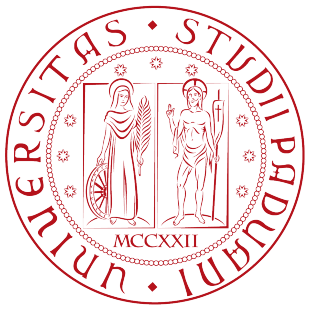
Università degli studi di Padova
Dipartimento di Scienze Statistiche
Corso di Laurea Magistrale in
Scienze Statistiche



Metodo di classificazione basato sulla topologia dei pathway biologici: il caso studio degli istotipi nel tumore ovarico

Relatore Prof.ssa Chiara Romualdi

Laureando: Giuseppe Arena
Matricola N 1130873

Anno Accademico 2016/2017

"Ma il guaio del caso Eichmann era che di uomini come lui ce n'erano tanti e che questi tanti non erano né perversi né sadici, bensí erano, e sono tuttora, terribilmente normali. Dal punto di vista delle nostre istituzioni giuridiche e dei nostri canoni etici, questa normalità è piú spaventosa di tutte le atrocità messe insieme, poiché implica – come già fu detto e ripetuto a Norimberga dagli imputati e dai loro patroni – che questo nuovo tipo di criminale, realmente hostis generis humani, commette i suoi crimini in circostanze che quasi gli impediscono di accorgersi o di sentire che agisce male."

(Hannah Arendt - La banalità del male)

Agli studenti che per colpa di un male diventato ormai banale hanno perso la vita, i sogni e i loro affetti.

Indice

Capitolo 1

Introduzione al caso studio e definizione degli obiettivi di ricerca

1.1 Il tumore ovarico

Il tumore all'ovaio consiste nella proliferazione non controllata delle cellule del parenchima ovarico. In particolare, la maggior parte delle volte tale rapida moltiplicazione avviene a carico delle cellule epiteliali, le quali non sono atte alla produzione di ovuli. In questi casi, tuttavia, possono risultare alterate le principali funzioni dell'organo stesso quali: la produzione di ormoni sessuali femminili e di ovociti. [10]

Come la maggior parte dei tumori, il carcinoma ovarico può presentarsi in due diverse forme: *benigno* o *maligno*. Il tumore benigno consiste in una cisti (ovarica) piena di materiale (liquido o solido), la quale può svilupparsi all'interno o sulla superficie dell'ovaio [9]. Invece, il tumore maligno si suddivide in tre principali tipologie a seconda delle cellule da cui esso origina: epiteliale (tessuto epiteliale che riveste la superficie delle ovaie), germinale (dalle cellule germinali che producono gli ovuli) e stromale (dallo stroma gonadico che costituisce il tessuto di sostegno dell'ovaio).

La classificazione del tumore ovarico è molto vasta e specifica ma il caso studio in esame si focalizzerà sui tumori maligni in fase precoce, i quali vengono distinti in quattro tipologie a seconda della loro istologia, ovvero in base a come le cellule che li costituiscono appaiono al microscopio [25]:

- **Clear Cell** (abbreviato in *"Cc"*): originano da cellule la cui struttura ed organizzazione richiamano le cellule del rene;
- **Endometrioide** (abbreviato in *"End"*): si formano da cellule simili a quelle del corpo uterino epiteliale/stromale;
- **Mucinoso** (abbreviato in *"Muc"*): aventi cellule che richiamano l'organizzazione delle cellule del tratto gastro-instestinale o dell'endocervice;
- **Sieroso** (abbreviato in *"Sier"*): hanno origine da cellule simili in organizzazione all'epitelio delle tube di falloppio.

Epidemioligia

Il tumore ovarico costituisce la settima tipologia di cancro maggiormente diagnosticata a livello mondiale. In Europa esso ricopre il 5% dei tumori femminili e si presenta con maggiore frequenza nella popolazione caucasica, in Europa nord-occidentale e negli Stati Uniti; meno frequente, invece, in Asia, Africa e Sud America.

In Italia, si trova al nono posto fra le forme tumorali presenti e rappresenta circa il 3% di tutte le diagnosi relative ai tumori. La diagnosi del tumore ovarico avviene il più delle volte dopo la menopausa. [10]

1.2 Dati e piattaforma in uso

Il caso studio riguarda l'espressione genica dei tessuti tumorali di 83 pazienti affette da tumore ovarico maligno in fase precoce. Le unità statistiche, quindi, saranno le 83 pazienti aventi età mediana all'ultimo follow-up di 63 anni (media 61 anni, deviazione standard 12.82 anni); in particolare, ciascuna paziente costituirà una replica biologica attraverso il corrispondente campione tumorale su cui viene misurata l'intensità d'espressione genica mediante la tecnologia dei *microarrays*. La piattaforma in uso è una piattaforma G4851B SurePrint G3 Human Gene Expression 8 x 60K v2 Microarray Kit (Agilent Technologies) le cui caratteristiche vengono specificate in Tabella 1.1. I dati di espressione saranno raccolti in una matrice di dimensioni $p \times n$, dove p è il numero di probes (sonde) dell'array, le quali rappresentano una parte del

Piattaforma G4851B	
Title	Agilent-014850 Whole Human Genome Microarray 4x44K G4112F (Feature Number version)
Tecnologia	oligonucleotide in situ
Distribuzione	uso commerciale
Organismo	Homo sapiens
Produttore	Agilent Technologies
Specifiche tecniche	
Agilent Product Number	G4851B
Design ID	039494
Format	8 x 60K
Arrays/Slide	8
Slides/Kit	3
Biological Features	50,599
Replicates of Biological Probes	900 x 10
Positive Controls	96 x 19 ERCC control probes 10 x 32 E10 spike-in control probes
Composition	Design based on: Ensemble, RefSeq, GenBank
Manufacturing	Agilent 60-mer SurePrint technology
Microarray Channel (singolo canale)	
Source name	tessuto ovarico
Organism	Homo sapiens
Caratteristica tessuto	tessuto tumorale
Molecola estratta	total RNA

Tabella 1.1: Informazioni sulla piattaforma Agilent utilizzata

trascritto di un gene (quindi per ogni gene esistono più sonde opportunamente disposte in modo ordinato sull'array) ed n è il numero di pazienti. Un esempio di matrice di espressione è possibile vederlo in Figura 1.1 dove accanto alle espressioni quantitative delle intensità di fluorescenza sono state accostate le due colonne relative agli identificativi dei probes (*ProbeName*) e dei geni (*GeneName*).

	A	B	C	D	E	F
1	ProbeName	GeneName	H252800415720_1_1	H252800415720_1_2	H252800415720_1_3	H252800415720_1_4
2	A_23_P326296	U2AF1L4	778	230	150,5	400
3	A_24_P287941	PSMC3IP	466	136,5	120	226,5
4	A_24_P325046	ZCCHC7	104	92	153,5	143
5	A_23_P200404	AK2	8673	864	2680	4444,5
6	A_19_P00800513	lincRNA:chr7:226042-232442_R	796	503	976	568
7	A_23_P15619	PRAC	59,5	53	55	58
8	A_33_P3402354	YLPM1	83	99	82,5	131
9	A_33_P3338798	SDR16C6	62,5	61,5	63	53,5
10	A_32_P98683	MLLT6	1683	2094,5	1114	2282
11	A_23_P137543	ZNF362	982	564	903	2584
12	A_19_P00803040	lincRNA:chr8:104254399-104295074_F	526	249,5	270	259,5
13	A_23_P117852	KIAA0101	6035	197	1567,5	487
14	A_33_P3285585	FLJ45256	68	57	61,5	63,5
15	A_24_P328231	CPSF3L	316	95	91,5	106
16	A_33_P3415668	LOC643923	65	80	51,5	56
17	A_23_P73609	NDP	223	83	110	116
18	A_24_P186124	MTERFD2	688,5	470	401	637
19	A_23_P369983	FAM98C	1242,5	715,5	420	382
20	ERCC-00071_128	ERCC-00071_128	70	63	62,5	70,5
21	ERCC-00142_99	ERCC-00142_99	70	73	63	69
22	A_23_P325676	ZNF653	277	153	155	184,5
23	A_24_P37441	PDK1	2203,5	402,5	1083	786,5
24	A_23_P20980	CYC1	12874	893	1487	1155
25	A_23_P109184	INSM1	64	55	54	53

Figura 1.1: Esempio matrice di espressione con accanto due colonne: identificativo probe (*ProbeName*) e identificativo gene (*GeneName*).

In quanto alle analisi dei dati di espressione, esse verranno condotte servendosi del programma di analisi statistica R [26] e dell'estensione Bioconductor [1]. Per quanto riguarda i grafici è stato utilizzato il pacchetto di R "ggplot2" [27].

1.3 Obiettivi di ricerca

Con i dati di espressione genica in possesso, l'obiettivo principale è quello di trovare dei pathway (o firme molecolari) che definiscono univocamente i quattro istotipi. Per perseguire tale obiettivo sono stati impiegati due approcci combinati:

- applicazione e valutazione delle performance di due nuovi metodi di analisi di pathway, i quali, al contrario delle tecniche classiche, cercano di caratterizzare il singolo campione senza alcuna informazione a priori;
- specificazione di un metodo di classificazione che lavori in modo non supervisionato basato su tecniche usate in contesti di reti sociali.

Nel Capitolo 2 verranno affrontate le fasi preliminari di elaborazione del dato di espressione (controllo qualità del dato, normalizzazione del dato, eliminazione di microarray con segnale di bassa qualità, identificazione dei geni differenzialmente espressi). Nel Capitolo 3 verrà trattata l'analisi di pathways attraverso due nuovi metodi (uno che lavora sui livelli delle densità, l'altro che opera secondo la topologia del pathway considerato). Il Capitolo 4 sarà una introduzione sulle reti di pazienti e sul metodo di stima della matrice di adiacenza. Nel Capitolo 5 si continuerà con l'applicazione di due metodi differenti: il primo di variational inference, il secondo un gibbs sampling; entrambi hanno l'obiettivo comune di identificare le quattro comunità di pazienti (le istologie tumorali nel caso studio). Il Capitolo 6 trarrà delle considerazioni finali circa i risultati ottenuti.

Capitolo 2

Analisi preliminare

Sulla piattaforma Agilent in uso viene ibridato il solo tessuto *tumorale* dello specifico paziente. Pertanto, nelle analisi di valutazione preliminare della qualità, ogni array sarà considerato come esperimento a *singolo canale*. In altri termini, le misure di intensità di espressione valutate a livello di probe saranno interpretate come misure di *espressione assoluta* della determinata sonda nel campione in esame.

L'analisi della qualità del dato di espressione grezzo che precede la normalizzazione dello stesso è finalizzata ad escludere dallo studio tutte quelle repliche biologiche che presentano: errori di lettura delle intensità, sovraesposizione o sottoesposizione dell'immagine risultante e altri artefatti che comprometterebbero l'affidabilità dei risultati, se non l'iter delle analisi stesse.

2.1 Controllo Qualità

Innanzi tutto, in ogni esperimento è possibile valutare come si distribuiscono le intensità relative a quelle *sonde di controllo* che risultano utili per comprendere se l'immagine risultante abbia sofferto o meno di artefatti probabilmente legati all'ambiente circostante in cui è avvenuta la scansione mediante laser. Tali sonde possono essere distinte in due tipologie: **controlli positivi** che misurano le intensità ad alti livelli, **controlli negativi** che misurano, invece, le intensità di background.

Dunque, quello che ci si aspetta da un esperimento andato a buon fine è che i controlli positivi e in controlli negativi abbiano una distribuzione rispettivamente sulle alte intensità e sulle basse intensità; inoltre, il campo di variazione dei controlli positivi deve essere distaccato da quello dei controlli negativi.

Considerando adesso gli 83 esperimenti, vengono riportati boxplot delle distribuzioni dei controlli negativi e positivi per ciascun array.

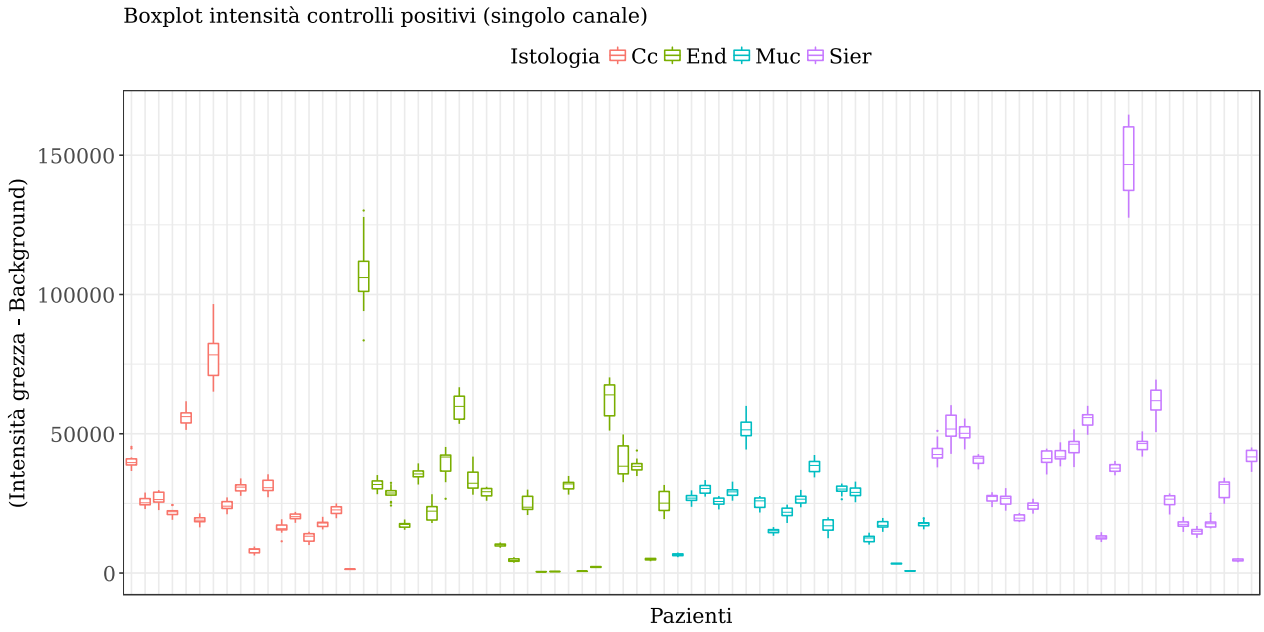


Figura 2.1: Boxplot intensità controlli positivi (singolo canale, 83 pazienti).

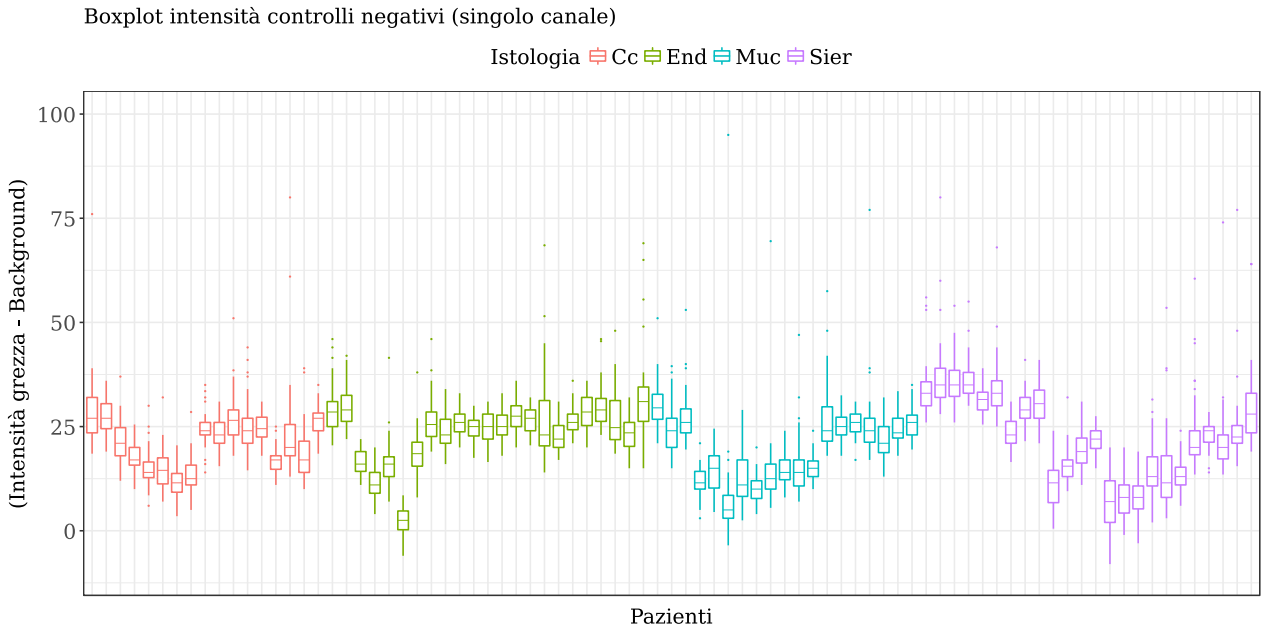


Figura 2.2: Boxplot intensità controlli negativi (singolo canale, 83 pazienti).

Dal grafico si osserva che i controlli negativi (di numerosità 35) hanno distribuzioni concentrate su valori bassi per tutte le repliche biologiche. I controlli positivi (di numerosità 14), invece, si distribuiscono su intensità di gran lunga più alte anche se alcuni arrays sembrano avere distribuzioni su basse intensità. Questo comportamento potrebbe essere dovuto anche alla basso numero di controlli.

Valutando i campi di variazione, è stato confermato, per ogni esperimento, che il livello

minimo di intensità dei controlli positivi risulta superiore al livello massimo di intensità dei controlli negativi.

Il dato di intensità grezza soffre della presenza di un rumore bianco di fondo che maschera una buona stima dell'informazione. Per tale motivo è necessaria una correzione dell'immagine che riesca a ripulire al meglio il segnale da questo rumore.

Un approccio comune consiste nella **sottrazione del background**. Dato il segnale del probe i nell'esperimento j, si è in possesso di due quantità: il segnale grezzo ($segnale+rumore$) X_{ij} e la stima del background ($rumore$) Z_{ij}. Secondo tale metodo, una stima migliorata del segnale (Y_{ij}) viene ricavata dalla semplice sottrazione del rumore al segnale grezzo.

$$Y_{ij} = X_{ij} - Z_{ij} \tag{2.1}$$

L'approccio è computazionalmente semplice ma causa spesso una perdita di informazione. Infatti, dopo la trasformazione, le analisi successive considerano sempre la trasformazione $log2$ delle intensità corrette e per tale ragione i valori minori o uguali a zero risulteranno come *dati mancanti*.

A dimostrazione di ciò, è stata: applicata la sottrazione del background agli 83 array, calcolato il $log2$ delle intensità risultanti e quantificata la percentuale di dati mancanti ottenuti dopo la trasformazione logaritmica.

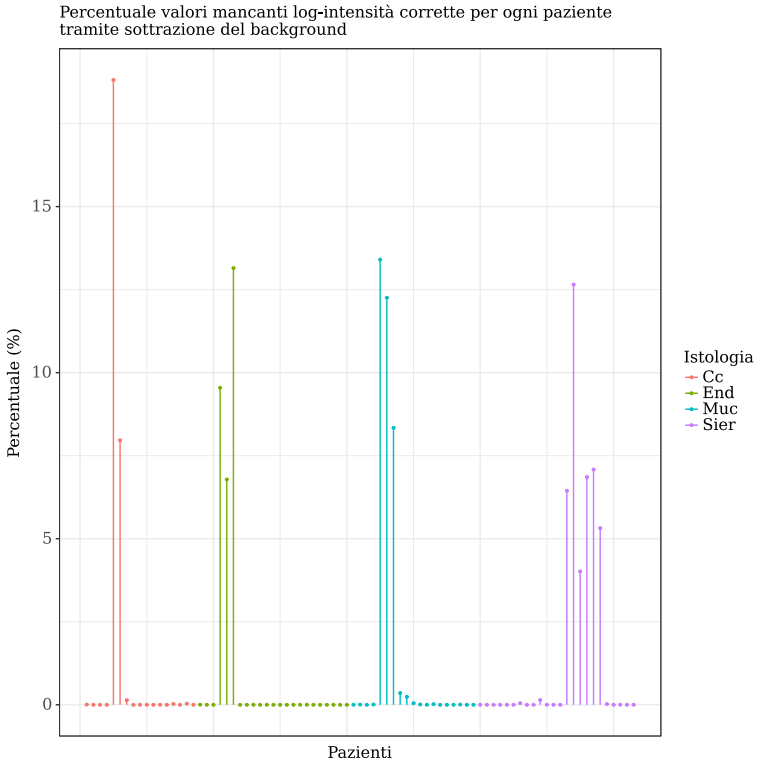


Figura 2.3: Percentuale dati mancanti log-intensità corrette tramite sottrazione del background.

Per tale ragione, è stato applicato il metodo di normalizzazione secondo **convoluzione normale-esponenziale** [22] il quale consiste nell'applicazione di un modello statistico-probabilistico all'intero vettore di intensità di un array. Secondo tale metodo, il segnale ottenuto nella (2.1) viene definito come segue,

$$Y_{ij} = B_{ij} + S_{ij} \tag{2.2}$$

ovvero come somma fra il background residuo (B_{ij}) non catturato da Z_{ij} e il vero valore di espressione S_{ij}. Inoltre, si assume che $B_{.j} \sim N(\mu_j, \sigma_j^2)$ e $S_{.j} \sim Exp(1/\alpha_j)$ e, quindi, la stima del segnale sarà data dal valore atteso condizionato $\mathbb{E}(S|Y = y)$. Mediante tale approccio non è stato perso alcun dato.

Un ulteriore fattore che permette di valutare la qualità di un esperimento è la percentuale di probe definiti *present* ovvero aventi una qualità molto soddisfacente. Si rappresenta, pertanto, la percentuale di elementi 'present' per ogni array.

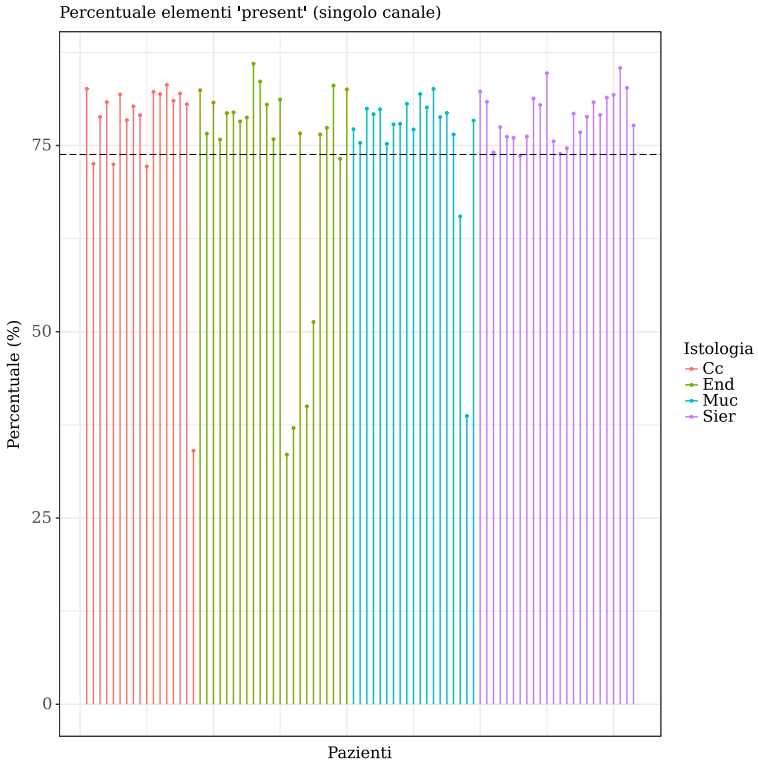


Figura 2.4: Percentuale di elementi 'present' (singolo canale).

La linea nera tratteggiata definisce il limite (73.8%) al di sotto del quale un esperimento è stato definito di qualità scadente ed è stato quindi eliminato dalle analisi. Tale livello è stato calcolato come l'estremo inferiore dell'intervallo di variabilità costruito intorno alla media e di ampiezza pari al 20% della variabilità (deviazione standard) della stessa misura.

In questo modo sono stati eliminati 7 delle 83 repliche biologiche e di conseguenza la numerosità del campione in esame è stata ridotta a 76.

Nelle pagine successive vengono riportati due dei grafici guida che permetteranno di valutare eventuali miglioramenti dovuti alle normalizzazioni. In particolare, si considereranno: i box-plot delle distribuzioni delle log-intensità e il grafico a dispersione delle prime due componenti principali. Le distribuzioni dei livelli di espressione devono essere confrontabili fra microarrays, cioè non molto diverse fra loro.

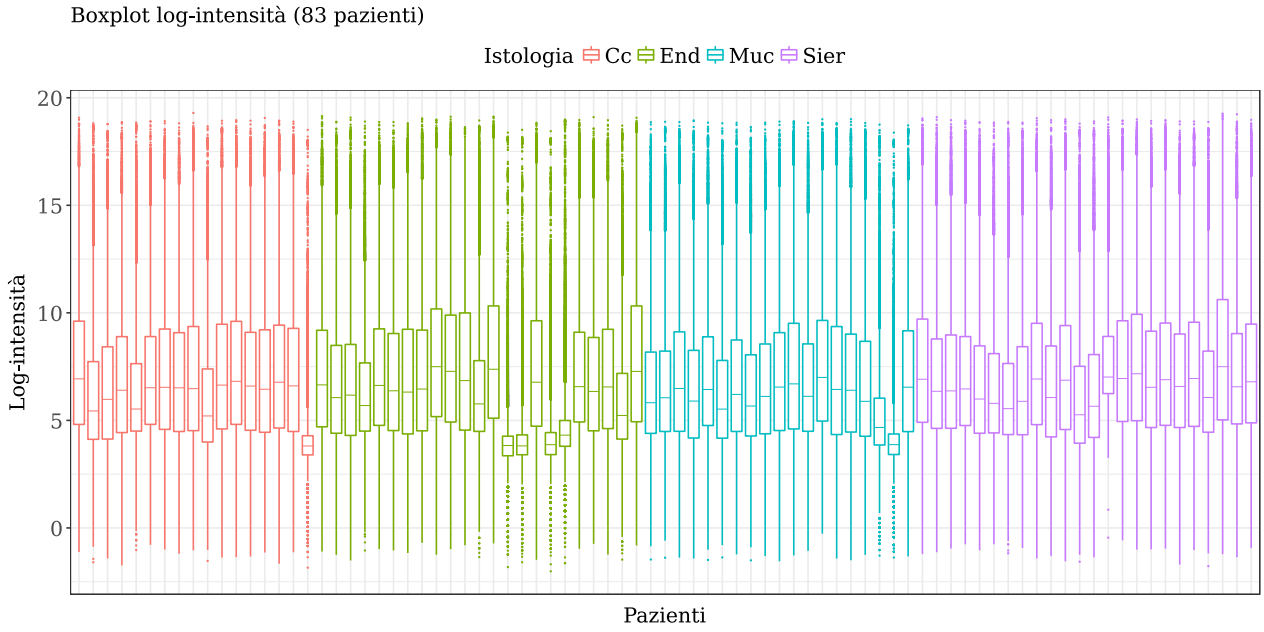


Figura 2.5: Boxplot log-intensità (83 pazienti).

La Figura 2.5 mostra come le distribuzioni delle log-intensità non siano molto simili fra gli array. La Figura 2.6 mette in evidenza un gruppo di repliche biologiche (a destra) che si

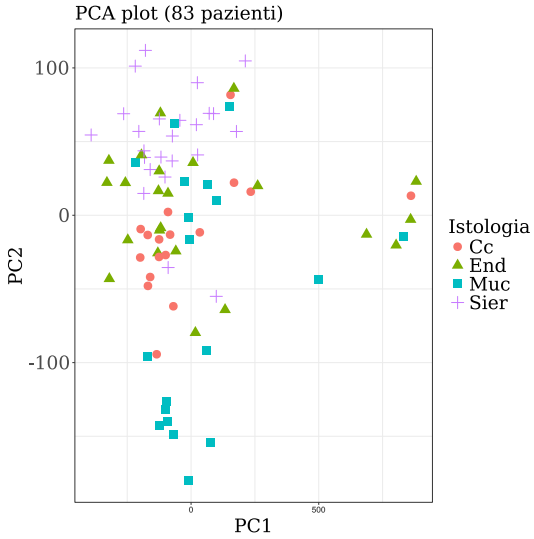


Figura 2.6: PCA plot (83 pazienti).

discostano dalle restanti. Quindi, è necessaria una normalizzazione del dato che permetta la confrontabilità fra gli esperimenti.

2.2 Normalizzazione dei dati

L'obiettivo in questa fase è quello di minimizzare l'interferenza di fattori come l'intensità del laser utilizzato per la scansione, dunque, di ridurre l'influenza sistematica nella lettura dell'immagine. Sui 76 microarray in analisi è stata applicata una normalizzazione tra array secondo il metodo della *loess ciclica "veloce" (cyclic loess fast)*. Di seguito, vengono riportati i due grafici relativi a detta normalizzazione.

Normalizzazione *tra array*: cyclic loess fast

La normalizzazione attraverso *cyclic loess fast* definisce un profilo di espressione fittizio (virtuale), il quale rappresenta l'espressione mediana di ogni gene nei 76 esperimenti. Per ciascuna replica biologica, rispetto a questo profilo viene effettuata una normalizzazione secondo il metodo *loess*.

