati da Internet sarà necessaria un piccola analisi preliminare per verificarne la coerenza.

I dataset si presentano sotto forma di matrice $N \times M$ dove i vari campioni sono disposti nelle colonne, mentre nelle righe sono presenti le variabili; la prima riga contiene le *label* dei diversi campioni che può assumere i valori di 1 o 0 in base classe di appartenenza del campione.

2.1 Dati genomici

Per questo tipo di studi vengono usati dati di tipo genomico e non genetico perché i dati genetici (DNA) risultano essere una costante nella vita cellulare (a meno di mutazioni) e per questa loro caratteristica non sono considerati un buon candidato per la determinazione di *signature* dato che non possono fornire informazioni sull'attività della cellula.

All'interno di questa categoria si trovano i dataset che si riferiscono ad mRNA e miRNA. Questi due dataset nonostante siano di ugual tipo hanno dimensionalità differenti, infatti per l'mRNA si ha un numero di variabili che si aggira sull'ordine dei 17 10^3, mentre per i miRNA si hanno circa 500 variabili. L'elaborazione dei dati sarà fatta solo sul dataset di miRNA, perché per elaborare la mole di dati contenuti nel dataset mRNA è necessario un $high - performance\ computer$, con un codice sviluppato in parallelo delle parti più onerose.

2.1.1 miRNA

Il miRNA è una piccola sequenza di RNA che ha l'importante compito di decidere quali sequenze di RNA devono essere tradotte, ovvero coordina le funzioni cellulari. Negli studi fatti negli ultimi anni sulle sequenze di miRNA in cellule malate sono state identificate due principali categorie di miRNA: una oncogena e l'altra tumore soppressore.

Per queste queste sue caratteristiche e il ruolo che ricopre all'interno delle cellule le sequenze di miRNA sono considerati dei target diagnostici e prognostici per diverse patologie come ad esempio il cancro.

Tra i vari compiti del miRNA c'è quello di decidere quali sequenze di mRNA devono essere tradotte in proteine. Per svolgere il suo ruolo il miRNA si accoppia al suo target, l'mRNA; se l'accoppiamento tra i due risulta perfetto la cellula prosegue nel processo di trasduzione imperturbata, mentre se l'accoppiamento tra i due blocchi risulta imperfetto la trasduzione si blocca con conseguenti danni alla cellula.

Il dataset che verrà esaminato è contenuto in una matrice 535×156 e viene analizzato con la stesso procedura con cui è stato analizzato il dataset di mRNA.

Partendo da un analisi generale con le funzioni *boxplot* e *mapcaplot* di ottengono i grafici in figura 2.1 e 2.2.

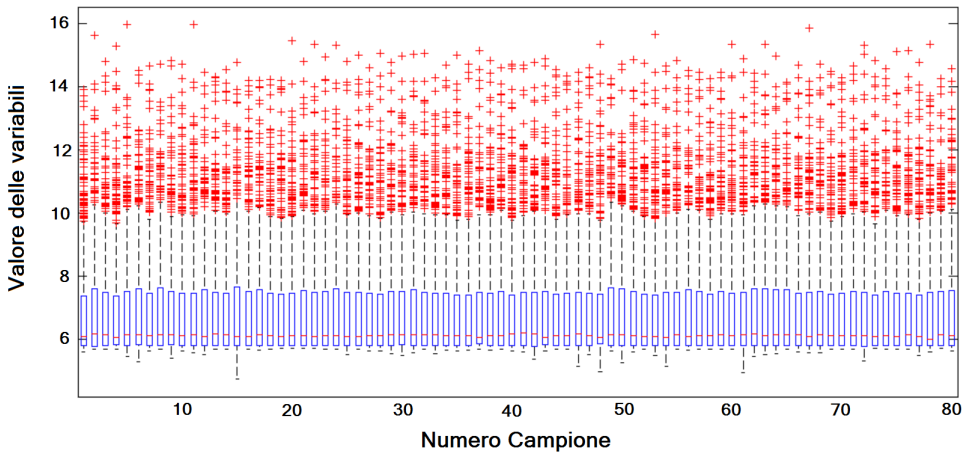

Figura 2.1: Boxplot dei dati di miRNA

Dal *boxplot* viene messo in evidenza come i valori assunti dalle variabili risultano essere prevalentemente maggiori di 6 e la presenza della mediana in prossimità di tale valore dimostra che la maggior parte delle variabili assume un valore vicino a 6.

