UNIVERSITÀ DEGLI STUDI DI PADOVA

Dipartimento di Scienze Statistiche

Corso di Laurea Magistrale in Scienze Statistiche

Cancro all'Ovaio di stadio I: uno studio comparativo per l'identificazione bioinformatica delle interazioni tra microRNA e mRNA

Stage I Ovarian Cancer:
a comparative study for the computational identification of microRNA and mRNA interactions

Relatore Prof. Romualdi Chiara

Dipartimento di Biologia

Laureanda: Salviato Elisa

Matricola N. 1014069

Anno Accademico 2012-2013

Indice

1 Introduzione

I microRNA sono stati definiti da alcuni autori[1] "Micromanagers" dell'espressione genica cioè, importanti molecole regolatrici nell'espressione genica degli eucarioti. Nonostante il ruolo dei miRNA nei meccanismi di regolazione siano a tutt'oggi poco conosciuti, molti studi hanno dimostrato il loro contributo fondamentale in molti processi chiave della cellula, dal differenziamento alla proliferazione o alla morte cellulare.

È stato però dimostrato che, nel caso i cui la loro espressione risulti essere alterata, i miRNA possono essere coinvolti in diverse malattie complesse, tra cui numerosi tumori[2].

L'azione ad ampio spettro dei miRNA costituisce pertanto un livello di controllo nuovo ed altamente regolato dell'espressione genica, che può portare ad un possibile impiego di queste molecole come marcatori diagnostici e prognostici o come strumenti terapeutici.

Il passaggio nelle applicazioni cliniche potrebbe quindi rappresentare la sfida futura in ambito scientifico.

1.1 Storia dei miRNA

Sebbene le tecniche per l'analisi dei livelli di espressione degli RNA siano conosciute da almeno un paio di decine di anni, i microRNA sono sempre stati poco considerati dagli studiosi per via della comune idea che l'unico destino dei filamenti di RNA fosse quello di dare origine a proteine. In base a tale dogma, tutti gli RNA di corte dimensioni venivano scartati, a volte venendo etichettati come miseri prodotti di degradazione di RNA più lunghi.

Tale dogma venne messo in discussione a partire dal 1993[3], quando alcuni ricercatori scoprirono in *C.elegans* le funzioni del gene *Lin-4*. Questo gene, essenziale regolatore della divisione cellulare allo stato larvale, dava origine ad un RNA di 61 nucleotidi, il quale veniva processato in un secondo

tempo in un RNA più corto di 22 nucleotidi di lunghezza. Tale prodotto mostrava una perfetta complementarietà antisenso ad un tratto della regione 3'UTR del gene *Lin-14*, e quando *Lin-4* veniva espresso, si asssisteva alla scomparsa sia della proteina *Lin-14* che del messaggero dal quale veniva tradotta. Questa scoperta destò forte scalpore ma non venne presa in debita considerazione poichè non esistevano ortologhi in *Lin-4* all'interno di altre specie.

La scoperta che conferì maggiore importanza a questa classe di molecole risale all'anno 2000, quando il gruppo Reinhart scoprì sempre in *C.elegans* il gene *Let-7*[4], codificante per un secondo miRNA di circa 22 nucleotidi. Questo era coinvolto nella transizione dallo stato larvale a quello adulto, con meccanismi d'azione analoghi al precedente. Tale gene rappresentava però delle sequenze molto conservate ed identificate in diverse specie, che andavano dalle mosche agli esseri umani[5], rivelando che anche gli RNA non codificanti possono giocare un ruolo fondamentale nella sintesi proteica dando una forte spinta allo studio dei miRNA in altre specie. Solo un anno dopo, infatti, vennero identificati oltre 100 miRNA, dei quali 20 in *Drosophila*, 30 nell'uomo e 60 nei vermi.

Al momento attuale, centinaia di miRNA sono stati predetti in animali, piante e virus attraverso differenti approcci, tra cui metodi sperimentali, approcci computazionali, analisi di sequenze genomiche ed EST. Ad oggi sono stati individuati un totale di 24.521 miRNA, di cui 1.872 umani[1].

[1] fonte: miRBase, giugno 2013 - *http://www.mirbase.org.*

1.2 Cosa sono i miRNA

I microRNA sono delle piccole molecole endogene di RNA non codificante a singolo filamento, la cui lunghezza varia da 21 a 25 nucleotidi, ampiamente caratterizzati nelle piante e nei nematodi - dove sono stati osservati per la prima volta - e differiscono tra loro per sequenza, quantità, pattern di espressione e localizzazione genomica.

Il meccanismo più noto dei miRNA si basa sul riconoscimento di specifiche sequenze target dell'RNA messaggero (mRNA) all'interno di un complesso proteico chiamato RISC[2] (RNA-induced silencing complex). Le proteine AGO costituiscono una componente chiave del RISC, rappresentando le proteine effettrici del meccanismo d'azione. Nei mammiferi sono rappresentate da quattro sottofamiglie da 1 a 4, di cui solo il tipo AGO2 è capace di esplicare la funzione del RISC.

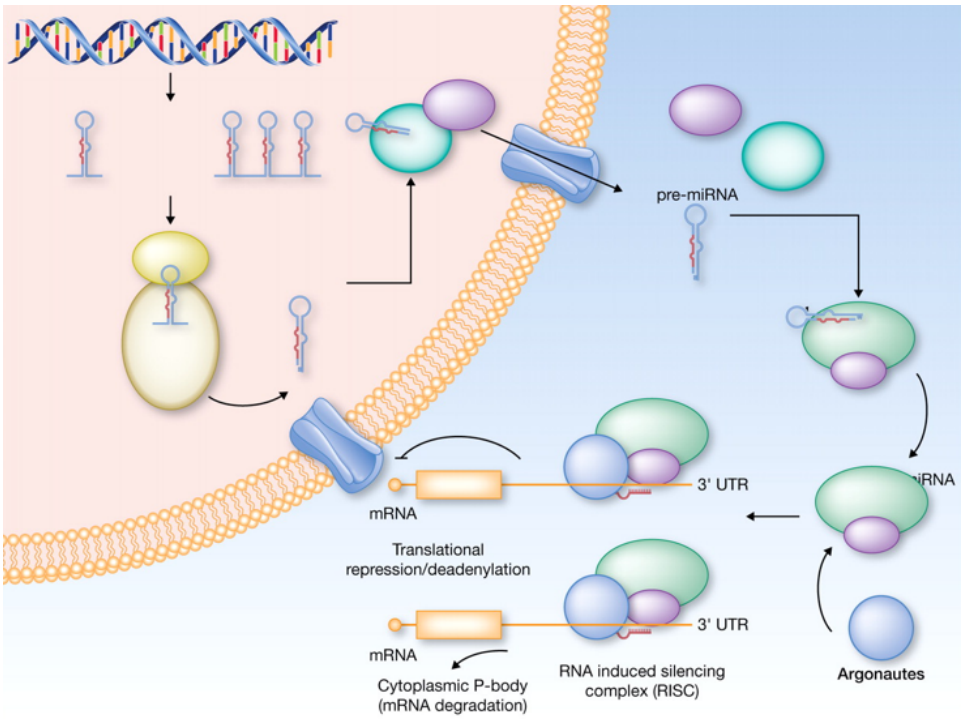
All'interno del complesso proteico il miRNA è in grado di appaiarsi, in maniera più o meno precisa, a sequenze complementari presenti nella regione 3'UTR di RNA messaggeri ed iniziare il meccanismo di azione nell'inibizione dell'espressione genica, denominata RNA interference (RNAi).

L'inibizione post trascrizionale dei miRNA avviene attraverso due differenti meccanismi d'azione, che possono riguardare la degradazione del RNA messaggero oppure l'inibizione della sua traduzione[6] [7]. La condizione che discrimina quale dei due meccanismi verrà attuato è il grado di appaiamento delle basi: con una perfetta complementarietà, l'appaiamento del miRNA determinerà la degradazione dell'mRNA target; con una complementarietà imperfetta avremo l'inibizione della traduzione (Figura 1).

Grazie alla concessione di mismatch, un singolo miRNA può potenzial-

[2] Complesso composto da siRNA e proteine in grado di legare alla regione proteica un filamento di RNA e di ricercare all'interno della cellula un filamento complementare.

Figura 1: Rappresentazione del meccanismo che lega i miRNA agli mRNA
(Fonte: American Association for Cancer Research, 2012)



mente legare centinaia di mRNA targets, che a sua volta può essere una sequenza bersaglio per diversi miRNA, creando fitte reti di regolazione caratterizzate da una natura molti-a-molti. Ne consegue la formazione di un Network cellulare complesso tra diversi RNAs, complicato dall'osservzione che alcuni miRNA possono controllare anche target costituiti da altri RNA non codificanti proteine, come i fattori di trascrizione (TF). Si è stimato che più di un terzo dei geni umani possono essere regolati da miRNA[8].

I miRNA possono essere quindi considerati come piccoli elementi di controllo di diversi pathways regolatori, posti alla base di fondamentali funzioni e numerosi meccanismi che incidono sull'omeostasi dell'organismo.

1.3 Identificazione dei geni target

Per quanto detto precedentemente, l'individuazione e la caratterizzazione dei geni target rappresentano un aspetto cruciale nella comprensione del ruolo funzionale dei miRNA stessi, e delle complesse reti molecolari alla base della regolazione genica.

Negli ultimi anni, si è cercato di estrapolare dagli studi sperimentali i principi alla base di questa interazione al fine di sviluppare algoritmi matematici per la predizione in silico[3] degli ipotetici mRNA bersaglio.

Tali approcci computazionali possono essere sostanzialmente suddivisi in due categorie: gli algoritmi basati sulla complementarietà di sequenza, eventualmente considerando siti di conservazione evolutiva in geni omologhi (miRanda, TargetScan, e PicTar); algoritmi basati sulla predizione della struttura secondaria più favorevole dal punto di vista energetico di molecole di RNA a doppio filamento (RNAhybrid e PITA).

La predizione bioinformatica degli mRNA target rappresenta certamente un primo importante approccio nello studio del ruolo funzionale dei miRNA, ma i target ipotetici predetti devono essere sempre validati da specifiche metodologie sperimentali.

Esistono diverse raccolte di miRNA-mRNA definite validate, in quanto riprodotte in vitro[4], attraverso le quali è possibile dimostrare l'effettiva regolazione dei geni bersaglio da parte dei miRNA e utili per un confronto sia per diversi algoritmi di predizione che per altre metodologie che hanno

[3] la locuzione è usata per indicare fenomeni di natura chimico biologica riprodotti in una simulazione matematica al computer, invece che in provetta o in un essere vivente.

[4] la locuzione è usata per indicare fenomeni biologici riprodotti in provetta e non nell'organismo vivente. Il vantaggio è quello di permettere di analizzare il fenomento isolandolo dal contesto che potrebbe creare un rumore di fondo troppo elevato per poter distinguere il fenomeno in maniera chiara

Database	ultimo aggiornamento	totale	specie	n° miRNA	n° mRNA
TarBase	2005 (v5.0)	1.331	8	178	998
miRecords	giugno 2013	2.705	9	644	1.901
mirTarBase	giugno 2013	34.083	15	1.1019	15.095
miRWalk	marzo 2011	67.598	3	2.044	3.821

Tabella 1: Elenco dei principali databases di associazioni validate

lo scopo di investigare l'azione regolatoria dei miRNA. Tra questi citiamo miRecords, TaRBase[9], mirWalk[10] e mirTarBase[11].

Mentre le interazioni incluse nei primi tre databases sono state curate manualmente, quelle contenute in mirTarBase sono state automaticamente estratte dalla letteratura scientifica attrevarso l'uso di tecniche di text mining[5], rendendo di conseguenza quest'ultima raccolta meno affidabile rispetto alle altre.

Nell'ottica di utilizzare questi database di iterazioni validate come risorse a livello sperimentale per chi studia i miRNA e per gli informatici che si occupano di sviluppare programmi di nuova generazione per la predizione dei target, si riscontrano un forte limite: oltre alla piccolissima frazione di validati rispetto al totale di possili iterazioni - dovuta ad un costo ancora elevato delle verifiche effettuate in laboratorio - l'omissione di quelle interazioni scoperte come veri negativi, che nei metodi di classificazione statistici risultano utili tanto quanto i veri positivi.

Per tale motivo le predizioni risultanti dai diversi software differiscono ampiamente rispetto ai metodi adottati. Al fine di confrontarli, con l'ausilio dei target validati, sono stati condotti diversi studi i quali hanno stabilito che nessun programma è superiore ad altri[12]. Per questo motivo, è pratica

[5]applicazione di tecniche di data mining per il processo di estrazione di informazioni significative da banche dati di grandi dimensioni di documenti testuali.

comune tra i ricercatori avvalersi di diversi programmi per la predizione dei target e focalizzarsi sul risultato della loro intersezione, o unione, a seconda delle esigenze.

1.4 Problematiche

Una delle problematiche principali che affligge tutti i software di predizioni disponibili è la significativa frazione di falsi positivi, che in questo contesto sono identificati come quelle associazioni predette come possibili interazioni ma che non risultano sperimentalmente validate.

Questo non solo è dovuto alla comprensione limitata che ancora abbiamo dei meccanismi di base dell'appaiamento tra miRNA e il suo target, ma anche dal fatto che, per la maggior parte dei miRNA noti, vengono predetti un gran numero di trascritti target, a causa della brevità della sequenza di interazione e della concessione di mismatch all'interno di questa. Il tutto è complicato ulteriormente dalla natura molti-a-molti delle relazioni predette che fanno assumere alle reti di regolazione post trascrizionale una natura altamente complessa e difficilmente esprimibile in algoritmi matematici.

Nell'ottica di migliorare la ricerca delle relazioni funzionali tra mRNA e miRNA, aumentando la specificità di questi software, è stata recentemente proposta l'integrazione delle predizioni dei target con l'informazione sui profili di espressione, ipotizzando una forma di relazione inversa tra le due molecole, sfruttando il numero crescente di evidenze sperimentali a supporto del meccanismo d'azione dei miRNA attraverso l'inbizione della traduzione - piuttosto che della sua repressione attraverso la degradazione.

Infatti, uno dei primi studi effettuati mediante microarray, dimostrò che i miRNA tendono a deregolare gli mRNA target, relazionanosi in maniera inversa con i profili di espressione[13]; lavori più recenti hanno confermato

questa correlazione inversa nonostante l'effetto biologico avvenga sui livelli della proteina[14][15].

Da allora gli approcci integrativi sono stati visti come una soluzione interessante per il raffinamento delle predizioni target e, in questa direzione, sono stati sviluppati numerosi programmi bioinformatici. Tra questi metodi, la differenza principale riguarda il modo in cui le informazioni sull'espressione vengono utilizzate. Per esempio, GenMir++[16] utilizza un contesto bayesiano, HOCTAR[17] ed altri autori la correlazione, MAGIA[18] la mutua informazione, mentre MMIA[19] si avvale di metodi utilizzati nell'analisi di arricchimento dei dati di microarray.

Nonostante sia stato dimostrato che aggiungendo i dati di espressione si riducono il numero di falsi positivi, gli strumenti a disposizione sono senza dubbio inadeguati sia alla rapidità crescente dell'ammontare di appaiamenti tra miRNA e profili genici, sia nello spiegare in maniera esaustiva la forza della loro relazione. In parte questo è dovuto all'eccessiva semplificazione del processo biologico adoperata dai modelli: si pensi che nessuno dei metodi precedentemente citati include in qualche modo la relazione molti-a-molti, elemento chiave dei network regolatori.

1.5 Scopo della tesi

In questo lavoro di tesi proponiamo diversi approcci integrativi, alcuni ampiamente già utilizzati in questo contesto ed altri del tutto inediti, cercando sia di raffinare le predizioni degli algoritmi basati sulla complementarietà della sequenza, che di caratterizzare meglio la relazione-legame tra microRNA e mRNA, attraverso un esperimento di microarray riferito a pazienti affetti dal Cancro Ovario (EOC) allo stadio I.

Inoltre vaglieremo la validità di alcune ipotesi formulate dagli autori dei modelli utilizzati, come ad esempio l'inserimento di un ulteriore livello di informazioni o l'efficacia dell'aggiunta di alcuni vincoli, che dovrebbero rispecchiare più fedelmente la natura della connessione tra queste due molecole. Di particolare interesse sarà l'inclusione della natura molti-a-molti nel tentativo di spiegare il profilo di espressione di un determinato mRNA bersaglio, rispetto all'espressione dei microRNA ad esso collegati, secondo una qualche lista di predizione iniziale derivante dagli algoritmi bioinformatici di previsione dei target.

Infine il nostro studio, attraverso l'utilizzo dei migliori approcci di integrazione individuati, cercherà di definire un pannello di RNA coinvolti nella tumorogenesi del Cancro Ovarico che potrebbero risultare utili nell'identificazione di bersagli terapeutici.

2 Materiali

Al pari di ogni altro gene, un microRNA può essere associato, in ragione della sua funzione biologica e della sua espressione specifica, ad un preciso stato fisiopatologico. Sin dalla loro scoperta, i miRNA sono oggetto di studio a tale proposito, e numerosissime pubblicazioni scientifiche negli ultimi anni hanno evidenziato correlazioni significative fra l'espressione di alcuni gruppi di miRNA e la presenza di determinate patologie, spesso neoplastiche[20].

Il meccanismo alla base della capacità dei miRNA di avere un ruolo attivo nello sviluppo del cancro può essere sintetizzato come segue: se un miRNA ha tra i suoi bersagli un determinato trascritto genico, e quel trascritto codifica per un oncosopressore, un aumento dell'espressione locale del miRNA avrà l'effetto di favorire l'insorgenza o lo sviluppo di una neoplasia. Simile sarà chiaramente l'effetto della ridotta espressione di un miRNA che invece abbia tra i suoi bersagli un oncogene[21].

L'osservazione dell'espressione dei vari miRNA nelle neoplasie ha fornito la possibilità di caratterizzare uno o più microRNA, indipendentemente dalla conoscenza della loro funzione specifica[22], come biomarcatori di una determinata patologia potenzialmente utilizzabili come indicatori diagnostico.

Nella maggior parte dei casi, questi studi sfruttano tecnologie di analisi su scala genomica, ovvero testano contemporaneamente l'espressione di tutti i miRNA conosciuti, utilizzando la tecnologia dei microarrays. Altri studi si basano invece sulla ricerca più circoscritta di gruppi ristretti di miRNA, con tecnologie standard come realtime PCR e Northern Blot. La combinazione delle due tipologie di metodiche appare, al momento, la via di elezione per studi che intendano evidenziare, attraverso la determinazione dell'espressione dei miRNA, un loro ruolo diretto o indiretto nello sviluppo e nella progressione di patologie neoplastiche.

Questo lavoro di tesi utilizza i dati di espressione di microarray di mRNA e miRNA, sul Cancro Ovarico Epiteliale (EOC), impiegati in un precedente studio per l'individuazione un insieme di miRNA marcatori associati alla recidività della malattia e alla sopravvivenza su pazienti classificati secondo criteri FIGO allo stadio I[70]. Lo scopo sarà quello di confrontare diversi algoritmi di integrazione delle predizioni dei target, ed individuare un set di possibili accoppiamenti miRNA-mRNA target associati alla malattia di stadio I.

2.1 Problema biologico: il Cancro Ovarico Epitalico

La sopravvivenza a 5 anni dei pazienti con il Cancro Ovarico (EOC) dipende dalla diffusione della malattia alla diagnosi: nei pazienti con una malattia limitata alle ovaie la sopravviavenza arriva all'80%; nei casi in cui la malattia coinvolga la parte superiore o inferirore dell'addome, la sopravvivenza nei 5 anni successiva di abbassa al 20%[24].

Considerando che tra i pazienti di stadio I, meno del 20% hanno una malattia aggressiva e recidiva entro i 5 anni dal primo intervento, diventa cruciale discriminare ad uno stadio primitivo della malattia i pazienti con uno stadio I curabile, da quelli che saranno recidivi dopo la chemioterpia. Questo permetterebbe di profilare una cura ottimale, con terapie più aggressive per i soli pazienti a rischio di recidività riducendo problemi di tossicità dei farmaci, non rare nel caso di cure chemioterapiche.

La predizione dei recidivi sulla base della attuali conoscenze cliniche e dei lineamenti patologici è difficoltosa, per questo una strada percorribile è quella di migliorare la conoscenza dei meccanismi genetici e molecolari a cui è associato ciascuno stadio del tumore. La conoscenza dei pathway molecolari, che sono alterati durante la trasformazione neoplastica, possono essere d'aiu-

to nella scoperta di biomarcatori per una precoce rivelazione della malattia, una predizione del responso clinico, e una guida per il trattamento[25][26][27].

Nonostante sia stata dimostrata la significatività clinica e patologica dell'azione dei miRNA nel cancro ovarico, questi studi si sono principalmente focalizzati sullo stadio III e IV della malattia[28]; molto poco si sa ancora sullo stadio I della malattia. Purtroppo i segni e i sintomi del carcinoma del'ovaio sono vaghi (dolori addominali o pelvici, senso di gonfiore) e la diagnosi è quindi molto spesso tardiva, rendendo i casi di EOC al primo stadio non comuni: ogni anno meno del 10% dei pazienti totali con EOC sono diagnosticati al primo stadio. Questo ha reso difficoltoso raccogliere una coorte di pazienti di grandezza sufficiente per avere un'adeguata potenza statistica.

Il primo studio che si è focalizzato sull'EOC di stadio I risale al 2011[70]: sono stati individuati 34 miRNA biomarcatori legati alla sopravvivenza, di cui 11 differenzialmente espressi nei recidivi. In particolare, è stato sottolineato il ruolo centrale del *miR-200c* nel generale il meccanismo della recidività, in accordo con altri studi su diverse forme neoplastiche che lo indicavano come un marcatore delle cellule epiteliali e un potente regolatore. Inoltre hanno individuato, utilizzando un intersezione tra i target predetti del *miR-200c* e una lista precedentemente identificata di potenziali marcatori, due target - VEGFA e TUBB3 - come significativamente sovra-regolati, rimandando però a futuri studi la loro potenziale rilevanza per nuovi approcci terapeutici nello stadio I dell'EOC.

Quindi, nonostante siano state confermate delle rilevanze significative del ruolo di alcuni miRNA nel Cancro Ovarico di primo stadio, il complessivo meccanismo dei pathway molecolari regolatori in questo contesto rimane tutt'ora un argomento poco esplorato, e quindi terreno fertile per nuovi studi e sperimentazioni.

Il lavoro di tesi utilizzerà i dati di espressione collezionati in questo ultimo articolo, non solo nel tentativo di studiare e valutare alcuni algoritmi di predizione dei target, ma anche con la speranza di fornire potenziali nuove proteine convolte nei meccanismi molecolari attivi in questa patologia e/o nuovi target terapeutici. La bioinformatica offre infatti un supporto fondamentale al fine di accellerare il percorso della ricerca di base verso la clinica.

2.2 Collezione di campioni

In questa tesi sono stati utilizzati i profili di espressione di miRNA e mRNA, ottenuti tramite tecnologia microarray, pubblicati nello studio *"Association between miR-200c and the survival of patients with stage I epithelian ovarian cancer: a retrospecitive study of two indipendent tumor rissue collections"*.

I dati si riferiscono a campioni tumorali di una coorte di 89 pazienti classificati con i criteri FIGO come stadio I dell'EOC, provenienti da una collezione di tessuto tumorale, depositata al dipartimento di Oncologia dell'Istituto Mario Negri di Milano. In particolare, ci siamo serviti esclusivamente della parte per cui erano stati rilevati anche i dati di espressione dei microRNA, ossia 76 di questi. I campioni sono stati raccolti tra il Settembre del 1992 e il Marzo 2005 - da pazienti sottoposti a chirurgia citoriduttiva - e che contenevano più del 70% di tessuto tumorale. Nella totalità si disponde della misurazione di 234 profili di miRNA e 13.952 di mRNA.

Essendo stati sviluppati diversi databases di annotazioni geniche[6] al cui interno, per ciascuno di questi, sono stati utilizzati codici identificativi diffe-

[6] Con il termine annotazione si intende l'inserimento di tutte le informazioni riguardanti la funzione di una determinata sequenza

renti, può capitare che alcuni strumenti preferiscono determinate nomeclature rispetto ad altre, rendendo necessario una processo di conversione.

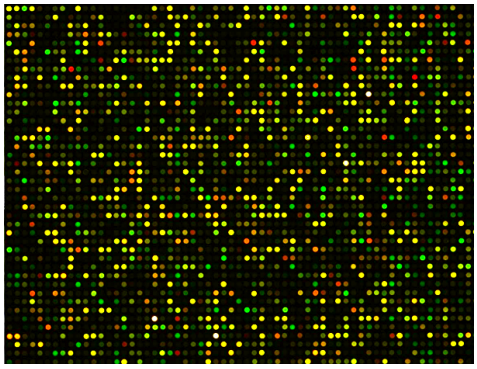
Nello specifico, abbiamo utilizzato sia le annotazioni Entrez che quelle Ensembl - rispettivamente utilizzate nei databases NCBI e EMBL - e questo ha portato ad una riduzione dei trascritti finali, a causa della relazione non univoca di questi id. Il totale di profili di espressione di mRNA utilizzati sono dunque pari a 13.795.

2.3 La tecnologia Microarray

In biologia molecolare, con il termine profilo di espressione si intende il processo attraverso cui l'informazione contenuta in un gene - codificante o non - viene convertita in una macromolecola funzionale. Attualmente la quantizzazione delle molecole di RNA è il dato di maggior interesse nello studio dei trascritti.

La tecnica classica di quantizzazione del mRNA e del miRNA è il Northern blot, un processo attraverso il quale un estratto di RNA proveniente dal campione biologico da analizzare viene dapprima separato in un gel di agarosio, quindi ibridato con una sonda radioattiva complementare al solo RNA di interesse. In questo modo, è possibile visualizzare e quantificare la presenza dell'RNA di interesse. Il Northern blot è una tecnica utilizzata sempre più raramente, a causa delle problematiche correlate all'uso di reagenti radiottivi e della scarsa sensibilità di quantizzazione. Due metodi più moderni per misurare la quantità di un singolo mRNA sono la real-time polymerase chain reaction (real-time PCR) o la reverse trascriptase-polymerase chain reaction (RT-PCR), che permettono di quantificare la presenza di un determinato trascritto con una elevatissima sensibilità.

Figura 2: Esempio della rilevazione dei segnali fluorescenti sulla superfice nella tecnologia Microarrey



I metodi precedentemente elencati però consentono di quantificare solo poche differenti molecole di miRNA per volta, ma esistono metodologie che permettono di effettuare screening molto ampi del trascritto complessivo di una cellula. Una di queste metodologie è qulla dei DNA Microarray, che attraverso l'utilizzo di supporti che contengono migliaia di sonde di DNA complementari ai trascritti della cellula, è in grado di fornire un'analisi contemporanea di migliaia di differenti trascritti.

Il primo lavoro sui microarray è stato pubblicato nel 1995 da Mark Schena dell'università di Stanford. L'idea ebbe origine dalla nacessità di studiare l'espressione genica delle piante attraverso la caratterizzazione dei loro fattori di trascrizione: la difficolità dovuta all'assenza di adeguati strumenti di analisi fece avanzare la proposta di sviluppare degli appositi chip come dispositivi utili allo studio dei trascritti. Così, il Davis Laboratory e il dipartimento di biochimica di Standford realizzarono microscopici array contenenti sequenze geniche di piante bloccate su un substrato di vetro; i micorarray fuono poi utilizzati per misurare l'espressione genica di tali piante in esperimenti di ibridazione con campioni di mRNA marcati in fluorescenza.

In condizioni sperimentali appropriate, i segnali fluorescenti sulla super-

fice del vetrino producono una misura dell'espressione di ogni gene rappresentato sul Microarray: dalla quantificazione di tale fluorescenza è possibile risalire al livello di espressione di ciascun gene. Questa tecnica è stata successivamente estesa anche nella misura dell'espressione dei microRNA.

2.4 Pre-processamento dei dati

Raramente i dati sperimentali sono pronti per essere utilizzati immediatamente così come sono stati raccolti, per questo si rende necessaria una prima fase di pre-processamento. Nel caso di dati di espressione provenienti da microarray i due passaggi fondamentali sono la normalizzazione e l'imputazione dei valori mancanti.

2.4.1 Normalizzazione degli array

Nella tecnologia microarray esistono molte fonti di variazione sistematica che possono influenzare i livelli misurati di espressione genica (distorsioni dovute all'efficienza dei diversi fluorofori, all'ibridazione, al processo di spottaggio..). Lo scopo della normalizzazione dei dati è quello di minimizzare gli effetti delle variazioni sperimentali e/o delle tecniche, in modo tale da rendere confrontabili i vari campioni biologici.

Nella letteratura scientifica sono stati proposti diversi approcci, e tutti hanno fornito buone performance al fine di ridurre gli errori sistematici, sia per la tencologia a singolo che a doppio canale[29][30][31]. Uno studio esplorativo per l'analisi comparativa dell'impatto delle diverse tecniche di normalizzazione [32] ha inoltre rivelato che esiste una differenza limitata di sensibilità e specifictà rispetto alla tecnica di normalizzazione scelta.

Alla luce di queste osservazioni, in questa analisi, si è scelto di utilizzare una tra le tecniche più semplici e di uso comune nella normalizzazione

dei profili di espressione genica: la normalizzazione quantile. L'obiettivo di questo metodo è quello di rendere uguali le distribuzioni empiriche di tutti gli array, prendendone uno come riferimento, e proiettando le osservazioni sulla diagonale del grafico quantile-quantile n-dimensionale.

2.4.2 Imputazione degli NA

Un altro problema che si incontra nell'analisi dei dati derivanti da tecnologie microarray è l'esistenza di valori mancanti nella matrice dei dati. La ragione per cui spesso vengono a mancare le misure relative ad un certo gene sono diverse e possono essere connesse agli strumenti utilizzati (presenza di polvere, graffi nei vetrini) oppure al trattamento computazionali dell'immagine e al processo di trasformazione del segnale luminoso in dato numerico (insufficiente risoluzione, alterazione dell'immagine).

Esistono diversi metodi per l'imputazione dei valori mancanti, alcuni costruiti sulla media generale o di gruppo, e altri sulla regressione (decomposizione in valori singolari, SVD) o sul cosidetto "metodo dei k vicini" (k-nearest neighbors). Mentre le prime due tecniche si sono rivelate inadeguate in questo contesto biologico, non tenendo in debita considerazione la struttura di correlazione dei dati, le utlime hanno fornito performance superiori in termini di accuratezza.

Poichè, per funzionare in maniera apprezzabile, l'approccio a decomposizione in valori singolari implica la disponibilità di un numero ragionevole di geni completi (nel caso della nostra matrice più del 50% dei profili di espressione dei miRNA contengono almeno un valore mancante, vedi Tabella 2), si è scelto di utilizzare la tecnica KNN: il metodo prevede l'utilizzo di quei profili che più somigliano a quello che presenta il valore mancante.

Tabella 2: Informazioni chiave per l'imputazione degli NA nei dati dell'EOC.

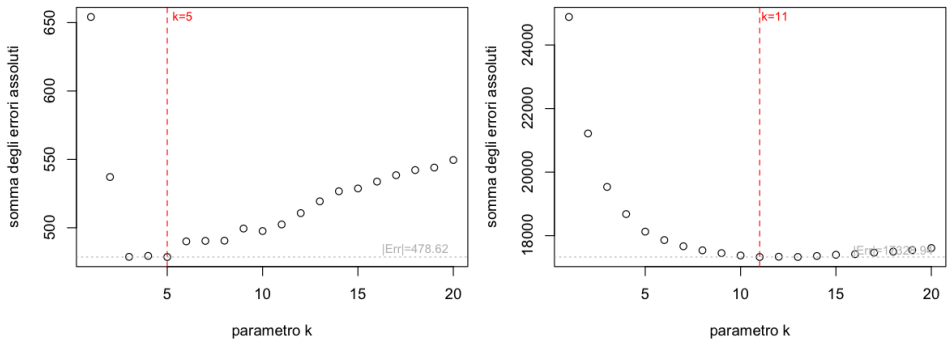
Profili di espressione	**originali**	**simulati**	**originali**	**simulati**
dimensione matrice	13.795 × 76	9.286 × 76	234 × 76	101 × 76
n° totale mRNA	13.795	9.286	234	101
n° mRNA con almeno un NA	4.509 (32.68%)	3.048 (32.82%)	133 (56.84%)	53 (52.48%)
n° totale valori mancanti	38.536 (3.68%)	25.877 (3.67%)	1661 (9.34%)	640 (8.34%)

Tale tecnica prevede (Algoritmo 1) di suddividere il dataset in due distinte matrici: X_c, che rappresenta il sottoinsieme dei geni completi, e X_m il sottoinsieme dei geni con almeno un valore mancante nei diversi campioni. Indicato con x^*, un qualsiasi valore appartenente alla matrice X_c, l'algoritmo consiste nel calcolare la distanza tra il profilo di x^* e quelli di tutti i profili (mRNA o miRNA) contenuti nella matrice X_c, usando solo le coordinate presenti nel profilo di x^*. Sucessivamente il dato da imputare viene calcolato come una media pesata delle coordinate corrispondenti a x^*, dei k geni che risultano più prossimi [33]. In particolare si usano pesi inversamente proporzionali alla distanza e tali da sommare ad uno, e per il calcolo della distanza si utilizza quella Euclidea che risulta sufficientemente accurata nonostante problemi di sensibilità agli outliers.

Per quanto concerne la scelta del parametro k, che regola quanti dei vicini debbano essere selezionati nel calcolo del valore mancante, abbiamo fatto riferimento alla procedura (Algoritmo 2) suggerita nell'articolo di riferimento del metodo. Poichè si assume che nei dati di microarray i valori mancanti non siano stati rilevati in maniera casuale, si cerca di sfruttare la composizione degli NA all'interno della matrice originale nel seguente modo.

Si campionano un numero di righe pari a quelle della matrice X_c, dalla matrice originale X, e, utilizzando le locazioni mancanti in quest'ultima, si assegnano gli NA alla matrice X_c. Questo porta ad avere una struttura simile di valori mancanti per il dataset di geni completi. Per questa ver-

Figura 3: Somma degli errori assoluti, calcolati per le due matrici miRNA (a) e mRNA (b), in funzione del parametro k. Con linea tratteggiata rossa è evidenziato il numero di vicini ottimali da utilizzare come scelta.



(a) Matrice miRNA (b) Matrice mRNA

sione "contaminata" di X_c possiamo imputare i valori mancanti tramite la tecnica KNN precedenetemente descritta utilizzando diversi valori per k: il valore che minimizzerà l'usuale somma degli errori assoluti verrà scelto come riferimento.

In Figura 3 sono rappresentati i risultati ottenuti utilizzando le matrici relative al Cancro Ovarico: il minimo per il nostro datase si raggiunge in corrispondenza del valore 11, per la matrice di espressione degli mRNA, e 5, per i miRNA (in linea con i valori standard che suggeriti dalla letteratura, pari rispettivamente a 10 e 5).

L'implementazione vera e propria del metodo è stata fatta utilizzando la funzione *impute.knn()* del pacchetto *impute* (versione 1.32.0), scaricabile collengandosi direttamente da R all'indirizzo *http://bioconductor.org/biocLite.R*.

Per ulteriori dettagli si veda l'Appendice A.1. dove è riportato il codice sviluppato per il calcolo del parametro k ottimale.

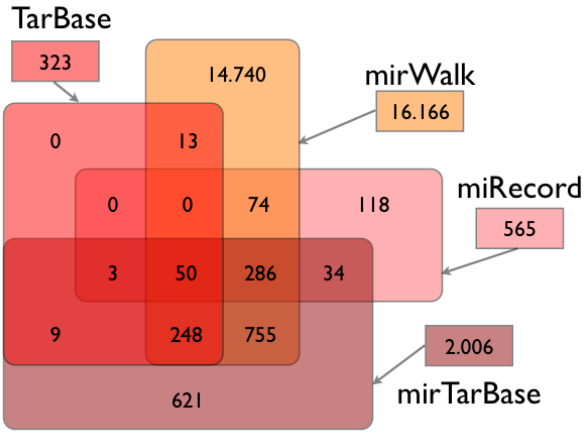
2.5 Collezione di interazioni validate

Nei capitoli successivi, ci troveremo di fronte alla necessità di valutare le performance dei vari approcci di integrazione: attualmente la via elettiva per effettuare questa analisi è condotta sulla base della ricerca di accoppiamenti miRNA-mRNA etichettati come "veri positivi" (validati).

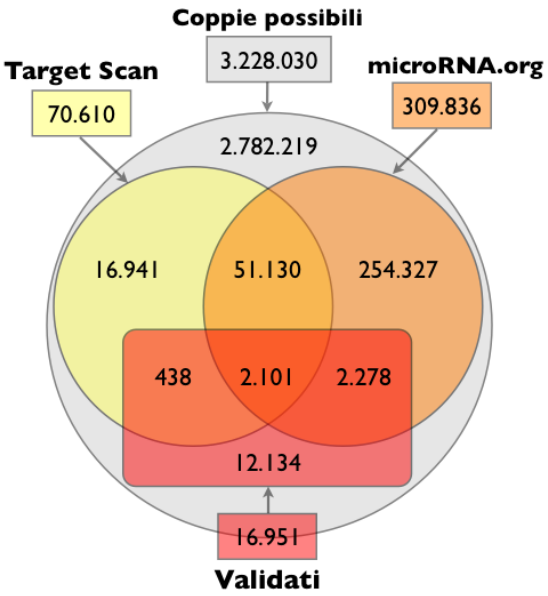
Abbiamo dunque creato una nostra collezione scaricando tutte le associazioni validate contenute nei più importanti database cunsultabili via web - miRecords, TarBase, miRWalk, mirTarBase - e intersecato il risultato con i nomi degli mRNA e dei miRNA presenti nel dataset dei dati dell'EOC: in questo modo, per alleggerire il carico computazionale, abbiamo mantenuto solo quelle associazioni che potessero essere individuate nelle nostre soluzioni.

