

Università degli studi di Padova
Dipartimento di Scienze Statistiche
Corso di Laurea Magistrale in
Scienze Statistiche



ANALISI DI RETI DINAMICHE IN CONTESTO SINGLE-CELL RNA-SEQ

Relatore Prof. Mauro Bernardi
Dipartimento di Scienze Statistiche

Correlatore Prof. Davide Risso
Dipartimento di Scienze Statistiche

Laureando Piero Trevisan
Matricola N 2022775

Anno Accademico 2023/2024

A tutti coloro che mi hanno sempre sostenuto

Indice

Introduzione	11
1 Introduzione ai dati single-cell	13
1.1 Dal DNA alle proteine	13
1.2 RNA-seq	15
1.3 scRNA-seq	16
1.4 Analisi di dati single-cell	18
1.5 Scopo della tesi	19
2 Analisi Descrittive	21
2.1 Contesto biologico	21
2.1.1 Cellule staminali	21
2.1.2 Analisi pseudotemporale	23
2.2 Analisi preliminari	26
2.3 Analisi esplorative	28
3 Modelli Grafici	35
3.1 Introduzione al problema	35
3.2 Modelli grafici Gaussiani	36
3.3 Graphical-LASSO	39
3.3.1 Introduzione alla regressione penalizzata	39
3.3.2 Graphical-LASSO: problema ed applicazione	40
3.4 Metodo delle direzioni alternate dei moltiplicatori	48
3.4.1 Problemi vincolati e discesa duale	48
3.4.2 Algoritmo ADMM	51
3.4.3 Convergenza dell'algoritmo	53

3.4.4	Condizioni di ottimalità e criteri per fermare l'algoritmo	54
3.5	ADMM per graphical-LASSO	56
4	Modelli Grafici dinamici	65
4.1	Introduzione al problema	65
4.2	Indici descrittivi	66
4.3	Analisi separata per tipo cellulare	68
4.4	Analisi dinamica via Rolling Window	71
4.5	Interpretazione biologica e conclusioni	73
A	Figure aggiuntive	77
B	Codice C++ degli algoritmi presentati	79
C	Codice R degli algoritmi presentati	83

Elenco dei codici

B.1	Algoritmo ADMM per graphical-LASSO.	79
C.1	Codice per la stima di Theta ottimo.	83
C.2	Codice per il calcolo degli indici descrittivi di una lista di grafi.	84

Elenco delle figure

2.1	Tipi cellulari dell'epitelio olfattivo.	22
2.2	Passaggi dell'analisi pseudotemporale.	25
2.3	Differenti traiettorie cellulari.	27
2.4	Ordinamento pseudotemporale stimato da Slingshot.	29
2.5	PCA colorata per i quattro tipi cellulari.	30
2.6	PCA colorata per lo pseudotempo.	31
2.7	t-SNE colorato per i quattro tipi cellulari.	32
2.8	t-SNE colorato per lo pseudotempo.	33
3.1	Grafo stimato via <i>CVglasso</i>	47
3.2	Andamento indice BIC.	62
3.3	Grafo stimato via ADMM graphical-LASSO.	63
4.1	Andamento indice BIC per i quattro gruppi cellulari.	70
4.2	Grafi stimati per i quattro gruppi cellulari.	70
4.3	Andamento indici descrittivi a livello di grafo.	72
4.4	Andamento indici descrittivi dei geni più importanti.	73
A.1	Grafi stimati per quattro i gruppi cellulari usando solo i geni comuni.	77
A.2	Andamento indici descrittivi a livello di grafo.	78
A.3	Andamento indici descrittivi dei geni più importanti.	78

Introduzione

L'idea principale di questa Tesi nasce dalla curiosità di studiare l'evolvere di un processo biologico nel tempo tramite opportune metodologie statistiche. Negli ultimi anni le tecnologie di sequenziamento del DNA si sono velocemente raffinate, dando vita a protocolli sempre più efficaci per il sequenziamento di mRNA a singola cellula (scRNA-seq). Tra le innumerevoli applicazioni di tale recente tecnologia, una delle più interessanti è l'analisi pseudotemporale, detta anche analisi di traiettoria. Questo tipo di analisi offre una visione diversa di ordinamento temporale rispetto al consueto tempo fisico, ma è molto utile in campo biologico per capire ed analizzare al meglio l'evolvere di un processo biologico nel suo compimento. Per una maggiore chiarezza dell'esposizione dei contenuti, in questa Tesi verranno alternati concetti e termini di biologia molecolare e di modellazione statistica, in particolare modelli grafici Gaussiani.

Il lavoro è suddiviso in 4 Capitoli: il Capitolo 1 fornisce una panoramica dei concetti principali di biologia e genomica, con particolare attenzione ai dati scRNA-seq e la loro analisi. Nel Capitolo 2 viene presentato il dataset oggetto di analisi ed una sua principale analisi descrittiva; viene inoltre descritto l'esperimento biologico da cui sono tratti i dati ed alcuni concetti fondamentali quali lo pseudotempo. Nel corso del Capitolo 3 si effettua una panoramica generale dei modelli di regressione penalizzati e modelli grafici penalizzati, in particolare il problema graphical-LASSO. Sono stati inoltre introdotti la teoria del metodo delle direzioni alternate dei moltiplicatori (ADMM) per problemi di ottimizzazione ed una sua formulazione in C++ per il problema biologico preso in esame. Per finire, vengono mostrati i risultati applicati al dataset di partenza, incluso il processo di selezione di alcuni parametri. Il

Capitolo 4 estende il caso statico del precedente Capitolo e tiene in considerazione lo pseudotempo e quindi la dinamicità del processo; oltre ad una breve introduzione sugli indici descrittivi dei grafi, tramite una tecnica Rolling Window sono stati studiati gli eventuali cambiamenti della struttura di dipendenza tra geni e quindi del processo biologico. È stata infine eseguita una rassegna della letteratura biologica per trovare eventuali riscontri con i risultati trovati.

Capitolo 1

Introduzione ai dati single-cell

In questo Capitolo vengono introdotti i concetti basilari di biologia molecolare e di genomica, per passare alla descrizione delle tecnologie utilizzate in questo campo, quali l'RNA-seq ed il scRNA-seq, assieme alle loro applicazioni. Vengono definite le caratteristiche principali dei dati single-cell ed alcune metodologie statistiche sviluppate per analizzare questa tipologia di dato.

1.1 Dal DNA alle proteine

La cellula è l'unità fondamentale di tutti gli organismi viventi, fondamentale per lo svolgimento dei processi biologici. La cellula è strutturalmente definita da una membrana cellulare che racchiude il citoplasma, una soluzione gelatinosa nella quale sono situati i vari organelli ed il nucleo. Quest'ultimo contiene gran parte del patrimonio genetico sottoforma di DNA, compresso in strutture chiamate cromosomi. Il DNA (*DeoxyriboNucleic Acid*, Acido Deossiribonucleico) è una molecola formata da tante unità che si ripetono, i nucleotidi; ogni nucleotide è caratterizzato da una base azotata. Nel DNA le basi azotate sono quattro: Adenina (A), Guanina (G), Citosina (C) e Timina (T). Il DNA è composto da due filamenti in modo tale che le basi azotate di un filamento si accoppino con quelle dell'altro secondo lo schema A-T e G-C. Per questo motivo si dice che ogni filamento è il complemento inverso dell'altro: l'Adenina si accoppia sempre con la Timina e viceversa, così come

la Guanina si accoppia sempre con la Citosina e viceversa. Il DNA non ha orientamento, in quanto l'inizio di un filamento è la fine dell'altro; i concetti di "inizio" e "fine" in genomica vengono "tradotti" rispettivamente in 5' e 3'. Esistono alcune sequenze specifiche di DNA chiamati geni, che codificano una proteina; nel genoma umano si contano oltre 20000 geni differenti. La procedura di sintesi delle proteine avviene però nei ribosomi, degli organelli situati nel citoplasma della cellula; per trasportare le informazioni genetiche del DNA presenti nel nucleo fino ai ribosomi nel citoplasma la cellula sfrutta l'mRNA o RNA messaggero. L'RNA (*RiboNucleic Acid*, Acido Ribonucleico) è una molecola a filamento unico composta da nucleotidi, le cui basi azotate sono Adenina (A), Guanina (G), Citosina (C) ed Uracile (U). Tra i diversi tipi di RNA esistenti, l'mRNA gioca un ruolo fondamentale nella sintesi proteica; le informazioni contenute in un gene sul DNA vengono infatti copiate in una molecola di mRNA mediante un processo chiamato trascrizione e trasportate dal nucleo al citoplasma della cellula. I geni sono formati da esoni, che rappresentano la parte codificante, ed introni che non codificano; dopo la trascrizione, un processo chiamato splicing ha il compito di rimuovere gli introni ed unire gli esoni in modo tale da formare una molecola di mRNA matura. Negli organismi biologicamente più semplici, come i lieviti, ad ogni gene corrisponde una sola proteina; gli organismi più complessi hanno però sviluppato un metodo che rende il DNA molto più efficiente. Tramite lo splicing alternativo, infatti, non vengono tagliati tutti gli introni e non vengono uniti tutti gli esoni, ma vengono ricombinati in modo diverso, in modo tale che ad ogni gene corrispondano più proteine, chiamate isoforme del gene. Ragionando in questi termini, un gene può essere quindi visto come l'insieme di tutte le sue isoforme, anche se può non esistere una sua isoforma che contiene tutti gli esoni. Completato il processo di splicing si arriva ad una molecola di mRNA matura ed i ribosomi sintetizzano le proteine a partire dal codice contenuto nell'mRNA tramite un processo chiamato traduzione. La cellula può richiedere un certo numero di proteine che le sono utili per una determinata funzione, di conseguenza per ogni gene possono essere create molte copie di RNA, dunque di proteine. Inoltre, ogni cellula richiede solamente le proteine che le sono funzionali in un certo momento e per un certo scopo; questo è il motivo per cui esistono molti diversi tipi cellulari nonostante ogni cellula

contienga la stessa dotazione di DNA. La misura dell'attività di migliaia di geni alla volta è detta analisi di espressione genica.

1.2 RNA-seq

La principale tecnologia utilizzata per lo studio dell'espressione genica è il sequenziamento dell'RNA (RNA-seq); sequenziare un filamento di RNA significa leggere la sequenza dei nucleotidi che lo compongono. I sequenziatori di seconda generazione sono una serie di tecnologie commercializzate a partire dal 2007 che hanno rimpiazzato la metodologia più comune usata precedentemente, quella dei *microarray*. Il vantaggio di questo metodo è l'uso della parallelizzazione, in questo modo è possibile sequenziare più filamenti contemporaneamente e con un costo più basso; è possibile inoltre lo studio dei geni non conosciuti. Prima del sequenziamento è necessaria una fase chiamata preparazione della libreria; essa è l'insieme dei trascritti estratti e preparati per il sequenziamento. Dal momento che nessun sequenziatore attuale è in grado di leggere l'intero filamento di RNA, esso viene frammentato in *reads*, delle sequenze più piccole lunghe dalle 50 alle 150 basi. Un'alternativa è frammentare una molecola di cDNA (DNA complementare), che viene creata a partire dall'RNA con un processo chiamato retrotrascrizione. Un vantaggio di questa procedura è che la molecola ottenuta risulta più stabile, dal momento che il DNA ha una struttura a doppia elica. Ad ogni *read* vengono attaccati degli adattatori, che servono ai macchinari, infine se il materiale a disposizione è poco si procede con l'amplificazione. In questo modo le molecole sono pronte ed il sequenziatore procede a leggerne le sequenze di nucleotidi. La fase successiva al sequenziamento è detta allineamento e consiste nell'identificare da quale gene è stata generata ogni *read*. È possibile allineare le *reads* sull'intero genoma oppure sul trascrittoma, ovvero l'insieme degli RNA trascritti a partire dal genoma. I vantaggi della prima opzione sono la possibilità di scoprire geni nuovi e l'identificazione di eventi di splicing alternativo, mentre la seconda opzione ha il grande vantaggio di ridurre lo spazio di ricerca, dal momento che solo il 5-10% del genoma è trascritto. Infine, grazie alla quantificazione, è sufficiente contare il numero di *reads* che mappano su ogni gene per ottenere un conteggio che rappresenta il livello

di espressione del gene: più il numero è alto, maggiormente è espresso il gene. È importante precisare che il tipo di conteggio che si ottiene da questi esperimenti non rappresenta una misura assoluta del numero di molecole di RNA, ma una misura relativa; pertanto, tali risultati hanno senso solo se confrontati con altri risultati dello stesso esperimento. Solitamente i conteggi vengono riportati in una tabella chiamata matrice di conteggio che presenta n righe corrispondenti ai campioni e J colonne corrispondenti ai geni; l'elemento y_{ij} della matrice rappresenta dunque il numero di *reads* nel campione i che mappano il gene j , quindi l'espressione del gene j nel campione i . Nell'ambito della biologia e della bioinformatica, è solito usare la trasposta di questa matrice, dove i geni sono sulle righe ed i campioni sulle colonne.

1.3 scRNA-seq

Il metodo di sequenziamento descritto nel precedente Paragrafo parte dall'estrazione dell'RNA di un campione biologico, che può essere un tessuto composto da migliaia di cellule, spesso appartenenti a diversi tipi. Di conseguenza l'RNA di tutte le cellule si meschia e quello che viene misurato per ogni gene è il suo livello di espressione medio tra tutte le cellule da cui è composto il campione. Queste tipologie di misurazioni vengono chiamate *bulk* RNA-seq (RNA-seq "all'ingrosso"). Ci sono però dei casi specifici in cui è di interesse misurare l'espressione genica a livello di singola cellula, se ad esempio l'obiettivo della ricerca è identificare nuovi tipi cellulari. Viene dunque sfruttata una tecnologia relativamente recente chiamata scRNA-seq (*Single Cell RNA Sequencing*, sequenziamento di RNA a singola cellula); lo scopo è quello di isolare le singole cellule tramite dei protocolli per poi estrarre l'RNA da ogni cellula e successivamente sequenziarlo. Di seguito vengono introdotti tre dei principali protocolli scRNA-seq:

- **SMART-seq2**: descritto per la prima volta nel 2014, rimane tuttora tra i metodi più efficienti. Questo protocollo prevede la separazione, manuale o attraverso smistamento automatico, delle cellule ed il loro posizionamento in micro-pozzetti. Il suo vantaggio è l'alta qualità delle cellule che si ottiene: prima di sequenziare l'RNA, infatti, è possibi-

le osservare le cellule al microscopio, evitando così di sequenziare due cellule che sono rimaste accattate (*doublets*) o cellule che stanno morendo. Uno svantaggio è che si tratta di un metodo laborioso e manuale, di conseguenza possono essere sequenziate solamente poche cellule per volta. Viene molto usato quando si è interessati ad identificare tutti i geni che vengono espressi da un certo tipo cellulare e servono quindi molte *reads*.

- **Fluidigm C1:** viene presentato per la prima volta nel 2014, anche se ora non è più molto in uso. In questo protocollo le cellule entrano in un chip trasparente che, grazie a principi di microfluidica, è in grado di separare le cellule ed amplificare il loro RNA separatamente. Il grande vantaggio di questo metodo è che le cellule vengono separate in maniera automatica e senza bisogno dell'intervento umano; in più, essendo il chip trasparente, si possono osservare al microscopio le cellule prima del sequenziamento e scartare le cellule morte. Lo svantaggio è che ogni chip ha solamente 96 micro-pozzetti, è dunque possibile sequenziare solamente 96 cellule per volta. Questo metodo è inoltre suscettibile alle *doublets*.
- **Drop-seq:** presentato nel 2015, questo protocollo sfrutta delle gocce (*droplet*) di gel per incapsulare le cellule assieme ad una perlina. Quest'ultima contiene gli enzimi che servono a costruire la libreria ed un codice a barre univoco che identifica ogni *droplet*, quindi ogni cellula. Un vantaggio di questo metodo è che si possono sequenziare migliaia di cellule per ogni campione, è dunque preferibile quando lo scopo della ricerca è caratterizzare tutti i tipi di cellule di un tessuto e serve un numero molto alto di cellule. Lo svantaggio è che i costi di sequenziamento sono alti, pertanto spesso si sequenziano poche *reads* per cellula; per questo motivo risulta complicato studiare geni poco espressi.

Nel contesto scRNA-seq esistono due tipi differenti di quantificazione: la quantificazione *full-length*, che permette di avere *reads* che coprono uniformemente tutto il gene e la quantificazione *tag al 3'*, in cui solo il 3' del gene viene sequenziato e di conseguenza tutte le *reads* andranno a mappare all'estremità del gene. La prima tipologia ha il vantaggio di poter osservare il gene

nella sua interezza e consente quindi di studiare l'espressione delle isoforme del gene e gli eventi di splicing alternativo. La seconda alternativa ha il vantaggio di richiedere meno *reads* per ogni gene, è dunque possibile sequenziare più cellule in un'unica corsa di sequenziamento. Di conseguenza ci si trova di fronte ad un trade-off: se l'obiettivo è identificare tutti i geni espressi da un certo tipo cellulare servono più *reads* per gene ed è quindi preferibile usare *SMART-seq2* o *Fluidigm C1*, che sfruttano la quantificazione *full-length*, con lo svantaggio che si avranno meno cellule a disposizione. Se invece è di interesse caratterizzare tutti i tipi cellulari di un tessuto servono molte cellule ed è dunque meglio usare *Drop-seq*, che sfrutta la quantificazione *tag 3'*, con lo svantaggio che si avranno meno *reads* per gene. Similarmente all'RNA-Seq, i dati single-cell vengono rappresentati da una matrice di conteggio con n righe corrispondenti alle cellule e J colonne corrispondenti ai geni. Nel successivo Capitolo sarà utilizzata la notazione usata in bioinformatica, ovvero una matrice di conteggio dove i geni sono sulle righe ed i campioni sulle colonne.

1.4 Analisi di dati single-cell

Come spiegato precedentemente, analizzando i dati *bulk* si ha a disposizione molto materiale, dal momento che si estrae l'RNA da un tessuto composto da molte cellule. Nel contesto single-cell si va invece ad estrarre l'RNA da ogni singola cellula, di conseguenza il materiale a disposizione è molto poco e va amplificato prima di procedere con il sequenziamento. Questa procedura comporta però due grossi problemi ai dati da studiare: il primo è un eccesso di zeri nella matrice di conteggio dovuto al fatto che alcuni geni, specialmente quelli poco espressi, non vengono mai amplificati e risultano come mancanti. Il secondo problema è l'*amplification bias*, vale a dire una distorsione nell'amplificazione dovuta al fatto che quei geni che vengono amplificati vengono amplificati a dismisura. Assieme, questi due problemi provocano un incremento della variabilità dei dati. Una possibile soluzione per contrastare l'*amplification bias* è quella di inserire nella libreria gli UMI (*Unique Molecular Identifiers*), ossia una sequenza di 4-10 nucleotidi che viene agganciata ai trascritti durante la retro-trascrizione. In questo modo, se dopo l'amplificazione si trovano due sequenze con lo stesso UMI, si sa che non si tratta

di due copie indipendenti di RNA ma di una copia esatta dovuta all'amplificazione. Anzichè quindi contare il numero di *reads* associato ad ogni gene si può contare il numero di UMI univoci per ogni gene; in tal modo è possibile ridurre la distorsione e la variabilità tecnica nei dati. Uno svantaggio di questo approccio è che il numero di UMI sarà sempre minore o al massimo uguale al numero di *reads*, poichè alcune *reads* sono copie della stessa molecola di RNA; visto che solitamente il numero di *reads* è basso, il numero di UMI sarà ancora più basso. Un altro classico problema dei dati single-cell è l'identificazione di *doublets*, *droplet* vuote e cellule che stanno morendo; si è dunque interessati alla qualità delle cellule che poi verranno studiate. A differenza dei dati *bulk*, nei dati single-cell si ha a disposizione un numero molto elevato di osservazioni; è quindi possibile eliminare campioni di bassa qualità e valori estremi. A questo scopo sono a disposizione alcune tecniche statistiche per valutare delle soglie ottimali riguardo la qualità dei campioni. Nel corso degli ultimi anni sono stati proposti protocolli per il sequenziamento di dati a singola cellula sempre più efficaci e meno costosi; assieme a questa relativamente recente tecnologia che si sta sviluppando, cresce anche il numero di possibili applicazioni bioinformatiche ad essa connessa. Questo tipo di dati può ad esempio essere sfruttato per rispondere ad una domanda ancora aperta nel mondo della biologia, ossia quanti sono i differenti tipi cellulari presenti nel corpo umano. Le analisi di clustering possono essere molto utili al caso, anche se i dati single-cell hanno solitamente numerosità molto elevata ed è quindi necessario trovare metodi computazionalmente efficienti. Una volta appurata l'esistenza di alcuni gruppi cellulari, un passo successivo è scoprire di che tipo cellulare si tratta. In questo ambito l'analisi di dati single-cell permette di identificare dei geni marcatori, ovvero geni la cui espressione è specifica di quel tipo cellulare. Un altro interessante campo di studio è l'analisi pseudotemporale, che viene spiegata in modo esaustivo nel seguente Capitolo.

1.5 Scopo della tesi

Lo scopo di questa Tesi è studiare il modo in cui la struttura di dipendenza tra geni cambia al variare dello stato di maturazione delle cellule staminali

lungo una traiettoria pseudotemporale. Sono state introdotte modellazioni statiche e dinamiche per la stima di reti geniche, grazie alle quali è possibile anche effettuare interessanti interpretazioni a livello biologico dei risultati trovati.

Capitolo 2

Analisi Descrittive

2.1 Contesto biologico

2.1.1 Cellule staminali

Le cellule staminali sono cellule primitive, non specializzate, che hanno la capacità di potersi trasformare in qualsiasi tipo cellulare esistente. È di interesse biologico studiare il loro processo di maturazione, ovvero il modo in cui si trasformano da cellule staminali immature a cellule specializzate adulte; ci si chiede quindi come e quando una cellula staminale "sceglie" quale cellula matura diventare. Un sistema biologico relativamente semplice e molto usato per cercare di rispondere a questa domanda è l'epitelio olfattivo del topo. L'epitelio olfattivo è un rivestimento che ricopre la parte esterna dell'apparato olfattivo; si tratta della zona dove vengono captate le molecole odorose, i cui stimoli vengono inviati al cervello per la loro elaborazione. In Figura 2.1 (Brann et al. (2020)) viene riportata una rappresentazione dell'epitelio olfattivo del topo; i tipi cellulari da cui è composto sono:

- HBC (*horizontal basal cells*, cellule basali orizzontali), si tratta di cellule staminali che fungono da riserva cellulare in caso di un danno al tessuto, ad esempio se viene annusata una sostanza tossica;
- GBC (*globose basal cells*, cellule basali globose), sono cellule staminali progenitrici dei neuroni;

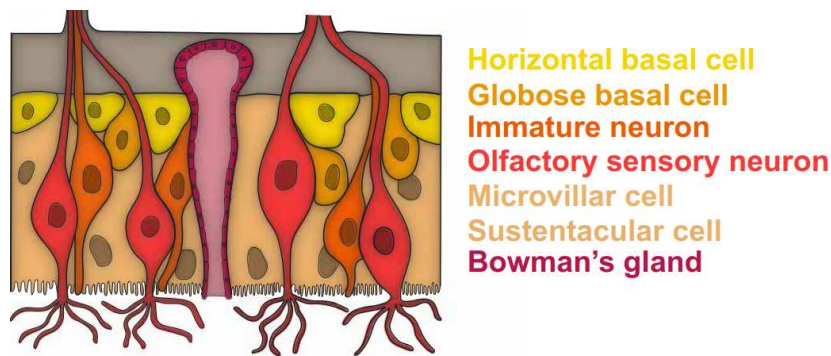


Figura 2.1: Tipi cellulari dell'epitelio olfattivo.