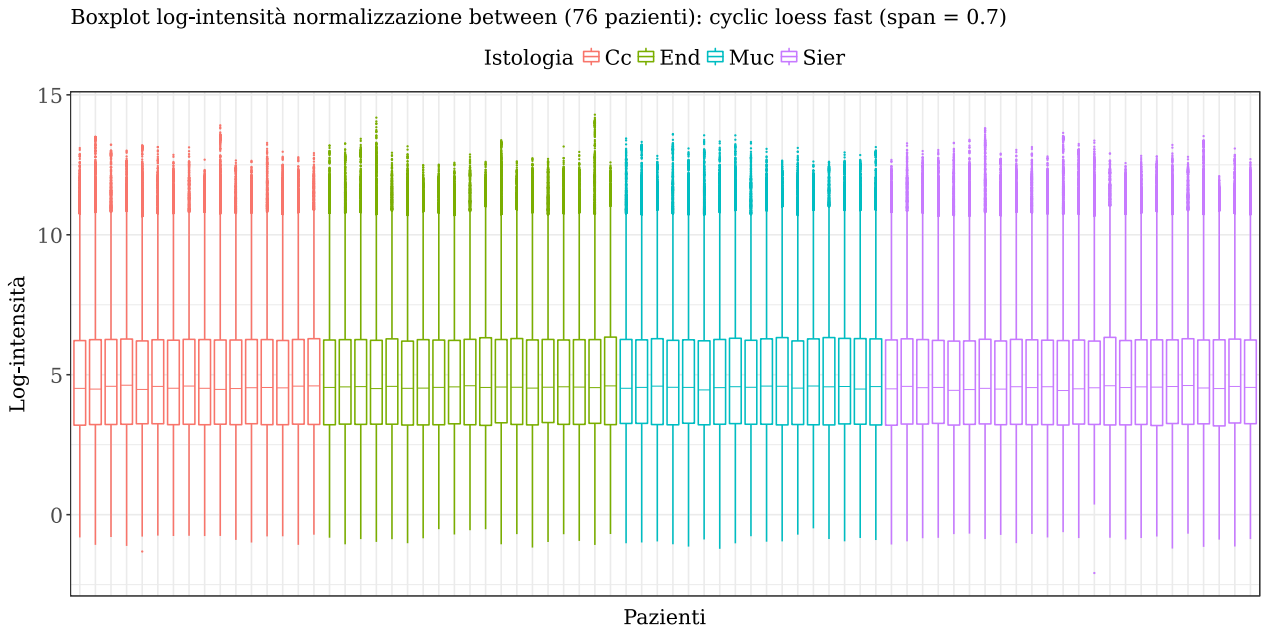


Figura 2.7: Boxplot log-intensità normalizzazione fra array (76 pazienti): cyclic loess fast (span = 0.7).

Le distribuzioni delle intensità risultanti dalla normalizzazione mediante cyclic loess si mostrano confrontabili e depurate dalla potenziale variabilità spuria. Inoltre, la maggiore comparabilità delle distribuzioni viene corroborata da un grafico delle prime due componenti principali più variabile ma soprattutto senza raggruppamenti isolati, lontano dai restanti.

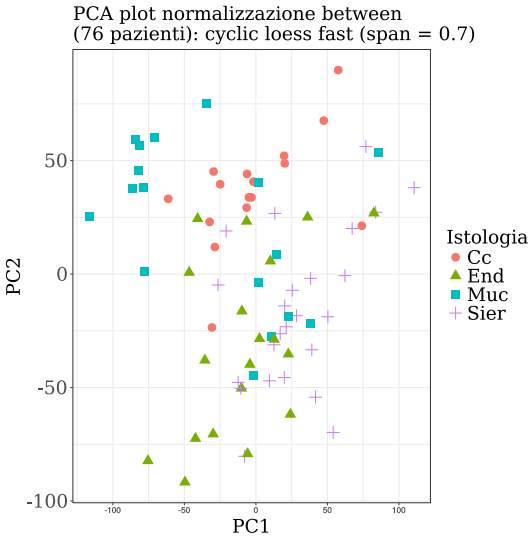


Figura 2.8: PCA plot normalizzazione array (76 pazienti) cyclic loess fast (span = 0.7).

2.3 Geni differenzialmente espressi (DEG)

Una volta normalizzato il dato, si procede alla valutazione della differenziale espressione a livello di probe fra le quattro istologie in esame (Clear Cell, Endometrioide, Mucinoso, Sieroso). Dunque, definendo un opportuno sistema d'ipotesi, si vuole verificare l'ipotesi nulla di uguaglianza delle medie di espressione contro l'ipotesi alternativa secondo la quale almeno uno dei 6 possibili confronti fra medie risulti significativo.

$$\begin{cases} H_0 : \mu_i = \mu_j \quad \forall i, j = 1, ..., 4 \\ H_1 : \mu_i \neq \mu_j \quad \text{per almeno una coppia} \quad i, j \end{cases}$$

Tale verifica d'ipotesi è stata condotta servendosi di più di una statistica test. In particolare, sono stati utilizzati il *test F* (parametrico), il *test di Welch* (parametrico), il *test di Kruskal-Wallis* (non parametrico, sulle mediane), il *test Empirical Bayes* (parametrico moderato) e il *test SAM* (parametrico moderato).

Inoltre, verranno eseguiti, per ciascuna tipologia di statistica test, tanti test quanti sono i probes del microarray (62,927). Successivamente è stata effettuata una correzione dei p-values secondo il metodo di Benjamini–Hochberg per procedere quindi con l'intersezione delle liste di probe differenzialmente espressi ricavate dall'applicazione di ciascuna statistica test. In questo modo, è stata ottenuta una lista di geni differenzialmente espressi nelle quattro condizioni del tumore ovarico e per i quali i risultati dei cinque test applicati sono stati fra loro concordanti in termini di significatività. Il livello di significatività nominale è stato fissato ad $\alpha = 0.05$.

Test F

Il *test F* confronta sostanzialmente due tipi di variabilità: fra i gruppi e interna ai gruppi. La statistica test costruita rifiuta l'ipotesi nulla di uguaglianza delle medie per valori emprici molto alti. In generale, il test per ogni probe, assume che vi sia omoschedasticità fra i gruppi e ha come distribuzione di riferimento sotto H_0 una F di Fisher-Snedecor con gradi di libertà (3,72) e zona di rifiuto $R \in (q_{F_{3,72},(1-\alpha)}, +\infty)$, dove $q_{F_{3,72},(1-\alpha)}$ è il quantile di livello $1-\alpha$ della distribuzione sotto H_0. Nel caso di $\alpha = 0.05$, $q_{F_{3,72},0.95} = 2.7318$.

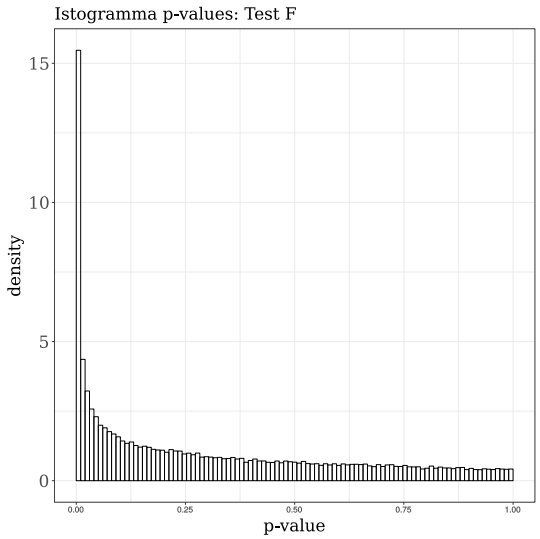


Figura 2.9: Istogramma p-values (non corretti): Test F.

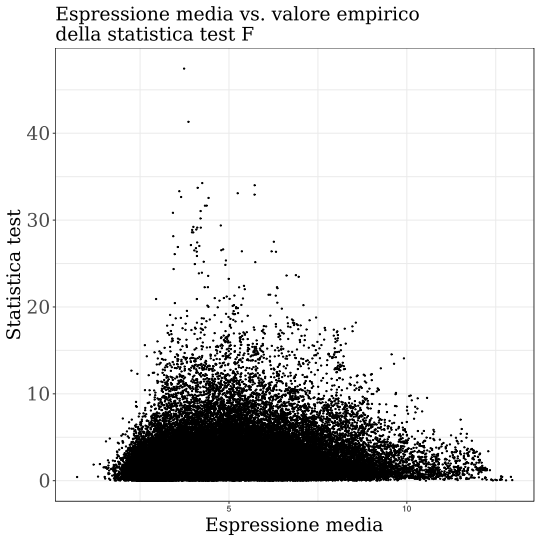


Figura 2.10: Espressione media vs. valore empirico della statistica test F.

La Figura 2.17 e la Figura 2.18 rappresentano rispettivamente l'istogramma dei p-values non corretti e il grafico a dispersione dell'espressione media del probe e il relativo valore empirico della statistica test. Quello che ci si aspetta è che l'istogramma sia fortemente asimmetrico, concentrato su valori bassi del pvalue ma che, tuttavia, esso presenti una distribuzione pressoché uniforme al crescere del valore dei pvalues (distribuzione teoricamente assunta sotto H_0); invece, il grafico a dispersione non deve mostrare evidenza di alcuna dipendenza fra le due misure. Nel caso del test F, entrambi i grafici risultano soddisfacenti. Il numero di probe per cui il test F è risultato significativo al 5% è pari a 5,148. In Tabella 2.1 vengono riportati i primi dieci geni[1] differenzialmente espressi fra le quattro condizioni.

[1]Il singolo *gene* a livello di microarray è rappresentato da più probes (ciascuno costituente una parte del suo trascritto) e, quindi, è possibile che fra i probes significativi risultino più sonde relative ad uno stesso gene.

Statistica test F: i primi 10 geni differenzialmente espressi		
Gene	T_F	**p-value**
LBP	47.442	4.9485E-17
FXYD2	34.013	8.3929E-14
F2	33.716	1.0077E-13
HAVCR1	33.325	1.2848E-13
PTP4A1	33.093	1.4843E-13
PTH1R	31.042	5.4675E-13
PTHLH	29.382	1.6279E-12
HNF1B	29.137	1.9172E-12
CTH	27.510	5.7986E-12
RBPMS	27.123	7.5846E-12

Tabella 2.1: Elenco dei primi 10 geni differenzialmente espressi secondo la statistica test F (nome del gene, valore empirico della statistica test (T_F) e p-values non corretti).

Test di Welch

Il *test di Welch* è simile ad un test T ma con assunzione di eteroschedasticità. Per questo motivo i gradi di libertà della distribuzione di riferimento F sotto H_0 non sono sempre gli stessi per ogni probe.

Pur non assumendo l'omoschedasticità fra i gruppi, il test conduce a risultati paragonabili a quelli ottenuti con il test F (si vedano la Figura 2.19 e la Figura 2.20). Inoltre, il numero di

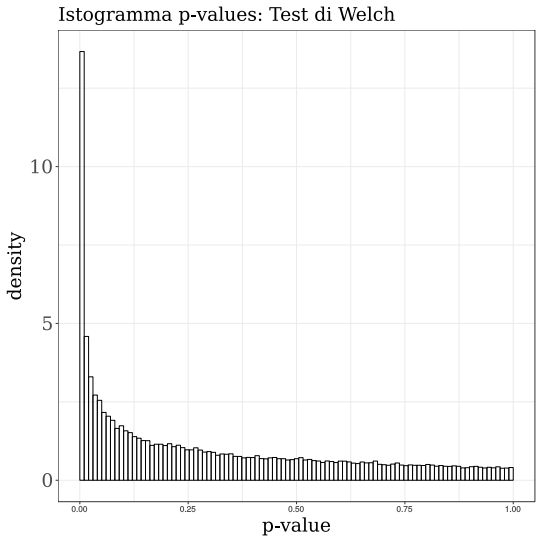


Figura 2.11: Istogramma p-values (non corretti): Test di Welch.

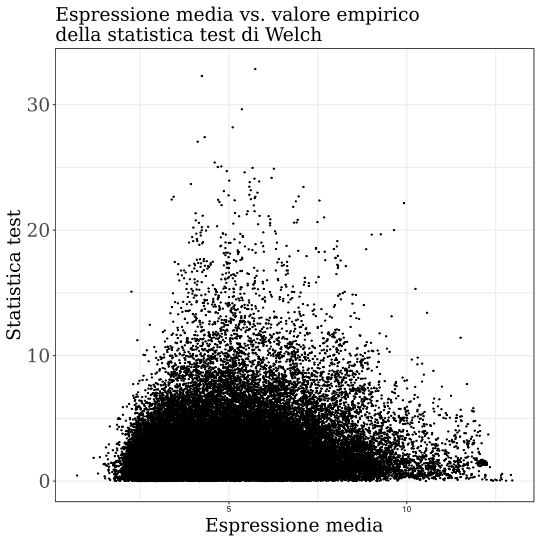


Figura 2.12: Espressione media vs. valore empirico della statistica test di Welch.

probe significativi risulta pari a 4,135. Nella Tabella 2.2 vengono riportati i primi dieci geni più significativi.

Statistica test di Welch: i primi 10 geni differenzialmente espressi

Gene	T_{Welch}	df_{num}	df_{den}	p-value
MPPED2	29.633	3	37.991	4.7727E-10
F2	27.046	3	36.797	2E-09
ANO1	24.603	3	39.515	3.8073E-09
NPAS3	24.707	3	38.622	4.3006E-09
LOX	24.962	3	37.396	4.8573E-09
DYSF	25.078	3	36.569	5.4616E-09
NXNL2	24.105	3	37.941	6.6856E-09
C12orf75	23.443	3	39.349	7.2107E-09
CTH	24.893	3	35.499	7.5239E-09
EPHB6	23.947	3	37.385	8.085E-09

Tabella 2.2: Elenco dei primi 10 geni differenzialmente espressi secondo la statistica test di Welch (nome del gene, valore empirico della statistica test (T_{Welch}), gradi di libertà al numeratore, df_{num}, al denominatore df_{den} e p-values senza correzione).

Test di Kruskal-Wallis

Il *testi di Krukal Wallis* [14] è un test non parametrico di analisi della varianza ad una via che viene applicato quando le assunzioni dell'ANOVA non vengono rispettate. In sostanza, esso confronta i ranghi delle osservazioni piuttosto che le osservazioni stesse. Quindi, esso costituirebbe uno strumento robusto attraverso cui ottenere dei risultati potenzialmente più affidabili rispetto a test parametrici le cui assunzioni possono non essere rispettate dai dati. La distribuzione della statistica test sotto H_0 è una χ^2_3 il cui quantile di livello $1-\alpha = 0.95$ è pari a 7.814728.

In Figura 2.21 e 2.22 i grafici risultano non alterati da alcun sistematicità e, quindi, soddisfacenti.

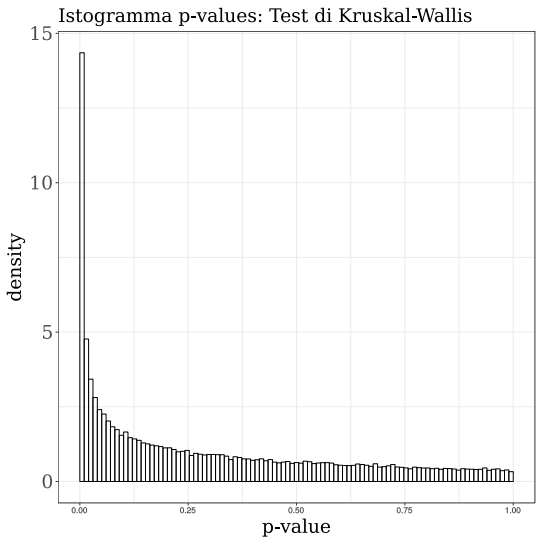


Figura 2.13: Istogramma p-values (non corretti): Test di Kruskal-Wallis.

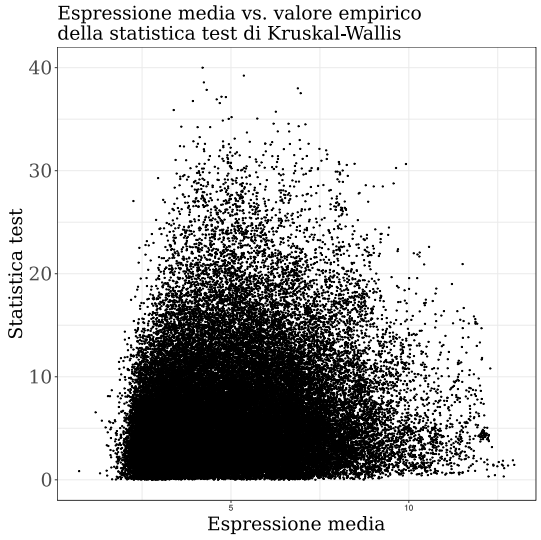


Figura 2.14: Espressione media vs. valore empirico della statistica test di Kruskal-Wallis.

Il numero di probes per cui è stata rifiutata l'ipotesi nulla è pari a 4,427 e in Tabella 2.3 vengono elencati i primi dieci geni più significativi.

Statistica test di Kruskal-Wallis: i primi 10 geni differenzialmente espressi

Gene	T_{KW}	**p-value**
PTH1R	39.996	1.0675E-08
MPPED2	39.227	1.5534E-08
MEIS1	37.992	2.8380E-08
NPAS3	37.142	4.2945E-08
RBPMS	36.760	5.1714E-08
CTH	35.725	8.5599E-08
AIF1L	35.199	1.1058E-07
PTHLH	34.865	1.3007E-07
C7orf49	34.5555	1.5126E-07
PLOD1	34.3152	1.7E-06

Tabella 2.3: Elenco dei primi 10 geni differenzialmente espressi secondo la statistica test di Kruskal-Wallis (nome del gene, valore empirico della statistica test (T_{KW}) e p-values senza correzione).

Test Empirical Bayes

Il *test Empirical Bayes* consiste nell'applicazione di un approccio bayesiano empirico per la stima a posteriori dei contrasti (differenze fra medie) il quale applica infine una statistica test T (frequentista) per valutare la loro significatività. Tale approccio attribuisce come parametri delle distribuzioni a priori dei contrasti e della varianza le stime empiriche ottenute attraverso il metodo dei momenti (per questo viene definito *bayesiano empirico*). Una volta stimati tali parametri, rispetto al modello assunto sui dati di espressione e alle distribuzioni a priori sui parametri del modello, vengono calcolate le stime a posteriori dei contrasti (differenze fra medie) e della loro relativa varianza (che figurerà al denominatore della statistica e sarà la caratteristica che definisce questo test un *test moderato*). La distribuzione della statistica test sotto l'ipotesi nulla è una F di Fisher-Snedecor con gradi di libertà $(3, 76.254)$, il cui quantile di riferimento per la definizione della regione di rifiuto è ≈ 2.7245.

Anche nel caso del test bayesiano empirico entrambi i grafici riportati in Figura 2.23 e 2.24 non presentano anomalie e il numero di probes significativi è pari a 5,303.

In Tabella 2.4 vengono riportati i primi dieci geni più significativi.

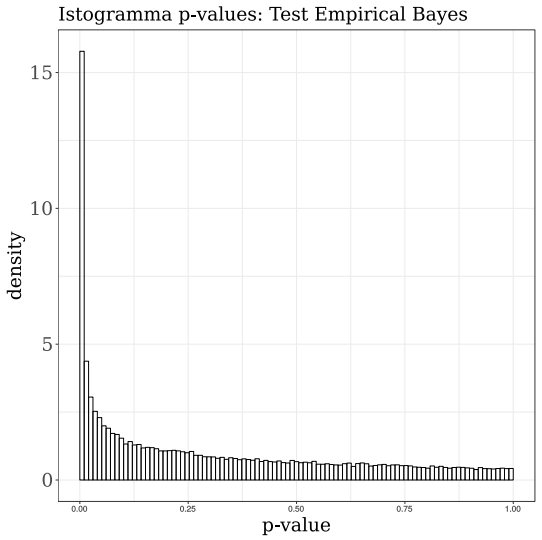


Figura 2.15: Istogramma p-values (non corretti): Test Empirical Bayes.

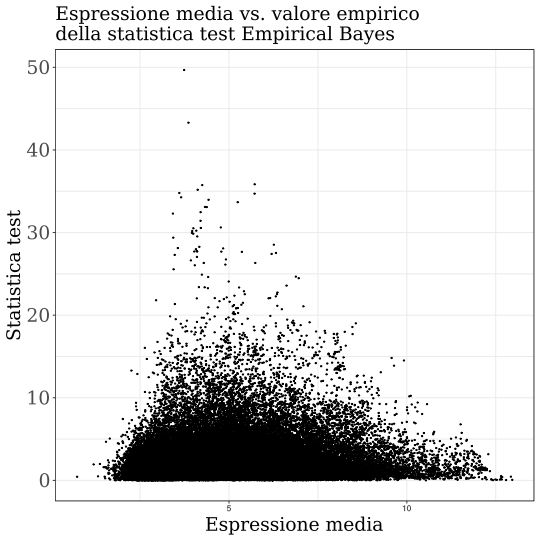


Figura 2.16: Espressione media vs. valore empirico della statistica test Empirical Bayes.

Statistica test di Empirical Bayes: i primi 10 geni differenzialmente espressi

Gene	T_{EB}	**p-value**
LBP	49.675	6.6571E-18
FXYD2	35.844	1.4735E-14
F2	35.175	2.2306E-14
HAVCR1	34.788	2.8430E-14
PTP4A1	33.677	5.7441E-14
PTH1R	32.467	1.2547E-13
PTHLH	30.617	4.2741E-13
HNF1B	29.528	8.9646E-13
CTH	28.530	1.7894E-12
IGFBP1	28.081	2.4522E-12

Tabella 2.4: Elenco dei primi 10 geni differenzialmente espressi secondo la statistica test Empirical Bayes (nome del gene, valore empirico della statistica test (T_{EB}) e p-values senza correzione).

Test SAM

Il *test SAM (Significance Analysis of Microarrays)* è un test parametrico moderato che *incrementa* la varianza al denominatore attraverso un criterio interno di analisi delle deviazioni standard dei singoli probe calcolate su tutti gli array. La significatività della statistica test risultante viene valutata mediante un approccio permutazionale (1,000 permutazioni) in cui si simula la distribuzione nulla (sotto H_0) e successivamente si stima il p-value come la proporzione dei valori più estremi (contro H_0) rispetto al valore campionario osservato. In Figura 2.25 l'istogramma dei pvalues non presenta artefatti di alcun tipo; invece, in Figura 2.26 il grafico a dispersione presenta una nuvola moderatamente dispersa che tuttavia non evidenzia alcun andamento sistematico sostantivo. Il numero di probe differenzialmente espressi è pari a 5,226.

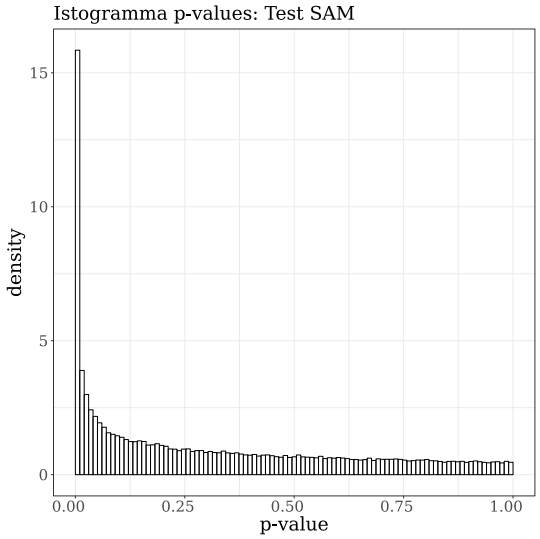


Figura 2.17: Istogramma p-values (non corretti): Test SAM.

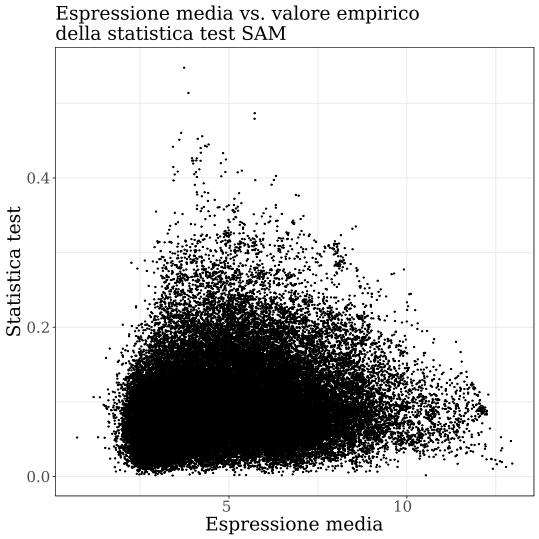


Figura 2.18: Espressione media vs. valore empirico della statistica test SAM.

In Tabella 2.5 è possibile consultare i primi dieci geni più significativi.

Statistica test SAM: i primi 10 geni differenzialmente espressi

Gene	T^{obs}_{SAM}	T^{*}_{SAM}	**p-value**
ESR1	0.2985	0.1623	1.5891E-05
OLIG3	0.3908	0.1589	1.5891E-05
ANXA4	0.3350	0.1528	1.5891E-05
LGALS4	0.3216	0.1513	1.5891E-05
RIMKLB	0.3635	0.1394	1.5891E-05
TM4SF5	0.3449	0.1377	1.5891E-05
SPINK1	0.3589	0.1366	1.5891E-05
CDX2	0.3017	0.1267	1.5891E-05
BBOX1	0.3130	0.1260	1.5891E-05
RASD1	0.3054	0.1232	1.5891E-05

Tabella 2.5: Elenco dei primi 10 geni differenzialmente espressi secondo la statistica test SAM (nome del gene, valore empirico della statistica test (T^{obs}_{SAM}), valore atteso della statistica test sotto H_0 (T^{*}_{SAM}) e p-values senza correzione ottenuti mediante approccio permutazionale).

Mettendo a confronto le tabelle sopra riportate, emergono in comune (almeno in tre test) quattro geni: PTHLH, PTH1R, CTH e F2.

PTHLH è il gene che codifica per la proteina paratiroidea che regola lo sviluppo osseo endocondrale e viene anche espresso in diversi tipi di tumore [6]. PTH1R è l'ormone recettore dell'ormone paratiroideo PTHLH [5]. Si osserva che l'espressione di entrambi i geni risulta potenzialmente diversa fra le istologie in esame.

Anche l'espressione del gene CTH è alterata e ciò costituisce un possibile segnale di diversa regolazione fra gli istotipi rispetto al pathway biochimico in cui esso è presente. Detto gene

codifica per un enzima citoplasmatico costituente una parte della catena di trans-sulfurazione della cisteina [2].

Altri geni che risultano espressi in modo diverso sono: il fattore di coagulazione II (F2) il quale è incaricato nella formazione della trombina durante il processo di coagulazione [3]; RASD1, il quale prodotto proteico è necessario per la definizione della morfologia della cellula, della sua crescita e delle interazioni fra essa e la matrice extracellulare [7]; HNF1B, gene coinvolto nello sviluppo del nefrone (unità funzionale del rene [28]), le cui mutazioni possono condurre a cisti renali, diabete mellito non insulino-dipendente (la sua espressione è stata trovata alterata in alcuni tipi di cancro) [4].

Quello che si evince con i primi risultati è che le differenze fra le istologie potrebbero interessare molto probabilmente l'alterazione dell'organizzazione e della struttura del tessuto al quale ciascun istotipo somiglia (clear cell al rene, sieroso all'epitelio delle tube di falloppio, mucinoso all'epitelio del tratto gastrointestinale o endocervicale, endometrioide al corpo uterino). Non si esclude che determinati pathways da un lato possano essere espressi in modo alterato rispetto a quanto accade in un tessuto normale, dall'altro possano risultare regolati in modo simile fra le istologie tumorali.

In altri termini, mettendo a confronto tumori che sicuramente avranno espressi alcuni geni in modo anomalo rispetto all'espressione in un tessuto normale, quello che ci si aspetta è di trovare una differenza quanto più legata alla diversa istologia che ad un altro tipo di variabilità, considerato che le neoplasie in esame sono tutte forme precoci del tumore ovarico.