L'intera collezione conta un totale di 16.951 associazioni: di queste 323 contenute in TarBase, 565 in miRecord, 2.006 in mirTarBase e 16.166 in mirWalk. L'intersezione degli elementi di ciascun database da cui è stata creata la nostra collezione è visibile in Figura 4.

Figura 4: Diagrammi di Venn per la raccolta di interazioni validate. Sono rappresentate le intersezioni rispetto ai databases di provenienza (a) e rispetto a tutte le interazioni possibili nei dati dell'EOC, suddivisi per liste di predizioni iniziali utilizzate (b).



(a) Collezione validati



(b) Collezione di campioni

3 Metodi & Razionale

Nonostante la crescita delle informazioni sui miRNA le loro funzioni specifiche non sono ancora ben note. Numerosi studi hanno però dimostrato che queste possono essere dedotte mediante le proprietà dei loro mRNA target. Questo ha dato centralità e importanza alla previsione bioinformatica dei geni bersaglio.

La disponibilità di dati di espressione di miRNA, provenienti da esperimenti di microarray, ha visto la nascita di numerosissimi algoritmi capaci di integrare i software di predizioni (basati principalmente sulla complementarietà delle sequenze), al fine di migliorare la loro specificità. Così, il crescente numero di metodi, ha portato a domandarsi quale fosse l'approccio di integrazione più adeguato per individuare le associazioni funzionali.
Ma, basandoci sulle nostre conoscenze attuali, non esiste in letteratura un confronto tra algoritmi di integrazione. Questo lavoro di tesi è dunque orientato in questa direzione: valutare quantitativamente diversi algoritmi di integrazione ed eventualmente fornire suggerimenti per i biologi che intentono scoprire il ruolo dei miRNA attraverso l'analisi dei target previsti.

In questo studio abbiamo deciso di analizzare l'impatto che i modelli proposti hanno sull'integrazione, cercando di spaziare il più possibile tra le diverse tecniche sviluppate, in manera tale da renere interessante ed esaustivo il confronto. Per fare questo, ci siamo avvalsi di procedimenti già utilizzati in esperimenti per la previsione degli appaiamenti miRNA-mRNA, e altri presi in prestito da differenti contesti biologici, come la creazione dei network genici. Inoltre abbiamo valutato uno degli aspetti ancora poco considerati, ossia la relazione molti-a-molti, attraverso due articoli di recente pubblicazione che utilizzando i cosidetti modelli di restringimento.

Le metodologie che verranno presentate nei prossimi capitoli possono essere quindi categorizzati sulla base dell'innovazione apportata ("stato dell'arte o nuove proposte") o della direzione della relazione esaminata ("uno-a-uno" o "molti-a-molti"); per una maggiore coerenza con l'obiettivo del lavoro di tesi si è scelta la prima suddivisione.

3.1 STATO DELL'ARTE

In questo capitolo presentiamo le metodologie che già sono state applicate al fine di integrare predizioni ed espressione. In particolare, la correlazione di Pearson, la Mutua Informazione, e infine la regressione Lasso, nelle due varianti proposte da diversi autori.

3.1.1 La Correlazione di Pearson (R)

L'indice di correlazione di Pearson [35] è sicuramente il metodo più utilizzato in moltissimi contesti per la quantificazione della relazione tra due variabili, e quello biologico non fa certo eccezione. Si tratta infatti di una misura che può essere applicata con facilità a dati continui e che richiede un numero relativamente poco elevato di campioni.

Numerosi studi [36][37][38] hanno dimostrato come l'azione di inibizione post-trascrizionale dei miRNA possa essere quantificata con successo attraverso questo indice sfruttando i profili di espressione. A questo proposito sono stati realizzati web tool che permettono di utilizzarlo, uno dei primi è HOCTAR[39]. Poichè queste procedure sono automatizzate, e funzionano come una sorta di "scatola nera" (integrando spesso anche altre informazioni), ai fini del confronto, in questo lavoro di tesi abbiamo deciso di implementare autonomamente il coefficiente di correlazione di Pearson attraverso la

funzione *corr()* della libreria standard di R.

Formalmente, siano G e M, due variabili aleatorie normali di cui si dispone di uno stesso numero limitato di campioni pari ad N, la correlazione di Pearson per una generica coppia mRNA(g_i)-miRNA(m_j) di profili d'espressione, può essere calcolata mediante il seguente stimatore:

$$r_{ij} = \frac{\frac{1}{N-1}\sum_{n=1}^{N}(g_{in} - \overline{g_i})(m_{jn} - \overline{m_j})}{\sqrt{\frac{1}{N-1}\sum_{n=1}^{N}(g_{in} - \overline{g_i})^2}} \tag{1}$$

dove $\overline{g_i}$ e $\overline{m_j}$ rappresentano le medie aritmetiche dei profili di g_i e m_j. Ovviamente è una misura simmetrica, ossia $r_{ij} = r_{ji}$ e assume valori compresi tra -1 e 1. Inoltre, è bene sottolineare che la correlazione di Pearson, è un indice parametrico che individua relazioni di tipo escusivamente lineare tra due variabili.

Un'importante proprietà del coefficiente di correlazione di Pearson è l'esistenza di una distribuzione esatta[40] che ci permette di valutare la significatività della relazione calcolata, sotto un ipotesi nulla. In particolare, nel nostro lavoro, assumento che i legami interessanti siano quelli inversi, andremo a vagliare per ciascun coefficiente r_{ij} calcolato, il seguente sistema di ipotesi:

$$\begin{cases} H_0 : r_{ij} = 0 \\ H_1 : r_{ij} < 0 \end{cases}$$

Sappiamo infatti che sotto l'ipotesi nulla H_0, r_{ij} si distribuisce come una distribuzione T di Student con $n-2$ gradi di libertà, secondo la formula:

$$t^{oss} = \frac{r_{i,j}}{\sqrt{1 - r_{i,j}^2}}\sqrt{n-2} \sim \mathbf{t_{n-2}} \tag{2}$$

Di contro alla semplice formulazione e alla conoscenza della distribuzione nulla, va ricordato che la correlazione di Pearson non è in grado di distin-

guere tra interazioni di tipo indiretto e diretto. Infatti, un elevato valore di $|r_{ij}|$ tra due variabili non è necessariamente indicativo di un'interazione diretta di regolazione tra un miRNA un mRNA: i profili possono essere altamente correlati anche nel caso in cui interagiscano in maniera indiretta, ad esempio per la presenza di un regolatore comune (come un fattore di trascrizione), o perchè la loro interazione è mediata da una terza variabile (come un secondo miRNA). Per superare, almeno in parte, questo limite si proporrà tra le nuove proproste un coefficiente di correlazione parziale (Paragrafo 3.2.1).

3.1.2 La Mutua Informazione (MI)

La Mutua Informazione, inizialmente nata nel contesto della teoria delle informazioni, si è poi affermata anche in quello biologico e biomedico fornendo buone performance nella previsione della struttura secondaria dell'RNA, nella ricostruzione delle reti geniche, nell'imaging biomedico per la registrazione delle immagini e anche nella predizione degli mRNA target. In particolare, di recente sviluppo è il web tool MAGIA[18]: un nuovo strumento che consente di integrare le predizioni dei target con i profili di espressione, usando tra le diverse misure possibili proprio la Mutua Informazione.

La Mutua Informazione è una misura della mutua dipendenza tra due variabili. Intuitivamente, cattura l'informazione che una variabile X e una variabile Y condividono: essa valuta quanto la conoscenza di una di queste variabili riduce la nostra intercezza rispetto all'altra. Può essere quindi interpretata come una misura di correlazione generalizzata, analogamente alla correlazione di Pearson, ma sensibile a qualsiasi relazione funzionale, non solo dunque alla dipendenza lineare.

Formalmente, la Mutua Informazione di due variabili casuali continue G

e M può essere definita come:

$$MI(G, M; b) = \int_M \int_G p(g, m) \log_b \left(\frac{p(g, m)}{p_1(g) p_2(m)} \right) \quad dgdm \tag{3}$$

dove $p(g, m)$ è la densità di probabilità congiunta delle variabili casuali G e M, $p_1(g)$ e $p_2(m)$ sono le funzioni di densità di probabilità marginale rispettivamente di G e M, e b la base del logaritmo (solitamente pari a 2). Nel caso i profili di espressione di G e M siano indipendenti, allora la conoscenza di G non darà nessuna informazione riguardo ad M - e viceversa, essendo una misura simmetrica -, perciò la loro informazione sarà zero; nel caso di dipendenza invece otteremo un valore maggiore di zero. Appare evidente quindi che la mutua informazione restuitirà solo valori positivi, il che si tradurrà nel contesto in esame, in una impossibilità nel distinguere tra regolazioni positive e negative.

Nella letteratura, sono stati proposti diversi approcci alla stima delle densità da utilizzare nel calcolo della Mutua Informazione, e tutte queste richiedono per l'implementazione un certo sforzo computazionale. Non essendo questo lo scopo della tesi, per calcolare la MI per i dati dell'EOC ci siamo limitati, in questo caso, ad avvalerci della versione aggiornata di Magia [41]. Una breve "guida" all'utilizzo del web tool è riportata in Figura 5.

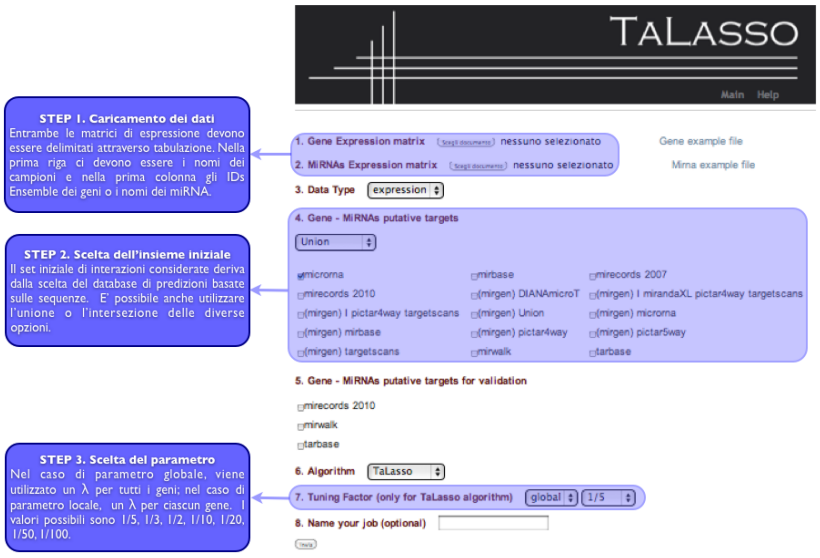
Questo strumento segue l'approccio proposto da Kraskov[42] basato sulla stima delle entropie, utilizzate nella Formula (3) in sostituzione delle funzioni di densità come misura dell'incertezza delle variabili, attraverso la statistica k-NN. In particolare, il calcolo della Mutua Informazione è data dalla stima separata di $H(G)$, $H(M)$ e $H(G, M)$, combinate poi usando la formulazione:

$$MI(G, M; k) = H(G; k) + H(M; k) - H(G, M; k) \tag{4}$$

La qualità dello stimatore $MI(G, M; k)$ sarà ovviamente relazionata al valore k di "vicini" utilizzati per la stima delle entropie: con valori piccoli di

Figura 5: Illustrazione dei passaggi chiave per l'utilizzo del web tool Magia[2], disponibile all'indirizzo http://gencomp.bio.unipd.it/magia

Figura 6: Illustrazione dei passaggi chiave per l'utilizzo del web tool Talasso disponibile all'indirizzo *http://talasso.cnb.csic.es/*.



k, lo stimatore avrà grande varianza e piccola distorsione; con valori grandi di k, si otterrà il risultato opposto. Seguendo le indicazioni dell'articolo da cui è tratto il metodo[42], Magia utilizza un unico numero k di "vicini" pari a 5.

3.1.3 Metodi di selezione delle variabili

Un modello consistente per interpretare l'idea centrale secondo cui differenti miRNA co-regolano uno stesso mRNA target (competendo nell'inbizione post-trascrizionale) è quello della regressione lineare multivriata.
Supponendo che g_i sia il profilo di espressione di un certo gene, e questo sia un'ipotetico target dei corrispondenti J_i miRNA previsti, possiamo interpretare questa relazione secondo la formua:

$$\mathbf{g}_i = \beta_{0i} + \sum_{j=1}^{J_i} \beta_{ji} \cdot \mathbf{m}_j + \epsilon_i \tag{5}$$

dove $\mathbf{g}_i$ e $\mathbf{m}_j$ sono i vettori contenenti i valori di espressione rilevati negli N campioni, e i coefficienti β_{ji} esprimono la forza e la direzione della relazione puntativa tra il gene i-esimo e il miRNA j-esimo.

Ci sono due motivi per cui nel contesto biologico le stime ai minimi quadrati del modello di regressione lineare multivariato non sono soddisfacienti e/o utilizzabili. In primo luogo il numero di miRNA che regolano puntativamente un mRNA sono, nella maggior parte dei casi, più del numero dei campioni, e di conseguenza l'equazione (5) diventa un sistema di equazioni indeterminate[7]. Anche quando si è in posssesso di un numero sufficiente di campioni, tali da permetterci di stimare i coefficienti, le stime ottenute risultano poco accurate a causa della comunque esigua numerosità campionaria.

[7] Un'equazone si dice indeterminata quando ha infinite soluzioni.

La seconda ragione è l'interpretazione: siamo interessati a determinare quel più piccolo sottoinsieme di miRNA responsabile degli effetti più marcati sull'espressione del gene target che condividono. La tecnica utilizzata per la selezione è la regressione stepwise: una procedura automatica che partendo dal modello con la sola intercetta (o completo) verifica se l'inserimento (o l'esclusione) di una variabile è significativamente utile nel prevedere la variabile risposta, attraverso l'utilizzo del test F (o l'ultizzo di altri test). E' stato però dimostrato che questo criterio non porta al miglior modello ottenibile, soprattutto quando il numero di variabili è elevato.

Le prime due tecniche, introdotte per migliorare le stime ai minimi quadrati, che cercano di risolvere i due problemi prima eposti, sono state la selezione subset[43] e la regressione ridge[44], ma entrambe hanno riportato degli svantaggi. La selezione subset riesce a fornire modelli interpretabili ma può essere estremamente variabile perche è un processo discreto: piccoli cambiamenti nei dati possono portare a selezionare modelli molto differenti, riducendo l'accuratezza delle predizioni. La regressione ridge è un processo continuo che costringe i coefficienti e quindi è più stabile: tuttavia, non fissa nessun coefficiente a 0 e quindi non restituisce un modello interpretabile. Per questo Tibshirani ha proposto nel 1994, una tecnica chiamata Lasso.

Il modello Lasso[45](Least Absolute Shrinkage and Selection Operator) cerca di conservare le migliori caratteristiche dei metodi precedentemente descritti, non focalizzandosi su dei sottoinsiemi ma definendo un'operazione di restringimento continua che può produrre alcuni coefficienti esattamente pari a 0, e costringendo altri vicino a questo valore. In particolare, il metodo Lasso grazie alle sue proprietà, è stato dimostrato essere la migliore scelta per individuare la presenza di un piccolo o moderato numero di effetti di media entità, come si suppone essere il processo di regolazione tra miRNA

e mRNA.

Recentemente sono stati pubblicati due articoli che utilizzano proprio il Lasso per l'integrazione dei profili di espressione al fine di individuare le associazioni significative. In entrambi gli studi vengono riportate prove a favore del buon funzionamento del metodo in termini di robustezza e efficacia nell'individuare i target validati, e quindi sono stati oggetti del nostro interesse. Nel seguito proponiamo il modello Lasso rivisitato per i dati di espressione e le due versioni proposte da entrambi gli articoli, mettendo in luce differenze e analogie.

3.1.3.1 Lasso :

Nel primo capitolo abbiamo descritto il meccanismo alla base dell'appaiamento di miRNA e mRNA, ossia la complementarietà di sequenza. Pertanto nell'equazione (5) i regressori di ciascun mRNA saranno ricavati da un sottoinsieme di tutte le possibili iterazioni, attraverso l'utilizzo di un algortimo di predizione basato sulle sequenze. Formalmente, esprimiamo questa idea nel seguente modo:

$$\mathbf{g}_i = \beta_{0i} + \sum_{j=1}^{J_i} \beta_{ji} \cdot c_{ij}^A \cdot \mathbf{m}_j + \epsilon_i \tag{6}$$

dove c_{ij} è una variabile indicatrice che varrà 1 se l'i-esimo mRNA è un potenziale target del j-esimo miRNA, secondo le predizioni del database di sequenze A, e 0 altrimenti.

Il Lasso per ciascun mRNA, propone di risolvere l'equazione (6) usando i minimi quadrati penalizzati regolarizzati tramite la norma L_1 e imponendo una penalità alla grandezza de coefficienti β_{ji} da stimare. Il nostro problema

di ottimizzazione avrà quindi la seguente forma:

$$\min_{\beta_{ji},\beta_{0i}} \left\{ \left\| \mathbf{g}_i - \beta_{0i} - \sum_{j=1}^{J_i} \beta_{ji} \cdot c_{ij}^A \cdot \mathbf{m}_j \right\| + \lambda_i \cdot \sum_{j=1}^{J_i} |\beta_{ji} \cdot c_{ij}^A| \right\} \tag{7}$$

dove λ_i è un parametro della complessità del modello che controlla il restringimento effettuato sui coefficienti: più grande è il valore di λ_j e più i coefficienti saranno costretti ad essere 0.

Nel seguito analizzeremo distintamente i metodi proposti nei due articoli dal momento che ciascuno di questi utilizza integrazioni differenti nella formulazione base (7), proponendo l'inserimento di altre informazioni e/o partendo da assunzioni differenti. Per non crerae confusione rinominiamo le due varianti come TaLasso[46] (TL) e ArgoLasso[47] (AL).

3.1.3.2 TaLasso (TL)

Formulazione Nella proposta di Ander Muniategui e colleghi, la principale variazione consiste nell'aggiunta di un ulteriore vincolo ai parametri β_{ji} per assicurare che il modello restituisca solo le relazioni negative tra mRNA e miRNA. La Formula 7 viene quindi riscritta nel seguente modo:

$$\min_{\beta_{ji},\beta_{0i}} \left\{ \left\| \mathbf{g}_i - \beta_{0i} - \sum_{j=1}^{J_i} \beta_{ji} \cdot c_{ij}^A \cdot \mathbf{m}_j \right\| + \lambda_i \cdot \sum_{j=1}^{J_i} |\beta_{ji} \cdot c_{ij}^A| \right\} \tag{8}$$
$$\text{soggetto a } \beta_{ji} \leq 0, \text{ per } j = 1, 2, \ldots, J_i$$

Algoritmo L'aggiunta di un vincolo si ripercuote sul risolutore utilizzato per il problema di ottimizzazione. Il pacchetto R suggerito è *Rcplex*[48]: un'interfaccia per il programma di ottimizzazine IBM ILOG CPLEX Optimization Studio sviluppato dall'IBM, ottenibile a pagamento. Nonostante esistano risolutori liberamente distribuiti e utilizzabili attraverso il software R che permetterebbero di aggiungere il vincolo proposto, questi, secondo gli

autori, si sono dimostrati meno efficienti nella ricerca del minimo assoluto.

Essendo impossibile riprodurre l'esatto funzionamento di Talasso, siamo stati costretti ad utilizzare lo strumento web messo a disposizione all'indirizzo *http://talasso.cnb.csic.es/* come una sorta di "scatola nera". Quindi il reale funzionamento dell'algoritmo non è stato verificato, ma ci si è limitati nel seguito a riportare quanto descritto nell'articolo di riferimento. Una breve "guida" all'utilizzo del web tool è riportata in 6.

Scelta del parametro λ La teoria del Lasso[49] afferma che il possibile valore del parametro di lisciamento λ_i deve trovarsi all'interno dell'intervallo $[0, \lambda_i^{max}]$, dove λ_i^{max} è definito come:

$$\lambda_i^{max} = 2 \cdot \max_j \left\{ |\mathbf{g}_i \cdot \mathbf{m}_j^t \cdot c_{ij}^A| \right\} \tag{9}$$

Ai due estremi, se il parametro λ_i è uguale a zero, otteniamo la soluzione standard dei minimi quadrati, mentre se il parametro λ_i è uguale a λ_i^{max} la soluzione ottima è il vettore nullo. Calcolando i λ_i in maniera indipendente per ciascun mRNA il modello ottenuto tende a sovra-adattarsi ai dati a causa del gran numero di parametri addizionali che richiede. Quindi, per ogni mRNA, gli autori propongono di selezionare il parametro di lisciamento come una frazione κ, comune a tutti, del valore massimo Λ_i, definita come $\kappa = \lambda_i / \Lambda_i$.

Talasso fornisce due possibili vie per selezionare Λ_i: il metodo globale

$$\Lambda_i = \Lambda = \max_i(\lambda_i^{max}) \quad \Rightarrow \quad \lambda_i = \lambda^G = \kappa^G \Lambda = \kappa^G \max_i\{\lambda_i^{max}\} \tag{10}$$

e il metodo locale

$$\Lambda_i = \lambda_i^{max} \qquad \Rightarrow \qquad \lambda_i = \lambda_i^L = \kappa^L \lambda_i^{max} \qquad \forall i \tag{11}$$

Come si vede nelle Formule (10) e (11), nel caso di parametro di lisciamento globale avremo un λ comune a tutti gli mRNA, mentre nel caso di parametro di lisciamento locale avremo un λ per ciascun mRNA. In ogni caso la frazione κ^L o κ^G sarà unica.

In realtà, per i dati dell'EOC, a causa dell'elevato costo computazionale, il web tool non riesce a fornire il risultato in caso di scelta del parametro locale. Quindi nel nostro lavoro verrà utilizzato solo il metodo globale. Inoltre, nonostante l'articolo suggerisca la convalida incrociata Leave One Out (LOOCV) come metodo per la scelta della frazione ottimale κ, non potendo utilizzare direttamente il codice R, non è possibile riprodurre il procedimento. La nostra scelta ricadrà sul parametro globale κ^G che fornirà i migliori risultati in termini di specificità: questo ovviamente porterà ad una stima "ottimistica" dei coefficienti β_{ji}.

Significatività dei coefficienti Nel contesto dei minimi quadrati regoralizzati con parametro di lisciamento non esiste nessun metodo che provveda al calcolo della significatività statistica della stima.

Gli autori propongono di utilizzare i p-value relativi ai coefficienti ottenuti attraverso la regressione lineare multipla (5), risolta con l'utilizzo dei minimi quadrati, utilizzando le sole relazioni mRNA-miRNA a cui l'algoritmo TaLasso assegna un β_{ij} minore di zero: il relativo p-value viene assegnato alla relazione puntativa. Questi valori non sono restituiti dalla procedura web, ma sono utilizzati all'interno dell'algoritmo per includere nei risultati forniti solo quelle interazioni che, con una confidenza dello 0.95, risultano significativamente minori di zero.

.

3.1.3.3 ArgoLasso (ALM - ALO)

Formulazione Lu e colleghi nel modello (7), oltre all'integrazione dei profili di espressione, aggiungono un ulteriore livello di informazioni. Poichè è stato dimostrata la centralità del RISC e delle proteine Argonaute nel processo di appaiamento tra miRNA e mRNA target, assumono che la diversa concentrazione delle proteine AGO 1- 3-4 influenzi la capacità di AGO 2 di legare il RISC e sfaldare l'mRNA target.

Formalmente, questa idea viene espressa attraverso il seguente modello:

$$\mathbf{g}_i = \beta_{0i} + \sum_{j=1}^{J_i} \beta_{ji,2} \cdot c_{ij}^A \cdot Ago_2 \cdot \mathbf{m}_j + \sum_{j=1}^{J_i} \beta_{ji,134} \cdot c_{ij}^A \cdot Ago_{134} \cdot \mathbf{m}_j + \epsilon_i \quad (12)$$

e trasformando il problema di minimo in:

$$\min_{\beta_{ji},\beta_{0i}} \left\{ \left\| \mathbf{g}_i - \beta_{0i} - \sum_{j=1}^{J_i} \beta_{ji,2} \cdot c_{ij}^A \cdot Ago_2 \cdot \mathbf{m}_j - \sum_{j=1}^{J_i} \beta_{ji,134} \cdot c_{ij}^A \cdot Ago_{134} \cdot \mathbf{m}_j \right\| + \lambda_i \cdot \sum_{j=1}^{J_i} |\beta_{ji,2} \cdot c_{ij}^A + \beta_{ji,134} \cdot c_{ij}^A| \right\} \quad (13)$$

Poichè le proteine AGO 1-3-4 sono tutte considerate dei competitori di AGO2 nel proccesso in cui il RISC lega i due filamenti complementari, le espressioni delle tre proteine sono unite in un unico coefficiente AGO 1-3-4. Si assume inoltre che il livello di concentrazione di ciascuna proteina Argonauta sia positivamente relazionato a livello del corrispondente mRNA AGO, e quindi sarà quest'ultimo a essere utilizzato come stima della concentrazioni.

Nessuna indicazione viene però data su come vengono combinati in un unico valore le espressioni dei geni AGO 1-3-4.

Algoritmo Per lo sviluppo dell'algoritmo relativo al metodo ArgoLasso abbiamo scaricato il codice R liberamente distribuito, ottenibile all'indirizzo *http://biocompute.bmi.ac.cn/CZlab/alarmnet/*.

La procedura base, utilizzata per l'implementazione della regressione Lasso e per la selezione delle variabili, si avvale della libreria *lars* e dell'omonima funzione. Quest'ultima prende in input l'intera matrice dei regressori (miRNA) previsti per un determinato gene, e partendo con tutti i coefficienti pari a zero aggiunge di volta in volta il primo β_{ji} stimato che risulta diverso da zero, facendo variare il parametro di lisciamento nell'intervallo [0, +Inf]. La funzione restituisce dunque un numero di modelli pari al numero di regressori contenuti nella matrice iniziale (Algoritmo 3).

Per determinare quale sottoinsieme di variabili è il migliore per prevedere la variabile risposta (mRNA), si stima l'errore commesso da ciascun modello previsto attraverso la statistica C_p e si sceglie quel sottoinsieme che minimizza questo valore. La statistica C_p è calcolata come:

$$C_p(\widehat{\mathbf{g}}_i^k) = \frac{\|\mathbf{g}_i - \widehat{\mathbf{g}}_i^k\|^2}{\widehat{\sigma}^2} - Nc + 2k$$

dove $\widehat{\mathbf{g}}_i{}^k$ è la stima dei valori di espressione del gene i-esimo al k-esimo step, $\widehat{\sigma}^2$ la varianza stimata ai minimi quadrati, N_c la numerosità campionaria e k un termine che rappresenta il numero di coefficienti inseriti nel modello. Quando la numerosità campionaria è particolarmente piccola - minore o uguale al numero delle variabili - la statistica C_p, non può essere calcolata.

Infine, non avendo imposto vincoli ai coefficienti come per il modello TL (8), è necessario selezionare come reali regolatori solo quei miRNA cui corrispondono β_{ji} negativi.

Raffinamenti: ***ArgoLasso-MultiRun (ALM) e ArgoLasso-OneRun (ALO)*** Nella pratica, in un contesto come il nostro, in cui la numerosità campionaria è prossima se non inferiore al numero dei regressori inseriti, il risolutore *lars()* si dimostra inefficiente nell'individuazione del vero sottoinsieme di regolatori, restituendo spesso il modello con la sola intercetta anche quan-

do tra i miRNA esistono dei veri regolatori. Per risolvere questo problema gli autori decidono di migliorare la stima adottando una procedura "Multi-Run" eseguendo più volte la regressione Lasso, ma su sottoinsiemi sempre più ristretti dei regressori iniziali.

Il procedimento (Algoritmo 4) può essere sintetizzato nel seguente modo. Vengono creati due gruppi: G_1, in cui inizialmente vengono inserite tutte le variabili, e G_2, un insieme vuoto. Per ciascuna *Run*, viene eseguita la funzione *lars()* utilizzando i regressori all'interno del gruppo G_1: le variabili a cui è assegnato un coefficiente diverso da zero vengono eliminate da G1; le variabili a cui è asegnato un coefficiente strettamente minore di zero vengono spostate in G_2. Questo ciclo si ferma quando non ci sono più variabili in G_1 o quando non vengono più trovate variabili diverse da zero nella stima della regressione Lasso.

Successivamente i coefficienti contenuti in G_2 vengono normalizzati e ordinati attraverso uno score basato sull'ordine del coefficiente, calcolato nel modo seguente, per ogni mRNA i-esimo:

$$RankScore_j = \frac{posizione\ del\ miRNA_j}{Numero\ totale\ di\ coefficienti\ in\ G_2} \times 100$$

dove $RankScore_j$ assumerà valori discreti all'interno dell'intervallo $[0, 100]$. Secondo questo ordinamento a grandi valori assoluti dei coefficienti β_{ji} corrisponderanno sempre grandi valori di score, e viceversa.

Il motivo dell'introduzione di questo score di ordinamento non è specificato dagli autori ma, nel capitolo successivo, proveremo a desumere le intenzioni di tale scelta.

Nella situazione in cui la numerosità campionaria è minore o uguale al numero di campioni, la procedura "*MultiRun*" è leggermente diversa da quanto descritto precedentemente: in ciascuna *Run*, invece di selezionare le variabili con i coefficienti diversi da zero nello step che minimizza il C_p, si seleziona sola la prima variabile che entra nel modello (ossia la più correlata

con la variabile risposta). Quest'ultima viene rimossa da G_1 e si procede in questo modo finchè il numero di variabili non è sufficiente per calcolare la statistica C_p come descritto precedentemente.

Per una maggiore chiarezza, in Figura 7 è riportata una rappresentazione schematica della procedura originale *ArgoLasso-OneRun* (ALO) e della versione migliorata introdotta dagli autori *ArgoLasso-MultiRun* (ALM).

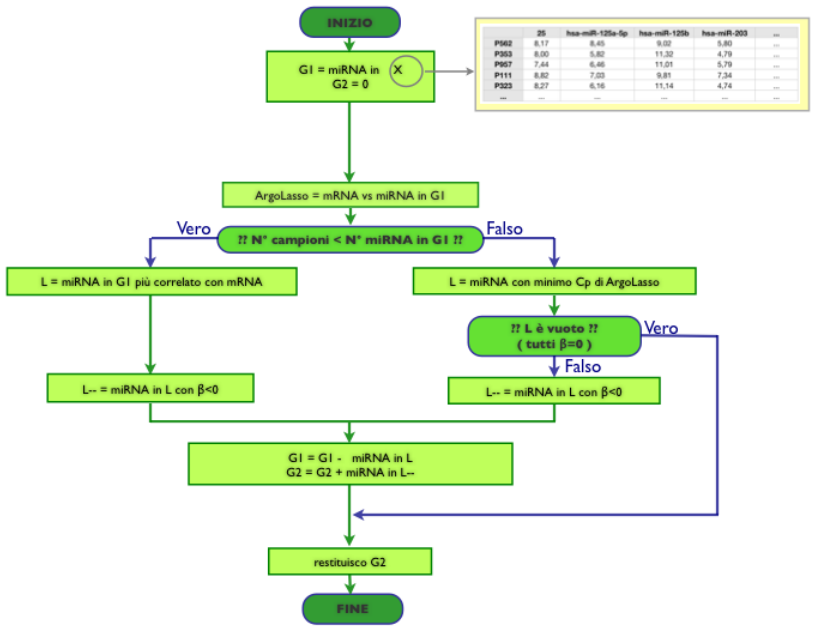
Bootstrap Anche per il modello ArgoLasso - sia nella versione ALM che in quella ALO - non esiste una distribuzione esatta della statistica di interesse, quindi, allo scopo di valutare la specificità dei modelli stimati gli autori si avvalgono del metodo Bootstrap [8].

Data una matrice contenente i profili di espressione dei miRNA associati ad un gene, la procedura consiste nel creare dei dati di controllo randomizzati, ricampionando i valori di uno dei miRNA, estraendolo a caso dall'insieme originale di candidati, ed inserendolo nel gruppo di stima G_1. Così, i valori dei miRNA originali e quello ricampionato sono simultaneamente applicati al modello di regressione.

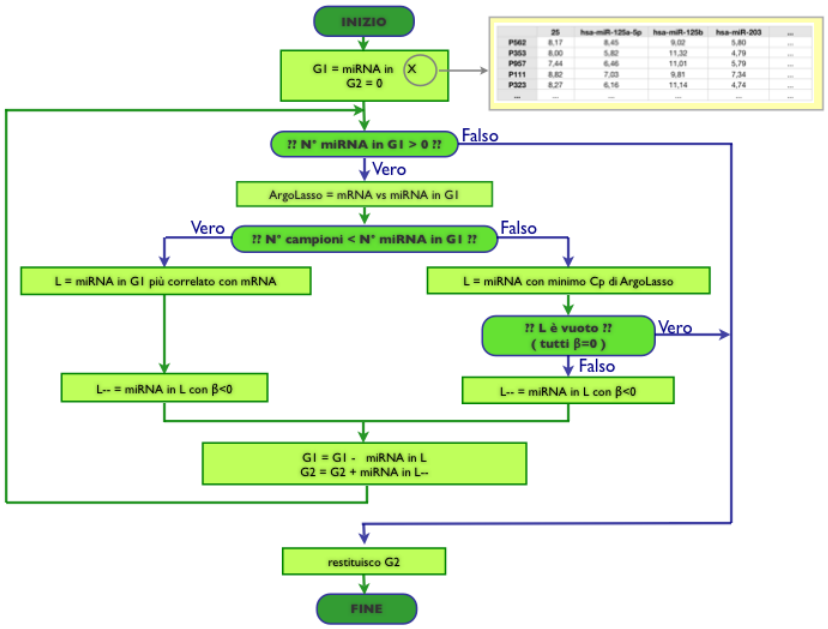
Per ciascun dataset si ripete la procedura più volte, registrando il numero della iterazione e lo score relativo alla variabile randomizzata, quando trovata significativa. Dalla combinazione di questi valori e degli score relativi alle associazioni validate, calcolate sui dati originali, vengono generate le curve ROC e calcolata l'area sottesa dalla curva (AUC, Area Under Curve): più questo indice si avvicinerà a uno, più la soluzione sarà arricchita.

[8] Tecnica statistica di ricampionamente con reimissione utilizzata per approssimare la distribuzione campionaria di una statistica

Figura 7: Rappresentazione sotto forma di diagramma di flusso degli algoritmi per la stima e la selezione dei coefficienti significativi attraverso due versioni del modello ArgoLasso.



(a) ArgoLasso-OneRun



(b) ArgoLasso-MultiRun

3.1.3.4 Penalized ($\mathbf{PZ}^{TL}$ - $\mathbf{PZ}^{AL}$) :

Il fine primo di questo lavoro di tesi è quello di confrontare i vari algoritmi di integrazione. Proponendo due diverse versioni di uno stesso modello matematico, un'aspetto importante da valutare sarà appunto quello di decidere quale dei due ottenga i migliori risultati e soprattutto perchè. In particolare, nell'articolo[46] relativo a TL, si sostiene con forza la tesi secondo cui il miglioramento delle performance rispetto ad AL sono ottenute grazie all'introduzione del vincolo sui coefficienti, visto nella formulazione (8).

Come vedremo nel prossimo capitolo, dove esporremo i risultati ottenuti, il metodo ArgoLasso e il metodo TaLasso non sono paragonabili per differenti motivi, quali la discretizzazione dello score di ordinamento del primo, e la mancanza di un codice sorgente del secondo.

Per ovviare a questo problema, abbiamo deciso di utilizzare un'ulteriore implementazione per la risoluzione del modello di regressione Lasso, che consente l'utilizzo di numerose opzioni, tra cui l'inserimento di vincoli sui β_{ji}. Così facendo, seppur in modo semplicistico, avremo la possibilità di emulare i due metodi e ottenere un confronto diretto che ci permetterà di vagliare la tesi sovracitata.

La funzione di R utilizzata sarà *penalized()*, contenuta nell'omonima libreria. L'algoritmo utilizzato in questa versione di stima penalizzata è documentato nell'articolo pubblicato da Goeman[50], a cui rimandiamo per maggiori dettagli. Oltre alla procedura base, il pacchetto fornisce anche la funzione *optL1()* che provvede a fornire la stima del parametro di lisciamento ottimale tramite convalida incrociata: questa restituisce un λ_i per ciascun mRNA, ciascuno di questi stimato indipendentemente dagli altri.

I due modelli, stimati attraverso questa libreria, utilizzeranno entrambi

la formulazione (7) ma il primo aggiungerà un vincolo sui beta, costringendoli ad essere minori di zero (PZ^{TL}); il secondo invece non imporrà nessuna condizione sui beta, oltre a quella prevista nel metodo di restringimento, e selezionerà solo a posteriori le associazioni con forza inversa (PZ^{AL}). Come i precedenti modelli, anche PZ, non fornisce nessuna significatività per i coefficienti stimati.

3.2 "NUOVE" PROPOSTE

In questo capitolo presentiamo tre metodi già proposti nell'analisi di dati provenienti da esperimenti di microarray, quindi già sviluppati per risolvere i problemi legati alla bassa numerosità campionaria ed alto numero di variabili, ma utilizzati principalmente per la costruzione delle reti geniche, e mai per l'integrazione delle previsione delle associazioni miRNA-mRNA. In particolare nel seguito proponiamo l'indice di correlazione Parziale, il coefficiente di correlazione di Gini e il GlobalTest.