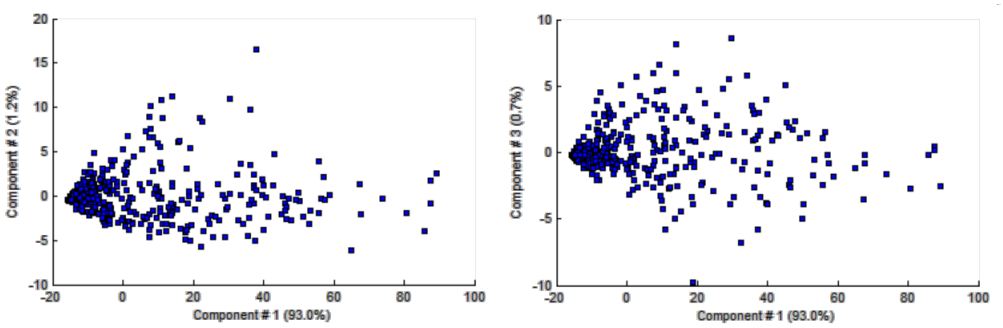

Figura 2.2: Analisi delle componenti principali dei dati di miRNA

Passando all'analisi dei singoli campioni attraverso l'istogramma in Fig. 2.3 è ben evidente che le variabili hanno un andamento esponenziale.

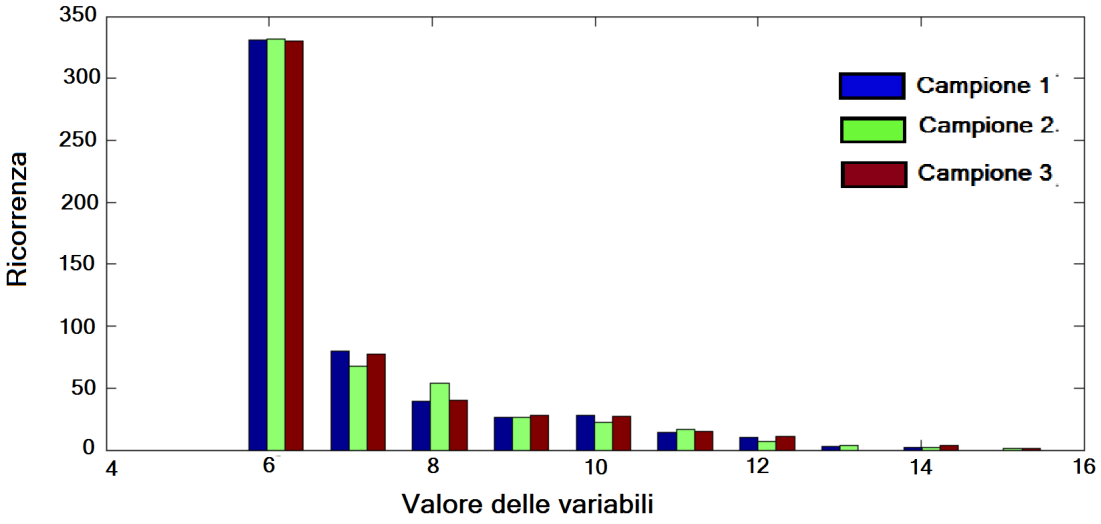

Figura 2.3: Istogrammi di tre campioni dei dati di miRNA

2.2 Dati proteomici

I dati proteomici di un sistema biologico sono raccolti all'interno del proteoma, il corredo proteico del sistema, che contiene tutte le informazioni sulle proteine che vengono prodotte. Il notevole vantaggio che si ha nello studio della proteomica, rispetto a quello della genetica, sta nel fatto che ogni tipo di cellula produce proteine diverse, mentre il genoma è identico in tutti i tipi di cellule.

Questa peculiarità del proteoma permette di associare ad un dato tipo di produzione proteica uno stato fisiologico e in presenza di alterazioni di identificare tre stati della malattia: quantitativo, funzionale e strutturale. L'analisi dei dati proteommici ha quindi permesso di identificare diverse proteine che possono essere utilizzate come target diagnostici, prognostici e terapeutici.

Come esposto precedentemente anche i dati proteomici vengono sottoposti a un'analisi preliminare con l'aiuto di alcune funzioni di MATLAB.