Intersezione delle liste di probes differenzialmente espressi

Una volta corrette le liste di probes differenzialmente espressi con il metodo di Benjamini–Hochberg, è stata valutata l'intersezione di queste ultime, rappresentata in Figura 2.27. Il

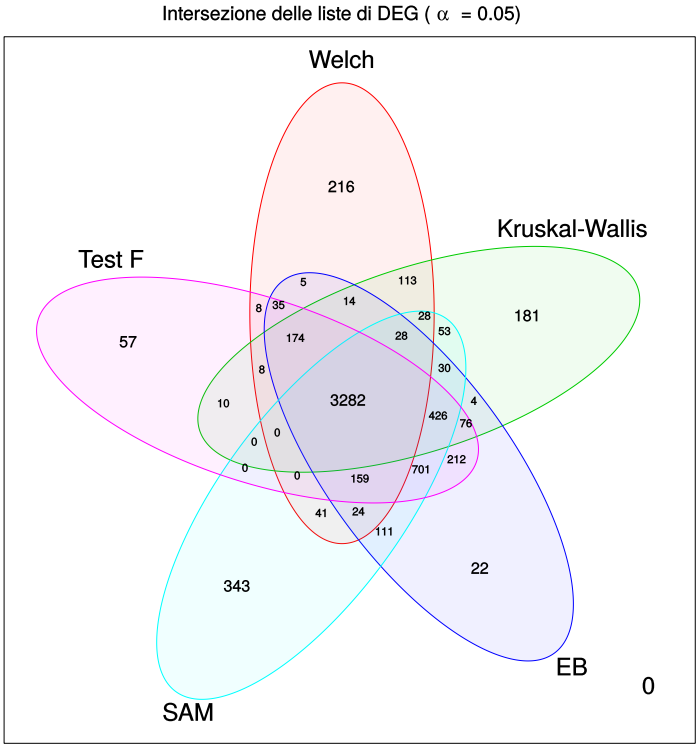


Figura 2.19: Intersezione delle liste di probes differenzialmente espressi ($\alpha = 0.05$).

numero di probes che sono stati definiti congiuntamente da tutte e cinque le liste come espressi in modo significativamente diverso fra le condizioni del tumore in analisi è pari a $3,282$ (in particolare 2,553 geni). Ciascun microarray contiene l'intero genoma umano (tutti i geni della specie *homo sapiens*), dove però ciascun gene viene suddiviso in più trascritti (probes) definendo così una corrispondenza uno-molti per gene-probe. Pertanto, la matrice di espressione in possesso per ogni paziente avrà un numero di celle pari al numero di probes, posizionati in modo ordinato affinché in fase di elaborazione dei dati si riconosca il trascritto e, quindi, il gene corrispondente. Allora, si è passati alla definizione di una nuova matrice di espressione di dimensioni ridotte e avente un gene per riga piuttosto che più probes relativi allo stesso gene. Per la scelta del probe rappresentativo di ciascun gene è stato condotto il seguente ragionamento: se il gene è risultato differenzialmente espresso allora è stata presa in considerazione l'intensità del probe più significativo (se più di un probe è risultato significativo, o semplicemente l'intensità d'espressione dell'unico probe significativo); invece, per i geni classificati come non significativi è stato scelto un valore di espressione casuale fra i valori dei probes corrispondenti. La matrice risultante ha dimensioni $19,681 \times 76$, circa il 70% di righe in meno rispetto alla matrice iniziale.

Capitolo 3

Analisi di pathways

Per *pathway* si intende un determinato insieme di entità biologiche aventi una funzione ben precisa a livello molecolare. In altri termini, esso consiste in un insieme di geni (*gene set*) in cui vengono annotate relazioni di ogni tipo intercorrenti fra due o più entità (proteine, complessi proteici, ormoni, etc...).

La loro principale classificazione consta di due grandi famiglie di pathways: i *pathways di segnale* e i *pathways metabolici*. I primi costituiscono quei pathways in cui vengono annotati i processi biologici che definiscono la trasduzione del segnale a livello cellulare. I secondi, invece, annotano le reazioni chimiche fra componenti finalizzate ai processi di anabolismo o catabolismo della cellula. Quindi, di rilevante importanza risulterebbe la possibilità di rispondere in modo quanto più affidabile alla domanda: *La differenziale espressione dello specifico pathway fra i fenotipi in esame è statisticamente significativa oppure è dovuta al caso?* Con ciò verrebbe condotto un confronto fra fenotipi sulla base di una misura che sintetizzi l'informazione a livello di pathway per ciascuna condizione sperimentale in analisi.

In letteratura, i metodi classici di *gene set analysis* vengono classificati in due principali categorie: i *metodi competitivi* e i *metodi indipendenti*. Queste due tipologie differiscono principalmente per la formulazione dell'ipotesi nulla (H_0), la quale ha ripercussioni sulla potenza del test, rendendo i secondi più potenti dei primi. Tuttavia, entrambi i metodi hanno in comune la necessità di essere a conoscenza della caratterizzazione fenotipica di ciascun campione (*supervisionati*). Quest'ultima non è un'informazione sempre nota a chi conduce le analisi e, inoltre, si potrebbe anche pensare che le differenze fra le repliche biologiche non seguano un pattern fenotipico già noto alla scienza ma possano definirsi secondo nuove classificazioni costituenti una evidenza empirica verso firme molecolari del tumore in esame non note finora.

Sulla base di queste considerazioni, negli ultimi anni sono state proposte metodologie inno-

vative che hanno permesso il calcolo di un punteggio (*score*) a livello di pathway ma soprattutto per ogni singola replica biologica.

Relativamente alle analisi sul tumore ovarico, sono stati scelti due metodi proposti di recente e che utilizzano due approcci diversi:

- *Gene Set Variation Analysis (GSVA)*: approccio non parametrico che calcola un punteggio per paziente e per singolo pathway sulla base della stima di una densità paziente-specifica;
- *Personalized Pathway Alteration Analysis (PerPAS)*: metodo che assegna un punteggio per paziente per singolo pathway tenendo in considerazione la topologia del determinato insieme di geni.

Prima di proseguire con l'esposizione più dettagliata di entrambi i metodi, è necessario definire i due dati di input che vengono richiesti:

- una matrice $X = \{x_{ij}\}_{p \times n}$ di valori di espressione normalizzati e avente dimensione $p \times n$ (p geni, n repliche biologiche). In genere, $p >> n$ ed x_{ij} definisce il valore di espressione dell' $i - esimo$ gene nel $j - esimo$ campione);
- una collezione di insiemi di geni (pathways) $\Gamma = \{\gamma_1, \gamma_2, \ldots, \gamma_m\}$ dove γ_k (pathway $k - esimo$) è costituito dal sottoinsieme di indici di riga della matrice X tale che $\gamma_k \subset \{1, \ldots, p\}$; la cardinalità del $k - esimo$ pathway verrà indicata con $|\gamma_k|$.

3.1 Gene Set Variation Analysis (GSVA)

L'algoritmo di Gene Set Variation Analysis [13] si sviluppa in quattro passi:

Passo 1. Stima kernel non parametrica della funzione di ripartizione di ciascun profilo di espressione relativo ad ogni gene $x_{i\cdot} = \{x_{i1}, \ldots, x_{in}\}$ per $i = 1, \ldots, p$ affinché i diversi profili di espressione vengano ricondotti ad una scala comune. Nel caso dei microarray, il kernel è di tipo Gaussiano,

$$\hat{F}_{h_i}(x_{ij}) = \frac{1}{n} \sum_{k=1}^{n} \int_{-\infty}^{\frac{x_{ij} - x_{ik}}{h_i}} \frac{1}{\sqrt{2\pi}} e^{-\frac{t^2}{2}} dt \tag{3.1}$$

dove h_i definisce la larghezza di banda gene-specifica impostata a $s_i/4$, con s_i deviazione standard campionaria dell'$i - esimo$ gene. Tale banda funge da parametro di regolazione che permette di modificare la risoluzione della stima kernel.

Passo 2. Si definisce $\hat{F}_{h_i}(x_{ij}) = z_{ij}$ per semplicità di notazione e, con l'intento di ridurre l'influenza di potenziali valori anomali, gli z_{ij} vengono convertiti in ranghi $z_{(i)j}$ per ciascun campione

j. Successivamente, detti ranghi vengono ulteriormente normalizzati

$$r_{ij} = |p/2 - z_{(i)j}| \tag{3.2}$$

rendendoli simmetrici attorno allo zero. Quest'ultimo calcolo viene applicato per dare maggior peso alle due code della distribuzione dei ranghi quando verrà calcolato il punteggio.

Passo 3. Viene calcolata la statistica di Kolmogorov-Smirnov (KS) con random walk confrontando l'andamento di due distribuzioni: la distribuzione dei ranghi dei geni costituenti il pathway in analisi e una passeggiata aleatoria (random walk) definita dai geni non appartenenti al pathway. Quindi, il valore della statistica per il campione j relativamente all'insieme di geni k e calcolata fino al rango ℓ è la seguente

$$\nu_{jk}(\ell) = \frac{\sum_{i=1}^{\ell} |r_{ij}|^{\tau} I(g_{(i)} \in \gamma_k)}{\sum_{i=1}^{p} |r_{ij}|^{\tau} I(g_{(i)} \in \gamma_k)} - \frac{\sum_{i=1}^{\ell} I(g_{(i)} \notin \gamma_k)}{p - |\gamma_k|} \tag{3.3}$$

dove τ è un parametro che definisce il peso della coda nella passeggiata aleatoria (impostato di default a 1), γ_k è il $k-esimo$ pathway, $I(g_{(i)} \in \gamma_k)$ è la funzione indicatrice se l'$i-esimo$ gene appartiene all'insieme γ_k, $|\gamma_k|$ è il numero di geni nel $k-esimo$ pathway, p è il numero totale di geni.

Passo 4. Vengono proposti due procedure differenti per calcolare da ν_{jk} un punteggio finale del campione j per il pathway k (*Enrichment Score (ES)*, chiamato anche *GSVA score*):

- il **metodo della massima deviazione**, il quale consiste nel calcolare la più grande deviazione da zero di ν_{jk}. Quindi,

$$ES_{jk}^{max} = \nu_{jk}[\arg \max_{\ell=1,\ldots,p} |\nu_{jk}(\ell)|] \tag{3.4}$$

  per ciascun pathway, tale procedura conduce a una distribuzione nulla bimodale dello ES. Questo comportamento è intrinseco nella statistica di KS con random walk in quanto sotto l'ipotesi nulla si presentano delle deviazioni massime non nulle e, quindi, la distribuzione nulla non è centrata in zero. Inoltre, la significatività può essere valutata indipendentemente utilizzando la parte positiva o la parte negativa della distribuzione nulla.

- il **metodo della normalizzazione dello ES** consiste, invece, nel calcolare la differenza fra la più grande deviazione positiva e la più grande deviazione negativa,

ovvero

$$ES_{jk}^{diff} = |ES_{jk}^{+}| - |ES_{jk}^{-}| = \max_{\ell=1,\dots,p}(0, \nu_{jk}(\ell)) - \min_{\ell=1,\dots,p}(0, \nu_{jk}(\ell)) \tag{3.5}$$

dove ES_{jk}^{+} e ES_{jk}^{-} rappresentano le deviazioni più grandi da zero, rispettivamente positiva e negativa e relative al campione j per il pathway k.

L'approccio di normalizzazione dello ES enfatizza quei geni che in un pathway sono espressi nella medesima direzione (sovraespressi o sottoespressi). Invece, rispetto ai pathways che hanno geni sia sovraespressi che sottoespressi (in riferimento ad una condizione tipo, in genere un tessuto sano) la normalizzazione cancellerebbe tale comportamento e il valore dello ES risulterebbe basso. Sotto H_0 la distribuzione dello score normalizzato è unimodale.

Una volta ottenuta la matrice di punteggi $S = \{s_{jk}\}_{n\times m}$ si possono rappresentare dei grafici per valutare le distribuzioni dei punteggi per singola replica biologica. Inoltre, è possibile valutare la significatività di ciascuno score implementando un algoritmo per simulare la distribuzione nulla attraverso un bootstrap.

Per condurre il calcolo dello score e la stima bootstrap del p-value paziente-specifico è stato utilizzato il pacchetto 'GSVA' disponibile su *Bioconductor* [12].

3.2 Personalized Pathway Alteration Analysis (PerPAS)

Il calcolo del punteggio PerPAS [17] avviene in tre passi:

Passo 1. Pre-elaborazione del dato (eseguita soltanto una volta): la quale consiste nella standardizzazione del valore di espressione del gene. In altri termini, l'intensità di espressione viene normalizzata rispetto alla media e alla deviazione standard di tutte le repliche biologiche. Quindi, verrà definita una matrice $Z = \{z_{ij}\}_{p\times n}$ in cui $z_{ij} = \frac{x_{ij}-\mu_{i\cdot}}{\sigma_{i\cdot}}$, con $\mu_{i\cdot}$ media d'espressione del gene $i-esimo$, $\sigma_{i\cdot}$ deviazione standard del gene $i-esimo$.

Passo 2. Quantificazione del contributo di un gene nel pathway: tale contributo viene misurato in base al ruolo di **hub** e di **bottleneck** che il gene esercita nel pathway considerato.

La **hubeness** di un gene può essere calcolata misurando il numero di geni sui quali esso ha un'influenza diretta (definiti *geni downstream diretti*). Per quantificare il ruolo di *hub*

del gene i nel pathway k relativamente al campione j si calcola

$$H_{ijk} = \frac{\sum_{l \in \omega_{ik}} z_{lj}}{|\omega_{ik}|} \tag{3.6}$$

dove ω_{ik} è l'insieme di geni downstream diretti del gene i nel pathway k, $|\omega_{ik}|$ è il numero totale di geni downstream diretti del gene i nel pathway k, z_{lj} è il valore d'espressione standardizzato del gene l nel campione j.

La **bottleneck** misura quanto un gene influisce nei flussi di segnale che caratterizzano il pathway in analisi. Per quantificare il ruolo di **bottleneck** viene stimata la percentuale di flussi di segnale in cui si ritrova il particolare gene. Pertanto, si propone di utilizzare il *cammino più corto (shortest path)* per definire un flusso di segnale fra due geni. Di conseguenza, l'assunzione intrinseca è che il segnale che viene trasmesso da un gene verso un altro avviene sempre attraverso il cammino più corto. Da qui, segue la formula per quantificare il ruolo di *bottleneck*

$$Q_{ik} = \frac{n_{ik}}{N_k} \quad Q_{ik} \in (0, 1) \tag{3.7}$$

dove N_k è il numero totale di flussi di segnale (shortest paths) presenti nel pathway k, n_{ik} è il numero di flussi di segnale del pathway k in cui è presente il gene i.

Per ciascun gene del pathway k vengono calcolate H_{ijk} e Q_{ik}. Quindi, data la replica biologica j, il contributo del gene i nel pathway k è definito dalla moltiplicazione di H_{ijk} e Q_{ik},

$$C_{ijk} = Q_{ik} \cdot H_{ijk} \tag{3.8}$$

Passo 3. Calcolo del punteggio PerPAS: dato il campione j e il pathway k, il punteggio s_{jk} viene calcolato come segue

$$s_{jk} = \sum_{i \in \gamma_k} C_{ijk} = \sum_{i \in \gamma_k} (Q_{ik} \cdot H_{ijk}) \tag{3.9}$$

dove γ_k rappresenta il sottoinsieme di geni costituenti il pathway k.

Ricavata la matrice dei punteggi $S = \{s_{jk}\}_{n \times m}$ è possibile anche per il PerPAS calcolare la significatività di ciascun punteggio paziente-specifico. In questo caso la distribuzione nulla può essere stimata in due modi diversi: scambiando casualmente i valori di espressione fra i geni che costituiscono il pathway; oppure, variando casualmente le relazioni fra i geni all'interno del

pathway, alterandone quindi la topologia. Per il calcolo dello score PerPAS è stato utilizzato il pacchetto 'PerPAS' disponibile sul sito *Systems Biology Laboratory* dell'Università di Helsinki [16].

In sintesi, volendo confrontare i due metodi, essi differiscono principalmente per il tipo di approccio che hanno con i dati di espressione: GSVA risulta di stampo più classico basandosi sulla densità dei ranghi delle misure di espressione; invece, il PerPAS è direttamente legato alla topologia del pathway e quindi influenzato dal livello di precisione con cui viene annotato e aggiornato un pathway, basandosi comunque sui livelli di espressione e non su una loro trasformazione in ranghi.

3.3 I risultati

L'applicazione di entrambi i metodi di Gene Set Analysis è stata condotta su 285 pathways (annotati nella libreria KEGG[1] e ricavati attraverso il pacchetto di R "graphite" [20]) dell'organismo *Homo Sapiens*. Una volta calcolate le due matrici di punteggi di dimensioni 285×76 ciascuna (ES^{diff} del GSVA e PerPAS), è stato condotto un test non parametrico di Kruskal-Wallis per verificare la differenziale espressione di ciascun pathway fra le quattro istologie tumorali (con livello di significatività fissato ad $\alpha = 0.05$).

Il bootstrap per il calcolo della stima della significatività campione-specifica è stato condotto soltanto su GSVA attraverso 10,010 simulazioni suddivise su 7 cores (1,430 simulazioni per singolo core), riducendo in questo modo la variabilità della stima del p-value. Il bootstrap relativo al PerPAS non è stato condotto per motivi di tempo in quanto la simulazione della distribuzione nulla risultava sempre più onerosa al crescere del numero di nodi e di relazioni all'interno del pathway. Per tale motivo è stato scelto di incrociare i risultati del PerPAS con quelli ottenuti con il GSVA.

Innanzi tutto, il numero di pathways significativi per il GSVA secondo il test di Kruskal-Wallis è risultato pari a 78. Sucessivamente è stata posta un'ulteriore restrizione sulla significatività campione-specifica secondo la quale mantenere soltanto quei pathways in cui almeno il 60% di repliche biologiche per ogni istologia sia risultato significativo al 5%. In Appendice B.1 e B.2 è possibile consultare l'istogramma dei p-values non corretti e i grafici a dispersione della statistica test vs. punteggio medio per lo score GSVA. Di seguito, in Tabella 3.1 vengono riportati i 25 pathways (dei 78) che rispettano la seconda restrizione.

[1] http://www.kegg.jp/

GSVA - ES^{diff} : Top 25 pathways differenzialmente espressi nelle quattro istologie

Pathway	KEGG id	T_{KW}	p-value	% sign.
Small cell lung cancer	hsa:05222	27.0505	5.7456E-06	71.1
Phosphonate and phosphinate metabolism	hsa:00440	18.9605	0.0003	72.4
Carbohydrate digestion and absorption	hsa:04973	17.4170	0.0006	78.9
Osteoclast differentiation	hsa:04380	16.3037	0.001	72.4
Phosphatidylinositol signaling system	hsa:04070	15.8311	0.0012	69.7
IL-17 signaling pathway	hsa:04657	15.7293	0.0013	73.7
Type II diabetes mellitus	hsa:04930	15.3368	0.0016	69.7
MicroRNAs in cancer	hsa:05206	13.8732	0.0031	67.1
Retrograde endocannabinoid signaling	hsa:04723	12.3170	0.0064	73.7
Caffeine metabolism	hsa:00232	11.7120	0.0084	69.7
N-Glycan biosynthesis	hsa:00510	11.3427	0.01	71.1
Autophagy - animal	hsa:04140	11.2674	0.0104	71.1
RIG-I-like receptor signaling pathway	hsa:04622	11.1521	0.0109	67.1
Breast cancer	hsa:05224	10.9404	0.0121	68.4
Adrenergic signaling in cardiomyocytes	hsa:04261	10.7237	0.0133	72.4
Prostate cancer	hsa:05215	10.0673	0.018	72.4
RNA degradation	hsa:03018	9.6014	0.0223	81.6
Lipoic acid metabolism	hsa:00785	9.4275	0.0241	81.6
mRNA surveillance pathway	hsa:03015	9.2803	0.0258	69.7
cGMP-PKG signaling pathway	hsa:04022	9.1808	0.027	75
Sphingolipid metabolism	hsa:00600	8.9580	0.0299	71.1
Biosynthesis of unsaturated fatty acids	hsa:01040	8.5285	0.0363	71.1
Glycosaminoglycan biosynthesis	hsa:00534	8.4947	0.0368	68.4
Phospholipase D signaling pathway	hsa:04072	8.1371	0.0433	73.7
Tight junction	hsa:04530	7.8648	0.0489	77.6

Tabella 3.1: GSVA - ES^{diff}: Top 25 pathways differenzialmente espressi nelle quattro istologie (nome del pathway, identificativo KEGG, valore empirico della statistica test Kruskal-Wallis, p-value non corretto, percentuale di campioni significativi al 5%).

L'applicazione del test di Kruskal-Wallis ai punteggi del PerPAS ha portato a 79 pathways significativi al 5%, dei quali soltanto 6 (annotati in Tabella 3.2 e scritti in corsivo in Tabella 3.1) sono in comune con la lista dei 25 pathways sopra riportati.

PerPAS : 6 pathways differenzialmente espressi nelle quattro istologie

Pathway	KEGG id	T_{KW}	p-value
RIG-I-like receptor signaling pathway	hsa:04622	14.7092	0.0021
Carbohydrate digestion and absorption	hsa:04973	12.7306	0.0053
Retrograde endocannabinoid signaling	hsa:04723	12.0979	0.0071
RNA degradation	hsa:03018	10.4946	0.0148
Adrenergic signaling in cardiomyocytes	hsa:04261	9.8165	0.0202
Type II diabetes mellitus	hsa:04930	9.1080	0.0279

Tabella 3.2: PerPAS: 6 pathways differenzialmente espressi nelle quattro istologie (nome del pathway, identificativo KEGG, valore empirico della statistica test Kruskal-Wallis, p-value non corretto).

Tale risultato comune va a corroborare l'affermazione riguardo la loro differenziale espressione. In Appendice B.3 e B.4 vengono riportati i grafici relativi all'istogramma dei p-values e al grafico a dispersione fra la statistica test e la media del punteggio PerPAS. In Figura 3.1 e

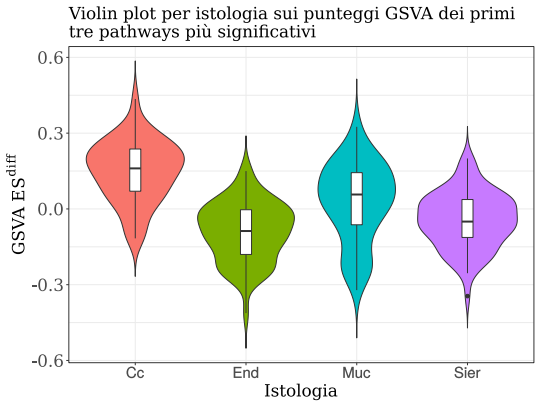


Figura 3.1: Violin plot per istologia sui punteggi del GSVA relativi ai primi tre pathways più significativi secondo il medesimo metodo.

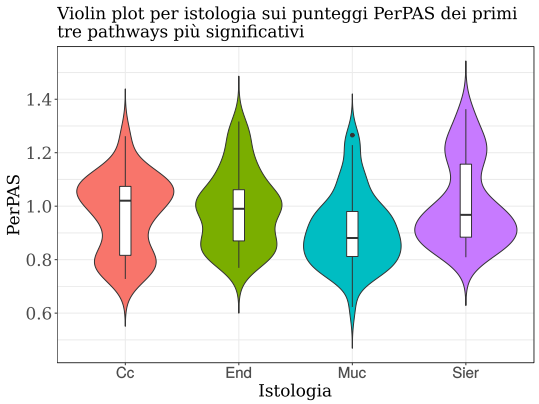


Figura 3.2: Violin plot per istologia sui punteggi del PerPAS relativi ai primi tre pathways più significativi secondo il medesimo metodo.

3.2 è possibile osservare i violin plots relativi ai punteggi GSVA e PerPAS per ciascuna istolgia e considerando soltanto i primi tre pathways più significativi per metodo.

Si noti come più con il GSVA che con il PerPAS si percepisce una differenza in termini di posizione fra le distribuzioni dei punteggi.

Affinché si possa evincere in modo più semplice e diretto la differenza di espressione fra i pathway, vengono riportati due grafici heatmap relativi alle due tipologie di score, ES^{diff} e PerPAS. La scala del colore è proporzionale al rango medio della determinata istologia (quindi sul vettore ordinato di punteggi di ciascun pathway è stato calcolato il rango medio rispetto ad ognuna delle quattro istologie).

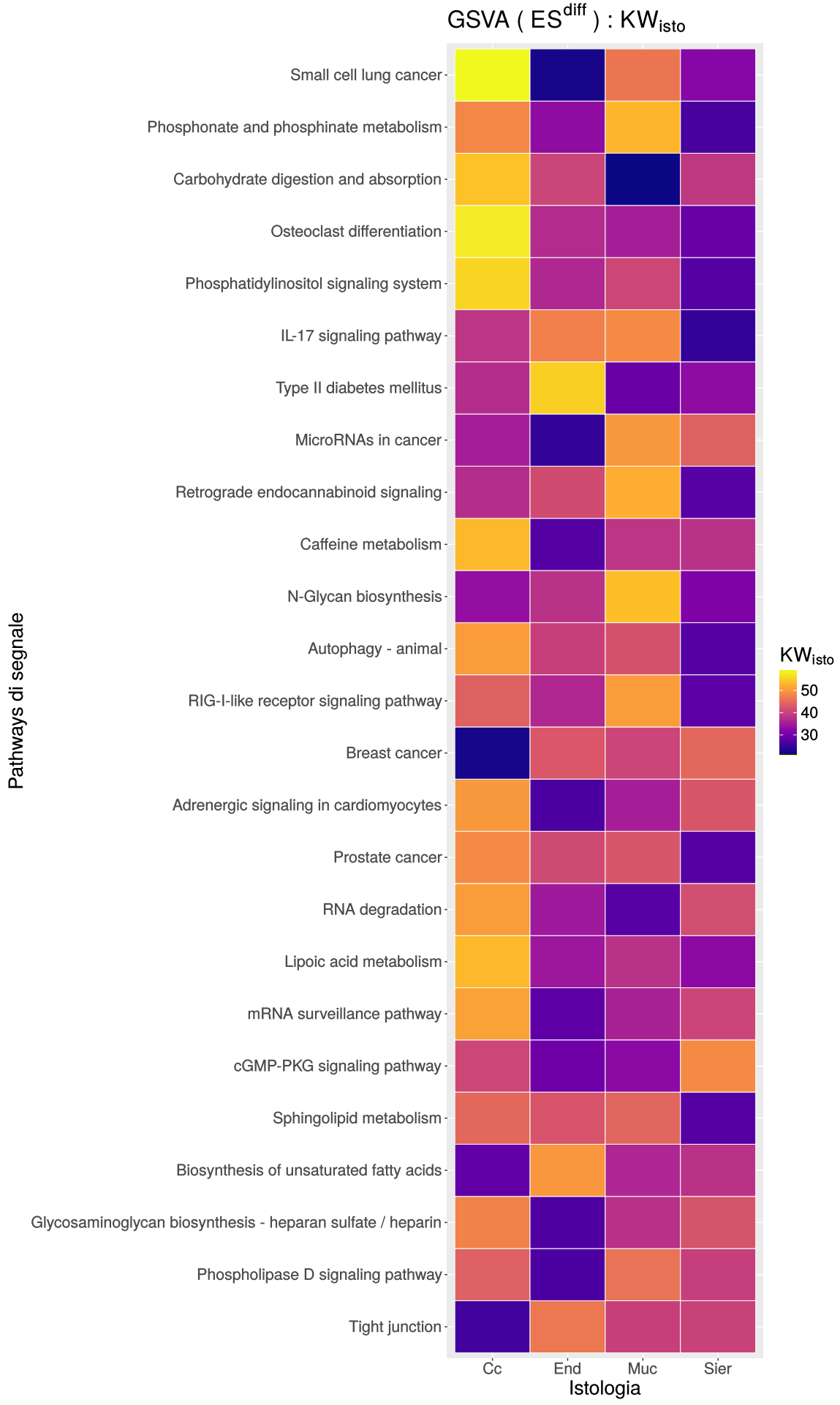


Figura 3.3: Heatmap Top 25 pathways GSVA (ES^{diff}).

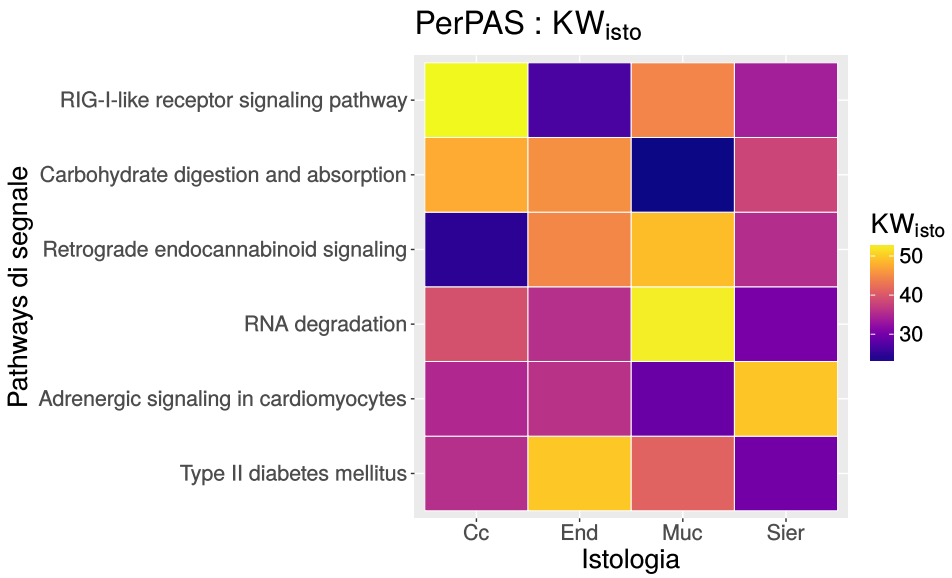


Figura 3.4: Heatmap 5 pathways PerPAS.

Da entrambe le heatmap in Figura 3.3 e 3.4 si osservano delle differenze sempre più marcate fra i colori dal basso verso l'alto. Il colore, proporzionale al rango medio, mette in evidenza la differenza fra i ranghi medi e quindi è direttamente legato alla significatività risultante del relativo pathway (questo comportamento è più chiaro in Figura 3.3 in cui il pathway *Tight junction* ha un p-value vicino a 0.05)

Nelle analisi successive verranno prese in considerazione le due matrici di punteggi relative al GSVA (ES^{diff}) e al PerPAS e la matrice di dati di espressione normalizzati relativa ai geni differenzialmente espressi (2,553 geni) nelle quattro istologie.

Capitolo 4

Le reti di pazienti: caratteristiche e descrizione

L'insieme delle 76 repliche biologiche può essere visto come una rete in cui i *nodi* sono rappresentati dalle singole repliche (pazienti) e la presenza di relazioni fra queste ultime è definita attraverso dei legami chiamati *archi*.

In questo capitolo verrà condotta una prima definizione ed analisi formale sulle caratteristiche delle reti in esame, la quale proseguirà nel Capitolo 5 con la definizione di due approcci metodologicamente differenti ma aventi un fine comune quale l'*identificazione di comunità* all'interno della rete di pazienti.