3.2.1 La Correlazione Parziale (RP)

Per misurare la relazione funzionale tra un miRNA e il suo ipotetico mRNA bersaglio e superare, almeno in parte, i limiti illustrati per il coefficiente di Correlazione di Pearson Paragrafo 3.1.1, è possibile utilizzare quello che viene definito il coefficiente di Correlazione Parziale: questo indice quantifica il grado di correlazione tra due varabili condizionandosi rispetto ad una o più delle restanti variabili in gioco. Il numero di variabili rispetto al quale viene eseguito il condizionamento definisce l'ordine del coefficiente di Correlazione Parziale.

Nelle analisi dei dati di microarray e di genomica funzionale la stima della matrice delle covarianze parziali è stata spesso utilizzata per il clustering o per la ricostruzione dei network genici [51][52][53], ottenendo buoni risultati.

In realtà, nel contesto delle integrazioni delle predizioni dei miRNA target, questo indice non rientra propriamente nemmeno nelle "nuove proposte". Infatti, uno studio ha già fatto uso della correlazione parziale come misura di associazione tra profili[54]. Ma, sembra giusto sottolineare che, seppur esistano dei precedenti, mai nessuno ha stimato il coefficiente di Correlazione Parziale in presenza di una così elevata numerosità di accoppiamenti: l'articolo prima citato utilizza solamente un piccolo sottoinsieme dei profili raccolti, pari a 71 mRNA e 31miRNA; il nostro database invece contiene 13795 profili di mRNA e 234 di miRNA.

Per questo motivo, ci sentiamo di dire che il precedente studio parte da presupposti concettualmente diversi e che il lavoro svolto fornirà comunque un progresso riguardo alla conoscenza del metodo in questo contesto.

Formalmente, siano G, M_1 e M_2 tre variabili aleatorie normali, la Correlazione Parziale (RP) del primo ordine, per la coppia G e M è definita come

$$rp_{g\,m_1|m_2} := \frac{r_{g\,m_1} - r_{g\,m_2} \cdot r_{m_1\,m_2}}{\sqrt{(1 - r_{g\,m_2}) \cdot (1 - r_{m_1\,m_2})}} \tag{14}$$

si ottiene cioè condizionando la coppia (g,m_1) rispetto ad una terza variabile m_2: se il valore di $rp_{g\,m_1|m_2}$ risulta prossimo a zero significa che la correlazione esistente le due variabili è unicamente dovuta alla correlazione che entrambi hanno con m_2 e quindi la coppia si può definire condizionatamente indipendente. La definizione di Correlazione Parziale può essere estesa ad ordini superiori, qualora il condizionamento rispetto ad una sola variabile non fosse sufficiente.

La difficoltà legata all'ottenimento di una stima affidabile della matrice delle Correlazioni Parziali, ha portato molti ricercatori a considerare solo ordini limitati; hanno comunque dimostrato che, nel caso della costruzione delle reti geniche, per individuare relazioni affidabili è suffciente considerare misure di Correlazione Parziale fino al secondo ordine [55][56][57].

Tuttavia, in questo lavoro, si è scelto di applicare dei condizionamenti di grado superiore per cercare di spiegare la relazione molti-a-molti alla base dell'effetto di regolazione svolto dai microRNA. Dato un mRNA e un set di miRNA puntativi, vogliamo dunque calcolare la Correlazione Parziale per ciascuna coppia possibile condizionandoci ai restanti miRNA; selezionando poi quelle relazioni che presentano un CP significativamente minore zero saremo capaci di distinguere la relazione di legame diretta da quella indiretta.

La teoria standard dei *Graphical Models*[9] mostra che la matrice delle Correlazioni Parziali ($\mathbf{P}$) è relazionata all'inversa della matrice delle covarianze[58] ($\mathbf{R}^{-1}$=$\mathbf{\Omega}$). Assumento che i dati di espressione siano estratti da una distribuzione normale mutlivariata e che la matrice di Correlazione sia invertibile, la relazione avviene secondo il legame

$$rp_{g\,m_1} = \widetilde{r}_{g\,m_1} = \frac{-\widehat{\omega}_{g\,m_1}}{\sqrt{\widehat{\omega}_{gg} \cdot \widehat{\omega}_{m_1 m_1}}} \tag{15}$$

dove $\widehat{\omega}_{ij}$ è l'elemento in posizione $[i, j]$ della matrice Ω.

Sfruttando la relazione (15), $\mathbf{P}$ può essere calcolata come la correlazione tra i residui ϵ_g e ϵ_{m_1} risultanti dalla regressione lineare di g_i e m_j rispetto ai profili dei restanti $(J_i - 1)$ miRNA: questa formulazione però eredita tutti i problemi discussi nel Paragrafo 3.1.1 relativi alla regressione lineare multivariata, risolta con l'utilizzo dei minimi quadrati. Infatti, quando il numero di variabili per cui si effettua il condizionamento è maggiore al numero di

[9] Il graphical model è un modello probabilistico che utilizza i grafi per rappresentare la struttura di dipendenza condizionata tra variabili casuali.

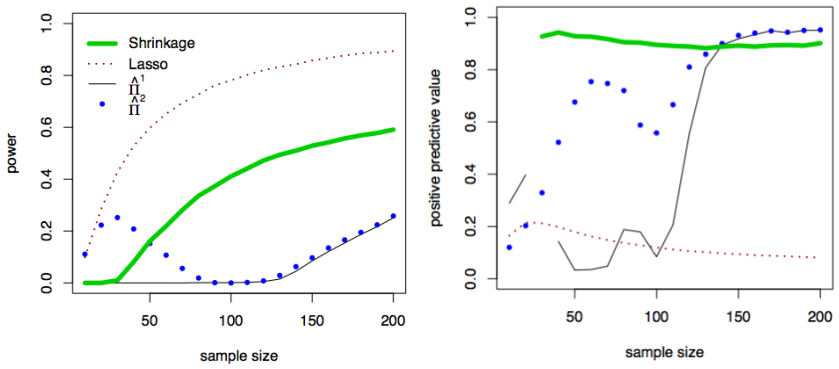
campioni disponibili, la matrice $\mathbf{R}$ non è invertibile e la formula non può essere applicata. Anche in questo caso, la soluzione più immediata è quella di utilizzare una versione di regressione penalizzata, come il Lasso o la regressione Ridge.

In questo lavoro utilizziamo uno stimatore appositamente ideato per il calcolo della matrice $\mathbf{P}$[59] che, confrontato sia con le soluzioni prima citate (Lasso e Ridge Reggresion) che con altri stimatori proposti, risulta essere la migliore scelta in termini di performance in caso di bassa numerosità campionaria e alto numero di regressori (Figura 8).

Tale stimatore ($\mathbf{S}^*$) utilizza un approccio di restringimento lineare secondo la formula:

$$\mathbf{S}^* = \lambda \cdot \mathbf{T} + (1 - \lambda) \cdot \mathbf{S} \tag{16}$$

Figura 8: Grafici tratti dall'articolo [59] relativi al confronto di quattro metodi proposti per la stima della matrice di Correlazione Parziale. Nel primo grafico (a) la potenza è definita come la frazione di veri positivi sul totale di quelli trovati; nel grafico (b) i veri positivi trovati sono definiti come il numero di veri positivi sul totale dei positivi esistenti. Il metodo utilizzato in questo lavoro di tesi è denominato *Shrinkage* e rappresentato dalla curva verde: confrontando i grafici appare evidente come quest'ultimo per piccole numerosità campionarie mostri le migliori performance.



(a) Potenza (b) Veri positivi previsti

Figura 9: Formule per il calcolo dello stimatore $\mathbf{S}^*$. I coefficienti s_{ij} e r_{ij} rappresentano rispettivamente la varianza empirica non distorta e la correlazione semplice. Per ulteriori dettagli sul calcolo delle altre misure, si veda l'articolo di riferimento [59].

"Small n, Large p" Covariance and Correlation Estimators $S^\star$ and $R^\star$:

$$s_{ij}^\star = \begin{cases} s_{ii} & \text{if } i = j \\ r_{ij}^\star \sqrt{s_{ii} s_{jj}} & \text{if } i \neq j \end{cases}$$

and

$$r_{ij}^\star = \begin{cases} 1 & \text{if } i = j \\ r_{ij} \min(1, \max(0, 1 - \hat{\lambda}^\star)) & \text{if } i \neq j \end{cases}$$

with

$$\hat{\lambda}^\star = \frac{\sum_{i \neq j} \widehat{\mathrm{Var}}(r_{ij})}{\sum_{i \neq j} r_{ij}^2}$$

dove $\mathbf{S}$ rappresenta la stima empirica non distorta della matrice delle covarianze, e $\mathbf{T}$ il Target, che rappresenta una versione ridimensionata di quest'ultima. Il nuovo stimatore così definito, attraverso una media pesata, cerca un semplice compromesso tra la alta varianza e bassa distorsione di $\mathbf{S}$ e la bassa varianza e grande distorsione di $\mathbf{T}$.

Il vantaggio principale di questa formulazione è che, scegliendo un adeguato Target, la matrice di covarianza $\mathbf{S}^*$ risulta essere sempre definita positiva e ben condizionata[10]. Questo ci permette di utilizzare gli elementi di $\mathbf{S}^*$ nella formulazione (15), in sostituzione dei valori ω_{ij}, ottenendo $\mathbf{P}$ attraverso la sua inversione. Di contro, lo svantaggio nell'utilizzo di questo stimatore, è la mancanza di una misura di significatività dei coefficienti facilmente calcolabile. Infatti, l'algoritmo di stima della significatività proposto è non banale e computazionalmente troppo oneroso per essere implementato con gli strumenti a disposizione.

[10] Un problema si dice ben condizionato quando con delle piccole variazioni la soluzione non differisce molto da quella del problema originale.

In Figura 9 riportiamo un riassunto delle misure finali da calcolare per ottenere la stima $\mathbf{S}^*$: si rimanda all'articolo di riferimento del metodo per ulteriori approfondimenti sulla scelta del parametro λ ottimale e della matrice Target utilizzata.

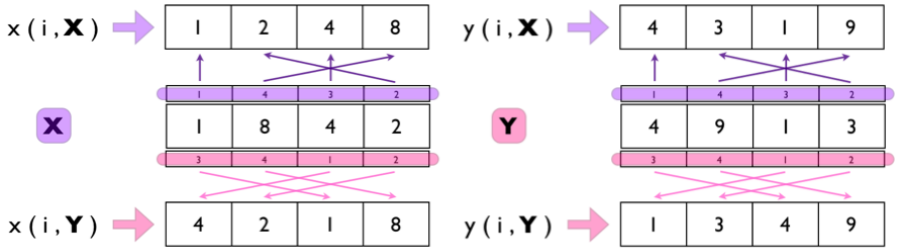
Il procedimento prima descritto è stato comodamente implementato dagli autori con il linguaggio R nel pacchetto *GeneNet*, scaricabile dall'archivio di *Bioconductor*. La stima dei coefficienti di Correlazione Parziale delle nostre coppie di associazioni miRNA-mRNA sarà ottenuta attraverso un'adattamento della funzione *pcor.shrink()*.

3.2.2 L'Indice di Correlazione di Gini (GC)

Il coefficiente di Correlazione di Gini è un membro della famiglia delle metodologie di Gini, ampliamente utilizzato nelle più disparate discipline, prima fra tutte l'economia, ma anche in sociologia, psicologia, ingegneria, informatica e ovviamente anche in biologia. Come la Mutua Informazione permette di catturare relazioni anche non lineari, ma al suo contrario, il range di variazione è l'intervallo $[-1, 1]$, rendendo così possibile distinguere la direzione della relazione: mentre 0 indica l'assoluta indipendenza delle variabili, -1 e 1 indicano l'assoluta relazione monotonica decrescente o crescente.

L'aspetto principale che distingue l'indice di Gini dagli altri metodi (Correlazione di Perason, Mutua Informazione, ma anche correlazione dei ranghi di Spearman o di Kendall) è l'integrazione non solo del profilo di espressione ma anche del rank, definito come la posizione del singolo valore nel profilo ordinato in senso crescente. In questo modo, il coefficiente di correlazione di Gini è più robusto ai dati non distribuiti normalmente permettendo di slegarsi dalle assunzioni sulla forma della distribuzione dei dati [60]. E' stato

Figura 10: Esempio del calcolo delle misure $x(i,X)$, $x(i,Y)$, $y(i,X)$ e $y(i,Y)$, per due generiche variabili X e Y di dimensione N_c, pari a 4.



dimostrato anche, che l'introduzione del rank, ha molti altri vantaggi tra i quali una maggiore robustezza, con un aumento della tolleranza degli ouliers e una minore dipendenza dalla numerosità campionaria[61].

Queste buone proprietà, dimostrate in recenti studi sui network regolatori, lo rendono un candidato ideale per essere inserito tra le "nuove proposte" di questo lavoro di tesi. La formulazione adottata è stata tratta da uno studio sui dati di espressione genica nell'*Arabidopis* e *Maize*[62] e riproposta in un'articolo di confronto di quattro diversi metodi di correlazione utilizzati nelle analisi di microarray[63].

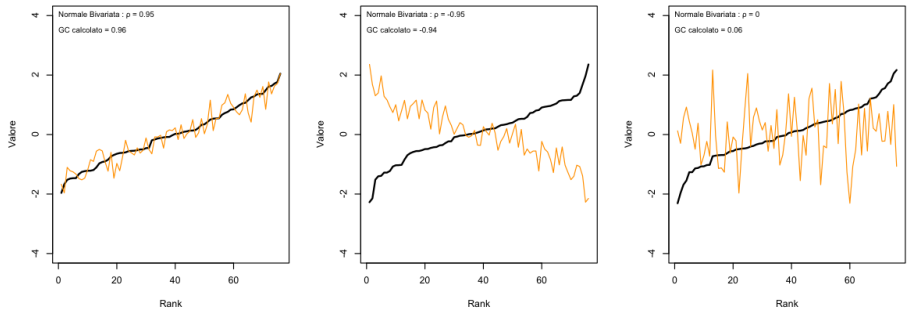
Dati i profili di espressione di due variabili G e M, di cui sono stati osservati N_c valori, il metodo utilizza reciprocamente l'informazione del valore di una variabile e l'informazione del rank dell'altra, producendo due diversi coefficienti di correlazione. Formalmente, definiamo questi due coefficienti come:

$$GC(G,M) = \frac{\sum_{i=1}^{N_c}(2i - N_c - 1)\cdot g(i,M)}{\sum_{i=1}^{N_c}(2i - N_c - 1)\cdot g(i,G)} \tag{17}$$

$$GC(M,G) = \frac{\sum_{i=1}^{N_c}(2i - N_c - 1)\cdot m(i,G)}{\sum_{i=1}^{N_c}(2i - N_c - 1)\cdot m(i,M)} \tag{18}$$

dove $x(i,X)$ rappresenta l'i-esimo valore della generica variabile X ordinata in senso crescente, e $x(i,Y)$ rappresenta il corrispondente valore della gene-

Figura 11: Sono rappresentate le curve relative ai profili simulati di due variabili X e Y, provenienti da una distribuzione Normale Standard, contenenti 76 valori ciascuna. Le tre curve rappresentano rispettivamente il caso di alta relazione lineare positiva, negativa, e di indipendenza. Il valore di $x(i, Y)$ è raffigurato con una curva nera, mentre il valore di $x(i, X)$ con una curva tratteggiata di colore arancione.



rica variabile X ordinato in senso crescente rispetto ai valori della generica variabile Y. Una esempio per il calcolo di queste misure è rappresentato in Figura(10). I due coefficienti GC così ottenuti sono solitamente simili, come il loro p-value.

Secondo le equazioni (20) e (21), possiamo interpretare il metodo GC come la differenza tra due curve pesate per l'informazione derivata dall'ordine del rank dei dati di espressione. In Figura(11) sono rappresentati alcuni esempi nel caso di relazione positiva, negativa o indipendenza, di due generiche variabili estratte da una distribuzione Normale Standard.

Uno svantaggio del GC è l'assenza di una distribuzione nulla di riferimento per vagliare l'ipotesi di significatività dei coefficienti. Come è uso comune in questo contesto, anche Wang e collega, propongono di calcolare il p-value attraverso l'utilizzo della procedura Boostrap (Algoritmo), randomizzando casualmente i dati di espressione della coppia di geni analizzata.

Il coefficiente di Correlazione di Gini, assieme agli altri quattro coefficienti analizzati nell'articolo di riferimento, sono stati implementati con lin-

guaggio R in un pacchetto liberamente scaricabile all'indirizzo *http://cran.r-project.org/web/packages/rsgcc*, denomintao *rsgcc*[11]. A causa di problemi di compatibilità tra la libreria *rsgcc*, la versione di R disponibile (R 1.15.1) e il sistema operativo utilizzato (Mac OS X - versione 10.5.8), è stato necessario installarla e implementarla su sistema operativo Linux.

[11] Attenzione: per utilizzare la libreria *rsgcc* è necessario scaricare *GTK+*: un insieme di strumenti per la creazione di interfacce grafiche.

4 Implementazione & Risultati

Allo scopo di identificare un pannello di microRNA che dimostrino di avere un effetto di regolazione nei confronti del loro gene target, in modo tale da ricondurci al significato biologico, abbiamo utilizzato i dati di espressione ottenuti attraverso piattaforme di microarray su pazienti al primo stadio del cancro ovarico. Il nostro studio è volto a quantificare in maniera contemporanea tutte le relazioni esisteni tra coppie di mRNA e miRNA, attraverso diverse metodologie che utilizzano i profili di espressione.

Partiremo da un set di intereazioni iniziali ottenuti dai database di predizione bioinformatica basati sulla complementarietà delle sequenze. Questa scelta, oltre ad essere in accordo con l'unico meccanismo molecolare noto, risolve anche alcune limitazioni intrinseche dei dati di microarry. Infatti, le tecnologie high-throughput forniscono un numero di trascritti nell'ordine delle decine di migliaia per ciascun esperimento: se da un lato la possibilità di monitorare l'intero trascrittoma permette di considerare il sistema nella sua globalità, dall'altro, il dover esaminare un così elevato numero di possibili combinazioni di appaiamenti miRNA-mRNA, complica notevolemente il problema sopratutto per i tempi di calcolo. Quindi, per ridurre il numero di accoppiamenti da analizzare, utilizzeremo distintamente le predizioni dei databases TargetScan (TS) e microRNA.org (SVR); ma, quando la complessità dell'algoritmo del metodo ce lo permetterà, proveremo ad analizzare ogni possibile combinazione per confermare la correttezza della scelta.

I metodi applicati al caso in esame sono gli stessi precedentemente esposti, ma per rendere agevole l'esposizione da un punto di vista implementativo, in questo capitolo preferiamo suddiverli sulla base del tipo di relazione espressa: per i confronti "pair-wise" utilizzeremo le due matrici principali dei valori di espressione di mRNA e miRNA da cui, attraverso la lista di

predizioni, estrarremo di volta in volta i due profili da computare; per i confronti "multivariati", saremo costretti a costruire una matrice per ciascun mRNA a cui verranno aggiunti tutti i valori dei miRNA che individuano il gene come un possibile target, secondo una qualche lista (Figura 12).

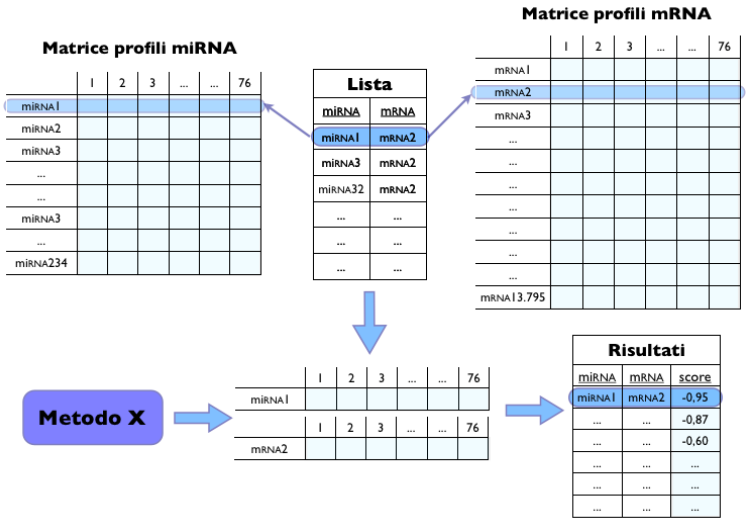
Nel terzo capitolo, sono contenuti i risultati relativi ai metodi per cui ci siamo affidati ad un web tool: per questi non siamo in grado di stabilire come vengano processati i dati, ma verranno descritti i passaggi necessari per l'utilizzazione degli strumenti e l'elaborazione dei risultati ottenuti. In ogni caso, a qualunque categoria appartenga la formulazione che decidiamo di utilizzare, quello che otterremo sarà una lista ordinata di appaiamenti definiti come reali deregolazioni tra un microRNA e il suo gene target, e la misura di questa forza.

Per valutare le prestazioni delle diverse misure proposte ci avvarremo dell'insieme di quelle interazioni validate ottenute dall'unione dei databases mirWalk, TarBase, mirTarBase e miRecord, per un totale di 16.951 interazioni (Paragrafo 2.5). Ad ogni lista restituita dai diversi metodi verrà poi applicata la funzione *ValidaLista*(4): dopo aver ordinato le associazioni fornite rispetto alla direzione decisa dal metodo (crescente/decrescente), la funzione restituisce una variabile indicatrice TRUE o FALSE (validata/non validata), e nel primo caso ulteriori informazioni accessorie, come il database in cui è contenuta la validazione e un'id identificativo da noi definito.

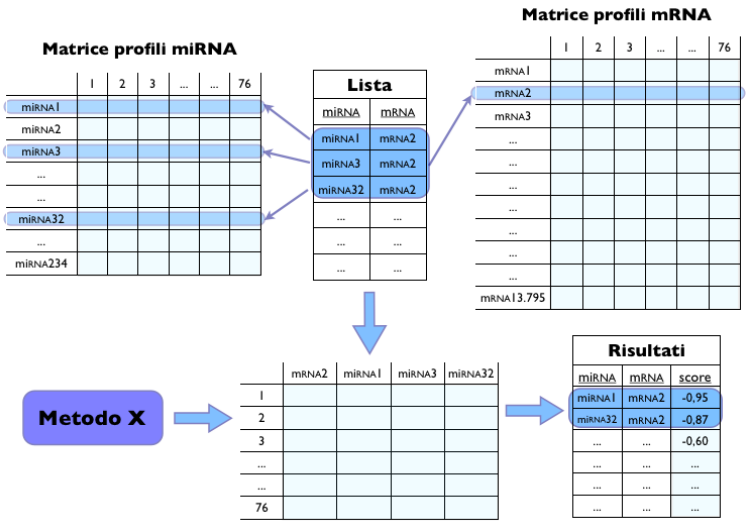
La sintassi del metodo può essere riepilogata nel seguente modo:

ValidaLista(Associazioni, listaValidati, nome.salvataggio=Associazione.txt, opzioneGene=1,
opzioneOrdina1=list(attivo=FALSE,colonna=0,metodo=0),
opzioneOrdina2=list(attivo=FALSE,colonna=0,metodo=0),
opzioneLasso=FALSE, nome.salvataggio.Lasso=")

Figura 12: E' rappresentata schematicamente la procedura di preparazione dei dati nel caso di modelli pair-wise (a) e nel caso di modelli multivariati (b).



(a) Pair-Wise



(b) Multivariato

Input:

Associazioni: matrice in cui ogni riga rappresenta un'associazione; sulla prima colonna devono esserci gli identificativi dei geni e sulla seconda colonna i nomi dei microRNA

listaValidati: matrice in cui ogni riga rappresenta una associazione sperimentalmente validata; nell'ordine le colonne devono contenere un id proprio della validazione; il nome, l'id Entrez, l'Id Ensebl dell'mRNA; una variabile indicatrice (presente/assente) per ciascun databases da cui le validazioni sono state estratte

opzioneGene: codice per scegliere il formato con cui l'identificativo del gene è stato salvato; 1 per il nome GO, 2 per l'id Entrez, 3 per l'id Ensembl

opzioneOrdina1-opzioneOrdina2: insieme di opzioni per l'ordinamento della lista di associazioni fornite; la variabile colonna deve indicare l'indice di colonna dei dati numerici da ordinare; la variabile metodo assume il valore -1 per un ordinamento decrescente e 1 per un ordinamento crescente

opzioneLasso: TRUE o FALSE

Output:

a schermo: viene stampato un riepilogo sulle principali informazioni della lista

file: viene salvato un file .txt denominato con il nome scelto nelle opzioni della funzione, contenente l'intera matrice passata e l'indicazione della validazione di ciascuna associazione; Nel caso di interazione validata viene fornito anche il suo id e le rispettive variabili indicatrici del databases di provenienza.

file: se l'opzioneLasso è uguale a TRUE viene restituito anche un file .Rdata contenente le sole associazioni validate.

Nel seguito, in maniera indipendente per ciascun metodo, riportiamo il procedimento con cui i risultati sono stati ottenuti, le scelte effettuate riguardo ai parametri ed eventuali altre misure di selezione e sintetizziamo attraverso degli indici la performance finale.

Riserviamo al quinto capitolo i confronti.

4.1 Sommario degli indici

Per rendere più agevole l'interpretazione delle misure adottate riserviamo a questa sezione il compito di descrivere nel dettaglio gli indici di cui ci siamo serviti. Ovviamente, si suppone che ciascuna misura venga applicata ad una lista finale precedentemente ordinata in maniera crescente o decrescente - in funzione della misura utilizzata per quantificare la forza di ciascuna relazione miRNA-mRNA.

L'assunzione alla base degli indicatori proposti prevede che, se una lista ordinata contiene nella parte alta della classifica più interazioni validate rispetto alle altre, ci si aspetta che questo algoritmo funzioni generalmente meglio.

Riportiamo quindi nel seguito le abbreviazioni e la descrizione completa degli indici e le notazioni che compariranno nel seguito del lavoro:

N_g/N_m: numero di geni/microRNA rilevati nell'esperimento di microarray.

N_g^{lista}/N_m^{lista}: numero di geni/microRNA rilevati nell'esperimento di microarray e a cui corrisponde almeno una associazione contentuta nella lista di predizioni iniziali specificata.

N^{TS}/N^{SVR}: numero totale di interazioni restituite dal databases di predizioni TargetScan/microRNA.org, riferite ad una coppia miRNA-mRNA realmente rilevata nell'esperimento di microarray.

N_i^{TS}/N_i^{SVR}: numero totale di microRNA associati al singolo mRNA i-esimo, secondo le predizioni restituite dai databases TargetScan/microRNA.org.

t: valore del taglio effettuato per discriminare le interazioni significative da quelle che non lo sono.

N^T: numero totale di interazioni restituite dal metodo senza nessun taglio sulla significatività dei valori.

N^S: numero totale di interazioni che risultano significative, a seguito di un taglio sulla significatività dei valori.

$t^{\%}$: percentuale di interazioni eliminate a seguito del taglio sulla significatività.

V^{tot}: numero di interazioni validate all'interno delle N^S interazioni

V^{500}: numero di interazioni validate all'interno delle prime 500 della lista

V^{1000}: numero di interazioni validate all'interno delle prime 1000 della lista

$\%^{tot}$: percentuale di interazioni validate sul totale N^S

$\%^{500}$: percentuale di validati all'interno delle prime 500 interazioni

$\%^{500}$: percentuale di validati all'interno delle prime 1000 interazioni

4.2 Confronti PAIR-WISE

Il termine "pair-wise" è stato coniato nel contesto della costruzione dei pathway genetici: questi metodi costruiscono le relazioni basandosi unicamente sul confronto di coppie di geni; tali relazioni, sono certamente indicative del sottostante sistema di regolazione, ma non necessariamente rappresentative di un'effettiva interazione a causa della possibilità di un effetto spurio. Questo è esattamente il principio alla base del coefficiente di Correlazione di Pearsone e dell'Indice di Gini - ma anche della Mutua Informazione - e perciò si è deciso di adottare questo nome per la nostra classificazione.

Dovremo quindi individuare di volta in volta, attraverso i suggerimenti di una delle due liste utilizzate in questo lavoro, le coppie miRNA-mRNA all'interno delle matrici dei valori di espressione a cui applicare una misura del tipo:

$$Metodo_i^X(G;M) = score_i^X \qquad \text{con } i \text{ da 1 a } N^{tot}$$

Il numero di operazioni totali da eseguire (N^{tot}), riducendole alle sole coppie previste dai databases TS e SVR e contenute nel nostro esperimento,

saranno rispettivamente pari a 70.610 (N^{TS}) e 309.836 (N^{SVR}). Nel caso in cui ci si svincolerà dall'assunzione della complementarietà di sequenza il numero totale di estrazioni sarà pari a 3.228.030 (N_gxN_m).

4.2.1 Il coefficiente di Correlazione di Pearson (R, R^{TS}, R^{SVR})

Formulazione Lo stimatore utilizzato per il calcolo del coefficiente di Correlazione di Pearson è il seguente:

$$r_{ij} = \frac{\frac{1}{N-1}\sum_{n=1}^{N}(g_{in} - \overline{g_i})(m_{jn} - \overline{m_j})}{\sqrt{\frac{1}{N-1}\sum_{n=1}^{N}(g_{in} - \overline{g_i})^2}} \tag{19}$$

In particolare, dai risultati estrarremo altre due diverse liste di stimatori (R^{TS} e R^{SVR}) che conterranno gli stessi valori calcolati per R ma relativi alle sole coppie reputate come possibili appaiamenti dagli algortimi di predizione basati sulla complementarietà delle sequenze.

Funzione R Per l'implementazione del coefficiente di correlazione di Pearson abbiamo utilizzato la funzione *cor()* contenuta nella libreria standard del linguaggio R. La sintassi della procedura adattata al nostro caso è riportata nel seguito.

cor*(x, y=NULL, use="everything", method=c("pearson","kendall","spearman")*

Input:

x: profilo di espressione del microRNA
y: profilo di espressione del gene
method: pearson

Output:

coefficiente di Correlazione di Pearson

Procedimento Il coefficiente R è stato calcolato per tutte le coppie mRNA-miRNA presenti nel nostro dataset per un totale di 3.228.030 possibili associazioni. Per farlo abbiamo costruito una matrice di dimensione 13.795x234 dove sono stati salvati i risultati ottenuti.

Abbiamo successivamente creato una lista per lo stimatore R salvando solo quei valori r_{ij} che risultassero strettamente minori di zero. Attraverso l'intersezione di R con le predizioni di TS e SVR, abbiamo ricavato altre due liste denominate R^{TS} e R^{SVR}.

Significatività Utilizzando la stessa numerosità campionaria per il calcolo di tutti gli r_{ij}, i valori contenuti nelle tre liste hanno la stessa distribuzione nulla. Abbiamo quindi ricavato attraverso la formulazione (2) il taglio che corrisponde al quantile 0.05 di una distribuzione t di Student con 74 gradi di libertà, pari a -0.1901. Tutti gli appaiamenti a cui corrispondeva un valore superiore o uguale al limite soglia sono stati considerati non rilevanti, e quindi eliminati dalle liste. In Figura 13 sono illustrate le distribuzioni dei coefficienti r_{ij}, r_{ij}^{TS} e r_{ij}^{SVR} e il relativo valore soglia.

Risultati In tabella riportiamo gli indici riassuntivi calcolati.

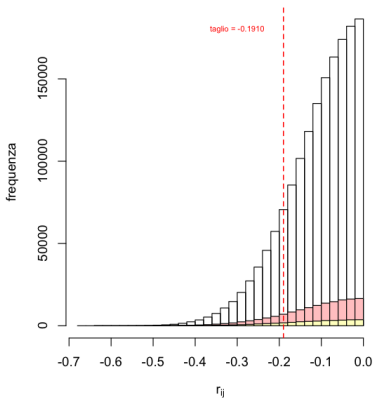
Lista	N^T	$t^{\%}$	N^S	V^{tot}	V^{500}	V^{1000}	$\%^{tot}$	$\%^{500}$	$\%^{1000}$
R	1.602.010	83,30	267.502	2.023	9	20	0,76	1,8	2,0
R^{TS}	36.441	77,20	8.310	393	35	63	4,73	7,0	6,3
R^{SVR}	151.015	80,77	26.215	636	21	37	2,01	4,1	3,7

.

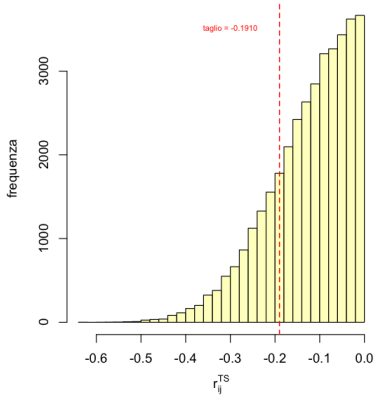
4.2.2 L'indice di Correlazione dei Gini (GC,GC^{TS}, GC^{SVR})

Formulazione L'indice di Correlazione di Gini, utilizzando congiuntamente il valore dei profili di espressione e il loro rank, può produrre due coefficienti di correlazione per una stessa coppia miRNA-mRNA, secondo le

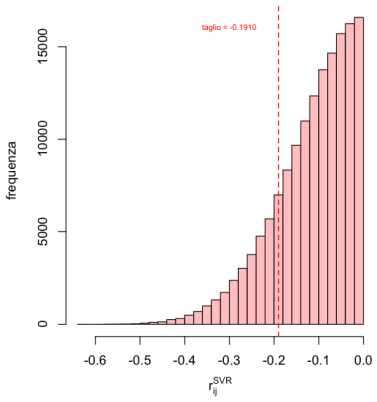
Figura 13: Sono rappresentate le distribuzione dei coefficienti di Correlazione di Pearson minori di zero. Nel primo grafico (a) sono rappresentate con istogrammi bianchi le correlazioni ottenute da tutti i possibili appaiamenti miRNA-mRNA, e con quelli gialli e rossi nell'ordine le correlazioni risultanti dall'intersezione con le liste di predizioni iniziali TS e SVR (rappresentati anche singolarmente nei restanti due grafici (b) e (c)). Con linea rossa tratteggiata è rappresentato il taglio posto sulla significatività.



(a) R



(b) R^{TS}



(c) R^{SVR}

formulazioni:

$$GC(G,M) = \frac{\sum_{i=1}^{N_c}(2i - N_c - 1) \cdot g(i,M)}{\sum_{i=1}^{N_c}(2i - N_c - 1) \cdot g(i,G)} \tag{20}$$

$$GC(M,G) = \frac{\sum_{i=1}^{N_c}(2i - N_c - 1) \cdot m(i,G)}{\sum_{i=1}^{N_c}(2i - N_c - 1) \cdot m(i,M)} \tag{21}$$

Nonostante i valori ottenuti siano solitamente simili, come il loro p-value, abbiamo deciso di ricavare dai risultati tre diverse liste: due che utilizzano rispettivamente solo la Formula 20 o solo la Formula 21 (GC_g GC_m), e una terza che, per ogni coppia di stimatori GC, scelga come valore finale il coefficiente con il più basso p-value (GC). Abbiamo poi intersecato ciascun insieme con i candidati contenuti in TS e SVR ottenendo un totale di nove liste: tre svincolate dall'ipotesi di complementarietà (GC, GC_g, GC_m), e sei date da quest'ultima operazione (GC^{TS}, GC_g^{TS} , GC_m^{TS}, GC^{SVR}, GC_g^{SVR}, GC_m^{SVR}).

Funzione R Per l'implementazione del metodo abbiamo utilizzato il pacchetto *rsgcc* sviluppato dagli autori dell'articolo di riferimento[63]. Tra le varie funzioni offerte dalla libreria abbiamo deciso di utilizzare *cor.pair()*, che rispetta la seguente sintassi:

cor.pair*(idxvec, GEMatrix, rowORcol = c("row","col"),*
cormethod = c("GCC","PCC","SCC","KCC","BiWi"), pernum=0,
sigmethod=c("two.sided","one.sided"))

Input:

idvecx: vettore che contiene gli indici della matrice dei dati a cui corrispondono la coppia di elementi miRNA-mRNA

GEMatrix: matrice dei dati, contenente sulle righe i diversi profili e sulle colonne i diversi campioni

rowORcol: row, calcolo l'indice rispetto alle righe

cormethod: GCC, calcolo l'indice di Correlazione di Gini

pernum: 50, numero di permutazioni per il calcolo della distribuzione nulla
sigmethod: one.sided, alternativa per il calcolo della significatività

Output:

gcc.rankx: GCC(G,M)
gcc.ranky: GCC(M,G)
gcc.rankx.pvalue: p-value relativo al coefficiente GCC(G,M)
gcc.ranky.pvalue: p-value relativo al coefficiente GCC(M,G)

La funzione, data un'unica matrice di correlazione e due indici di riga (o colonna), può calcolare la correlazione di una coppia di profili (o campioni) scegliendo tra cinque diversi metodi di correlazione. Il livello di significatività della correlazione computata può essere stimato con un test di permutazione, e rispetto sia all'alternativa unilaterale che a quella bilaterale.

Procedimento La libreria *rsgcc* è stata ideata per l'analisi dei pattern dell'espressione genica, per questo vuole in ingresso un'unica matrice: tutte le possibili coppie di geni sono contenuti in un unico esperimento di microarray.

Per adattarla al nostro scopo abbiamo dovuto quindi costruire un unico contenitore di 14.029 righe (dove le prime 13.795 posizioni si riferiscono agli mRNA rilevati e le restanti 234 ai profili di miRNA) e 76 colonne rappresentanti i diversi campioni raccolti. Abbiamo così fatto variare gli indici in modo tale da ottenere tutti i possibili 3.228.030 accoppiamenti. Di volta in volta, abbiamo salvato i risultati in un unico file contenente, oltre al nome della coppia per cui era stato calcolato l'indice di Correlazione di Gini, i quattro valori restituiti dalla funzione.