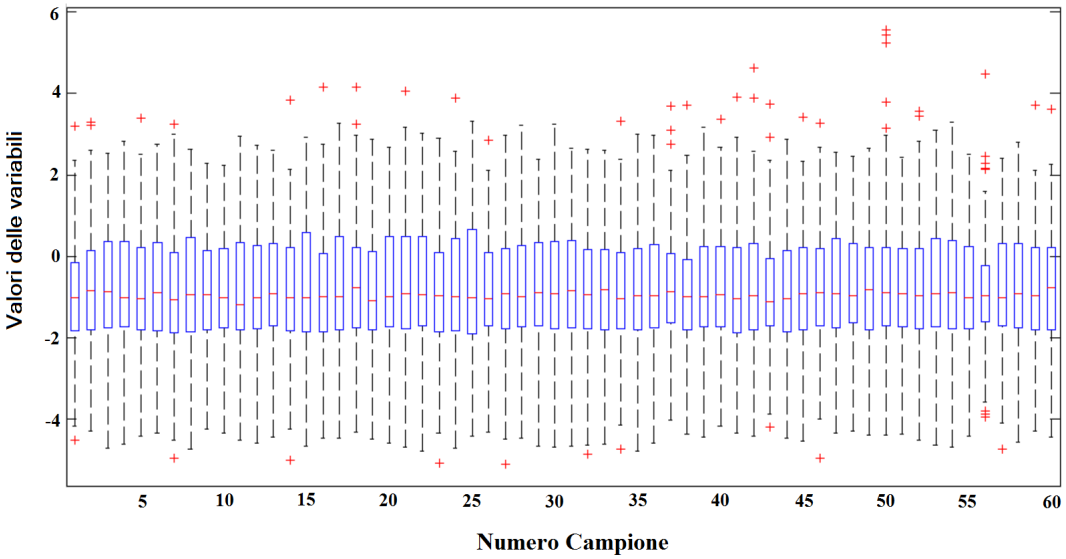

Figura 2.4: Boxplot dei dati proteomici

L'analisi di tipo generale grazie alla funzione *boxplot* evidenzia un andamento costante dei dati per quanto riguarda il loro posizionamento rispetto la mediana e a livello della ampiezza di variazione dei valori per ogni campione.

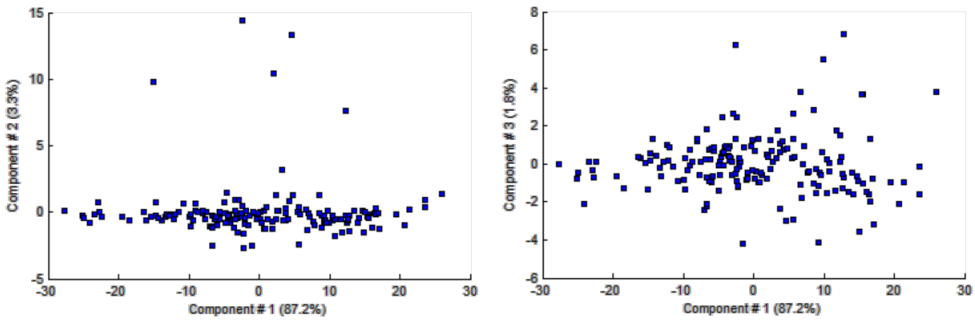

Figura 2.5: Analsi delle componenti principali

I dati proteomici sembrano non essere raggruppati, quindi ogni variabile risulta particolarmente importante perché segue un certo comportamento del sistema.

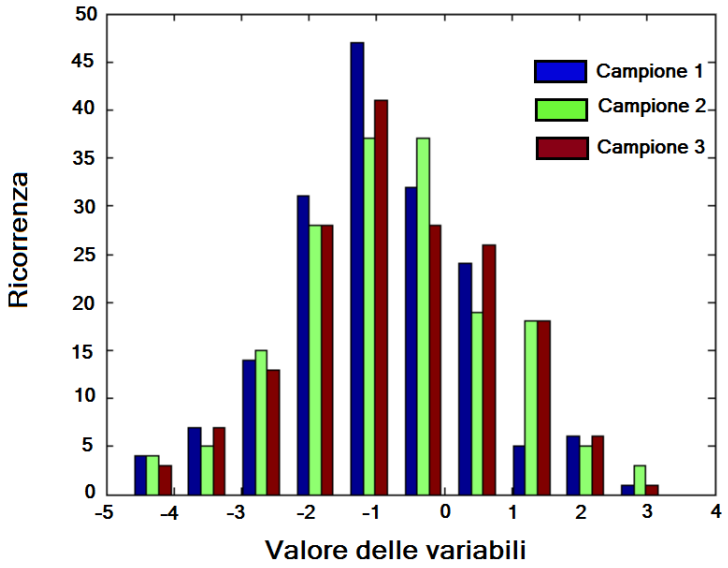

Figura 2.6: Istogramma di tre campioni di dati proteomici

Come mostra la Fig. 2.6 i campioni hanno un andamento di tipicamente gaussiano, coerente con la loro natura.