Per quanto riguarda il calcolo delle statistiche descrittive di rete e la loro rappresentazione grafica è stato usato il pacchetto di R *"igraph"* [11].

Una rete binaria non direzionata

Volendo trattare l'insieme dei 76 pazienti come una rete, il **nodo** sarà rappresentato dal singolo **paziente** e la presenza di un **arco** andrà a specificare se fra il paziente l e il paziente j ($l \neq j$) è presente una relazione di qualsiasi entità o meno. Dunque, la rete sarà **non direzionata** in quanto non risulterebbe importante ai fini interpretativi la *direzione* con cui i pazienti entrano in relazione; inoltre, la rete sarà trattata per semplicità come **binaria**, ovvero sarà d'interesse soltanto la presenza o meno di una relazione fra il paziente l e il paziente j (si potrebbe comunque definire una rete non direzionata *pesata* ma questa assunzione non è stata trattata nelle seguenti analisi, soffermandosi, dunque, su un'ipotesi semplificatrice di rete binaria non direzionata).

L'analisi delle interazioni (links, connessioni) fra pazienti verrà condotta sulla base di due differenti misure: la *matrice dei punteggi* (GSVA, 25 pathways, PerPAS, 79 pathways), la *matrice di espressione normalizzata dei geni differenzialmente espressi* (2,553 geni).

Nel primo caso, sia per GSVA che per PerPAS, si definisce **profilo di punteggio** del paziente j relativo all'insieme di pathways λ il vettore di punteggi (GSVA o PerPAS) dell'esperimento j, costituenti una misura quantitativa personalizzata dell'espressione dei pathways considerati. Quindi, definita la matrice dei punteggi $S = \{s_{jk}\}_{n \times m}$, il profilo di score del paziente j sarà

$$p_j = s_{j\cdot} \tag{4.1}$$

ovvero la riga $j - esima$ della matrice S.

Nel secondo caso, si definisce **profilo di espressione** del paziente j relativo all'insieme di geni γ_{DEG} il vettore di intensità normalizzate della replica biologica j e costituenti il gruppo di geni che sono differenzialmente espressi nelle quattro condizioni istologiche. Quindi, data la matrice di espressione normalizzata $X = \{x_{ij}\}_{p \times n}$ e l'insieme di *DEG* ($\gamma_{DEG} \subset \{1, \ldots, p\}$) il profilo di espressione del paziente j sarà

$$p_{j,\gamma_{DEG}} = \{x_{ij} \mid i \in \gamma_{DEG}\} \tag{4.2}$$

Di rilevante importanza è la definizione di **legame fra due pazienti** (rappresentato da un *arco*). Pertanto, se due pazienti l e j ($l \neq j$) hanno dei profili di espressione o di punteggio molto simili, cioè $p_l \approx p_j$, allora è verosimile che l'istologia tumorale sarà la stessa. Al contrario, se i profili di espressione o di punteggio fra due pazienti stanno su livelli diversi è plausibile che i due nodi non avranno una relazione. Quindi, formalizzando quanto assunto finora, si definisce un *grafo binario* e *non direzionato* $G = \{N, A\}$, il quale è costituito da due elementi principali:

- N : *insieme di nodi* (pazienti) $N = \{1, \ldots, n\}$;
- A : *matrice di adiacenza* $A_{n \times n} = \{a_{ij}, \quad (i,j) : i = 1, 2, \ldots, n, j = 1, 2, \ldots, n\}$, simmetrica ($a_{ij} = a_{ji}$), dove $a_{ij} = 1$ se (i,j) hanno un legame, $a_{ij} = 0$ altrimenti e $a_{ii} = 1$ (self-loop) $\quad \forall i = 1, \ldots, n$.

Un esempio di rete binaria non direzionata è rappresentato in Figura 4.1 , in cui 20 pazienti (P1, P2,..., P20) raggruppati secondo le quattro istologie (nell'esempio: 6 sierosi, 5 endometriodi, 4 mucinosi, 5 clear cell) entrano in relazione secondo la matrice di adiacenza A_1.

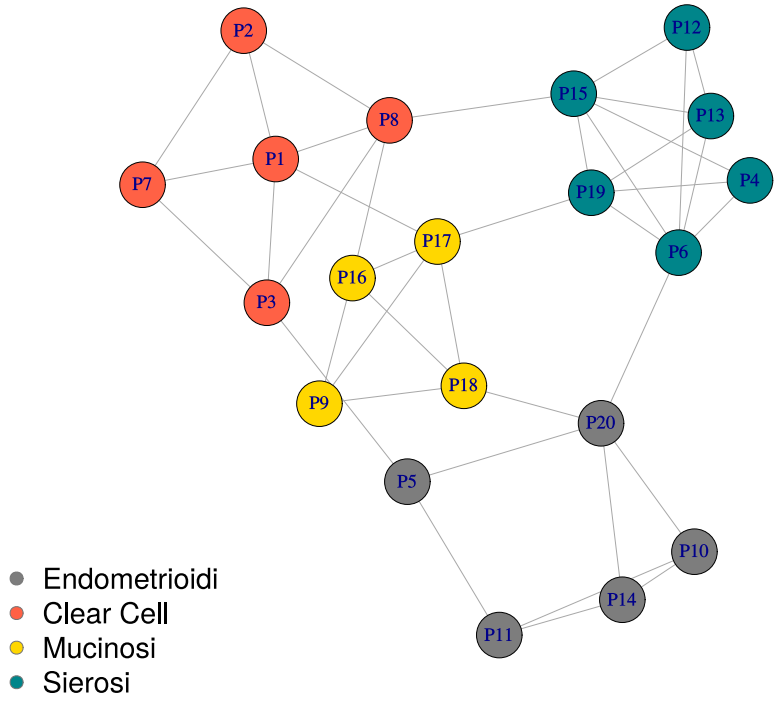


Figura 4.1: Rete binaria non direzionata di 20 pazienti (6 sierosi, 5 endometrioidi, 4 mucinosi, 5 clear cell)

Nell'esempio si osserva che i membri di ogni gruppo sono molto collegati fra loro (ad esempio il gruppo dei sierosi); invece, sono pochi i legami fra quelle coppie di nodi appartenenti ciascuno a gruppi diversi (ad esempio fra sierosi ed endometrioidi). Questo farebbe notare che quanto più simili sono le due tipologie di profili dei pazienti tanto più probabile è la presenza di una relazione fra loro e quindi la formazione di un gruppo molto coeso. Tuttavia, esiste anche la possibilità, seppur auspicabilmente sporadica, secondo la quale il paziente appartenente al gruppo k abbia un profilo simile ad un paziente che appartiene al gruppo r ($k \neq r$).

La *matrice di adiacenza*, risulta, quindi, avere un ruolo fondamentale nell'analisi delle relazioni fra pazienti. Un esempio di matrice di adiacenza è la matrice relativa al grafo in Figura 4.1.

$$A_1 = \begin{bmatrix} & P1 & P2 & P3 & P4 & \cdots & P20 \\ P1 & 1 & 1 & 1 & 0 & \cdots & 0 \\ P2 & 1 & 1 & 0 & 0 & \cdots & 0 \\ P3 & 1 & 0 & 1 & 0 & \cdots & 0 \\ P4 & 0 & 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ldots & \vdots \\ P20 & 0 & 0 & 0 & 0 & \cdots & 1 \end{bmatrix} \tag{4.3}$$

Nell'Equazione 4.3 viene riportata una parte della matrice di adiacenza A_1 relativa ai primi quattro pazienti e al paziente 20: si osserva, quindi, la presenza di 1 quando due pazienti hanno un legame (arco), 0 altrimenti. Le statistiche descrittive a livello di rete (transitività globale, densità, shortest path medio, diametro) e a livello di nodo (grado di un nodo, transitività locale, betweenness) sono direttamente calcolabili attraverso la matrice di adiacenza relativa alla determinata misura in analisi (sia essa una matrice di punteggi a livello di pathways, sia essa la matrice di espressione dei DEG) [19].

4.1 Misure di dissimilarità

Dalle precedenti considerazioni risulta fondamentale utilizzare una metrica che permetta di definire quantitativamente la diversità (o, per converso, la similarità) fra due profili di espressione o di punteggio relativi a due pazienti distinti. Esistono delle misure classiche che permettono di quantificare la dissimilarità fra due vettori di dimensioni specifiche e relativi a due entità distinte. Nelle successive analisi verranno applicate due misure di distanza: la classica **distanza euclidea** ed una nuova misura di distanza basata su una misura di similarità a livello di rango, la **Fraction Enrichment Sum**.

La distanza euclidea (d_E)

Una *misura di distanza* che viene comunemente utilizzata è la **distanza euclidea**. Dati due pazienti l e j ($l \neq j$) e i loro profili (di espressione o di punteggio) p_l e p_j, si definisce la

distanza euclidea $d_E(p_l, p_j)$ come

$$\begin{cases} d_E(p_l, p_j) = \sum_{k=1}^{m} (s_{lk} - s_{jk})^2 & \text{, per } \textbf{profili di punteggio} \\ d_E(p_l, p_j) = \sum_{i \in \gamma_{DEG}} (x_{il} - x_{ij})^2 & \text{, per } \textbf{profili di espressione} \end{cases} \tag{4.4}$$

In Figura 4.2 vengono rappresentate le heatmap relative relative alle tre matrici prese in considerazione.

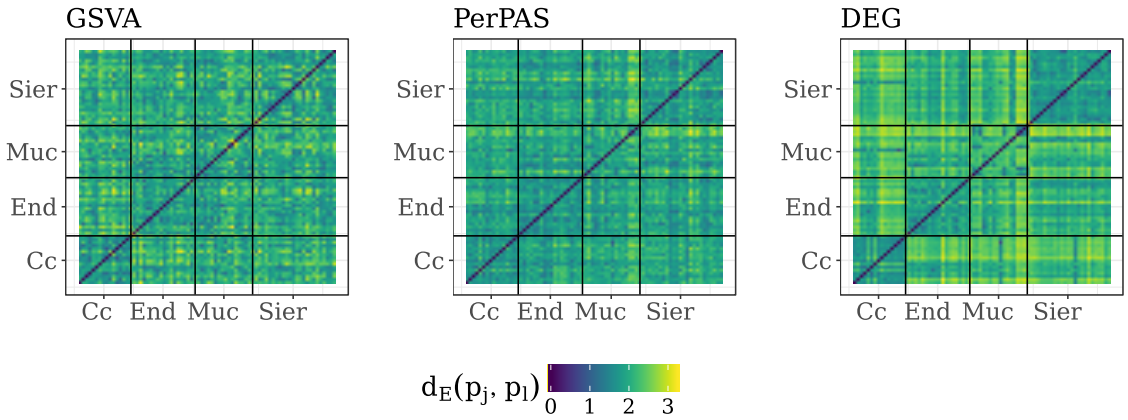


Figura 4.2: Heatmap distanza euclidea (GSVA, PerPAS e DEG).

Si osserva nel caso del gruppo di geni *DEG* come la misura di distanza euclidea riesca a far emergere la forte somiglianza interna ai gruppi istologici (colorazione più scura) e la non debole diversità fra repliche biologiche appartenenti ad istologie differenti (colorazione più chiara). Per GSVA e PerPAS , invece, le distanze osservate non evidenziano in modo chiaro i potenziali raggruppamenti sottostanti.

La distanza basata sulla Fraction Enrichment Sum (d_{FES})

La distanza basata sulla **Fraction Enrichment Sum** (d_{FES}) [21] rappresenta una particolare misura di dissimilarità complementare ad una misura di similarità (basata sui ranghi), la *FES*, la quale quantifica la sovrapposizione di vettori di misure ordinate in funzione della loro entità e relative a due pazienti j ed l ($j \neq l$).

Pertanto, prima di definire la misura di distanza, è necessario presentare la misura di similarità su cui essa si basa.

Si consideri la matrice Z di valori di espressione normalizzati dei DEG o dei valori dei punteggi (GSVA, PerPAS), standardizzati secondo media e deviazione standard del relativo gene/pathway rispetto alle 76 repliche biologiche.

Dati due pazienti j ed l ($j \neq l$) è possibile definire tre misure che quantificano il loro grado di somiglianza in termini di espressione (DEG) o di score relativo ai pathways: *Fraction Enrichment (FE)*, *Fraction Enrichment p-value (FEP)* e *Fraction Enrichment Sum (FES)*. Dati i due corrispondenti vettori di valori numerici di j e di l, si chiamino *elementi* le componenti degli stessi, ciascuno dei quali può essere l'*espressione di un gene* o il *punteggio di un pathway* a seconda se si stia considerando rispettivamente la matrice dei DEG o la matrice dei punteggi dei pathways.

La **Fraction Enrichment (FE)** consiste nel conteggio del numero di elementi in comune nei primi m elementi ordinati in senso crescente e costituenti le due liste (vettori) relative ai pazienti j ed l. Tale misura può essere calcolata per $m \in [1, n_k]$ dove n_k indica l'ordine massimo delle due liste di elementi (e rappresenta anche il numero totale di elementi). Dunque, considerati due pazienti j e l e definite le rispettive liste dei primi m elementi ordinati come $S_{j,m}$ e $S_{l,m}$, la *Fraction Enrichment (FE)* a livello m, n_k fra j ed l sarà

$$FE_{m,n_k}(j,l) = |S_{j,m} \cap S_{l,m}| \tag{4.5}$$

La FE può essere calcolata a qualsiasi livello $m \in \{1, \ldots, n_k\}$ ed è possibile anche valutare il relativo p-value (considerando una distribuzione ipergeometrica). In particolare, quest'ultimo viene definito come la probabilità che dati n_k elementi, presi a caso (con reinserimento) due insiemi di numerosità m ciascuno, essi condividono la FE_{m,n_k} di elementi in comune o un numero maggiore; quindi, formalmente, $p - value = Pr(FE_m > FE_m^{obs} | n_k, m)$. Tale misura può essere calcolata fra due pazienti j e l e per ogni valore di m. Inoltre, il *valore minimo* del vettore di p-values di lunghezza m, può essere usato come misura di similarità e viene chiamato **Fraction Enrichment p-value**, $FEP(j,l)$. Per un minore peso computazionale, il calcolo del p-value (basato sulla distribuzione ipergeometrica) può avvenire ad intervalli di ampiezza definita rispetto alle esigenze del ricercatore. Tuttavia, per le tre matrici in analisi non è stata applicata tale esemplificazione.

Sempre per ogni valore di $m \in \{1, \ldots, n_k\}$ è possibile calcolare la somma cumulata fino all'ordine t della $FE_{m,n_k}(j,l)$, ovvero

$$FE_t = \sum_{m=1}^{t} FE_{m,n_k}(j,l) \quad per \quad t = 1, \ldots, n_k$$

Come misura riassuntiva della similarità fra j ed l viene utilizzata la somma all'$n_k - esimo$ ordine della FE. Quindi, calcolando la formula di cui sopra per $t = n_k$ e normalizzandola, si

ottiene la **Fraction Enrichment Sum (FES)**

$$FES = \frac{\sum_{m=1}^{n_k} FE_{m,n_k}(j,l) - min}{max - min} \tag{4.6}$$

dove

$$max = \frac{n_k(n_k+1)}{2} \qquad min = \left(\frac{n_k + n_k \mod 2}{2}\right)\left(\frac{n_k - n_k \mod 2}{2} + 1\right)$$

Su tale misura è stato proposto [21] un sistema di *pesi esponenzialmente decrescenti* $e^{-\alpha(i,j)m}$,

$$FES = \frac{\sum_{m=1}^{n_k} FE_{m,n_k}(j,l)e^{-\alpha(i,j)am} - min(\alpha(i,j))}{max(\alpha(i,j)) - min(\alpha(i,j))} \tag{4.7}$$

secondo il quale viene dato un peso sempre più basso al crescere dell'ordine m considerato. Di conseguenza, anche il calcolo del valore minimo e del valore massimo saranno condotti in funzione di tale nuova definizione e dipenderanno dal parametro $\alpha(i,j)$ (come esplicitato nella Equazione (4.7)). Il massimo e il minimo rappresentano due circostanze contrastanti, relative ai possibili ordinamenti delle due liste di elementi: il massimo si riferisce al caso in cui le due liste per ogni ordine $m \in \{1,\ldots,n_k\}$ si sovrappongono perfettamente dal primo all'ultimo elemento; invece, il minimo si osserva quando le due liste sono ordinate nel modo opposto e quindi, semplificando i ragionamenti, non si ha alcun elemento in comune fra esse per almeno metà lista.

Il parametro α è stato scelto in base al criterio proposto da Serra et al. (2016) [21], secondo il quale $\alpha(i,j) = 1/m_{opt}(i,j)$, dove $m_{opt}(i,j)$ è definito per *ogni coppia di pazienti* (j,l) (con $j \neq l$) come il numero di elementi in comune corrispondente al valore della $FEP(j,l)$.

Definite le nozioni necessarie, dalla **FES** è possibile ricavare la **misura di dissimilarità** d_{FES}

$$d_{FES} = 1 - \frac{FES + FES_r}{2} \tag{4.8}$$

dove FES_r è la FES calcolata secondo l'ordinamento inverso delle liste di elementi (dal valore di espressione/punteggio più grande al valore di espressione/punteggio più piccolo). Come definito dagli autori, tale scelta è finalizzata a dare il medesimo peso agli elementi sottoespressi e sovraespressi.

L'applicazione della misura di distanza d_{FES} alle matrici prese in considerazione ha condotto alle heatmap rappresentate in Figura (4.3). Si osserva come nel caso dell'insieme *DEG* la d_{FES}

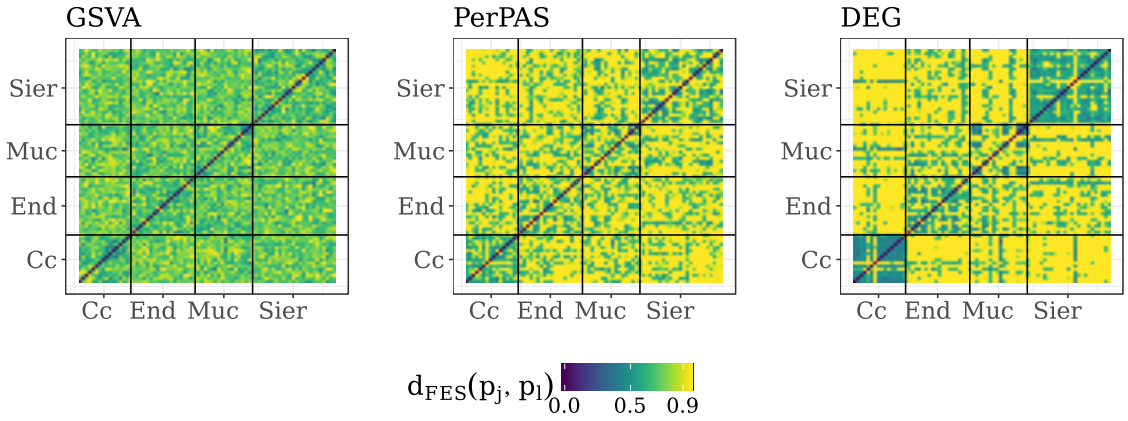


Figura 4.3: Heatmap misura di distanza basata sulla FES (GSVA, PerPAS e DEG).

sia riuscita a delineare in modo soddisfacente le forti somiglianze interne ai quattro gruppi di pazienti. Invece, per quanto concerne gli altri due, la misura non riesce in modo chiaro a definire un raggruppamento sottostante, ad eccezione del PerPAS in cui si osserva un andamento migliore che permette, anche se in modo poco chiaro, di evincere somiglianze interne ai gruppi ed un comportamento opposto tra elementi di gruppi diversi.

4.2 Stima della matrice di adiacenza

Un passo fondamentale per ricavare l'evidenza empirica base degli algoritmi che verranno proposti nel Capitolo 5 è la *stima della matrice di adiacenza*. In altri termini, è necessario definire una procedura che a partire da un *input* quale la *matrice di distanza* (sia euclidea che FES) restituisca un *output*, ovvero la *matrice di adiacenza*.

Un criterio per la definizione del cut-off

Le misure di distanza (o di dissimilarità) definite nel Paragrafo 4.1 rappresentano l'elemento iniziale per la specificazione di un metodo che permetta di stimare la matrice di adiacenza.

Per prima cosa, è necessario specificare che la misura di distanza euclidea (d_E) assume valori in $\mathbb{R}_0^+$; invece, la misura di distanza basata sulla FES (d_{FES}) assume valori in $[0, 1]$. Pertanto, data una qualsiasi misura di distanza d, la **matrice di adiacenza** A_q definita dal cut-off delle

distanze in $q \in S_q$ (S_q insieme dei possibili cut-off) sarà ricavata nel seguente modo:

$$\{A_q\}_{n\times n} = \begin{cases} a_{ij} = 1 \quad \text{se} \quad d \leq q \quad \forall (i,j), \quad i < j, \quad i,j = 1,\ldots,n \\ a_{ij} = 0 \quad \text{se} \quad d > q \quad \forall (i,j), \quad i < j, \quad i,j = 1,\ldots,n \end{cases}$$

Dunque, q definisce il **cut-off** secondo cui le distanze più piccole rispetto ad esso costituiranno l'evidenza empirica verso una relazione fra le specifiche coppie di repliche biologiche; il contrario, invece, sarà per i casi in cui si osserveranno distanze più grandi di q. Inoltre, si assume che gli elementi nella diagonale principale siano $a_{ii} = 1$ $(i = 1,\ldots,n)$. In particolare, questa caratteristica in un grafo prende il nome di *self-loop* ma non influirà nello svolgimento delle analisi successive.

Il passo fondamentale consiste nel trovare il modo più semplice possibile attraverso il quale ricondursi ad una matrice di adiacenza che possa descrivere una rete le cui caratteristiche siano quelle classiche di una *rete genica (o sociale)*, cioè una rete dotata di **invarianza di scala (scale-free)** [29]. In altri termini, si ricerca quel cut-off q tale che la **distribuzione del grado (degree) di nodo** segua la *legge di potenza*

$$p(k) \approx f(k,\gamma) = k^{-\gamma} \quad \text{dove} \quad \gamma \in (2,3) \tag{4.9}$$

secondo cui la probabilità che si osservi un grado pari a k sia approssimativamente uguale a $k^{-\gamma}$. In Figura 4.4 viene riportato un esempio di distribuzione del grado relativa ad una rete scale-free,

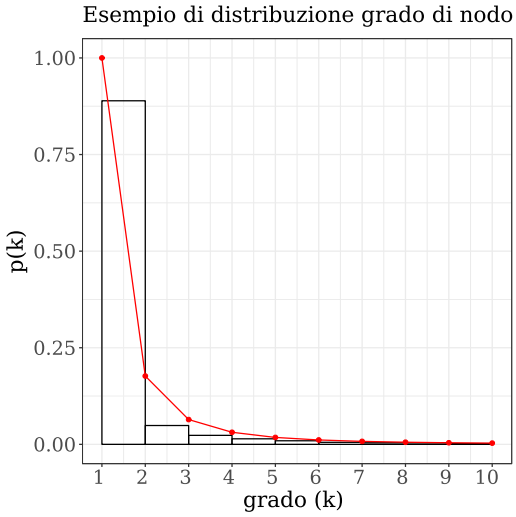


Figura 4.4: Esempio di distribuzione *scale-free* del grado di nodo con $\gamma = 2.5$ (l'istogramma rappresenta la distribuzione empirica del grado di nodo $k \in \{1,2,\ldots,75\}$ e zoomata sui primi 10 valori di k; la linea rossa con i punti rappresenta l'andamento funzionale $k^{-\gamma}$).

La specificazione dell'Equazione 4.9 può essere riformulata nel seguente modo,

$$p(k) = f(k, \beta, \gamma) = \beta k^{-\gamma} \tag{4.10}$$

in cui β è un parametro che, come verrà mostrato in seguito, permetterà di avere una migliore stima di γ.

Quindi, è necessario stimare γ (parametro di interesse) al variare del cut-off q. Tuttavia, l'evidenza empirica a disposizione per la stima di γ risente di alcune problematiche comuni alle matrici di distanza ricavate con qualsiasi misura di dissimilarità. Infatti, definito S_q l'insieme dei cut-off osservabili nella determinata matrice di dissimilarità, per valori di q vicini a $\min\{S_q\}$ e per valori prossimi al $\max\{S_q\}$, la distribuzione del grado (k) risulta fortemente asimmetrica e avere soltanto al più due (o tre) gradi osservati (nel primo caso $k \in \{1, 2, 3\}$, nel secondo caso $k \in \{73, 74, 75\}$). Nelle due circostanze estreme, dunque, il valore di γ tenderà rispettivamente a $+\infty$ e a $-\infty$. Pertanto, definiti i cut-off q osservati come le $n(n-1)/2 = (76 \times 75)/2 = 2850$ misure di distanza potenzialmente diverse e considerate le loro distribuzioni rispetto ad ogni matrice, queste risultano approssimativamente simmetriche e verrà escluso il 5% dei valori più

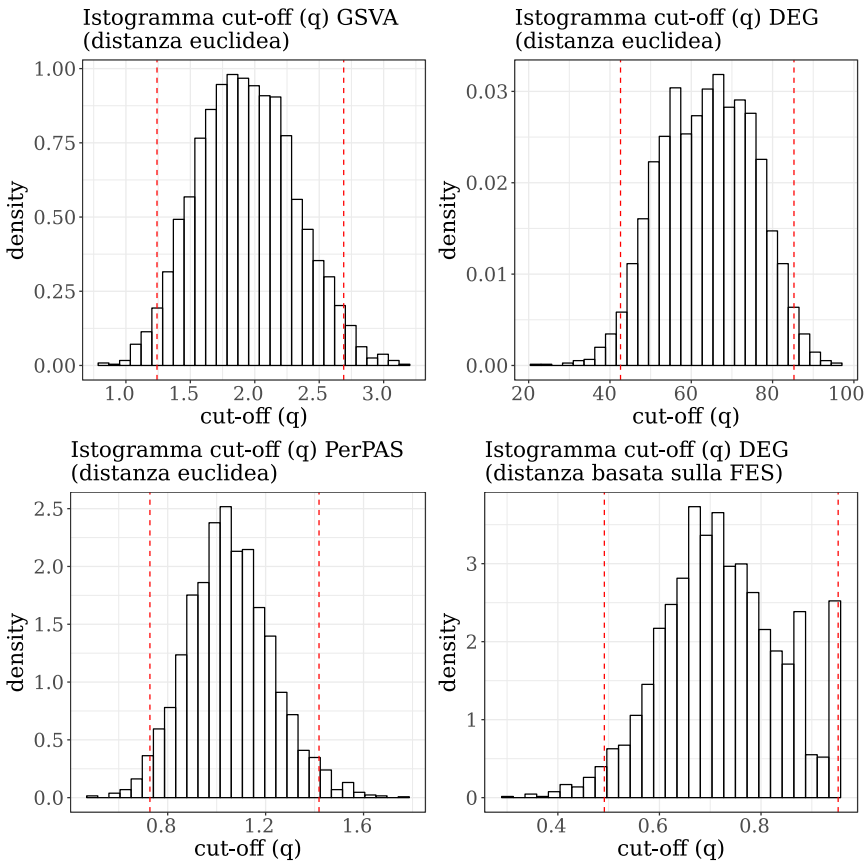


Figura 4.5: Istogrammi cut-off (q): distanza eucliea (GSVA, PerPAS, DEG), distanza basata sulla FES (DEG).

estremi di ogni distribuzione di cut-off (il 2.5% di valori più bassi e il 2.5% di valori più alti); in Figura 4.5 sono riportate le tre distribuzioni dei cut-off delle matrici di distanza euclidea e la distribuzione dei cut-off relativa alla distanza basata sulla FES dei DEG. Inoltre, su ogni distribuzione sono segnati attraverso una linea rossa tratteggiata i valori soglia che hanno determinato l'esclusione del 5% dei cut-off estremi per i quali si presenterebbe la problematica appena discussa. Prima di definire come stimare γ, è necessario conoscere la distribuzione di probabilità del grado, ovvero $p(k)$ per $k \in \{1, 2, \ldots, 75\}$ al variare del valore di $q \in S_q$. La prima e più semplice procedura consisterebbe nella stima di $p(k)$ direttamente dalla distribuzione osservata; ciò però, a diversi valori di q, non è possibile in quanto si conoscono le $p(k)$ soltanto per un sottoinsieme di $k \in \{1, 2, \ldots, 75\}$. Dunque, è utile stimare dette probabilità, anche quando non vengono osservate, sfruttando un'assunzione sulla distribuzione del grado di nodo.