Sono state così create le tre liste principali: nella lista GC abbiamo salvato, per ciascuna coppia, l'indice corrispondente al minor p-value tra i due restituiti; nelle liste GC_g e GC_m gli indici e i p-value che utilizzano

rispettivamente il rank del profilo di G e il rank del profilo di M. Da questi insiemi abbiamo mantenuto esclusivamente i valori dei coefficienti strettamente minori di zero.

Significatività Per calcolare la significatività, la funzione *cor.pair()*, utilizza una stima basata sul metodo Bootstrap: ottiene la distribuzione nulla - e il relativo p-value - attraverso la permutazione dei profili originali. E' risaputo che la bontà della stima ottenuta tramite questa procedura è legata al numero B di replicazioni effettutate: più B è grande e più ci si avvicina al vero valore dell'ignoto stimatore.

Nonostante l'articolo suggerisca di effettuare un numero di ripetizioni pari a 2.000, con i mezzi a nostra disposizione, il processo si è dimostrato eccessivamente dispendioso per essere computato in termini di tempo[12]. Per questo ci siamo limitati ad utilizzare un numero di permutazioni pari a 200, previa verifica che tale numerosità ci permettesse di ottenere una precisione pari al secondo decimale[13].

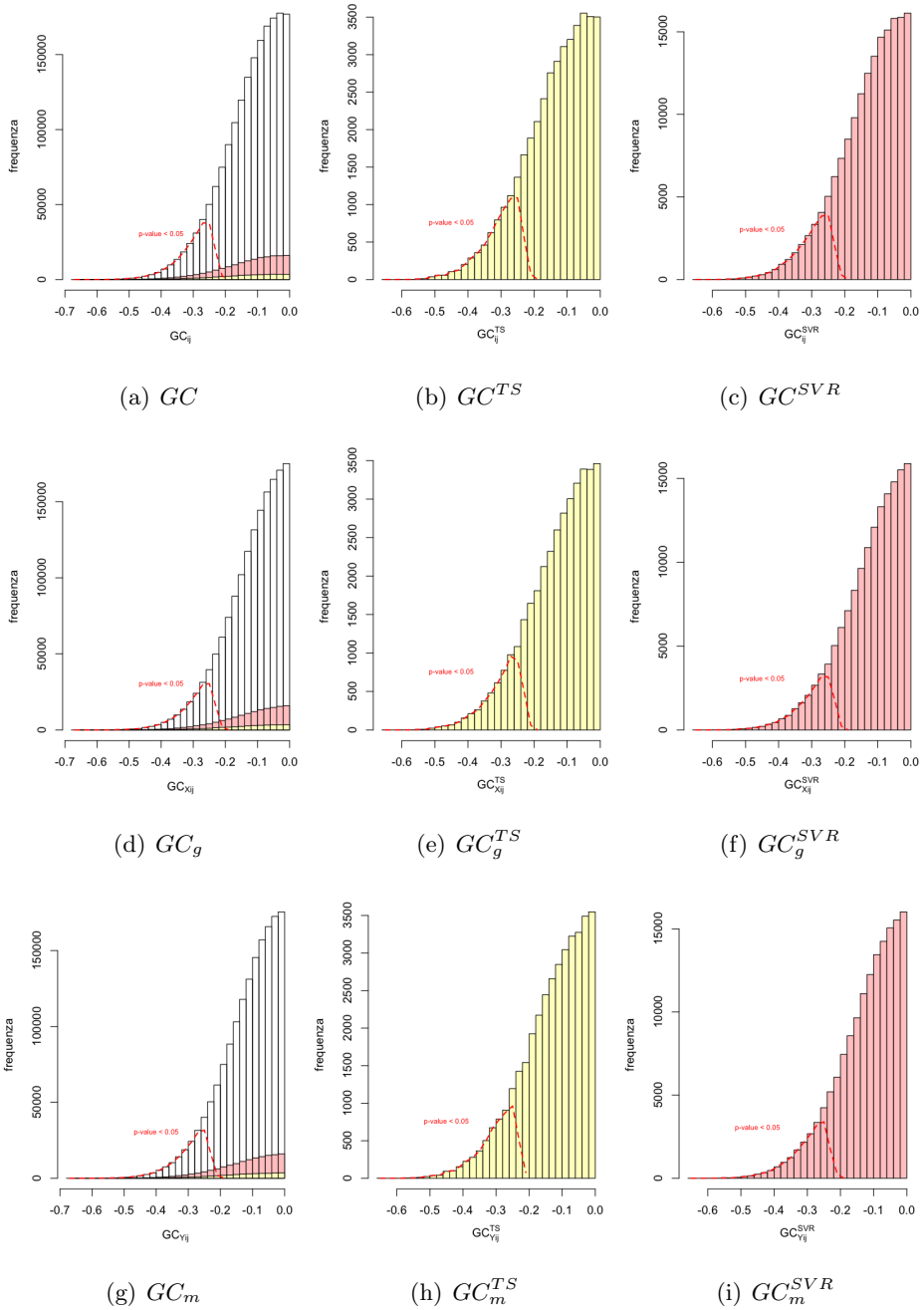
I p-value così ottenuti sono stati utilizzati per discriminare dalle liste GC, GC_g e GC_m le relazioni significative da quelle che non lo erano, scegliendo un livello di pari a 0,95. Infine si è proceduto ad intersecare questi tre insiemi con le predizioni dei database TS e SVR ottenendo le altre sei liste descritte precedentemente (GC^{TS}, GC_g^{TS} , GC_m^{TS}, GC^{SVR}, GC_g^{SVR}, GC_m^{SVR}). In Figura 14 sono illustrate tutte le distribuzioni dei coefficienti calcolati e il relativo taglio basato sul p-value ottenuto con la procedura Bootstrap.

.

[12]Tempo stimato per l'elaborazione di 3.228.030 interazioni pari a circa 110 giorni.

[13]Controllo effettuato su un sottoinsieme di interazioni che comprendeva 4 miRNA e per ciascuno di questi tutti i 13.795 mRNA rilevati, per un totale di 55.180 associazioni.

Figura 14: Sono rappresentate le distribuzioni dei coefficienti di Correlazione di Gini minori di zero. Nei primi grafici sulla sinistra (a,d,g) sono rappresentate con istogrammi bianchi le distribuzioni delle correlazioni ottenute da tutti i possibili appaiamenti miRNA-mRNA per le tre diverse formulazioni proposte - combinazione (20) e (21), formula (20), formula (21) - e con quelli gialli e rossi, nell'ordine, le correlazioni risultanti dall'intersezione con le liste di predizioni iniziali TS e SVR (rappresentati anche singolarmente nei restanti due grafici (b,e,h) e (c,f,i)). Con linea rossa tratteggiata è rappresentato il taglio posto sulla significatività.



(a) GC (b) GC^{TS} (c) GC^{SVR}

(d) GC_g (e) GC_g^{TS} (f) GC_g^{SVR}

(g) GC_m (h) GC_m^{TS} (i) GC_m^{SVR}

Risultati In tabella riportiamo i risultati ottenuti dalle diverse liste.

Lista	N^T	$t^{\%}$	N^S	V^{tot}	V^{500}	V^{1000}	$\%^{tot}$	$\%^{500}$	$\%^{1000}$
GC	1.799.208	88,15	213.304	1.677	7	16	0,79	1,4	1,6
GC_g	1.593.439	89,60	165.649	1.353	10	17	0,82	2,0	1,7
GC_m	1.605.470	89,40	170.047	1.392	5	15	0,82	1,0	1,5
GC^{TS}	40.585	83,05	6.880	351	37	59	5,10	7,4	5,9
GC_g^{TS}	36.425	84,71	5.571	279	33	57	5,00	6,6	5,9
GC_m^{TS}	36.984	84,56	5.709	298	37	63	5,22	7,4	6,3
GC^{SVR}	169.537	83,05	23.707	533	22	45	2,25	4,4	4,5
GC_g^{SVR}	150.562	87,50	18.818	429	25	44	2.28	5,0	4,4
GC_m^{SVR}	153.042	87,33	19.393	446	22	39	2,30	4,4	3,9

.

Come ci sia attende, utilizzando una combinazione dei due coefficienti basati sui diversi rank, il numero totale di coppie significative è maggiore rispetto alle altre due soluzioni, qualsiasi sia la lista a cui ci condizioniamo o meno.

In termini di validati però, non riscontrando una netta preferenza tra le formulazioni (20) e (21) o l'unione delle due, preferiamo rimanere in linea con la metodologia suggerita dall'articolo di riferimento riportando nel seguito esclusivamente i risultati ottenuti nelle liste GC, GC^{TS}, GC^{SVR}.

4.3 Confronti MULTIVARIATI

In questo capitolo presentiamo quei metodi che, per ottenere la forza della relazione tra due variabili, non si limitano ad utilizzare solo i loro profili di espressione ma si avvalgono anche delle informazioni di ulteriori variabili legate alla coppia in un qualche modo: nello specifico il legame che verrà utilizzato sarà la loro complementarietà di sequenza.

Per tutti i modelli, i valori che rappresentano la variabile dipendente saranno i profili di espressione dei geni, mentre le covariate, che nel loro insieme incideranno sull'espressione di quest'ultimi, saranno rappresentate dai microRNA.

La misura ottenuta sarà quindi esprimibile in forma generale come:

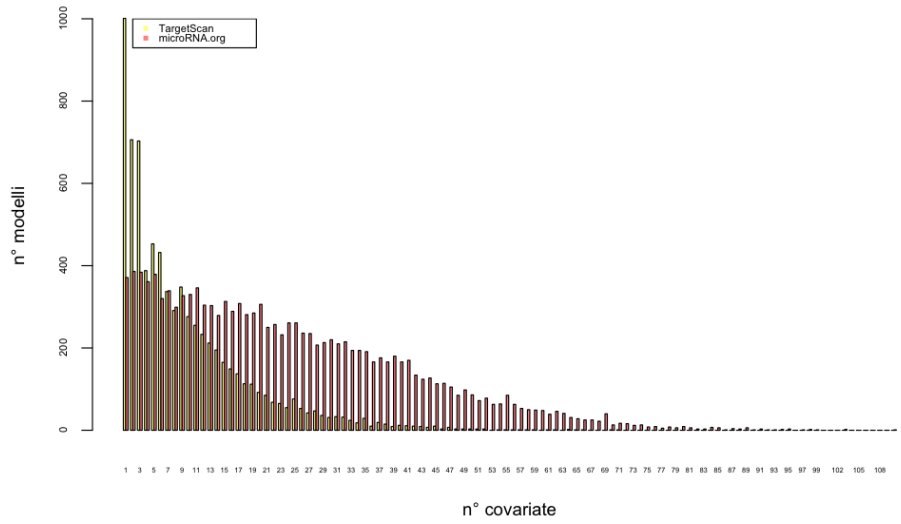
$$Metodo_i^X(G; M_1, ..., M_{N_i}) = [score_1^X, ..., score_{N_i}^X] \qquad \text{con } i \text{ da 1 a } N^{tot}$$

Il numero di modelli da stimare (N^{tot}) dipenderà quindi dal numero di distinti mRNA contenuti nelle liste ottenute attraverso i databases TS e SVR, pari nell'ordine a 12.991 (N_g^{SVR}) e 7.443 (N_g^{TS}) (Figura 15). Quando invece, non assumeremo nessuna relazione legame di base, il numero totale di modelli stimati corrisponderà al numero di mRNA rilevati nel nostro esperimento di microarray, pari a 13.795 (N_g).

La dipendenza alla relazione legame si traduce, dal un punto di vista computazionale, nella necessità di disporre di matrici dei dati con un diverso numero di colonne, riferite appunto alle covariate coinvolte nella stima. A differenza dei metodi "pair-wise" quindi, la divisione dei risultati in liste diverse, è fatta a priori rispetto al processo di imputazione, più che raddoppiando il tempo necessario per l'intera procedura. In questo lavoro di tesi, il problema è stato risolto creando tre cartelle differenti: due contenenti tutti i modelli previsti da TS e SVR, e una contenente 13.795 matrici al cui interno si trova il profilo del gene a cui è riferita la matrice stessa e tutti i 234 miRNA rilevati.

Un'aspetto che verrà analizzato per ciascun modello multvariato sarà la dipendenza tra lo score ottenuto e il numero di covariate utilizzate. Ipotizzando infatti che ciascuna covariata spieghi una piccola parte dell'espressione totale della risposta, ci si aspetta che più il numero di covariate del modello (N_i) è grande, più gli score ottenuti saranno marginalmente piccoli. Se questo fosse vero, nascerebbe un problema legato alla confrontabilità degli N^{tot} indici di una stessa lista, dovuto all'esigenza di restituire un'insieme di interazioni ordinate: tendenzialmente, i modelli con meno covariate pro-

Figura 15: Numero di covariate N_i previste per ciascun modello, in funzione delle due liste di predizioni TS e SVR.



durranno degli score che saranno avvantaggiati nel rank e si posizioneranno nella parte più alta della classifica.

.

4.3.1 Il Coefficiente di Correlazione Parziale (R_p, R_p^{TS}, R_p^{SVR})

Formulazione Le formule utilizzate per il calcolo del coefficiente di Correlazione Parziale ponno essere riassunte nel modo seguente:

$$\widetilde{r}_{g_i\, m_{j_i}} = \frac{-\omega^*_{g_i\, m_{j_i}}}{\sqrt{\omega^*_{g_i g_i} \cdot \omega^*_{m_{j_i} m_{j_i}}}} \qquad \text{con } i \text{ da 1 a } N^{tot}\text{, e } j_i \text{ da 1 a } N_i \qquad (22)$$

dove $\omega^*_{g_i g_i}$ è l'elemento in posizione [1,1] e $\omega^*_{g_i k}$ è l'elemento in posizione $[1, k]$ della i-esima matrice Ω^*_i, definita come l'inversa di S^*_i e calcolata secondo la formulazione (16). Si rimanda al Sottoparagrafo 1.2.1 e all'articolo di riferimento[59] per ulteriori approfondimenti sulle altre misure utilizzate nella stima.

In particolare, abbiamo deciso di calcolare tre diversi indici: $\widetilde{r}$, $\widetilde{r}^{TS}$ e $\widetilde{r}^{SVR}$. Il primo stimatore si svincola dalla selezione di un sottoinsieme iniziale di covariate e rappresenta il coefficiente di Correlazione Parziale di ordine 233 (N_m-1): per ciascuna coppia mRNA-miRNA, $\widetilde{r}$ è calcolato condizionatamente ai valori dei rimanenti 233 miRNA rilevati. I restanti due stimatori, utilizzano la relativa lista - nell'ordine TS e SVR - per decidere per ciascun mRNA quanti sono i miRNA associati (N_i^{TS} e N_i^{SVR}) e di conseguenza l'ordine del coefficiente di Correlazione Parziale. Ad esempio, se al gene g sono associati 10 miRNA con sequenze complementari, allora per ciascuna delle deci possibili coppie mRNA-miRNA il grado della correlazione parziale sarà pari a 9 ($N_i^{lista} - 1$).

Funzione R Per il calcolo delle matrici di Correlazone Parziale abbiamo utilizzato la funzione *pcor.shrink()*, contenuta nella libreria *corpcor*, sviluppata dagli autori dell'articolo di riferimento. Per essere utilizzata la procedura vuole la seguente sintassi:

pcor.shrink*(x, lambda, w, verbose=TRUE)*

Input:

x: matrice dei dati, contenente sulla prima colonna il profilo di espressione del gene e sulle rimanenti quelli dei microRNA associati; le righe rappresentano i campioni.

lambda: parametro di liscimento, se mancante viene stimato attraverso una formula analitica descritta nell'articolo di riferimento

w: peso dei campioni, se mancante pari ad uno sul numero di campioni

---

Output:

matrice delle correlazioni parziali

La funzione, data una matrice contenente i valori relativi a tutte le variabili implicate nel modello, restituisce la matrice stimata S^* contenente sulla

diagonale principale le varianze e sulle restanti celle le Correlazioni Parziali delle relative coppie. Ovviamente, è una matrice simmetrica e quadrata, di dimensione pari al numero di variabili contenute nella matrice dei dati fornita come input.

Procedimento Per ciascuna cartella creata (vedi Paragrafo 4.3), abbiamo imputato la funzione *pcor.shrink()* in maniera sequenziale per ciascun file contenuto. Dalla matrice delle correlazioni parziali restituita abbiamo estratto la prima riga, ed eliminato la prima cella di quest ultima: avendo inserito nella prima colonna di ogni matrice iniziale il profilo del "gene risposta", il vettore così ottenuto rappresenta tutti i valori $\widetilde{r}$ per le possibili coppie mRNA-miRNA in essa contenuta.

Tutti i valori così ottenuti, che avessero un coefficiente stimato strettamente inferiore a zero, sono stati salvati in un unica lista. La procedura descritta produrrà quindi un totale di tre liste (R_p, R_p^{TS} e R_p^{SVR}) ricavate in maniera indipendente l'una dall'altra.

Distorsione rispetto al grado Ciascun coefficiente di Correlazione Parziale ha un grado che dipende dal numero di miRNA che condizionano la stima (N_i). Mentre gli score contenuti nella lista R_p hanno tutti grado 233, e quindi possono essere confrontati tra loro, quelli delle due restanti classifiche, avendo un grado che varia nell'intervallo dei numeri naturali [1,71] per R_p^{TS} e [1,110] per R_p^{SVR}, soffrono della distorsione descritta all'inizio di questo sottocapitolo.

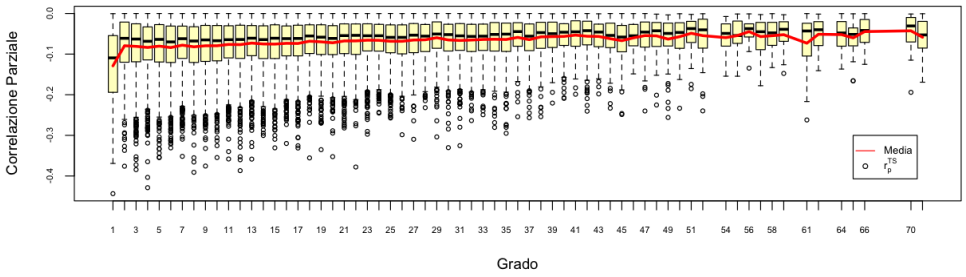
Per valutare l'entità del problema abbiamo rappresentato graficamente le distribuzioni dei coefficienti in funzione del loro grado, attraverso dei boxplot suddivisi sulla base delle liste di predizioni iniziali utilizzate. Il risultato ottenuto è riportato in Figura 16.

Anche se gli $\widetilde{r}$ outliers[14] hanno effettivamente una relazione positiva rispetto al grado, i restanti coeffienti di Correlazione Parziale sembrano avere la medesima distribuzione, con dei quartili paragonabili tra loro. Questo comportamento sembra avvalorare la comune scelta dei biologi di non andare oltre al secondo grado di condizionamento: da un certo punto in poi l'informazione spuria risultante dall'introduzione di ulteriori covariate è approssimabile a zero.

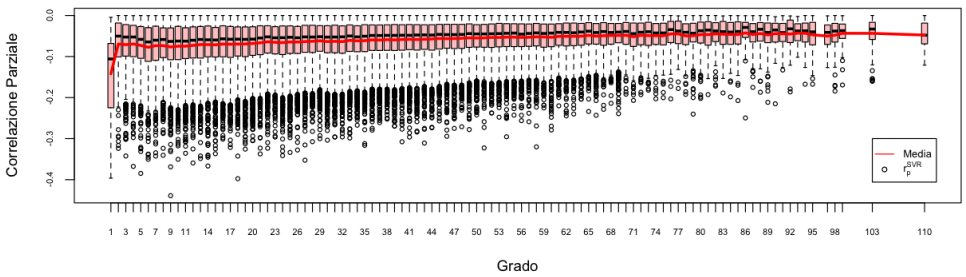
Il vero bias sembra esserci solamente per gli $\widetilde{r}$ condizionati ad un so-

[14]Definiamo outliers quei valori che si distribuiscono su valori maggiori del baffo superiorie del box-plot rappresentato.

Figura 16: Sono rappresentate le distribuzioni dei coefficienti di Correlazione Parziale in funzione del loro grado, suddivisi per liste di predizioni iniziali - TargetScan (a) e microRNA.org (b). I pallini rappresentano i singoli valori dei coefficienti, mentre la linea rossa la media calcolata per ciascun gruppo.



(a) Distribuzione degli $\widetilde{r}$ calcolati per TS



(b) Distribuzione degli $\widetilde{r}$ calcolati per SVR

lo microRNA, dovuto probabilmente all'ottimizzazione del metodo per alte dimensionalità della matrice dei dati richiesta in ingresso. Le oscillazioni della linea media, rapresentata con linea rossa continua, per i valori delle ascisse più estremi sono dovute invece alla minore numerosità campionaria dei gruppi, e quindi non risultano particolarmente allarmanti.

Per i motivi sovraesposti decidiamo di assumere che i coefficienti di Correlazione Parziale stimati contenuti in una stessa lista provengano tutti da un'unica distribuzione, eliminando di fatto l'informazione sul grado del condizionamento.

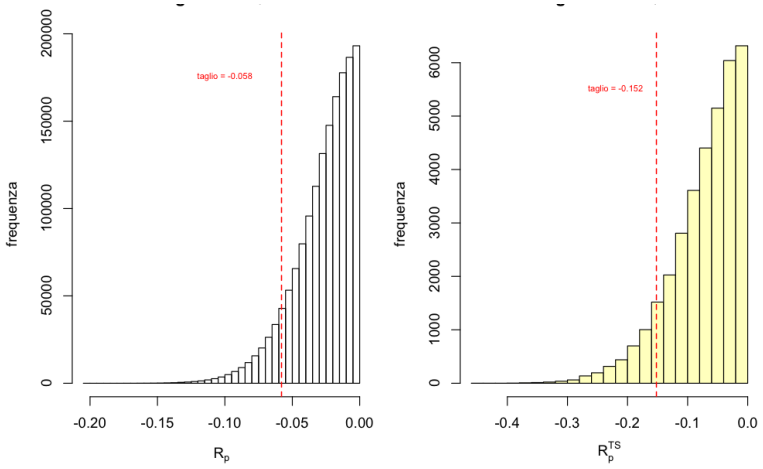
Significatività La significatività associata al coefficiente di Correlazione Parziale ottenuto attraverso la Formula 22, non è calcolabile con i mezzi a nostra disposizione, quindi non è possibile discriminare in maniera sistematica le reali interazioni con forza inversa da quelle dovute al caso.

L'assunzione di un'unica distribuzione per ciascuna lista ottenuta ci permette però di imporre un taglio basato sulla distribuzione stessa: come è uso comune, in mancanza di un p-value, si decide una soglia di discriminazione pari al quantile 0,05 calcolato sulle stime $\widetilde{r}$.

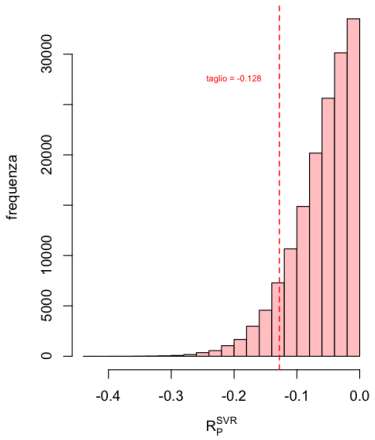
In particolare, per le liste R_p, R_p^{TS} e R_p^{SVR}, i valori discriminanti sono rispettivamente pari a -0,058, -0,152 e -0,128, riducendo il numero delle interazioni restituite dalla procedura del 95%. Il bias (trascurabile) che ci attendevamo è osservabile attraverso le soglie ottenute: dove il grado è mediamente sposato su valori più alti la soglia tende ad abbassarsi.

In Figura 17 sono rappresentate le distribuzioni dei diversi coefficienti di Correlazione Parziale calcolati e le relative soglie. .

Figura 17: Sono rappresentate le distribuzione dei coefficienti di Correlazione Parziale minori di zero. Nel primo grafico (a) sono rappresentate con istogrammi bianchi le correlazioni di grado 233; nei restanti due grafici troviamo le correlazioni ottenute rispettivamente con le liste di predizioni TargetScan (b) e microRNA.org (c). Con linea rossa tratteggiata è riportato il taglio posto sulla significatività, pari al quantile 0,05 della distribuzione stessa.



(a) R_p (b) R_p^{TS}



(c) R_p^{SVR}

Risultati In tabella riportiamo i risultati ottenuti dalle diverse liste.

Lista	$\mathbf{N^T}$	t	$\mathbf{N^S}$	$\mathbf{V^{tot}}$	$\mathbf{V^{500}}$	$\mathbf{V^{1000}}$	$\mathbf{\%^{tot}}$	$\mathbf{\%^{500}}$	$\mathbf{\%^{1000}}$
R_p	1.591.166	-0,058	161.402	1023	2	3	0,63	0,4	0,3
R_p^{TS}	34.807	-0,152	3.520	154	26	51	4,37	5,2	5,1
R_p^{SVR}	153.817	-0,128	15.492	2.324	11	17	1,51	2,2	1,7

.

4.3.2 ArgoLasso (L_O^{TS}, L_O^{SVR}, L_M^{TS}, AL_O^{TS}, AL_O^{SVR}, AL_M^{TS})

Formulazione Seguendo le indicazioni dell'articolo di Lu e colleghi[45] riproponiamo con formulazioni equivalenti i due modelli per la regressione penalizzata L_1, nel seguente modo:

$$\mathbf{g}_i = \beta_{0i} + \sum_{j=1}^{J_i} \beta_{ji} \cdot c_{ij}^A \cdot \mathbf{m}_j + \epsilon_i \qquad \text{soggetta a } \textstyle\sum_{j=1}^{J_i} |\beta_{ji}| < t_i \tag{23}$$

$$\mathbf{g}_i = \beta_{0i} + \sum_{j=1}^{J_i} \beta_{ji,2} \cdot c_{ij}^A \cdot Ago_2 \cdot \mathbf{m}_j + \sum_{j=1}^{J_i} \beta_{ji,134} \cdot c_{ij}^A \cdot Ago_{134} \cdot \mathbf{m}_j + \epsilon_i \tag{24}$$
$$\text{soggetta a } \textstyle\sum_{j=1}^{J_i} |\beta_{ji,2} + \beta_{ji,134}| < t_i$$

dove esiste una corrispondenza uno a uno tra il λ_i in (13) e il t_i in (23 24).

Nella Formula 23 (L) troviamo la classica stima di regressione Lasso, ottenuta attraverso l'algoritmo implementativo fornito dall'articolo di riferimento; nella Formula 24 (AL) viene aggiunta l'informazione sulle proteine Argonaute nel tentativo di migliorare le previsioni ottenute con L.

Come abbiamo visto nel capitolo precedente (Paragrafo 1.1.3.3.), per ottenenere un'aumento della specificità, abbiamo introdotto una variazione nell'implementazione della stima, ottenendo nella totalità due possibili versioni di calcalcolo: *ArgoLasso-MultiRun* - quella proposta dagli autori - che pro-

durrà un totale di due liste[15] (L_M^{TS}, AL_M^{TS}); e *ArgoLasso-OneRun* - la nostra rivisitazione - che fornirà un totale di quattro classifiche ($L_O^{TS}, L_O^{SVR}, AL_O^{TS}$, AL_O^{SVR}). Questo cambiamento non influirà in alcun modo nelle forulazioni (23) e (24) ma semplicemente forzerà l'uscita della funzione alla prima stima ottenuta per ciascun "microRNA risposta".

Funzione R Nel modello ArgoLasso abbiamo utilizzato quattro diverse funzioni scritte con linguaggio R: *lars()* - contenuta nell'omonima libreria standard - per il calcolo vero e proprio della stima lasso; *ArgoLasso()* per la selezione delle associazioni interessanti e per l'esecuzione del ciclo ricorsivo nel caso *MultiRun*; *BootstrapArgoLasso()* per produrre gli "score falsi positivi", che ci aiuteranno nell'individuazione del taglio per la discriminazione delle associazioni significative; *ROCArgoLasso()* per la rappresentazione delle curve ROC attraverso l'utilizzo delle associazioni validate e degli score forniti dalla precedente funzione.

Le ultime tre procedure sono il risultato di un riadattamento del codice sviluppato dagli autori del metodo, liberamente scaricabile all'indirizzo *http://biocompute.bmi.ac.cn/CZ/lab/alarmnet*. Il listato relativo alle tre funzioni è riportato nell'Appendice A.2..

Documentiamo nel seguito le funzioni utilizzate.

lars(*x,y,type=c("lasso","lar","forward.stagewise","stepwise"),trace=FALSE, normalize=TRUE,intercept=TRUE,Gram,max.step,use.Gram=TRUE)*

Input:

x: matrice dei predittori
y: vettore della variabile risposta
type: lasso

[15] Il modello *MultiRun* riferito agli accoppiamenti della lista SVR (L_M^{SVR} e AL_M^{SVR}) non è stato stimato per motivi di complessità computazionale, relativi al calcolo della significatività.

Output:

beta: matrice quadrata di dimensione pari al numero di predittori; ciascuna riga rappresenta uno step di stima e al suo interno troviamo un numero di coefficienti (uno per colonna) diversi da zero pari al numero di passaggi fatti

entry: vettore contenente gli indici di colonna del regressore che entra nel modello in ciascuno step

lambda: vettore contenente il primo valore del parametro di lisciamento in corrispondenza del quale viene aggiunta una nuova variabile ai risultati, per ciascuno step

Cp: vettore contenente la stima degli errori commessi per ciascun modello ottenuto in ciascuno step

ArgoLasso*(data.dir,data.dir.salvataggio,MRun=TRUE, AGO=TRUE,max.run=1000)*

Input:

data.dir: directory in cui sono contenuti i file delle matrici dei dati; ciascun file deve essere nominato come l'mRNA risposta e la prima colonna deve contenere il suo profilo di espressione

data.dir.salvataggio: directory in cui vengono salvati i risultati ottenuti

MRun: TRUE, per la versione *MultiRun*, FALSE per quella *OneRun*

AGO: FALSE per la formulazione (25), TRUE per la formulazione (**??**)

max.run: se MRun settato a TRUE, numero massimo di cicli per ciascuna matrice

Output:

a schermo: vengono stampati i progressi del processo, il tempo impiegato, un breve riassunto sulle directory utilizzate e il modello, e il numero di associazioni finali selezionate

selectXY.Rdata: file contenente il dataframe caricabile su piattaforma R, denominata nell'ambiente interno *select.dat*; al suo interno si trovano nell'ordine: il codice relativo all'"mRNA risposta" estratto dal nome del file della matrice dei dati, il nome del miRNA appaiato, il coefficiente stimato e lo score calcolato, per le sole associazioni con coefficiente minore di zero

Dipendenze: funzione *lars()*

BootstrapArgoLasso*(data.dir,data.dir.salvataggio,MultiRun=TRUE,AGO=TRUE, repeat.times=5,max.run=1000)*

Input:

data.dir, data.dir.salvataggio, MultiRun, AGO, max.run:
vedi funzione *ArgoLasso()*

repeat.times: numero di volte in cui il procedimento Bootstrap deve essere ripetuto; per ciascuna iterazione e per ciascuna matrice contenuta nella directory dei dati, viene generato un "miRNA causale"

Output:

a schermo: vengono stampati i progressi del processo, il tempo impiegato per ciascuna operazione bootstrap e un breve riassunto sulle directory e il modello utilizzato

randomNXY.Rdata: file contenente una lista caricabile su piattaforma R, denominata nell'ambiente interno *random.score*; la lista è composta da un numero di elementi pari al numero di operazioni bootstrap eseguite, e al suo interno un numero variabile di score, relativi ai "miRNA casuali" che sono risultati significativi a seguito della selezione

---

Dipendenze: funzione *lars()* e funzione *ArgoLasso()*

ROCArgoLasso*(random, overlap, dir.salvataggio, nome.plot, plot.pdf=list(crea=TRUE,nome=""), plot.add=list(unico=FALSE,col="black")*

Input:

random: lista degli score ottenuti con la funzione *BootstrapArgoLasso* (variabile *random.score*), considerati come falsi positivi

overlap: dataframe delle associazioni validate, considerate come veri positivi; è possibile creare il file attraverso un'opzione della funzione *ValidaLista()*

dir.salvataggio: directory in cui sarà salvato il grafico ottenuto

nome.plot, plot.pdf, plot.add: opzioni grafiche relative alla curva ROC prodotta

---

Output:

a schermo: viene stampato lo stato del processo e l'AUC calcolato

Proteina	Simbolo	Id Entrez	id Ensembl
Ago_1	EIF2C1	26523	ENSG00000092847
Ago_2	EIF2C2	27161	ENSG00000123908
Ago_3	EIF2C3	192669	ENSG00000126070
Ago_4	EIF2C4	192670	ENSG00000134698

Tabella 3: Nomi e identificativi delle diverse proteine Argonaute presenti nell'uomo

quartz: viene prodotto il plot relativo alla curva ROC prodotta
ROCNXY: viene salvato un file .pdf contenente l'immagine della curva ROC ottenuta nella directory indicata tra le opzioni

---

Dipendenze: funzione *prediction* e funzione *performance()*, contenute nella libreria *ROCR*

Scelta del valore AGO Per poter stimare il modello (24) è necessario definire i termini Ago_2 e Ago_{134}, che sintetizzano la capacità del RISC di legarsi alla regione 3'UTR. Nell'articolo di riferimento, viene assunto che il livello delle proteine Argonaute sia positivamente relazionato al livello di espressione del relativo mRNA che codifica per esse (Tabella 3), ma non vi è alcuna indicazione su come inglobare in un unico valore gli Ago competitori (AGO_1, AGO_3, AGO_4).

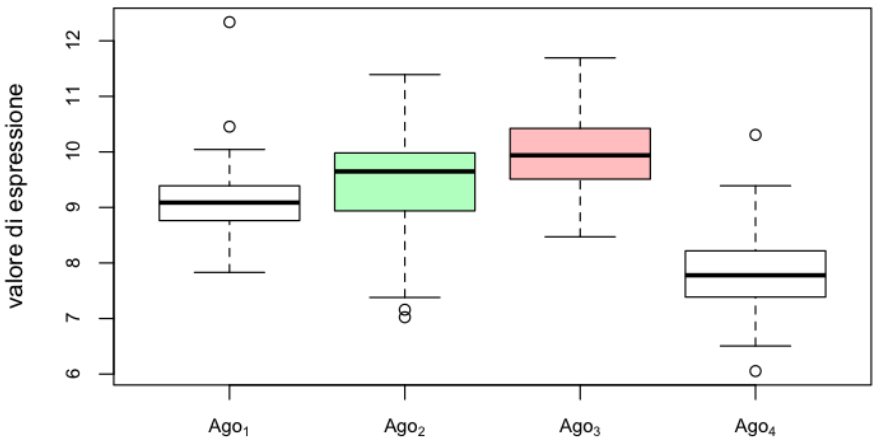
Tra i metodi statistici di sintesi a disposizione il più indicato è sicuramente quello delle componenti principali (PCA): lo scopo di questa tecnica infatti è la riduzione del numero delle variabili attraverso una trasformazione lineare di queste in un nuovo sistema cartesiano, che darà vita a delle nuove variabili latenti[64]. Ciascuna componente così creata, essendo di fatto una sintesi, spiegherà una certa percentuale della varianza del modello e quindi dell'informazione complessiva dei dati originali: attraverso i pesi assegnati alle variabili originali dalla componente principale è possibile assegnare anche un significato a posteriori a ciascuna di queste.

Per utilizzare la PCA nel nostro dataset abbiamo nell'ordine: creato una matrice composta dai tre profili di espressione degli mRNA codificani per le proteine AGO_1, AGO_3, AGO_4; calcolato la matrice di correlazione, attraverso la funzione R *cor()*; estratto i relativi autovalori e autovettori, attraverso le funzioni R *eigen()*; infine, ottenuto la varianza spiegata dividento gli autovalori per il numero di componenti totali.

I risultati ottenuti (Tabella 4) sugeriscono che la prima componente è in grado di spiegare da sola più del 70% dell'informazione contenuta nei dati originali ma, osservando l'autovettore stimato, è evidente come ciascuna Agoproteina apporti un'informazione equiparabile. Questa osservazione, e il fatto che la combinazione lineare dei profili degli mRNA associati ai diversi AGO produrrebbe una stima negativa[16], ci ha spinti ad adottare una soluzione alternativa più semplice per la scelta della nuova variabile: poichè il termine Ago_{134} rappresenta un competitore di Ago_2, decidiamo di scegliere il profilo dell'mRNA che si distribuisce su valori più grandi rispetto agli

[16] I valori di espressione sono sempre maggiori di zero

Figura 18: Distribuzione dei valori di espressione dei geni che codificano per le quattro proteine Argonaute.



	autovalore	**varianza spiegata**	**autovettore** (Ago_1, Ago_3, Ago_4)
λ_1^a	2,232	0,744	$[\ -0,579 \quad -0.581 \quad -0,572\ ]$
λ_2^a	0,401	0,134	$[\ -0,489 \quad -0.314 \quad 0,814\ ]$
λ_3^a	0,367	0,122	$[\ 0,652 \quad -0.751 \quad 0,103\ ]$

Tabella 4: Nomi e identificativi delle diverse proteine Argonaute presenti nell'uomo

altri, nel nostro caso AGO_3 (Figura 18).

Procedura Per ottenere i coefficienti stimati attraverso la Formula 25 abbiamo richiamato la funzione *ArgoLasso()* settando l'opzione AGO a FALSE e scegliendo come directory dei dati le cartelle contenenti le matrici descritte per i metodi multivariati. Ripetendo l'operazione per entrambe le versioni di calcolo possibili otteniamo le tre liste L_M^{TS}, L_O^{TS} e L_O^{SVR}.

L'implementazione della Formula 24 ha rchiesto la creazione di matrici diverse da quelle fino ad ora utilizzate. L'aggiunta dell'informazione sul RISC comporta la duplicazione delle N_i colonne relative ai microRNA associati all'i-esimo mRNA: le prime N_i devono essere moltiplicate per il profilo del gene Ago_2 mentre le restanti per i valori di espressione di Ago_3.