2.3 Dataset Unico

Prendendo spunto dal lavoro esposto nell'articolo *Assessing the clinical utility of cancer genomic and proteomic data across tumor types* (Yuan Yuan et al., 2014) i dataset di miRNA e proteomica vengono uniti.

Come prima cosa viene verificata la corrispondeza dei 60 sample proteomici anche nel dataset di miRNA, permettendo quindi di proseguire l'analisi.

Il fatto che i dati provengono da dataset di natura diversa non dovrebbe disturbare l'analisi, dato che tutti i dati sono comunque dello stesso ordine di grandezza, come si nota nel seguente istogramma in Fig. 2.7.

Passando all'analisi attraverso la funzione *boxplot* si prevede che la mediana sarà maggiore rispetto a quella del dataset proteomico dato che, come visto in precedenza, i valori di miRNA sono praticamente tutti maggiori di

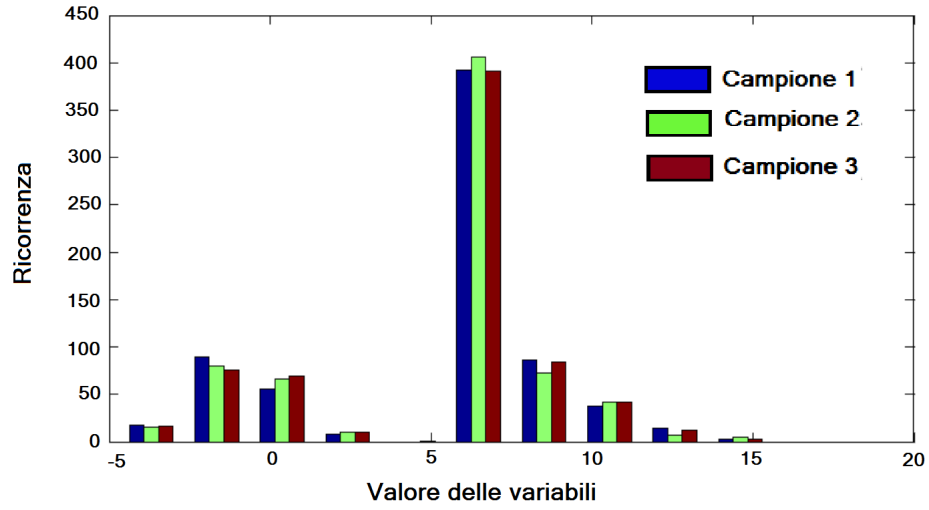

Figura 2.7: Istogramma di tre campioni del dataset unificato

6. Questo spostamento della mediana comporta inevitabilmente la presenza di molti dati che verranno considerati anomali.

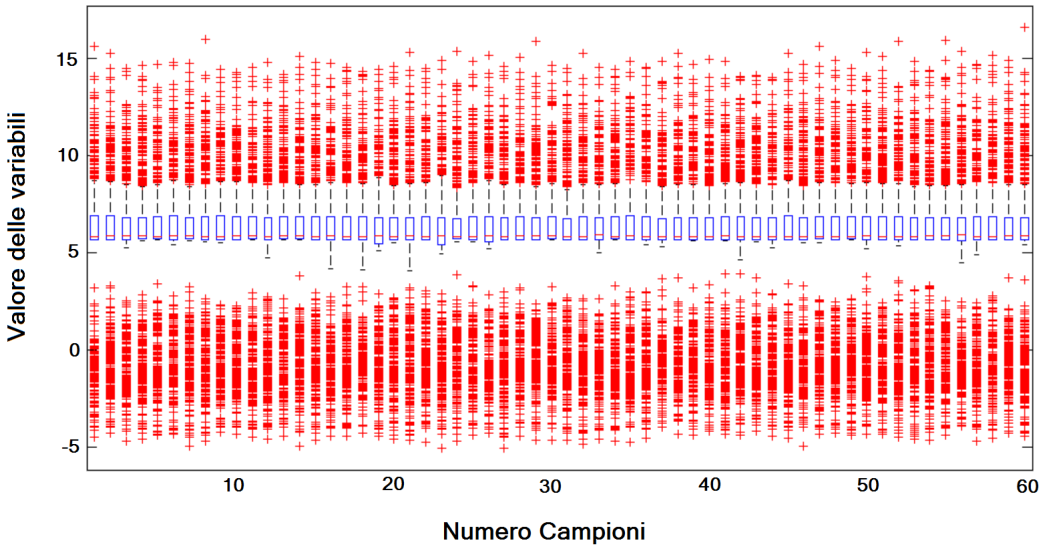

Figura 2.8: Boxplot del dataset unificato

Per quanto riguarda l'analisi delle componenti principali attravereso la funzione *mapcaplot* il risultato è abbastanza prevedibile trattandosi dell'unione di due dataset distinti si ha esattamente l'unione dell'analisi sulle componenti principali dei due dataset originali.