Dato il cut-off $q \in S_q$ e la matrice di adiacenza risultante A_q, si definisce la variabile casuale *grado del nodo* $K \sim Bin(r, p)$ dove $S_K = \{0, 1, \ldots, r\}$, $r = 75$ e $p \in (0, 1)$. Il supporto S_K deve essere opportunamente ridefinito in $\{1, 2, \ldots, r\}$ quindi le probabilità stimate attraverso la distribuzione Binomiale dovranno essere divise per la costante di normalizzazione corretta:

$$1 - Pr(K = 0) = 1 - (1 - p)^r. \tag{4.11}$$

La forma della distribuzione Binomiale non è in grado di adattarsi bene all'asimmetria delle distribuzioni dei gradi di nodo che si osservano al variare di q, in quanto impone implicitamente una simmetria distributiva non necessariamente osservabile; quindi, è opportuno affinare il modello, definendo un modello Bayesiano in cui la distribuzione assunta per i dati rimane, tuttavia, Binomiale

$$Pr(K = k|p, r) = L(k; p, r) = \binom{r}{k} p^k (1 - p)^{r-k} \tag{4.12}$$

e per p viene assunta una distribuzione *a priori* $p \sim Beta(\alpha, \beta)$

$$\pi(p; \alpha, \beta) = \frac{\Gamma(\alpha)\Gamma(\beta)}{\Gamma(\alpha + \beta)} p^{\alpha-1} (1 - p)^{\beta-1} \tag{4.13}$$

con $S_p \in (0, 1)$, $\alpha > 0$, $\beta > 0$. Di conseguenza, l'interesse passa alla distribuzione *a posteriori* di K , per cui la stima della probabilità $p(k)$ che avviene attraverso detta probabilità a posteriori viene riportata nell'Equazione 4.14

$$\widehat{p(k)} = \pi(K = k|r, \alpha, \beta) = \binom{r}{k} \frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha)\Gamma(\beta)} \frac{\Gamma(k + \alpha)\Gamma(r - k + \beta)}{\Gamma(r + \alpha + \beta)}. \tag{4.14}$$

Per i passaggi che hanno condotto al risultato in Equazione 4.14 si consulti l'Appendice A.1. I parametri α e β della distribuzione a priori vengono posti pari alle stime degli stessi ottenute con il metodo dei momenti (MM), ovvero

$$\hat{\alpha}_{MM} = \frac{\bar{p}^2}{\sigma_p^2} \quad \text{e} \quad \hat{\beta}_{MM} = \frac{\bar{p}}{\sigma_p^2} \tag{4.15}$$

dove $\bar{p}$ è la media delle $p(k)$ osservate e σ_p^2 è la varianza empirica delle $p(k)$. Sostituendo tali stime nella Formula 4.14, verrà stimata la probabilità a posteriori che $K = k$.

In Figura 4.6 viene riportato un esempio sulla matrice dei punteggi GSVA e cutoff $q = 2$ (in viola la stima delle probabilità con il modello Binomiale; in rosso la stima delle probabilità con il modello Beta-Binomiale).

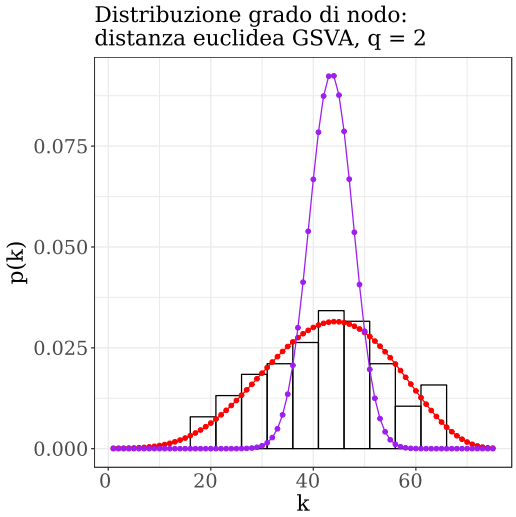


Figura 4.6: Esempio distribuzione grado di nodo: distanza euclidea GSVA, $q = 2$ (in viola la stima delle probabilità con il modello Binomiale; in rosso la stima delle probabilità con il modello Beta-Binomiale).

Anche nel caso del modello Beta-Binomiale sarà necessario rinormalizzare le probabilità dividendo esse per

$$1 - Pr(K = 0|p, r, \alpha, \beta) = 1 - \frac{\Gamma(\alpha + \beta)\Gamma(r + \beta)}{\Gamma(\beta)\Gamma(r + \alpha + \beta)} \tag{4.16}$$

dove in α e β vengono sostituite le stime ottenute con il metodo dei momenti nella Formula 4.15.

Una volta ottenuta una stima parametrica della distribuzione del grado di nodo, è possibile stimare attraverso i **minimi quadrati ordinari (OLS)** il parametro di interesse γ del modello lineare in Equazione 4.17 (trasformazione logaritmica della forma specificata nell'Equazione

4.10)

$$log(p(k)) = log(f(k, \beta, \gamma)) = log(\beta) - \gamma log(k). \tag{4.17}$$

La funzione di perdita da minimizzare sarà quella classica,

$$(\hat{\gamma}, \hat{\beta}) = \arg\min_{\gamma,\beta} \left\{ \sum_{k=1}^{75} [log(\widehat{p(k)}) - log(\beta) + \gamma log(k)]^2 \right\} \quad \beta > 0, \gamma \in \mathbb{R} \tag{4.18}$$

Dunque, data la matrice di dissimilarità basata sulla misura di distanza d e l'insieme di cut-off S_q, al variare di $q \in S_q$ vengono stimati $\hat{\gamma}(q)$ e $\hat{\beta}(q)$, minimizzando la quantità in Equazione 4.18 dove $\widehat{p(k)}$ dipenderà da q e verrà stimato attraverso il modello Beta-Binomiale, $k \in \{1, 2, \ldots, 75\}$ $\forall q \in S_q$. In questo modo, disinteressandosi del parametro di disturbo β e focalizzandosi sul parametro di interesse γ, è possibile rappresentare l'andamento della stima di γ in funzione di q, ovvero $\gamma(q)$ dove $q \in S_q$. In Figura 4.7, vengono riportati i grafici di $\gamma(q)$ relativi alle tre matrici di distanza euclidea (GSVA, PerPAS, DEG) e alla matrice di distanza basata sulla FES (DEG). Si osserva l'andamento decrescente di γ al crescere del cut-off q, indicando la presenza

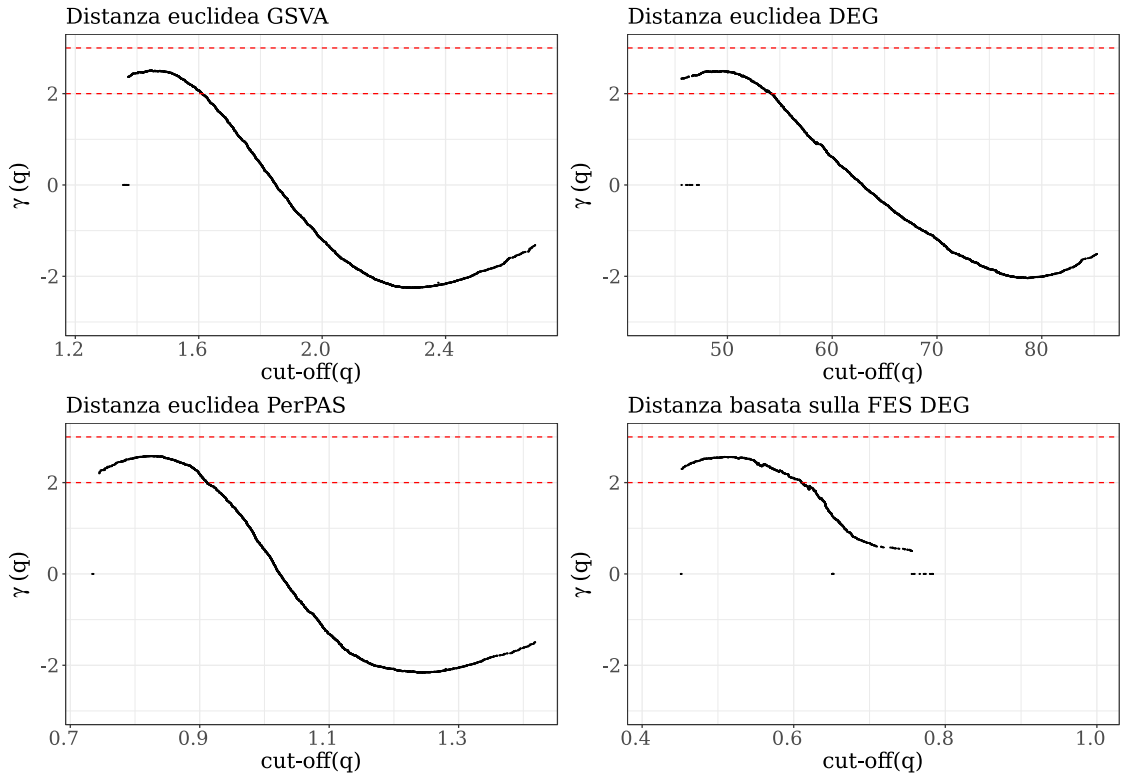


Figura 4.7: Stima di γ al variare del cut-off q (matrici di distanza euclidea: GSVA, PerPAS, DEG; matrice di distanza basata sulla FES: DEG); le linee tratteggiate in rosso delimitano l'insieme di valori $\gamma \in (2, 3)$.

di reti *scale-free* per cut-off q vicini a $\min\{S_q\}$. Quindi, si evince una propensione ad avere delle reti potenzialmente poco dense ma con una distribuzione del grado di nodo che soddisfi

la forma funzionale finora studiata.

In Figura 4.8 viene riportato un ingrandimento dei grafici in Figura 4.7 per i valori di $\gamma \in (2, 3)$, il quale mostra implicitamente l'insieme di q che conducono a delle reti *scale-free*.

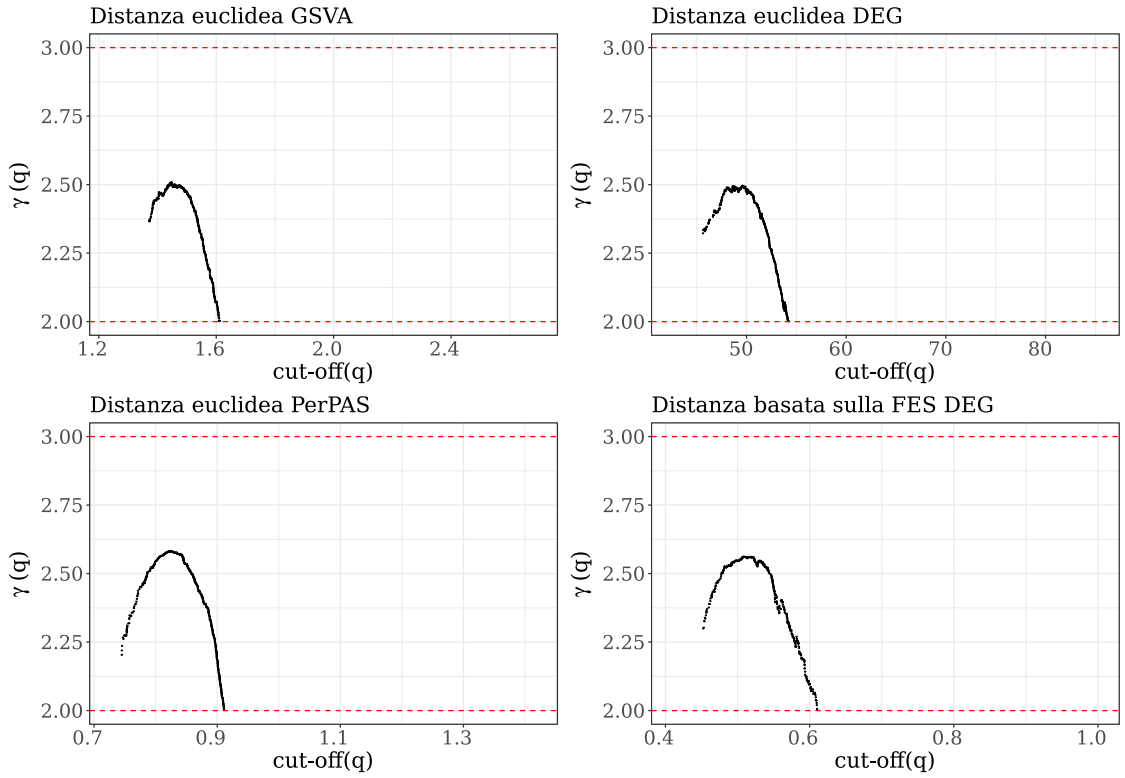


Figura 4.8: Ingrandimento sull'intervallo $\gamma \in (2, 3)$ (matrici di distanza euclidea: GSVA, PerPAS, DEG; matrice di distanza basata sulla FES: DEG); le linee tratteggiate in rosso delimitano gli estremi dell'intervallo.

Tuttavia, manca un criterio secondo cui scegliere una q fra quelle per le quali $\gamma(q) \in (2, 3)$. In modo semplice, è stato scelto il cutt-off che non restituisca una matrice troppo sparsa.

Come scritto all'inizio di questo paragrafo, la forma funzionale $p(k) = \beta k^{-\gamma}$ non solo offre un adattamento relativamente migliore rispetto alla forma più semplice $p(k) = k^{-\gamma}$ ma permette anche di ottenere una stima corretta di γ per quei cut-off in cui con il modello semplificato le stime del parametro di interesse non rispecchierebbero il reale andamento della distribuzione del grado di nodo. In Tabella 4.1, il decremento che si osserva, seppur di lieve entità ha, tuttavia,

Devianze residue modello per $p(k)$ con e senza la stima del parametro β

Devianza residua	**GSVA**(d_E)	**PerPAS**(d_E)	**DEG**(d_E)	**DEG**(d_{FES})
con β	253.247	168.438	234.394	264.404
senza β	253.193	165.335	233.813	263.440

Tabella 4.1: Confronto devianze residue del modello per $log(p(k))$ con e senza la stima del parametro $log(\beta)$.

un modesto impatto sulla possibilità di stimare γ in modo più corretto. Lo si può osservare

riportando la stima di γ con il modello semplice insieme alla stima ottenuta con l'inserimento del parametro β in Figura 4.10, dove si osserva che per valori di q alti, la stima di γ del modello

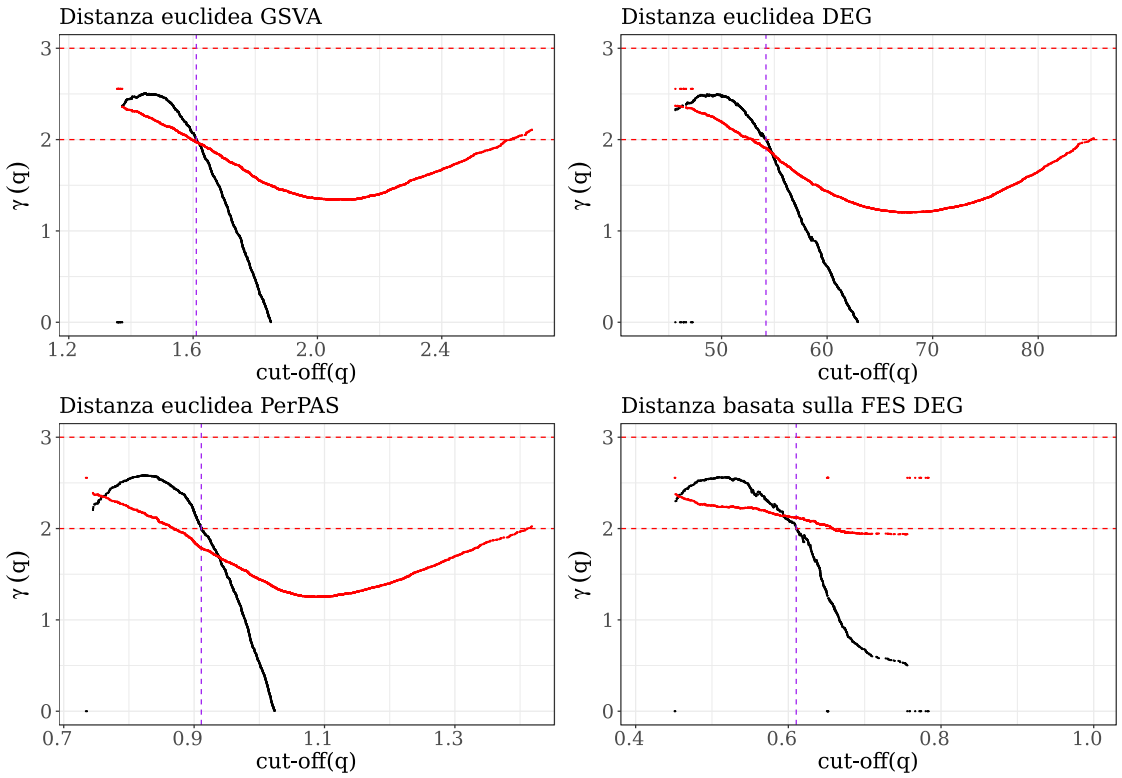


Figura 4.9: Confronto andamento stima di γ al variare di q ottenuta attraverso il modello con e senza β (in rosso: modello senza β; in nero: modello con β). La linea tratteggiata viola indica il cut-off ottimale (q_{opt}, ottenuto dalla stima di γ del modello con β); le due linee tratteggiate in rosso indicano gli estremi dell'intervallo $(2, 3)$.

senza β cresce e in alcuni casi assume persino valori compresi in $(2, 3)$. Questo comportamento non è concorde con l'andamento della distribuzione del grado di nodo a valori alti del cut-off in quanto evidenzia un andamento completamente opposto a quello che dovrebbe assumere: ovvero γ dovrebbe essere negativo.

Adesso, non rimane che stimare le matrici di adiacenza secondo i q_{opt} precedentemente ricavati. Quindi, data la matrice di distanza d, si definisce $A_{q_{opt}}$ come

$$\{A_{q_{opt}}\}_{n\times n} = \begin{cases} a_{ij} = 1 \quad \text{se} \quad d \leq q_{opt} \quad \forall (i,j), \quad i < j, \quad i,j = 1,\ldots,n \\ a_{ij} = 0 \quad \text{se} \quad d > q_{opt} \quad \forall (i,j), \quad i < j, \quad i,j = 1,\ldots,n \end{cases}$$

In Figura 4.11 vengono riportate le quattro matrici di adiacenza risultanti. Si noti come nel caso dell'insieme dei DEG le matrici riescano ad evidenziare in modo soddisfacente i raggruppamenti secondo le quattro condizioni istologiche in analisi. Invece, nelle matrici relative al GSVA e PerPAS i raggruppamenti si evincono in modo meno evidente. A scopo descrittivo sono

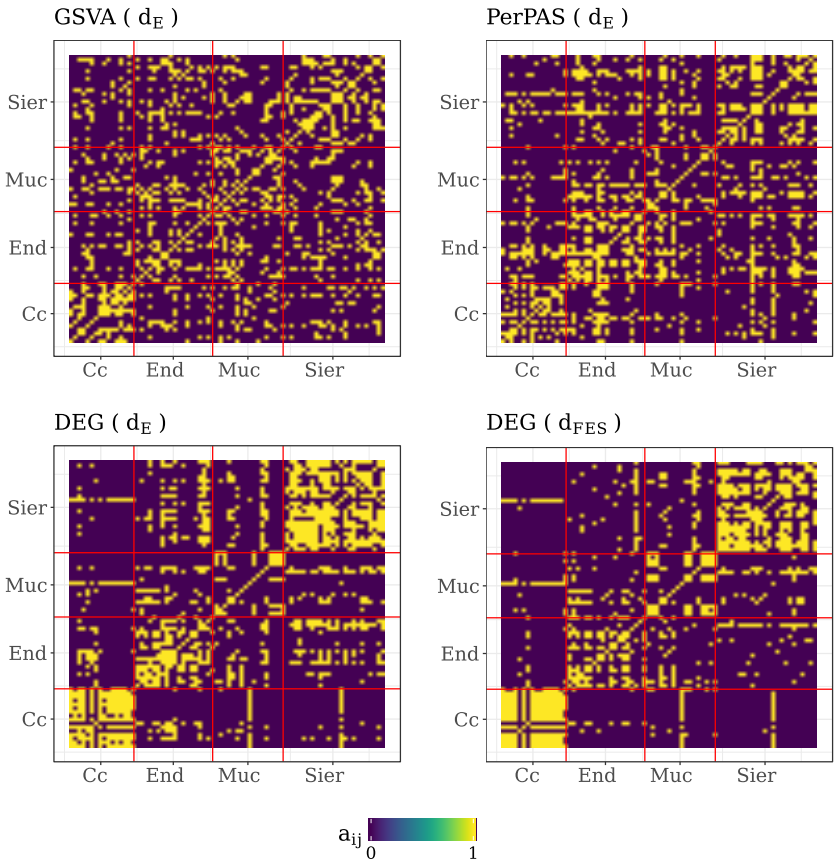


Figura 4.10: Matrici di adiacenza stimate sulla base delle quattro matrici di distanza iniziali (distanza euclidea e distanza basata sulla FES).

state calcolate le principali statistiche di rete (riportate in Tabella 4.2) per condurre le prime considerazioni sulle reti stimate.

Statistche descrittive sulle quattro reti stimate							
rete	**cut-off**	**densità**	**transitività globale**	**betweenness normalizzata (mediana)**	**degree mediano**	**shortest path medio**	**diametro**
GSVA (d_E)	1.61	0.208	0.455	0.011	16	2.011	4
PerPAS (d_E)	0.911	0.211	0.507	0.006	14	2.027	4
DEG (d_E)	54.207	0.213	0.618	0.007	15	2.279	5
DEG (d_{FES})	0.61	0.176	0.689	0.009	14	2.685	6

Tabella 4.2: Statistiche descrittive sulle quattro reti stimate. Le colonne indicano: la rete a cui si fa riferimento, il cut-off che ha determinato la loro stima, la densità della rete risultante, la transitività globale, la mediana della distribuzione delle betweenness normalizzate, la mediana della distribuzione del grado di nodo, la media dei cammini più corti (shortest paths), il diametro della rete.

Le reti stimate sono poco dense (la densità si aggira intorno a 0.2), hanno una transitività globale moderata, quindi i nodi risultano abbastanza interconnessi fra loro; la mediana della distribuzione delle betweenness normalizzate risulta molto bassa e vicina allo zero, indice che almeno metà dei nodi hanno un'influenza quasi nulla all'interno delle rispettive reti. Il grado di nodo mediano evidenzia un numero di connessioni non troppo elevato per almeno metà dei

nodi (tra 14 e 16 legami); la media dei cammini più corti, tuttavia, pone in risalto la vicinanza media fra due nodi non connessi; il diametro delle reti risulta relativamente basso, risultando poco disconnesse.

Capitolo 5

Modelli a Blocchi Stocastici: applicazione sulle reti di pazienti

I **modelli a blocchi stocastici (stochastic block models, SBM)** costituiscono un insieme di *modelli generativi* i quali, mediante assunzioni probabilistiche su delle specifiche **dimensioni latenti** e sul **processo generatore delle relazioni** osservate in un grafo, hanno il fine ultimo di ricavare informazioni circa le comunità sottostanti la rete in analisi e le loro caratteristiche.

Le tecniche che permettono agli *SBM* di perseguire l'obiettivo o di *detection (weak recovery)*, o di *partial recovery*, fanno parte di quelle metodologie di **apprendimento non supervisionato** (unsupervised learning), ad eccezione, infatti, delle tecniche di *exact recovery* in cui è già necessario che l'agoritmo sia a conoscenza, ad esempio, del numero e della dimensione delle comunità che sottostanno ad una rete. Nelle analisi successive verranno proposte due tecniche che, in base alle informazioni in uso dagli algoritmi, sono da inquadrare in un contesto di **partial recovery**, in quanto non solo si ricercheranno i raggruppamenti latenti ma questi verranno anche confrontati con il vero raggruppamento dei pazienti. Quindi, per definizione di apprendimento non supervisionato, gli algoritmi non saranno a conoscenza del vero raggruppamento [30] . In breve, le informazioni che si cercherà di ricavare mediante gli algoritmi che implementano gli SBM sono le caratteristiche delle comunità quali:

- le probabilità di appartenenza agli m gruppi (comunità);
- la classificazione dei nodi della rete secondo le m comunità;
- una matrice di dimensioni $m \times m$ che definisca quantitativamente le relazioni intra e fra i gruppi.

5.0.1 Le dimensioni latenti

Gli SBM che verranno trattati considereranno dei *grafi non direzionati* e *binari* $G = \{N, A\}$, dove N è l'insieme di nodi (numero di pazienti) e A è la matrice di adiacenza. Sulla base di queste informazioni, si cercherà di ottenere nel modo più affidabile possibile delle informazioni circa le tre *dimensioni latenti* presenti nella specificazione del modello [24]:

- il vettore di probabilità $\alpha = (\alpha_1, \ldots, \alpha_m)$ di appartenenza agli m gruppi.
- matrice $\{Z\}_{N \times m}$ di classificazione di ciascun paziente in uno degli m gruppi;
- matrice $\{\theta\}_{m \times m}$ che descrive le relazioni quantitative intra e fra i gruppi.

5.0.2 Formulazione dei modelli probabilistici

Sono stati specificati due modelli diversi:

- *modello di Bernoulli* in cui:
  - $Z_i \sim \mathcal{M}\big(1; \alpha = (\alpha_1, \ldots, \alpha_m)\big), \quad \forall i = 1, \ldots, N$, ovvero la distribuzione di Z_i è *multinomiale* di parametri $\alpha = (\alpha_1, \ldots, \alpha_m)$ vettore (latente) di probabilità di appartenenza agli m gruppi;
  - $A_{ij}|Z_i = k, Z_j = l \sim Ber(\eta_{kl}) \quad \forall k \leq l, (k, l) = 1, \ldots, m$, dove η_{kl} (latente) definisce la probabilità che un elemento appartenente al gruppo k e un elemento appartenente al gruppo l abbiano una relazione (connessione, arco);
- *modello di Poisson* in cui:
  - $Z_i \sim \mathcal{M}\big(1; \alpha = (\alpha_1, \ldots, \alpha_m)\big), \quad \forall i = 1, \ldots, N$ come nel modello di Bernoulli, $\alpha = (\alpha_1, \ldots, \alpha_m)$ è il vettore (latente) di probabilità di appartenenza ai m gruppi;
  - $A_{ij}|Z_i = k, Z_j = l \sim Poisson(\omega_{kl}) \quad \forall k \leq l, (k, l) = 1, \ldots, m$, dove ω_{kl} (latente) specifica il numero di relazioni (archi) medio fra un elemento appartenente al gruppo k e un elemento appartenente al gruppo l.

Nel modello di Poisson la matrice di adiacenza a cui si farà riferimento sarà calcolata come somma delle due matrici di adiacenza ottenute con GSVA, PerPAS e DEG (d_{FES}). Questa integrazione di informazioni, tuttavia, accumulerà del rumore di fondo presente nelle due matrici distinte.

Ulteriori specificazioni

Potrebbe essere utile specificare nei modelli la presenza di caratteristiche o di nodo o di rete che vadano ad influire sul processo di *partial recovery* ottenendo dei raggruppamenti auspicabilmente ottimali (ad esempio dei biomarcatori tumorali). Tuttavia, nei modelli oggetto di studio non è stata specificata alcuna caratteristica a livello di nodo/rete.

5.1 Variational Bayes Expectation-Maximization

Tutti gli aspetti metodologici del Paragrafo 5.1 sono stati presi dal lavoro di Blei et. al (2017) [8] per quanto riguarda la parte di Variational Inference, dal lavoro di Latouche et. al (2012) [15] per quanto riguarda l'approccio di Variational Inference applicato in un contesto Bayesiano.

La **Variational Inference** consiste in una tecnica di machine learning che cerca di ricavare densità di probabilità mediante algoritmi di ottimizzazione. L'obiettivo per cui viene utilizzato un approccio di *variational inference* è quello di approssimare distribuzioni di probabilità condizionata di *variabili latenti* date delle *variabili osservate*.

Una soluzione proposta è quella di introdurre una famiglia di densità sulle variabili latenti, parametrizzata da "*parametri variazionali a priori*" e che sia auspicabilmente simile alla vera famiglia da cui esse provengono.

Definizione metodologica

Sia $Y = (Y_1, \ldots, Y_n)$ un insieme di n variabili osservate e $Z = (Z_1, \ldots, Z_m)$ un insieme di m variabili latenti con distribuzione congiunta

$$p(Y, Z). \tag{5.1}$$

L'obiettivo della *variational inference* è di esplicitare la distribuzione delle variabili latenti Z date le variabili osservate Y, ovvero

$$p(Z|Y). \tag{5.2}$$

La distribuzione condizionata potrà essere utilizzata per fornire delle stime (sia puntuali che intervallari) sulle variabili latenti o per scopi predittivi. Essa può essere scritta come

$$p(Z|Y) = \frac{p(Y, Z)}{p(Y)}, \quad \text{dove} \quad p(Y) = \int_Z p(Y, Z) dZ \tag{5.3}$$

L'integrale in Formula 5.3 rappresenta la densità marginale (chiamata anche *evidence*) il cui calcolo ha un peso computazionale non trascurabile.