Le matrici dei dati così ottenute, distinte per sottoinsieme di interazioni iniziali utilizzate, conterranno un totale di $1 + (N_i \times 2)$ colonne dove, quelle relative ai valori di espressione dei microRNA, saranno contrassegnate - oltre che dal nome - anche dalla locuzione Ago2/Ago3: questo servirà alla procedura *ArgoLasso()* per distinguere i coefficienti stimati e restituire solo quelli effettivamente indicatori della forza dell'associazione, secondo le ipotesi volute dal metodo.

Quindi, settando questa volta l'opzione AGO a TRUE e fornendo in ingresso le nuove matrici costruite, la funzione *ArgoLasso()* restituitirà altre tre liste di interazioni: AL_M^{TS}, AL_O^{TS} e AL_O^{SVR}. In questo caso è la funzione

stessa ad occuparsi della selezione dei coefficienti minori di zero e/o della discriminazione di quelli associati ad Ago2.

Score e distorsione rispetto al numero di covariate I coefficienti β_{ji} o $\beta_{ji,2}$, ottenuti rispettivamente dai modelli Lasso o ArgoLasso, per definizione rappresentano la forza diretta stimata che lega le coppie mRNA-miRNA e per questo motivo sembrano essere gli indici più naturali da utilizzare nell'ordinamento delle liste restituite. A questo scopo gli autori del metodo propongono però di utilizzare un nuovo score, senza indicare alcuna motivazione circa tale scelta.

Confrontando i due indici (Figura 19) notiamo che pur essendo inversamente relazionati - coefficienti prossimi a -1 implicano alti valori dello score, e coefficienti prossimi a 0 implicano valori bassi dello score - questo legame non può essere definito lineare. Il sospetto è che l'introduzione di un nuovo criterio di ordinamento serva a risolvere la distorsione legata al numero di miRNA utilizzati nel set iniziale di candidati: il nuovo score infatti utilizza, al posto del valore del coefficiente, il suo "rank relativo" all'interno del sottoinsieme riferito ad uno stesso mRNA che contiene quei microRNA risultati significativi. L'indice così definito, moltiplicato per cento, assume per qualsiasi appaiamento un numero naturale compreso tra 1 e 100, rendendo così gli elementi di una stessa lista confrontabili tra loro.

Nonostante questa soluzione riduca effettivamente il bias (Figura 20), soprattutto per numerosità medio/alte di covariate che entrano nel modello iniziale di stima, quando i regressori sono inferiori a dieci la distorsione rimane, influenzando in maniera drastica il top della classifica finale e di conseguenza le performance del metodo.

Le considerazioni fatte valgono per entrambe le versioni di calcolo e per entrambe le liste di predizioni iniziali, salvo accentuarsi nel caso di utilizzo

Figura 19: Relazione tra i coefficienti restituiti dalla funzione *ArgoLasso* e i relativi score - versione *MultiRun* - per quattro differenti datasets; le lettere colorate rappresentano i risultati ottenuti nell'articolo di riferimento (b: broad, m: medison, k: mskcc) mentre i puntini neri rappresentano quelli ottenuti con i dati dell'EOC.

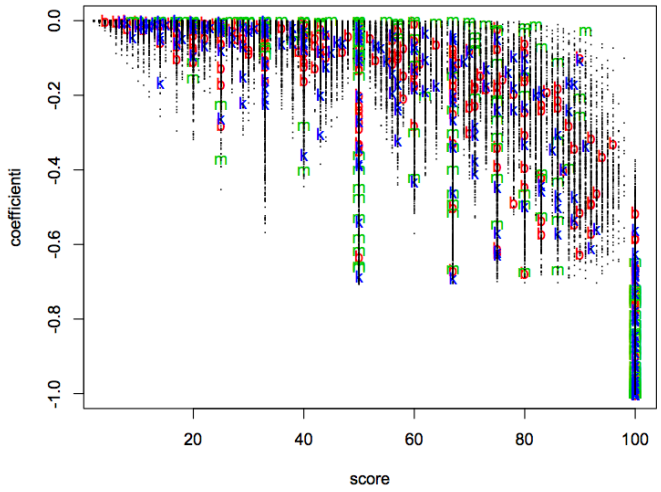
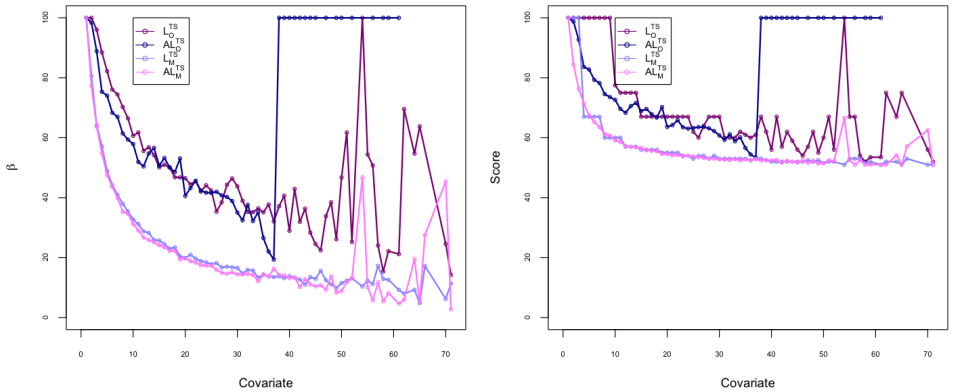


Figura 20: Sono rappresentati i valori medi dei coefficienti stimati (a) e degli score calcolati (b) in funzione del numero di microRNA utilizzati come set iniziale del modello. I coefficienti sono stati moltiplicati per -100 per confrontare i valori sulla medesima scala.



(a) Coefficienti (b) Score

della procedura OneRun a causa di una minore numerosità interna ai gruppi.

Significatività In generale, per i modelli che utilizzano una regressione penalizzata, non esiste una distribuzione nulla esatta e il procedimento comunemente utilizzato è il ricampionamento Bootstrap.

Per il nostro dataset utilizziamo la funzione *BootstrapArgoLasso()* che, date le directory contenenti le matrici dei dati, ripete l'algoritmo di stima descritto precedentemente, ma inserendo ad ogni interazione per ogni mRNA un profilo casuale, simulato attraverso ricampionamento di uno dei miRNA associati estratto a sorte: il procedimento restituirà un insieme di liste, di numerosità pari al numero di cicli fatti, contenenti gli score legati alle variabili casuali che sono risultate significative in ciascuna iterazione - ossia strettamente minori di zero e/o riferite alla proteina AGO_2. Ovviamente il numero di elementi restituiti per ciascun ciclo sarà variabile.

Come abbiamo già detto, parlando del coefficiente di Correlazione di Gini (Paragrafo 4.2.2.), la bontà delle stime ottenute attraverso il metodo Bootstrap dipende principlmente dal numero B di ripetizioni fatte: l'articolo suggerisce di applicare il controllo randomizzato 100 volte.

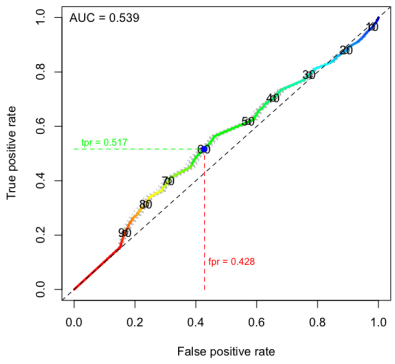
A causa dell'elevato costo computazionale[17] abbiamo deciso di eseguire la funzione solamente 20 volte: si può dimostrare infatti che essendo il numero di mRNA elevato rispetto all'esperimento degli autori, dopo 5 ripetizioni la curva ROC, ottenuta dalle medie delle singole interpolazioni, rimane pressochè inalterata[18].

A differenza del coefficiente di Correlazione di Gini, in questo modello non utilizziamo questa procedura per stimare la vera distribuzione nulla, ma piuttosto per quantificare la probabilità che un'appaiamento casuale simulato, risulti erroneamente significativo.

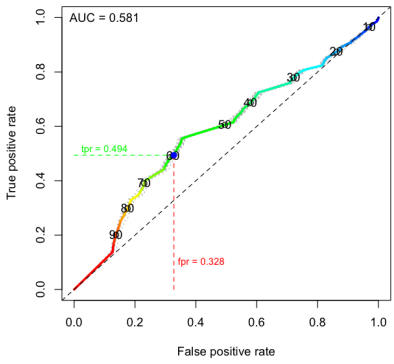
[17] Tempo medio stimato per 100 iterazioni del controllo randomizzato per modello e lista di predizione: 3giorni e 10 ore circa.

[18] verificato con una prova completa di 100 iterazioni bootstrap per il modello ArgoLasso-MultiRun con lista di predizioni iniziali TargetScan.

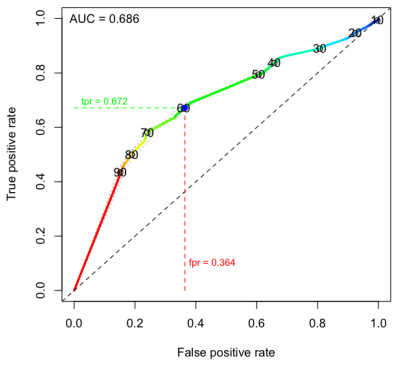
Figura 21: Sono rappresentate le curve ROC ottenute attraverso la funzione *ROCArgo-Lasso()*. Nella prima colonna (a,c,e) le curve si riferiscono ai modelli calcolati attraverso la Formula (25) (L_M^{TS}, L_O^{TS}, L_O^{SVR}), mentre nella seconda (b,d,f) si riferiscono ai modelli calcolati attaraverso la Formula (**??**) (AL_M^{TS}, AL_O^{TS}, AL_O^{SVR}). Il taglio scelto è in corrispondenza dello score 60.



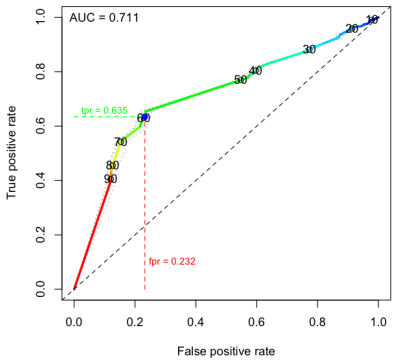
(a) L_M^{TS}



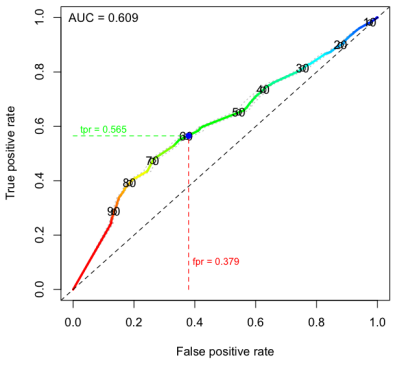
(b) AL_M^{TS}



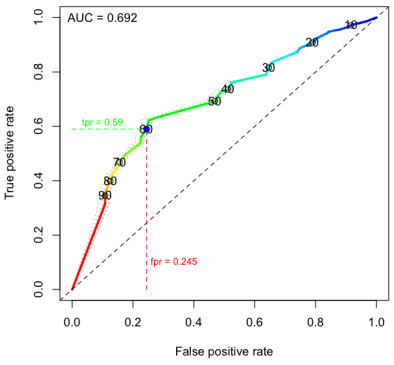
(c) L_O^{TS}



(d) AL_O^{TS}



(e) L_O^{SVR}



(f) AL_O^{SVR}

ROC & AUC Per valutare le performance dei modelli nella sua interezza, rappresentiamo quindi - attraverso la procedura *RocArgoLasso()* - le curve ROC risultanti dalla combinazione degli score riferiti a quelle coppie mRNA-miRNA contenute nella lista e che risultano sperimentalmente validate - ottenute attraverso l'utilizzo di *ValidaLista()* - e gli score restituiti dalla procedura Boostrap appena descritta. La curva ROC è ottenuta calcolando i valori di specificità[19] e sensibilità[20] ricavati dalla classifica dei valori ordinati: in corrispondenza di ogni possibile taglio viene costruita una tabella di contingenza 2x2 - dove sono rappresentate le quattro possibilità (veri positivi, veri negativi, falsi positivi falsi negativi) - da cui sono calcolate le due misure. Per ottenere la migliore soglia di discriminazione per la significatività dei nostri score, analizziamo quindi le curve ROC mostrate in Figura 21.

In questo contesto, essendo la proporzione di validati poco rappresentata rispetto al totale di osservazioni, quello che vogliamo fare è trovare quel valore che ci permetta di controllare il FPR (false positive rate) e a cui corrisponda il miglior numero di TPR (true positive rate) possiibile: il compromesso ottimale è rappresentato dal punto della curva più vicino all'angolo superiore sinistro, che per i dati dell'EOC è pari a 60.

Con il taglio scelto procediamo quindi ad eliminare, dalle sei liste a nostra disposizione, tutte quelle assocazioni a cui corrispondono valori dello score inferiori a 60.

Un'altro utile indicatore di confrontro tra i modelli Lasso e ArgoLasso è il cosidetto AUC: questo indice quantifica la capacità discriminatoria di un modello, ossia la sua attitudine a separare propriamente la popolazione in interazioni validate e non, attraverso una misura proporzionale all'esten-

[19] Specificità = veri negativi / (veri negativi + falsi positivi)

[20] Sensibilità = veri positivi / (veri positivi + falsi negativi)

sione dell'area sottesa alla curva ROC (*Area Under Curve*). Nel caso di un modello perfetto, ossia che non restituisce alcun falso positivo né falso negativo, la AUC corrisponde all'area dell'interno quadrato e assume valore 1; al contrario, la curva ROC per un modello assolutamente privo di valore informativo è rappresentata dalla diagonale passante per l'origine, con un AUC pari a 0,5.

Quindi, in questo paragrafo, oltre agli indici proposti in precedenza, nella sezione dedicata ai risultati riporteremo anche il valore dell'AUC stimato.

Risultati Per i motivi sovraesposti la misura adottata per l'ordinamento è lo score, creato sulla base dei rank dei coefficienti raggruppati per mRNA. Essendo un valore discreto, con un range limitato rispetto al numero di associazione restituite, quella che otterremo sarà una "classifica a scalini" (Figura 22) dovuta al pari-merito di molte coppie miRNA-mRNA: per risolvere il problema abbiamo deciso di eseguire un'ordinamento su due livelli, prima rispetto allo score e poi al coefficiente beta stimato.

Quello che però viene evidenziato dai dati è che, per tutte le liste, oltre mille interazioni ottengono congiuntamente un punteggio pari a cento e un coefficiente pari a -1 (n_{-1}^{100}): questo rende inutile l'utilizzo della maggior parte degli indicatori di performance e, da un punto di vista pratico, l'utilizzo stesso del metodo da parte dei biologici per identificare un ristretto sottogruppo di associazioni interessanti.

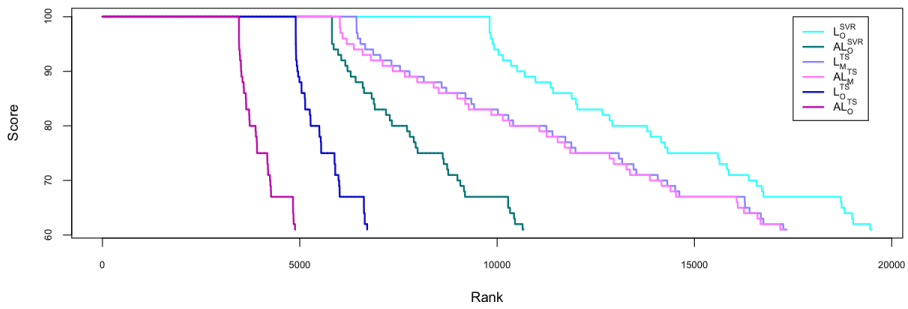
Per concludere, nel seguito riportiamo i risultati ottenuti con una breve discussione sulle implicazioni delle scelte fatte per l'implementazione di ArgoLasso.

Lista	N^T	$t^{\%}$	N^S	V^{tot}	$\%^{tot}$	AUC	$n_{c=-1}^{s=100}$	$\%_{c=-1}^{s=100}$
L_M^{TS}	36.303	52,75	17.334	693	4,00	0,539	1.372	7,91
L_O^{TS}	10.455	35,83	6.709	290	4,32	0,686	2.908	43,35
AL_M^{TS}	36.534	52,69	17.283	652	3,77	0,581	1.334	7,72
AL_O^{TS}	7.908	38.19	4.888	184	3,76	0,711	1.718	35,15
L_O^{SVR}	37.161	47,53	19.497	360	1,85	0,609	3.692	18,94
AL_O^{SVR}	19.597	45,55	10.671	147	1,38	0,692	2.490	23,33

Come vediamo, l'introduzione di una versione *OneRun* dell'algortimo di calcolo, oltre all'ovvio abbassamento del numero di interazioni restituite, si ripercuote soprattutto sulla capacità discriminatoria del modello a cui si riferisce: l'AUC da un valore prossimo alla scelta casuale di classificazione, aumenta notevolmente fino ad arrivare a circa 0.70, per entrambe le formulazioni. Nonostante questo miglioramento, è bene sottolineare che i valori rimangono ben al di sotto di quelli ottenuti nell'esperimento degli autori, pari rispettivamente - per ciascun databases - a (0.72, 0.70 e 0.78) per la Formula 25, e a (0.75, 0.8 e 0.86) per la Formula 24.

Se da un lato l'aggiustamento migliora la discriminazione tra validati e non validati, dall'altro utilizzando una sola volta la funzione *lars()* per ciascun mRNA e forzando la procedura quando la numerosità delle covariate è superiore al numero di campioni, si ottengono più associazioni che ottengono congiuntamente score uguale a 100 e coefficiente pari a -1. Per le liste L_O^{TS} e AL_O^{TS} il problema è drammaticamente evidenziato: rispettivamente al 43%

Figura 22: Sono rappresentati i valori degli score ordinati, per ciascuna delle 6 liste.



e al 35% delle associazioni trovate significative viene assegnata la prima posizione pari-merito. Appare quindi chiaro perchè nel seguito questi risultati non potranno essere in alcun modo utilizzati per il confronto diretto con gli altri metodi proposti, e per quale motivo sia stata utile l'introduzione dell'algortimo di stima Penalized.

4.3.3 Penalized (PZ_{AL}^{TS}, PZ_{TL}^{TS}, PZ_{AL}^{SVR}, PZ_{TL}^{SVR})

Risultati L'inserimento del metodo Penalized all'interno di questo lavoro di tesi è nato dallla necessità di un confronto tra le proposte TaLasso e ArgoLasso, viste le difficoltà per quest'ultimo dovute all'ordinamento dello score. Le formulazioni riprendono quella del modello Lasso classico nel seguente modo:

$$\mathbf{g}_i = \beta_{0i} + \sum_{j=1}^{J_i} \beta_{ji} \cdot c_{ij}^A \cdot \mathbf{m}_j + \epsilon_i \qquad \text{soggetta a } \textstyle\sum_{j=1}^{J_i} |\beta_{ji}| < t_i \tag{25}$$

$$\mathbf{g}_i = \beta_{0i} + \sum_{j=1}^{J_i} \beta_{ji} \cdot c_{ij}^A \cdot \mathbf{m}_j + \epsilon_i \qquad \text{soggetta a } \textstyle\sum_{j=1}^{J_i} |\beta_{ji}| < t_i \text{ , ed a } \beta_{ji} \leq 0 \tag{26}$$

La prima è la stessa utilizzata in ArgoLasso, ma senza l'inserimento dell'informazione sul RISC, mentre la seconda emula TaLasso, imponendo un ulteriore vincolo ai coefficienti da stimare. In entrambi i casi useremo un'implementazione completamente differente da quella originale, che si ripercuterà anche sulla scelta del parametro di lisciamento, che per il pacchetto *penalized* prevede una stima di λ_i per ogni matrice dei dati, attraverso convalida incrociata.

Funzione R La libreria utilizzata per l'implementazione è *penalized*, a cui interno troviamo l'omonima funzione, per la stima vera e propria del modello, e una procedura che fornisce il valore del λ_i ottimale da assegnare a ciascuna matrice. La sintassi di ciascuna funzione è riassunta nel seguito.

penalized*(response, penalized, unpenalized, lambda1=0, lambda2=0, positive=FALSE, data, fusedl=FALSE, model=c("cox","logistic","linear","poisson"), starbeta, stargamma, steps=1, epsilon=1e-10, maxiter, standardize=FALSE, trace=TRUE))*

Input:

response: vettore del profilo della variabile risposta

penalized: matrice, in cui sono contenuti i profili delle covariate (microRNA) sottoposte alla penalizzazione

unpenalized: covariate che non sono penalizzate, di default l'intercetta è compresa

lambda1: valore del parametro di lisciamento

positive: se uguale a TRUE viene aggiunto un vincolo ai beta che sono costretti ad essere non negativi

model: "linear"

Output:

penalized: un vettore contenente i coefficienti penalizzati stimati

optL1*(response, penalized, unpenalized, minlambda1, maxlambda2, base1, lambda2=0, fusedl=FALSE, positive=FALSE, data, model=c("cox","logistic","linear","poisson"), starbeta, startgamma, fold, epsilon=1e-10, maxiter=Inf, standardize=FALSE, trace=TRUE)*

Input:

response, penalized, unpenalized, positive, model: vedi funzione *penalized()*

Output:

lambda: lambda stimato attraverso la convalida incrociata

Per completezza, sottolineiamo che questo pacchetto, oltre alla penalizzazione attraverso valore assoluto (L_1), permette di utilizzare anche quella quadratica (L_2) e una combinazione delle due.

Procedura Per ciascun file, contenuto nelle cartelle relative alle previsioni fornite dagli algoritmi TargetScan e microRNA.org, abbiamo richiamato la funzione *optL1()* per la ricerca del parametro di lisciamento ottimale, utilizzato poi per la stima dei coefficienti attraverso la funzione *penalized()*. Poichè quest'ultima procedura permette di forzare i parametri β_{ji} ad essere esclusivamente non negativi, per ottenere il modello previsto nella Formula 26, dobbiamo cambiare di segno tutti i profili di espressione delle covariate, prima di fornirle in ingresso.

Quindi eseguiamo due volte la stima, per ciascuna lista di predizioni iniziali, settando rispettivamente a FALSE e a TRUE l'opzione *positive*: nel primo caso, prima del salvataggio in PZ_{AL}^{TS} e PZ_{AL}^{SVR}, abbiamo eliminato quelle associazioni maggiori o uguali a zero; mentre nel secondo abbiamo riportato le stime ottenute al segno originale, per poi salvarle nelle liste PZ_{TL}^{TS} e PZ_{TL}^{SVR}.

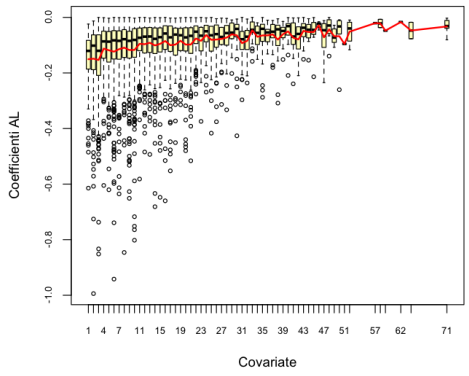
Distorsione rispetto al numero di covariate Come per gli altri modelli multivariati, valutiamo la distorsione presente nei coefficienti stimati, rappresentando questi ultimi in funzione del numero di covariate utilizzate nel modello a cui si riferiscono.

Osservando i grafici in Figura 23, vediamo che la situazione è conforme a quanto mostrato per gli altri algoritmi, quindi ci limitiamo a confermare che esiste una effettiva relazione tra microRNA e le stime ottenute, maggiormente presente quando il numero dei primi è molto basso.

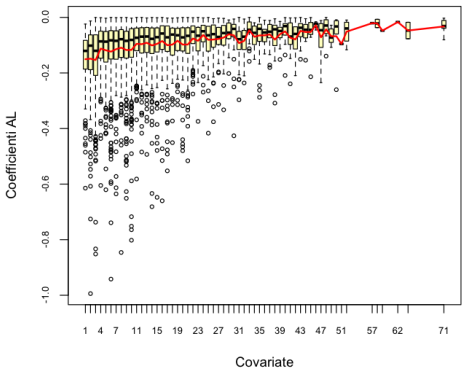
Significatività Rientrando nella classe degli stimatori per i modelli lineari penalizzati, anche Penalized non fornisce nessuna indicazione sulla significatività delle stime.

In questo caso però, essendo il numero di coefficienti restituiti molto basso, decidiamo di non applicare il "taglio quantile" ma di mantenere l'intera

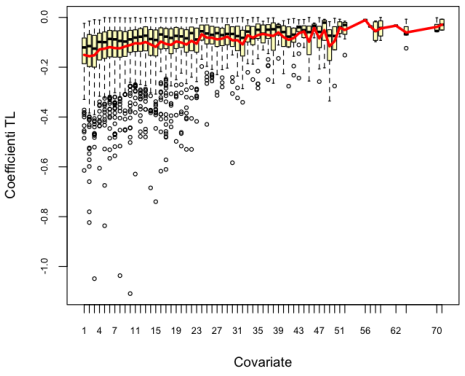
Figura 23: Sono rappresentate le distribuzioni dei coefficienti stimati attraverso la procedura *penalized* in funzione del loro grado, suddivisi per liste di predizioni iniziali - TargetScan (a,b) e microRNA.org (c,d). La prima colonna utilizza la Formula 25 che emula l'algoritmo ArgoLasso, mentre la seconda colonna utilizza la Formula 26, emulazione del modello Talasso. La linea rossa continua rappresenta la media calcolata per ciascun gruppo.



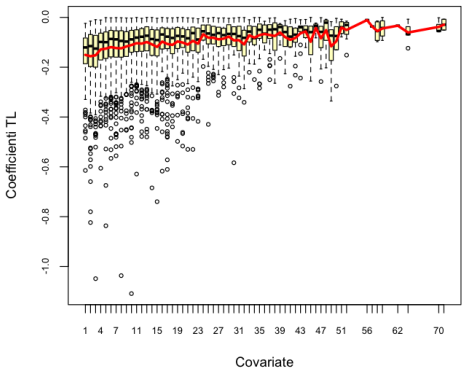
(a) PZ_{AL}^{TS}



(b) PZ_{TL}^{TS}



(c) PZ_{AL}^{SVR}



(d) PZ_{TL}^{SVR}

lista restituita: in questo modo potremo calcolare anche gli indici relativi ai Top^{500} e Top^{1000} della classifica ordinata.

Risultati Riportiamo nella seguente tabella i risultati ottenuti attraverso l'utilizzo della librera Penalized. Ulteriori commenti sulla base di queste informazioni commenti vengono rimandati al seguente capitolo, dove verrà dedicato un intero paragrafo al confrontro tra ArgoLasso e TaLasso.

Lista	N^T	t	N^S	V^{tot}	V^{500}	V^{1000}	$\%^{tot}$	$\%^{500}$	$\%^{1000}$
PZ_{AL}^{TS}	5.372	-	5.372	228	33	58	4,24	6,6	5,8
PZ_{TL}^{TS}	5.924	-	5.924	243	30	59	4,10	6,0	5,9
PZ_{AL}^{SVR}	14.114	-	14.114	241	13	27	1,71	2,6	2,7
PZ_{TL}^{SVR}	17.568	-	17.568	283	13	24	1,61	2,6	2,4

.

4.4 WEB TOOL

Questo capitolo è dedicato ai modelli per cui si è reso necessario l'utilizzo di uno strumento web per l'ottenimento delle stime realtive agli appaiamenti miRNA-mRNA target.

In particolare descriveremo la formattazione necessaria per il corretto caricamento dei dati, il download dei risultati ed eventuali elaborazioni di questi. Il web tool utilizzati sono Magia[2], per il calcolo della Mutua Informazione, e Talasso, per la stima dell'omonimo modello.

4.4.1 La Mutua Informazione (MI^{TS}, MI^{SVR})

Matrice dei dati Le matrici necessarie per il calcolo della Mutua informazione sono due: i profili di espressione di microRNA e di mRNA ottenuti attraverso un'esperimento microarray.

I dati dovranno essere salvati su un file di testo, delimitati attraverso tabulazione, dove la prima riga deve contenere il nome dei campioni e la

prima colonna gli identificativi dei miRNA o dei geni. E' fondamentalmente che gli esperimenti siano appaiati, ossia le colonne di entrambe le matrici si riferiscano nell'ordine ai medesimi pazienti, e che non risultino valori di espressione mancanti. Per i codici identificativi dei geni è possibile utilizzare sia gli Ensembl che gli Entrez Id.

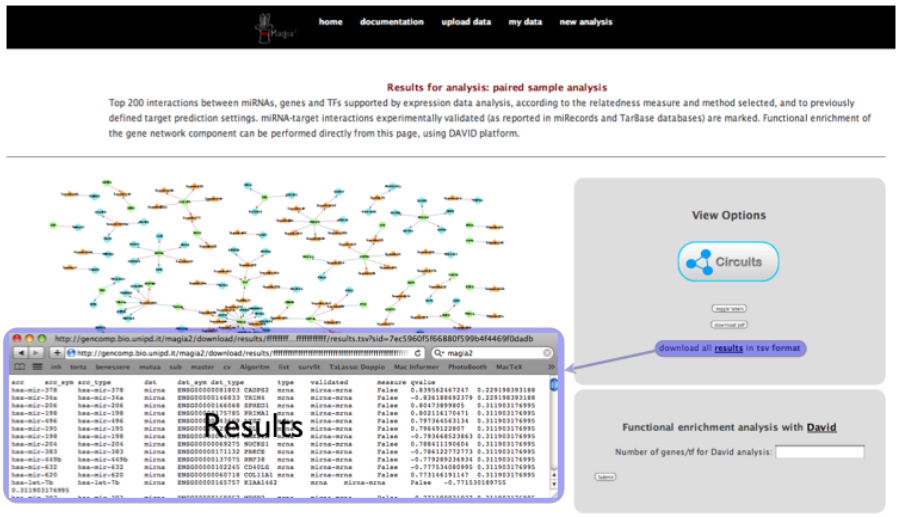
Upload Il primo passo consiste nella creazione del proprio ambiente di lavoro, che consentirà di rivedere i risultati anche in un momento successivo alla sessione di caricamento e download. Basterà portarsi nella sezione "Data Upload" e caricare le due matrici selezionando le opzioni desiderate: Magia^2 provvederà automaticamente a controllare che non siano presenti errori riguardanti gli identificativi utilizzati o la struttura della matrice.

I dati caricati saranno immediatamente visibili nella sezione "new analys": selezionando la coppia di matrici, sarà possibile scegliere tra i quattro indici disponibili - tra cui la Mutua Informazione - e successivamente il predittore iniziale da utilizzare. Per ciascun predittore è inoltre possibile inserire un taglio sulla lista restituita da quest'ultimo.

Download I risultati sono esplorabili in "upload Data" dove è schematizzato un riepilogo di tutte le operazioni svolte: quando visualizzeremo "done" sullo stato dell'analisi di interesse allora potremo cliccare sul nome precedentemente scelto e i risultati saranno consultabili.

Selezionando "download all results in tsv format" (Figura 24) si aprirà un link ad un file, contenente tutte le informazioni sulle associazioni restituite: ciascuna riga rappresenterà un appaiamento mRNA-miRNA già valutato come significativo per il metodo scelto, e contenuto nella lista di predizioni iniziali selezionato tra le opzioni. Ci basterà dunque salvare le informazioni contenute ed elaborale per le successive analisi.

Figura 24: Screenshot della pagina di Magia[2] relativa al download dei risultati



Risultati Utilizzando le stesse matrici e le stesse opzioni, tranne che per la scelta del predittore inziale, abbiamo così ottenuto e scaricato le due liste relative alle stime della Mutua Informazione per l'algoritmo TargetScan (MI^{TS}) e per SVR (MI^{SVR}).

I risultati sono riepilogati nella tabella sottostante.

Lista	N^T	t	N^S	V^{tot}	V^{500}	V^{1000}	$\%^{tot}$	$\%^{500}$	$\%^{1000}$
MI^{TS}	na	na	5.628	240	23	39	4,26	4,6	3,9
MI^{SVR}	na	na	4.715	62	7	18	1,31	1,4	1,8

.

4.4.2 TaLasso ($TL^{TS}_{1/10}$, $TL^{SVR}_{1/10}$)

Matrice dei dati I dati caricati in Talasso devono essere organizzati in due matrici di espressione, salvate su file di testo, con i nomi dei campioni inseriti nella prima riga e gli identificativi delle molecole di RNA sulla prima colonna[21]. Le informazioni contenute devono essere delimitate attraverso

[21]NB: La prima riga dev'essere di lunghezza pari esattamente al numero di campioni; dalla seconda in poi la lunghezza sarà maggiorata di una unità contenente in testa il codice relativo al gene o al miRNA a cui si riferisce.

tabulazione.

A differenza di Magia i campioni contenuti nelle due matrici non devono necessariamente essere riportati nello stesso ordine - anche se i pazienti devono essere gli stessi - e per i geni è possibile utilizzare esclusivamente gli Ensemble Id.

Upload L'interfaccia grafica è molto semplice. Il primo passo consiste nel caricamento delle matrici, scegliendole attraverso il loro percorso assoluto all'interno della postazione di lavoro; poi si scelgono le diverse impostazioni disponibili: il tipo di dati utilizzati, uno o più databases di interazioni iniziali (in caso di scelta multipla l'intersezione o l'unione di questi), il tipo di parametro e uno tra i valori proposti per κ (veid Paragrafo 1.1.3.2).

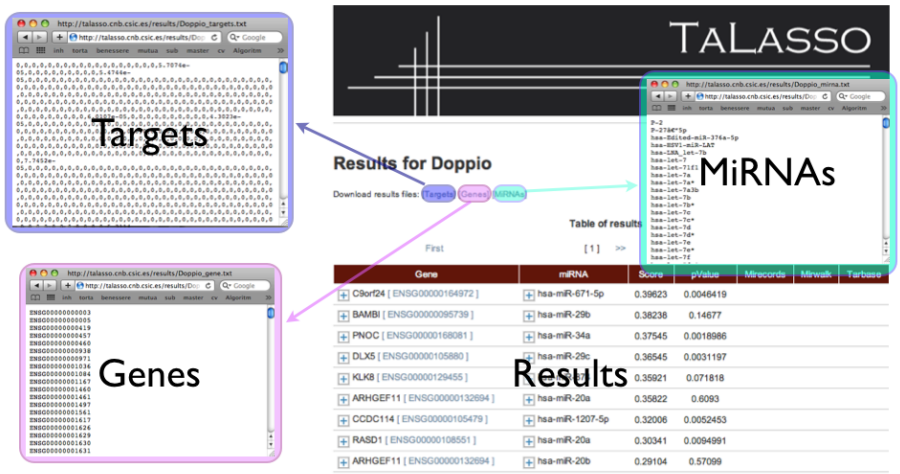
Opzionalmente è possibile assegnare un nome all'analisi, avendo così la possibilità di esplorarla in un secondo momento, digitando nella barra degli indirizzi del proprio browser l'URL relativa al web tool, seguita dalla locuzione */NomeAnalisi*.

Nonostante TaLasso offra la possibilità di scegliere il parametro di lisciamento locale, quando si utilizzano matrici di espressione di elevata dimensionalità, il web tool non restituisce nessun risultato perchè non è in grado di eseguire l'intero algoritmo a causa dell'eccessivo carico computazionale.

Per questo motivo, nel seguito di questo lavoro, considereremo esclusivamente i risultati ottenuti attraverso l'utilizzo del parametro κ globale.

Download Una volta inviata al server la domanda di elaborazione dei nostri dati, si viene indirizzati ad una pagina, che dopo qualche minuto, presenta i risultati ottenuti schematizzati in una tabella - suddivisa su più pagine - composta da sette colonne e un numero di righe pari alle interazioni totali restituite. Nell'ordine questa contiene: il simbolo del gene affiancato al suo id Ensembl, il nome del microRNA associato, lo score ottenuto dal modello, il suo p-value e tre colonne rappresentanti i databases di accop-

Figura 25: Screenshot della pagina web di TaLasso relativa al download dei risultati



piamenti *in silico* considerati dallo strumento, che nel caso di validazione riportano un simbolo di spunta. Cliccando sull'identificativo di una delle molecole di RNA mostrate (mRNAo miRNA) è possibile esplorare i soli risultati riferiti a quest'ultima.

In questa pagina, cliccando su uno dei tre link disponibili, è possibile ottenere: la matrice dei coefficienti β_{ji} stimati, i vettori degli id dei geni e dei simboli dei microRNA utilizzati, che corrispondono rispettivamente alle righe e alle colonne della precedente matrice (Figura 25). Il web tool restituisce esclusivamente quelle associazioni con un p-value inferiore a 0.05, senza la possibilità di poter scaricare anche quest'ultima informazione.

Risultati Non avendo a disposizione il softwer adeguato per l'utilizzo del codice messo a disposizione dagli autori, è chiaro come il web tool non possa permetterci di calcolare, tramite la convalida incrociata, il parametro globale ottimale per i nostri dati.

La soluzione adottata sarà dunque la più semplice: il valore κ^G, utilizzato per ottenere i risultati che verranno confrontati con gli altri modelli nel successivo capitolo, sarà quello che produrrà il miglior esito in termini

di validati, con una preferenza per gli indicatori relativi alle prime 500 e 1000 posizioni della classifica. Ovviamente va ricordato che questa strategia porterà a misuratori di performance distorti in senso ottimistico.

Riportiamo nella seguente tabella i risultati ottenuti, per diverse scelte del parametro di lisciamento globale.