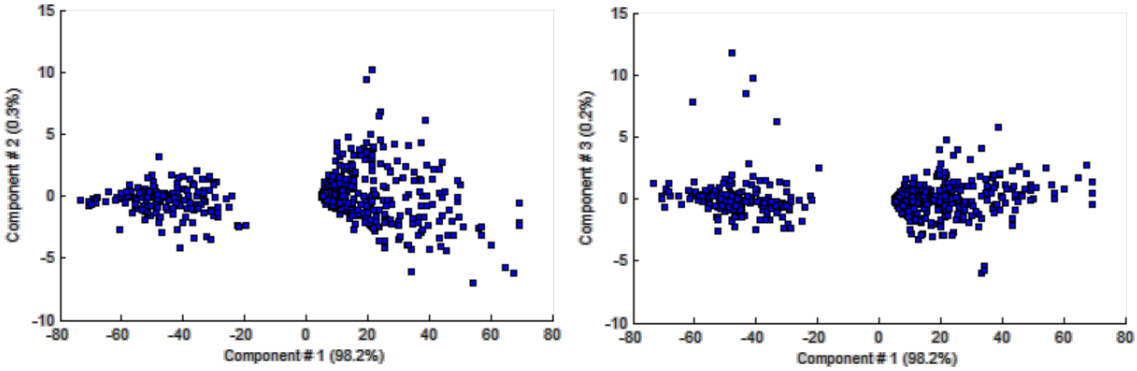

Figura 2.9: Analisi delle componenti principali del dataset unificato

Capitolo 3

Applicazione dell'algoritmo ai dati del glioblastoma

A seguito dell'analisi preliminare appena esposta i dati vengono elaborati attraverso il metodo QDanet_PRO dove ad ogni elemento della coppia e alla coppia stessa vengono assegnati i rispettivi indici di performance. L'analisi sulle coppie continua su MATLAB per la determinazione di network performanti.

L'obbiettivo finale è quello di ottenere *signature* altamente performanti composte dal minor numero di variabili possibile.

Il metodo viene applicato ai dataset che sono resi pubblici grazie al progetto TCGA, che ha l'obbiettivo di sequenziare l'intero genoma di alcuni tumori, tra cui il gliobastoma multiforme.

I dataset vengono divisi in due parti:

- *training set*, corrispondente al 70% dei sample;
- *test set*, corrispondente al restante 30% dei sample.

Questa divisione dei dataset permette di provare e validare il metodo per i diversi tipi di dati.

Nei paragrafi successivi vengono esposti i risultati ottenuti per i dataset di miRNA e proteomica.

3.1 Risutati dei dati di miRNA

Dall'algoritmo QDanet_PRO oltre alle coppie con le rispettive performance vengono restituite informazioni generali sul dataset sui tempi di elaborazione dei dati, di seguito viene riportato come esempio i dati relativi al dataset di miRNA.

```
Numero di pazienti  108  di cui :
63  nel tipo  1
45  nel tipo  0
Geni presenti:533
Tempo impiegato per analizzare 142311 coppie: 6.58657 s
```

Il file contenente le coppie con le performance viene elaborato su MATLAB per la determinazione della *best signature* e durante il processo sulla command window viene riportata una tabella contenente le informazioni sulla media e sul valore massimo, come riportato di seguito:

Classe	Media	Max
1 (45)	14	15
2 (63)	22.5	24
T (108)	36.5	37

Posta come soglia 35 vengono selezionate 2 coppie a 4 *probes* che tramite l'analisi con la funzione *components* di MATLAB corrispondono a 4 nodi a 2 componenti. Sono stati fatti diversi tentativi con soglie inferiori ma i *probes* selezionati risultavo essere sempre gli stessi.

Proseguendo la classificazione con il metodo *diagquatratic* si ottiengono due *signature*, dove viene selezionata come più performante quella composta dai *probes*: hsa-miR-611 e hsa-miR-9. Di seguito vengono riportati i valori di performance della *signature*.