La Mean-field variational family e l'Evidence Lower BOund (ELBO)

Nel problema di ottimizzazione, viene specificata una particolare *variational family* $\mathcal{Q}$ di densità a priori sulle variabili latenti che assume la *mutua indipendenza* di queste ultime. Quindi, date m variabili latenti $Z = (Z_1, \ldots, Z_m)$ la **mean-field variational family** $\mathcal{Q}$ è definita come

$$q(Z) = \prod_{j=1}^{m} q_j(Z_j) \tag{5.4}$$

dove ciascuna variabile latente Z_j $(j = 1, \ldots, m)$ è specificata da una propria componente variazionale $q_j(Z_j) \in \mathcal{Q}$ $(j = 1, \ldots, m)$. Si noti che la $q(Z)$ non risulta dipendere dai dati (variabili osservate) Y. Le componenti di $\mathcal{Q}$ sono considerate come distribuzioni candidate nell'approssimazione della vera distribuzione condizionata (la complessità della famiglia determinerà la complessità dell'ottimizzazione.). Pertanto, lo scopo è quello di trovare le distribuzioni migliori delle dimensioni latenti la cui divergenza di Kullback-Leibler con la vera distribuzione risulti minima,

$$q^*(Z) = \arg\min_{q(Z)\in\mathcal{Q}} KL\big(q(Z)||p(Z|Y)\big) \tag{5.5}$$

Tuttavia, tale ottimizzazione richiede il calcolo della *evidence* $p(Y)$ come mostrato in Appendice A.2. Dato che non è possibile calcolare la KL, viene ottimizzata una funzione obiettivo chiamata **Evidence Lower BOund (ELBO)** uguale alla KL cambiata di segno e sommata alla *evidence*,

$$ELBO(q) = \mathbb{E}\big[\ln p(Y, Z)\big] - \mathbb{E}\big[\ln q(Z)\big]. \tag{5.6}$$

Quindi, massimizzare la $ELBO$ equivale a minimizzare la divergenza di KL. È possibile riscrivere la Formula in 5.6 come

$$\begin{aligned} ELBO(q) = \mathbb{E}\big[\ln p(Z)\big] + \mathbb{E}\big[\ln p(Y|Z)\big] - \mathbb{E}\big[\ln q(Z)\big] = \\ = \mathbb{E}\big[\ln p(Y|Z)\big] - KL\big(q(Z)||p(Z)\big) \end{aligned} \tag{5.7}$$

ovvero come il valore atteso della log-verosimiglianza e la divergenza di KL fra $p(Z)$ e $q(Z)$. Una proprietà della $ELBO$ è che essa definisce il limite inferiore della *log-evidence*, $\log p(Y)$, ovvero $\ln p(Y) \geq ELBO(q)$, dove $\ln p(Y) = KL\big(q(Z)||q(Z|Y)\big) + ELBO(q)$ e $KL(\cdot) \geq 0$ sempre. Per tale motivo, $ELBO(q)$ si propone come una buona approssimazione della log-verosimiglianza marginale e quindi costituisce un buon criterio per la selezione del modello.

Algoritmo CAVI

Quello che manca è un algoritmo che permetta di risolvere il problema di ottimizzazione in questione. Uno degli algoritmi più comuni per la risoluzione di tale problema è il **Coordinate Ascent Variational Inference (CAVI)**. Tale algoritmo ottimizza in modo iterativo ciascuna componente della *mean-field variational family*, separatamente dalle restanti componenti; inoltre, esso conduce l'$ELBO$ ad un *ottimo locale*.

Si consideri la $j-esima$ variabile latente Z_j e la distribuzione condizionata di Z_j rispetto alle restanti variabili latenti $p(Z_j|Z_{-j}, Y)$. L'ottimizzazione della componente $q_j(Z_j)$ è proporzionale all'elevamento a potenza del valore atteso del logaritmo della distribuzione condizionata della componente $j - esima$ (calcolato rispetto alle restanti componenti),

$$q_j^*(Z_j) \propto \exp\{\mathbb{E}_{-j}[\ln p(Z_j|Z_{-j}, Y)]\} \tag{5.8}$$

L'equazione 5.8 è proporzionale all'elevamento a potenza della distribuzione congiunta

$$q_j^*(Z_j) \propto \exp\{\mathbb{E}_{-j}[\ln p(Z_j, Z_{-j}, Y)]\} \tag{5.9}$$

Dato che le variabili latenti sono mutuamente indipendenti, i valori attesi non coinvolgono la componente $j - esima$. Dopo aver ottimizzato le variabili latenti, viene calcolata la $ELBO(q)$ ed iterato il processo fino a convergenza, ovvero fin quando la differenza fra il nuovo $ELBO(q)$ e quello calcolato al passo precedente non risulti più piccola di un valore specificato dal programmatore.

5.2 In un contesto Bayesiano: Variational Bayes

In un contesto Bayesiano [15] in cui le dimensioni latenti riguardano sia le variabili casuali che i parametri trattati come variabili casuali, si definisce la seguente specificazione di un SBM:

- le variabili casuali:
  - $A_{ij}|Z_i = k, Z_j = l \sim \mathcal{P}(\theta_{kl})$ per $k \leq l$, dove $\mathcal{P}$ è una distribuzione di probabilità (ad esempio appartenente alla famiglia esponenziale) assunta per descrivere le relazioni fra le componenti i e j (per $(i, j) = 1, \ldots, N, \quad i < j$) dei gruppi rispettivamente k ed l (con supporto $S_{\mathcal{P}}$) e θ_{kl} è il parametro che riassume la relazione fra i due gruppi con spazio parametrico S_θ. La matrice risultante avrà dimensioni $m \times m$, definita come $\{\theta\}_{m\times m}$;

- $Z_i \sim \mathcal{M}(1;\alpha)$ per $i = 1, \ldots, N$, *variabile casuale latente* con distribuzione *multinomiale* parametrizzata da $\alpha = (\alpha_1, \ldots, \alpha_m)$.

- le distribuzioni a priori sui parametri α e θ_{kl} (per $k \leq l$) (trattate quindi come variabili casuali, anch'esse sono *latenti*):

  - $\alpha \sim Dir(n^0)$, il cui supporto sarà $S_\alpha = (0,1)^m$ ed $n^0 = (n_1^0, \ldots, n_m^0)$ è il vettore di parametri *a priori*;
  - $\theta_{kl} \sim \mathcal{H}(\upsilon), \quad k \leq l$ per $(k,l) = 1, \ldots, m$, dove $\mathcal{H}$ è una distribuzione di probabilità a priori sui parametri che descriveranno la relazione fra due gruppi (ad esempio una distribuzione a priori coniugata al modello assunto per la variabile casuale A), avente supporto S_θ; υ è il vettore di m parametri a priori definiti in un opportuno spazio parametrico S_υ di dimensione m.

Dunque, le dimensioni latenti saranno (Z, α, θ) e l'obiettivo è quello di approssimare nel miglior modo possibile la distribuzione $p(Z, \alpha, \theta|A)$ in quanto si è interessati a conoscere le sue caratteristiche.

La log-verosimiglianza marginale (*evidence*) $\ln p(A)$ sarà calcolata come

$$\ln p(A) = \mathbb{E}\big[\ln p(A, Z, \alpha, \theta)\big] + KL\big(q(Z, \alpha, \theta)||p(Z, \alpha, \theta|A)\big) \tag{5.10}$$

dove

$$\begin{aligned} ELBO(q(Z,\alpha,\eta)) = \mathbb{E}\big[\ln p(A, Z, \alpha, \theta)\big] &= \sum_Z \int\int q(Z,\alpha,\theta) \ln\left\{\frac{p(A,Z,\alpha,\theta)}{q(Z,\alpha,\theta)}\right\} d\alpha d\theta \\ KL\big(q(Z,\alpha,\theta)||p(Z,\alpha,\theta|A)\big) &= -\sum_Z \int\int q(Z,\alpha,\theta) \ln\left\{\frac{p(Z,\alpha,\theta|A)}{q(Z,\alpha,\theta)}\right\} d\alpha d\theta \end{aligned} \tag{5.11}$$

Allo stesso modo della Formula 5.7, minimizzare la divergenza di KL nella Formula 5.11 equivale a massimizzare il limite inferiore $ELBO$ rispetto a $q(Z, \alpha, \theta)$.

In questo modo, cercando di approssimare $p(Z, \alpha, \theta|A)$ con $q(Z, \alpha, \theta)$, ci si riconduce ad un problema di ottimizzazione mediante *Variational Inference*. Pertanto, la *mean-field variational family* $q(Z, \alpha, \theta)$ viene fattorizzata nel seguente modo (assumendo, quindi, la mutua indipendenza)

$$q(Z, \alpha, \theta) = q(\alpha)q(\theta)q(Z) = q(\alpha)q(\theta)\prod_{i=1}^{n} q(Z_i). \tag{5.12}$$

E-step ed M-step

La procedura di ottimizzazione, sulla base dell'algortimo CAVI, sarà definita in tre passi passi principali (predizione, massimizzazione mediante aggiornamento di parametri e calcolo dell'$ELBO$), iterati fin quando il limite inferiore non raggiungere la convergenza (definita attraverso dei parametri scelti dall'operatore, ad esempio un parametro numerico ϵ sulla differenza fra due limiti inferiori successivi o il *numero di iterazioni* massimo):

- **Expectation-step (E-step)**: il quale consiste nel passo di *predizione* delle probabilità di ciascuna delle Z_i in ognuno dei m gruppi. Quindi fissato il gruppo k verrà calcolato per $i = 1, \ldots, N$

$$\ln q(Z_i)_k = \mathbb{E}_{Z \setminus i, \alpha, \theta}\big[\ln p(A, Z, \alpha, \theta)\big]_k + cost. \tag{5.13}$$

  dove il valore valore atteso delle parti del modello che dipendono funzionalmente soltanto da Z_i viene calcolato rispetto a tutte le variabili casuali esclusa la componente $i - esima$ di Z (come specificato con le due Formule equivalenti 5.8 e 5.9);

- **Maximization-step (M-step)**: il passo di *massimizzazione* si divide in due parti:

  - *ottimizzazione di α (vettore di probabilità di appartenenza a ciascuno dei m gruppi)*

$$\ln q(\alpha) = \mathbb{E}_{Z, \theta}\big[\ln p(A, Z, \alpha, \theta)\big] + cost. \tag{5.14}$$

    dove il valore atteso di α viene calcolato rispetto alle restanti variabili casuali.

  - *ottimizzazione di θ (per ogni componente di $\{\theta\}_{m \times m}$)*

$$\ln q(\theta) = \mathbb{E}_{Z, \alpha}\big[\ln p(A, Z, \alpha, \theta)\big] + cost. \tag{5.15}$$

- **calcolo dell'$ELBO$**:

$$ELBO(q) = \mathcal{L}(q(Z, \alpha, \theta)) = \mathbb{E}\big[\ln p(A, Z, \alpha, \theta)\big] \tag{5.16}$$

  dove il valore atteso è calcolato rispetto a tutte le componenti della *mean-field variational family*, quindi (Z, α, θ).

Inizializzazione dell'algoritmo: Spectral clustering

L'approccio *Variational Bayes Expectation-Maximization (VBEM)*, basandosi su un'ottimizzazione iterativa risente di una debolezza relativa all'inizializzazione dell'algoritmo stesso

che potrebbe far ricadere la procedura in un ottimo locale. Pertanto, risulta necessario inizializzare l'algoritmo in modo tale che venga minimizzata la probabilità di ritrovarsi con risultati relativi ad un ottimo locale.

Una delle soluzioni al problema di inizializzazione è l'implementazione della tecnica di **spectral clustering** [18], la quale consiste nella decomposizione del *Laplaciano normalizzato* relativo al grafo in analisi. Sia dato un grafo con matrice di adiacenza A e matrice diagonale dei gradi di nodo D, si definisce il *Laplaciano* la matrice $L = D - A$ dove l'elemento L_{ij} per $i < j$ sarà uguale a

$$L_{ij} = \begin{cases} d_i & \text{se} \quad i = j \\ -1 & \text{se fra} \quad (i,j) \quad \text{c'è un arco (relazione)} \\ 0 & \text{altrimenti} \end{cases} \tag{5.17}$$

La normalizzazione del Laplaciano consiste nel pre-moltiplicare e post-moltiplicare il Laplaciano per la matrice $D^{-1\backslash 2}$, quindi

$$L_{norm} = D^{-1\backslash 2} L D^{-1\backslash 2} \tag{5.18}$$

Tale normalizzazione viene anche definita simmetrica. Successivamente, bisogna calcolare gli autovalori λ e gli autovettori x della matrice L_{norm}; la decomposizione della Laplaciano normalizzato condurrà ad n autovettori e verranno considerati soltanto i primi m (in base al numero di gruppi che si sta provando ad individuare). Nella fase finale, ognuno degli n elementi sarà descritto da un vettore $m - dimensionale$ e verrà applicato un algoritmo di clustering non gerarchico classico, il *k-means* per trovare il miglior raggruppamento a m comunità degli n pazienti. In altri termini, l'inizializzazione dello spectral clustering prenderà forma grazie all'unica evidenza empirica in possesso: la matrice di adiacenza. Tuttavia, potrebbe anche essere utilizzata una matrice di similarità.

Scelta del numero di gruppi

L'obiettivo principale del VBEM è quello di stimare la distribuzione a posteriori delle variabili che sono considerate *latenti* e su cui si vogliono dare delle stime (sia puntuali che intervallari). Fra le informazioni non conosciute figurerebbe anche il numero di gruppi m e risulterebbe opportuno tenere in considerazione questa mancanza di informazione nello stesso processo di ottimizzazione della distribuzione a posteriori. Pertanto, il calcolo sarebbe impraticabile in quanto significherebbe che per ciascun valore di m bisognerebbe considerare tutti i possibili

modelli sulla base dei supporti delle variabili casuali considerate, ovvero

$$\ln p(A|m) = \ln \left\{ \sum_Z \int \int p(A, Z, \alpha, \theta|m) d\alpha d\theta \right\} \tag{5.19}$$

Per riuscire a discriminare qual è il raggruppamento migliore, viene proposto da Latouche et al. (2012) [15] l'uso dell'approssimazione della log-verosimiglianza marginale, piuttosto che $\ln p(A|m)$. Dunque, per diversi valori di $m = \{2, 3, \dots\}$ viene avviato più volte l'algoritmo di ottimizzazione e valutata la distribuzione dell'$ELBO$. Dunque, ci si affiderebbe a questa approssimazione per valutare quale risulterebbe essere il miglior raggruppamento.

Nel caso in analisi, però, si è già interessati ad un particolare numero di gruppi, quindi, si proseguirà con l'applicazione dell'approccio VBEM per l'identificazione di *quattro* comunità.

5.2.1 Modello di Bernoulli

Riprendendo le notazioni nel Paragrafo 5.0.2, data la matrice di adiacenza A e stabilito il numero di m gruppi, si definiscono:

- $A_{ij}|Z_i = k, Z_j = l \sim Ber(\eta_{kl}) \quad \forall k \leq l$, $(k, l) = 1, \dots, m$ dove η_{kl} definisce la probabilità che vi sia una relazione (arco) fra un elemento del gruppo k e un elemento del gruppo l. Dunque, si assume l'identica distribuzione e l'indipendenza per tutte le coppie (i, j) dove $i \in k$ e $j \in l$;
- $Z_i \sim \mathcal{M}(1; \alpha = (\alpha_1, \dots, \alpha_m)))$, $\forall i = 1, \dots, n$, dove n è il numero di pazienti; il vettore di parametri α descrive la probabilità di appartenere a ciascuno dei m gruppi;
- $\alpha \sim Dir(n^0)$, distribuzione a priori di Dirichlet, parametrizzata dal vettore $n^0 = (n_1^0, \dots, n_m^0)$;
- $\eta_{kl} \sim Beta(\delta_{kl}^0, \xi_{kl}^0)$ per $k \leq l$, $(k, l) = 1, \dots, m$; la probabilità di relazione fra gruppo k e gruppo l sono descritte da una distribuzione Beta con parametri a priori δ_{kl}^0 e ξ_{kl}^0.

Dunque, il vettore di variabili latenti sarà (Z, α, η). Il modello a cui si farebbe riferimento (qualora si conoscessero gli m raggruppamenti) sarebbe:

$$\begin{aligned} p(A, Z, \alpha, \eta) = p(A|Z, \eta)p(Z, \alpha) = \prod_{i<j} p(A_{ij} = a_{ij}|Z_i = k, Z_j = l, \eta_{kl}) \prod_{i=1}^{n} p(Z_i, \alpha) = \\ = \prod_{i<j} \prod_{k,l} \left[\eta_{kl}^{a_{ij}} (1 - \eta_{kl})^{1-a_{ij}} \right]^{Z_{ik} Z_{jl}} \prod_{i=1}^{n} \prod_{k=1}^{m} \alpha_k^{Z_{ik}}. \end{aligned} \tag{5.20}$$

La distribuzione a posteriori che si intende approssimare avrà la seguente forma a partire dalla distribuzione congiunta (a meno di costanti di normalizzazione: si noti il segno di proporzionalità)

$$\begin{aligned} p(A,Z,\alpha,\eta) \propto p(Z,\alpha,\eta|A) = p(A|Z,\eta)p(Z|\alpha)p(\alpha)p(\eta) = \\ = \prod_{i<j} p(A_{ij}=a_{ij}|Z_i=k,Z_j=l,\eta_{kl}) \prod_{i=1}^{n} p(Z_i|\alpha)p(\alpha) \prod_{1\leq k\leq l\leq m} p(\eta_{kl}) \propto \\ \propto \prod_{i<j}\prod_{k,l}\left[\eta_{kl}^{a_{ij}}(1-\eta_{kl})^{1-a_{ij}}\right]^{Z_{ik}Z_{jl}} \prod_{i=1}^{n}\prod_{k=1}^{m}\alpha_k^{Z_{ik}} \prod_{k=1}^{m}\alpha_k^{n_k^0-1} \prod_{1\leq k\leq l\leq m}\eta_{kl}^{\delta_{kl}^0-1}(1-\eta_{kl})^{\xi_{kl}^0-1} \end{aligned} \tag{5.21}$$

Le variabili latenti non permettono di stimare mediante un approccio Bayesiano le distribuzioni a posteriori di ciascuna delle variabili di interesse ma ricorrendo alla tecnica di VBEM si definisce l'**Algoritmo VBEM: modello di Bernoulli** in Appendice A.8 (Input, output ed inizializzazione), A.9 (algoritmo). Inoltre, per l'implementazione in R è stata utilizzata la funzione già presente nel pacchetto *'mixer'* [15]. Per quanto concerne i calcoli relativi alle probabilità φ_{ik}^{new}, alle matrici δ e ξ dei parametri per η, al vettore n di parametri per α si consultino le Appendici A.3, A.4 e A.5.

5.2.2 Modello di Poisson

Per quanto riguarda il *modello di Poisson*, si faccia sempre riferimento alle notazioni nel Paragrafo 5.0.2 e considerati m gruppi di pazienti e la matrice di adiacenza A, si definiscono:

- $A_{ij}|Z_i=k,Z_j=l \sim Poisson(\omega_{kl}) \quad k\leq l,(k,l)=1,\ldots,m$ dove ω_{kl} definisce il numero medio di connessioni (archi, collegamenti) fra un elemento del gruppo k e un elemento del gruppo l. Pertanto viene assunta l'identica disitribuzione e l'indipendenza per tutte le coppie (i,j) dove $i\in k$ e $j\in l$;

- $\omega_{kl} \sim Gamma(\lambda_{kl}^0,\tau_{kl}^0)$ per $k\leq l$, $(k,l)=1,\ldots,m$; il numero medio di relazioni fra gruppo k e gruppo l è descritto da una distribuzione Gamma con parametri a priori λ_{kl}^0 e τ_{kl}^0.

Rispetto alle variabili casuali Z_i (per $i=1,\ldots,N$) e α le assunzioni sulle distribuzioni rimangono uguali a quelle esplicitate per il modello di Bernoulli nel Paragrafo 5.1.1.

Il vettore di variabili latenti sarà (Z,α,ω). Se si conoscesse il raggruppamento degli N

pazienti negli m gruppi, il modello sarebbe il seguente:

$$p(A,Z,\alpha,\omega) = p(A|Z,\omega)p(Z,\alpha) = \prod_{i<j} p(A_{ij}=a_{ij}|Z_i=k,Z_j=l,\omega_{kl}) \prod_{i=1}^{n} p(Z_i,\alpha) = \\ = \prod_{i<j}\prod_{k,l}\left[\frac{\omega_{kl}^{a_{ij}}}{a_{ij}!}e^{-\omega_{kl}}\right]^{Z_{ik}Z_{jl}} \prod_{i=1}^{n}\prod_{k=1}^{m}\alpha_k^{Z_{ik}}. \tag{5.22}$$

La distribuzione a posteriori $p(Z,\alpha,\omega|A)$ a partire dalla distribuzione congiunta (a meno di costanti di normalizzazione: si noti il segno di proporzionalità) sarà la seguente:

$$p(A,Z,\alpha,\omega) \propto p(Z,\alpha,\omega|A) = p(A|Z,\omega)p(Z|\alpha)p(\alpha)p(\omega) = \\ = \prod_{i<j} p(A_{ij}=a_{ij}|Z_i=k,Z_j=l,\omega_{kl}) \prod_{i=1}^{n} p(Z_i|\alpha)p(\alpha) \prod_{1\le k\le l\le m} p(\omega_{kl}) \propto \\ \propto \prod_{i<j}\prod_{k,l}\left[\frac{\omega_{kl}^{a_{ij}}}{a_{ij}!}e^{-\omega_{kl}}\right]^{Z_{ik}Z_{jl}} \prod_{i=1}^{n}\prod_{k=1}^{m}\alpha_k^{Z_{ik}} \prod_{k=1}^{m}\alpha_k^{n_k^0-1} \prod_{1\le k\le l\le m}\frac{{\tau_{kl}^0}^{\lambda_{kl}^0}}{\Gamma(\lambda_{kl}^0)}\omega_{kl}^{\lambda_{kl}^0-1}e^{-\tau_{kl}^0\omega_{kl}} \tag{5.23}$$

Anche nel modello di Poisson, le variabili latenti non permettono di stimare con un approccio Bayesiano le distribuzioni a posteriori di ciascuna delle variabili di interesse ma ricorrendo alla tecnica di VBEM si definisce l'**Algoritmo VBEM: modello di Poisson** in Appendice A.10 (Input, output ed inizializzazione), A.11 (algoritmo).

È opportuno precisare che la matrice di adiacenza nel caso del modello di Poisson non è la stessa utilizzata nel caso del modello di Bernoulli ma consiste nella somma di matrici di adiacenza definite in $(0,1)$ per le quali si assume la loro indipendenza. Ad esempio, se fossero costruite delle matrici a livello di pathway, due o più pathways potrebbero avere uno o più geni in comune e quindi le matrici di adiacenza verrebbero stimate sulla base di matrici di distanza in cui avrà sicuramente influito un certo numero di geni *ridondanti*. In quanto ai passaggi algebrici relativi al calcolo delle probabilità φ_{ik}^{new}, alle matrici λ e τ dei parametri per ω, al vettore n di parametri per α si consultino le Appendici A.3, A.6 e A.7; invece, per consultare il codice in linguaggio R si faccia riferimento all'Appendice C.1.

I risultati dell'approccio VBEM

Per quanto riguarda l'applicazione dell'approccio VBEM con modello di Bernoulli è stato utilizzato il pacchetto R di nome *'mixer'* [15]; invece, per il modello di Poisson è stata implementata una funzione in R consultabile in Appendice C.1. Per ogni matrice di adiacenza è stato applicato l'algoritmo VBEM e trovata sia la classificazione finale che la matrice di relazione fra i gruppi. Le comunità ritrovate (vb) sono state confrontate con quelle vere (T) in una tabella a doppia entrata ed un'ulteriore tabella è stata costruita fra un metodo di classificazione classico (kmeans, km, basato sulla matrice dei dati su cui sono state calcolate le misure di dissimilarità) e la vera classificazione (T). Nelle Tabelle 5.1 e 5.2 vengono riportate tali informazioni rispetto alle quattro applicazioni del modello di Bernoulli e all'unica applicazione del modello di Poisson. Inoltre, sono stati calcolati due indici di entropia: l'entropia della classificazione proposta condizionata alla vera classificazione ($H(km|T)$ e $H(vb|T)$) e l'entropia *marginale* della classificazione basata sui due metodi proposti ($H(km)$ e $H(vb)$). L'indice marginale servirà principalmente a valutare se la distribuzione risultante dal kmeans o dal VBEM sia abbastanza omogenea o se invece si osservano delle eterogeneità particolari. Date due variabili discrete X (con supporto S_X) e (Y con supporto S_Y), si definiscono le Formule 5.24 e 5.25 per calcolare l'entropia marginale $H(Y)$ e l'entropia condizionata $H(Y|X = x)$ rispettivamente

$$H(Y) = -\sum_{y \in S_Y} p(y) \ln \big(p(y)\big) \tag{5.24}$$

dove $p(y) = Pr(Y = y)$.

$$H(Y|X = x) = -\sum_{x \in S_X} \sum_{y \in S_Y} p(y|X = x) \ln \big(p(y|X = x)\big) p(x) \tag{5.25}$$

dove $p(y|X = x) = Pr(Y = y|X = x)$.

VBEM (Bernoulli e Poisson)

GSVA (Bernoulli)

$km\backslash T$	Cc	End	Muc	Sier	$vb\backslash T$	Cc	End	Muc	Sier	η	1	2	3	4
1	2	4	2	9	1	2	3	2	8	1	0.55	0.07	0.17	0.16
2	10	1	4	0	2	4	6	5	11	2	-	0.52	0.15	0.13
3	4	7	4	12	3	0	8	6	4	3	-	-	0.47	0.004
4	0	7	7	3	4	10	2	4	1	4	-	-	-	0.39
$H(km\|T)$	0.474				$H(vb\|T)$	0.505								
$H(km)$	0.59				$H(vb)$	0.592								

PerPAS (Bernoulli)

$km\backslash T$	Cc	End	Muc	Sier	$vb\backslash T$	Cc	End	Muc	Sier	η	1	2	3	4
1	2	13	4	8	1	7	3	11	15	1	0.08	0.19	0.11	0.1
2	1	1	9	0	2	0	5	2	7	2	-	0.91	0.38	0.38
3	3	1	3	10	3	8	3	2	2	3	-	-	0.69	0.08
4	10	4	1	6	4	1	8	2	0	4	-	-	-	0.84
$H(km\|T)$	0.452				$H(vb\|T)$	0.44								
$H(km)$	0.581				$H(vb)$	0.55								

DEG(d_E, Bernoulli)

$km\backslash T$	Cc	End	Muc	Sier	$vb\backslash T$	Cc	End	Muc	Sier	η	1	2	3	4
1	14	0	1	1	1	0	5	4	23	1	0.59	0.11	0.01	0.05
2	0	4	2	22	2	2	14	4	0	2	-	0.47	0.03	0.02
3	0	0	9	0	3	0	0	8	0	3	-	-	0.83	0.02
4	2	15	5	1	4	14	0	1	1	4	-	-	-	0.88
$H(km\|T)$	0.246				$H(vb\|T)$	0.238								
$H(km)$	0.569				$H(vb)$	0.556								

DEG(d_{FES}, Bernoulli)

$km\backslash T$	Cc	End	Muc	Sier	$vb\backslash T$	Cc	End	Muc	Sier	η	1	2	3	4
1	0	0	9	0	1	0	1	2	20	1	0.72	0.08	0.02	0.003
2	14	0	1	1	2	2	16	5	2	2	-	0.32	0.05	0.003
3	0	4	2	22	3	1	2	9	1	3	-	-	0.75	0.07
4	2	15	5	1	4	13	0	1	1	4	-	-	-	0.99
$H(km\|T)$	0.246				$H(vb\|T)$	0.307								
$H(km)$	0.569				$H(vb)$	0.586								

Poisson(somma matrici di adiacenza GSVA, PerPAS e DEG(d_{FES}))

$km\backslash T$	Cc	End	Muc	Sier	$vb\backslash T$	Cc	End	Muc	Sier	ω	1	2	3	4
1	1	3	11	1	1	14	2	3	1	1	2.49	0.29	0.38	0.49
2	13	1	1	1	2	1	4	2	21	2	-	2.07	0.37	0.62
3	1	4	2	21	3	0	0	9	0	3	-	-	4.88	0.42
4	1	11	3	1	4	1	13	3	2	4	-	-	-	2.65
$H(km\|T)$	0.35				$H(vb\|T)$	0.311								
$H(km)$	0.587				$H(vb)$	0.573								

Tabella 5.1: Tabella riassuntiva dei risultati relativi all'applicazione dell'approccio VBEM alle matrici di adiacenza GSVA, PerPAS e DEG (euclidea e DISFES) con modello di Bernoulli e alla matrice integrata con modello di Poisson. Per ciascuna matrice si riportano: la tabella kmeans (km) vs. classificazione vera (T), la tabella VBEM vs. classificazione vera, la matrice di relazioni stimata fra i gruppi (o η o ω) attraverso VBEM e due indici: uno di entropia della classificazione VBEM o kmeans ($H(vb)$, $H(km)$) e uno di entropia delle due classificazioni condizionatamente ai veri gruppi ($H(vb|T)$, $H(km|T)$).

Le matrici che descrivono le relazioni fra i quattro gruppi (sia η che ω, escluso il caso del PerPAS) soddisfano una delle caratteristiche che ci si aspettava, secondo cui: le relazioni sti-

mate fra i gruppi sarebbero state poco probabili (nel caso delle probabilità di legame, η) o poco numerose (nel caso nel numero medio di relazioni, ω); invece, le relazioni stimate dentro i gruppi sarebbero state in media o maggiori di numero (nel caso del modello di Poisson) oppure più probabili (nel caso del modello di Bernoulli). Il confronto fra classificazione trovata e classificazione vera potrebbe essere effettuato costruendo la relativa tabella a doppia entrata. Pertanto, è possibile confrontare l'entropia condizionata calcolata sul kmeans con quella calcolata sul VBEM. Si osserva, infatti, che quest'ultima risulta (per ogni grafo) inferiore rispetto alla prima ad eccezione del GSVA e del DEG (d_{FES}) dove si evince un incremento piuttosto che un decremento. In altri termini, un decremento di entropia condizionata osservato passando da un metodo di classificazione ad un altro significa che condizionandosi ad ogni vero gruppo, la distribuzione nei quattro gruppi trovati con VBEM è meno omogenea rispetto alla distribuzione trovata con il metodo kmeans, e quindi ogni singola istologia risulterebbe più concentrata in un solo gruppo dei quattro identificati piuttosto che sparsa.

In particolare, tra i modelli di Bernoulli calcolati, il migliore risulta essere quello basato sui DEG(d_E) anche se già un kmeans offrirebbe una classificazione non lontana da quella trovata con VBEM.