- OSN (*olfactory sensory neurons*, neuroni sensoriali olfattivi) noti anche come ORN (*olfactory receptor neurons*, neuroni recettori olfattivi), sono cellule mature, nello specifico neuroni, che si occupano di captare gli odori. Si dividono in immaturi (iOSN) e maturi (mOSN);
- SUS (*sustentacular cells*, cellule sustentacolari), sono cellule che hanno la funzione di supporto dell'epitelio olfattivo;
- MV (*microvillar cells*, microvilli), sono cellule differenti dalle cellule di tipo SUS e che hanno comunque funzione di supporto;
- BG (*Bowman's gland*, ghiandole di Bowman), sono cellule che secernono un muco con lo scopo di proteggere e mantenere umida la superficie della mucosa olfattiva.

Le cellule staminali presenti nell'epitelio olfattivo possono attivarsi per riparare l'epitelio in caso di danneggiamento dovuto a fattori esterni. A seguito di un'incidente, infatti, le cellule HBC che normalmente sono quiescenti, si attivano per differenziarsi e ricostituire tutti i principali tipi di cellule mature dell'epitelio, quali i mOSN, le BG ed i MV. In particolare, per arrivare allo stadio di neuroni maturi, le cellule passano dallo stato HBC a GBC, da GBC a iOSN ed infine da iOSN a mOSN. I dati utilizzati in questa Tesi provengono da uno studio (Brann et al. (2020)) che si prefigge l'obiettivo di capire il meccanismo attraverso il quale il virus SARS-CoV-2 attacca l'apparato respiratorio umano, provocando talvolta la perdita temporanea dell'olfatto,

detta anosmia. Nello studio sono stati danneggiati gli epitelii olfattivi dei topi con il metimazolo, un comune farmaco per la tiroide che viene spesso usato in questi contesti in quanto causa una tossicità tessuto-specifica nella mucosa olfattiva dei roditori.

2.1.2 Analisi pseudotemporale

Una delle più interessanti applicazioni dei dati single-cell riguarda l'analisi dello pseudotempo, detta anche analisi di traiettoria. Molti processi biologici sono caratterizzati da una transizione delle cellule da uno stato cellulare ad un altro che avviene in modo continuo; non esiste quindi una chiara distinzione tra i differenti stati cellulari. Un esempio di questo tipo di processo può essere il ciclo di divisione cellulare, che permette alle cellule di replicarsi. In un processo di questo tipo, dunque, le cellule cambiano stato grazie ad un graduale cambiamento dell'espressione genica. Un'interessante domanda di interesse biologico è capire quali geni ed in che modo sono connessi al processo, studiando come cambia l'espressione genica nel corso del processo biologico di riferimento. Una possibile soluzione può essere quella di quantificare l'espressione cellulare di un tessuto composto da tante cellule e vedere come cambia a diversi istanti temporali. Ad esempio, se si è particolarmente interessati all'espressione di un gene, si può valutare se e come cambia la sua espressione media nel corso del tempo. Il principale problema di questo approccio è che le cellule ricevono il segnale di "partenza" del processo biologico in maniera sparsa e dunque non tutte si attivano allo stesso istante temporale. Ad esempio, se viene danneggiato l'epitelio del topo, il segnale di attivazione arriva alle cellule staminali immature ad istanti temporali diversi e non risultano quindi essere sincronizzate; di conseguenza, i dati che riguardano l'espressione genica ad istanti temporali differenti sono caratterizzati da una forte eterogeneità. Una possibile alternativa per far fronte a questo problema è cercare un'ordinamento delle cellule non in base al tempo di "orologio", ma di istante di reazione al segnale cellulare. Si possono vedere dunque le cellule come dei punti lungo una traiettoria (*lineage*) attraverso lo spazio ad alta dimensionalità dell'espressione genica, in cui esse sono ordinate dallo pseudotempo, una variabile latente (non osservata) che viene stimata a

partire dall'espressione genica, che indica la posizione di ogni cellula lungo la traiettoria di riferimento. In Figura 2.2 (Campbell and Yau (2016)) vengono riportate le fasi appena descritte: in Figura 2.2A si vede come le cellule sono ordinate secondo il tempo fisico ed ogni colore rappresenta un diverso istante temporale in cui le cellule vengono catturate. Come detto in precedenza, misurare l'espressione genica delle cellule ad ogni istante temporale fornisce risultati difficilmente interpretabili. Infatti, come si vede dalla Figura 2.2B, nel momento in cui si selezionano le cellule per il scRNA-seq, l'ordinamento temporale viene perso. Di conseguenza, le cellule possono essere ordinate lungo una traiettoria pseudotemporale (Figura 2.2C), in cui le prime cellule sono quelle che si sono attivate dopo e che quindi hanno appena iniziato il processo biologico, mentre le ultime si sono attivate prima. Il punteggio di pseudotempo assegnato ad ogni cellula sta quindi a raffigurare una misura relativa di "progressione" del processo biologico. Grazie allo pseudotempo è possibile identificare dei geni che sono differenzialmente espressi lungo la traiettoria e quali sono differenzialmente espressi prima e dopo, stando ad indicare che tali geni sono in qualche modo responsabili del processo biologico di riferimento (Figura 2.2D). Nei casi più semplici le traiettorie avranno un punto di inizio ed un unico punto finale, ma possono esistere anche traiettorie più complesse con diversi punti finali. Nel caso del processo di maturazione delle cellule staminali, i punti finali saranno più di uno dal momento che le cellule di partenza possono trasformarsi in diversi tipi cellulari maturi. Esistono molti metodi statistici per stimare le traiettorie pseudo-temporali in dati singlecell. Il primo è stato proposto in uno studio del 2014 (Trapnell et al. (2014)), in cui viene presentato Monocle, un algoritmo non supervisionato che permette di ricostruire la traiettoria pseudo-temporale. Vengono riportati in maniera sintetica i passi dell'algoritmo.

- Il profilo di espressione di ogni cellula viene rappresentato da un punto in uno spazio euclideo ad alta dimensionalità, con una dimensione per ogni gene;
- Lo spazio ad alta dimensionalità viene ridotto tramite l'analisi delle componenti indipendenti (ICA)

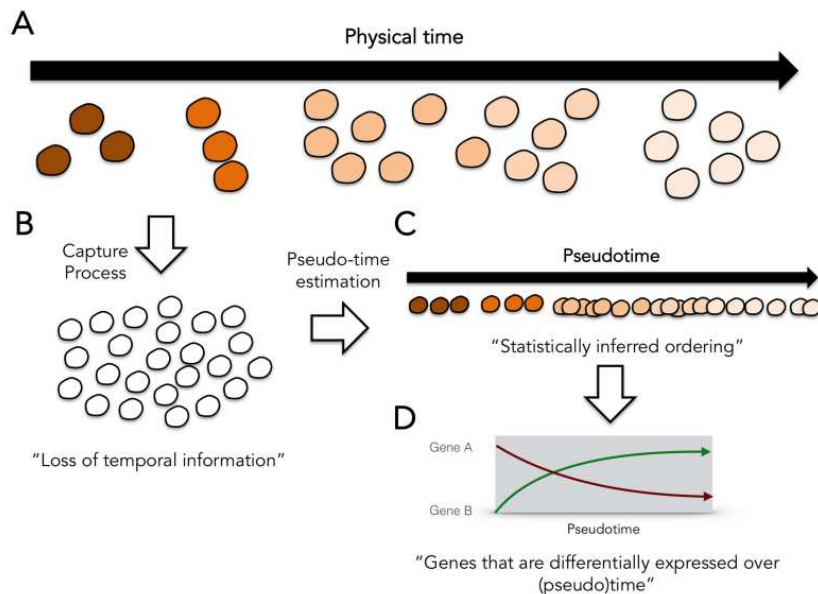


Figura 2.2: Passaggi dell'analisi pseudotemporale.

- Viene costruito un albero ricoprente minimo (*minimum spanning tree*, MST) tra le cellule. Nella teoria dei grafi, un albero è un particolare tipo di grafo non orientato all'interno del quale non possono esistere percorsi chiusi e per ogni coppia di nodi esiste un unico collegamento che li congiunge. Un albero ricoprente è un albero che contiene tutti i vertici del grafo e contiene soltanto un sottoinsieme degli archi, vale a dire solo quelli necessari per connettere tra loro tutti i vertici con uno e un solo cammino. L'albero ricoprente minimo è un albero ricoprente nel quale sommando i pesi degli archi si ottiene un valore minimo;
- Monocle trova il cammino più lungo attraverso il MST, corrispondente alla sequenza più lunga di cellule trascrizionalmente simili. L'algoritmo usa questa sequenza per produrre una traiettoria del progresso di una singola cellula attraverso la differenziazione.

Lo pseudotempo utilizzato in questa Tesi è stato calcolato grazie all'algoritmo Slingshot (Street et al. (2018)), che parte da una rappresentazione dei dati a bassa dimensionalità e da un risultato di clustering. In seguito vengono riportati i passaggi principali.

- Vengono identificate le linee di discendenza: si trattano i cluster come nodi di una rete, viene costruito l'MST tra i nodi, vengono definite le linee di discendenza come un insieme ordinato di nodi ed infine si sceglie in modo semi-supervisionato il nodo iniziale;
- Si traccia un cammino “liscio” attraverso le linee attraverso il metodo delle curve principali (Hastie and Stuetzle (1989)). Infine, si schiacciano le curve insieme nel cammino condiviso e si proiettano le cellule sulle curve principali per ottenere lo pseudotempo;

Il processo di maturazione delle cellule staminali descritto nel precedente sottoparagrafo è un classico caso in cui è applicabile l'analisi dello pseudotempo. Si vogliono infatti identificare e stimare le traiettorie attraverso cui le cellule passano dallo stato immaturo a maturo. Sono svariate le traiettorie in questo contesto biologico; in uno studio (Fletcher et al. (2017)) in cui è stato condotto l'esperimento, sono state identificate tre differenti traiettorie a partire dalle cellule HBC. In particolare, una traiettoria che porta ad avere cellule SUS, una che porta ad avere microvilli ed una, la più lunga, che porta a mOSN. È stato dimostrato che anche le cellule BG provengono dalle cellule HBC. Questa Tesi si concentra sull'ultima traiettoria, è di interesse quindi studiare il processo attraverso cui le cellule passano dallo stadio più immaturo (HBC) allo stadio più maturo (mOSN). Questa traiettoria viene chiamata *lineage* neuronale. Come si può facilmente notare dalla Figura 2.3 (Duggan and Ngai (2007)), la traiettoria neuronale consiste nel passaggio delle cellule da HBC a GBC, da GBC a iOSN e da iOSN a mOSN.

2.2 Analisi preliminari

Prima di procedere con le analisi preliminari è necessario chiarire alcuni passaggi di analisi che sono stati già eseguiti da alcuni ricercatori.

- È stato eseguito un controllo sulle cellule, eliminando quelle di scarsa qualità. Di conseguenza, in questa Tesi non è stato svolto il passaggio di pulizia delle cellule;

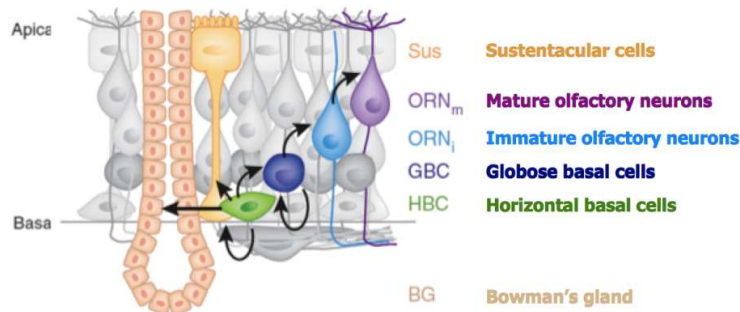


Figura 2.3: Differenti traiettorie cellulari.

- È stata eseguita un'analisi di cluster, in cui sono state etichettate le cellule come HBC, GBC, iOSN, mOSN o SUS;
- A partire da tale analisi di cluster e da una dimensionalità ridotta dei dati, è già stata eseguita un'analisi pseudotemporale in cui sono state individuate tre traiettorie: quella che viene presa in considerazione in questa Tesi è la più lunga, quella neuronale, che parte dalle cellule di tipo HBC per arrivare ai neuroni maturi.

Delle 20426 cellule originali sono state considerate solo le 6810 appartenenti alle *lineage* neuronale. In questo tipo di analisi non è raro etichettare come appartenenti ad una traiettoria delle cellule che in realtà sono estranee. Per questo motivo sono state eliminate 312 cellule di tipo SUS in quanto non sono di interesse per la *lineage* neuronale; sono state eliminate anche 85 cellule HBC dormienti. Da questi step di analisi ne risulta una matrice di conteggio di 14618 righe e 6413 colonne; le righe corrispondono ai geni, le colonne sono riferite alle cellule, con la rispettiva etichetta. Come è lecito aspettarsi da un contesto single-cell, i conteggi sono tendenzialmente bassi; infatti, solo 242 geni su 14618 presentano in media più di 5 reads. Alla luce di queste considerazioni, viene effettuato un filtraggio sui geni, ovvero vengono considerati solo i geni che presentano un numero sufficiente di reads. In questo caso, vengono tenuti in considerazione soli i geni che presentano almeno 5 reads in

almeno 5 cellule; così facendo i geni rimangono 5679 e la matrice di conteggio si riduce ad avere 5679 righe e 6413 colonne. Un altro passaggio importante da effettuare prima delle analisi è la normalizzazione. In contesto di dati a singola cellula, ed in generale nei dati di genomica, la normalizzazione serve a minimizzare gli errori sistematici e distorsioni relative alla tecnologia del sequenziamento, preservando la variabilità biologica. Teoricamente, il livello di espressione normalizzato di un gene non dovrebbe essere correlato con il numero di reads sequenziate per cellula (profondità di sequenziamento) e la varianza della distribuzione normalizzata di un gene lungo le cellule dovrebbe riflettere principalmente la variabilità biologica, indipendentemente dalla profondità di sequenziamento. È stato dimostrato da Ahlmann-Eltze and Huber (2022) che uno dei metodi più semplici ed efficaci per la normalizzazione di dati scRNA-Seq è usare i dati log-normalizzati. In altre parole, si calcola un fattore di normalizzazione per ogni cellula chiamato s_c , $c = 1, \dots, n$ dove n è il numero di cellule, e viene applicata ad ogni colonna la funzione

$$g(y_c) = \log_2 \left(\frac{y_c}{s_c} + 1 \right), \quad (2.1)$$

dove con y_c si rappresenta la c -esima colonna, ovvero la c -esima cellula. Esistono molti metodi per stimare s_i , uno dei più semplici è

$$s_c = \frac{\sum_c y_{gc}}{n^{-1} \sum_{gc} y_{gc}}, \quad (2.2)$$

dove il numeratore sta ad indicare la somma di reads tra tutti i geni per la cellula c , mentre il denominatore è la media dei numeratori tra tutte le cellule.

2.3 Analisi esplorative

Una volta eseguite le analisi preliminari descritte nel Paragrafo precedente, il risultato è una matrice di conteggio con 5679 geni sulle righe e 6413 cellule sulle colonne. I geni poco espressi sono stati filtrati ed è stata applicata la normalizzazione. Inoltre, le cellule sono etichettate in base al loro tipo cellulare e sono ordinate lungo lo pseudotempo. In particolare, su 6413 cellule il 30% sono etichettate come HBC, il 10% come GBC, il 44% come

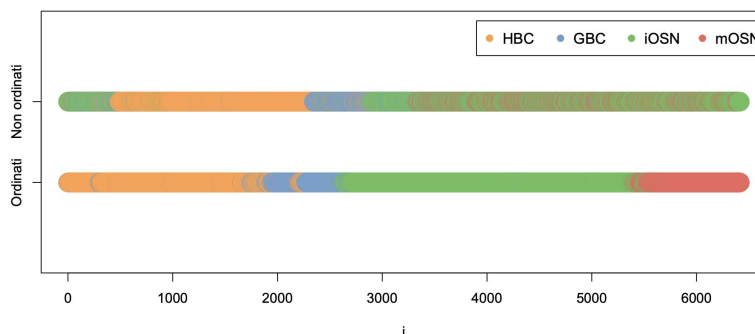


Figura 2.4: Ordinamento pseudotemporale stimato da Slingshot.

iOSN ed il 14% come mOSN; si nota quindi che quasi la metà delle cellule sono etichettate come neuroni immaturi. Il primo grafico esplorativo (Figura 2.4) mostra come l'algoritmo Slingshot ha ordinato le 6413 cellule lungo lo pseudotempo assegnando ad ogni cellula un punteggio che sta ad indicare la posizione relativa di ogni cellula lungo la traiettoria di riferimento. Come è facilmente visibile dalla legenda, ognuno dei quattro colori si riferisce ad un differente tipo cellulare (HBC, GBC, iOSN, mOSN), mentre sull'asse delle ordinate c'è un ipotetico indice i , $i = 1, \dots, 6413$ per ogni cellula. Come prima cosa si può notare che l'ordinamento stimato da Slingshot coincide con l'ordinamento biologico del processo di riferimento; infatti prima sono situate le cellule HBC che fungono da riserva, poi le GBC che sono cellule staminali progenitrici dei neuroni, poi i neuroni immaturi ed infine i neuroni maturi. Si nota inoltre come, una volta applicato l'algoritmo, le cellule ordinate lungo la traiettoria sono più raggruppate per ogni tipo cellulare rispetto alle cellule non ordinate. I metodi di riduzione della dimensionalità sono molto usati in contesti single-cell, dove in presenza di spazi ad alta dimensionalità è complicata anche la visualizzazione. Tra i metodi più semplici e più usati a questo scopo c'è la PCA (*Principal component analysis*, analisi delle componenti principali). Viene riportato in Figura 2.5 un grafico delle prime due componenti principali, dove ogni punto dello spazio rappresenta ognuna delle 6413 cellule ed ogni colore corrisponde ad un tipo cellulare differente. La curva nera è la traiettoria stimata dall'algoritmo Slingshot a partire dalla PCA,

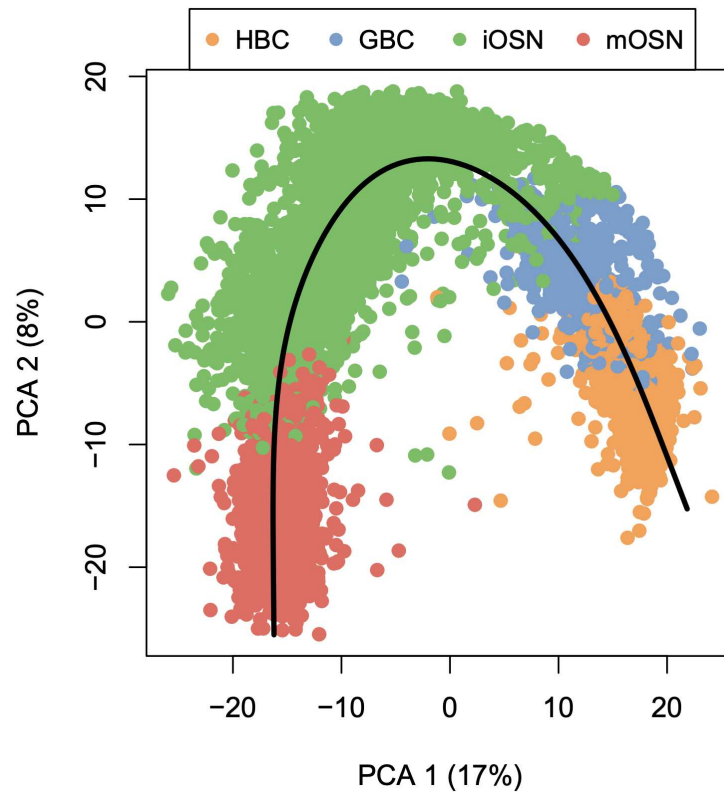


Figura 2.5: PCA colorata per i quattro tipi cellulari.

dalle 6413 cellule considerate e dalle loro etichette già presenti. Come si può facilmente notare, i quattro tipi cellulari sono molto ben distinti tra loro e si vede inoltre a livello di cluster di tipi cellulari l'andamento di interesse biologico, ovvero il passaggio delle cellule da HBC a GBC, da GBC a iOSN e da iOSN a mOSN. Tale aspetto viene sottolineato dalla curva nera, che parte dal cluster di cellule HBC per arrivare al cluster di cellule mOSN. In Figura 2.6 viene riportato lo stesso grafico dove le 6413 cellule sono colorate in base allo pseudotempo. Come si vede facilmente dalla legenda, i colori più scuri (viola, blu) corrispondono a pseudotempi bassi, vale a dire a cellule relativamente immature lungo la traiettoria. Viceversa i colori più chiari (verde, giallo) corrispondono a pseudotempi alti, vale a dire cellule relativamente mature o quasi lungo la traiettoria. Si nota un'interessante corrispondenza col grafico mostrato in precedenza: infatti, le cellule più immature (HBC, GBC) che

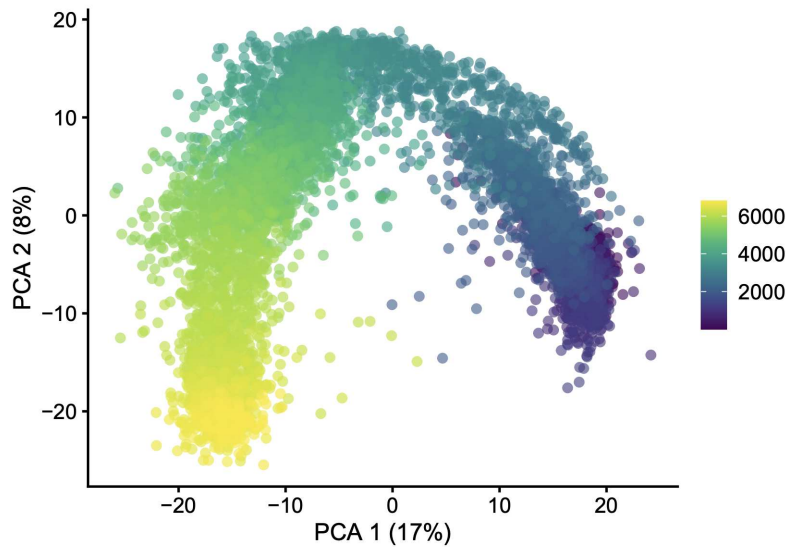


Figura 2.6: PCA colorata per lo pseudotempo.

sono situate alla destra del grafico hanno colori più scuri, viceversa le cellule più mature (iOSN, mOSN) che sono alla sinistra del grafico hanno colori più chiari. Viene effettuato lo stesso tipo di analisi esplorativa ma usando come metodo di riduzione della dimensionalità il t-SNE anziché la PCA. Il t-SNE (*t-distributed stochastic neighbor embedding*) è un metodo non lineare di riduzione della dimensionalità, utile in contesto di dati a singola cellula per la visualizzazione dei dati. Lo scopo del t-SNE è trovare una rappresentazione dei dati a dimensione ridotta (2 o 3 dimensioni) preservando la struttura di similarità delle osservazioni (nel caso in esame, le cellule) presente nello spazio originale ad alta dimensionalità. L'elemento stocastico è dato dal fatto che la distanza tra punti ha interpretazione probabilistica. Nonostante sia molto utilizzato anche in contesti diversi dal single-cell, il t-SNE presenta anche dei problemi, come il fatto che non esiste una semplice interpretazione dei fattori, a contrario della PCA. Inoltre, il t-SNE preserva la struttura locale dei dati ma non quella globale; di conseguenza, risulta utile per trovare dei cluster, ma non è interpretabile la loro relazione. In Figura 2.7 viene riportato il t-SNE dei dati in cui le cellule sono colorate per tipo cellulare. Anche in questo caso la curva nera è la traiettoria stimata dall'algoritmo

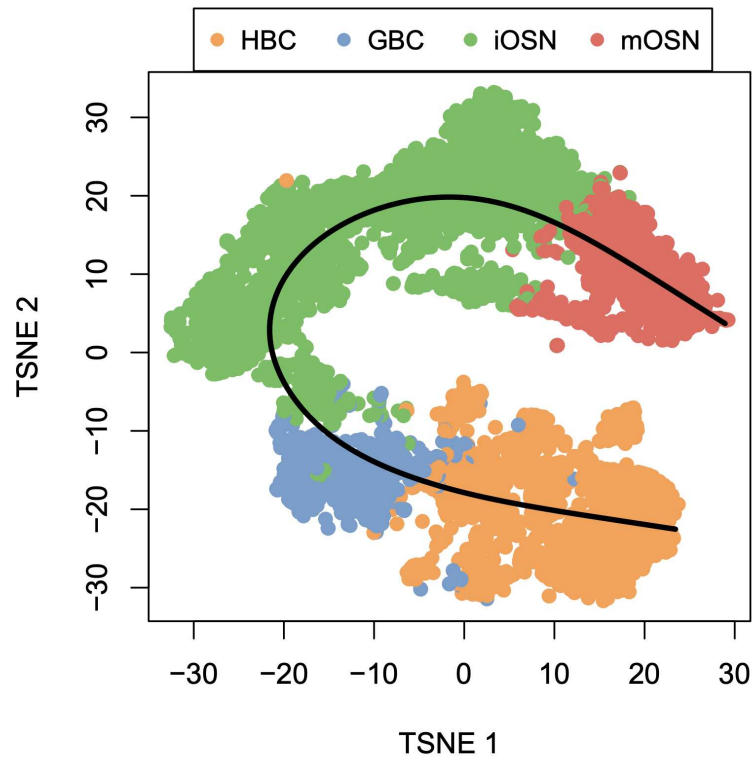


Figura 2.7: t-SNE colorato per i quattro tipi cellulari.