Classe	Performance
1	30/45 (66.7%)
2	23 (41.3%)
T	56/108 (51.9%)

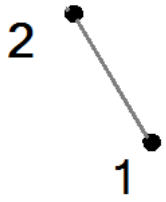

Figura 3.1: Signature del training set dei dati di miRNA

L'analisi prosegue con un $5 - fold\ cross\ validation$ a 200 iterazioni portando a nuove performance:

Classe	perf. media	perf. min	perf. max
1	28 (62.2%)	22 (48.9%)	33 (73.3%)
2	30 (47.6%)	23 (36.5%)	36 (57.1%)
T	58 (53.7%)	49 (45.4%)	65 (60.1%)

I valori medi dopo la $5 - fold\ cross\ validation$ si innalzano solo per la seconda classe e solo di qualche punto percentuale.

Essendo tale signature composta da solo due nodi non è possibile ridurla ulteriormente, quindi si procede alla validazione.

Per la validazione delle *signature* il dataset viene ridimensionato a una matrice contenente solo i valori dei *probes* selezionati per i tutti i campioni.

Tramite alla funzione $classify$ di MATLAB si hanno i seguenti valori di performance riferiti alla buona classificazione o meno della *signature* in base ai tre tipi di classificazione che $classify$ permette di selezionare:

diagquadratic

Corretto	27/47 (57.4%)
Errato	20/47 (42.6%)

quadratic

Corretto	22/47 (46.8%)
Errato	25/47 (53.2%)

mahalanobis

Corretto	26/47 (55.3%)
Errato	21/47 (44.7%)

La miglior performance è data dal tipo *diagquadratic*, dove dei 27 campioni che sono stati ben classificati 14 appartengono alla *label* 1 e i restanti 13 alla *label* 0; si evidenzia una buona classificazione per entrambe le classi. Questo si traduce in:

Sensitivity	51.9%
Specificity	48.1%

3.2 Risultati dei dati di proteomica

Passando al dataset proteomico si ricorda che sia i campioni che le variabili hanno un numero ridotto, quindi ci aspettiamo tempi di elaborazione molto minori, come dimostrano i dati ricavati dal file di output del metodo QDanet_PRO.

```
Numero di pazienti  42  di cui :
22  nel tipo  1
20  nel tipo  0
Geni presenti:171
Tempo impiegato per analizzare 14706 coppie: 0.421745 s
```

Anche in questo caso il processo continua su MATLAB dove vengono registrati i seguenti dati per la media e il massimo del dataset set:

Classe	Media	Max
1 (20)	10	10
2 (22)	8	8
T (42)	18	18

Per proseguire nell'analisi viene scelta come soglia 15; che corrisponde alla selezione di 2 coppie a 3 *probes*. Anche in questo caso la diminuzione

della soglia non ha portato alla selezione di più *probes*. Grazie alla funzione *components* di MATLAB si ottengono 3 nodi a 1 componente. Su questi ogni componente viene applicata la funzione $classify$ con il metodo *diagquadratic* per la *cross validation* con $leave - one - out$.

Risulta una sola *signature* costituita da 3 *probes* (Collagen_VI-R-V, COX-2-R-C e GSK3_pS9-R-V) che possiede le segeunti performance:

Classe	Performance
1	8/20 (40%)
2	11/22 (50%)
T	19/42 (45.2%)

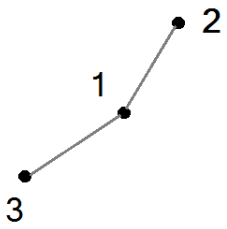

Figura 3.2: Schematizzazione della signature del dataset proteomico

Essento questa l'unica *signature* presente la si assume come *best signature* e si prosegue con una $5 - fold$ *cross validation* con 200 iterazioni che restituisce le seguenti performance:

Classe	perf. media	perf. min	perf. max
1	9 (45%)	4 (20%)	14 (70%)
2	12 (54.5%)	7 (31.8%)	16 (72.7%)
T	20 (47.6%)	14 (33.3%)	27 (64.3%)

I valori medi ottenuti a seguito della *cross validation* sono abbastanza vicini a quelli ottenuti inizialmente.