Per quanto riguarda il modello di Poisson, si osserva un modesto decremento dell'entropia condizionata del VBEM rispetto al kmeans. Inoltre, è possibile avere una stima del numero medio di relazioni intra e fra i gruppi (ω), la quale evidenzia poche interazioni fra elementi di gruppi diversi e più di due interazioni fra elementi dello stesso gruppo.

Infine è da notare come la classificazione in GSVA risulta piuttosto sparsa e in PerPAS si osserva come il VBEM formi un gruppo con molti *mucinosi* e *sierosi*. Tale comportamento viene meno nelle restanti applicazioni dell'algoritmo VBEM.

I grafici relativi alle rappresentazioni delle reti possono essere consultati in Appendice B.5, B.6, B.7, B.8 e B.9.

5.3 Approccio Bayesiano: Gibbs sampling

L'obiettivo degli SBM può essere perseguito anche attraverso l'applicazione di un Gibbs sampling, il quale seguirà le assunzioni probabilistiche finora esplicitate sia sul modello e che sui parametri. La proposta di usare il campionamento di Gibbs è stata definita da Snijders e Nowicki prima nel 1997 [23] e successivamente sviluppata in modo più completo nel 2001 [24] dagli stessi autori.

L'intento generale rimane quello di fornire delle stime (puntuali o intervallari) dei parametri delle distribuzioni che descrivono le relazioni fra i gruppi e la composizione degli stessi. Pertanto, è necessario specificare un passo ulteriore per la predizione della classificazione delle unità statistiche negli m gruppi, equivalente alla fase di *Expectation* nel modello VBEM.

Un particolare dei modelli SBM in cui viene applicato il metodo del Gibbs sampling è la classificazione iterativa delle unità statistiche negli m gruppi dati i parametri generati nella iterazione corrente dell'algoritmo.

Burn-in

Snijders e Nowicki (2001) [24] specificano che il burn-in dovrebbe essere suddiviso in due parti: le prime $M0$ simulazioni del Gibbs sampling dovranno aggiornare i parametri ad ogni singola iterazione in modo particolare seguendo delle specifiche indicazioni per assicurare auspicabilmente la simulazione dalla maggior parte dello spazio parametrico; le seconde $M0$ simulazioni dovranno aggiornare i parametri come in un normale Gibbs sampling. Dopo le $2M0$ simulazioni di burn-in, gli autori spiegano che l'algoritmo avrà raggiunto la convergenza e che quindi le simulazioni successive permetteranno di dare delle stime sui parametri di interesse. Nelle simulazioni per i quattro grafi è stato impostato $M0 = 5,000$.

5.3.1 Modello di Bernoulli

Per quanto riguarda il modello di Bernoulli [24], le assunzioni rimangono le stesse scritte nel Paragrafo 5.1.1:

- $A_{ij}|Z_i = k, Z_j = l \sim Ber(\eta_{kl}) \quad \forall k \leq l,\ (k,l) = 1, \ldots, m$;
- $Z_i \sim \mathcal{M}(1; \alpha = (\alpha_1, \ldots, \alpha_m))), \forall i = 1, \ldots, n$;
- $\alpha \sim Dir(n^0)$;
- $\eta_{kl} \sim Beta(\delta^0_{kl}, \xi^0_{kl})$ per $k \leq l,\ (k,l) = 1, \ldots, m$.

Gli aggiornamenti dei parametri saranno i seguenti:

$$\alpha|A, Z, \eta \sim Dir\left(\sum_k \left[\left(\sum_{i=1}^{n} Z_{ik}\right) + n^0_k\right]\right) \tag{5.26}$$

$$\text{per } k < l \quad \eta_{kl}|A, Z, \alpha \sim Beta(\delta^0_{kl} + \sum_{i \neq j}^{n} Z_{ik} Z_{jl} a_{ij}, \xi^0_{kl} + \sum_{i \neq j}^{n} Z_{ik} Z_{jl}(1 - a_{ij})) \tag{5.27}$$

$$\text{per } k = l \quad \eta_{kk}|A, Z, \alpha \sim Beta(\delta^0_{kk} + \sum_{i<j}^{n} Z_{ik}Z_{jk}a_{ij}, \xi^0_{kk} + \sum_{i<j}^{n} Z_{ik}Z_{jk}(1 - a_{ij})) \tag{5.28}$$

I parametri delle distribuzioni a priori (non informative) saranno impostati nel seguente modo: $\{n^0\}_k = \{100\}_k$ e $\delta^0_{kl} = \xi^0_{kl} = 1 \quad \text{per } k \leq l$. Ad ogni passo dell'algoritmo verranno generati singoli valori dalle distribuzioni a posteriori e successivamente verrà eseguito un passo di riclassificazione dei pazienti negli m gruppi. Rispetto alle assunzioni probabilistiche relative al modello di Bernoulli, l'aggiornamento della classificazione del generico elemento i per $i = 1, \ldots, N$ sarà

$$\begin{aligned} k_i &= \arg\max_k \left\{ Pr(Z_i = k|A, \alpha, \eta, \{Z\}_{-i}) \right\} = \\ &= \arg\max_k \left\{ \mathcal{C}\alpha_k \prod_{l=1}^{m} \prod_{j \neq i}^{n} \eta_{kl}^{d(i,l)} (1 - \eta_{kl})^{1 - d(i,l)} \right\} \end{aligned} \tag{5.29}$$

dove $\mathcal{C}$ è una costante che non dipende da l e $d(i,l) = I(a_{ij} = 1)I(Z_j = l)$. Quindi la Formula 5.27 potrà essere scritta come

$$\begin{aligned} k_i &\propto \arg\max_k \left\{ \alpha_k \prod_{l=1}^{m} \eta_{kl}^{\sum_{j \neq i}^{n} d(i,l)} (1 - \eta_{kl})^{\sum_{j \neq i}^{n} (1 - d(i,l))} \right\} = \\ &= \arg\max_k \left\{ \alpha_k \prod_{l=1}^{m} \eta_{kl}^{\sum_{j \neq i}^{n} a_{ij} Z_{jl}} (1 - \eta_{kl})^{\sum_{j \neq i}^{n} (1 - a_{ij}) Z_{jl}} \right\} \end{aligned} \tag{5.30}$$

Nelle prime 5,000 simulazioni del burn-in si osserveranno degli specifici aggiornamenti dei parametri:

- per i parametri di $\alpha|A, Z, \eta$ viene creata una sequenza decrescente di lunghezza 5,000 che parte da $10n$ (nel caso studio $10 \times 76 = 760$) e termina a 100 (parametro a priori per $k = 1, \ldots, 4$); ad ogni passo si somma il parametro aggiornato n_k al valore corrispondente della sequenza
- per i parametri di $\eta_{kl}|A, Z, \alpha$ per $k \leq l$ viene creata una sequenza di lunghezza 5,000 e crescente da $1/n$ (nel caso studio 1/76) a 1; ad ognuna delle 5,000 iterazioni si moltiplicano entrambi i parametri aggiornati δ e ξ per il corrispondente valore della sequenza.

Il codice dell'algoritmo in R è consultabile in Appendice C.2

5.3.2 Modello di Poisson

Anche per il modello di Poisson, le assunzioni probabilistiche rimangono quelle esplicitate nel Paragrafo 5.1.2:

- $A_{ij}|Z_i = k, Z_j = l \sim Poisson(\omega_{kl}) \quad k \leq l, (k,l) = 1, \ldots, m;$
- $\omega_{kl} \sim Gamma(\lambda^0_{kl}, \tau^0_{kl})$ per $k \leq l$, $(k,l) = 1, \ldots, m$.

Le assunzioni su α e Z sono le stesse del modello di Bernoulli (si veda Paragrafo 5.2.1).

Gli aggiornamenti dei parametri saranno:

$$\alpha|A, Z, \omega \sim Dir\left(\sum_k \left[\left(\sum_{i=1}^{n} Z_{ik}\right) + n^0_k\right]\right) \tag{5.31}$$

$$\text{per } k < l \quad \omega_{kl}|A, Z, \alpha \sim Gamma(\lambda^0_{kl} + \sum_{i \neq j}^{n} Z_{ik} Z_{jl} a_{ij}, \tau^0_{kl} + \sum_{i \neq j}^{n} Z_{ik} Z_{jl}) \tag{5.32}$$

$$\text{per } k = l \quad \omega_{kk}|A, Z, \alpha \sim Gamma(\lambda^0_{kk} + \sum_{i<j}^{n} Z_{ik} Z_{jk} a_{ij}, \tau^0_{kk} + \sum_{i<j}^{n} Z_{ik} Z_{jk}) \tag{5.33}$$

I parametri delle distribuzioni a priori (non informative) saranno impostati nel seguente modo: $\{n^0\}_k = \{100\}_k$ e $\lambda^0_{kl} = \tau^0_{kl} = 1 \quad$ per $k \leq l$. Ad ogni passo dell'algoritmo verranno prima generati singoli valori dalle distribuzioni a posteriori e dopo verrà eseguito un passo di ri-classificazione dei pazienti negli m gruppi. Rispetto alle assunzioni probabilistiche relative al modello di Poisson, l'aggiornamento della classificazione del generico elemento i per $i = 1, \ldots, N$ sarà

$$\begin{aligned} k_i &= \arg\max_k \left\{ Pr(Z_i = k|A, \alpha, \omega, \{Z\}_{-i}) \right\} = \\ &= \arg\max_k \left\{ \mathcal{C} \alpha_k \prod_{l=1}^{m} \prod_{j \neq i}^{n} \left[\frac{\omega_{kl}^{a_{ij}}}{a_{ij}!} e^{-\omega_{kl}} \right]^{Z_{jl}} \right\} \end{aligned} \tag{5.34}$$

dove $\mathcal{C}$ e $a_{ij}!$ sono costanti che non dipendono da l. La Formula 5.32 potrà essere scritta come

$$k_i \propto \arg\max_k \left\{ \alpha_k \prod_{l=1}^{m} \omega_{kl}^{\sum_{j \neq i}^{n} a_{ij} Z_{jl}} e^{-\omega_{kl} \sum_{j \neq i}^{n} Z_{jl}} \right\} \tag{5.35}$$

Nelle prime 5,000 simulazioni del burn-in si osserveranno degli specifici aggiornamenti dei parametri:

- per i parametri di $\alpha|A, Z, \omega$ viene creata una sequenza decrescente di lunghezza 5,000 che parte da $10n$ (nel caso studio $10 \times 76 = 760$) e termina a 100 (parametro a priori per $k = 1, \ldots, 4$); ad ogni passo si somma il parametro aggiornato n_k al valore corrispondente della sequenza
- per i parametri di $\omega_{kl}|A, Z, \alpha$ per $k \leq l$ viene creata una sequenza di lunghezza 5,000 e crescente da $1/n$ (nel caso studio 1/76) a 1; ad ognuna delle 5,000 iterazioni si moltiplicano entrambi i parametri aggiornati λ e τ per il corrispondente valore della sequenza.

Il codice dell'algoritmo in R è consultabile in Appendice C.3

I risultati

Per il caso studio in esame (sia per Bernoulli che per Poisson) sono stati simulati 15,000 valori: 5,000 con aggiornamenti dei parametri secondo il metodo specificato dagli autori, 5,000 con aggiornamenti classici, e un numero di 5,000 simulazioni delle quali 3,000 sono state utilizzate per le analisi successive. In particolare, il modello di Bernoulli è stato applicato ad ogni matrice di adiacenza definita nel Paragrafo 4.2; invece, il modello di Poisson è stato applicato alla somma delle matrici di adiacenza GSVA, PerPAS e DEG (d_{FES}). Per valutare la convergenza (solo nel caso del modello di Bernoulli) sono state simulate 6 catene di lunghezza 3,000 per ogni matrice e valutato l'andamento dell'indice I_y [24] e della sua media al susseguirsi delle iterazioni (ci si aspetta che essa abbia un andamento costante).

$$I_y = -\frac{2}{\sharp(\mathcal{N})} \sum_{(i,j)\in\mathcal{N}, i<j} \log\left(\eta_{y_{ij}}(X_i, X_j)\right) \tag{5.36}$$

dove $\mathcal{N}$ definisce l'insieme delle coppie per le quali si osserva una relazione (1 nella matrice di adiacenza), con $\sharp$ si indica la sua cardinalità. Si assume che $\mathcal{N}$ sia simmetrico nel senso che sia (i, j) che (j, i) appartengono ad $\mathcal{N}$.

Modello di Bernoulli

L'applicazione del Gibbs sampling con modello di Bernoulli ha condotto ai risultati presenti nelle Tabelle 5.3 e 5.4. In essa vengono riportate le tabelle a due vie fra la classificazione ottenuta con il Gibbs sampling e quella vera, la matrice di probabilità di relazione η (stimata come la media a posteriori di ciascuna distribuzione), le probabilità di appartenenza ai gruppi α (stimate con le medie a posteriori), l'entropia marginale della classificazione con Gibbs sampling e quella condizionata alla vera classificazione.

Gibbs sampling (Bernoulli)

GSVA

$Gibbs \backslash T$	Cc	End	Muc	Sier	η	1	2	3	4		α
1	2	2	5	3	1	0.64	0.15	0.29	0.26	1	0.2499103
2	3	4	2	7	2	-	0.28	0.05	0.16	2	0.2498592
3	8	3	4	7	3	-	-	0.34	0.01	3	0.2501084
4	3	10	6	7	4	-	-	-	0.50	4	0.2501221
$H(Gibbs\|T)$	0.55										
$H(Gibbs)$	0.58										

PerPAS

$Gibbs \backslash T$	Cc	End	Muc	Sier	η	1	2	3	4		α
1	1	11	3	6	1	0.49	0.02	0.46	0.07	1	0.2503032
2	10	3	1	0	2	-	0.70	0.49	0.09	2	0.2499716
3	0	3	2	8	3	-	-	0.67	0.14	3	0.2499618
4	5	2	11	10	4	-	-	-	0.09	4	0.2497635
$H(Gibbs\|T)$	0.44										
$H(Gibbs)$	0.58										

Tabella 5.2: Tabella riassuntiva dei risultati relativi all'applicazione dell'approccio Gibbs Samlpling alle matrici di adiacenza GSVA, PerPAS con modello di Bernoulli. Per ciascuna matrice si riportano: la tabella Gibbs vs. classificazione vera, la matrice di relazioni stimata fra i gruppi (η) e due indici: uno di entropia marginale del Gibbs ($H(Gibbs)$) e uno di entropia della classificazione Gibbs condizionatamente ai veri gruppi ($H(Gibbs|T)$).

Gibbs sampling (Bernoulli)

DEG (d_E)

$Gibbs\backslash T$	Cc	End	Muc	Sier	η	1	2	3	4		α
1	14	0	1	1	1	0.88	0.05	0.03	0.02	1	0.2503279
2	0	5	3	23	2	-	0.59	0.13	0.01	2	0.2497792
3	2	12	3	0	3	-	-	0.63	0.05	3	0.2498054
4	0	2	10	0	4	-	-	-	0.37	4	0.2500875
$H(Gibbs\|T)$	0.26										
$H(Gibbs)$	0.57										

DEG (d_{FES})

$Gibbs\backslash T$	Cc	End	Muc	Sier	η	1	2	3	4		α
1	13	0	1	1	1	0.99	0.003	0.003	0.04	1	0.2500841
2	2	15	6	2	2	-	0.30	0.07	0.05	2	0.2498643
3	0	1	1	20	3	-	-	0.72	0.02	3	0.2499442
4	1	3	9	1	4	-	-	-	0.73	4	0.2501074
$H(Gibbs\|T)$	0.31										
$H(Gibbs)$	0.59										

Tabella 5.3: Tabella riassuntiva dei risultati relativi all'applicazione dell'approccio Gibbs Samlpling alle matrici di adiacenza DEG (euclidea e DISFES) con modello di Bernoulli. Per ciascuna matrice si riportano: la tabella Gibbs vs. classificazione vera, la matrice di relazioni stimata fra i gruppi (η) e due indici: uno di entropia marginale del Gibbs ($H(Gibbs)$) e uno di entropia della classificazione Gibbs condizionatamente ai veri gruppi ($H(Gibbs|T)$).

Innanzi tutto si osserva come le probabilità di appartenenza ai gruppi siano approssimativamente pari a 0.25 e che quindi, almeno per quanto riguarda i dati osservati, non si evince una distribuzione particolare delle quattro istologie. In quanto ai parametri (η) che definiscono le probabilità di relazione intra e fra i gruppi, si osserva che per GSVA e PerPAS alcune probabilità di relazione fra gruppi non risultano abbastanza basse; in particolare nel caso del PerPAS risultano persino più alte delle probabilità di relazione intra i gruppi (si veda: 0.49 la probabilità che vi sia una interazione fra un elemento del gruppo 2 e uno del gruppo 3, è maggiore di 0.09 , che è la probabilità di interazione fra elementi appartenenti al gruppo 4). Ciò fa notare che il raggruppamento ottenuto, almeno alla luce dell'unica evidenza empirica (matrice di adiacenza PerPAS), non risulta in accordo con quanto ci si aspetta, ovvero alte probabilità di relazione fra elementi dello stesso gruppo e basse probabilità di relazione fra elementi appartenenti a gruppi diversi. Nel caso del GSVA l'entropia condizionata relativa al risultato del Gibbs sampling risulta persino più alta di quella calcolata sui risultati del kmeans (vedi Tabella 5.1).

I risultati ottenuti con i DEG (sia d_E che d_{FES}) sono i migliori. In particolare, una classificazione soddisfacente si osserva con DEG d_E (matrice di adiacenza stimata sulla base della distanza euclidea), in quanto i gruppi risultano molto coesi (si vedano le elevate probabilità di relazione dentro i gruppi), e l'entropia condizionata risultante è abbastanza bassa.

I grafici in Figura 5.1 mostrano l'andamento per ciascun caso (GSVA, PerPAS, DEG (d_E e d_{FES}) l'andamento di I_y e della sua media al susseguirsi delle iterazioni delle 6 catene simulate ($6 \times 3,000 = 18,000$ iterazioni concatenate). Si osserva una convergenza soddisfacente per

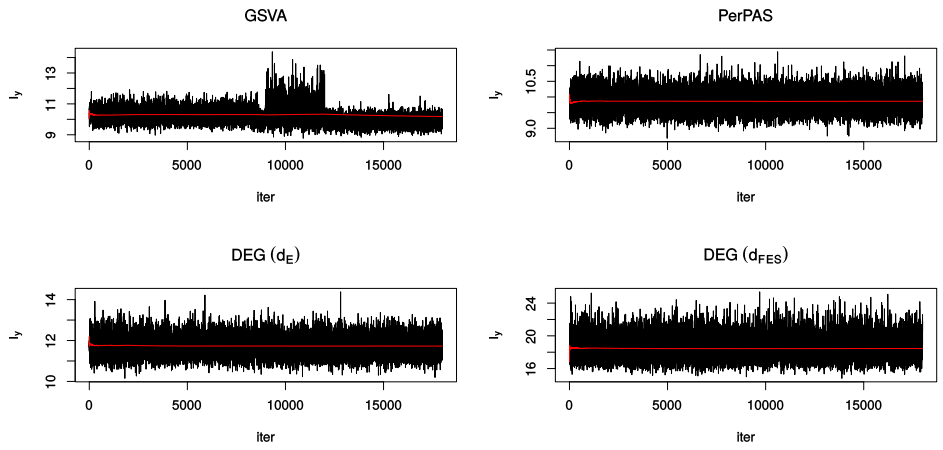


Figura 5.1: Andamento I_y e media (in rosso) per ciascun modello (GSVA, PerPAS, DEG (d_E e d_{FES}).

PerPAS e DEG (sia d_E che d_{FES}). Invece, per GSVA si nota che una delle 6 catene ha una tendenza ad assumere valori più grandi ed in quel tratto risulta anche più variabile rispetto alle restanti.

Per quanto riguarda i grafici relativi alle serie di valori simulati (η e α, sia grafici del loro andamento, della loro media che istogrammi delle distribuzioni) e alla rete risultante, si consulti l'Appendice da B.10 a B.29.

Modello di Poisson

Per quanto concerne il modello di Poisson (applicato sull'integrazione di tre matrici di adiacenza: GSVA, PerPAS e DEG (d_{FES})), sono stati simulati 5,000 valori oltre ai 10,000 di burn-in (stessa procedura utilizzata per il modello di Bernoulli) e di seguito vengono riportate in Tabella 5.5 le medesime informazioni già osservate per il modello di Bernoulli nelle Tabelle 5.3 e 5.4.

Gibbs sampling (Poisson)

					DEG (d_{FES})					
$Gibbs \backslash T$	Cc	End	Muc	Sier	ω	1	2	3	4	α
1	1	10	11	1	1	1.05	0.66	0.21	0.17	0.25012
2	0	6	4	3	2	0.66	1.35	0.50	0.71	0.25008
3	14	0	1	1	3	0.21	0.50	1.86	0.24	0.25000
4	1	3	1	19	4	0.17	0.71	0.24	1.08	0.24980
$H(Gibbs\|T)$	0.34									
$H(Gibbs)$	0.59									

Tabella 5.4: Tabella riassuntiva dei risultati relativi all'applicazione dell'approccio Gibbs Samlpling con modello di Poisson. Si riportano: la tabella Gibbs vs. classificazione vera, la matrice di relazioni stimata fra i gruppi (ω) e due indici: uno di entropia marginale del Gibbs ($H(Gibbs)$) e uno di entropia della classificazione Gibbs condizionatamente ai veri gruppi ($H(Gibbs|T)$).

Si osserva che il modello di Poisson fornisce una classificazione abbastanza soddisfacente a meno del gruppo 1 costituito da una maggioranza di *mucinosi* e *sierosi*. Il numero di relazioni medio (ω) stimato risulta essere maggiore di uno per coppie di nodi (pazienti) dello stesso gruppo e minore di uno per coppie di nodi (pazienti) appartenenti a gruppi differenti. Anche nel caso del modello di Poisson si osserva che la probabilità di appartenere ad un gruppo qualsiasi dei quattro è ≈ 0.25. I grafici corrispondenti alla catena simulata sono riportati in Appendice da B.30 a B.34.

In quanto alla convergenza non è stato valutato l'andamento dell'indice I_y in quanto nel caso specifico del modello di Poisson non si tratterebbe più del logaritmo di una probabilità ma del logaritmo del numero medio di relazioni. Pertanto, è stato scelto di soffermarsi alla sola generazione di una catena di simulazioni.

Capitolo 6

Conclusioni

Una volta concluse le analisi è opportuno esporre delle conclusioni circa i risultati ottenuti e definire dei possibili miglioramenti delle analisi stesse.

Nel corso dell'applicazione dei due nuovi metodi di pathway analysis, è risultato necessario implementare un bootstrap per entrambi i metodi (GSVA e PerPAS) ma è stato possibile farlo soltanto per il GSVA. Quindi, sarebbe stato interessante poter applicare un bootstrap anche per il PerPAS e valutare in modo più corretto la signifcatività paziente-specifica degli score ottenuti.

Tuttavia, entrambi i metodi sono risultati abbastanza soddisfacenti nella discriminazione dei gruppi istologici, un po' meno nella stima della matrice di adiacenza dove, l'applicazione del metodo di stima bayesiano empirico ha condotto a delle matrici di adiacenza per GSVA e PerPAS che non sono riuscite a porre in evidenza i veri raggruppamenti. Il contrario è stato osservato per la matrice di DEG (geni differenzialmente espressi), per cui la matrice di adiacenza risultante è riuscita a mettere in risalto i veri gruppi.

Le matrici di adiacenza stimate hanno costituito l'unica evidenza empirica alla base dei modelli definiti nel Capitolo 5 (VBEM e Gibbs sampling).

Sono stati proposti due approcci per la classificazione: un approccio di variational inference applicato ai modelli bayesiani, un approccio bayesiano con gibbs sampling specifico per i modelli a blocchi stocastici. Entrambi hanno avuto in comune l'obiettivo di ricavare informazioni circa la composizione dei gruppi e le relazioni intra e fra i gruppi. Le due specificazioni proposte sui modelli (Bernoulli e Poisson) hanno dimostrato di fornire dei risultati apprezzabili sulle classificazioni sia nel caso del variational inference che nel Gibbs sampling (in particolare per la matrice di adiacenza ricavata dai DEG). Tuttavia i due metodi sono molto diversi per quanto concerne il loro peso computazionale. In particolare, il Gibbs sampling offre delle stime delle

distribuzioni a posteriori migliori rispetto al variational bayes il quale però richiede tempi più brevi e permette così di indagare in modo veloce un maggior numero di specificazioni rispetto alle assunzioni sui modelli e/o sulle distribuzioni dei parametri.

In conclusione, dei possibili miglioramenti riguardo i modelli per le classificazioni potrebbero consistere in: un maggior numero di iterazioni del burn-in (nell'articolo di riferimento viene spiegato un esempio in cui vengono utilizzate $2M0 = 50,000$ simulazioni); una diversa specificazione dei modelli in cui figureranno anche dei potenziali predittori, ad esempio dei biomarcatori tumorali che si esprimono in modo significativamente diverso fra le istologie, oppure altri fattori siano essi endogeni o esogeni relativi all'istologia.

Appendice A

Appendice metodologica

$$\begin{aligned}\pi(K=k|r,\alpha,\beta) &= \int_0^1 L(k;p,r)\pi(p;\alpha,\beta)dp = \\ &= \int_0^1 \binom{r}{k} p^k(1-p)^{r-k}\frac{\Gamma(\alpha)\Gamma(\beta)}{\Gamma(\alpha+\beta)}p^{\alpha-1}(1-p)^{\beta-1}dp = \\ &= \binom{r}{k}\frac{\Gamma(\alpha)\Gamma(\beta)}{\Gamma(\alpha+\beta)}\int_0^1 p^{k+\alpha-1}(1-p)^{r-k+\beta-1}dp = \\ &= \binom{r}{k}\frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}\frac{\Gamma(k+\alpha)\Gamma(r-k+\beta)}{\Gamma(r+\alpha+\beta)}\end{aligned} \tag{A.1}$$

$$\begin{aligned}KL\big(q(Z)||p(Z|Y)\big) &= \mathbb{E}\big[\ln q(Z)\big] - \mathbb{E}\big[\ln p(Z|Y)\big] = \\ &= \mathbb{E}\big[\ln q(Z)\big] - \mathbb{E}\big[\ln p(Y,Z)\big] + \mathbb{E}\big[\ln p(Y)\big]\end{aligned} \tag{A.2}$$

$$\begin{aligned}\ln q(\alpha) &= \mathbb{E}_{Z,\eta}[\ln p(A,Z,\alpha,\eta)] + cost. = \\ &= \mathbb{E}_Z[\ln p(Z|\alpha)] + \ln p(\alpha) + cost. = \\ &= \sum_{i=1}^{N}\sum_{k=1}^{m}\varphi_{ik}\ln\alpha_k + \sum_{k=1}^{m}(n_k^0-1)\ln\alpha_k + cost. = \\ &= \sum_{k=1}^{m}\left(\sum_{i=1}^{N}\varphi_{ik} + n_k^0 - 1\right)\ln\alpha_k + cost.\end{aligned} \tag{A.3}$$

$$\ln q(Z_i) = \mathbb{E}_{Z\setminus i,\alpha,\eta}[\ln p(A,Z,\alpha,\eta)] + cost. =$$

$$= \mathbb{E}_{Z\setminus i,\eta}[\ln p(A|Z,\eta)] + \mathbb{E}_{Z\setminus i,\alpha}[\ln p(Z|\alpha)] + cost. =$$

$$= \mathbb{E}_{Z\setminus i,\eta}\left[\sum_{i<j}\sum_{k,l} Z_{ik}Z_{jl}\Big[a_{ij}\ln(\eta_{kl}) + (1-a_{ij})\ln(1-\eta_{kl})\Big]\right] +$$

$$+\mathbb{E}_{Z\setminus i,\alpha}\left[\sum_{i=1}^{N}\sum_{k=1}^{m} Z_{ik}\ln(\alpha_k)\right] + cost.$$

Fissato un generico i

$$\sum_{k=1}^{m} Z_{ik}\left[\mathbb{E}_{\alpha_k}[\ln(\alpha_k)] + \sum_{j\neq i}^{N}\sum_{l=1}^{m}\mathbb{E}_{Z\setminus i}[Z_{jl}]\big[a_{ij}\mathbb{E}_{\eta}[\ln(\eta_{kl})] + (1-a_{ij})\mathbb{E}_{\eta}[\ln(1-\eta_{kl})]\big]\right] + cost. =$$

$$= \sum_{k=1}^{m} Z_{ik}\left[\psi(n_k) - \psi\Big(\sum_{k=1}^{m} n_k\Big) + \sum_{j\neq i}^{N}\sum_{l=1}^{m}\varphi_{jl}\big[a_{ij}\big[\psi(\delta_{kl}) - \psi(\delta_{kl}+\xi_{kl})\big] +\right.$$

$$\left. +(1-a_{ij})\big[\psi(\xi_{kl}) - \psi(\delta_{kl}+\xi_{kl})\big]\big]\right] + cost. =$$

$$= \sum_{k=1}^{m} Z_{ik}\left[\psi(n_k) - \psi\Big(\sum_{k=1}^{m} n_k\Big) + \sum_{j\neq i}^{N}\sum_{l=1}^{m}\varphi_{jl}\big[a_{ij}\psi(\delta_{kl}) - a_{ij}\psi(\delta_{kl}+\xi_{kl}) +\right.$$

$$\left. +\psi(\xi_{kl}) - \psi(\delta_{kl}+\xi_{kl}) - a_{ij}\psi(\xi_{kl}) + a_{ij}\psi(\delta_{kl}+\xi_{kl})\big]\right] + cost. =$$

$$= \sum_{k=1}^{m} Z_{ik}\left[\psi(n_k) - \psi\Big(\sum_{k=1}^{m} n_k\Big) + \sum_{j\neq i}^{N}\sum_{l=1}^{m}\varphi_{jl}\big[a_{ij}[\psi(\delta_{kl}) - \psi(\xi_{kl})] +\right.$$

$$\left. +\psi(\xi_{kl}) - \psi(\delta_{kl}+\xi_{kl})\big]\right] + cost. =$$

per $\quad k \in \{1,\dots,m\}$

$$\varphi_{ik} \propto \exp\left\{\psi(n_k) - \psi\Big(\sum_{k=1}^{m} n_k\Big) + \sum_{j\neq i}^{N}\sum_{l=1}^{m}\varphi_{jl}\big[a_{ij}[\psi(\delta_{kl}) - \psi(\xi_{kl})] + \psi(\xi_{kl}) - \psi(\delta_{kl}+\xi_{kl})\big]\right\}$$