Slingshot a partire dal t-SNE, dalle 6413 cellule considerate e dalle loro etichette già presenti. Si vedono molto bene i quattro cluster distinti e si può notare un simile andamento ordinato tra cluster, anche se la loro relazione è meno chiara rispetto alla PCA. Il grafico in Figura 2.8 mostra il t-SNE in cui le cellule sono colorate dallo pseudotempo. Si può vedere anche qui come le cellule relativamente più immature (HBC, GBC) che sono situate in basso hanno pseudotempi minori mentre le cellule relativamente più mature (iOSN, mOSN) che sono situate in alto hanno pseudotempi maggiori.

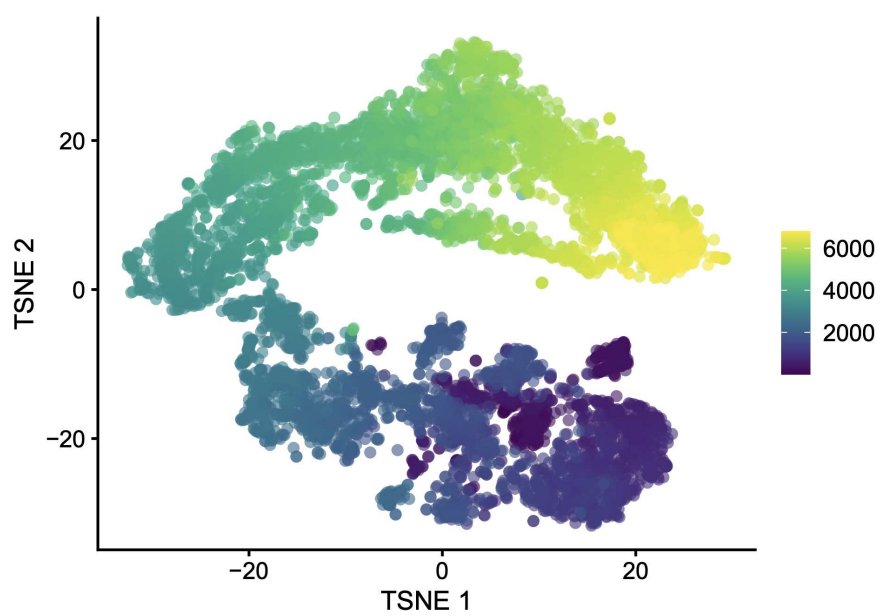


Figura 2.8: t-SNE colorato per lo pseudotempo.

Capitolo 3

Modelli Grafici

In questo Capitolo viene affrontata la fase di modellazione delle reti geniche, con particolare attenzione agli algoritmi di stima.

3.1 Introduzione al problema

Come spiegato nel Capitolo 1, l'espressione genica è un complesso processo grazie al quale le proteine sono sintetizzate per svolgere le funzioni della cellula. Sulla base di una specifica funzione biologica, infatti, uno o più geni vengono espressi in quantità sufficiente per il suo svolgimento. Un aspetto molto interessante riguardo il modo in cui avviene questo processo è che l'espressione di due o più geni può non essere indipendente tra loro. Un esempio può essere l'epistasi, che in forma molto sintetica è definita come una forma di interazione fra geni in cui una mutazione genetica dipende dalla presenza o dall'assenza di mutazioni in uno o più altri geni. Diventa quindi di grande interesse biologico studiare e stimare la struttura di dipendenza che esiste tra l'espressione di diversi geni presenti in un genoma; uno dei possibili modi per rappresentarla è attraverso le reti di regolazione genica (GRNs, *gene regulatory networks*). Le GRNs sono reti biologiche in cui i geni sono rappresentati dai nodi mentre gli archi sono le connessioni tra geni. In seguito verranno riportati alcuni possibili modelli usati per fare inferenza sulle GRNs. Per ragioni strettamente computazionali, i modelli affrontati in questo Capitolo vengono adattati a partire da un sottoinsieme di geni ricavato a partire dal-

le analisi preliminari. Per comprendere al meglio il processo di selezione dei geni è importante introdurre il concetto di *biological pathway*. Per *pathway* si intende un gruppo di geni organizzati in un certo ordine allo scopo di eseguire una determinata funzione biologica. In essi, vengono illustrate le singole interazioni biologiche tra geni. Uno dei database più utilizzati per l'annotazione dei *pathway* è la Kyoto Encyclopedia of Genes and Genomes (KEGG), un dettagliato database che aiuta a capire il funzionamento di molti sistemi biologici. Nel già citato studio (Fletcher et al. (2017)) viene scoperto che l'attivazione di un *pathway* chiamato Wnt Signaling è necessario e sufficiente per attivare il processo di differenziazione delle cellule HBC del topo; il Wnt Signaling Pathway è presente in KEGG ed è formato da 172 geni. Quello che è stato fatto è trovare i geni comuni tra tali 172 geni e i 5679 già filtrati del Capitolo precedente: in questo modo, si è ottenuta una lista di 53 geni alla quale verranno applicati i modelli grafici.

3.2 Modelli grafici Gaussiani

Un modello grafico, o modello grafico probabilistico (PMG, *Probabilistic Graphical models*) è un modello statistico in cui la struttura di indipendenza condizionata di un insieme di variabili aleatorie è rappresentata da un grafo. In particolare, in un modello grafico i nodi rappresentano le variabili aleatorie e gli archi descrivono la relazione di indipendenza condizionata tra nodi associati.

Un grafo $G = (V, E)$ è una coppia di insiemi disgiunti: V è un insieme non vuoto, discreto, finito, i cui elementi sono chiamati vertici del grafo, mentre E , detto insieme degli archi del grafo, è un sottoinsieme del prodotto cartesiano $V \times V$, ossia un insieme di coppie di vertici del grafo. Un grafo può essere orientato, se gli archi sono considerati come coppie ordinate di vertici, in tal caso vale $(u, v) \neq (v, u)$, oppure non orientato, se gli archi sono considerati come coppie non ordinate, ovvero $(u, v) = (v, u)$.

Due variabili aleatorie X ed Y si dicono condizionatamente indipendenti data una variabile aleatoria Z , e si scrive $X \perp Y|Z$, se $P(X|Y, Z) = P(X|Z)$.

Un grafo G ed un vettore aleatorio \mathbf{X} possono o meno essere in qualche tipo di relazione; il concetto di fattorizzazione spiega la densità congiunta di \mathbf{X}

tramite un grafo G . Sia $G = (V, E)$ un grafo ordinato ed $\mathbf{X} = (\mathbf{X}_1, \dots, \mathbf{X}_p)$ un vettore aleatorio. Nella teoria dei grafi, una cricca (*clique*) $C \subseteq V$ di un grafo è un sottoinsieme dei vertici di un grafo orientato, completamente connesso, vale a dire $(s, t) \in E \quad \forall (s, t) \in C$. Viene definito \mathcal{S} l'insieme di tutte le possibili cricche del grafo. Si dice che \mathbf{X} fattorizza rispetto al grafo G se esistono delle funzioni $\psi_C \geq 0$ dette *funzioni di compatibilità* tali che

$$p(x_1, \dots, x_p) = \frac{1}{Z} \prod_{C \in \mathcal{S}} \psi_C(x_C), \quad (3.1)$$

dove $Z = \int \prod_{C \in \mathcal{S}} \psi_C(x_C) dx_1, \dots, dx_d$ è detta funzione di partizione.

Sia $\mathbf{X} = (\mathbf{X}_1, \dots, \mathbf{X}_n)$ un vettore di variabili aleatorie con distribuzione P e $G = (V, E)$ un grafo non orientato. Allora la distribuzione di probabilità P soddisfa la *Global Markov Property* rispetto al grafo G se per ogni A, B e C sottoinsiemi disgiunti di V tale che C separa A e B , vale

$$X_A \perp X_B | X_C. \quad (3.2)$$

Se viene soddisfatta tale proprietà si dice che il vettore casuale \mathbf{X} soddisfa la proprietà di Markovianità rispetto al grafo G .

Sia $\mathbf{X} = (\mathbf{X}_1, \dots, \mathbf{X}_n)$ un vettore di variabili aleatorie con distribuzione P e $G = (V, E)$ un grafo non orientato. Allora la distribuzione di probabilità P soddisfa la *Pairwise Markov Property* rispetto al grafo G se, per ogni coppia di nodi $(i, j) \in V$

$$\mathbf{X}_i \perp \mathbf{X}_j | \mathbf{X}_{V \setminus \{i, j\}} \iff (i, j) \notin E, \quad (3.3)$$

vale a dire che \mathbf{X}_i ed \mathbf{X}_j sono condizionatamente indipendenti se e solo se è assente l'arco (i, j) che collega \mathbf{X}_i con \mathbf{X}_j .

La *Global Markov Property* implica la *Pairwise Markov Property*.

Il Teorema di Hammersley–Clifford enuncia il seguente risultato.

Sia $G = (V, E)$ un grafo ed $\mathbf{X} = (\mathbf{X}_1, \dots, \mathbf{X}_n)$ un vettore di variabili casuali. Sia $p(\mathbf{X})$ la densità congiunta di \mathbf{X} . Se $p(\mathbf{X}) > 0$, allora \mathbf{X} fattorizza secondo il grafo G se e solo se \mathbf{X} soddisfa la proprietà di Markovianità rispetto al grafo G .

Finora non si è fatto cenno alla distribuzione di \mathbf{X} ; nei modelli grafici Gaussiani si prende in considerazione il caso in cui \mathbf{X} assume distribuzione normale multivariata. Sia \mathbf{X} un vettore aleatorio p -dimensionale ed \mathbf{x} realizzazione di

\mathbf{X} . \mathbf{X} ha distribuzione normale multivariata se, indicata con p la sua funzione di densità, è

$$p(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) = (2\pi)^{-p/2} |\boldsymbol{\Sigma}|^{-1/2} \exp\left\{-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\mathbf{x} - \boldsymbol{\mu})\right\}, \quad (3.4)$$

e si scrive $\mathbf{X} \sim N_p(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, con $E[\mathbf{X}] = \boldsymbol{\mu} \in \mathbb{R}^p$ e $V[\mathbf{X}] = \boldsymbol{\Sigma} \in \mathbb{S}_{++}^p$ matrice simmetrica definita positiva.

Ponendo $\boldsymbol{\Theta} = \boldsymbol{\Sigma}^{-1}$ si ha

$$p(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Theta}) = (2\pi)^{-p/2} |\boldsymbol{\Theta}|^{1/2} \exp\left\{-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^\top \boldsymbol{\Theta}(\mathbf{x} - \boldsymbol{\mu})\right\}. \quad (3.5)$$

Sia $\mathbf{X} = (\mathbf{X}_1, \dots, \mathbf{X}_n)$ un vettore aleatorio con distribuzione p e $G = (V, E)$ un grafo non orientato. Si dice che il vettore aleatorio \mathbf{X} soddisfa il modello grafico Gaussiano con grafo G se $\mathbf{X} \sim N_p(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ dove $\boldsymbol{\Theta}_{ij} = 0 \quad \forall (i, j) \notin E$. In altre parole, in un modello grafico Gaussiano la presenza di zeri nella matrice di precisione del vettore aleatorio associato corrisponde ad archi mancanti nel corrispondente grafo.

Si noti che la distribuzione di una normale multivariata si può scrivere come

$$\begin{aligned} p(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Theta}) &= (2\pi)^{-p/2} |\boldsymbol{\Theta}|^{1/2} \exp\left\{-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^\top \boldsymbol{\Theta}(\mathbf{x} - \boldsymbol{\mu})\right\} = \\ &= (2\pi)^{-p/2} |\boldsymbol{\Theta}|^{1/2} \exp\left\{-\frac{1}{2} \sum_{i,j} (\mathbf{x}_i - \boldsymbol{\mu}_i)^\top \boldsymbol{\Theta}_{ij} (\mathbf{x}_j - \boldsymbol{\mu}_j)\right\} = \\ &= (2\pi)^{-p/2} |\boldsymbol{\Theta}|^{1/2} \prod_{i,j} \exp\left\{-\frac{1}{2} (\mathbf{x}_i - \boldsymbol{\mu}_i)^\top \boldsymbol{\Theta}_{ij} (\mathbf{x}_j - \boldsymbol{\mu}_j)\right\}. \end{aligned} \quad (3.6)$$

Dunque $p(\mathbf{x})$ si può rappresentare come $p(\mathbf{x}) = h_{ij}(\mathbf{x}_i, \mathbf{x}_j)$ se $\boldsymbol{\Theta}_{ij} \neq 0$ e quindi fattorizza rispetto ad un grafo $G = (V, E)$. Dal momento che $p(\mathbf{X}) > 0$ si può usare il teorema di Hammersley–Clifford e dire che la distribuzione di probabilità p soddisfa la *Global Markov Property* rispetto al grafo G . Come spiegato precedentemente, se vale la *Global Markov Property* vale anche la *Pairwise Markov Property* e si può scrivere dunque

$$\boldsymbol{\Theta}_{ij} = 0 \iff (i, j) \notin E \iff \mathbf{X}_i \perp \mathbf{X}_j | \mathbf{X}_{V \setminus \{i, j\}}. \quad (3.7)$$

In altre parole, in un modello grafico Gaussiano la presenza di zeri nella matrice di precisione del vettore aleatorio associato corrisponde ad archi mancanti nel corrispondente grafo, che a sua volta significa indipendenza condizionata tra le corrispondenti variabili.

3.3 Graphical-LASSO

3.3.1 Introduzione alla regressione penalizzata

Il più comune problema di ottimizzazione nel campo della regressione è il problema ai minimi quadrati, dove per trovare il β ottimo si cerca il β che minimizza la somma dei quadrati dei residui:

$$\operatorname{argmin}_{\beta \in \mathbb{R}^p} \|\mathbf{y} - \mathbf{X}\beta\|_2^2, \quad (3.8)$$

dove $\mathbf{X} \in \mathbb{R}^{n \times p}$ è la matrice del disegno, $\mathbf{y} \in \mathbb{R}^n$, $\beta \in \mathbb{R}^p$. Tale problema ha soluzione esplicita per β , ossia lo stimatore dei minimi quadrati:

$$\hat{\beta} = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{y}. \quad (3.9)$$

Questo metodo richiede che $p < n$ affinché abbia soluzione unica. Per questo motivo sono stati sviluppati alcuni metodi di regolarizzazione in cui si aggiunge al problema appena descritto una penalità ai coefficienti β , al fine di rendere il modello più flessibile.

Uno dei più importanti metodi di regolarizzazione è la regressione RIDGE, proposta per la prima volta in Hoerl and Kennard (1970) che aggiunge una penalità quadratica ai coefficienti, risolvendo il problema di minimo

$$\min_{\beta} \|\mathbf{y} - \mathbf{X}\beta\|_2^2 \quad \text{sotto il vincolo} \quad \sum_j \beta_j^2 < s, \quad (3.10)$$

o equivalentemente in forma Lagrangiana

$$\min_{\beta} \|\mathbf{y} - \mathbf{X}\beta\|_2^2 + \lambda \|\beta\|_2^2, \quad (3.11)$$

dove $\lambda \in \mathbb{R}^+$ è detto parametro di regolazione o di penalizzazione. Se $\lambda = 0$ il problema ritorna ad essere quello dei minimi quadrati ordinari. Tra i vari vantaggi della regressione ridge, si ha il fatto che la soluzione è esplicita:

$$\hat{\beta}_{ridge} = (\mathbf{X}^\top \mathbf{X} + \lambda \mathbf{I}_p)^{-1} \mathbf{X}^\top \mathbf{y}. \quad (3.12)$$

È inoltre utile nel caso di collinearità tra variabili esplicative ed imponendo una penalità a β si evita il sovradattamento della stima.

Un altro tipo di regressione penalizzata è la regressione LASSO (*least absolute shrinkage and selection operator*), proposta per la prima volta da Tibshirani (1996), che risolve il problema di minimo

$$\min_{\boldsymbol{\beta}} \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 \quad \text{sotto il vincolo} \quad \sum_j |\beta_j| < s, \quad (3.13)$$

o equivalentemente in forma Lagrangiana

$$\min_{\boldsymbol{\beta}} \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \lambda \|\boldsymbol{\beta}\|_1. \quad (3.14)$$

Come si vede, rispetto alla regressione RIDGE cambia il tipo di penalità imposta a $\boldsymbol{\beta}$.

La regressione LASSO non ha una soluzione esplicita, di conseguenza è necessario ricorrere ad alcuni algoritmi per risolvere il problema di ottimo. Cionostante, ha il vantaggio di stimare alcuni parametri esattamente a 0, mimando dunque un continuo processo di selezione delle variabili.

3.3.2 Graphical-LASSO: problema ed applicazione

Sia \mathbf{X} un vettore aleatorio p -dimensionale e \mathbf{x} una realizzazione di \mathbf{X} . Come spiegato precedentemente, \mathbf{X} ha distribuzione normale multivariata con media $\boldsymbol{\mu} \in \mathbb{R}^p$ e matrice di varianze e covarianze $\boldsymbol{\Sigma} \in \mathbb{S}_{++}^p$ se, indicata con f la sua funzione di densità, è

$$f(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) = (2\pi)^{-p/2} |\boldsymbol{\Sigma}|^{-1/2} \exp\left\{-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\mathbf{x} - \boldsymbol{\mu})\right\}, \quad (3.15)$$

e si scrive $\mathbf{X} \sim N_p(\boldsymbol{\mu}, \boldsymbol{\Sigma})$.

Posto $\boldsymbol{\theta} = (\boldsymbol{\mu}, \boldsymbol{\Sigma})$ e date $\mathbf{x}_1, \dots, \mathbf{x}_n$ osservazioni indipendenti ed identicamente distribuite da una $N_p(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, la funzione di verosimiglianza per $\boldsymbol{\theta}$ è

$$\begin{aligned} L(\boldsymbol{\theta}) &= \prod_{i=1}^n (2\pi)^{-p/2} |\boldsymbol{\Sigma}|^{-1/2} \exp\left\{-\frac{1}{2}(\mathbf{x}_i - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\mathbf{x}_i - \boldsymbol{\mu})\right\} \\ &\propto |\boldsymbol{\Sigma}|^{-n/2} \exp\left\{-\frac{1}{2} \sum_{i=1}^n (\mathbf{x}_i - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\mathbf{x}_i - \boldsymbol{\mu})\right\}. \end{aligned} \quad (3.16)$$

Lo stimatore di massima verosimiglianza di $\boldsymbol{\theta} = (\boldsymbol{\mu}, \boldsymbol{\Sigma})$ è $\widehat{\boldsymbol{\theta}} = (\widehat{\boldsymbol{\mu}}, \widehat{\boldsymbol{\Sigma}})$ dove

$$\widehat{\boldsymbol{\mu}} = \frac{1}{n} \sum_{i=1}^n \mathbf{x}_i \text{ e}$$

$$\widehat{\boldsymbol{\Sigma}} = \frac{1}{n} \sum_{i=1}^n (\mathbf{x}_i - \widehat{\boldsymbol{\mu}})(\mathbf{x}_i - \widehat{\boldsymbol{\mu}})^\top, \quad (3.17)$$

quindi la matrice di precisione può essere stimata con $\widehat{\boldsymbol{\Theta}} = \widehat{\boldsymbol{\Sigma}}^{-1}$. Cionostante, questa soluzione presenta dei problemi. Per prima cosa $\boldsymbol{\Sigma}$ ha dimensioni $p \times p$ ed essendo simmetrica contiene $\frac{p(p-1)}{2}$ parametri di varianza e covarianza da stimare, il che rende $\widehat{\boldsymbol{\Sigma}}$ uno stimatore poco stabile quando p è grande. Inoltre, tale stimatore non riesce a catturare l'eventuale sparsità della matrice. Un altro problema è rappresentato dal fatto che se $p > n$, allora non esiste $\widehat{\boldsymbol{\Sigma}}^{-1}$ in quanto $\widehat{\boldsymbol{\Sigma}} \in \mathbb{R}^{p \times p}$ ha rango massimo n .

Il graphical-LASSO, proposto per primo in Friedman et al. (2007) è un metodo penalizzato per ottenere la stima della matrice di precisione sparsa di una distribuzione normale multivariata; viene introdotto per risolvere il problema della stabilità dello stimatore di massima verosimiglianza e per aggiungere l'informazione dell'eventuale sparsità alla stima.