Eliminando i nodi a connettività 1 la *signature* si riduce al solo *probe* Collagen_VI-R-V comportando un aumento della performance della seconda classe e della performance totale:

Classe	perf. media	perf. min	perf. max
1	8 (40%)	2 (10%)	12 (60%)
2	15 (68.2%)	11 (50%)	18 (81.8%)
T	23 (54.8%)	18 (42.9%)	27 (64.3%)

Il dataset con cui si procede all'analisi è composto da tutti i campioni e solo dalle *probes* costituenti la *signature*. Dato che la *signature* è stata ridotta si vuole vedere se è presente una differenza nella performance di classificazione tra la *signature* originale e quella ridotta.

Cominciando dalla *signature* originale (Collagen_VI-R-V, COX-2-R-C e GSK3_pS9-R-V) i campioni vengono elaborati grazie alla funzione $classify$ di MATLAB; dato che la suddetta funzione ha la possibitià di selezionare il tipo di classificazione, sono stati testati diversi tipi per trovare il più performante.

Il tipo di classificazione più performante in questo caso risulta il *quadratic*:

Corretto	11/18 (61.1%)
Errato	7/18 (38.9%)

mentre per i tipi *diagquadratic* e *mahalanobis* si hanno le seguenti percentuali:

Corretto	7/18 (38.9%)
Errato	11/18 (61.1%)

Corretto	9/18 (50%)
Errato	9/18 (50%)

Analizzando i risultati raggiunti con la classificazione di tipo *quadratic*, che risulta essere la più performante, si mette in evidenza che tra gli 11 campioni classificati correttamente 8 appartengono alla *label* 1 e le restanti 3 alla *label* 0; questo evidenzia una migliore classificazione dei campioni appartenenti alla classe con *label* 1. Di seguito vengono anche riportati i valori si $Sensibility$ e $Specificity$:

Sensitivity	72.7%
Specificity	27.2%

Passando alla validazione della *signature* composta dal solo *probe* Collagen_VI-R-V si riportano i valori di performance per i tre diversi tipi di classificazione:

diagquadratic e *quadratic*

Corretto	10/18 (55.6%)
Errato	8/18 (44.4%)

mahalanobis

Corretto	9/18 (50%)
Errato	9/18 (50%)

I risultati dovuti alla *signature* ridotta sono decisamente peggiori a quelli della *signature* originale.

3.3 Risultati del dataset unico

Il dataset ottenuto dall'unione dei due dataset precedenti ha un numero di campioni pari a 60 ma ben 704 variabili, come mostrano i dati che si possono ricavare dal file di output del metodo QDanet:

```
Numero di pazienti  42  di cui :
22  nel tipo  1
20  nel tipo
Geni presenti:704
Tempo impiegato per analizzare 14706 coppie: 2.621745 s
```

Proseguendo con l'analisi delle coppie su MATLAB si ottengono i seguenti valori per media e massimo:

Classe	Media	Max
1 (20)	22	22
2 22)	19	19
T (42)	41	41

Continuando nell'analisi si seleziona la soglia di 35 e si evidenziano 2 coppie a 3 *probes* che con l'analisi *diagquadratic* portano ad avere una sola *signature* composta da 3 *probes*: Cyclin_E2-R-C, hsa-let-7e, hsa-miR-769-3p, con le seguenti performance:

Classe	Performance
1	12/20 (60%)
2	4/22 (18.2%)
T	16/42 (38.1%)

Proseguendo con la $5-fold\ cross\ validation$ si hanno le seguenti performance:

Classe	perf. media	perf. min	perf. max
1	11 (55%)	7 (35%)	17 (85%)
2	5 (22.7%)	1 (4.5%)	10 (45.5%)
T	17 (40.5%)	10 (23.8%)	22 (52.4%)

Le performance medie non si discostano molto da quelle iniziali.

Come per il caso precendente la *signature* viene ridotta eliminando i *probe* con grado di connettività pari a 1 e la *signature* ridotta risulta caratterizzata dal solo *probe* Cyclin_E2-R-C. I valori di performance a differenza di quelli ottenuti per le *signature* degli altri dataset giovano di questa riduzione, in particolare la seconda classe e la classe totale:

Classe	perf. media	perf. min	perf. max
1	11 (55%)	8 (40%)	14 (70%)
2	11 (50%)	5 (22.7%)	14 (63.6%)
T	21.5 (51.2%)	15 (35.7%)	27 (64.3%)

Proseguendo nella validazione si parte dalla *signature* originale e grazie alla funzione $classify$ si ottengono le senguenti performance per i tre tipi di classificazione possibile:

diagquadratic e *quadratic*

Corretto	7/18 (38.9%)
Errato	11/18 (61.1%)

mahalanobis

Corretto	6/18 (33.3%)
Errato	12/18 (66.7%)

I risultati per la *signature* originale sono decisamente deludenti in quanto non vengono classificati bene neanche il 50% dei campioni.