Lista	$\mathbf{N^T}$	**t**	$\mathbf{N^S}$	$\mathbf{V^{tot}}$	$\mathbf{V^{500}}$	$\mathbf{V^{1000}}$	$\mathbf{\%^{tot}}$	$\mathbf{\%^{500}}$	$\mathbf{\%^{1000}}$
$TL^{TS}_{1/2}$	na	na	30.071	1.608	31	69	5,35	6,2	6,9
$TL^{TS}_{1/3}$	na	na	30.071	1.608	37	70	5,35	7,4	7,0
$TL^{TS}_{1/5}$	na	na	30.071	1.608	46	95	5,35	9,2	9,5
$TL^{TS}_{1/10}$	na	na	30.071	1.608	58	94	5,35	11,6	9,4
$TL^{TS}_{1/50}$	na	na	30.071	1.608	54	92	5,35	10,8	9,2
$TL^{SVR}_{1/2}$	na	na	178.800	2.862	6	14	1,6	1,2	1,4
$TL^{SVR}_{1/3}$	na	na	178.800	2.862	9	18	1,6	3,4	1,8
$TL^{SVR}_{1/5}$	na	na	178.800	2.862	17	26	1,6	3,4	2,6
$TL^{SVR}_{1/10}$	na	na	178.800	2.862	17	34	1,6	3,4	3,4
$TL^{SVR}_{1/50}$	na	na	178.800	2.862	12	29	1,6	2,4	2,8

.

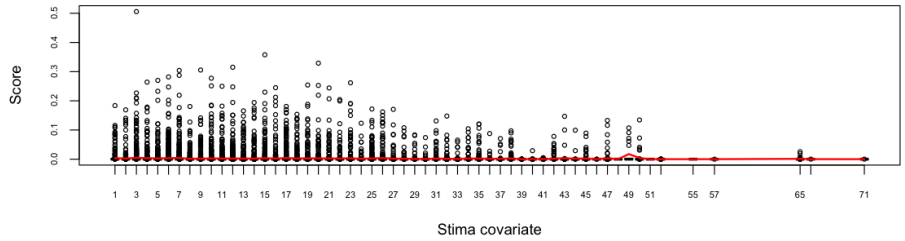
Appare evidente come l'algortimo TaLasso si dimostri robusto a variazioni del parametro di lisciamento: le associazioni restituite, nei gruppi di liste dello stesso predittore, sono le medesime sia per TS che per SVR.

Valutando la percentuale di validati nel top delle classifiche, la nostra scelta ricade quindi sui modelli $TL^{TS}_{1/10}$ e $TL^{SVR}_{1/10}$.

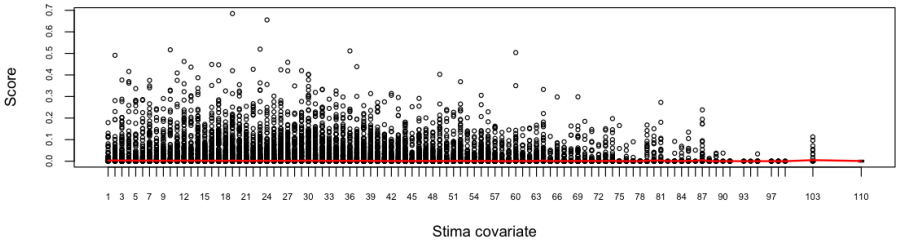
Distorsione rispetto al numero di covariate Il modello TaLasso rientra a tutti gli effetti tra i modelli multivariati, per questo è di interesse analizzare la distribuzione dei coefficienti rispetto al numero di covariate utilizzate nella stima.

Intersecando la lista di predizione dei due databases utilizzati per l'implementazione in R dei precedenti metodi, notiamo che alcuni appaiamenti

Figura 26: Sono rappresentate le distribuzioni dei coefficienti ottenuti attraverso il web tool TaLasso in funzione del numero di covariate, stimate utilizzando le nostre liste di predizioni - TS (a) e SVR (b). La linea rossa continua rappresenta la media per gruppo.



(a) $TL^{TS}_{1/10}$



(b) $TL^{SVR}_{1/10}$

restituiti dal web tool non rientrano all'interno di questi insiemi. Trattandosi di un numero irrisorio di associazioni rispetto al totale restituito - 876 su 30.071 per TS, e 956 su 178.800 per SVR - decidiamo di utilizzare le nostre informazioni sulla numerosità delle covariate per ciascun mRNA, con la consapevolezza che potrebbero non essere del tutto corrette ma anche con la speranza che non si discostino molto dai nostri valori.

Quello che otteniamo in Figura 26 è un risultato sorprendente: al contrario di tutti gli altri metodi multivariati, l'algoritmo TaLasso non sembra risentire minimamente del bias dovuto al diverso numero di miRNA entrati in ciascun modello. Le distribuzioni degli score infatti sono completamente schiacciate sullo zero e la linea continua rossa, che rappresenta la media per gruppo, è del tutto piatta.

5 Verifiche & Confronti

Al fine di migliorare la conoscenza dei meccanismi genetici e molecolari che legano miRNA e mRNA, in questo lavoro di tesi sono stati proposti diversi metodi per integrare le predizioni *in silico* dei profili di espressione.

Alcuni di questi sono incentrati sulla forza e sulla direzione di coefficienti che basano la loro stima su un tipo di relazione uno-a-uno - rinominati metodi pair-wise - mentre altri sfruttano le informazioni date dall'insieme dei microRNA che agiscono su uno stesso target, esprimendo così una relazione molti-a-molti - rinominati metodi multivariati. Per questo ultimo gruppo, ad eccezione della Correlazione Parziale, il modello di ispirazione è stata la regressione lineare penalizzata attraverso il vincolo L_1, ossia il Lasso. La scelta di questa penalizzazione rende però il problema di minimizzazione non lineare rispetto alla variabile risposta, fornendo così una soluzione non esprimile in forma chiusa che necessita di una soluzione numerica: per questo motivo abbiamo proposto tre differenti implementazioni dell'algoritmo di stima.

Questo capitolo verrà dedicato al confronto quantitativo tra i metodi proposti utilizzando, come criterio elettivo per valutare le diverse performance, le interazioni riprodotte *in vitro* che hanno dato esito positivo sulla reale regolazione tra le due molecole, contenute in quattro diverse raccolte: TarBase, miRecord, mirTarBase e mirWalk. L'assunzione alla base del nostro criterio di valutazione sarà la seguente: ordinando ciascuna coppia miRNA-mRNA sulla base della forza inversa del legame - espressa dallo score restituito da ciascun modello - quante più coppie validate verranno posizionate in cima alla classifica, tanto più si suppone che il metodo colga la vera forma della regolazione.

Come prima cosa dedicheremo un paragrafo alla valutazione dell'influenza che la scelta o meno di un sottoinsieme iniziale ha sui diversi modelli. Poi, passeremo a stabilire se le ipotesi avanzate negli articoli passati in rassegna trovano riscontro anche nel nostro datasets, e quindi sono effettivamente generalizzabili al meccanismo di appaiamento delle nostre molecole di RNA.

In particolare, ci domanderemo se l'inserimento dell'informazione sul RISC apporti un vantaggio competitivo in termini di predizione dei veri accoppiamenti; se la scelta delle interazioni negative attraverso l'imposizione di un ulteriore vincolo sui coefficienti da stimare, sia migliore rispetto a quella fatta a posteriori; e infine avanzeremo e cercheremo di dimostrare una nostra congettura nata dalle conclusioni tratte alle diverse ipotesi.

Per concludere, procederemo al vero e proprio confronto dei risultati ottenuti dai modelli candidati nel capitolo precedente, al fine di migliorare la comprensione dei meccanismi di base dell'appaiamento tra i microRNA e i loro geni target.

Per valutare in termini di arricchimento le classifiche restituite dai diversi modelli, oltre ad alcune semplici analisi descrittive o grafiche, ci avvarremo anche della distribuzione Ipergeometrica, opportunamente rivisitata per il caso in esame. Per questo motivo inseriamo un breve sottoparagrafo finalizzato all'introduzione di questa variabile casuale.

La distribuzione Ipergeometrica In teoria delle probabilità, la distribuzione Ipergeometrica rappresenta una variabile casuale discreta che può essere descritta come l'estrazione senza reinserimento di alcune palline - perdenti e vincenti - da un urna.

Formalmente $\mathbf{H}(n, n_v, k)$, descrive la variabile aleatoria N_h che conta, per k elementi distinti estratti a caso - in modo equiprobabile - da un insieme $\mathcal{A}$ di cardinalità n, quanti di quelli estratti appartengono al sottoinsie-

me $\mathcal{B}$ di cardinalità n_v. La probabilità di ottenere esattamente q elementi appartenenti all'insieme $\mathcal{B}$ è dunque:

$$\mathbb{P}(q) = \frac{\binom{n_v}{q}\binom{n-n_v}{q-k}}{\binom{n}{q}} \tag{27}$$

mentre il valore atteso è dato dalla formula:

$$\mathbb{E}[N_h] = \frac{k \times n_v}{n} \tag{28}$$

In termini più concreti, se applicata al nostro contesto, $\mathbf{H}(n, n_v, k)$ descriverà la probabilità di ottenere per caso una conformazione pari a quella della lista restituita dal $Modello^X$, partendo da un insieme di interazioni iniziali n, al cui interno risultano validate n_v di queste. Il parametro k rappresenterà il totale di accoppiamenti contenuti nella classifica analizzata, o il taglio effettuato ad una certa posizione, e q il numero di validati trovati in corrispondenza di quest'ultimo sottoinsieme. In questo modo sarà possibile valutare la capacità discriminatoria del modello rispetto alla scelta e all'ordinamento casuale delle interazioni, in funzione delle predizioni iniziali restituite da TS e SVR.

5.1 Validità delle Assunzioni e delle Ipotesi

5.1.1 Insieme iniziale

L'obiettivo di questo paragrafo non è quello di condurre uno studio comparativo degli algoritmi basti sulla complementarietà delle sequenze (A), argomento già trattato in letteratura in maniera esauriente, quanto quello di valutare l'influenza netta delle loro predizioni sul metodo (M) utilizzato per la loro intergrazione.Per cercare di studiare questo aspetto ci siamo avvalsi della distribuzione Ipergeometrica per calcolare - in relazione a ciascuna coppia A-M - la probabilità di ottenere un risultato migliore per caso, in termini di validati nelle prime posizioni della classifica restituita dal metodo.

Tabella 5: Principali valori delle distribuzioni Ipergeometriche riferite alle diverse scelte per l'insieme di interazioni iniziali.

Lista ($\mathcal{A}$)	**Variabile**	$\mathbf{N^T}$ ($\#\mathcal{A}$)	$\mathbf{V^{tot}}$ ($\#\mathcal{B}$)	$\mathbb{E}[\mathbf{N}]\vert_{\mathbf{k=500}}$	$\mathbb{E}[\mathbf{N}]\vert_{\mathbf{k=1000}}$	$\%^{\mathbf{k}}$
Nessuna	N_h^{na}	3.228.030	16.951	~ 3	~ 5	0,52
microRNA.org	N_h^{SVR}	309.836	4.379	~ 7	~ 14	1,40
Target Scan	N_h^{TS}	70.610	2.539	~ 18	~ 36	3,60

Concretamente, quando utilizzeremo TS la variabile casuale N si distribuirà come $\mathbf{H}(70.610, 2.535, k)$, mentre per SVR la distribuzione di N sarà $\mathbf{H}(309.836, 4.379, k)$, dove k rappresenterà in entrambi i casi il taglio effettuato in corrispondenza dei primi 1000 valori (Tabella 5).

Nel seguito riportiamo la probabilità $\mathbb{P}(q)$ calcolata per i tre metodi che, avendo una complessità computazionale relativamente bassa, ci hanno concesso piena libertà nella scelta dell'insieme iniziale di stima: in questo modo rendiamo interessante il confronto, mettendo in luce la risposta del metodo.

Lista	TargetScan	microRNA.org	Nessuna
R	$8{,}32^{e-06}$	8.21^{e-07}	1.58^{e-07}
GC	9.59^{e-05}	2.35^{e-07}	3.38^{e-05}
R_p	0,0056	0,1805	0,7692

.

Come è logico attendersi per i metodi pair-wise (R e GC), qualsiasi sia la scelta fatta, la probabilità che otteniamo risulta essere dello stesso ordine di grandezza: il sottoinsieme iniziale di variabili che decidiamo di utilizzare non aiuta il metodo a funzionare meglio, quanto piuttosto il miglioramento è dovuto alle caratteristiche intrinseche della lista stessa.

Per l'unico metodo multivariato invece, l'utilizzo di uno specifico set di predizioni iniziali, tende a decretare il successo o il totale insuccesso del metodo stesso: per R_p, la scelta di non ridurre in nessun modo le covariate iniziali entranti nella stima, portano i risultati ad avvicinarsi alla scelta casuale.

In altre parole, possiamo supporre che il miglioramento di ciascun $Metodo^X$ possa essere suddiviso, in funzione della forma di relazione espressa, nel seguente modo:

- Pair-Wise: $\text{Miglioramento}(M,A) = \text{Miglioramento}(M) + \text{Miglioramento}(A)$
- Multivariato: $\text{Miglioramento}(M,A) = \text{Miglioramento}(\text{M}|\text{A})$

Questo tipo di congettura verrà confermata poi, nel paragrafo sui confronti, anche per i restanti modelli utilizzati.

5.1.2 Vincolo di non-positività

Nella discussione finale dell'articolo da cui è stato tratto il modello TaLasso, troviamo la seguente affermazione:

> "Poco prima della pubblicazione di questo manoscritto siamo venuti a conoscenza di un lavoro simile di Lu e colleghi. [...] La principale differenza tra questo lavoro e il nostro è l'inclusione di un vincolo di non-positività nella regressione regolarizzata L_1. Qui, abbiamo dimostrato che l'aggiunta del vincolo di non-positività è cruciale ai fini della ricerca delle interazioni mRNA-miRNA."

Il lavoro a cui si riferiscono è quello concernente l'algoritmo ArgoLasso: non riportando nessuna prova tangibile a sostegno di questa tesi, abbiamo deciso di verificare l'effettiva utilità di questo vincolo aggiuntivo nella stima di regressione penalizzata.

TaLasso e ArgoLasso, pur utilizzando la stessa formulazione base, fatta appunto eccezione per il vincolo aggiuntivo del primo, in realtà differiscono per tutti i restanti aspetti, anch'essi cruciali ai fini della qualità dei risultati. In particolare sono discordanti per algoritmo di stima, scelta del parametro di lisciamento, scelta del taglio sulla significatività e scelta sui criteri di

ordinamento. Queste motivazioni, aggiunte all'impossibilità di modificare attraverso il codice l'algoritmo di TL, impediscono di fatto un confronto sull'influenza del vincolo al netto delle restanti condizioni, attraverso l'utilizzo delle procedure originali.

Per risolvere queste complicazioni abbiamo optato per l'utilizzo di un ulteriore implementazione del Lasso, che permettesse una maggiore flessibilità sulle scelte dei parametri e sulle restanti opzioni, consentendoci di emulare i due modelli da confrontare: la funzione R di cui stiamo parlando è *penalized()*. In particolare abbiamo per entrambi adoperato il parametro di lisciamento ottimale, ottenuto attraverso una funzione prevista nella libreria che sfrutta la convalida incrociata, mantenendo la stessa funzione di stima ma variando l'opzione sui vincoli aggiuntivi. Per maggiori informazioni sulla procedura si rimanda al Paragrafo 1.3.3 .

Riportiamo i risultati ottenuti nella seguente tabella, suddividendoli per algoritmo di predizioni iniziali utilizzato.

Lista	$\mathbf{N^T}$	**t**	$\mathbf{N^S}$	$\mathbf{V^{tot}}$	$\mathbf{V^{500}}$	$\mathbf{V^{1000}}$	$\%^{\mathbf{tot}}$	$\%^{\mathbf{500}}$	$\%^{\mathbf{1000}}$
PZ_{AL}^{TS}	5.372	-	5.372	228	33	58	4,24	6,6	5,8
PZ_{TL}^{TS}	5.924	-	5.924	243	30	59	4,10	6,0	5,9
PZ_{AL}^{SVR}	14.114	-	14.114	241	13	27	1,71	2,6	2,7
PZ_{TL}^{SVR}	17.568	-	17.568	283	13	24	1,61	2,6	2,4

.

Alla luce dei risultati ottenuti è evidente come la tesi sostenuta con fermezza da Muniategui e colleghi sia del tutto incosistente: con l'uso di entrambi i sottoinsiemi iniziali, il numero di interazioni restituite e le percentuali di validazioni sui diversi tagli della lista risultano molto simili tra loro, se non addirittura migliori per l'emulazione di ArgoLasso, che ricordiamo non utilizza nessun vincolo. D'altro canto però, notiamo anche che siamo ben

distanti dall'ottenere per il sostituito PZ_{TL}^{TS} gli ottimi risultati originali.

In conclusione ipotizzando che, come dicono gli autori stessi la scelta dell'algoritmo di ottimizzazione di TL sia dipesa principalmente dalla sua efficienza computazionale in termini di tempo, avanziamo l'ipotesi secondo cui l'unica vera differenza tra i due modelli non sia l'aggiunta del vincolo, ma piuttosto il processo di stima del parametro di lisciamento ottimale.

5.1.3 Parametro di lisciamento

Negli esperimenti in cui l'obiettivo principale è la selezione delle variabili attraverso gli algoritmi basati sul modello Lasso, è stato dimostrato che quando l'accuratezza delle previsioni è utilizzata come criterio di scelta per il parametro di lisciamento, in generale la procedura risulta inconsistente indipendentemente dalla numerosità campionaria a disposizione[65]. Ovvero, reinterpretando il significato nel nostro contesto, il sottoinsieme di microRNA selezionati non risultano essere il vero sottoinsieme di regolatori del gene target a loro associato.

Nell'ottica di quanto riportato ci siamo allora focalizzati sul modo in cui il parametro di lisciamento ottimale viene scelto per i tre algoritmi della stima Lasso proposti in questo lavoro, notando subito la principale differenza: mentre Penalized e ArgoLasso, si avvalgono entrambi di una misurazione basata sull'errore commesso dalla predizione - nell'ordine la convalida incorciata e la statistica Cp - TaLasso attraverso la Formula 9 sfrutta esclusivamente l'informazione dei dati. Ci sembra dunque ragionevole estendere al nostro dataset le considerazioni fatte dagli autori dell'articolo citato, e verificare i risultati ottenuti.

Per confermare la fondatezza delle nostre affermazioni, abbiamo utilizzato la Formula 10 per la stima del parametro globale, ottenendo un numero di λ_i^{max} pari al numero di mRNA contenuti nella lista iniziale TS[22] e scegliendo di utilizzare diverse frazioni κ del massimo di questi ultimi.

I risultati, accostati a quelli derivanti dall'utilizzo del lambda basato sull'accuratezza delle previsioni, sono riassunti nella seguente tabella.

Lista	κ^G	N^T	V^{tot}	V^{500}	V^{1000}	$\%^{tot}$	$\%^{500}$	$\%^{1000}$
PZ_{AL}^{TS}	-	5.372	228	33	58	4,24	6,6	5,8
PZ_{ALg}^{TS}	$\frac{1}{5}$	12.114	495	40	68	4,09	8,0	6,8
PZ_{ALg}^{TS}	$\frac{1}{10}$	19.058	755	40	78	3,96	8,0	7,8
PZ_{ALg}^{TS}	$\frac{1}{20}$	25.148	967	43	76	3,84	8,6	7,6
PZ_{ALg}^{TS}	$\frac{1}{50}$	30.689	1.148	38	71	3,74	7,6	7,1
PZ_{ALg}^{TS}	$\frac{1}{100}$	32.897	1.224	37	69	3,72	7,4	6,9
PZ_{TL}^{TS}	-	5.924	243	30	59	4,10	6,0	5,9
PZ_{TLg}^{TS}	$\frac{1}{5}$	11.279	471	39	65	4,18	7,8	6,5
PZ_{TLg}^{TS}	$\frac{1}{10}$	15.762	613	38	68	3,89	7,6	6,8
PZ_{TLg}^{TS}	$\frac{1}{20}$	18.501	706	40	67	3,82	8,0	6,7
PZ_{TLg}^{TS}	$\frac{1}{50}$	20.268	766	40	64	3,78	8,0	6,4
PZ_{TLg}^{TS}	$\frac{1}{100}$	20.838	786	39	64	3,77	7,8	6,4

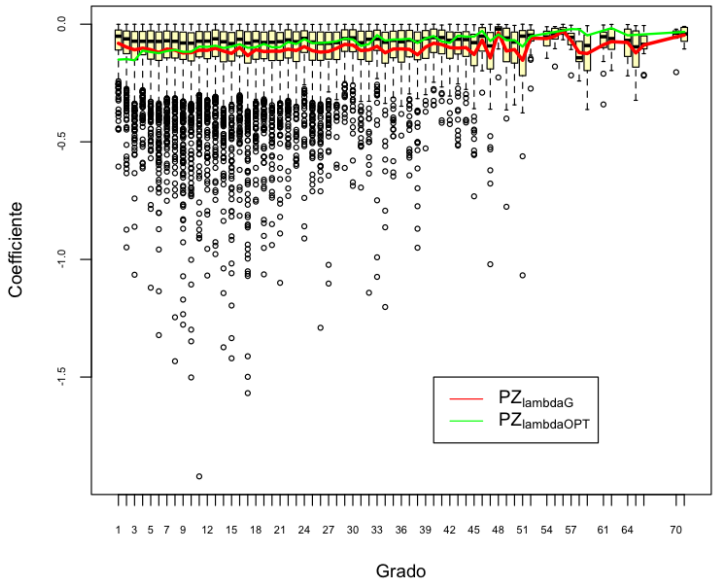
.

Risulta evidente che basando la scelta del parametro di lisciamento sulle informazioni dei dati, rispetto che sull'accuratezza del modello, si assiste ad un netto miglioramento delle percentuali di validati nelle prime posizioni delle classifiche restituite, indipendentemente dalla frazione κ^G scelta. Inoltre, si registra anche ad un'attenuazione della distorsione dovuta al numero di covariate iniziali introdotte per la stima (Figura 27).

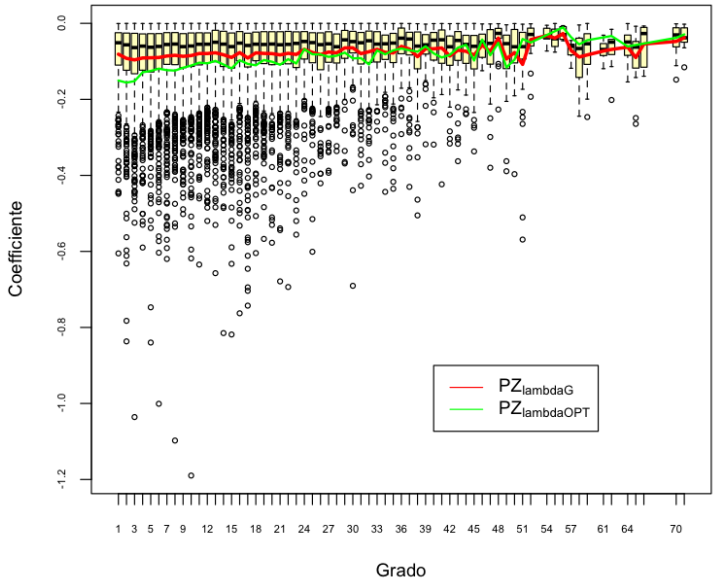
L'effetto rilevato, anche se non risulta ancora pari a quello di TL, ci porta a confermare la nostra ipotesi suggerendo l'utilizzo di un metodo di scelta differente per il lambda ottimale da utilizzare.

[22] Viene utilizzata esclusivamente la lista di predizioni iniziali TargetScan per rendere più agevole l'esposizione.

Figura 27: Distribuzioni dei coefficienti stimati per i modelli che emulano rispettivamente ArgoLasso (a) e TaLasso (b), attraverso la funzione *penalized()* e parametro di stima basato sull'informazione dei dati. Con linea continua rossa sono rappresenate le medie dei coefficienti in funzione del numero di covariate dei modelli appena descritti, mentre in verde la media dei modelli con parametro di lisciamento stimato tramite convalida incrociata.



(a) PZ^{TS}_{ALg}



(b) PZ^{TS}_{TLg}

Nello studio citato all'inizio di questo paragrafo, vengono suggerite delle soluzioni alternative da quella utilizzata per il modello TaLasso: un lavoro futuro potrebbe studiare in maniera più approfondita l'influenza di queste proposte sui dati di espressione,nel contesto della predizone dei target dei microRNA.

5.1.4 Proteine Argonaute

Nell'articolo relativo al metodo ArgoLasso, si sostiene che l'inserimento dell'informazione relativa alla concentrazione del RISC, aumenti sensibilità e specificità delle nostre predizioni. Non avendo però ottenuto segnali forti quanto quelli ottenuti dagli autori del metodo, tali da supportare a pieno questa ipotesi, abbiamo deciso di approfondire l'analisi conducendo un piccolo esperimento.

Nel tentativo di cogliere variazioni sistematiche che facessero intuire il meccanismo alla base del modello ArgoLasso, abbiamo sostituito ai veri valori Ago_2 e Ago_3 in (?), i profili di espressione di alcuni geni estratti casualmente all'interno del nostro esperimento di microarray (Figura 28). In particolare abbiamo suddiviso i geni estratti secondo la seguente classificazione:

$Ago_j^{=}$: distribuzione simile a quella del vero Ago_j

$Ago_j^{<}$: distribuzione con valori di espressione inferiori a quelli del vero Ago_j

$Ago_j^{>}$: distribuzione con valori di espressione superiori a quell del vero Ago_j

A seguito della costruzione delle matrici necessarie per l'implementazione di ArgoLasso (vedi Paragrafo 4.3.2) abbiamo utilizzato la versione *OneRun* dell'algoritmo e calcolato l'AUC - misura di sintesi per esprimere il miglioramento ottenuto in termini di specificità e sensibilità - attraverso il metodo Bootstrap utilizzando solamente 5 ripetizioni, ossia il numero minimo di ite-

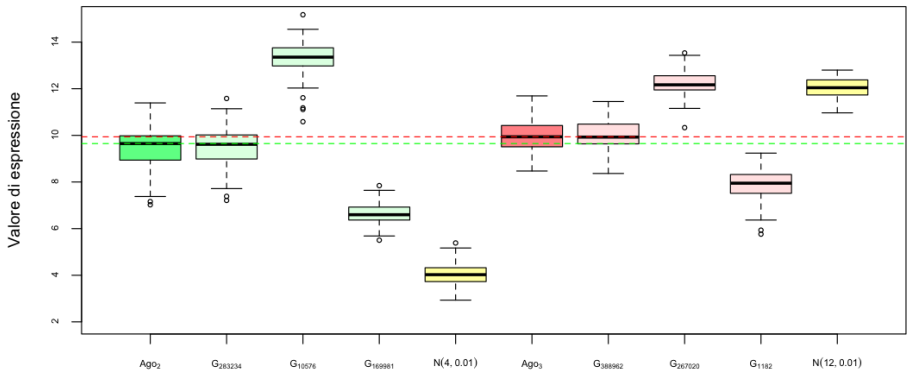
razioni che garantivano un adeguata sensibilità alle variazioni registrate.

I risultati ottenuti per tutte le combinazioni calcolate sono riportate nella seguente tabella, dove in cima troviamo il vero modello, poi i modelli che contengono rispettivamente il valore di Ago_2 o Ago_3 originale, e infine quelli che contengono solo geni estratti in maniera casuale.

Ago	$Me^{A_3} - Me^{A_2}$	N^T	V^T	$\%^{tot}$	AUC
Ago_2 vs Ago_3	0,29	7.908	301	3,81	0,711
Ago_2 vs $Ago_3^{=}$	0,28	8.515	339	3,98	0,727
Ago_2 vs $Ago_3^{>}$	2,52	7.861	327	4,16	0,719
Ago_2 vs $Ago_3^{<}$	-1,70	9.354	330	3,53	0,728
$Ago_2^{=}$ vs Ago_3	0,32	6.602	252	3,87	0,721
$Ago_2^{>}$ vs Ago_3	-3,42	9.196	348	378	0,700
$Ago_2^{<}$ vs Ago_3	3,34	7.421	297	4,00	0,716
$Ago_2^{<}$ vs $Ago_3^{>}$	5,57	6.807	250	3,68	0,732
$Ago_2^{>}$ vs $Ago_3^{<}$	-5,40	10.385	420	4,04	0,676
$Ago_2^{>}$ vs $Ago_3^{>}$	-1,19	7.401	274	3,70	0,720
$Ago_2^{<}$ vs $Ago_3^{<}$	1,36	9.748	394	4,04	0714

Anzitutto è evidente come, qualsiasi scelta dei valori rappresentativi delle proteine Argonaute, apporti un beneficio in termini di AUC rispetto al

Figura 28: Distribuzione dei geni scelti per l'esperimento: in verde chiaro sono rappresentati quelli utilizzati in sostituzione del termine Ago_2, mentre in rosso chiaro quelli utilizzati in sostituzione di Ago_3. Le due linee tratteggiate rappresentano il valore mediano degli AGO originali.



modello L_O^{TS} - che ricordiamo essere pari a 0.686 - e addirittura come il miglior risultato corrisponda all'utilizzo di entrambi termini casuali. Notando inoltre che sembra esserci una relazione inversa che lega la distanza delle distribuzioni dei due profili AGO, calcolata come la differenza tra le mediane, presumiamo che il miglioramento delle performance del modello siano legate a quest'ultimo aspetto.

Per confermare questa ipotesi ripetiamo l'esperimento, ma questa volta utilizzando due variabili casuali indipendenti con medie posizionate su valori distanti tra loro e varianza molto bassa: in particolare utilizzeremo un campione di 76 osservazioni provenienti per Ago_3 da una Normale(μ=12,σ=0.01) (z_3), e per Ago_2 da una Normale(μ=4,σ=0.01) (z_2). I risultati sono riassunti nel seguito.

Ago	$Me^{A_3} - Me^{A_2}$	N^T	V^T	$\%^{tot}$	AUC
Ago_2 vs Ago_3	0,29	7.908	301	3,81	0,711
$Ago_2^<$ vs $Ago_3^>$	5,57	6.807	250	3,68	0,732
$Ago_2^>$ vs $Ago_3^<$	-5,41	10.385	420	4,04	0,676
z_2 vs z_3	8	5.230	212	4,05	0.740

Come previsto, all'aumentare della distanza mediana - con segno positivo - aumentiamo la specificità del modello di previsione. Ripetendo il procedimento, per distribuzioni sempre più distanziate, il miglioramento del valore dell'AUC raggiunge ad un certo punto una stabilità, nonostante per tutti gli esperimenti si migliori il valore originale[23].

Possiamo quindi concludere che l'effetto, ottenuto attraverso l'introduzione del termine che rappresenta la concentrazione del RISC, è dovuto ad un artificio dell'algoritmo di stima di ArgoLasso: poiché abbiamo precedentemente dimostrato che la scelta del parametro di lisciamento, nei metodi di restringimento, è centrale ai fini di una corretta classificazione, la distorsione

[23]Nell'esperimento con variabili casuali Normali abbiamo riportato il miglior risultato ottenuto, e l'AUC ottenuto può essere definito come la soglia superiore.

introdotta aiuta in questo senso, funzionando come un aggiustamento per il parametro di penalità.

Utilizzando nuovamente Penalized per imitare il meccanismo di ArgoLasso, introducendo questa volta l'informazione sulle proteine Argonaute e facendo affidamento alla stima del parametro di lisciamento del metodo TaLasso che dovrebbe rendere più stabile le predizioni finali, abbiamo riprodotto il precedente esperimento. Riportiamo alcune delle prove effettuate.

Modello	Ago	κ^G	N^T	V^T	V^{500}	V^{1000}	$\%^{tot}$	$\%^{500}$	$\%^{1000}$
PZ_{AL}^{TS}	-	$\frac{1}{20}$	25.148	967	43	76	3,84	8,6	7,6
$ArgoPZ_{AL}^{TS}$	Ago_2 vs Ago_3	$\frac{1}{20}$	12.847	500	40	65	3,89	8,0	6,5
$ArgoPZ_{AL}^{TS}$	$Ago_2^{=}$ vs $Ago_3^{=}$	$\frac{1}{20}$	14.199	551	22	61	3,88	4,4	6,1
$ArgoPZ_{AL}^{TS}$	$Ago_2^{<}$ vs $Ago_3^{<}$	$\frac{1}{20}$	14.781	584	44	76	3,95	8,8	7,6
$ArgoPZ_{AL}^{TS}$	z^2 vs z^3	$\frac{1}{20}$	1.729	66	17	37	3,82	3,4	3,7

.

Con questa diversa implementazione vediamo che, se confrontato con il modello originale, il miglioramento dell'inserimento dei veri termini *Ago* non sussiste e, utilizzando geni estratti a caso, il numero di validati nel top della classifica è molto variabile, riuscendo in alcuni casi ad ottenere la migliore performance assoluta.

Concludendo ci sentiamo di sostenere che per come è stato proposto l'inserimento dell'informazione sul RISC, ossia attraverso il livello dell'mRNA che codifica per la determinata proteina AGO, questa non è utile ai fini della predizione quando si utilizza un modello di regressione penalizzata come il Lasso.

5.2 Confronti finali

In questa sezione dedicata al confronto, per una maggiore chiarezza espositiva, non tutti i modelli utilizzati verranno presentati, quindi appare opportuno fare alcune precisazioni sui motivi di tale scelta.

Anzitutto è chiaro che per ArgoLasso, visti i problemi sulla discretizzazione dello score utilizzato - discussi nel Paragrafo 4.3.2 - il confronto sulla base del top della classifica sarebbe del tutto inefficace e per questo viene escluso. Al suo posto decidiamo di utilizzare PZ_{AL} ma di omettere PZ_{TL} poichè, oltre all'inutilità dimostrata dell'inserimento del vincolo, dall'intersezione delle due liste, notiamo che il contenuto informativo è il medesimo: circa l'88% delle validazioni sul totale risultano essere comuni ad entrambi.

Inoltre, nonostante le buone qualità dei modelli PZ_{ALg} e PZ_{TLg}, ispirati in tutto e per tutto a TaLasso, non raggiungendo gli stessi risultati ci sembra ragionevole decidere di utilizzare solamente la proposta originale, senza costringerci a duplicare inutilmente nel seguito osservazioni e risultati. In particolare, scorrendo la classifica completa si può osservare che, nonostante le interazioni siano buona parte comuni, l'emulatore di TL tende a posizionare tutte le validazioni all'interno dei top 500 verso le ultime posizioni.

Escludiamo inoltre, tutte quelle misure applicate ai dati senza l'utilizzo di un algoritmo di predizioni iniziali, in seguito alla conferma della correttezza dell'utilità dell'informazione sul meccanismo della complementarietà di sequenza tra le coppie miRNA-mRNA.

5.2.1 Analisi descrittive

La sezione sulle analisi descrittive, prevede principalmente un confronto sulla base delle percentuali di interazioni corrette trovate su diversi tagli della lista finale ottenuta.

La tabella di riepilogo per i soli modelli mantenuti, suddivisi per lista di predizioni iniziali fornite e relazione espressa, è riportata nel seguito.

Lista	N^T	N^S	V^{tot}	V^{500}	V^{1000}	$\%^{tot}$	$\%^{500}$	$\%^{1000}$
R^{TS}	36.441	8.310	393	35	63	4,73	7,0	6,3
GC^{TS}	40.585	6.880	351	37	59	5,10	7,4	5,9
MI^{TS}	na	5.628	240	23	39	4,26	4,6	3,9
R_p^{TS}	34.807	3.520	154	26	51	4,37	5,2	5,1
PZ_{AL}^{TS}	5.372	5.372	228	33	58	4,24	6,6	5,8
$TL_{1/10}^{TS}$	na	30.071	1.608	58	94	5,35	11,6	9,4
R^{SVR}	151.015	26.215	528	21	37	2,01	4,1	3,7
GC^{SVR}	169.537	23.707	533	22	45	2,25	4,4	4,5
MI^{SVR}	na	4.715	62	7	18	1,31	1,4	1,8
R_p^{SVR}	153.817	15.492	2324	11	17	1,51	2,2	1,7
PZ_{AL}^{SVR}	14.114	14.114	241	13	27	1,71	2,6	2,7
$TL_{1/10}^{SVR}$	na	178.800	2.862	17	34	1,6	3,4	3,4

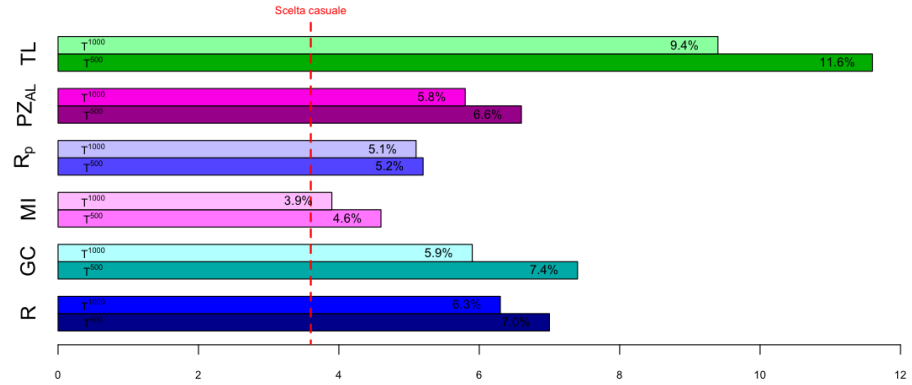
.

Poichè il numero di interazioni significative (N^S), restituite per i diversi metodi, ricopre un range di valori molto ampio - [3.520,30.071] per TS e [4.715.178.800] per SVR - e ad alcune liste non è stato applicato nessun taglio sulla significativà, ci sembra poco opportuno commentare le percentuali di validati ottenute rispetto al totale.