Per arrivare alla formulazione finale del problema occorrono alcuni passaggi algebrici. Come descritto precedentemente, la funzione di verosimiglianza di una normale multivariata con parametri $\boldsymbol{\theta} = (\boldsymbol{\mu}, \boldsymbol{\Sigma})$ è esprimibile come

$$L(\boldsymbol{\theta}) = |\boldsymbol{\Sigma}|^{-n/2} \exp \left\{ -\frac{1}{2} \sum_{i=1}^n (\mathbf{x}_i - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1} (\mathbf{x}_i - \boldsymbol{\mu}) \right\}. \quad (3.18)$$

Detta $\boldsymbol{\Theta} = \boldsymbol{\Sigma}^{-1}$ la matrice di precisione e riscrivendo la verosimiglianza per $(\boldsymbol{\mu}, \boldsymbol{\Theta})$ si ottiene

$$L(\boldsymbol{\mu}, \boldsymbol{\Theta}) = |\boldsymbol{\Theta}|^{n/2} \exp \left\{ -\frac{1}{2} \sum_{i=1}^n (\mathbf{x}_i - \boldsymbol{\mu})^\top \boldsymbol{\Theta} (\mathbf{x}_i - \boldsymbol{\mu}) \right\}. \quad (3.19)$$

Quindi la log-verosimiglianza è

$$\begin{aligned} l(\boldsymbol{\mu}, \boldsymbol{\Theta}) &= \frac{n}{2} \log |\boldsymbol{\Theta}| - \frac{1}{2} \sum_{i=1}^n (\mathbf{x}_i - \boldsymbol{\mu})^\top \boldsymbol{\Theta} (\mathbf{x}_i - \boldsymbol{\mu}) \propto \\ &\propto \log |\boldsymbol{\Theta}| - \frac{1}{n} \sum_{i=1}^n (\mathbf{x}_i - \boldsymbol{\mu})^\top \boldsymbol{\Theta} (\mathbf{x}_i - \boldsymbol{\mu}). \end{aligned} \quad (3.20)$$

Dal momento che è di interesse la stima di Θ e non particolarmente quella di μ , si massimizza parzialmente la log-verosimiglianza rispetto a μ , sostituendo il rispettivo stimatore di massima verosimiglianza $\hat{\mu}$, ottenendo

$$l(\Theta) = \log |\Theta| - \frac{1}{n} \sum_{i=1}^n (\mathbf{x}_i - \hat{\mu})^\top \Theta (\mathbf{x}_i - \hat{\mu}). \quad (3.21)$$

Usando la proprietà della traccia

$$\mathbf{x}^\top \mathbf{A} \mathbf{x} = \text{tr}(\mathbf{A} \mathbf{x} \mathbf{x}^\top), \quad \mathbf{x} \in \mathbb{R}^d, \mathbf{A} \in \mathbb{R}^{d \times d}, \quad (3.22)$$

e sfruttando il fatto che la traccia è operatore lineare, si ottiene

$$\begin{aligned} l(\Theta) &= \log |\Theta| - \frac{1}{n} \sum_{i=1}^n \text{tr}[\Theta (\mathbf{x}_i - \hat{\mu})(\mathbf{x}_i - \hat{\mu})^\top] = \\ &= \log |\Theta| - \text{tr}\left\{\frac{1}{n} \sum_{i=1}^n (\mathbf{x}_i - \hat{\mu})(\mathbf{x}_i - \hat{\mu})^\top \Theta\right\} = \log |\Theta| - \text{tr}(\mathbf{S}\Theta), \end{aligned} \quad (3.23)$$

indicando \mathbf{S} lo stimatore di massima verosimiglianza di Σ . Si ottiene dunque

$$l(\Theta) = \log |\Theta| - \text{tr}(\mathbf{S}\Theta) \quad (3.24)$$

Anzichè trovare il Θ che massimizza la log-verosimiglianza, il metodo graphical-LASSO propone di trovare il Θ che massimizza la log-verosimiglianza penalizzata, applicando una penalità di tipo L_1 a Θ . La formulazione del problema graphical-LASSO è quindi

$$\hat{\Theta} = \underset{\Theta}{\text{argmax}} \ l(\Theta) - \lambda \|\Theta\|_1, \quad (3.25)$$

o equivalentemente

$$\hat{\Theta} = \underset{\Theta}{\text{argmax}} \ \log |\Theta| - \text{tr}(\mathbf{S}\Theta) - \lambda \|\Theta\|_1. \quad (3.26)$$

Il parametro di penalità λ può essere scelto in molti modi; in questo Capitolo, viene scelto il λ che minimizza l'errore di convalida incrociata.

Tra i vari metodi possibili per risolvere il problema del graphical-LASSO, uno dei più utilizzati è il metodo *Block Coordinate Descent*.

L'espressione di partenza è la (3.26) e la corrispondente equazione del subgradiente è:

$$\frac{\partial}{\partial \Theta} \left\{ \log |\Theta| - \text{tr}(\mathbf{S}\Theta) - \lambda \|\Theta\|_1 \right\} = \mathbf{W} - \mathbf{S} - \lambda \mathbf{\Gamma} = 0, \quad (3.27)$$

dove $\mathbf{W} = \Theta^{-1}$ e la matrice simmetrica $\mathbf{\Gamma}$ ha elementi, $\Gamma_{ij} = \text{sign}(\Theta_{ij})$ se $\Theta_{ij} \neq 0$, mentre $\Gamma_{ij} \in [-1, 1]$ se $\Theta_{ij} = 0$. Viene effettuata una partizione a blocchi per le matrici simmetriche \mathbf{W} , Θ , \mathbf{S} e $\mathbf{\Gamma}$ in modo tale che il blocco in alto a sinistra comprenda tutti gli elementi della rispettiva matrice tranne l'ultima riga e l'ultima colonna:

$$\begin{aligned} \mathbf{W} &= \begin{bmatrix} \mathbf{W}_{11} & \mathbf{w}_{12} \\ \mathbf{w}_{12}^\top & w_{22} \end{bmatrix}, & \mathbf{S} &= \begin{bmatrix} \mathbf{S}_{11} & \mathbf{s}_{12} \\ \mathbf{s}_{12}^\top & s_{22} \end{bmatrix}, \\ \Theta &= \begin{bmatrix} \Theta_{11} & \boldsymbol{\theta}_{12} \\ \boldsymbol{\theta}_{12}^\top & \theta_{22} \end{bmatrix}, & \mathbf{\Gamma} &= \begin{bmatrix} \mathbf{\Gamma}_{11} & \boldsymbol{\gamma}_{12} \\ \boldsymbol{\gamma}_{12}^\top & \gamma_{22} \end{bmatrix}. \end{aligned} \quad (3.28)$$

Tutte e quattro le matrici sono simmetriche di dimensioni $p \times p$; di conseguenza \mathbf{W}_{11} , \mathbf{S}_{11} , Θ_{11} , $\mathbf{\Gamma}_{11}$ sono matrici di dimensioni $(p-1) \times (p-1)$. Si noti inoltre che \mathbf{w}_{12} , \mathbf{s}_{12} , $\boldsymbol{\theta}_{12}$, $\boldsymbol{\gamma}_{12}$ sono vettori di dimensione $p-1$ mentre w_{22} , s_{22} , θ_{22} , γ_{22} sono scalari. Alla luce di questa notazione, in termini matriciali l'equazione del subgradiente può esser riscritta come:

$$\begin{bmatrix} \mathbf{W}_{11} & \mathbf{w}_{12} \\ \mathbf{w}_{12}^\top & w_{22} \end{bmatrix} - \begin{bmatrix} \mathbf{S}_{11} & \mathbf{s}_{12} \\ \mathbf{s}_{12}^\top & s_{22} \end{bmatrix} - \lambda \begin{bmatrix} \mathbf{\Gamma}_{11} & \boldsymbol{\gamma}_{12} \\ \boldsymbol{\gamma}_{12}^\top & \gamma_{22} \end{bmatrix} = \begin{bmatrix} \mathbf{0}_{(p-1) \times (p-1)} & \mathbf{0}_{(p-1) \times 1} \\ \mathbf{0}_{1 \times (p-1)} & 0 \end{bmatrix} \quad (3.29)$$

Sviluppando la (3.29) si ottiene il sistema di equazioni

$$\begin{cases} \mathbf{W}_{11} - \mathbf{S}_{11} - \lambda \mathbf{\Gamma}_{11} = \mathbf{0}_{(p-1) \times (p-1)} \\ \mathbf{w}_{12} - \mathbf{s}_{12} - \lambda \boldsymbol{\gamma}_{12} = \mathbf{0}_{(p-1) \times 1} \\ \mathbf{w}_{12}^\top - \mathbf{s}_{12}^\top - \lambda \boldsymbol{\gamma}_{12}^\top = \mathbf{0}_{1 \times (p-1)} \\ w_{22} - s_{22} - \lambda \gamma_{22} = 0 \end{cases} \quad (3.30)$$

Vista la sua definizione, γ_{22} è l'elemento in posizione p, p di $\mathbf{\Gamma}$, ovvero $\Gamma_{p,p}$. Per definirne il valore è necessario studiare l'eventuale nullità del corrispondente elemento di Θ , vale a dire $\Theta_{p,p}$: trovandosi nella diagonale di una matrice di

precisione, $\Theta_{p,p}$ è un valore strettamente positivo e quindi diverso da 0. Alla luce di ciò, data la definizione della matrice Γ si può scrivere

$$\gamma_{22} = \Gamma_{p,p} = \text{sign}(\Theta_{p,p}) = 1, \quad (3.31)$$

ed il sistema (3.30) diventa

$$\begin{cases} \mathbf{W}_{11} - \mathbf{S}_{11} - \lambda\Gamma_{11} = \mathbf{0}_{(p-1)\times(p-1)} \\ \mathbf{w}_{12} - \mathbf{s}_{12} - \lambda\gamma_{12} = \mathbf{0}_{(p-1)\times 1} \\ \mathbf{w}_{12}^\top - \mathbf{s}_{12}^\top - \lambda\gamma_{12}^\top = \mathbf{0}_{1\times(p-1)} \\ w_{22} - s_{22} - \lambda = 0 \end{cases} \quad (3.32)$$

Dal momento che $\mathbf{W} = \Theta^{-1}$ allora si può scrivere

$$\mathbf{W}\Theta = \mathbf{I}_p = \begin{bmatrix} \mathbf{I}_{(p-1)} & \mathbf{0}_{(p-1)\times 1} \\ \mathbf{0}_{1\times(p-1)} & 1 \end{bmatrix}, \quad (3.33)$$

quindi

$$\begin{bmatrix} \mathbf{W}_{11} & \mathbf{w}_{12} \\ \mathbf{w}_{12}^\top & w_{22} \end{bmatrix} \begin{bmatrix} \Theta_{11} & \theta_{12} \\ \theta_{12}^\top & \theta_{22} \end{bmatrix} = \begin{bmatrix} \mathbf{I}_{p-1} & \mathbf{0}_{(p-1)\times 1} \\ \mathbf{0}_{1\times(p-1)} & 1 \end{bmatrix} \quad (3.34)$$

Sviluppando si ottiene il sistema di equazioni

$$\begin{cases} \mathbf{W}_{11}\Theta_{11} + \mathbf{w}_{12}\theta_{12}^\top = \mathbf{I}_{p-1} \\ \mathbf{W}_{11}\theta_{12} + \mathbf{w}_{12}\theta_{22} = \mathbf{0}_{(p-1)\times 1} \\ \mathbf{w}_{12}^\top\Theta_{11} + w_{22}\theta_{12}^\top = \mathbf{0}_{1\times(p-1)} \\ \mathbf{w}_{12}^\top\theta_{12} + w_{22}\theta_{22} = 1 \end{cases} \quad (3.35)$$

risolvendo la seconda equazione di (3.35) per \mathbf{w}_{12} si ottiene

$$\mathbf{w}_{12} = -\mathbf{W}_{11}\theta_{12}\theta_{22}^{-1} \quad (3.36)$$

ed indicando $\boldsymbol{\beta} = -\theta_{12}\theta_{22}^{-1}$ si può scrivere $\mathbf{w}_{12} = \mathbf{W}_{11}\boldsymbol{\beta}$. In tal modo, la seconda equazione di (3.32) si può riscrivere come

$$\mathbf{W}_{11}\boldsymbol{\beta} - \mathbf{s}_{12} - \lambda\gamma_{12} = \mathbf{0}. \quad (3.37)$$

Aggiungendo il fatto che

$$\gamma_{12} = \text{sign}(\Theta_{12}) = \text{sign}(-\theta_{22}\boldsymbol{\beta}) = -\text{sign}(\boldsymbol{\beta}) \quad (3.38)$$

perchè $\theta_{22} > 0$, si può riscrivere (3.37) come

$$\mathbf{W}_{11}\boldsymbol{\beta} - \mathbf{s}_{12} + \lambda \text{sign}(\boldsymbol{\beta}) = \mathbf{0}. \quad (3.39)$$

Tale equazione può essere vista come l'equazione del subgradiente associato ad un regolare problema LASSO. Infatti, se \mathbf{y} è il vettore risposta e \mathbf{Z} la matrice di predittori, il LASSO minimizza rispetto a $\boldsymbol{\beta}$

$$\frac{1}{2N}(\mathbf{y} - \mathbf{Z}\boldsymbol{\beta})^\top(\mathbf{y} - \mathbf{Z}\boldsymbol{\beta}) + \lambda\|\boldsymbol{\beta}\|_1, \quad (3.40)$$

e l'equazione del subgradiente associata è

$$\frac{1}{N}\mathbf{Z}^\top\mathbf{Z}\boldsymbol{\beta} - \frac{1}{N}\mathbf{Z}^\top\mathbf{y} + \lambda \text{sign}(\boldsymbol{\beta}) = \mathbf{0}, \quad (3.41)$$

si noti che essa coincide con (3.39) se si sostituisce $\mathbf{W}_{12} = \frac{1}{N}\mathbf{Z}^\top\mathbf{Z}$ e $\mathbf{s}_{12} = \frac{1}{N}\mathbf{Z}^\top\mathbf{y}$. Secondo tale notazione, il problema LASSO il cui subgradiente è (3.39) si può scrivere come

$$\hat{\boldsymbol{\beta}} = \min_{\boldsymbol{\beta} \in \mathbb{R}^{p-1}} \left\{ \frac{1}{2} \|\mathbf{W}_{11}^{\frac{1}{2}}\boldsymbol{\beta} - \mathbf{W}_{11}^{-\frac{1}{2}}\mathbf{s}_{12}\|_2^2 + \lambda\|\boldsymbol{\beta}\|_1 \right\}. \quad (3.42)$$

Inoltre, ponendo a sistema la seconda e la quarta equazione di (3.35)

$$\begin{cases} \mathbf{W}_{11}\boldsymbol{\theta}_{12} + \mathbf{w}_{12}\theta_{22} = \mathbf{0}_{(p-1) \times 1} \\ \mathbf{w}_{12}^\top\boldsymbol{\theta}_{12} + w_{22}\theta_{22} = 1 \end{cases} \quad (3.43)$$

si può ottenere

$$\theta_{22} = [w_{22} - \mathbf{w}_{12}^\top\mathbf{W}_{11}^{-1}\mathbf{w}_{12}]^{-1} = [w_{22} - \mathbf{w}_{12}^\top\boldsymbol{\beta}]^{-1}. \quad (3.44)$$

Algoritmo 1 Graphical-LASSO**Data:** Matrice di dati $\mathbf{X} \in \mathbb{R}^{n \times p}$;**Input:** Matrice di covarianza campionaria $\mathbf{S} \in \mathbb{R}^{p \times p}$, parametro di regolazione $\lambda > 0$;**Inizializza:** Inizializzare $\mathbf{W} = \mathbf{S} + \lambda \mathbf{I}$, $j = 0$, $\hat{\Theta} = \mathbf{I}$;**for** $j = 1, \dots, p, 1, \dots, p \dots$ fino a convergenza **do**

- Partizionare le righe e le colonne di \mathbf{S} e \mathbf{W} come in (3.28) in modo tale che la colonna target sia sempre l'ultima;
- risolvere il problema LASSO in equazione (3.42) ed ottenere $\hat{\beta}$;
- aggiornare l'elemento off-diagonal $\hat{w}_{12} = \mathbf{W}_{11} \hat{\beta}$ tramite (3.36);

nel ciclo finale, per ogni j , calcolare gli elementi diagonali di $\hat{\Theta}$ usando (3.44), ovvero $\hat{\theta}_{22} = 1/(w_{22} - \mathbf{w}_{12}^\top \hat{\beta})$ e gli elementi fuori dalla diagonale di $\hat{\Theta}$ grazie a (3.36) con $\hat{\theta}_{12} = -\hat{\beta} \hat{\theta}_{22}$ e $\hat{\theta}_{21} = \hat{\theta}_{12}^\top$

end**Output:** $\hat{\Theta}$

Restringendo l'analisi ai soli 53 geni descritti nel Paragrafo 3.1, la matrice di dati risulta avere 53 geni sulle righe e 6413 cellule sulle colonne. Come spiegato nel Capitolo 1, però, tale notazione viene spesso usata in contesto biologico e bioinformatico. In statistica si usa invece usare il trasposto di tale matrice, dove le cellule che rappresentano le unità statistiche sono sulle righe mentre i geni che rappresentano le variabili sono sulle colonne. Una volta effettuata la normalizzazione, è possibile assumere per ogni gene una distribuzione normale univariata,

$$x_i \sim N(\mu_i, \sigma_i^2), \quad i = 1, \dots, 53, \quad (3.45)$$

La distribuzione congiunta di p variabili casuali normali univariate segue la distribuzione normale multivariata, quindi

$$\mathbf{X} = (\mathbf{x}_1, \dots, \mathbf{x}_p) \sim N_p(\boldsymbol{\mu}, \boldsymbol{\Sigma}), \quad (3.46)$$

dove $\boldsymbol{\mu} \in \mathbb{R}^p$, $\boldsymbol{\Sigma} \in \mathbb{R}^{p \times p}$ e $p = 53$. Le 6413 cellule sono osservazioni, per ora assunte i.i.d. da una $N_p(\boldsymbol{\mu}, \boldsymbol{\Sigma})$.

La matrice di varianze e covarianze di \mathbf{X} , chiamata $\boldsymbol{\Sigma}$, ha dimensioni 53×53 ,

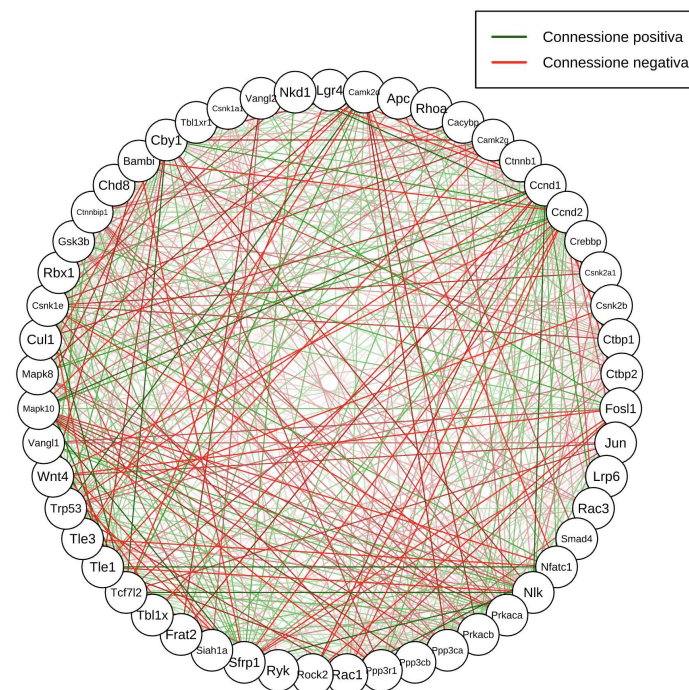


Figura 3.1: Grafo stimato via *CVglasso*.

così come la matrice di precisione $\Theta = \Sigma^{-1}$. Quest'ultima matrice è assunta sparsa; questo vuol dire che sono pochi i geni che dipendono da altri. Il graphical-LASSO può essere usato in questo caso per ricostruire la struttura sparsa di Θ , il che significa studiare la struttura di relazione tra geni tramite un grafo.

Applicando la funzione *CVglasso* del pacchetto *CVglasso* in R viene scelto il parametro di penalità λ che minimizza l'errore di convalida incrociata; in questo caso si ha $\lambda = 0.0006$ e tale funzione restituisce la stima sparsa di Θ . Dal momento che $\Theta \in \mathbb{R}^{53 \times 53}$, allora i valori stimati dalla funzione sono $53(53-1)/2 = 1378$; di questi, il graphical-LASSO ne stima esattamente nulli 105, vale a dire il 7.6%. In Figura 3.1 viene raffigurato il grafo corrispondente alla matrice di precisione stimata dalla funzione; i 53 nodi sono i geni mentre gli archi rappresentano le relazioni tra essi. Archi verdi corrispondono a relazioni positive, archi rossi corrispondono a relazioni negative.

3.4 Metodo delle direzioni alternate dei moltiplicatori

3.4.1 Problemi vincolati e discesa duale

Come spiegato precedentemente, molti problemi in statistica sono riconducibili a problemi di ottimizzazione convessa. In seguito viene data la formulazione del problema data da Boyd et al. (2010). Si consideri un generale problema di ottimizzazione, chiamato problema primale:

$$\begin{aligned} \min_x \quad & f(x) \\ \text{soggetto a} \quad & h_i(x) \leq 0, \quad i = 1, \dots, m \\ & l_j(x) = 0, \quad j = 1, \dots, r. \end{aligned} \quad (3.47)$$

Viene definito il suo Lagrangiano come

$$L(x, u, v) = f(x) + \sum_{i=1}^m u_i h_i(x) + \sum_{j=1}^r v_j l_j(x), \quad (3.48)$$

o equivalentemente

$$L(x, \mathbf{u}, \mathbf{v}) = f(x) + \mathbf{u}^\top \mathbf{h}(x) + \mathbf{v}^\top \mathbf{l}(x), \quad (3.49)$$

dove $\mathbf{u} \in \mathbb{R}^m$ e $\mathbf{v} \in \mathbb{R}^r$ sono i moltiplicatori di Lagrange. Il valore ottimo del problema primale, chiamato f^* , è

$$f^* = \inf_x \sup_{\mathbf{u} \geq 0, \mathbf{v}} L(x, \mathbf{u}, \mathbf{v}). \quad (3.50)$$

Dato un Lagrangiano, si definisce la sua funzione duale

$$g(\mathbf{u}, \mathbf{v}) = \inf_x L(x, \mathbf{u}, \mathbf{v}). \quad (3.51)$$

Dato un problema primale (3.47), il suo rispettivo problema duale è

$$\begin{aligned} \max_{\mathbf{u}, \mathbf{v}} \quad & g(\mathbf{u}, \mathbf{v}) \\ \text{soggetto a} \quad & \mathbf{u} \geq 0, \end{aligned} \quad (3.52)$$

che ha soluzione pari a

$$g^* = \sup_{\mathbf{u} \geq 0, \mathbf{v}} \inf_x L(x, \mathbf{u}, \mathbf{v}). \quad (3.53)$$

Si definisce dualità debole la condizione tale per cui $f^* \geq g^*$ e dualità forte la condizione tale per cui $f^* = g^*$.

Si consideri il problema di ottimizzazione convessa vincolata:

$$\begin{aligned} \min_{\boldsymbol{\beta}} f(\boldsymbol{\beta}), \\ \text{sotto il vincolo } \mathbf{A}\boldsymbol{\beta} = \mathbf{b}, \end{aligned} \quad (3.54)$$

dove $\boldsymbol{\beta} \in \mathbb{R}^n$, $\mathbf{A} \in \mathbb{R}^{m \times n}$ e $f : \mathbb{R}^n \rightarrow \mathbb{R}$ funzione convessa.