Passando alla *signature* ridotta invece i risultati che si ottengono sono più soddisfancenti di quelli appena ottenuti, a differenza dei casi precedenti in cui la *signature* originale era più performante:

diagquadratic e *quadratic*

Corretto	11/18 (61.1%)
Errato	7/18 (38.9%)

mahalanobis

Corretto	10/18 (55.6%)
Errato	8/18 (44.4%)

Considerando la performance dovuta alla classificazione di tipo *diagquadratic* e *quadratic* si evidenziano una Sensitivuty e una Specificity pari a:

Sensitivity	54.5%
Specificity	45.6%

dovute alla classificazione di 6 campioni per la *label* 1 e 5 per la *label* 0, su un totale di 11 campioni classificati correttamente.

Capitolo 4

Conclusioni

Lo studio svolto nella ricerca delle *signature* più performanti dei dataset omici riferiti al glioblastoma ha portato all'identificazione di *signature* caratterizate da un basso numero di variabili sia per il dataset genomico che per quello proteomico. Per entrambi i dataset il risultato di performance classificante risulta aggirarsi intorno al 60%, un buon risultato ma non abbastanza soddisfacente per dare un reale supporto in applicazioni mediche.

Avendo analizzato gli stessi dati usati nello studio pubblicato su Nature dal titolo *Assessing the clinical utility of cancer genomic and proteomic data across tumor types* (Yuan Yuan et al., 2014) si è voluta seguire la stessa traccia esposta nell'articolo, che consiste nell'unire i dataset di tipo genomico con quelli di tipo proteomico.

Questa dataset unificato ha portato a risultati paragonabili a quelli ottenuti per i singoli dataset ma solo dopo una riduzione della *signature* ottenuta inizialmente.

I risultati ottenuti da questi tre dataset rappresentano un buon punto di partenza per approfondimenti sul metodo che possano portare ad un miglioramento della performance delle *signature* che vengono individuate.

Bibliografia

[1] Yuan Yuan et al., *Assessing the clinical utility of cancer genomic and proteomic data across tumor type*, Nature, 2014.

[2] Malagoli Stefano, *Approccio network-based alla Discriminant analysis mediante HPC per la ricerca di signature ottimali in dati ad alta dimensionalità.*

[3] Dudoit et al., *Comparison of Discrimination Methods for the Classification of Tumors Using Gene Expression Data,* Journal of the American Statisical Association, 2002.

[4] Shi et al., *Top scoring pairs for freature selection in machine learnning and applications to carcer outcome*, MC Bioninformatics, 2011.

[5] W. Etienne et al., *Comparison of mRNA gene expression by RT-PCR and DNA microarray*, BioTecniques, 2004.

[6] Geman et al., *Classifying Gene Expression Profiles from pairwise mRNA Comparison*, Statistical Application in Genetics and Molecular Biology, 2004.

[7] T. Hastie, Tibshirani and Friedman, *The Elements of Statistical Learning,* seconda edizione, Springer-Verlag, 2009.

[8] T. Heinze, von Löwis, Polze, *Feature Salience for Neural Networks: Comparing Algorithms in Neural Information Processing*, Springer, 2012.

[9] Mozer et al., *Using Relevance to Reduce Network Size Automatically,* Connection Science, 1989.

[10] M. Shekhtman et al., *Robustness of skeletons and salient features in network*, Journal of Complex Network, 2014.

[11] Zhu et al., *Proteomics*, Annual Review of Biochemistry, 2003.

[12] Tyers et al., *From genomics to proteomics,* Nature, 2003.

[13] L. Wong. et al., *Real-time PCR for mRNA quantitation,* BioTecniques, 2005.

Ringrazimenti

Vorrei cogliere questa occasione per ringraziare le persone che mi sono state accanto in questa esperienza a partire dal Prof. Daniel Remondini e il Dott. Giuseppe Levi che mi hanno seguito nella stesura della tesi.

Vorrei ringraziare i miei genitori, Mara e Gian Carlo, per avermi supportato in ogni momento e in ogni decisione e i miei compagni di studi con cui ho condiviso ansie e gioie.

Inoltre vorrei ringraziare anche tutti i miei amici che hanno saputo distrarmi e alleggerirmi i pensieri in momenti difficili.

Infine vorrei ringraziare Andrea per non avermi mai fatto perdere il sorriso.