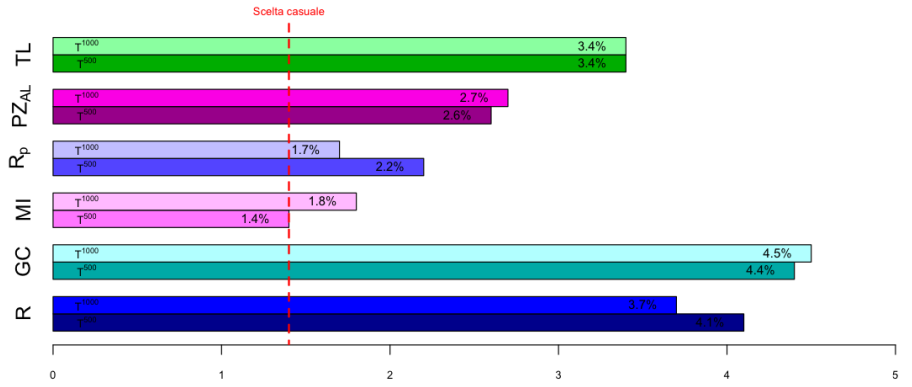
Nell'ottica di utilizzare un piccolo sottoinsieme di queste liste, come risorsa a livello sperimentale per i biologi che si occupano dello studio del ruolo funzionale dei miRNA, il vero contenuto informativo è da ricercarsi all'interno delle frazioni di associazioni validate tra le prime 500/1000 posizioni della classifica: i barplot riportati in Figura 29 ci danno immediatamente un'idea grafica di come agiscono i modelli.

La prima cosa che notiamo è che per la Correlazione Parziale e la Mutua Informazione, in entrambi i grafici, il vantaggio apportato è decisamente modesto. In particolare per MI raggiungiamo addititura l'inefficienza: questa evidenza non stupisce vista l'impossibilità del metodo di distinguere la direzione della relazione regolatoria; al contrario, fornisce un'ulteriore riprova a

Figura 29: Grafico a barre della percentuale di iterazioni validate nelle prime 500 (colori più scuri) e 1000 (colori più chiari) posizioni della classifica ordinata, suddivista per lista di predizioni iniziali fornite da TS (a) e SVR (b). La linea rossa tratteggiata rappresenta la percentuale che si otterrebbe attraverso l'ordinamento casuale delle coppie miRNA-mRNA.



(a) TargetScan



(b) microRNA.org

supporto della regolazione inversa dei microRNA rispetto al loro gene target.

Per quanto concerne i modelli restanti, riscontriamo lo stesso fenomeno riportato nel primo paragrafo di questo capitolo: quando ai nostri metodi multivariati, basati sulla sparsità delle soluzioni, si forniscono un numero troppo elevato di interazioni iniziali, questi sono portati a funzionare peggio rispetto ai metodi pair-wise, che non risentono di questa problematica direttamente all'interno della stima dei coefficienti utilizzati per l'ordinamento. Quando però la lista fornita è quella dell'algortimo TargetScan, il modello TL non ha rivali, dimostrandosi abbondantemente superiore in termini di percentuali.

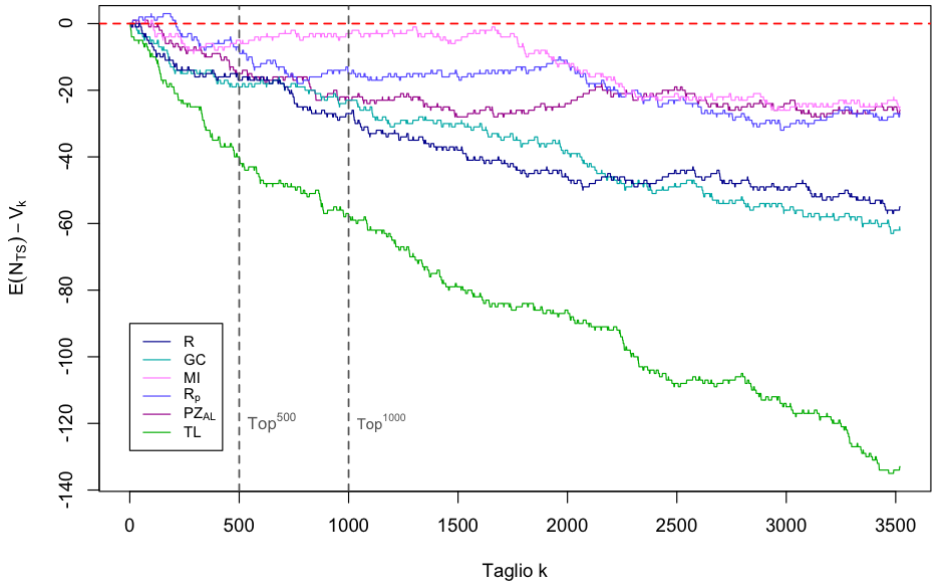
5.2.2 Analisi Ipergeometrica

Utilizzando la distribuzione Ipergeometrica, come descritto nell'introduzione, cerchiamo di investigare l'intera struttura delle liste, invece che analizzare le singole percentuali. In particolare, la nostra analisi mostrerà il numero di validati in più previsti, rispetto al valore atteso che ci si attenderebbe di osservare nell'ordinamento casuale della lista di predizioni iniziali, fornita in ingresso a ciascun modello.

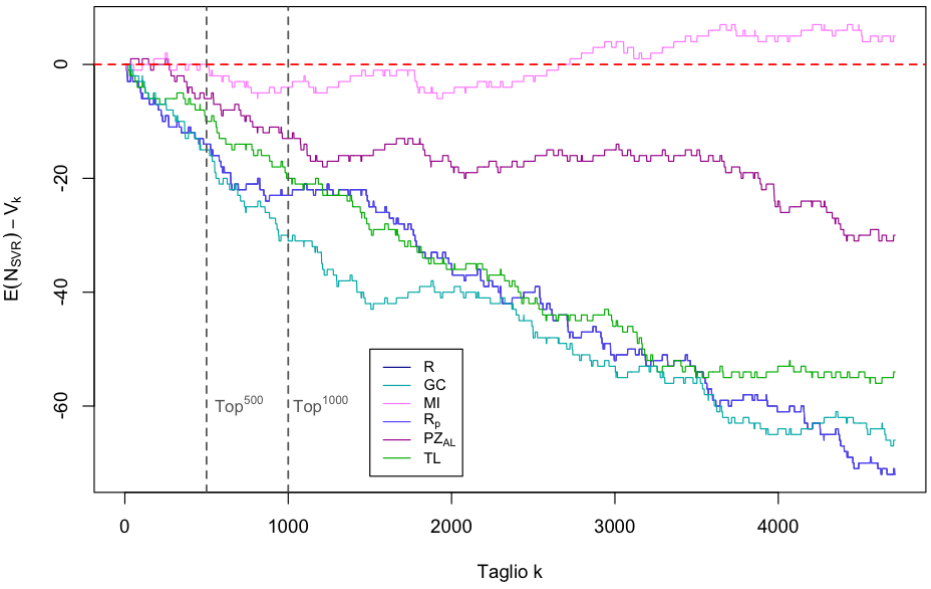
Per fare questo, abbiamo prima calcolato per ogni taglio possibile (k) il numero di validati atteso per TS e SVR, ed a questo valore abbiamo poi sottratto il numero di validati che il $Modello^X$ trova effettivamente (V^k). Unendo questi valori, quello che estraiamo è una rappresentazione grafica del miglioramento apportato dal nostro modello lungo tutte le posizioni della graduatoria (Figura 30).

Notiamo subito come, per le primissime posizioni, le uniche a permanere al di sotto alla soglia casuale sono TL, GC e R, per poi protrarre il mi-

Figura 30: Curve del miglioramento ottenuto da ciascun metodo, in termini di numero di validati previsti rispetto al valore atteso per le liste TS (a) e SVR(b)



(a) TargetScan



(b) microRNA.org

glioramento lungo tutta la finestra di osservazione mantenendo la medesima pendenza, ad eccezione di TaLasso che in realtà decresce con una maggiore angolazione nelle prime 500 posizioni. Questa proprietà rende TL un ottimo candidato ai fini della restituzione di un piccolo set di interazioni molto arricchite.

Penalized invece, sebbene riesca a recuperare la défaillance iniziale portandosi velocemente su valori comparabili a GC e R, dalla posizione 1500 in poi si riposiziona su valori più alti, stabilizzandosi alla stessa altezza delle inefficienti MI e R_p.

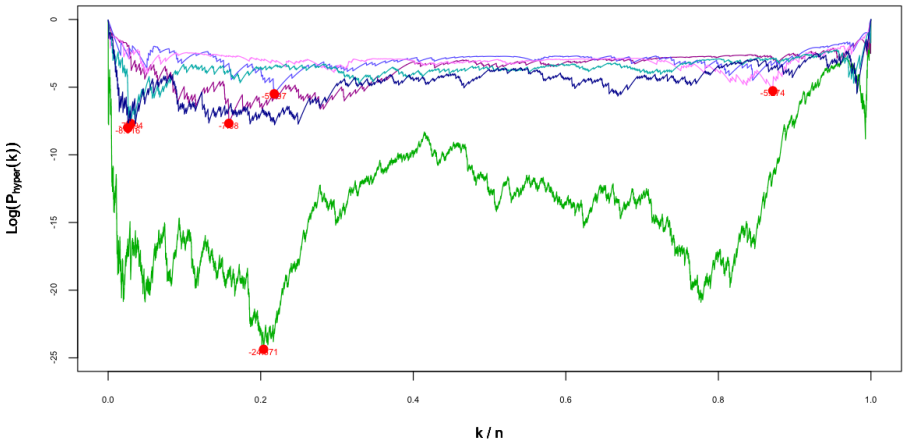
Possiamo estendere le precedenti riflessioni anche alla seconda lista di predizioni iniziali utilizzata, ad esclusione di TL: pur non esibendo le stesse apprezzabili qualità viste con l'utilizzo di TargetScan, notiamo che, considerando uno spettro di osservazione più ampio, è in grado di difendersi efficacemente rispetto agli altri modelli Paire-wise, fornendo buone prestazioni nella parte iniziale, e riportandosi oltre le mille osservazioni su valori prossimi a quelli di GC ed R; questo a dimostrazione del fatto che i soli valori percentuali possono alle volte trarre in inganno.

Non fa eccezione invece la Mutua Informazione che continua a dimostrarsi inadeguata, peggiorando considerevolmente nel secondo caso.

5.2.3 Analisi di arricchimento individuale

Con questa analisi vogliamo brevemente studiare le caratteristiche intrinseche di ciascuna lista nel posizionamento dei validati all'interno della classifica. Ci siamo serviti sempre della distribuzione geometrica, ma utilizzando questa volta come insieme $\mathcal{A}$ la classifica fornita da ciascun modello e come numero di estrazioni "vincenti" il numero di validati totali (V^{tot}) all'interno di queste. Le curve così definite sono rappresentate in Figura 31.

Figura 31: Arricchimento di ciascun metodo, sulla base delle caratteristiche intrinseche di ciascuna lista, per i risultati ottenuti con le sole predizioni inziali di TargetScan.



Ovviamente trattandosi di classifiche con lunghezze ben diverse, e al cui interno esistono percentuali di validati differenti, le curve non possono essere direttamente confrontate tra loro in termini di logaritmo di probabilità, ma offrono comunque alcuni spunti. Ad esempio, osservando la posizione dei pallini rossi, che rappresentano il punto di maggior arricchimento, possiamo stabilire la frazione in corrispondenza della quale il modello funziona meglio nell'identificare le reali regolazioni: più il taglio è su frazioni basse, più la lista tende a funzionare meglio nelle prime posizioni.

Esaminando le curve ottenute non ci stupisce vedere che sia GC che R ottengano il loro miglior arricchimento nel primo 5% della lista restituita, seguite nell'ordine, con una valore che si aggira attorno allo 0.2, PZ, TL e R_p. TaLasso in questo caso non si dimostra nettamente al di sopra degli altri a causa di alcuni minimi locali: in ogni caso è evidente come la rapida discesa iniziale inidichi un buon comportamento anche nelle primissime frazioni.

6 Discussione & Conclusioni

Negli ultimi anni, lo sviluppo di piattaforme di microarry per l'analisi dell'espressione dei microRNA, ha rivelato che molte di queste molecole sono espresse in maniera anomala nei tumori, rappresentando quindi una nuova classe di oncogeni e geni oncosopressori.

A tale riguardo, l'identificazione di miRNA ´´cancro-specifici" e dei loro bersagli molecolari, rappresenta un passaggio chiave per caratterizzare il loro ruolo nella tumorogenesi e potrebbe essere importante per l'identificazione di nuovi bersagli terapeutici.

Un primo passo è stato fatto attraverso la creazione di software bioinformatici che sfruttassero il principale meccanismo noto di appaiamento, ossia la complementarietà della sequenza nucleotidica, ma si sono dimostrati afflitti da una significativa frazione di falsi positivi, a causa della natura molti-a-molti delle relazioni predette.

Recentemente si è cercato di migliorare la specificità di questi algoritmi attraverso l'integrazione dei profili di espressione di microRNA e mRNA.

In questo lavoro di tesi, abbiamo proposto e confrontato diversi approcci integrativi, alcuni mai impiegati in questo contesto, cercando delle soluzioni interessanti per il raffinamento delle predizioni dei target e introducendo anche metodologie che esprimessero la natura molti-a-molti del meccanismo di appaiamento. I campioni biologici utilizzati sono riferiti a 76 pazienti al primo stadio del Cancro Ovarico (EOC).

L'indagine condotta ha consentito di giungere a diverse conclusioni interessanti sulla forza-legame di queste molecole, che possono essere utilizzate in futuri lavori come punto di partenza per il raffinamento dei metodi attuali e la proposta di nuovi algoritmi. In particolare, i punti chiave della nostra indagine possono essere riassunti come segue:

MODELLI MULTIVARIATI: la natura molti-a-molti della relazione regolatori tra miRNA e mRNA può essere efficaciemente colta attraverso l'utilizzo di modelli di regressione multivariati in grado di gestire l'elevata dimensionalità delle variabili in gioco, ad esempio attraverso una regressione penalizzata come il Lasso. La considerazione concomitante di più microRNA in gioco apporta ad un netto miglioramento della specifictà delle predizioni basate sulla complementarità di sequenza rispetto a metodi univariati.

Va precisato però, che questa buona proprietà, è fortemente dipendente sia alla scelta del sottoinsieme inziale di variabili introdotte nella stima, sia alla scelta del parametro di penalizzazazione. Per quest'ultimo, in particolare si è visto che la scelta ottimale non dev'essere fatta sulla base di misure che coinvolgono l'accuratezza delle previsioni del modello.

RISC: nonostante le proteine Argonaute rappresentino le molecole effettrici del meccanismo di riconoscimento delle specifiche sequenze complementari tra mRNA e microRNA, la loro informazione, inserita nel modello di stima come fattore moltiplicativo dei profili di espressione, non apporta nessun reale miglioramento in termini di specificità, quando il modello utilizzato è robusto.

FORMA DELLA RELAZIONE: anche se il modello multivariato, quando rispetta le proprietà sovra-citate, si dimostra migliore rispetto a quelli univariati, la frazione ancora elevata di falsi positivi dimostra che la regolazione molecolare dei microRNA rispetto ai suoi geni target risulta ancora troppo semplificata nelle nostre formulazioni matematiche.

In particolare i nostri dati suggeriscono che il legame tra microRNA e mRNA può essere ben catturato sia da misure che esprimono una relazione lineare, univariata (Correlazione Parziale di Pearson) o mul-

tivariata (TaLasso), che da quelle che esprino altre forme di relazioni (Indice di Correlazione di Gini).

DIREZIONE DELLA RELAZIONE: nonostante recenti studi abbiano indicato che i miRNA possono anche mediare la traduzione attivandola, per migliorare la specificità delle predizioni in termini di validati, è necessario considerare esclusivamente l'effetto di inbizione attravareso relazioni inverse tra profili di mRNA e miRNA. I metodi che non sono in grado di distinguere tra forza-legame postiva e negativi risultano totalmente inadeguati nell'identificazione dei geni bersaglio (Mutua Informazione).

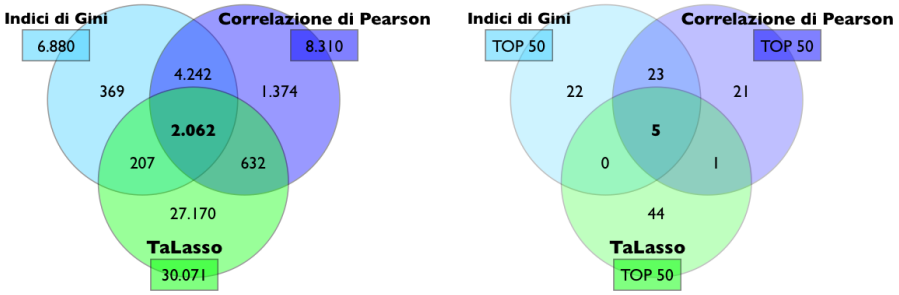
Il nostro studio inoltre, attraverso l'utilizzo dei migliori approcci di integrazione individuati per i dati dell'EOC, ha anche definito una *signature* di miRNA coinvolti nella tumorigenesi che potrebbero risultare utili nell'identificazione di bersagli terapeutici.

Come conclusione riportiamo dunque i risultati ottenuti in questo senso e una breve discussione sul ruolo biologico dei microRNA e degli mRNA definiti come possibili marcatori del Cancro Ovarico al primo stadio.

Liste di interazioni Il sottoinsieme finale di accoppiamenti interessanti è stato ottenuto attraverso l'intersezione delle liste restituite dal modello di regressione penalizzata TaLasso, dalla Correlazione di Pearsone e dall'Inidice di Gini. Visto il sostanziale miglioramento ottenuto attraverso l'utilizzo delle predizioni iniziali dell'algoritmo TargetScan, sono state considerate solo le classifiche ad esso associate.

In Tabella 7, Tabella 8 e Tabella 9 abbiamo riportato le liste complete, tagliate alla cinquantesima posizione; mentre in Figura 32 abbiamo rappresentato i diagrammi di Venn, per evidenziare le interazioni comuni a tutte le liste sull'intera graduatoria e sui Top^{5}00.

Figura 32: Diagrammi di Venn, su diverse porzioni delle liste ottenute da TL^{TS}(verde), R^{TS}(blu) e GC^{TS}(azzurro)



(a) Liste complete (b) Top^{50}

Confrontrando tra loro le liste restituite dai due metodi pair-wise notiamo che buona parte delle associazioni significative finali sono comuni, ottenendo anche coefficienti molto simili tra loro: il 91.63% delle interazioni contenute in R^{TS} sono anche anche contenute in GC^{TS}, e 28 di queste sono comuni considerando le prime 50 posizioni. Questi risultati fanno dunque pensare che la relazione che lega il microRNA al suo mRNA target sulla base della sua espressione sia effettivamente di tipo lineare.

Le intersezioni comuni a tutti e tre gli insiemi invece sono pari 2.062 su un totale di 36.056, pari rispettivamente al 24,81%, 29,97% e 6,68% sul totale di interazioni predette da Correlazione, Gini e TaLasso.

Considerando un taglio finale pari a 50 otteniamo un insieme di 5 accoppiamenti mRNA-miRNA che consideriamo il nostro pannello di accoppiamenti interessanti finale (Tabella 6).

Significato biologico L'indagine sui profili di espressione, di pazienti con Cancro Ovarico di stadio I, ha individuato una *signature* finale di due microRNA che dimostrano un forte legame inverso con cinque differenti mRNA.

Tabella 6: Sottoinsieme finale restituito dall'intersezione delle liste TL^{TS}, GC^{TS} e R^{TS}, considerando un taglio variabile da 10 a 45.

k	mRNA	microRNA	Validato
10	KLF4	hsa-miR-200c	T
26	MAP3K5	hsa-miR-20b	F
40	KIAA0513	hsa-miR-20b	F
42	TIMP2	hsa-miR-20b	F
44	ZEB1	hsa-miR-200c	T

A conferma della validità scientifica della nostra analisi, e della scelta dei modelli finali individuati, due su cinque delle interazioni predette sono state precedentemente individuate in esperimenti validati in vitro e coinvolgono molecole notoriamente attive nei tumori umani.

Tra questi, la famiglia dei *miR-200* si è rivelata essere implicata nella crescita e nello sviluppo metastatisco di numerosi tumori[69]: in particolare, si è osservato che il *miR-200c*, attraverso la deregolazione del suo gene target ZEB1, portava ad aumentare l'invasività delle cellule del cancro al seno e induceva un processo matastiaico[68].

Altri studi invece hanno suggerito un ruolo del *miR-200c* proprio nel Cancro Ovarico di primo stadio[70]: questa molecole è stata associata sia alla sopravvivenza complessiva, che alla recidività dei pazienti in seguito ad intervento di rimozione del tessuto neoplastico. Infine hanno dimostrato il loro potenziale anche come strumento diagnostico nella capacità di predire la risposta clinica alle cure chemioterapiche[66].

Di notevole interesse, nel nostro caso, è suo target KLF4: un fattore di trascrizione che è risultato funzionare come un oncogene sopressore del tuomore, nel cancro mammario. Questo mRNA sembra essere responsabile

della diminuzione della proliferazione e dell'aumento dell'apoptosi, rivelandosi un mediatore chiave nella progressione delle matastasi[67].

Le prosepettive future di questa analisi sono dunque quelle di un ulteriore caratterizzazione *in vitro* di questo appaiamento individuato, al fine di delucidare i meccanismi molecolari coinvolti nel processo di trasformazione maligna. Suggeriamo pertanto che questa nuovo target di *miR-200c* vada approfonditamente analizzato come possibile marcatore diagnstico ed eventuale bersaglio per una innovativa terapia.

Tabella 7: Prime 50 posizioni degli accoppiamenti miRNA-mRNA risultanti dall'integrazione di Target Scan attraverso il modelllo TALSSO. In verde sono sottolineate le iterazioni validate, mentre in grassetto quelle comuni alle tre liste finali scelte.

k	mRNA	microRNA	Score	Validato
1	PNOC	hsa-miR-34a	0.506	F
2	RASD1	hsa-miR-20a	0.358	F
3	**KLF4**	**hsa-miR-200c**	**0.329**	**T**
4	DACT1	hsa-miR-200c	0.315	F
5	BAMBI	hsa-miR-20a	0.306	T
6	FOXL2	hsa-miR-93	0.304	F
7	LHFP	hsa-miR-200c	0.287	T
8	FN1	hsa-miR-200c	0.282	T
9	**KIAA0513**	**hsa-miR-20b**	**0.277**	**F**
10	FAP	hsa-miR-30c	0.27	F
11	C16orf45	hsa-miR-203	0.264	F
12	PDE5A	hsa-miR-19a	0.262	F
13	CNN1	hsa-miR-335	0.254	F
14	NPTX1	hsa-miR-130a	0.253	T
15	CLDN11	hsa-miR-25	0.251	F
16	CDH11	hsa-miR-200c	0.25	F
17	COL15A1	hsa-miR-29b	0.248	F
18	JPH4	hsa-miR-203	0.247	F
19	F3	hsa-miR-20b	0.245	F
20	**ZEB1**	**hsa-miR-200c**	**0.244**	**T**
21	RASL11B	hsa-miR-20a	0.233	F
22	CXCL12	hsa-miR-141	0.23	F
23	CACNA1H	hsa-miR-25	0.227	F
24	EPHA4	hsa-let-7g	0.219	F
25	FOXL2	hsa-miR-23a	0.219	F
26	**MAP3K5**	**hsa-miR-20b**	**0.212**	**F**
27	KLF2	hsa-miR-25	0.21	F
28	PAX8	hsa-miR-134	0.208	F
29	GPR124	hsa-miR-96	0.207	F
30	RASGEF1A	hsa-miR-125b	0.204	F
31	CAMK2N1	hsa-miR-20a	0.202	F
32	HS3ST1	hsa-miR-200c	0.202	F
33	**TIMP2**	**hsa-miR-20b**	**0.199**	**F**
34	HOXA5	hsa-miR-143	0.198	F
35	PDGFRB	hsa-miR-29b	0.194	F
36	DKK3	hsa-miR-19a	0.194	F
37	MFAP2	hsa-miR-29b	0.193	F
38	PDGFRA	hsa-miR-96	0.19	F
39	RASD1	hsa-miR-130a	0.19	F
40	PDGFRA	hsa-miR-181d	0.19	F
41	SPARC	hsa-miR-203	0.189	T
42	ANTXR2	hsa-miR-96	0.188	F
43	FXYD2	hsa-miR-205	0.186	F
44	SMPD3	hsa-miR-506	0.185	F
45	RERG	hsa-miR-203	0.184	F
46	ADAMTS9	hsa-miR-29c	0.183	F
47	VCAN	hsa-miR-101	0.183	F
48	EDNRA	hsa-miR-200c	0.182	F
49	DUSP5	hsa-miR-25	0.181	F
50	PRRX1	hsa-miR-20a	0.181	F

Tabella 8: Prime 50 posizioni degli accoppiamenti miRNA-mRNA risultanti dall'integrazione di Target Scan attraverso il modelllo relativo al coefficiente di Correlazione di GINI. In verde sono sottolineate le iterazioni validate, mentre in grassetto quelle comuni alle tre liste finali scelte.

k	mRNA	microRNA	Score	Validato
1	PEAK1	hsa-miR-200c	-0.653	F
2	SH3PXD2A	hsa-miR-200c	-0.614	F
3	**KIAA0513**	**hsa-miR-20b**	**-0.582**	**F**
4	TIMM17B	hsa-let-7b	-0.58	F
5	PALLD	hsa-miR-92a	-0.571	F
6	AKAP13	hsa-miR-106b	-0.557	F
7	LHFPL2	hsa-miR-30e	-0.55	F
8	RAB22A	hsa-miR-21	-0.549	F
9	MAPRE2	hsa-miR-141	-0.549	F
10	**KLF4**	**hsa-miR-200c**	**-0.547**	**T**
11	DOCK5	hsa-miR-19b	-0.545	F
12	**MAP3K5**	**hsa-miR-20b**	**-0.542**	**F**
13	PEAK1	hsa-miR-30c	-0.541	F
14	LRRC8A	hsa-miR-200c	-0.54	F
15	PEAK1	hsa-miR-141	-0.539	F
16	UBFD1	hsa-let-7c	-0.538	F
17	MGLL	hsa-miR-93	-0.536	F
18	FOXP1	hsa-miR-141	-0.531	F
19	SLC12A6	hsa-miR-30d	-0.531	F
20	BEND3	hsa-miR-199a-5p	-0.53	F
21	MAP3K3	hsa-miR-106b	-0.53	F
22	LMOD1	hsa-miR-96	-0.525	F
23	LARP4	hsa-miR-199a-5p	-0.525	F
24	E2F2	hsa-miR-125b	-0.523	F
25	BCL2L11	hsa-miR-222	-0.52	T
26	SYNE1	hsa-miR-93	-0.519	F
27	CSGALNACT1	hsa-miR-106b	-0.519	F
28	PRRX1	hsa-miR-106b	-0.518	F
29	MYCL1	hsa-let-7b	-0.518	F
30	NRP2	hsa-miR-141	-0.517	F
31	**TIMP2**	**hsa-miR-20b**	**-0.514**	**F**
32	MARCKS	hsa-miR-200c	-0.513	T
33	TIGD5	hsa-miR-199a-5p	-0.513	F
34	TIMP2	hsa-miR-93	-0.512	F
35	SOX17	hsa-miR-194	-0.512	F
36	CSGALNACT1	hsa-miR-20a	-0.511	F
37	KIAA0513	hsa-miR-93	-0.511	F
38	APP	hsa-miR-20a	-0.511	T
39	PCNX	hsa-miR-200c	-0.51	F
40	NDST1	hsa-miR-200c	-0.51	F
41	ZNF469	hsa-miR-19a	-0.509	F
42	TIMP2	hsa-miR-425	-0.509	F
43	TSHZ3	hsa-miR-93	-0.508	F
44	**ZEB1**	**hsa-miR-200c**	**-0.507**	**T**
45	NFIB	hsa-miR-30a	-0.506	F
46	YOD1	hsa-miR-15b	-0.506	F
47	PDGFRA	hsa-miR-106b	-0.506	F
48	WASF3	hsa-miR-146a	-0.506	F
49	DUSP3	hsa-miR-141	-0.505	F
50	CSGALNACT1	hsa-miR-17	-0.505	F

Tabella 9: Prime 50 posizioni degli accoppiamenti miRNA-mRNA risultanti dall'integrazione di Target Scan attraverso il modelllo relativo al coefficiente di Correlazione di PEARSON. In verde sono sottolineate le iterazioni validate, mentre in grassetto quelle comuni alle tre liste finali scelte.

k	mRNA	microRNA	Score	Validato
1	PEAK1	hsa-miR-200c	-0.638	F
2	SH3PXD2A	hsa-miR-200c	-0.593	F
3	LRRC8A	hsa-miR-200c	-0.573	F
4	NFIB	hsa-miR-30a	-0.565	F
5	PALLD	hsa-miR-92a	-0.552	F
6	PEAK1	hsa-miR-141	-0.543	F
7	**KLF4**	**hsa-miR-200c**	**-0.541**	**T**
8	MAPRE2	hsa-miR-141	-0.538	F
9	AKAP13	hsa-miR-106b	-0.536	F
10	LHFPL2	hsa-miR-30e	-0.531	F
11	MGLL	hsa-miR-93	-0.522	F
12	FOXP1	hsa-miR-141	-0.522	F
13	LRRC8A	hsa-miR-141	-0.522	F
14	DOCK5	hsa-miR-19b	-0.513	F
15	TIMM17B	hsa-let-7b	-0.507	F
16	ATP8A1	hsa-miR-30a	-0.507	F
17	NDST1	hsa-miR-200c	-0.506	F
18	**MAP3K5**	**hsa-miR-20b**	**-0.506**	**F**
19	C16orf45	hsa-miR-203	-0.504	F
20	VASH1	hsa-miR-200c	-0.502	F
21	MFSD6	hsa-miR-30a	-0.501	F
22	CREB5	hsa-miR-29c	-0.499	F
23	SOX17	hsa-miR-194	-0.499	F
24	MAP3K3	hsa-miR-106b	-0.498	F
25	LRP1	hsa-miR-200c	-0.497	F
26	PEAK1	hsa-miR-30c	-0.497	F
27	**ZEB1**	**hsa-miR-200c**	**-0.495**	**T**
28	ARHGEF7	hsa-miR-200c	-0.495	F
29	**TIMP2**	**hsa-miR-20b**	**-0.493**	**F**
30	AKAP13	hsa-miR-93	-0.492	F
31	SH3PXD2A	hsa-miR-301a	-0.491	F
32	LMOD1	hsa-miR-96	-0.491	F
33	TARBP2	hsa-miR-497	-0.491	F
34	TIGD5	hsa-miR-199a-5p	-0.491	F
35	SH3PXD2A	hsa-miR-106b	-0.49	F
36	NRP2	hsa-miR-141	-0.488	F
37	C1orf21	hsa-miR-141	-0.487	F
38	APBB2	hsa-miR-141	-0.486	F
39	CD40	hsa-miR-145	-0.486	F
40	**KIAA0513**	**hsa-miR-20b**	**-0.484**	**F**
41	CHAC1	hsa-miR-214	-0.484	F
42	TIMP2	hsa-miR-20b	-0.483	F
43	CLIP2	hsa-miR-200c	-0.483	F
44	SLC12A6	hsa-miR-30d	-0.482	F
45	CERS6	hsa-miR-30a	-0.481	F
46	FOXP1	hsa-miR-200c	-0.481	F
47	IVNS1ABP	hsa-miR-19b	-0.481	F
48	YOD1	hsa-miR-15b	-0.48	F
49	CD276	hsa-miR-29b	-0.479	T
50	ZCCHC24	hsa-miR-200c	-0.478	F

A Algoritmi

.

.

.

Algoritmo 1 KNN - Metodo k-nearest neighbors

Input :

$X_{(N \times p)}$: matrice originale dei dati

k: numero di vicini scelti

1: Per ciascuna riga della matrice X $\{i$: da 1 a $N\}$:

1.1 : se la riga i-esima contiene tutti i p valori viene salvata in X^c. Si ottiene una nuova matrice X^c di dimensione $_{N_c \times p}$

2: Calcolo n, il numero totale di valori mancanti

3: Per ogni valore mancante in X di coordinate $[i, y]$ $\{z$: da 1 a $n\}$

3.1 : Per ogni riga da X_c $\{j$: da 1 a $N_c\}$

3.1.1 : Calcolo D_e: la distanza euclidea tra la riga j-esima di X_c e la riga i-esima di X

3.2 : Trovo le k righe di X_c a cui corrispondono i piú piccoli valori di D_e

3.3 : Prendo i valori in corrispondenza della colonna y delle k righe, e ne faccio la media pesata per i valori di D_e corrispondenti

3.4 : Sostituisco a $X[i, y]$ il valore ottenuto al passo precedente.

Algoritmo 2 KNN - Scelta del parametro k

Input :

$X_{(N\times p)}$: matrice originale dei dati

$L_k(1 \times n_k)$: vettore che contiene gli n_k possibili valori del parametro k

1: Per ciascuna riga della matrice X $\{i$: da 1 a $N\}$:

1.1 : se la riga i-esima contiene tutti i p valori viene salvata in X^c. Si ottiene una nuova matrice X^c di dimensione $_{N_c\times p}$

2: Estraggo casualmente N_c numeri da 1 a N

3: Creo una nuova matrice X^* prendendo le righe relative agli indici estratti al punto precedente. Si ottiene una nuova matrice X^* di dimensioni $_{N_c\times p}$.

4: Creo una matrice X^{c*} identica ad X^c

5: Creo un vettore Err di dimensione $(1\times n_k)$ che conterrà la somma degli errori assoluti relativi a ciascun k

6: Per ogni riga si X^* i: da 1 a N_c

6.1 : Per ogni colonna di X^* j: da 1 a p

6.1.1 : Se $X^*[i,j]$=NA, allora forzo X^{c*} ad NA e salvo le coordinate $[i,j]$ nella lista E

7: Per ogni valore di L_k i: da 1 a n_k

7.1 : Inizializzo la matrice $X^{c*}_{na} = X^{c*}$

7.2 : Imputo i valori mancanti tramite l'Algoritmo **??**, fornendo come matrice originale X^{c*}_{na} e il k i-esimo. Ottengo una matrice X^{c*}_{na} senza valori mancanti.

7.3 : Ottengo $Err[i]$, calcolando $|X^{c*} - X^{c*}_{na}|$ e sommando tutte quelle celle cui coordinate sono contenute nella lista E.

8: Trovo il minimo valore contenuto in Err, e scelgo il k corrispondente del vettore L_k.

Algoritmo 3 stima attraverso la funzione LARS

Input :

$Y_{(N^c \times 1)}$: vettore contenente i valori della variabile risposta

$Y_{(N^c \times N^m)}$: matrice contenente i valori degli N^m regressori, negli N^c campioni

1: Inizializzo λ con un valore prossimo a +Inf.

2: Inizializzo Step a zero.

3: Inizializzo $Scarto_{[1x(1+N^m)]}$ a zero. Il vettore conterrà la somma degli scarti a quadrato del modello $Lasso_\lambda$ stimato.

4: Inizializzo a zero la matrice *Mbeta*, che contiene i beta stimati per ciascun sottoinsieme di regressori scelto. STATE Stimo $Lasso_\lambda$: regressione Lasso con il rispettivo valore di λ.

5: Salvo in $B_{[1xN_m]}$ i valori di beta stimati. Allo step zero otterremo il modello con la sola intercetta. Salvo in E la somma dei residui al quadrato del modello.

6: Salvo B in $M[step,]$.

7: Salvo E in $Scarto[step]$.

8: Finchè $\lambda > 0$ o non ho ottenuto N^m modelli:

8.1 : $\lambda = \lambda$ - ϵ

8.2 : Stimo $Lasso_\lambda$, salvo in B i valori di beta stimati e salvo in E la somma dei residui al quadrato.

8.3 : Calcolo N_b: numero di elementi di B diversi da zero.

8.3.1 : Se $N_b = step$:

8.3.2 : Salvo B in $M[step,]$

8.3.3 : Salvo E in $Scarto[step]$

8.3.4 : Incremento *step* di 1.

9: Calcolo $Sigma^2$: la varianza stimata con il modello lineare di regressione multivariato.

10: Per ogni modello stimato (k: 0 a Nm)

10.1 : Calcolo $C_p[k]$ come $Scarto[k]/Sigma^2$ - $N^c + 2 * k$

11: Restituisco B e C_p.

Algoritmo 4 stima attraverso la funzione LARS

Input :

A: algoritmo di predizione basato sulla complementarietà di sequenze

$X i_{[N^c \times (1+N_i^m))]}$: matrice contenente i valori degli N^m regressori, negli N^c campioni

N: numero di mRNA per cui sono stati trovati dei miRNA associati, attraverso A

$a^2_{[N^c \times 1]}$: vettore dei valori di espressione relativo al gene Ago_2 negli N^c campioni

$a^{134}_{[N^c \times 1]}$: vettore dei valori di espressione risultante da una qualche combinazione dei profili dei geni Ago_1, Ago_3 e Ago_4

1: Per ciascuna matrice X_i costruita (i: da 1 a N):

1.1 : Costruire una nuova matrice X_i^* con i valori di espressione relativi al mRNA i-esimo contenuti nella matrice X_i

1.2 : Prendere le N_i^m colonne dei valori di espressione di X_i, moltiplicarle per a^2 e inserirle nella nuova matrice X_i^*

1.3 : Prendere le N_i^m colonne dei valori di espressione di X_i, moltiplicarle per a^{134} e inserirle nella nuova matrice X_i^*

1.4 : La matrice X_i^* finale conterrà $1+(N_i^m \times 2)$ colonne

2: Per ciascuna matrice X_i^* costruita (i: da 1 a N):

2.1 : Inizializzo G_1 con gli indici di tutte le $(N_i^m \times 2)$ colonne relative ai valori dei miRNA associati, e G_2 come insieme vuoto

2.2 : Inizializzo come insieme vuoto il vettore L_b

2.3 : Finche G_1 non contiene più indici (o non trovo più variabili significative in G_1, vedi punto 2.3.4.)