Il Lagrangiano del problema è

$$L(\boldsymbol{\beta}, \mathbf{u}) = f(\boldsymbol{\beta}) + \mathbf{u}^\top (\mathbf{A}\boldsymbol{\beta} - \mathbf{b}), \quad (3.55)$$

con $\mathbf{u} \in \mathbb{R}^m$ moltiplicatori di Lagrange, mentre la funzione duale è

$$g(\mathbf{u}) = \inf_{\boldsymbol{\beta}} L(\boldsymbol{\beta}, \mathbf{u}) = \inf_{\boldsymbol{\beta}} [f(\boldsymbol{\beta}) + \mathbf{u}^\top \mathbf{A}\boldsymbol{\beta}] - \mathbf{b}^\top \mathbf{u}. \quad (3.56)$$

Sia $f : \mathbb{R}^n \rightarrow \mathbb{R}$; la funzione coniugata di f è la funzione $f^* : \mathbb{R}^n \rightarrow \mathbb{R}$ definita come

$$f^*(\mathbf{y}) = \sup_{\mathbf{x} \in \text{dom} f} [\mathbf{y}^\top \mathbf{x} - f(\mathbf{x})] = - \inf_{\mathbf{x} \in \text{dom} f} [f(\mathbf{x}) - \mathbf{y}^\top \mathbf{x}]. \quad (3.57)$$

Di conseguenza si può scrivere

$$\inf_{\boldsymbol{\beta}} [f(\boldsymbol{\beta}) + \boldsymbol{\beta}^\top \mathbf{A}\mathbf{u}] = -f^*(-\mathbf{A}^\top \mathbf{u}), \quad (3.58)$$

quindi

$$g(\mathbf{u}) = \inf_{\boldsymbol{\beta}} L(\boldsymbol{\beta}, \mathbf{u}) = -f^*(-\mathbf{A}^\top \mathbf{u}) - \mathbf{b}^\top \mathbf{u}. \quad (3.59)$$

Il problema duale è quindi

$$\max_{\mathbf{u}} g(\mathbf{u}), \quad (3.60)$$

con $\mathbf{u} \in \mathbb{R}^m$

Assumendo che ci sia dualità forte, la soluzione del problema primale equivale alla soluzione del problema duale. Di conseguenza, si può trovare il punto primale ottimo $\boldsymbol{\beta}^*$ dal punto duale ottimo \mathbf{u}^* tramite

$$\boldsymbol{\beta}^* = \underset{\boldsymbol{\beta}}{\text{argmin}} L(\boldsymbol{\beta}, \mathbf{u}^*). \quad (3.61)$$

Nel metodo della discesa duale questo tipo di problema viene risolto attraverso l'ascesa del gradiente. Assumendo $g()$ differenziabile, si può trovare $\boldsymbol{\beta}^+ = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} L(\boldsymbol{\beta}, \mathbf{u})$ che porta a $\nabla g(\mathbf{u}) = \mathbf{A}\boldsymbol{\beta}^+ - \mathbf{b}$, vale a dire i residui del vincolo. Il metodo dell'ascesa duale consiste in due step iterativi:

$$\begin{aligned}\boldsymbol{\beta}^{(k+1)} &:= \underset{\boldsymbol{\beta}}{\operatorname{argmin}} L(\boldsymbol{\beta}, \mathbf{u}^{(k)}), \\ \mathbf{u}^{(k+1)} &:= \mathbf{u}^{(k)} + \alpha^{(k)}(\mathbf{A}\boldsymbol{\beta}^{(k+1)} - \mathbf{b}),\end{aligned}\tag{3.62}$$

dove k rappresenta la k -esima iterazione e $\alpha^{(k)}$ è l'ampiezza del passo verso la minimizzazione. Il primo step è di minimizzazione della L mentre il secondo è l'aggiornamento della variabile duale. La variabile \mathbf{u} può essere interpretata come un vettore di "prezzi", e il suo aggiornamento viene descritto come passo per "l'aggiustamento dei prezzi".

Il maggior vantaggio del metodo dell'ascesa duale è il fatto di potersi in alcuni casi ricondurre ad un algoritmo decentralizzato; in altre parole, riesce a raggiungere un obiettivo globale attraverso diverse dinamiche locali che comunicano tra loro. Si supponga, ad esempio, di poter dividere il vettore $\boldsymbol{\beta}$ in N sotto-vettori, rendendo la funzione f separabile e quindi:

$$f(\boldsymbol{\beta}) = \sum_{i=1}^N f_i(\boldsymbol{\beta}_i),\tag{3.63}$$

dove $\boldsymbol{\beta} = (\boldsymbol{\beta}_1, \dots, \boldsymbol{\beta}_N)$ e le variabili $\boldsymbol{\beta}_i \in \mathbb{R}^{n_i}$ sono sotto-vettori di $\boldsymbol{\beta}$. Partizionando \mathbf{A} come $\mathbf{A} = [\mathbf{A}_1, \dots, \mathbf{A}_N]$ tale che $\mathbf{A}\boldsymbol{\beta} = \sum_{i=1}^N \mathbf{A}_i\boldsymbol{\beta}_i$ il Lagrangiano può essere scritto come

$$L(\boldsymbol{\beta}, \mathbf{u}) = \sum_{i=1}^N L_i(\boldsymbol{\beta}_i, \mathbf{u}) = \sum_{i=1}^N [f_i(\boldsymbol{\beta}_i) + \mathbf{u}^\top \mathbf{A}_i \boldsymbol{\beta}_i] - (1/N)\mathbf{u}^\top \mathbf{b},\tag{3.64}$$

che è a sua volta separabile in $\boldsymbol{\beta}$. Ciò significa che lo step di minimizzazione di L descritto prima può essere diviso in N passi paralleli, mentre lo step di aggiornamento di \mathbf{u} rimane uguale andando a raggruppare l'informazione degli N passi indipendenti.

$$\begin{aligned}\boldsymbol{\beta}_i^{(k+1)} &:= \underset{\boldsymbol{\beta}_i}{\operatorname{argmin}} L(\boldsymbol{\beta}_i, \mathbf{u}^{(k)}), \\ \mathbf{u}^{(k+1)} &:= \mathbf{u}^{(k)} + \alpha^{(k)}(\mathbf{A}\boldsymbol{\beta}^{(k+1)} - \mathbf{b}).\end{aligned}\tag{3.65}$$

In questo caso quindi si fa riferimento al metodo come “decomposizione duale”. Il metodo del Lagrangiano aumentato è stato proposto da Powell (1969) e permette di aumentare la robustezza del metodo della discesa duale e di arrivare alla convergenza dell’algoritmo senza assunzioni quali la convessità stretta e la finitezza di f . La forma del Lagrangiano aumentato è:

$$L_\rho(\boldsymbol{\beta}, \mathbf{u}) = f(\boldsymbol{\beta}) + \mathbf{u}^\top(\mathbf{A}\boldsymbol{\beta} - \mathbf{b}) + \frac{\rho}{2}\|\mathbf{A}\boldsymbol{\beta} - \mathbf{b}\|_2^2, \quad (3.66)$$

dove ρ è un parametro positivo di penalità. Si noti che se tale parametro fosse pari a zero, la forma ritorna quella del Lagrangiano standard. Inoltre, il Lagrangiano aumentato può essere visto come un Lagrangiano non aumentato associato al problema

$$\begin{aligned} \min f(\boldsymbol{\beta}) + \frac{\rho}{2}\|\mathbf{A}\boldsymbol{\beta} - \mathbf{b}\|_2^2, \\ \text{sotto il vincolo } \mathbf{A}\boldsymbol{\beta} = \mathbf{b}. \end{aligned} \quad (3.67)$$

La funzione duale $g_\rho(\boldsymbol{\beta})$ sarà $\inf_{\boldsymbol{\beta}} L_\rho(\boldsymbol{\beta}, \mathbf{u})$. Grazie all’aggiunta del termine di penalizzazione, si indeboliscono le condizioni che permettono la differenziabilità di g_ρ . Applicando il metodo della ascesa duale si ottiene l’algoritmo

$$\begin{aligned} \boldsymbol{\beta}^{(k+1)} &:= \operatorname{argmin}_{\boldsymbol{\beta}} L_\rho(\boldsymbol{\beta}, \mathbf{u}^{(k)}), \\ \mathbf{u}^{(k+1)} &:= \mathbf{u}^{(k)} + \rho(\mathbf{A}\boldsymbol{\beta}^{(k+1)} - \mathbf{b}), \end{aligned} \quad (3.68)$$

conosciuto come metodo dei moltiplicatori. Le modifiche rispetto all’algoritmo della discesa duale sono due: nel passo della minimizzazione di $\boldsymbol{\beta}$ viene sostituito il Lagrangiano con il Lagrangiano aumentato, mentre nel secondo step si sostituisce $\alpha^{(k)}$ con ρ . L’algoritmo ha il vantaggio di migliorare le proprietà di convergenza ma ha lo svantaggio che L_ρ non è separabile, quindi non si può usare la decomposizione duale.

3.4.2 Algoritmo ADMM

Il metodo delle direzioni alternate dei moltiplicatori (ADMM, *alternating direction method of multipliers*), proposto per la prima da Glowinski and Marocco (1975) e Gabay and Mercier (1976), è un algoritmo che unisce le proprietà di convergenza del metodo dei moltiplicatori con le proprietà della

decomposizione duale dell'ascesa duale. Tale algoritmo risolve problemi della forma:

$$\begin{aligned} \min_{\boldsymbol{\beta}} f(\boldsymbol{\beta}) + g(\mathbf{z}), \\ \text{sotto il vincolo } \mathbf{A}\boldsymbol{\beta} + \mathbf{B}\mathbf{z} = \mathbf{c}, \end{aligned} \quad (3.69)$$

dove $\boldsymbol{\beta} \in \mathbb{R}^p$, $\mathbf{z} \in \mathbb{R}^m$, $\mathbf{A} \in \mathbb{R}^{n \times p}$, $\mathbf{B} \in \mathbb{R}^{n \times m}$ e $\mathbf{c} \in \mathbb{R}^n$. Si assume che le funzioni f e g siano convesse. La differenza rispetto al problema precedente è che $\boldsymbol{\beta}$ viene divisa in due parti denominate $\boldsymbol{\beta}$ e \mathbf{z} , così come il vincolo. Viene definito in questo caso il Lagrangiano aumentato come:

$$L_\rho(\boldsymbol{\beta}, \mathbf{z}, \mathbf{u}) = f(\boldsymbol{\beta}) + g(\mathbf{z}) + \mathbf{u}^\top(\mathbf{A}\boldsymbol{\beta} + \mathbf{B}\mathbf{z} - \mathbf{c}) + \frac{\rho}{2} \|\mathbf{A}\boldsymbol{\beta} + \mathbf{B}\mathbf{z} - \mathbf{c}\|_2^2. \quad (3.70)$$

L'algoritmo ADMM consiste di tre iterazioni:

$$\begin{aligned} \boldsymbol{\beta}^{(k+1)} &:= \operatorname{argmin}_{\boldsymbol{\beta}} L_\rho(\boldsymbol{\beta}, \mathbf{z}^{(k)}, \mathbf{u}^{(k)}), \\ \mathbf{z}^{(k+1)} &:= \operatorname{argmin}_{\mathbf{z}} L_\rho(\boldsymbol{\beta}^{(k+1)}, \mathbf{z}, \mathbf{u}^{(k)}), \\ \mathbf{u}^{(k+1)} &:= \mathbf{u}^{(k)} + \rho(\mathbf{A}\boldsymbol{\beta}^{(k+1)} + \mathbf{B}\mathbf{z}^{(k+1)} - \mathbf{c}). \end{aligned} \quad (3.71)$$

L'algoritmo è molto simile a quello dell'ascesa duale: un primo step per la minimizzazione di $\boldsymbol{\beta}$, un secondo per la minimizzazione di \mathbf{z} ed uno infine per l'aggiornamento della variabile duale \mathbf{u} .

La versione scalata dell'algoritmo ADMM è lievemente modificata e verrà applicata in seguito. Definito il residuo $\mathbf{r} = \mathbf{A}\boldsymbol{\beta} + \mathbf{B}\mathbf{z} - \mathbf{c}$ si avrà:

$$\mathbf{u}^\top \mathbf{r} + \frac{\rho}{2} \|\mathbf{r}\|_2^2 = \frac{\rho}{2} \|\mathbf{r}\|_2^2 + \frac{1}{\rho} \|\mathbf{u}\|_2^2 - \frac{1}{2\rho} \|\mathbf{u}\|_2^2 = \frac{\rho}{2} \|\mathbf{r} + \tilde{\mathbf{u}}\|_2^2 - \frac{\rho}{2} \|\tilde{\mathbf{u}}\|_2^2, \quad (3.72)$$

dove $\tilde{\mathbf{u}} = \frac{1}{\rho} \mathbf{u}$ ed il Lagrangiano aumentato in (3.70) diventa

$$\begin{aligned} L_\rho(\boldsymbol{\beta}, \mathbf{z}, \mathbf{u}) &= f(\boldsymbol{\beta}) + g(\mathbf{z}) + \mathbf{u}^\top \mathbf{r} + \frac{\rho}{2} \|\mathbf{r}\|_2^2 = \\ &= f(\boldsymbol{\beta}) + g(\mathbf{z}) + \frac{\rho}{2} \|\mathbf{r} + \tilde{\mathbf{u}}\|_2^2 - \frac{\rho}{2} \|\tilde{\mathbf{u}}\|_2^2. \end{aligned} \quad (3.73)$$

Di conseguenza si può riscrivere la procedura iterativa come:

$$\begin{aligned} \boldsymbol{\beta}^{(k+1)} &= \operatorname{argmin}_{\boldsymbol{\beta}} \left(f(\boldsymbol{\beta}) + \frac{\rho}{2} \|\mathbf{A}\boldsymbol{\beta} + \mathbf{B}\mathbf{z}^{(k)} - \mathbf{c} + \tilde{\mathbf{u}}^{(k)}\|_2^2 \right), \\ \mathbf{z}^{(k+1)} &= \operatorname{argmin}_{\mathbf{z}} \left(f(\mathbf{z}) + \frac{\rho}{2} \|\mathbf{A}\boldsymbol{\beta}^{(k+1)} + \mathbf{B}\mathbf{z} - \mathbf{c} + \tilde{\mathbf{u}}^{(k)}\|_2^2 \right), \\ \tilde{\mathbf{u}}^{(k+1)} &= \tilde{\mathbf{u}}^{(k)} + \mathbf{A}\boldsymbol{\beta}^{(k+1)} + \mathbf{B}\mathbf{z}^{(k+1)} - \mathbf{c}. \end{aligned} \quad (3.74)$$

3.4.3 Convergenza dell'algoritmo

In Boyd et al. (2010) viene descritto inoltre un risultato riguardante le proprietà di convergenza dell'algoritmo ADMM. La prima assunzione necessaria riguarda le funzioni f e g ed è la seguente:

Assunzione 1.

Le funzioni $f : \mathbb{R}^p \rightarrow \mathbb{R} \cup \{+\infty\}$ e $g : \mathbb{R}^m \rightarrow \mathbb{R} \cup \{+\infty\}$ sono chiuse, proprie e convesse.

Questa prima assunzione implica che esistono β e \mathbf{z} non necessariamente uniche che minimizzano il Lagrangiano aumentato. In aggiunta, l'assunzione riguardante il codominio permette alle funzioni di essere non differenziabili e di poter assumere il valore $+\infty$.

La seconda assunzione è la seguente:

Assunzione 2.

Il Lagrangiano non aumentato L_0 ha un punto di sella

Ciò significa che esiste un $(\beta^*, \mathbf{z}^*, \mathbf{u}^*)$, non necessariamente unico, tale per cui

$$L_0(\beta^*, \mathbf{z}^*, \mathbf{u}) \leq L_0(\beta^*, \mathbf{z}^*, \mathbf{u}^*) \leq L_0(\beta^*, \mathbf{z}, \mathbf{u}). \quad (3.75)$$

Grazie alla prima assunzione, ciò implica che $L_0(\beta^*, \mathbf{z}^*, \mathbf{u}^*)$ è finito per ogni punto di sella $(\beta^*, \mathbf{z}^*, \mathbf{u}^*)$. Ciò implica che (β^*, \mathbf{z}^*) è soluzione di 3.2, quindi $\mathbf{A}\beta^* + \mathbf{B}\mathbf{z}^* = \mathbf{c}$, $f(\beta^*) < \infty$ e $g(\mathbf{z}^*) < \infty$. Questo implica inoltre che \mathbf{u}^* è soluzione del problema primale e duale.

Sotto le due assunzioni descritte precedentemente, l'algoritmo ADMM soddisfa i seguenti punti:

- $\mathbf{r}^{(k)} \rightarrow 0$ per $k \rightarrow \infty$, vale a dire che per k che tende all'infinito, $\mathbf{r}^{(k)}$ tende a 0;
- $f(\beta^{(k)}) + g(\mathbf{z}^{(k)}) \rightarrow \mathbf{p}^*$ per $k \rightarrow \infty$, vale a dire che per k che tende all'infinito, la funzione obiettivo iterata converge al valore ottimale \mathbf{p}^* ;
- $\mathbf{u}^{(k)} \rightarrow \mathbf{u}^*$ per $k \rightarrow \infty$ vale a dire che per k che tende all'infinito, la variabile duale \mathbf{u} converge al suo valore ottimo \mathbf{u}^* .

3.4.4 Condizioni di ottimalità e criteri per fermare l'algoritmo

Le condizioni necessarie e sufficienti tali per cui $(\boldsymbol{\beta}^*, \mathbf{z}^*, \mathbf{u}^*)$ è la soluzione ottima per il problema ADMM (3.68) sono l'esistenza primale

$$\mathbf{A}\boldsymbol{\beta}^* + \mathbf{B}\mathbf{z}^* - \mathbf{c} = 0, \quad (3.76)$$

e l'esistenza duale

$$\begin{aligned} 0 &\in \partial f(\boldsymbol{\beta}^*) + \mathbf{A}^\top \mathbf{u}^*, \\ 0 &\in \partial g(\mathbf{z}^*) + \mathbf{B}^\top \mathbf{u}^*, \end{aligned} \quad (3.77)$$

dove ∂ sta ad indicare l'operatore subdifferenziale. Dal momento che per definizione \mathbf{z}^* minimizza $L_\rho(\boldsymbol{\beta}^{(k+1)}, \mathbf{z}, \mathbf{u}^{(k)})$, allora si avrà

$$\mathbf{A}\boldsymbol{\beta}^* + \mathbf{B}\mathbf{z}^* - \mathbf{c} = 0, \quad (3.78)$$

e si avrà

$$\begin{aligned} 0 &\in \partial g(\mathbf{z}^{(k+1)}) + \mathbf{B}^\top \mathbf{u}^{(k)} + \rho \mathbf{B}^\top (\mathbf{A}\boldsymbol{\beta}^{(k+1)} + \mathbf{B}\mathbf{z}^{(k+1)} - \mathbf{c}) \\ &= \partial g(\mathbf{z}^{(k+1)}) + \mathbf{B}^\top \mathbf{u}^{(k)} + \rho \mathbf{B}^\top (\mathbf{r}^{(k+1)}) \\ &= \partial g(\mathbf{z}^{(k+1)}) + \mathbf{B}^\top \mathbf{u}^{(k+1)}, \end{aligned} \quad (3.79)$$

dove

$$\mathbf{r}^{(k+1)} = \mathbf{A}\boldsymbol{\beta}^{(k+1)} + \mathbf{B}\mathbf{z}^{(k+1)} - \mathbf{c} \quad (3.80)$$

è detto residuo primale al passo $(k+1)$. Questo significa che $\mathbf{z}^{(k+1)}$ e $\mathbf{u}^{(k+1)}$ soddisfano sempre l'equazione (3.76) e di conseguenza anche le due condizioni (3.77). Per definizione $\boldsymbol{\beta}^{(k+1)}$ minimizza $L_\rho(\boldsymbol{\beta}, \mathbf{z}^{(k)}, \mathbf{u}^{(k)})$, quindi si avrà:

$$\begin{aligned} 0 &\in \partial f(\boldsymbol{\beta}^{(k+1)}) + \mathbf{A}^\top \mathbf{u}^{(k)} + \rho \mathbf{A}^\top (\mathbf{A}\boldsymbol{\beta}^{(k+1)} + \mathbf{B}\mathbf{z}^k - \mathbf{c}) \\ &= \partial f(\boldsymbol{\beta}^{(k+1)}) + \mathbf{A}^\top \mathbf{u}^{(k)} + \rho \mathbf{r}^{(k+1)} + \rho \mathbf{B}(\mathbf{z}^{(k)} - \mathbf{z}^{(k+1)}) \\ &= \partial f(\boldsymbol{\beta}^{(k+1)}) + \mathbf{A}^\top \mathbf{u}^{(k+1)} + \rho \mathbf{A}^\top \mathbf{B}(\mathbf{z}^{(k)} - \mathbf{z}^{(k+1)}), \end{aligned} \quad (3.81)$$

o equivalentemente

$$\rho \mathbf{A}^\top \mathbf{B}(\mathbf{z}^{(k)} - \mathbf{z}^{(k+1)}) \in \partial f(\boldsymbol{\beta}^{(k+1)}) + \mathbf{A}^\top \mathbf{u}^{(k+1)}. \quad (3.82)$$

Questo significa che la quantità

$$\mathbf{s}^{(k+1)} = \rho \mathbf{A}^\top \mathbf{B}(\mathbf{z}^{(k)} - \mathbf{z}^{(k+1)}), \quad (3.83)$$

detta residuo duale, può essere vista come il residuo per la condizione duale d'esistenza al passo $(k + 1)$. Riassumendo, le condizioni di ottimo per il problema ADMM sono tre, ovvero (3.76) e (3.77). L'ultima condizione di (3.77) vale sempre per $(\boldsymbol{\beta}^{(k+1)}, \mathbf{z}^{(k+1)}, \mathbf{u}^{(k+1)})$; i residui per le altre due sono rispettivamente i residui primali e duali $\mathbf{r}^{(k+1)}$ ed $\mathbf{s}^{(k+1)}$. Entrambe queste quantità convergono a zero all'aumentare delle iterazioni.

In Boyd et al. (2010) vengono suggeriti alcuni criteri per cui terminare le iterazioni dell'algoritmo. Si può dimostrare che

$$f(\boldsymbol{\beta}) + g(\mathbf{z}^{(k)}) - p^* \leq -(\mathbf{u}^{(k)})^\top \mathbf{r}^{(k)} + (\boldsymbol{\beta}^{(k)} - \boldsymbol{\beta}^*)^\top \mathbf{s}^{(k)}, \quad (3.84)$$

dove p^* indica il valore ottimo della funzione di perdita e $\boldsymbol{\beta}^*$ le variabili corrispondenti. Tale disuguaglianza mostra che se quando i residui $\mathbf{r}^{(k)}$ e $\mathbf{s}^{(k)}$ sono piccoli, allora anche la differenza tra il valore obiettivo alla k -esima iterazione ed il vero valore ottimale deve essere piccola. Ciononostante, non conoscendo $\boldsymbol{\beta}^*$ è impossibile usare direttamente la disuguaglianza per un criterio di fermata. Se si assume però che $\|\boldsymbol{\beta}^{(k)} - \boldsymbol{\beta}^*\|_2$ sia più piccolo di un certo valore d , si ha che

$$f(\boldsymbol{\beta}) + g(\mathbf{z}^{(k)}) - p^* \leq -(\mathbf{u}^{(k)})^\top \mathbf{r}^{(k)} + d\|\mathbf{s}^{(k)}\|_2 \leq \|\mathbf{u}^{(k)}\|_2 \|\mathbf{r}^{(k)}\|_2 + d\|\mathbf{s}^{(k)}\|_2. \quad (3.85)$$

Questo suggerisce quindi che un possibile criterio di fermata per l'algoritmo ADMM sia che i due residui siano "piccoli":

$$\|\mathbf{r}^{(k)}\|_2 \leq \varepsilon^{\text{pri}}, \quad \|\mathbf{s}^{(k)}\|_2 \leq \varepsilon^{\text{dual}}, \quad (3.86)$$

dove $\varepsilon^{\text{pri}} > 0$ e $\varepsilon^{\text{dual}} > 0$ sono tolleranze per le condizioni di esistenza. Questi valori possono essere scelti tramite un criterio rispettivamente assoluto e relativo, come ad esempio:

$$\begin{aligned} \varepsilon^{\text{pri}} &= \sqrt{n}\varepsilon^{\text{ass}} + \varepsilon^{\text{rel}} \max\{\|\mathbf{A}\boldsymbol{\beta}^{(k)}\|_2, \|\mathbf{B}\mathbf{z}^{(k)}\|_2, \|\mathbf{c}\|_2\}, \\ \varepsilon^{\text{dual}} &= \sqrt{p}\varepsilon^{\text{ass}} + \varepsilon^{\text{rel}} \|\mathbf{A}^\top \mathbf{u}^{(k)}\|_2, \end{aligned} \quad (3.87)$$

dove $\varepsilon^{\text{ass}} > 0$ e $\varepsilon^{\text{rel}} > 0$ sono rispettivamente tolleranza assoluta e relative e vengono in generale determinate dall'utente. Un valore ragionevole per ε^{rel} può essere 10^{-4} o 10^{-3} , mentre la scelta di ε^{ass} dipende dalla scala o dal tipo di variabili.

3.5 ADMM per graphical-LASSO

L'algoritmo ADMM può essere applicato anche alla risoluzione del problema graphical-LASSO, descritto nel Paragrafo 3.3. Si parte dal problema definito in (3.26)

$$\underset{\Theta}{\text{maximize}} \quad \log |\Theta| - \text{tr}(\mathbf{S}\Theta) - \lambda \|\Theta\|_1,$$

vale a dire

$$\underset{\Theta}{\text{minimize}} \quad -\log |\Theta| + \text{tr}(\mathbf{S}\Theta) + \lambda \|\Theta\|_1, \quad (3.88)$$

che può anche essere espresso come

$$\underset{\Theta}{\text{minimize}} \quad -\log |\Theta| + \text{tr}(\mathbf{S}\Theta) + \lambda \|\mathbf{Z}\|_1 \quad (3.89)$$

$$\text{sotto il vincolo} \quad \Theta - \mathbf{Z} = 0.$$

Sfruttando la forma scalata dell'algoritmo ADMM, è possibile scrivere il Lagrangiano aumentato associato al problema come:

$$L_p(\Theta, \mathbf{Z}, \tilde{\mathbf{U}}) = -\log |\Theta| + \text{tr}(\mathbf{S}\Theta) + \lambda \|\mathbf{Z}\|_1 + \frac{\rho}{2} \|\Theta - \mathbf{Z} + \tilde{\mathbf{U}}\|_F^2 - \frac{\rho}{2} \|\tilde{\mathbf{U}}\|_F^2, \quad (3.90)$$

dove $\|\mathbf{A}\|_F = \sqrt{\mathbf{A}^\top \mathbf{A}}$ è la norma di Frobenius.

In questo caso i tre step saranno:

$$\begin{aligned} \Theta^{(k+1)} &= \underset{\Theta}{\text{argmin}} \left(-\log |\Theta| + \text{tr}(\mathbf{S}\Theta) + \frac{\rho}{2} \|\Theta - \mathbf{Z}^{(k)} + \tilde{\mathbf{U}}^{(k)}\|_F^2 \right), \\ \mathbf{Z}^{(k+1)} &= \underset{\mathbf{Z}}{\text{argmin}} \left(\lambda \|\mathbf{Z}\|_1 + \frac{\rho}{2} \|\Theta^{(k+1)} - \mathbf{Z} + \tilde{\mathbf{U}}^{(k)}\|_F^2 \right), \\ \tilde{\mathbf{U}}^{(k+1)} &= \tilde{\mathbf{U}} + \Theta^{(k+1)} - \mathbf{Z}^{(k+1)}. \end{aligned} \quad (3.91)$$

Si procede ora a sviluppare l'aggiornamento per Θ ponendo a zero il gradiente rispetto a Θ e si ottiene

$$\frac{\partial L_\rho(\Theta, \mathbf{Z}, \tilde{\mathbf{U}})}{\partial \Theta} = \mathbf{S} - \Theta^{-1} + \rho(\Theta - \mathbf{Z}^{(k)} + \tilde{\mathbf{U}}^{(k)}) = 0, \quad (3.92)$$

quindi

$$\rho \Theta - \Theta^{-1} = \rho(\mathbf{Z}^{(k)} - \tilde{\mathbf{U}}^{(k)}) - \mathbf{S}. \quad (3.93)$$

Da qui si cerca la matrice Θ che soddisfa tale condizione e che minimizzi la funzione obiettivo del primo step. Si esegue la decomposizione spettrale della parte di destra,

$$\mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^\top = \rho(\mathbf{Z}^{(k)} - \tilde{\mathbf{U}}^{(k)}) - \mathbf{S}, \quad (3.94)$$

dove $\mathbf{\Lambda} = \text{diag}(\lambda_1, \dots, \lambda_n)$ è una matrice diagonale che contiene gli autovalori di $\rho(\mathbf{Z}^{(k)} - \tilde{\mathbf{U}}^{(k)}) - \mathbf{S}$ e \mathbf{Q} è una matrice ortogonale che contiene i suoi autovettori normalizzati; vale quindi $\mathbf{Q}^\top\mathbf{Q} = \mathbf{Q}\mathbf{Q}^\top = \mathbf{I}$. Moltiplicando ambo i membri a destra per \mathbf{Q} e a sinistra per \mathbf{Q}^\top si ottiene

$$\begin{aligned} \mathbf{\Lambda} &= \mathbf{Q}^\top[\rho(\mathbf{Z}^{(k)} - \tilde{\mathbf{U}}^{(k)}) - \mathbf{S}]\mathbf{Q} = \mathbf{Q}^\top[\rho\mathbf{\Theta} - \mathbf{\Theta}^{-1}]\mathbf{Q} = \\ &= \mathbf{Q}^\top\rho\mathbf{\Theta}\mathbf{Q} - \mathbf{Q}^\top\mathbf{\Theta}^{-1}\mathbf{Q}. \end{aligned} \quad (3.95)$$

Se si indica con $\tilde{\mathbf{\Theta}} = \mathbf{Q}^\top\mathbf{\Theta}\mathbf{Q}$ si ottiene

$$\mathbf{\Lambda} = \rho\tilde{\mathbf{\Theta}} - \tilde{\mathbf{\Theta}}^{-1}, \quad (3.96)$$

vista poi la struttura diagonale di $\mathbf{\Lambda}$, è possibile scrivere (3.96) come

$$\begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_p \end{bmatrix} = \rho \begin{bmatrix} \tilde{\theta}_1 & & \\ & \ddots & \\ & & \tilde{\theta}_p \end{bmatrix} - \begin{bmatrix} \tilde{\theta}_1^{-1} & & \\ & \ddots & \\ & & \tilde{\theta}_p^{-1} \end{bmatrix}. \quad (3.97)$$

Si evince quindi che deve valere l'uguaglianza $\lambda_i = \rho\tilde{\theta}_i - \tilde{\theta}_i^{-1}$, $\forall i = 1, \dots, p$, che equivale a risolvere in $\tilde{\theta}_i$ l'equazione di secondo grado $\rho\tilde{\theta}_i^2 - \lambda_i\tilde{\theta}_i - 1 = 0$, che ha come soluzioni positive

$$\tilde{\theta}_i = \frac{\lambda_i + \sqrt{\lambda_i^2 + 4\rho}}{2\rho}, \quad \forall i = 1, \dots, p \quad (3.98)$$

Una volta trovati gli elementi della matrice diagonale $\tilde{\mathbf{\Theta}}$, sapendo che $\tilde{\mathbf{\Theta}} = \mathbf{Q}^\top\mathbf{\Theta}\mathbf{Q}$ allora $\mathbf{\Theta} = \mathbf{Q}\tilde{\mathbf{\Theta}}\mathbf{Q}^\top$. Così costruita, $\mathbf{\Theta}$ soddisfa l'equazione ottima (3.93). Viene definito l'operatore

$$\mathcal{F}_\rho(\mathbf{X}) = \mathbf{Q}\text{diag}\left\{\frac{\lambda_i + \sqrt{\lambda_i^2 + 4\rho}}{2\rho}\right\}\mathbf{Q}^\top, \quad (3.99)$$

dove $\mathbf{X} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^\top$ è la decomposizione spettrale di \mathbf{X} e $\mathbf{\Lambda} = \text{diag}\{\lambda_1, \dots, \lambda_p\}$. Si può dunque scrivere l'aggiornamento per $\mathbf{\Theta}$ come

$$\mathbf{\Theta}^{(k+1)} = \mathcal{F}_\rho[\rho(\mathbf{Z}^{(k)} - \tilde{\mathbf{U}}^{(k)}) - \mathbf{S}]. \quad (3.100)$$

Per quanto riguarda il passo di aggiornamento per \mathbf{Z} , è necessario porre uguale a 0 il gradiente rispetto a \mathbf{Z} , ovvero

$$\frac{\partial L_\rho(\Theta, \mathbf{Z}, \tilde{\mathbf{U}})}{\partial \mathbf{Z}} = 0, \quad (3.101)$$

ma il Lagrangiano, considerando solo le componenti che dipendono da \mathbf{Z} , contiene una parte non differenziabile rispetto a \mathbf{Z} , denominata con $L_\rho(\Theta, \mathbf{Z}, \tilde{\mathbf{U}})^A = \lambda \|\mathbf{Z}\|_1$ e da una seconda parte differenziabile rispetto a \mathbf{Z} , ovvero $L_\rho(\Theta, \mathbf{Z}, \tilde{\mathbf{U}})^B = \frac{\rho}{2} \|\Theta - \mathbf{Z} + \tilde{\mathbf{U}}\|_F^2$. Si può scrivere quindi

$$\frac{\partial L_\rho(\Theta, \mathbf{Z}, \tilde{\mathbf{U}})}{\partial \mathbf{Z}} = \frac{\partial L_\rho(\Theta, \mathbf{Z}, \tilde{\mathbf{U}})^A}{\partial \mathbf{Z}} + \frac{\partial L_\rho(\Theta, \mathbf{Z}, \tilde{\mathbf{U}})^B}{\partial \mathbf{Z}}. \quad (3.102)$$

Sulla prima parte, nonostante non sia differenziabile rispetto a \mathbf{Z} , è possibile applicare il subdifferenziale ottenendo:

$$\frac{\partial L_\rho(\Theta, \mathbf{Z}, \tilde{\mathbf{U}})^A}{\partial \mathbf{Z}} = \frac{\partial \lambda \|\mathbf{Z}\|_1}{\partial \mathbf{Z}} = \begin{cases} \lambda, & \text{se } \mathbf{Z} > 0 \\ [-\lambda, \lambda], & \text{se } \mathbf{Z} = 0 \\ -\lambda, & \text{se } \mathbf{Z} < 0 \end{cases} \quad (3.103)$$

Derivando la seconda parte differenziabile rispetto a \mathbf{Z} si ottiene invece

$$\frac{\partial L_\rho(\Theta, \mathbf{Z}, \tilde{\mathbf{U}})^B}{\partial \mathbf{Z}} = \frac{\partial \frac{\rho}{2} \|\Theta - \mathbf{Z} + \tilde{\mathbf{U}}\|_F^2}{\partial \mathbf{Z}} = \rho[\mathbf{Z} - (\Theta + \tilde{\mathbf{U}})]. \quad (3.104)$$

Ponendo uguale a 0 le due parti assieme risulta dunque:

$$0 = \begin{cases} \rho[\mathbf{Z} - (\Theta + \tilde{\mathbf{U}})] - \lambda, & \text{se } \mathbf{Z} < 0 \\ [-\rho(\Theta + \tilde{\mathbf{U}}) - \lambda, -\rho(\Theta + \tilde{\mathbf{U}}) + \lambda], & \text{se } \mathbf{Z} = 0 \\ \rho[\mathbf{Z} - (\Theta + \tilde{\mathbf{U}})] + \lambda, & \text{se } \mathbf{Z} > 0 \end{cases} \quad (3.105)$$

Nel secondo caso si ha:

$$\begin{aligned} 0 &\in [-\rho(\Theta + \tilde{\mathbf{U}}) - \lambda, -\rho(\Theta + \tilde{\mathbf{U}}) + \lambda] \\ -\rho(\Theta + \tilde{\mathbf{U}}) - \lambda &\leq 0 \\ -\rho(\Theta + \tilde{\mathbf{U}}) + \lambda &\geq 0, \end{aligned} \quad (3.106)$$

e si ottiene dunque

$$-\frac{\lambda}{\rho} \leq (\Theta + \tilde{\mathbf{U}}) \leq \frac{\lambda}{\rho}. \quad (3.107)$$

Risolvendo anche per chi altri 2 casi si ottiene

$$\begin{cases} \widehat{\mathbf{Z}} = (\mathbf{\Theta} + \tilde{\mathbf{U}}) - \frac{\lambda}{\rho}, & \text{se } \widehat{\mathbf{Z}} > 0 \\ \widehat{\mathbf{Z}} = 0, & \text{se } -\frac{\lambda}{\rho} \leq (\mathbf{\Theta} + \tilde{\mathbf{U}}) \leq \frac{\lambda}{\rho} \\ \widehat{\mathbf{Z}} = (\mathbf{\Theta} + \tilde{\mathbf{U}}) + \frac{\lambda}{\rho}, & \text{se } \widehat{\mathbf{Z}} < 0 \end{cases} \quad (3.108)$$

ottenendo l'operatore chiamato *soft-thresholding*:

$$\mathbf{S}_{\frac{\lambda}{\rho}}(\mathbf{\Theta} + \tilde{\mathbf{U}}) = \begin{cases} (\mathbf{\Theta} + \tilde{\mathbf{U}}) - \frac{\lambda}{\rho}, & \text{se } (\mathbf{\Theta} + \tilde{\mathbf{U}}) > \frac{\lambda}{\rho} \\ 0, & \text{se } |(\mathbf{\Theta} + \tilde{\mathbf{U}})| \leq \frac{\lambda}{\rho} \\ (\mathbf{\Theta} + \tilde{\mathbf{U}}) + \frac{\lambda}{\rho}, & \text{se } (\mathbf{\Theta} + \tilde{\mathbf{U}}) < -\frac{\lambda}{\rho} \end{cases} \quad (3.109)$$

Dunque l'aggiornamento per \mathbf{Z} è

$$\mathbf{Z}^{(k+1)} = \mathbf{S}_{\frac{\lambda}{\rho}}(\mathbf{\Theta}^{(k+1)} + \tilde{\mathbf{U}}^{(k)}). \quad (3.110)$$

Viene riportato in seguito l'algoritmo ADMM per la stima del problema graphical-LASSO. Tale algoritmo è stato implementato in linguaggio C++ grazie al pacchetto *rcpp* di R, che permette il dialogo tra l'ambiente R e la programmazione in C++. Il codice per esteso viene riportato in Appendice C.2.

Come si può facilmente notare, tale funzione, che restituisce tra i vari oggetti anche la stima della matrice di precisione $\widehat{\mathbf{\Theta}}$, dipende da una serie di parametri; i due principali presi in studio sono il parametro di regolazione $\lambda > 0$ ed il parametro di penalità $\rho > 0$. I restanti parametri sono fissati ai valori presenti in **Algoritmo 2**.

Una scelta standard del parametro di penalità ρ è fissarlo ad un valore piccolo come 0.01 e tenerlo costante per tutte le iterazioni. Wohlberg (2017) propone un metodo per far variare ρ ad ogni iterazione, con l'obiettivo di migliorare la convergenza dell'algoritmo, oltre a rendere le prestazioni meno dipendenti dalla scelta iniziale del parametro di penalità. Una semplice formulazione di quello che viene definito ρ adattivo è:

$$\rho^{(k+1)} = \begin{cases} \tau\rho^{(k)}, & \text{se } \|\mathbf{r}^{(k)}\|_2 > \mu\|\mathbf{s}^{(k)}\|_2 \\ \tau^{-1}\rho^{(k)}, & \text{se } \|\mathbf{s}^{(k)}\|_2 > \mu\|\mathbf{r}^{(k)}\|_2 \\ \rho^{(k)} & \text{altrimenti,} \end{cases} \quad (3.111)$$

dove μ e τ sono parametri positivi fissati.

Algoritmo 2 ADMM per graphical-LASSO**Data:** Matrice di dati $\mathbf{X} \in \mathbb{R}^{n \times p}$;**Input:** Matrice di covarianza campionaria $\mathbf{S} \in \mathbb{R}^{p \times p}$, parametro di penalità $\rho > 0$, parametro di regolazione $\lambda > 0$, tolleranza assoluta $\varepsilon^{\text{ass}} = 10^{-4}$, tolleranza relativa $\varepsilon^{\text{rel}} = 10^{-4}$, $\mu = 10$, $\tau = 2$, $\text{maxiter} = 1000$;**Inizializza:** $k = 0$, inizializzare i parametri $\Theta^{(0)} = \mathbf{I}$, $\mathbf{Z}^{(0)} = \mathbf{I}$, $\tilde{\mathbf{U}}^{(0)} = \mathbf{I}$, inizializzare i vettori di residui primali e duali $\hat{\mathbf{r}}^{(k+1)} = \hat{\mathbf{s}}^{(k+1)} = \infty$, calcolare la tolleranza primale ε^{pri} e duale $\varepsilon^{\text{dual}}$ secondo l'equazione(3.87);**while** ($\|\hat{\mathbf{r}}^{(k)}\|_2 > \varepsilon^{\text{pri}}$ e $\|\hat{\mathbf{s}}^{(k)}\|_2 > \varepsilon^{\text{dual}}$) **do** **for** $k = 1, \dots, \text{maxiter}$ **do**

- aggiornare $\Theta^{(k+1)} = \mathcal{F}_\rho[\rho(\mathbf{Z}^{(k)} - \tilde{\mathbf{U}}^{(k)}) - \mathbf{S}]$, come in equazione (3.99);
- aggiornare $\mathbf{Z}^{(k+1)} = \mathbf{S}_{\frac{\lambda}{\rho}}(\Theta^{(k+1)} + \tilde{\mathbf{U}}^{(k)})$, come in equazione (3.110);
- aggiornare $\tilde{\mathbf{U}}^{(k+1)} = \tilde{\mathbf{U}} + \Theta^{(k+1)} - \mathbf{Z}^{(k+1)}$.

end calcolare i vettori di residui primali e duali $\hat{\mathbf{r}}^{(k+1)}$ e $\hat{\mathbf{s}}^{(k+1)}$ come in equazione (3.80) e (3.83) ed aggiornare le tolleranze primali e duali ε^{pri} e $\varepsilon^{\text{dual}}$ come in equazione (3.87) $k = k + 1$ **end****Output:** $\hat{\Theta} \leftarrow \Theta^{(k+1)}$

Per la scelta del parametro λ ottimo ci si è affidati al criterio BIC (*Bayesian Information Criterion*) (Gao et al. (2012), Yuan and Lin (2007)), un criterio molto usato in statistica per la selezione del modello e del numero dei suoi parametri; solitamente si preferisce un modello con BIC più basso possibile. In questo caso si è scelto il parametro λ che minimizzasse il criterio BIC. Dato un generico modello, la formulazione generale del BIC è

$$BIC = p \cdot \log(n) - 2l(\hat{\theta}), \quad (3.112)$$

dove p è il numero di parametri del modello, n il numero di osservazioni, $l(\hat{\theta})$ è la log-verosimiglianza massimizzata. Nel caso preso in considerazione in questo capitolo, la log-verosimiglianza è funzione del parametro Θ , la cui stima dipende dal parametro di regolazione λ . La formulazione del BIC in

(3.112) diventa

$$BIC_\lambda = p \cdot \log(n) - 2l(\widehat{\Theta}_\lambda). \quad (3.113)$$

Come spiegato precedentemente, p è il numero di parametri stimati dal modello; chiamato $\widehat{\theta}_{ij,\lambda}$ il generico elemento (i, j) della matrice $\widehat{\Theta}_\lambda$, allora il numero di parametri stimati dal graphical-LASSO è

$$p = \sum_{i \leq j} |\widehat{\theta}_{ij,\lambda}| > 0.0001 \quad (3.114)$$

dove 0.0001 è il valore prefissato della tolleranza assoluta. Un arco può essere considerato zero se è in valore assoluto uguale o inferiore alla tolleranza assoluta prefissata

Riprendendo l'equazione della log-verosimiglianza in (3.20) si ha

$$l(\boldsymbol{\mu}, \boldsymbol{\Theta}) = \frac{n}{2} \log |\boldsymbol{\Theta}| - \frac{1}{2} \sum_{i=1}^n (\mathbf{x}_i - \boldsymbol{\mu})^\top \boldsymbol{\Theta} (\mathbf{x}_i - \boldsymbol{\mu}) \quad (3.115)$$

che, massimizzata per $\boldsymbol{\mu}$ diventa

$$l(\boldsymbol{\Theta}) = \frac{n}{2} \log |\boldsymbol{\Theta}| - \frac{1}{2} \sum_{i=1}^n (\mathbf{x}_i - \widehat{\boldsymbol{\mu}})^\top \boldsymbol{\Theta} (\mathbf{x}_i - \widehat{\boldsymbol{\mu}}). \quad (3.116)$$

Diventa quindi

$$\begin{aligned} 2l(\boldsymbol{\Theta}) &= n \log |\boldsymbol{\Theta}| - n \cdot \frac{1}{n} \sum_{i=1}^n (\mathbf{x}_i - \widehat{\boldsymbol{\mu}})^\top \boldsymbol{\Theta} (\mathbf{x}_i - \widehat{\boldsymbol{\mu}}) = \\ &= n \log |\boldsymbol{\Theta}| - n \cdot \text{tr}(\mathbf{S}\boldsymbol{\Theta}) \end{aligned} \quad (3.117)$$

Alla luce di questi risultati, l'equazione del criterio BIC per il problema graphical-LASSO in (3.114) diventa

$$BIC_\lambda = -n \cdot \text{tr}(\mathbf{S}\widehat{\Theta}_\lambda) + n \log |\widehat{\Theta}_\lambda| + \log(n) \sum_{i \leq j} |\widehat{\theta}_{ij,\lambda}| > 0.0001 \quad (3.118)$$

Si è fatto variare λ da un valore minimo di 0.001 ad un valore massimo di 6; in corrispondenza di $\lambda = 6$, tutti gli elementi non diagonali di $\widehat{\Theta}_\lambda$ sono considerati zero in base alla tolleranza.