2.3.1 : Stimo il Lasso utilizzando come variabile risposta la colonna relativa ai valori di espressione dell'mRNA contenua in X_i^*, e come regressori le restanti colonne.

2.3.2 : Assegno a N^g1 il numero di variabili contenute in G_1 per questa iterazione.

2.3.3 : Se $N^g1 < N^c$: (altrimenti prendo il miRNA più correlato e salvo in IND il suo indice e in B il relativo β, e vado al punto 2.3.4)

2.3.3.1 : Trovo il valore che minimizza la statistica C_p ottenuta con la funzione Lasso, e salvo gli indici diversi da 0 nel vettore IND, e i relativi β nel vettore B.

2.3.4 : Se il vettore IND è di lunghezza non nulla: (altrimenti esco dal ciclo e proseguo al punto 2.4)

2.3.4.1 : Elimino in G_1 gli indici contenuti in IND

2.3.4.2 : Aggiungo in G_2 gli indici contenuti in IND a cui corrispondono valori di B strettamente minori di 0, e i rispettivi β li aggiungo alla lista L_b

2.4 : Tengo solo gli indici in G_2 che si riferiscono agli N_i^m valori moltiplicati per a_2

2.5 : Cambio di segno e Normalizzo i coefficienti contenuti in L_b. L_b conterrebbe solo valori positivi, più grande è il valore e più l'associazione è forte.

2.6 : Ordino in maniera decrescente i valori contenuti in L_b, e i rispettivi indici contenuti in G_2.

2.7 : A N^{g2} assegno il numero di elementi salvati nel vettore G_2.

2.8 : Creo un vettore S di grandezza N^{g2}

2.9 : Per ogni elemento contenuto in G_2 (j: da 1 a N^{g2})

2.9.1 : Assegno a $S[j]$ uno score calcolato in funzione della posizione secondo la formula $(N^{g2} - j + 1)/(N^{g2}) * 100$. Più il β cambiato di segno sarà grande più il miRNA relativo sarà associato al mRNA, e più lo score sarà vicino a 100.

3: Restituisco G_2, L_b e S.

B Codice R

B.1 Validadazione

Codice 1: *ValidaLista()*

```
ValidaLista<-function(Associazioni,listaValidati,nome.salvataggio="Associazione.txt",
opzioneGene=1,opzioneOrdina1=list(attivo=FALSE,colonna=0,metodo=0),
opzioneOrdina2=list(attivo=FALSE,colonna=0,metodo=0),
opzioneLasso=FALSE,nome.salvataggio.Lasso="")
{
        ## ORDINO LA LISTA ASSOCIAZIONI SULLA BASE DELLO SCORE
        # primo criterio di ordine
        if(opzioneOrdina1$attivo & opzioneOrdina2$attivo)
        {
                ordina<-order(opzioneOrdina1$metodo*Associazioni[,opzioneOrdina1$colonna],
                opzioneOrdina2$metodo
                *Associazioni[,opzioneOrdina2$colonna])
                Associazioni<-Associazioni[ordina,]
        }
        # secondo criterio di ordine
        if(opzioneOrdina1$attivo & !opzioneOrdina2$attivo)
        {
                ordina<-order(opzioneOrdina1$metodo*Associazioni[,opzioneOrdina1$colonna])
                Associazioni<-Associazioni[ordina,]
        }

        ## VALIDO LE ASSOCIAZIONI
        contatore<-1:nrow(Associazioni)
        n<-nrow(Associazioni)
        Geni<-switch(opzioneGene,paste(listaValidati[,3]),as.numeric(listaValidati[,4]),
        paste(listaValidati[,5]))
        Mirn<-paste(listaValidati[,2])
        Id<-paste(listaValidati[,1])

        print("---INIZIO␣VALIDAZIONE---")
        inizio<-proc.time()
        RisValida<-apply(cbind(Associazioni,contatore),1,valida,Geni,Mirn,Id,n,opzioneGene)
        fine<-proc.time()
        print("---␣FINE␣VALIDAZIONE␣---")
        print(paste("Tempo␣di␣validazione:␣",(fine-inizio)[3],sep=""))

        validated<-as.logical(RisValida[1,])
        idValida<-as.numeric(RisValida[2,])

        ## NUOVA LISTA
        # Variabili originali
        # validated: TRUE/FALSE
        # Id (mio)
        # TarBase(presente/assente)
        # miRecord(presente/assente)
        # mirTarBase(presente/assente)
        # mirWalk(presente/assente)
        Nuovo<-cbind(Associazioni,validated,idValida,listaValidati[idValida,6],listaValidati
```

```
49          [idValida,7],listaValidati[idValida,8],listaValidati[idValida,9])
50          colnames(Nuovo)<-c(colnames(Associazioni),"validated","Id","TarBase","miRecord",
51          "mirTarBase","mirWalk")
52          write.table(Nuovo,file=nome.salvataggio,col.names=TRUE,row.names=FALSE)
53          print(paste("File salvato con successo in: ",getwd(),"/",nome.salvataggio,sep=""))
54          print("SUMMARY operazione:")
55          print(paste("File: ",nome.salvataggio,sep=""))
56          print(paste("Totale Associazioni: ",nrow(Associazioni),sep=""))
57          print(paste("Totali validate: ",sum(validated),sep=""))
58          if(nrow(Associazioni)>=500)
59          {
60                  print(paste("Totali validati TOP500: ",sum(validated[1:500]),sep=""))
61                  if(nrow(Associazioni)>=1000)
62                  {
63                          print(paste("Totali validate TOP1000: ",sum(validated[1:1000]),sep=""))
64                  }
65          }
66
67          if(opzioneLasso)
68          {
69                  overlap.dat<-Associazioni[validated,]
70                  save(overlap.dat,file=paste("overlap",nome.salvataggio.Lasso,".Rdata",sep=""))
71                  print(paste("File -overlap",nome.salvataggio.Lasso,".Rdata- salvato!",sep=""))
72          }
73  }
74
75  valida<-function(riga,listaG,listaM,Id,tot,opzioneGene)
76  {
77          i<-as.numeric(riga[[length(riga)]])
78          if((i %% 2000)==0)
79          {
80                  print(paste("Valida: ",i,"/",tot,sep=""))
81          }
82
83          ident<-""
84          G<-switch(opzioneGene,paste(riga[[1]]),as.numeric(riga[[1]]),paste(riga[[1]]))
85          M<-paste(riga[[2]])
86
87          lG<-which(G==listaG,arr.ind=TRUE)
88          lM<-which(M==listaM,arr.ind=TRUE)
89
90          if(!length(lM)==0 & !length(lG)==0)
91          {
92                  l<-intersect(lM,lG)
93                  if(!length(l)==0)
94                  {
95                          risp<-"TRUE"
96                          ident<-paste(Id[l])
97                  } else { risp<-"FALSE" }
98          } else { risp<-"FALSE" }
99          out<-matrix(c(risp,ident),nrow=2,ncol=1)
100         out
101
102 }
```

B.2 ArgoLasso

Codice 2: *ArgoLasso()*

```
ArgoLasso<-function(data.dir,data.dir.salvataggio,MRun=TRUE,AGO=TRUE,max.run=1000)
{
        library(lars)
                # Inizializzazione variabili
        count<-1
        inizio<-proc.time()
        select.rna <- c()
        select.mir <- c()
        select.coeff <- c()
        select.score <- c()
        n<-length(dir(data.dir))
        print(paste("Geni da analizzare:",n,sep=" "))

        ## Inizio ad analizzare i dati contenuti nella directory data.dir
        for (filename in dir(data.dir))
        {
                if((count %% 1000)==0)
                { print(paste("Analizzo:",count,"/",n)) }
                print(filename)

        # Carico il file
                filename = paste(data.dir,filename,sep="/")
                dfr<-read.table(filename,header=TRUE,check.name=FALSE)

           if (ncol(dfr) > 1)
           {
                        y <- as.matrix(dfr[1])
                        x <- as.matrix(dfr[2:ncol(dfr)])

                # Inizializzo i due gruppi

                        ## (INIZIO) della Regressione MultiRunLasso
                        risultato<-MultiRun(MRun,AGO,y,x,max.run)
                        groupii<-risultato$mir
                        groupiicoeffs<-risultato$coeff
                        ## (FINE) della Regressione MultiRun Lasso

                        ## Creo la Lista di associazioni per il Gene che sto analizzando e
                        ## associo uno score di ranking
                        if (length(groupii) > 0)
                        {
                        # Inizializzo le variabili per il rank
                        rank.mir <- c()
                                rank.coeff <- c()
                                rank.rank <- 1
                                for (i in seq(length(groupii)))
                        {
                                        mir <- groupii[i]
                                        coeff <- groupiicoeffs[i]
                                        if(AGO)
                                        {
                                                # Se l'opzione AGO e settata a TRUE salvo in
```

```
53                                              # G2 solo i miRNA relativi ad AGO2
54                                              if (grepl("AGO2",mir))
55                                              {
56                                                      # Formattazione
57                                              mir <- gsub("\\-AGO2","",mir)
58                                              mir <- gsub("\\.$","\\*",mir)
59                                                      mir <- gsub("\\.","-",mir)
60                                              rank.mir <- append(rank.mir,mir)
61                                              rank.coeff <- append(rank.coeff,coeff)
62                                              }
63                                      }
64                                      else
65                                      {
66                                              # Se l'opzione AGO e settata a False salvo
67                                              # tutte le variabili presenti in G2
68                                      rank.mir <- append(rank.mir,mir)
69                                      rank.coeff <- append(rank.coeff,coeff)
70                                      }
71                              }
72
73                              # Normalizzo i coefficienti selezionati
74                              rank.coeff <- abs(as.numeric(rank.coeff))
75                              rank.coeff <- rank.coeff/sqrt(sum(rank.coeff^2))
76
77                              # Asegno lo score per il Ranking a ciascuna associazione
78                              for (i in order(rank.coeff))
79                              {
80                                      select.rna <- append(select.rna,colnames(y))
81                                      select.mir <- append(select.mir,rank.mir[i])
82                                      select.coeff <- append(select.coeff,-rank.coeff[i])
83                      select.score <- append(select.score,round((rank.rank/length(rank.mir))*100))
84                                      rank.rank <- rank.rank + 1
85                              }
86
87                              } ## Ho finito di creare la lista di associazioni per il gene
88              }
89              # Incremento il numero di geni analizzati
90              count<-count+1
91              } ## chiudo il for che analizza tutti i dati
92
93              fine<-proc.time()
94              print(paste("Tempo totale per il dataset ",data.dir," ",(fine-inizio)[3],sep=""))
95
96              select.dat <- data.frame(select.rna,select.mir,select.coeff,select.score)
97              if(MRun)
98              { stringa2<-"Multi" }
99              else { stringa2<-"One" }
100 save(select.dat,file=paste(data.dir.salvataggio,"/select",stringa2,data.dir,".Rdata",sep=""))
101
102             print("FINE")
103             print("Summary:")
104             print(paste("Directory dati: ",data.dir,sep=""))
105             print(paste("Directory salvataggio: ",data.dir.salvataggio,sep=""))
106
107             stringa1<-"Regressione "
108             if(MRun)
```

```
109         {
110                 stringa1<-paste(stringa1,"Multirun␣Lasso␣")
111         } else { stringa1<-paste(stringa1,"OneRun␣Lasso␣") }
112         if(AGO)
113         {
114                 stringa1<-paste(stringa1,"con␣AGO␣")
115         }else { stringa1<-paste(stringa1,"senza␣AGO␣") }
116         print(paste("Opzioni:␣",stringa1,sep=""))
117
118         print(paste("numero␣associazioni␣trovate:␣",nrow(select.dat),sep=""))
119
120 }
121
122 MultiRun<-function(MRun,AGO,y,x,max.run)
123 {
124         library(lars)
125
126         groupii <- c()
127         groupiicoeffs <- c()
128
129         run<-0
130         ## (INIZIO) della Regressione MultiRunLasso
131         while (TRUE)
132         {
133         # Se G1 e vuoto esco dal MultiRun Lasso
134                 if (ncol(x) == 0) { break }
135         M <- lars(x,y,type="lasso",normalize=TRUE,intercept=TRUE,use.Gram=TRUE)
136
137                 ## CASO (1): Cp nullo  (nx > ncampioni)
138                 # -> scelgo la x maggiormente correlata
139                 if (is.nan(M$Cp[1]))
140                 {
141                         # Cerco lo step in cui la prima x entra nel modello
142                         for (step.i in c(1:1:length(M$Cp)))
143                         {
144                             if (length(which(M$beta[step.i,]!=0)) != 0)
145                         {
146                                         m.beta <- as.data.frame(M$beta)
147                                         all.names <- colnames(m.beta)
148                                         minCp.line <- m.beta[step.i,]
149                                         nzcnames <- all.names[which(minCp.line!=0)]
150                                         ngtnames <- all.names[which(minCp.line<0)]
151                                         # Salvo i coefficienti < 0
152                                         if (identical(ngtnames,character(0)) == FALSE)
153                                         {
154                                         groupii <- append(groupii,ngtnames)
155                                     ngtcoeffs <- M$beta[step.i,][(colnames(x)%in%ngtnames)]
156                                         groupiicoeffs <- append(groupiicoeffs,ngtcoeffs)
157                                         }
158                                         # Rimuovo i coefficienti <> 0 per la prossima Run
159                                         if (identical(nzcnames,character(0)) == FALSE)
160                                         {
161                                                 x <- x[,!(colnames(x)%in%nzcnames),drop=FALSE]
162                                                 break
163                                         }
164                         }
```

```
                }
        } # chiudo (1)
            ## CASO (2): Cp non nullo
        else
        {
                minCpIndex <- as.numeric(which.min(M$Cp))

                # se il min e' il primo o l'ultimo e il num dei Cp calcolati e >1
                if ((minCpIndex == 1 | minCpIndex == length(M$Cp)) & length(M$Cp) > 1)
                {
                for (minCpIndex in c(1:1:length(M$Cp)))
                {
                        #minCpIndex<-1
                                if (length(which(M$beta[minCpIndex,]!=0)) != 0)
                                {
                                        m.beta <- as.data.frame(M$beta)
                                        all.names <- colnames(m.beta)
                                        minCp.line <- m.beta[minCpIndex,]
                                        nzcnames <- all.names[which(minCp.line!=0)]
                                        ngtnames <- all.names[which(minCp.line<0)]
                                        # Salvo i coefficienti < 0
                                        if (identical(ngtnames,character(0)) == FALSE)
                                        {
                                        groupii <- append(groupii,ngtnames)
                                ngtcoeffs <- M$beta[minCpIndex,][(colnames(x)%in%ngtnames)]
                                        groupiicoeffs <- append(groupiicoeffs,ngtcoeffs)
                                        }
                                    # Rimuovo i coefficienti <> 0 per la prossima Run
                                    if (identical(nzcnames,character(0)) == FALSE)
                                    {
                                        x <- x[,!(colnames(x)%in%nzcnames),drop=FALSE]
                                                break # esce dal for
                                }
                                }
                }
                }
                else
                {
                        if (length(which(M$beta[minCpIndex,]!=0)) != 0)
                        {
                                m.beta <- as.data.frame(M$beta)
                                all.names <- colnames(m.beta)
                                minCp.line <- m.beta[minCpIndex,]
                                nzcnames <- all.names[which(minCp.line!=0)]
                                ngtnames <- all.names[which(minCp.line<0)]
                                # Salvo i coefficienti < 0
                                if (identical(ngtnames,character(0)) == FALSE)
                                {
                                        groupii <- append(groupii,ngtnames)
                         ngtcoeffs <- M$beta[minCpIndex,][(colnames(x)%in%ngtnames)]
                                groupiicoeffs <- append(groupiicoeffs,ngtcoeffs)
                                }
                                # Rimuovo i coefficienti <> 0 per la prossima Run
                                if (identical(nzcnames,character(0)) == FALSE)
                                {
                                x <- x[,!(colnames(x)%in%nzcnames),drop=FALSE]
```

```
                                        }
                        }
                        # Quando tutte le variabili hanno coeff==0 esco dal MultiRun Lasso
                        else { break }
                        }
            } # chiudo (2)
                # Se l'opzione MultiRun e settata a False esce dal MultiRun Lasso
                if (!MRun) { break }

                run <- run + 1
                if (run >= max.run) { break }
        } ## (FINE) della Regressione MultiRun Lasso
        # Restituisce groupii e groupiicoeff
        list(mir=groupii,coeff=groupiicoeffs)

}
```

Codice 3: *BootstrapArgoLasso()*

```
BootstrapArgoLasso<-function(data.dir,data.dir.salvataggio,MultRun=TRUE,AGO=TRUE,
repeat.times=5,max.run=1000)
{
        n<-length(dir(data.dir))
        print(paste("Geni da analizzare:",n,sep=" "))
        random.score <- c()

    for (l in seq(repeat.times))
    {
                print(paste("INIZIO ripetizione numero ",l,sep=""))
                inizio<-proc.time()
                # Initializing mRNA and miRNA list
                random.score.1 <- c()
                random.coeff.1 <- c()
                count<-1

                for (filename in dir(data.dir))
                {
                        if( (count %% 2000 )==0 )
                        { print(paste("(",l,") Analizzo: ",count,"/",n,sep="")) }
                        filename = paste(data.dir,filename,sep="/")
                dfr=read.table(filename,header=TRUE,check.names=FALSE)

                if (ncol(dfr) > 3)
                {
                                y <- as.matrix(dfr[1])
                                x <- as.matrix(dfr[2:ncol(dfr)])
                                # Creazione del miRNA random
                        x.random.name = sample(colnames(x),1,replace=FALSE)
                                x.random.1 <- sample(x[,x.random.name],nrow(y),replace=FALSE)
                                x <- cbind(x,x.random.1)

                                risultato<-MultiRun(MultRun,AGO,y,x,max.run)

                                groupii<-risultato$mir
                                groupiicoeffs<-risultato$coeff
                                if (length(groupii) > 0)
```

```
38                              {
39                              # Initializing rank data
40                              rank.mir <- c()
41                              rank.mir.2 <- c()
42                              rank.coeff <- c()
43                              rank.coeff.2 <- c()
44                              rank.score <- c()
45                              rank.rank <- 1
46                              for (i in seq(length(groupii)))
47                              {
48                                      mir <- groupii[i]
49                                      coeff <- groupiicoeffs[i]
50                                      if(AGO)
51                                      {
52                                      if (grepl("AGO2",mir) | grepl("random",mir))
53                                              {
54                                              # Formatting miRNA id
55                                              mir <- gsub("\\-AGO2","",mir)
56                                              mir <- gsub("\\.$","\\*",mir)
57                                              mir <- gsub("\\.","-",mir)
58
59                                              rank.mir <- append(rank.mir,mir)
60                                              rank.coeff <- append(rank.coeff,coeff)
61                                              }
62                                      }
63                                      else
64                                      {
65                                              if (grepl("random",mir))
66                                              {
67                                              # Formatting miRNA id
68                                              mir <- gsub("\\-AGO2","",mir)
69                                              mir <- gsub("\\.$","\\*",mir)
70                                              mir <- gsub("\\.","-",mir)
71                                              }
72
73                                              rank.mir <- append(rank.mir,mir)
74                                      rank.coeff <- append(rank.coeff,coeff)
75                                      }
76                              }
77                              if (length(rank.mir)>1)
78                              {
79                                      # Normalizing coefficients
80                                      rank.coeff <- abs(as.numeric(rank.coeff))
81                                rank.coeff <- rank.coeff/sqrt(sum(rank.coeff^2))
82                                      # Ranking
83                                      for (i in order(rank.coeff))
84                                      {
85                                      rank.mir.2 <- append(rank.mir.2,rank.mir[i])
86                                rank.coeff.2 <- append(rank.coeff.2,rank.coeff[i])
87                  rank.score <- append(rank.score,round((rank.rank/length(rank.mir))*100))
88                                      rank.rank <- rank.rank + 1
89                                      }
90                                      for (i in seq(length(rank.mir.2)))
91                                      {
92                                              # controllo se nella lista dei miRNA
93                                              # significativi c'e' quello casuale
```

```
94                                              if (rank.mir.2[i] == "x-random-1")
95                                              {
96                                      random.score.1 <- append(random.score.1,rank.score[i])
97                                      random.coeff.1 <- append(random.coeff.1,rank.coeff.2[i])
98                                              }
99                                              }
100                                     }
101                                     } # fine controllo se groupii > 0
102                             }
103                     count<-count+1
104                     } # fine del for per ogni file
105
106                     if (identical(random.score.1,NULL)==FALSE)
107                     {
108                             # inserisco lo score con l'indice riferito alla run (da uno a 100)
109                     random.score[[l]] <- random.score.1
110                     }
111                     fine<-proc.time()
112                     print(paste("Tempo ripetizione ",l," ",(fine-inizio)[3],sep=""))
113     } # fine delle ripetizioni
114
115         if(MultRun)
116         { stringa2<-"Multi" }
117         else { stringa2<-"One" }
118
119         save(random.score,file=paste(data.dir.salvataggio,"/random",
120         repeat.times,stringa2,data.dir,".Rdata",sep=""))
121
122         print("FINE BOOTSTRAP")
123         print("Summary:")
124         print(paste("Directory dati: ",data.dir,sep=""))
125         print(paste("Directory salvataggio: ",data.dir.salvataggio,sep=""))
126
127         stringa1<-"Bootstrap con Regressione "
128         if(MultRun)
129         {
130                 stringa1<-paste(stringa1,"Multirun Lasso ")
131         } else { stringa1<-paste(stringa1,"OneRun Lasso ") }
132         if(AGO)
133         {
134                 stringa1<-paste(stringa1,"con AGO ")
135         }else { stringa1<-paste(stringa1,"senza AGO ") }
136         print(paste("Opzioni: ",stringa1,sep=""))
137
138         print(paste("numero ripetizioni: ",repeat.times,sep=""))
139
140 }
```

Codice 4: *ROCArgoLasso()*

```
1  ROCArgoLasso<-function(random,overlap,dir.salvataggio,nome.plot,
2  plot.pdf=list(crea=TRUE,nome=""),plot.add=list(unico=FALSE,col="black"))
3  {
4          library(ROCR)
5          repeat.times<-length(random)
6          roc.score <- c()
```

```
7     roc.label <- c()
8     roc.label.2<-c()
9     j <- 1
10    for (i in seq(repeat.times))
11    {
12                if (identical(random.score[[i]],NULL)==FALSE)
13                {
14                        roc.score[[j]] <- c(overlap$select.score,random[[i]])
15  roc.label.2[[j]]<-c(rep(1,length(overlap$select.score)),rep(0,length(random[[i]])))
16  roc.label[[j]] <- c(rep("True",length(overlap$select.score)),rep("False",length(random[[i]])))
17                j <- j + 1
18                }
19    }
20        roc.pred <- prediction(roc.score,roc.label)
21    roc.perf <- performance(roc.pred,"tpr","fpr")
22    roc.pref2 <- performance(roc.pred,"auc")
23
24        print("---INIZIO␣ELABORAZIONE␣IMMAGINE---")
25
26    if(plot.add$unico)
27    {
28    plot(roc.perf,avg="threshold",col=plot.add$col,add=TRUE,lwd=3)
29    }
30    else
31    {
32             if(plot.pdf$crea)
33                {
34                pdf(paste(dir.salvataggio,"/ROC",plot.pdf$nome,".pdf",sep=""),height=8,width=8)
35                }
36        plot(roc.perf,lty=3,col="grey")
37        title(main=nome.plot,sub=paste("AUC:",mean(as.numeric(roc.pref2@y.values))))
38   plot(roc.perf,avg="threshold",colorize=TRUE,add=TRUE,lwd=5,print.cutoffs.at=seq(10,90,by=10))
39                abline(a=0,b=1,lty=2)
40                if(plot.pdf$crea)
41                {
42                dev.off()
43   print(paste("Curva␣ROC␣salvata␣come:␣",dir.salvataggio,"/ROC",plot.pdf$nome,".pdf",sep=""))
44                }
45    }
46        print("---FINE␣ELABORAZIONE␣IMMAGINE---")
47
48        print("SUMMARY␣Curva␣ROC")
49        print(paste("AUC␣medio:␣",mean(as.numeric(roc.pref2@y.values)),sep=""))
50  }
```

Riferimenti bibliografici

[1] Bartel DP, Chen CZ. Micromanagers of gene expression: the potentially widespread influence of metazoan microRNAs. Nat Rev Genet 2004;5:396-400

[2] Esquela-Kersher A and Slack FJ. Oncomirs-microRNAs with a role in cancer. Nat. Rev. Cancer 2006; 6:259-269

[3] Lee RC, Feinbaum RL, Ambros V. The C. elegans heterochronic gene lin-4 encodes small RNAs with antisense complementarity to lin-14. Cell 1993;75:843-54

[4] Reinhart BJ, Slack FJ, Basson M, et al. The 21-nucleotide let-7 RNA regulates developmental timing in Caenorhabditis elegans. Nature 2000:403:901-6

[5] Pasquinelli AE, Reinhart BJ, Slack FJ et al. Conservation of the sequence and tempoal expression of the let-7 heterochronic regulatory RNA. Nature 2000;408:86-9

[6] O'Connell RM, Raso DS, Chaudhuri AA, and Baltimore . Physiological and pathological roles for microRNAs in the immune system. Nat. Rev. Immunol. 2010; 10:11-122

[7] Rajewsky N. microRNA target predictions in animals. Nat. Genet. 2006; 38(suppl):S8-S13

[8] Brown JR, Sanseau P. A computational view of microRNAs and their target. Drug Discov 2005;10:595-601

[9] Sethupathy P, Corda B, Hatzigeorgiou AG. TarBase: A comprehensive database of experimentally supported animal microRNA targets. RNA 2006; 12(2): 192-7

[10] Dweep H, Sticht C, Pandey P, Gretz N. miRWalk–database: prediction of possible miRNA binding sites by walking the genes of three genomes. J Biomed Inform 2011; 44(5): 839-47

[11] Hsu SD, Lin FM, Wu WY, Liang C, Huang WC, et al. miRTarBase: a database curates experimentally validated microRNA-target interactions. Nucleic Acids Res 2011; 39: D163-9

[12] Sethupathy P, Megraw M and Hatzigeorgiou AG. A guide through present computational approaches for the identification of mammalian microRNA target. Nat. Methods 2006. 3:881-886

[13] Lim LP, Lau NC, Garrett-Engele P, Grimson A, Schelter JM, Castle J, Bartel DP, Linsley PS and Johonson JM. Microarray analysis shows that some microRNAs downregulate large numbers of target mRNAs. Nature 2005. 433:769-773

[14] Wang X. Systematic identification of microRNA functions by combining target prediction and expression profiling. Nucleic Acid Res 2006. 34:1646-1652

[15] Creigthon CJ, Nagaraja AK, Hanash SM, Matzuk MM and Gunaratne PH. A bioinformatic tool for linking gene espression profiling result with public databases of miRNA target predictions. RNA 2008; 14:2290-2296

[16] Huang JC, MorrisQD, Frey BJ. Bayesian inference of MicroRNA targets from sequence and expression data. Comput. Biol. 2007; 14(5):550-563

[17] Gennarino VA, Sardiello M, Avellino R, Meola N, Maselli V et al. MicroRNA target prediction by expression analysis of host genes. Genome Res 2009; 19(3): 481-490

[18] Sales G, Coppe A, Bisognin A, Biasiolo M, Bortoluzzi S, Romualdi C. MAGIA, a web-based tool for miRNA and genes integrated analysis. Nucleic Acids Res 2010; 38(Suppl):W352-9

[19] Nam S, Li M, Choi K, Balch C, Kim S and Nephew KP. MicroRNA and mRNA integrated analysis(MMIA): a web tool for examining biological functions of microRNA expression. Nucleic Acids Res 2009; 37:W356-W362

[20] Volinia S, Calin GA, Liu CG, Ambs S, Cimmino A, Petrocca F, Visone R, Iorio M, Roldo C, Ferracin M, Pruiett RL, Yanaihara N, Lanza G, Scarpa A, Vecchione A, Negrini M, Harris CC and Croce CM. A microRNA expressione signature of human solid tumors defines cancer gene targets. Proc Natl Acad Sci USA 2006; 103:2257-2261

[21] Lim LP, Lau NC, Garrett-Engele P et al. Microarray analysis shows that some microRNAs downregulate large numbers of target mRNAs. Nature 2005; 433: 769-73

[22] Lu J, Getz G, Miscka EA, Alvarez-Saavedra E, Lamb J, Peck D, Sweet-Cordero A, Ebert BL, Mak RH, Ferrando AA, Downing JR, Jacks T, Horvits HR and Golub TR. MicroRNA expression profiles classify human canceres. Nature 2005; 435:834-838

[23] Marchini S, Cavalieri D, Fruscio R, Calura E, Garavaglia D, Nerini IF, Mangioni C, Cattoretti G, Clivio L, Beltrame L, Katsaros D, Scarampi L, Menato G, Perego P, Chiorino G, Buda A, Romualdi C, D'Incalci M. Association between miR-200c and the survival of patients with stage I epithelial ovarian cancer: a retrospective study of two indepedent tumor tissue collection. 2011.

[24] Cannista SA. Cancer of the ovary. N Engl J Med 2004; 351: 2519-29.

[25] Cummins JM, Velculescu VE. Implications of micro-RNA profiling for cancer diagnosis. Oncogene 2006; 25: 6220-27.

[26] Dalmay T, Edwards DR. MicroRNAs and hellmarks of cancer. Oncogene 2006; 25: 6170-75.

[27] Tricoli JV, Jacobson JW. MicroRNA: potential for cancer detection, diagnosis and prognosis. Cancer res 2007; 67:4553.55.

[28] (Merritt WM, Lin YG, Han LY et al. Dicer, Drosha and outcomes in patients with ovarian cancer. N Engl J Med 2008; 14: 7850-60).

[29] Quackenbush J. Computational analysis of microarray data. Nat Rev Genet 2001; 2(6): 418-427.

[30] Bilban M, Buehler LK, Head S, Desoye G, Quaranta V. Normalizing DNA microarray data. Curr Issues Mol Biol 2002; 4:57-64.

[31] Smyth G, Speed T. Normalization of cDNA microarray data. Methods 2003; 31(4): 265-273.

[32] Chiogna M, Massa MS, Risso D, Romualdi C. A comparison on effects of normalisations in the detection od differentially espressed genes. BMA Bionformatics 2009; 10:61.

[33] Troyanskaya O, Cantor M, Sherlock G, Brown P, Hastie T, Tibshirani R, Bostein D, Altman R. Bioinformatics 2001; 17(6): 520-525.

[34] K. Pearson, "Mathematical Contributions to the Theory of Evolution. III. Regression, Heredity and Panmixia". Philosophical Transactions of the Royal Society of London 1896; A: 253-318

[35] K. Pearson, "Mathematical Contributions to the Theory of Evolution. III. Regression, Heredity and Panmixia". Philosophical Transactions of the Royal Society of London 1896; A: 253-318

[36] Wang YP, Li KB. Correlation of expression profiles between microRNAs and mRNA target using NCI-60 data. BMC Genomics 2009; 10:28.

[37] Jayaswal V, LutherborrowM, Ma DD, Hwa Yung Y. Identification of microRNA with regulatory potential using a matchet microRNA-mRNA timecourse data. Nucleic Acids Res 2009; 37(8): e60

[38] Ruike Y, Ichimura A, Tsuchya S, Shimizu K, Kunimoto R, et al. Global correlation analysis for micro-RNA and mRNA expression profiles in human cell lines. J Hum Genet 2008; 53(6): 515-523.

[39] Gennarino VA, Sardiello M, Mutarelli M, Dharmalingam G, Maselli V, Lago G, Banfi S. HOCTAR database: A unique resource for microRNA target prediction. Gene 2011; 480(1-2): 51-58.

[40] L. Euler. De progressionibus transcendentibus seu quarum termini generales algebraice dari nequeunt. Comm. Acad. Sci. Petropolitanae 1730; vol. 5: 36-57.

[41] Bisognin A, Sales G, Coppe A, Bortoluzzi S, Romuladi C. Magia^2: from miRNA and genes expression data integrative analysis to microRNA-trascription factor mixed regulatory circuits (2012 update). Nucleic Acids Research 2012; Vol.40: W13-W21.

[42] Kraskov A, Stogbauer H and Grassberger P. Estimatink mutual information. Phys. Rev. E Stat. Nonlin. Soft Matter Phys 2004; 69: 066138.

[43] Graham E, Gargano A, Timmermann A. Complete subset regressions 2013. Journal of Econometrics.

[44] Arthur E, Hoerl and Robert W, Kennard. Technometrics 1970; Vol.12: 55-67.

[45] Tibshirani R. Regression Shrinkage and Selection via the Lasso. J R Statist Soc 1996; 58:267-288.

[46] Muniategui A, Nogales-Cadenas R, Vazaquez M, Arangurel LX, Agirre X, et al. Quantification of miRNA-mRNA Interaction. PLoS ONE 2012; Vol.7(2): e30766.

[47] Yiming L, Yang Z, Wubin Q, Minghua D and Chenggang Z. A Lasso regression model for the construction of microRNA-target regulatory networks. System biology 2012; Vol.27: 2406-2413.

[48] Corrada D, Theussl S. Rcplex: Rinterface to CPLEX. 2009

[49] Kim SJ, Khon K, Lusting M, BoydS, Gorinevsky D. An interior-point method for large-scale L1-regularized least sqares. IEEE Journal on selected Topics in Signal Processing 2007; I(4): 606-617.

[50] Goemann JJ. L_1 Penalized Estimation in the Cox Proportional Hazards Model. Biometrical Journal 2010; Vol.52(1): 70-84.

[51] Nguyen DV, Rocke DM. Tumor classification by partial least squares using microarray gene expression data. Bioinformatics 2002; 18a: 39–50.

[52] Purohit P, Rocke DM. Discriminant models for high-throughput proteomics mass spectrometer data. Proteomics 2003; 3: 1699–1703.

[53] Datta S, Le-Rademacher J, Datta S. Predicting patient survival from microarray data by accelerated failure time modeling using partial least squares and LASSO. Biometrics 2007; 63: 259–271.

[54] Li X, Gill R, Cooper NGF, Yoo JK, Datta S. Modeling microRNA-mRNA Interaction Using PLS Regression in Human Colon Cancer. BMC Med Genomics 2011; 4:44.

[55] de la Fuente A, Bing N, Hoeschele I, Mendes P. Discovery of meaningful associations in genomic data using partial correlation coefficients. Bioinformatics 2004; 20: 3565-3574.

[56] Mangwene PM, Kim J. Estimating genomic coexpression networks using first-order conditional independence. Genome Biology 2004; 5: R100.

[57] Willie A, Zimmermann P, Vranova E, et al. Sparse graphical Gaussian modeling of the isoprenoid gene network in Arabidopsis thaliana. Genome Biology 2004; 5: R92.

[58] Whittaker J. Graphical Models in Applied Multivariate Statistics. 1990.

[59] Schafer J, Strimmer K. An empirical Bayes approach to inferring large-scale gene association networks. Bioinformatics 2005a; 21: 754-764

[60] Yitzhaki S. Gini's mean difference: a superior measure of variability for non-normal distributions. METRON Internetional Journal of Statistics 2003; LXI: 285-316.

[61] Ma C, Zhou Y, Huang SH. Inequalities and duality in gene coexpression networks of HIV-1 infection revelaed by the combination of the double-connectivity approach and Gini's method. J Biomed Biotechnol 2011; 926407.

[62] Berry S, Abbruscato P, Faivre-Rampant O, et al. Characterization of WRKY co-regulatory networks in rice and Arabidopsis. BMC Plant Biol 2009; 9:120

[63] Ma C, Wang X. Application of the Gini Correlaton Coefficient to Infer Regulatory Relationships in Tarscriptome Analysis. Bioinformatics 2012; Vol.160: 192-203.

[64] Bolasco S. Analisi multidimensionale dei dati. Metodi, strategie e criteri d'interpretazione, 1999, Roma, Carocci

[65] Leng C, Lin Y and Wahba G. A note on the Lasso and related procedures in model selection. Sat. Sinica 2006; 16: 1273-1284.

[66] Cittelly DM, Dimitrova I, Howe EN, et al. Restoration of miR-200c to ovarian cancer reduces tumor burden and increases sensitivity to paclitaxel. Mol Cancer Ther 2012; 11(12): 2556-65

[67] Yori JL, Seachrist DD, Johnson E, ed al. Krüppel-like Factor 4 Inhibits Tumorigenic Progression and Metastasis in a Mouse Model of Breast Cancer. NEOPLASIA 2011; 13(7)

[68] Derek C Radisky. miR-200c at the nexus of epithelial-mesenchymal transition, resistance to apoptosis, and the breast cancer stem cell phenotype. Breast Cancer Res 2011; 13(3)

[69] Gianpiero Di Leva and Carlo M. Croce. The Role of microRNAs in the Tumorigenesis of Ovarian Cancer. Front Oncol 2013; v.3

[70] Marchini S, Cavalieri D, Fruscio R, Calura E, Garavaglia D, Nerini IF, Mangioni C, Cattoretti G, Clivio L, Beltrame L, Katsaros D, Scarampi L, Menato G, Perego P, Chiorino G, Buda A, Romualdi C, D'Incalci M. Association between miR-200c and the survival of patients with stage I epithelial ovarian cancer: a retrospective study of two indepedent tumor tissue collection. 2011.