Come si può vedere dalla Figura 3.2, l'indice BIC raggiunge un minimo in corrispondenza di λ pari a 0.0865, in corrispondenza del quale il modello

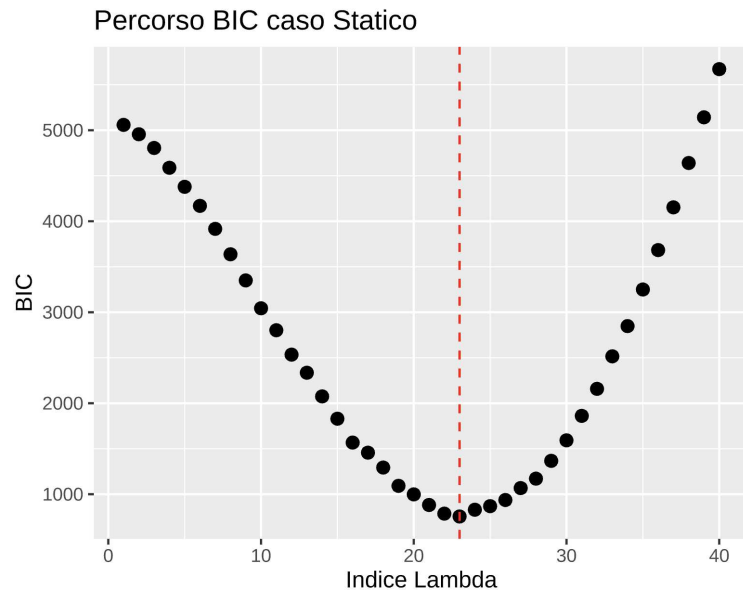


Figura 3.2: Andamento indice BIC.

stima 106 archi diversi da 0. Viene verificata inoltre la convergenza effettiva di ogni modello per ogni λ corrispondente. In Figura 3.3 viene riportato il grafo stimato in corrispondenza del parametro λ scelto. I 53 nodi sono i geni mentre le linee che li connettono sono gli archi; si tratta di un grafo molto più sparso rispetto a quello stimato nel Paragrafo 3.3, anche se la struttura globale sembra essere rimasta simile.

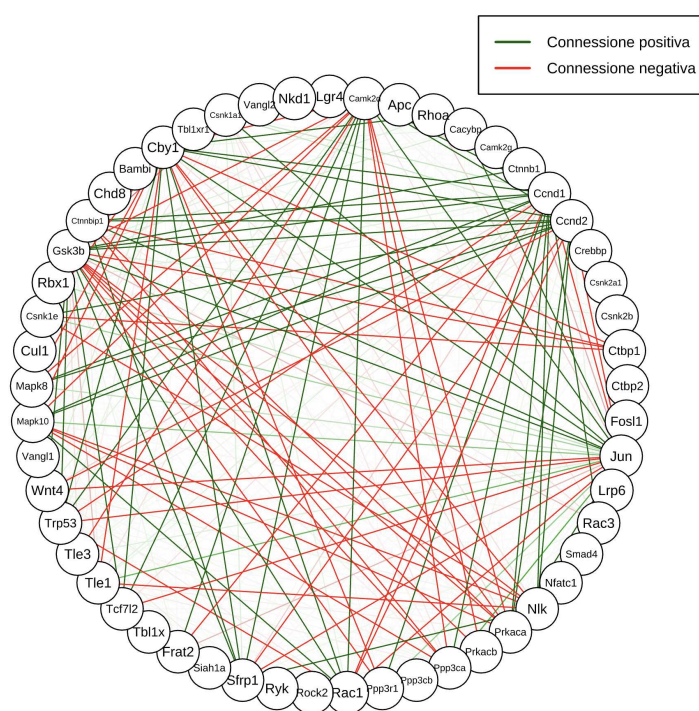


Figura 3.3: Grafo stimato via ADMM graphical-LASSO.

Capitolo 4

Modelli Grafici dinamici

In questo Capitolo si estende il modello statico presentato nel Capitolo 3. Tenendo conto dell'ordinamento pseudotemporale delle cellule, infatti, è stato possibile valutare se e come varia la stima della matrice di precisione sparsa al variare dello pseudotempo. Alla luce dei risultati ottenuti è stato inoltre possibile trarre alcune interessanti interpretazioni a livello biologico.

4.1 Introduzione al problema

Nel corso del precedente Capitolo sono stati introdotti ed applicati alcuni modelli per stimare la matrice di precisione sparsa di una distribuzione normale multivariata. A partire dalla matrice di espressione genica, che conta 6413 cellule sulle righe e 53 geni sulle colonne, è stata stimata un'unica matrice di precisione $\hat{\Theta}$ di dimensioni 53×53 . L'assunzione che è stata fatta alla base del modello è che le 6413 cellule fossero osservazioni indipendenti ed identicamente distribuite provenienti da una distribuzione normale multivariata 53-dimensionale; tale assunzione, però, non può essere verificata. Infatti, come spiegato dettagliatamente nel Capitolo 2, le 6413 cellule sono state ordinate secondo una traiettoria pseudotemporale, tale per cui esse sono state disposte dalla più immatura alla più matura. Di conseguenza, le osservazioni non possono essere assunte indipendenti poichè sono riportate in un certo ordine, secondo lo pseudotempo, che ordina le cellule secondo un indice pseudotemporale $t = 1, \dots, 6413$. Sotto quest'ottica, la matrice dei

dati segue una distribuzione normale multivariata 53-dimensionale, dove i parametri μ e Σ cambiano ad ogni istante pseudotemporale, quindi

$$\mathbf{X} \sim N_{53}(\mu_t, \Sigma_t), \quad t = 1, \dots, 6413, \quad (4.1)$$

o equivalentemente

$$\mathbf{X} \sim N_{53}(\mu_t, \Theta_t^{-1}), \quad t = 1, \dots, 6413. \quad (4.2)$$

Rispetto al caso i.i.d. quindi, il problema diventa nettamente più oneroso computazionalmente. Tralasciando la media, se prima i parametri da stimare erano $\frac{p(p-1)}{2} = 1378$, pur assumendo la sparsità della matrice Θ , nel caso dinamico i parametri da stimare diventano $\frac{np(p-1)}{2} = 8837114$. A bilanciare tale evidente aumento di difficoltà computazionale c'è la possibilità di andare a studiare momento per momento se e come cambia il sistema di connessione dei geni al variare del livello di maturità cellulare.

4.2 Indici descrittivi

Riprendendo la definizione riportata nel Paragrafo 3.2, un grafo $G = (V, E)$ è una coppia di insiemi disgiunto dove $V = \{1, \dots, N\}$ è l'insieme dei vertici (o nodi) del grafo, mentre $E \subseteq \{\{i, j\} : i, j \in V\}$ è l'insieme degli archi del grafo. Una rappresentazione alternativa è quella della matrice di adiacenza Y , matrice quadrata simmetrica di dimensione $N \times N$, in cui i vertici sono disposti in riga e colonna e vale $Y_{ij} = Y_{ji} = 1$ se $\{i, j\} \in E$ e $Y_{ij} = Y_{ji} = 0$ altrimenti. Esistono alcuni indici descrittivi che permettono di delineare in termini quantitativi le caratteristiche di un grafo, sia globalmente che a livello di singolo vertice o gruppi di vertici.

- Il grado di un nodo è il numero di archi con cui è connesso:

$$v_i = \sum_{j=1}^N Y_{ij}, \quad (4.3)$$

e viene normalizzato con

$$\tilde{v}_i = \frac{v_i}{(N-1)} \quad (4.4)$$

per effettuare confronti.

- La *betweenness* b_i di un nodo i è una sua misura di centralità basata sugli *shortest path*. Nella teoria dei grafi, per ogni coppia di nodi u e v gli *shortest paths* sono i cammini più corti tra nodi interconnessi che uniscono u a v . La *betweenness* di un nodo i è calcolata come la somma, fatta su tutte le coppie di nodi u e v diverse da i del rapporto tra il numero degli *shortest paths* tra u e v che passano per i , ossia $n_{uv}(i)$ ed il totale degli *shortest paths* tra u e v , ovvero n_{uv} :

$$b_i = \sum_{u \neq i \neq v} \frac{n_{uv}(i)}{n_{uv}}. \quad (4.5)$$

Per fare in modo che $b_i \in [0, 1]$, per i grafi non orientati si usa normalizzarlo con

$$\tilde{b}_i = \frac{2b_i}{(N-1)(N-2)}. \quad (4.6)$$

Alta *betweenness* comporta un ruolo decisivo nella struttura della rete.

- La *closeness* c_i di un nodo i è una misura della sua vicinanza agli altri nodi nel grafo; un valore alto di *closeness* per un nodo indica che è facilmente raggiungibile dagli altri nodi nel grafo. Viene definita come

$$c_i = \frac{1}{\sum_{j=1}^N s_{ij}}, \quad (4.7)$$

dove s_{ij} è la lunghezza dello *shortest path* tra i e j . Viene solitamente normalizzata con

$$\tilde{c}_i = c_i(N-1). \quad (4.8)$$

- La densità di Y è definita come la frequenza relativa del numero totale di archi osservati, sul totale degli archi possibili; nel caso di grafi non orientati è pari a

$$D = \frac{2E}{N(N-1)}, \quad (4.9)$$

dove $E = \sum Y_{ij}$ è il numero totale di archi del grafo.

- La distribuzione del grado $p(k)$ di un grafo è definita come la probabilità che un nodo abbia grado k e si scrive come la frazione di nodi del grafo che ha grado k , ovvero

$$p(k) = \frac{N_k}{N}, \quad (4.10)$$

dove N_k è il numero di nodi del grafo con grado k . Tale distribuzione ha media

$$\mu = E[p(k)] = \sum_k k \cdot p(k) \quad (4.11)$$

detta grado medio, e varianza

$$\sigma^2 = V[p(k)] = \sum_k (k - \mu)^2 \cdot p(k). \quad (4.12)$$

Si faccia presente inoltre che, in un grafo non orientato, il grado medio si può scrivere anche come

$$\mu = \frac{2E}{N} \quad (4.13)$$

e presenta una relazione lineare con la densità in quanto

$$D = \frac{2E}{N(N-1)} = \frac{\mu}{(N-1)}. \quad (4.14)$$

- Il diametro d di un grafo è la lunghezza del suo più lungo *shortest path*.
- La *transitivity* t di un grafo, detta anche *clustering coefficient* è una misura della tendenza dei nodi a raggrupparsi insieme. Nei grafi non orientati è definita come il rapporto tra il numero di triplette aperte e le triplette chiuse. Nella teoria dei grafi una tripletta è un insieme di tre nodi connessi da due archi (tripletta aperta) o da tre archi (tripletta chiusa o triangolo). Alta transitivity significa che la rete contiene comunità o gruppi di nodi che sono densamente connessi internamente.

4.3 Analisi separata per tipo cellulare

Come preliminare analisi dinamica si è deciso di stimare 4 matrici di precisione in corrispondenza dei 4 diversi tipi cellulari. Come illustrato in

(Figura 2.4), l'algoritmo Slingshot è stato in grado di ordinare le 6413 cellule secondo un ordine pseudotemporale di maturazione: grazie a tale ordinamento, si può dire con un certo grado di precisione che le cellule dalla prima alla 1900-esima sono di tipo HBC, dalla 1901-esima alla 2600-esima sono di tipo GBC, dalla 2601-esima alla 5300-esima sono di tipo iOSN e dalla 5301-esima alla 6413-esima sono di tipo mOSN. Volendo stimare differentemente la struttura di precisione in questi 4 gruppi, le numerosità campionarie saranno rispettivamente 1900, 700, 2700 e 1113. L'algoritmo ADMM-Glasso presentato nel Capitolo precedente è stato quindi applicato alle 4 matrici di covarianze empiriche ottenute dalla suddivisione appena descritte, ottenendo 4 stime distinte di Θ , chiamate $\hat{\Theta}_{HBC}$, $\hat{\Theta}_{GBC}$, $\hat{\Theta}_{iOSN}$ e $\hat{\Theta}_{mOSN}$. In Figura 4.1 viene riportato l'andamento del criterio BIC al variare del parametro λ per i quattro casi differenti; i valori ottimi scelti per λ sono stati rispettivamente 0.086, 0.108, 0.108, 0.108, in base al quale vengono stimati rispettivamente 9, 25, 48 e 31 archi.

In Figura 4.2 vengono raffigurati i 4 grafi corrispondenti alle 4 stime della matrice di precisione; si vede molto bene come all'aumentare del grado di maturazione delle cellule aumenta anche la densità del grafo. Il picco di densità sembra giungere in concomitanza dei neuroni immaturi e negli ultimi due grafi è possibile inoltre notare la presenza di alcuni geni con più connessioni degli altri, segno che è possibile il loro coinvolgimento nel processo biologico preso in esame.

In Tabella 4.1 vengono riassunte alcune misure descrittive dei 4 grafi. Per ognuno di essi sono state calcolate la densità D , la media e varianza della distribuzione del grado (μ e σ^2), il diametro d , la *transitivity* t , il gene con maggior grado normalizzato $MAX(\tilde{v}_i)$, quello con maggior *betweenness* normalizzata $MAX(\tilde{b}_i)$ e quello con maggiore *closeness* normalizzata $MAX(\tilde{c}_i)$. Come era lecito aspettarsi la densità, così come la varianza, raggiunge il suo massimo in corrispondenza del terzo grafo, relativo ai neuroni immaturi. Il diametro rimane circa costante, mentre la *transitivity* è l'unico indice a livello di grafo globale che raggiunge il suo picco massimo nel primo grafo. I geni più importanti sembrano essere rispettivamente *Ccnd2* e *Ccnd1* per i primi due gruppi cellulari e *Jun* per gli ultimi due. La densità è espressa in percentuale.

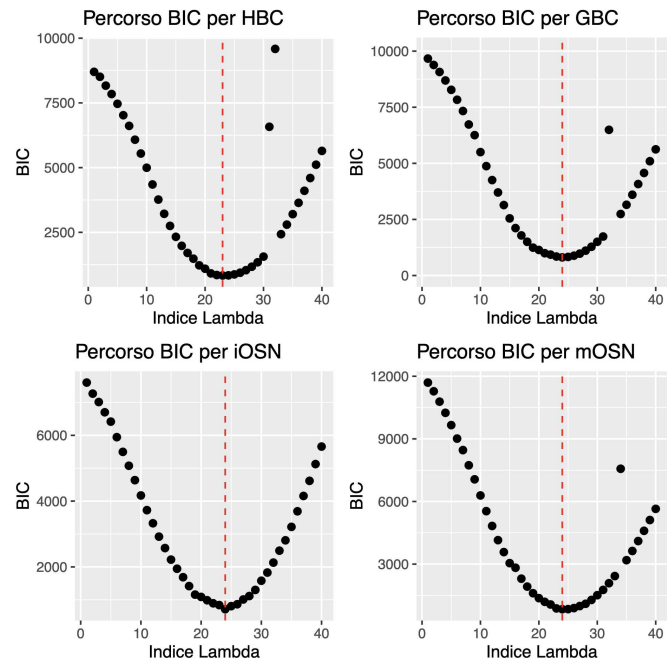


Figura 4.1: Andamento indice BIC per i quattro gruppi cellulari.

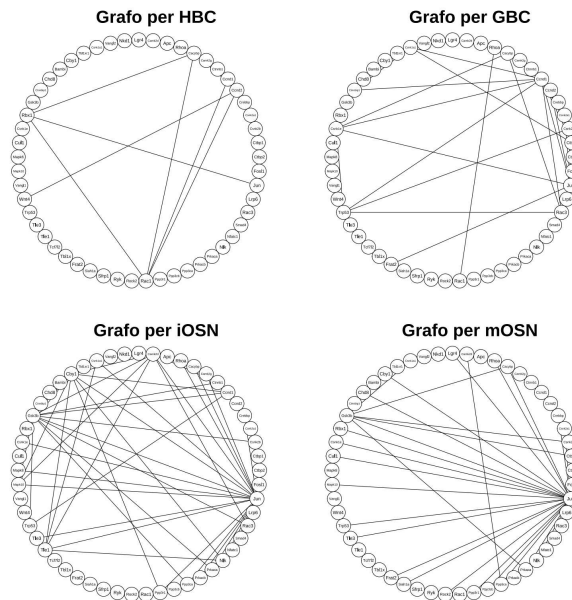


Figura 4.2: Grafi stimati per i quattro gruppi cellulari.

Tabella 4.1: Indici descrittivi per i quattro gruppi cellulari.

	$\widehat{\Theta}_{HBC}$	$\widehat{\Theta}_{GBC}$	$\widehat{\Theta}_{iOSN}$	$\widehat{\Theta}_{mOSN}$
D	0.653	1.814	3.483	2.249
μ	0.339	0.943	1.811	1.169
σ^2	0.254	0.259	2.316	0.632
d	4	3	3	4
t	0.473	0.336	0.24	0.04
$MAX(\tilde{v}_i)$	<i>Ccnd2</i>	<i>Ccnd1</i>	<i>Jun</i>	<i>Jun</i>
$MAX(\tilde{b}_i)$	<i>Ccnd2</i>	<i>Ccnd1</i>	<i>Jun</i>	<i>Jun</i>
$MAX(\tilde{c}_i)$	<i>Rac1</i>	<i>Rhoa</i>	<i>Jun</i>	<i>Jun</i>

4.4 Analisi dinamica via Rolling Window

Dalle analisi del precedente Paragrafo è emerso in maniera evidente che la struttura di dipendenza tra geni cambia allo scorrere dello pseudotempo, sia a livello di grafo globale che a livello di singoli nodi. Alla luce di questi risultati si è deciso di studiare il comportamento dinamico dei grafi in maniera più dettagliata. È stata applicata una tecnica di tipo Rolling Window, vale a dire una stima della matrice di precisione sparsa partendo da blocchi di dati successivi, dette finestre. Prendendo una serie di finestre consecutive in cui la finestra successiva esclude l'ultima osservazione della precedente ed include un'osservazione in più, è possibile notare in maniera precisa gli eventuali cambiamenti tra grafi consecutivi. Si è scelta un'ampiezza delle finestre pari a 1400 osservazioni ed avendo a disposizione 6413 cellule sono stati stimati 5014 grafi successivi. Per ognuno di essi poi sono stati calcolati i medesimi indici descrittivi del precedente Paragrafo, con i risultati mostrati in Figura 4.3. Le due linee tratteggiate verticali rappresentano la prima window che esclude le cellule HBC e la prima window che esclude le cellule GBC. I primi due grafici riferiscono un'evidente aumento della densità dei grafi e della varianza della distribuzione dei loro nodi in concomitanza delle finestre dalla 2000 alla 2500 circa; tale picco coincide con l'intervallo compreso tra le due rette verticali descritte precedentemente. Come era lecito aspettarsi dalle analisi del Paragrafo precedente, la *transitivity* presenta un picco iniziale ed

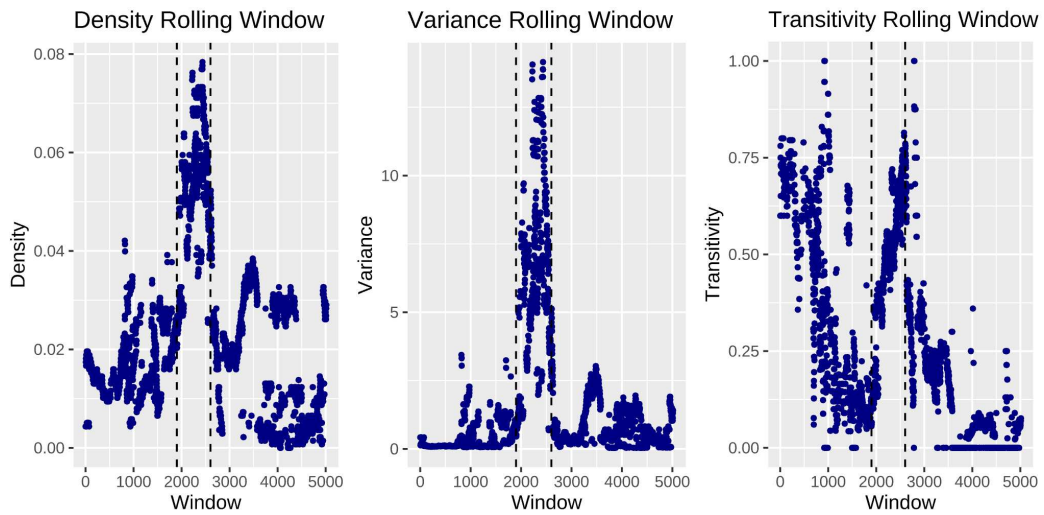


Figura 4.3: Andamento indici descrittivi a livello di grafo.

un secondo, più modesto, simile agli altri due indici.

Un secondo passaggio è stato quello di individuare i geni più importanti individuati nel precedente Paragrafo per vedere se e come cambiasse la loro importanza allo scorrere dello pseudotempo. Per ogni grafo sono stati quindi calcolati il grado normalizzato \tilde{v} , la *betweenness* normalizzata \tilde{b} e la *closeness* normalizzata \tilde{c} solo per alcuni geni (*CCnd2*, *Ccnd1*, *Jun*, *Rhoa*, *Rac1*); i risultati sono mostrati in Figura 4.4. Si vede piuttosto chiaramente che i tre geni hanno proprio picco di importanza in tre istanti pseudotemporalmente differenti; il primo gene *Ccnd2* è molto importante nei primi grafi, che includono anche le cellule HBC. Il secondo gene *Ccnd1* presenta un picco sovrapponibile a quello in Figura 4.3, mentre *Jun* sembra essere molto importante alla fine, quando le finestre non includono più neanche le cellule GBC e comprendono quindi solo neuroni. Ulteriori grafici sono presenti in Appendice A.

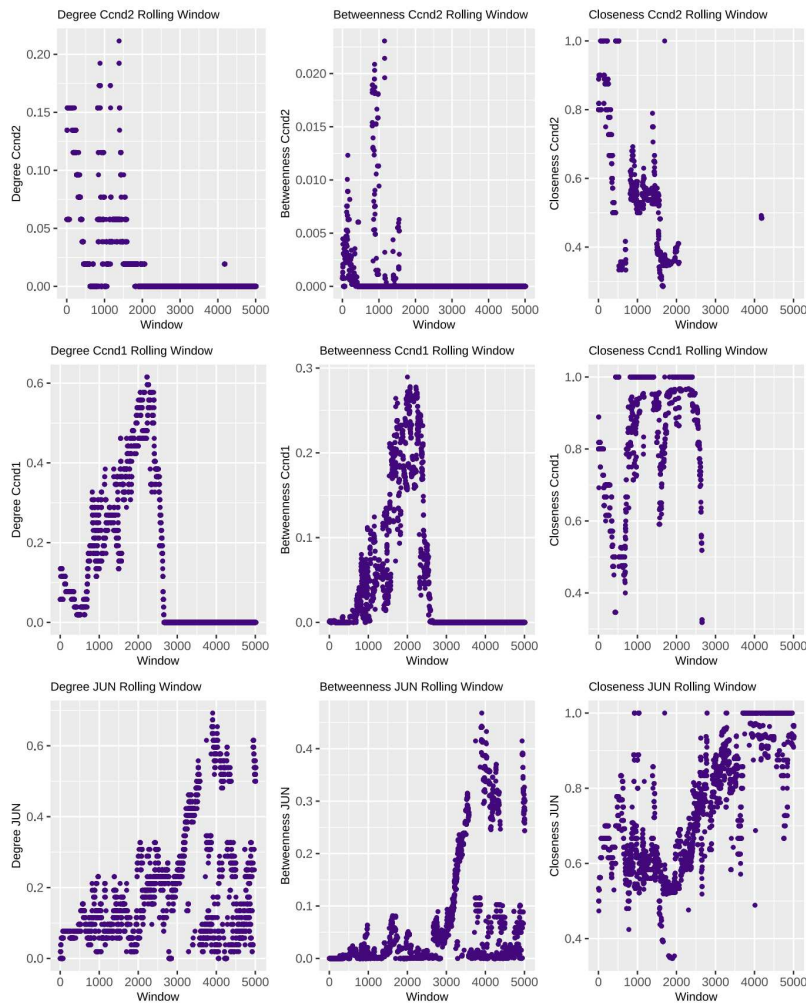


Figura 4.4: Andamento indici descrittivi dei geni più importanti.

4.5 Interpretazione biologica e conclusioni

Dalle analisi del capitolo precedente sono emersi cinque geni che sembrano essere i più importanti nel processo di maturazione delle cellule staminali: *Cnd2*, *Cnd1*, *Jun*, *Rac1*, *Rhoa*. Di questi, si è fatta una rassegna della letteratura scientifica in merito ad un loro possibile coinvolgimento nel fenomeno biologico per confermare o confutare i risultati trovati finora. Si ricordi che, come spiegato dettagliatamente nel corso del Capitolo 2, il

processo biologico studiato in questa Tesi è quello della differenziazione delle cellule staminali da immature a cellule neuronali mature; questo processo in biologia è detto neurogenesi.

I geni *Ccnd1* e *Ccnd2*, i cui nomi completi sono rispettivamente *CyclinD1* *CyclinD2*, sono dei geni codificanti appartenenti alla famiglia *CyclinsD* che svolgono un ruolo chiave nel ciclo cellulare. Uno studio (Artegiani et al. (2011)) ha studiato il processo di neurogenesi per quanto riguarda le cellule staminali neurali (*neural stem cells, NSCs*) del cervello di topi adulti. Quello che è stato trovato è che un aumento significativo dell'espressione del gene *Ccnd1* è associato alla proliferazione dei neuroni. Tale risultato può essere la motivazione tale per cui nel Paragrafo precedente il gene assume molta importanza nel passaggio tra cellule HBC e neuroni immaturi. Questo risultato sembra essere confermato anche da un altro studio (Lukaszewicz and Anderson (2011)), in cui è stato scoperto che *Ccnd1*, a differenza di *Ccnd2*, è espresso nelle cellule neuronali mature del midollo spinale degli embrioni di topo. Tuttavia sembra emergere anche un'importanza del gene *Ccnd2*, come in un articolo (Jedynaka et al. (2011)) che evidenzia la rilevanza di tale gene nel processo di neurogenesi delle cellule di cervello di topo e della comprensione del funzionamento dell'ippocampo. *Jun* è un proto-oncogene, ossia un gene coinvolto in genere nei processi di controllo della crescita cellulare e, in caso di lesioni genetiche a loro carico, può diventare oncogene, modificando la propria funzione originale e dando il via ai processi di sviluppo del cancro. Agisce a monte o all'interno di diversi processi, tra cui lo sviluppo di organi animali e codifica il fattore di trascrizione *Jun*. In uno studio del 2017 (Kawashima et al. (2017)) si è voluto studiare il ruolo di tale gene nella proliferazione delle *NSCs* del topo; non solo *Jun* gioca effettivamente un ruolo importante in questo processo ma entra in molte funzioni biologiche. Inoltre, è stata riscontrata una sua espressione molto alta anche nelle cellule adulte mature, e questo può essere un motivo per cui il picco di importanza di tale gene superava quello degli altri geni. Per quanto riguarda infine il gene *Rac1* (nome completo *RacFamilySmallGTPase1*) in uno studio (Haditsch et al. (2013)) si è voluto studiare il ruolo di tale gene nel processo di neurogenesi negli ippocampi di topi adulti. Si scopre che l'espressione di *Rac1* è strettamente necessario per la proliferazione delle cellule neurali.

Da queste analisi si è trovato quindi un riscontro di quanto descritto nel precedente Paragrafo circa l'importanza di alcuni geni nel processo di neurogenesi. Ulteriori studi biologici potranno essere svolti per confermare tale ipotesi scientifica.

Appendice A

Figure aggiuntive

In questa Appendice sono riportate alcune figure aggiuntive riferite al Capitolo 4.

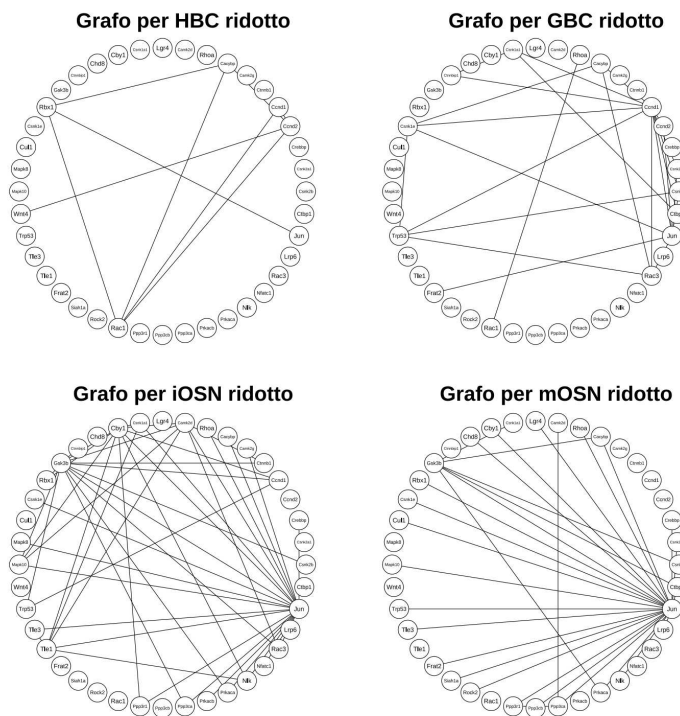


Figura A.1: Grafi stimati per quattro i gruppi cellulari usando solo i geni comuni.

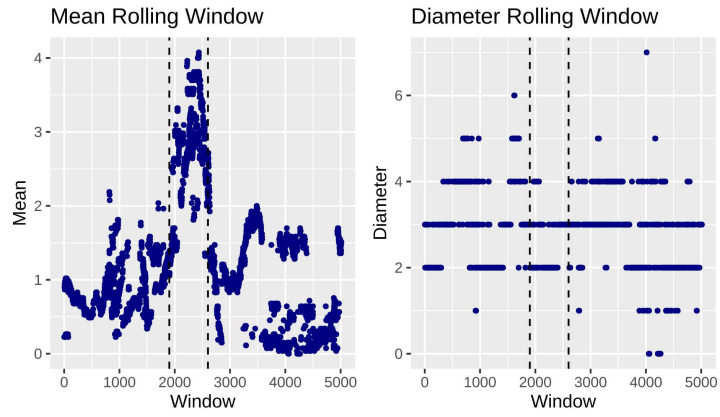


Figura A.2: Andamento indici descrittivi a livello di grafo.

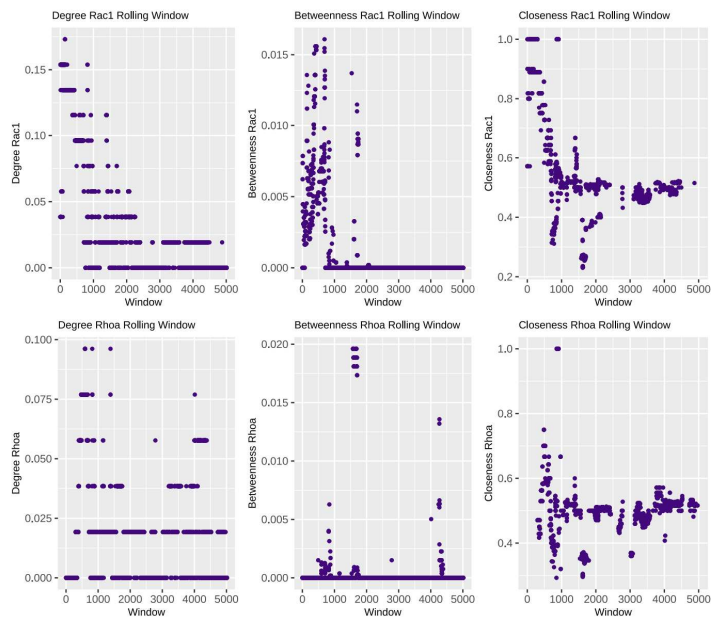


Figura A.3: Andamento indici descrittivi dei geni più importanti.

Appendice B

Codice C++ degli algoritmi presentati

In questa appendice viene riportato il codice C++ dell'algoritmo ADMM graphical-LASSO presentato nel Capitolo 3.

Codice B.1: Algoritmo ADMM per graphical-LASSO.

```
/* Definizione della proxy del Graphical-LASSO */
arma::mat glasso_proxy(arma::mat A, double rho) {
    arma::vec d;
    arma::mat Q;
    arma::eig_sym(d,Q,A);
    arma::mat L;
    arma::vec values;
    values=(d+arma::sqrt(d*d+4*rho))/(2*rho);
    L=Q*diagmat(values)*Q.t();
    return L;
}

/* Definizione del soft-tresholding scalare */
double scalar_soft(const double a, const double kappa) {
    double y=0;

    if (a-kappa > 0.0) {
        y = a-kappa;
    }
    else if (-a-kappa > 0.0) {
        y = a + kappa;
    }
    return y;
}
```

```

/* Definizione del soft-tresholding matriciale */
arma::mat matr_soft(arma::mat A, double kappa) {
    const int nrows = A.n_rows;
    const int ncols = A.n_cols;
    arma::mat AA(nrows, ncols, fill::zeros);

    for (int i=0; i<nrows; i++) {
        for (j = 0; j < ncols; ++j) {
            AA(i,j)=scalar_soft(A(i,j),kappa);
        }
    }
    return(AA);
}

/* Definizione della funzione obiettivo del Graphical-LASSO */
double glasso_objfun(arma::mat Theta, arma::mat S, const double lambda) {
    const int m = S.n_rows;
    return(0.5 * log(det(Theta))/ static_cast<float>(m)- 0.5 * trace(S*Theta)/ static_cast<float>(m)-
           lambda * norm(Theta, 1));
}

# S          : matrice di varianza e covarianza campionaria
# rho        : variabile per la convergenza dell'algoritmo
# lambda     : parametro di regolazione
# maxiter    : numero massimo di iterazioni
# alpha      : parametro per l'adattivita' dell'algoritmo
# tau        : parametro per l'adattivita' dell'algoritmo
# reltol     : soglia per l'errore relativo
# abstol     : soglia per l'errore assoluto
# rho_adaptation : algoritmo con adaptation

/* Definizione dell algoritmo ADMM per Graphical-LASSO */
Rcpp::List admm_glasso(const arma::mat& S, double rho, const double lambda, const int maxiter,
                      const double alpha, const double tau, const double reltol, const double abstol
                      , bool rho_adaptation){

    /* Definizioni variabili */
    const int n = S.n_rows;
    const double sqrtn = std::sqrt(static_cast<float>(n));
    int k=0;
    double elTime=0;
    double rho_old=0;

    /* Definizione di vettori e matrici */
    arma::vec h_objval(maxiter, fill::zeros);
    arma::vec h_r_norm(maxiter, fill::zeros);
    arma::vec h_s_norm(maxiter, fill::zeros);
    arma::vec h_eps_pri(maxiter, fill::zeros);
    arma::vec h_eps_dual(maxiter, fill::zeros);

```



```

arma::vec rho_store(maxiter+1, fill::zeros);
arma:: mat Theta=arma::eye(n,n);
arma:: mat z=arma::eye(n,n);
arma:: mat z_old=arma::eye(n,n);
arma:: mat u=arma::eye(n,n);

/* Tempo computazionale */
wall_clock timer;
timer.tic();

/* rho per caso adaptation */
rho_store(0)=rho;
rho_old=rho;

if (S.is_symmetric()) {
  /* Ciclo principale */
  for (int k=0; k<maxiter; k++) {
    /* Aggiornamento per Theta */
    Theta=glasso_proxy(rho*(z-u)-S, rho);

    /* Aggiornamento per Z */
    z_old = z;
    z = matr_soft(Theta + u, lambda/rho);

    /* Aggiornamento variabile duale */
    u = u + (Theta - z);

    /* Diagnostica */
    h_objval(k) = glasso_objfun(Theta, S, lambda);
    h_r_norm(k) = norm(Theta - z, 2);
    h_s_norm(k) = norm(-rho * (z - z_old), 2);
    if (norm(Theta, 2) > norm(-z, 2)) {
      h_eps_pri(k) = sqrtn * abstol + reltol * norm(Theta, 2);
    } else {
      h_eps_pri(k) = sqrtn * abstol + reltol * norm(-z, 2);
    }
    h_eps_dual(k) = sqrtn * abstol + reltol * norm(rho * u, 2);

    /* adaptation */
    rho_old = rho;
    if (rho_adaptation) {
      if (h_r_norm(k) > alpha * h_s_norm(k)) {
        rho = rho * tau;
        u = u / tau;
      } else if (h_s_norm(k) > alpha * h_r_norm(k)) {
        rho=rho/tau;
        u = u * tau;
      }
    }
  }

  rho_store(k+1) = rho;

```

```
        /* Criterio di fermata */
        if ((h_r_norm(k) < h_eps_pri(k)) && (h_s_norm(k) < h_eps_dual(k))) {
            break;
        }
    }

    /* Tempo per la stima */
    elTime = timer.toc();

    /* Risultati */
    Rcpp::List output;

    output["Theta"] = Theta;
    output["u"] = u;
    output["z"] = z;
    output["objval"] = h_objval;
    output["niter"] = k;
    output["r_norm"] = h_r_norm;
    output["s_norm"] = h_s_norm;
    output["eps_pri"] = h_eps_pri;
    output["eps_dual"] = h_eps_dual;
    if ((k+1) < maxiter) {
        output["convergence"] = 1.0;
    } else {
        output["convergence"] = 0.0;
    }
    output["rho"] = rho;
    output["rho_store"] = rho_store;
    output["etime"] = elTime;
    /* Output */
    return output;
}

else {
    Rcpp::stop("Error. Not symmetric matrix.");
}
}
```



```

niter_vet <- unlist(lapply(lista, function(list) {list[[5]]}))
convergence_vet <- unlist(lapply(lista, function(list) {list[[10]]}))
BIC_vet <- sapply(lapply(lista, function(list) {list[[1]]}),BIC_lambda)
sparsity_vet <- sapply(lapply(lista, function(list) {list[[1]]}),index)

lambda_opt <- lambdas[which.min(BIC_vet)]
sparsity_opt <- sparsity_vet[which.min(BIC_vet)]
Theta_opt <- lista[[which.min(BIC_vet)-2]]$Theta

output <- list(
  convergence_vet = convergence_vet,
  BIC_vet = BIC_vet,
  sparsity_vet = sparsity_vet,
  niter_vet = niter_vet,
  lambda_opt = lambda_opt,
  sparsity_opt = sparsity_opt,
  Theta_opt = Theta_opt)
return(output)
}

```

Codice C.2: Codice per il calcolo degli indici descrittivi di una lista di grafi.

```

compute_descriptives <- function(graph_list)
{
  result <- list()

  density <- sapply(graph_list, function(gr) {
    edge_density(gr)})
  mean <- sapply(graph_list, function(gr) {
    distr(gr)$mean})
  var <- sapply(graph_list, function(gr) {
    distr(gr)$var})
  diameter <- sapply(graph_list, function(gr) {
    diameter(gr, directed = FALSE)})
  transitivity <- sapply(graph_list, function(gr) {
    transitivity(gr,type="globalundirected")})
  genemaxdegree <- sapply(graph_list, function(graph) {
    names(degree(graph))[which.max(degree(graph, normalized=TRUE))])})
  genemaxbetweenness <- sapply(graph_list, function(graph) {
    names(degree(graph))[which.max(betweenness(graph, directed = FALSE, normalized=TRUE))])})
  genemaxcloseness <- sapply(graph_list, function(graph) {
    names(degree(graph))[which.max(closeness(graph, normalized=TRUE))])})
  degreeJUN <- sapply(graph_list, function(graph) {
    degree(graph,normalized=TRUE)[names(degree(graph))=="Jun"]})
  betweennessJUN <- sapply(graph_list, function(graph) {
    betweenness(graph,normalized=TRUE, directed = FALSE)[names(degree(graph))=="Jun"]})
  closenessJUN <- sapply(graph_list, function(graph) {
    closeness(graph,normalized=TRUE)[names(degree(graph))=="Jun"]})
  degreeCnd2 <- sapply(graph_list, function(graph) {
    degree(graph,normalized=TRUE)[names(degree(graph))=="Cnd2"]})
}

```

```
betweennessCcnd2 <- sapply(graph_list, function(graph) {
  betweenness(graph,normalized=TRUE, directed = FALSE)[names(degree(graph))=="Ccnd2"]}
)
closenessCcnd2 <- sapply(graph_list, function(graph) {
  closeness(graph,normalized=TRUE)[names(degree(graph))=="Ccnd2"]}
)
degreeCcnd1 <- sapply(graph_list, function(graph) {
  degree(graph,normalized=TRUE)[names(degree(graph))=="Ccnd1"]}
)
betweennessCcnd1 <- sapply(graph_list, function(graph) {
  betweenness(graph,normalized=TRUE, directed = FALSE)[names(degree(graph))=="Ccnd1"]}
)
closenessCcnd1 <- sapply(graph_list, function(graph) {
  closeness(graph,normalized=TRUE)[names(degree(graph))=="Ccnd1"]}
)
degreeRac1 <- sapply(graph_list, function(graph) {
  degree(graph,normalized=TRUE)[names(degree(graph))=="Rac1"]}
)
betweennessRac1 <- sapply(graph_list, function(graph) {
  betweenness(graph,normalized=TRUE, directed = FALSE)[names(degree(graph))=="Rac1"]}
)
closenessRac1 <- sapply(graph_list, function(graph) {
  closeness(graph,normalized=TRUE)[names(degree(graph))=="Rac1"]}
)
degreeRhoa <- sapply(graph_list, function(graph) {
  degree(graph,normalized=TRUE)[names(degree(graph))=="Rhoa"]}
)
betweennessRhoa <- sapply(graph_list, function(graph) {
  betweenness(graph,normalized=TRUE, directed = FALSE)[names(degree(graph))=="Rhoa"]}
)
closenessRhoa <- sapply(graph_list, function(graph) {
  closeness(graph,normalized=TRUE)[names(degree(graph))=="Rhoa"]}
)

result$density <- density
result$mean <- mean
result$var <- var
result$diameter <- diameter
result$transitivity <- transitivity
result$genemaxdegree <- genemaxdegree
result$genemaxbetweenness <- genemaxbetweenness
result$genemaxcloseness <- genemaxcloseness
result$degreeJUN <- degreeJUN
result$betweennessJUN <- betweennessJUN
result$closenessJUN <- closenessJUN
result$degreeCcnd2 <- degreeCcnd2
result$betweennessCcnd2 <- betweennessCcnd2
result$closenessCcnd2 <- closenessCcnd2
result$degreeCcnd1 <- degreeCcnd1
result$betweennessCcnd1 <- betweennessCcnd1
result$closenessCcnd1 <- closenessCcnd1
result$degreeRac1 <- degreeRac1
result$betweennessRac1 <- betweennessRac1
result$closenessRac1 <- closenessRac1
result$degreeRhoa <- degreeRhoa
result$betweennessRhoa <- betweennessRhoa
result$closenessRhoa <- closenessRhoa
return(result)
}
```


Bibliografia

- Ahlmann-Eltze, C. and Huber, W. (2022). Comparison of transformations for single-cell rna-seq data. *Nature methods*, 20:655–672.
- Artegiani, B., Lindemann, D., and Calegari, F. (2011). Overexpression of *cdk4* and *cyclind1* triggers greater expansion of neural stem cells in the adult mouse brain. *Journal of Experimental Medicine*, 5:937–948.
- Boyd, S., Parikh, N., Chu, E., et al. (2010). Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 6(31):1–122.
- Brann, D., Tsukahara, T., Weinreb, C., et al. (2020). Non-neuronal expression of sars-cov-2 entry genes in the olfactory system suggests mechanisms underlying covid-19–associated anosmia. *Science Advances*, 6(31).
- Campbell, K. R. and Yau, C. (2016). Order under uncertainty: Robust differential expression analysis using probabilistic models for pseudotime inference. *PLoS Computational Biology*, 12(11).
- Duggan, C. D. and Ngai, J. (2007). Scent of a stem cell. *Nature Neuroscience*, 10(6):673–4.
- Fletcher, R., Das, D., Gadye, L., et al. (2017). Deconstructing olfactory stem cell trajectories at single-cell resolution. *Cell Stem Cell*, 20:817–830.
- Friedman, J., Hastie, T., and Tibshirani, R. (2007). Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3):432–441.

- Gabay, D. and Mercier, B. (1976). A dual algorithm for the solution of nonlinear variational problems via finite element approximation. *Camp Maths. with Appls.*, 2:17–40.
- Gao, X., Pu, D. Q., Wu, Y., and Xu, H. (2012). Tuning parameter selection for penalized likelihood estimation of inverse covariance matrix. *Statistica Sinica*, 122:1123–1146.
- Glowinski, R. and Marocco (1975). Sur l’approximation, par éléments finis d’ordre un, et la résolution, par pénalisation-dualité d’une classe de problèmes de dirichlet non linéaires. *Revue française d’automatique, informatique, recherche opérationnelle. Analyse numérique*, 9:41–76.
- Haditsch, U., Anderson, M., et al. (2013). Neuronal *rac1* is required for learning-evoked neurogenesis. *The Journal of Neuroscience*, 33(30):12229–12241.
- Hastie, T. and Stuetzle, W. (1989). Principal curves. *Journal of the American Statistical Association*, 84(406):502–16.
- Hoerl, E. A. and Kennard, R. W. (1970). Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67.
- Jedynaka, P., Jaholkowsk, P., et al. (2011). Lack of cyclin d2 impairing adult brain neurogenesis alters hippocampal-dependent behavioral tasks without reducing learning ability. *Behavioural Brain Research*, 227:159–166.
- Kawashima, F., Saito, K., et al. (2017). *c-jun* is differentially expressed in embryonic and adult neural precursor cells. *Histochem Cell Biol*, 147:721–731.
- Lukaszewicz, A. and Anderson, D. (2011). Cyclin d1 promotes neurogenesis in the developing spinal cord in a cell cycle-independent manner. *Proceedings of the National Academy of Sciences*, 108.
- Powell, M. J. D. (1969). A method for nonlinear constraints in minimization problems. *Academic Press*.

-
- Street, K., Risso, D., Fletcher, R., et al. (2018). Slingshot: cell lineage and pseudotime inference for single-cell transcriptomics. *BMC Genomics*, 19:477.
- Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58(1):267–288.
- Trapnell, C., Cacchiarelli, D., et al. (2014). Pseudo-temporal ordering of individual cells reveals dynamics and regulators of cell fate decisions. *Nature Biotechnology*, 32(4).
- Wohlberg, B. (2017). Admm penalty parameter selection by residual balancing. *Cornell University*.
- Yuan, M. and Lin, Y. (2007). Model selection and estimation in the gaussian graphical model. *Biometrika*, 94:19–35.