(A.4)

$$
\begin{aligned}
\ln q(\eta) = \mathbb{E}_{Z,\alpha}\big[\ln p(A,Z,\alpha,\eta)\big] + cost. = \mathbb{E}_Z[\ln p(A|Z,\eta)] + \ln p(\eta) + cost. = \\
= \mathbb{E}_Z\left[\sum_{i<j}\sum_{k,l} Z_{ik}Z_{jl}\Big[a_{ij}\ln(\eta_{kl}) + (1-a_{ij})\ln(1-\eta_{kl})\Big]\right] + \\
+ \sum_{1\le k\le l\le m}\Big((\delta^0_{kl}-1)\ln(\eta_{kl}) + (\xi^0_{kl}-1)\ln(1-\eta_{kl})\Big) + cost. = \\
= \sum_{i<j}\sum_{k,l}\varphi_{ik}\varphi_{jl}\Big[a_{ij}\ln(\eta_{kl}) + (1-a_{ij})\ln(1-\eta_{kl})\Big] + \\
+ \sum_{1\le k\le l\le m}\Big((\delta^0_{kl}-1)\ln(\eta_{kl}) + (\xi^0_{kl}-1)\ln(1-\eta_{kl})\Big) + cost. = \\
= \sum_{k<l}^{m}\sum_{i\neq j}^{N}\varphi_{ik}\varphi_{jl}\Big[a_{ij}\ln(\eta_{kl}) + (1-a_{ij})\ln(1-\eta_{kl})\Big] + \\
+ \sum_{k=1}^{m}\sum_{i<j}^{N}\varphi_{ik}\varphi_{jk}\Big[a_{ij}\ln(\eta_{kk}) + (1-a_{ij})\ln(1-\eta_{kk})\Big] + \\
+ \sum_{1\le k\le l\le m}\Big((\delta^0_{kl}-1)\ln(\eta_{kl}) + (\xi^0_{kl}-1)\ln(1-\eta_{kl})\Big) + cost. = \\
= \sum_{k<l}^{m}\sum_{i\neq j}^{N}\varphi_{ik}\varphi_{jl}\Big[a_{ij}\ln(\eta_{kl}) + (1-a_{ij})\ln(1-\eta_{kl})\Big] + \\
+ \sum_{k=1}^{m}\sum_{i<j}^{N}\varphi_{ik}\varphi_{jk}\Big[a_{ij}\ln(\eta_{kk}) + (1-a_{ij})\ln(1-\eta_{kk})\Big] + \\
+ \sum_{k<l}^{m}\Big((\delta^0_{kl}-1)\ln(\eta_{kl}) + (\xi^0_{kl}-1)\ln(1-\eta_{kl})\Big) + \\
+ \sum_{k=1}^{m}\Big((\delta^0_{kk}-1)\ln(\eta_{kk}) + (\xi^0_{kk}-1)\ln(1-\eta_{kk})\Big) + cost. = \\
= \sum_{k<l}^{m}\left[\Big(\sum_{i\neq j}^{N}\varphi_{ik}\varphi_{jl}a_{ij} + \delta^0_{kl} - 1\Big)\ln(\eta_{kl}) + \Big(\sum_{i\neq j}^{N}\varphi_{ik}\varphi_{jl}(1-a_{ij}) + \xi^0_{kl} - 1\Big)\ln(1-\eta_{kl})\right] + \\
+ \sum_{k=1}^{m}\left[\Big(\sum_{i<j}^{N}\varphi_{ik}\varphi_{jk}a_{ij} + \delta^0_{kk} - 1\Big)\ln(\eta_{kk}) + \Big(\sum_{i<j}^{N}\varphi_{ik}\varphi_{jk}(1-a_{ij}) + \xi^0_{kk} - 1\Big)\ln(1-\eta_{kk})\right] + cost.
\end{aligned}
$$

(A.5)

$$\ln q(Z_i) = \mathbb{E}_{Z\setminus i,\alpha,\omega}[\ln p(A,Z,\alpha,\omega)] + cost. =$$

$$= \mathbb{E}_{Z\setminus i,\omega}[\ln p(A|Z,\omega)] + \mathbb{E}_{Z\setminus i,\alpha}[\ln p(Z|\alpha)] + cost. =$$

$$= \mathbb{E}_{Z\setminus i,\omega}\left[\sum_{i<j}\sum_{k,l} Z_{ik}Z_{jl}\left[a_{ij}\ln(\omega_{kl}) - \omega_{kl}\right]\right] +$$

$$+\mathbb{E}_{Z\setminus i,\alpha}\left[\sum_{i=1}^{N}\sum_{k=1}^{m} Z_{ik}\ln(\alpha_k)\right] + cost.$$

Fissato un generico i

$$\sum_{k=1}^{m} Z_{ik}\left[\mathbb{E}_{\alpha_k}[\ln(\alpha_k)] + \sum_{j\neq i}^{N}\sum_{l=1}^{m}\mathbb{E}_{Z\setminus i}[Z_{jl}]\left[a_{ij}\mathbb{E}_{\omega}[\ln(\omega_{kl})] - \mathbb{E}_{\omega}[\omega_{kl}]\right]\right] + cost. =$$

$$= \sum_{k=1}^{m} Z_{ik}\left[\psi(n_k) - \psi\left(\sum_{k=1}^{m} n_k\right) + \sum_{j\neq i}^{N}\sum_{l=1}^{m}\varphi_{jl}\left[a_{ij}\left[\psi(\lambda_{kl}) - \ln(\tau_{kl})\right] - \frac{\lambda_{kl}}{\tau_{kl}}\right]\right] + cost.$$

per $k \in \{1,\ldots,m\}$

$$\varphi_{ik} \propto \exp\left\{\psi(n_k) - \psi\left(\sum_{k=1}^{m} n_k\right) + \sum_{j\neq i}^{N}\sum_{l=1}^{m}\varphi_{jl}\left[a_{ij}\left[\psi(\lambda_{kl}) - \ln(\tau_{kl})\right] - \frac{\lambda_{kl}}{\tau_{kl}}\right]\right\}$$

(A.6)

$$\ln q(\omega) = \mathbb{E}_{Z,\alpha}\big[\ln p(A,Z,\alpha,\omega)\big] + cost. = \mathbb{E}_Z[\ln p(A|Z,\omega)] + \ln p(\omega) + cost. =$$

$$= \mathbb{E}_Z\left[\sum_{i<j}\sum_{k,l} Z_{ik}Z_{jl}\Big[a_{ij}\ln(\omega_{kl}) - \omega_{kl}\Big]\right] +$$

$$+ \sum_{1\le k\le l\le m}\Big((\lambda^0_{kl}-1)\ln(\omega_{kl}) - \tau^0_{kl}\omega_{kl}\Big) + cost. =$$

$$= \sum_{i<j}\sum_{k,l}\varphi_{ik}\varphi_{jl}\Big[a_{ij}\ln(\omega_{kl}) - \omega_{kl}\Big] +$$

$$+ \sum_{1\le k\le l\le m}\Big((\lambda^0_{kl}-1)\ln(\omega_{kl}) - \tau^0_{kl}\omega_{kl}\Big) + cost. =$$

$$= \sum_{k<l}^{m}\sum_{i\neq j}^{N}\varphi_{ik}\varphi_{jl}\Big[a_{ij}\ln(\omega_{kl}) - \omega_{kl}\Big] + \sum_{k=1}^{m}\sum_{i<j}^{N}\varphi_{ik}\varphi_{jk}\Big[a_{ij}\ln(\omega_{kk}) - \omega_{kk}\Big] +$$

$$+ \sum_{1\le k\le l\le m}\Big((\lambda^0_{kl}-1)\ln(\omega_{kl}) - \tau^0_{kl}\omega_{kl}\Big) + cost. =$$

$$= \sum_{k<l}^{m}\sum_{i\neq j}^{N}\varphi_{ik}\varphi_{jl}\Big[a_{ij}\ln(\omega_{kl}) - \omega_{kl}\Big] + \sum_{k=1}^{m}\sum_{i<j}^{N}\varphi_{ik}\varphi_{jk}\Big[a_{ij}\ln(\omega_{kk}) - \omega_{kk}\Big] +$$

$$+ \sum_{k<l}^{m}\Big((\lambda^0_{kl}-1)\ln(\omega_{kl}) - \tau^0_{kl}\omega_{kl}\Big) + \sum_{k=1}^{m}\Big((\lambda^0_{kk}-1)\ln(\omega_{kk}) - \tau^0_{kk}\omega_{kk}\Big) + cost. =$$

$$= \sum_{k<l}^{m}\left[\Big(\sum_{i\neq j}^{N}\varphi_{ik}\varphi_{jl}a_{ij} + \lambda^0_{kl} - 1\Big)\ln(\omega_{kl}) + \Big(\sum_{i\neq j}^{N}\varphi_{ik}\varphi_{jl} + \tau^0_{kl}\Big)\omega_{kl}\right] +$$

$$+ \sum_{k=1}^{m}\left[\Big(\sum_{i<j}^{N}\varphi_{ik}\varphi_{jk}a_{ij} + \lambda^0_{kk} - 1\Big)\ln(\omega_{kk}) + \Big(\sum_{i<j}^{N}\varphi_{ik}\varphi_{jk} + \tau^0_{kk}\Big)\omega_{kk}\right] + cost.$$

(A.7)

Algoritmo VBEM: modello di Bernoulli (Input, output ed inizializzazione)

Input:	
	grafo $G = \{N, A\}$: matrice di adiacenza A, il cui elemento a_{ij} indica se fra il nodo i e il nodo j è presente (1) o assente (0) una relazione; numero di nodi pari a N numero di gruppi m vettore di inizializzazione raggruppamenti $X = (X_1, \ldots, X_N)$, dove $X_i = k$ con $k \in \{1, \ldots, m\}$ ϵ_0 : soglia per la convergenza delle probabilità di classificazione negli m gruppi ϵ_1 : soglia per la convergenza dell'algoritmo di ottimizzazione
Output:	
	φ (probabilità di classificazione negli m gruppi per ogni paziente) n (composizione degli m gruppi) δ e ξ (matrici $m \times m$ i cui elementi δ_{kl} e ξ_{kl} sono direttamente collegati agli elementi η_{kl} della matrice $\{\eta\}_{m \times m}$ che definisce le probabilità di relazione intra e fra gli m gruppi)
Inizializzazione:	
	$\varphi_i^0 = [1/m]_m$ vettore $m - dimensionale$ $i = 1, .., N$ $n^0 = [1/2]_m$ vettore $m - dimensionale$ $\delta_{kl}^0 = \sum_{(i,j):i<j} a_{ij} I(i \in k) I(j \in l)$ per $k \leq l$ $\xi_{kl}^0 = n_k n_l - \delta_{kl}^0$ per $k \leq l$, dove $n_k = \sum_{i=1}^{N} I(i \in k)$ per $k = 1, \ldots, m$

Tabella A.8: Algoritmo VBEM: modello di Bernoulli (Input, output ed inizializzazione).

Algoritmo VBEM: modello di Bernoulli

set $ELBO^{old} = 10^4$
$\varphi^{old} = \varphi^0$
$\delta_{kl} = \delta^0_{kl}, \quad k \leq l$
$\xi_{kl} = \xi^0_{kl}, \quad k \leq l$
$n_k = n^0_k, \quad k = 1, \ldots, m$
while $|ELBO^{new} - ELBO^{old}| > \epsilon_1$ **do**

while $|\varphi^{new} - \varphi^{old}| > \epsilon_0$ **do**
for $i \in \{1, \ldots, N\}$
for $k \in \{1, \ldots, m\}$
set $\varphi^{new}_{ik} \propto \exp\left\{\psi(n_k) - \psi\left(\sum_k^m n_k\right) + \right.$
$\left. + \sum_{j \neq i}^N \sum_{l=1}^m \varphi^{old}_{jl}\big[a_{ij}[\psi(\delta_{kl}) - \psi(\xi_{kl})] + \psi(\xi_{kl}) - \psi(\delta_{kl} + \xi_{kl})\big]\right\}$
compute $|\varphi^{new} - \varphi^{old}|$
set $\varphi^{new} = \varphi^{old}$
end

set $\varphi = \varphi^{new}$

for $k \in \{1, \ldots, m\}$
$n_k = n^0_k + \sum_{i=1}^N \varphi_{ik}$

for $k = l$
$\delta_{kk} = \delta^0_{kk} + \sum_{i<j}^N a_{ij}\varphi_{ik}\varphi_{jk}$
$\xi_{kk} = \xi^0_{kk} + \sum_{i<j}^N (1 - a_{ij})\varphi_{ik}\varphi_{jk}$

for $k < l$
$\delta_{kl} = \delta^0_{kl} + \sum_{i \neq j}^N a_{ij}\varphi_{ik}\varphi_{jl}$
$\xi_{kl} = \xi^0_{kl} + \sum_{i \neq j}^N (1 - a_{ij})\varphi_{ik}\varphi_{jl}$

compute $ELBO^{new} = \ln\left\{\frac{\Gamma\left(\sum_{k=1}^m n^0_k\right)\prod_{k=1}^m \Gamma\left(n_k\right)}{\Gamma\left(\sum_{k=1}^m n_k\right)\prod_{k=1}^m \Gamma\left(n^0_k\right)}\right\} +$
$+ \sum_{1 \leq k \leq l \leq m} \ln\left\{\frac{\Gamma\left(\delta^0_{kl}+\xi^0_{kl}\right)\Gamma\left(\delta_{kl}\right)\Gamma\left(\xi_{kl}\right)}{\Gamma\left(\delta_{kl}+\xi_{kl}\right)\Gamma\left(\delta^0_{kl}\right)\Gamma\left(\xi^0_{kl}\right)}\right\} -$
$- \sum_{i=1}^N \sum_{k=1}^m \varphi_{ik} \ln(\varphi_{ik})$

compute $|ELBO^{new} - ELBO^{old}|$
set $ELBO^{new} = ELBO^{old}$
end
return φ, n, δ, ξ

Tabella A.9: Algoritmo VBEM: modello di Bernoulli.

Algoritmo VBEM: modello di Poisson (Input, output ed inizializzazione)

Input:	grafo $G = \{N, A\}$: matrice di adiacenza A (somma di matrici di adiacenza dove i singoli elementi sono definiti in $(0, 1)$)), il cui elemento a_{ij} indica il numero di relazioni (archi) presenti fra il nodo i e il nodo j; numero di nodi pari a N numero di gruppi m vettore di inizializzazione raggruppamenti $X = (X_1, \ldots, X_N)$, dove $X_i = k$ con $k \in \{1, \ldots, m\}$ ϵ_0 : soglia per la convergenza delle probabilità di classificazione negli m gruppi ϵ_1 : soglia per la convergenza dell'algoritmo di ottimizzazione
Output:	φ (probabilità di classificazione negli m gruppi per ogni paziente) n (composizione degli m gruppi) λ e τ (matrici $m \times m$ i cui elementi λ_{kl} e τ_{kl} sono direttamente collegati agli elementi ω_{kl} della matrice $\{\omega\}_{m \times m}$ che definisce il numero medio di relazioni intra e fra gli m gruppi)
Inizializzazione:	$\varphi_i^0 = [1/m]_m$ vettore $m - dimensionale$ $i = 1, .., N$ $n^0 = [1/2]_m$ vettore $m - dimensionale$ $\lambda_{kl}^0 = \left(\bar{\omega}_{kl}\right)^2 / \sigma_{kl}^2$ per $k \leq l$ $\tau_{kl}^0 = \bar{\omega}_{kl} / \sigma_{kl}^2$ per $k \leq l$ dove $\bar{\omega}_{kl} = \sum_{(i,j):i<j, i \in k, j \in l} a_{ij} / \sum_{(i,j):i<j} I(i \in k) I(j \in l)$ $\sigma_{kl}^2 = \sum_{(i,j):i<j, i \in k, j \in l} a_{ij}^2 / \sum_{(i,j):i<j} I(i \in k) I(j \in l) - (\bar{\omega}_{kl})^2$ (λ e τ inizializzati attraverso le stime ottenute con il metodo dei momenti)

Tabella A.10: Algoritmo VBEM: modello di Poisson (Input, output ed inizializzazione).

Algoritmo VBEM: modello di Poisson

set $ELBO^{old} = 10^4$
$\varphi^{old} = \varphi^0$
$\lambda_{kl} = \lambda^0_{kl}, \quad k \leq l$
$\tau_{kl} = \tau^0_{kl}, \quad k \leq l$
$n_k = n^0_k, \quad k = 1, \ldots, m$
while $|ELBO^{new} - ELBO^{old}| > \epsilon_1$ **do**

while $|\varphi^{new} - \varphi^{old}| > \epsilon_0$ **do**
for $i \in \{1, \ldots, N\}$
for $k \in \{1, \ldots, m\}$
set $\varphi^{new}_{ik} \propto \exp\left\{\psi(n_k) - \psi\left(\sum_k^m n_k\right) + \right.$
$\left. + \sum_{j \neq i}^N \sum_{l=1}^m \varphi^{old}_{jl}\left[a_{ij}[\psi(\lambda_{kl}) - \ln(\tau_{kl})] - \frac{\lambda_{kl}}{\tau_{kl}}\right]\right\}$
compute $|\varphi^{new} - \varphi^{old}|$
set $\varphi^{new} = \varphi^{old}$
end

set $\varphi = \varphi^{new}$

for $k \in \{1, \ldots, m\}$
$n_k = n^0_k + \sum_{i=1}^N \varphi_{ik}$

for $k = l$
$\lambda_{kk} = \lambda^0_{kk} + \sum_{i<j}^N a_{ij}\varphi_{ik}\varphi_{jk}$
$\tau_{kk} = \tau^0_{kk} + \sum_{i<j}^N \varphi_{ik}\varphi_{jk}$

for $k < l$
$\lambda_{kl} = \lambda^0_{kl} + \sum_{i \neq j}^N a_{ij}\varphi_{ik}\varphi_{jl}$
$\tau_{kl} = \tau^0_{kl} + \sum_{i \neq j}^N \varphi_{ik}\varphi_{jl}$

compute $ELBO^{new} = \ln\left\{\frac{\Gamma\left(\sum_{k=1}^m n^0_k\right)\prod_{k=1}^m \Gamma\left(n_k\right)}{\Gamma\left(\sum_{k=1}^m n_k\right)\prod_{k=1}^m \Gamma\left(n^0_k\right)}\right\} +$
$+ \sum_{1 \leq k \leq l \leq m} \ln\left\{\frac{\Gamma(\lambda_{kl})\tau^{0\,\lambda^0_{kl}}_{kl}}{\tau^{\lambda_{kl}}_{kl}\Gamma(\lambda^0_{kl})}\right\} -$
$- \sum_{i=1}^N \sum_{k=1}^m \varphi_{ik}\ln(\varphi_{ik})$

compute $|ELBO^{new} - ELBO^{old}|$
set $ELBO^{new} = ELBO^{old}$
end
return $\varphi, n, \lambda, \tau$

Tabella A.11: Algoritmo VBEM: modello di Poisson.

Appendice B

Grafici

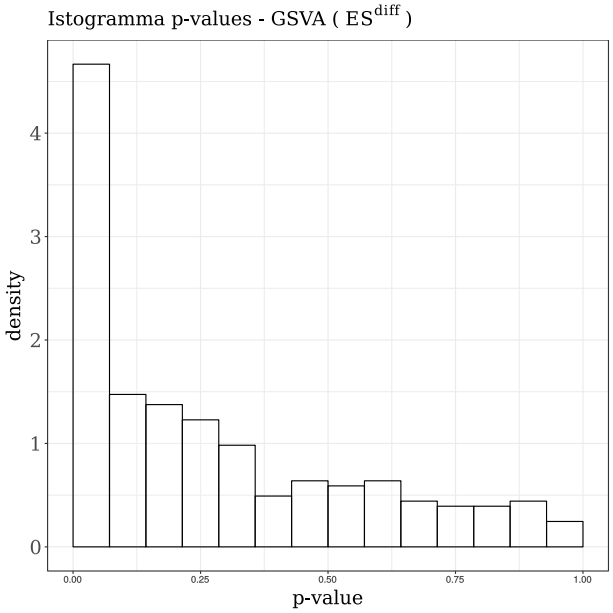


Figura B.1: Istogramma p-values - GSVA (ES^{diff}).

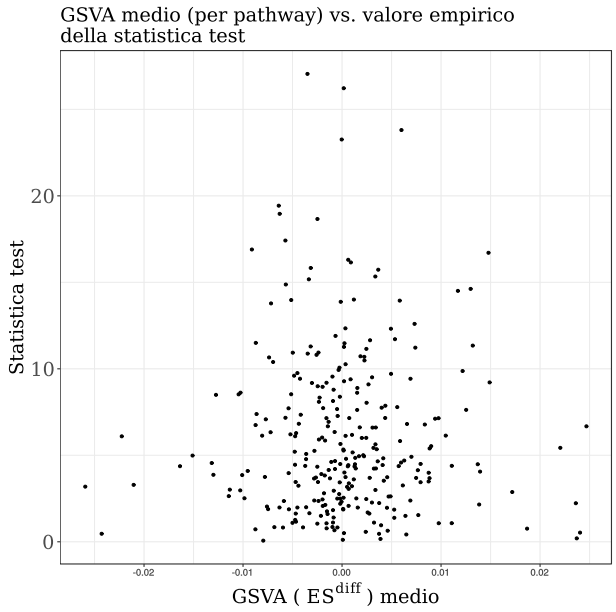


Figura B.2: GSVA medio (per pathway) vs. valore empirico della statistica test (Kruskal-Wallis).

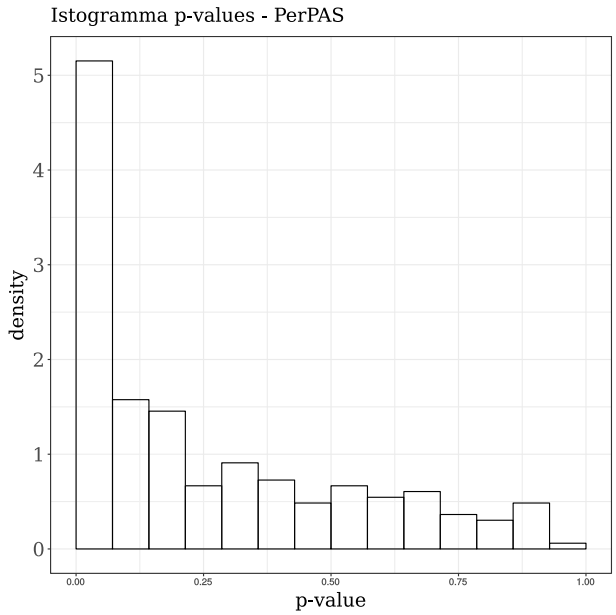


Figura B.3: Istogramma p-values - PerPAS.

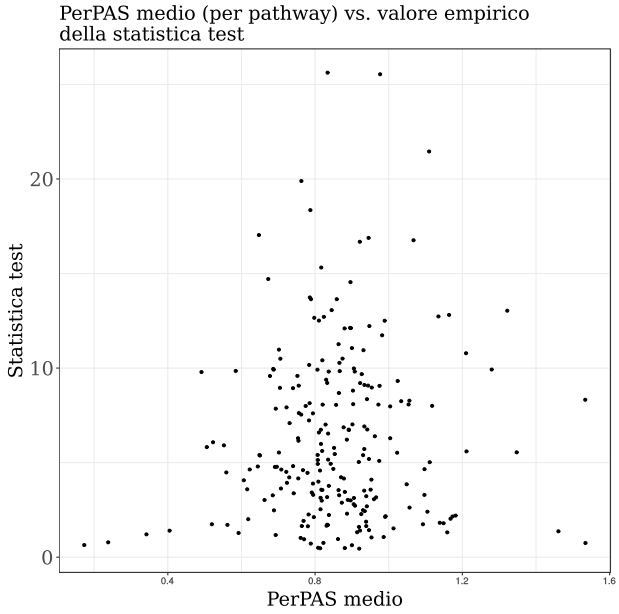


Figura B.4: PerPAS medio (per pathway) vs. valore empirico della statistica test (Kruskal-Wallis).

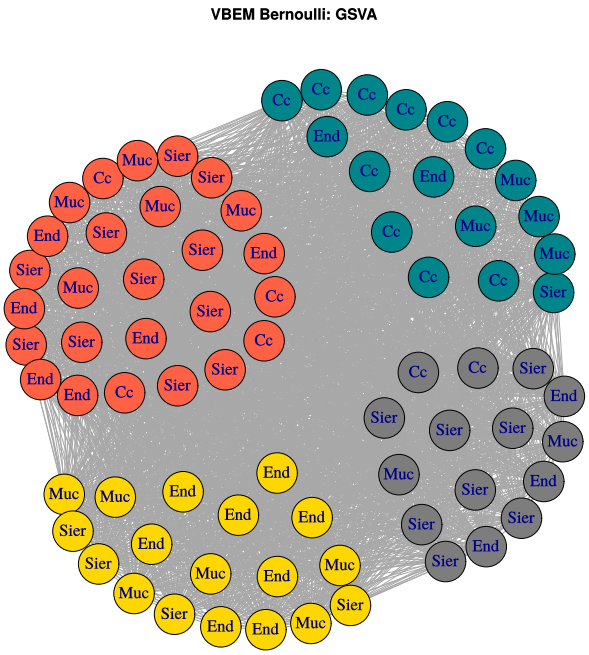


Figura B.5: Rappresentazione grafica rete di pazienti (approccio VBEM con modello di Bernoulli su matrice di adiacenza GSVA). I colori indicano i quattro gruppi identificati dall'algoritmo; su ogni nodo viene riportata l'abbreviazione della vera istologia.

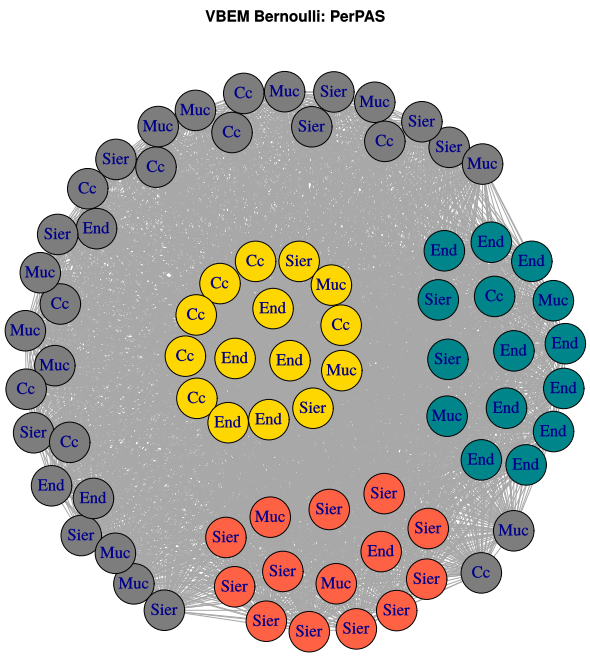


Figura B.6: Rappresentazione grafica rete di pazienti (approccio VBEM con modello di Bernoulli su matrice di adiacenza PerPAS). I colori indicano i quattro gruppi identificati dall'algoritmo; su ogni nodo viene riportata l'abbreviazione della vera istologia.

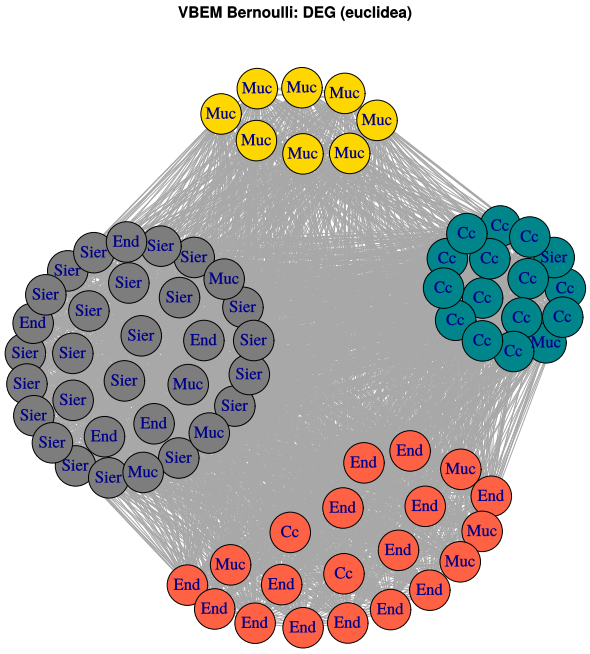


Figura B.7: Rappresentazione grafica rete di pazienti (approccio VBEM con modello di Bernoulli su matrice di adiacenza DEG(d_E)). I colori indicano i quattro gruppi identificati dall'algoritmo; su ogni nodo viene riportata l'abbreviazione della vera istologia.

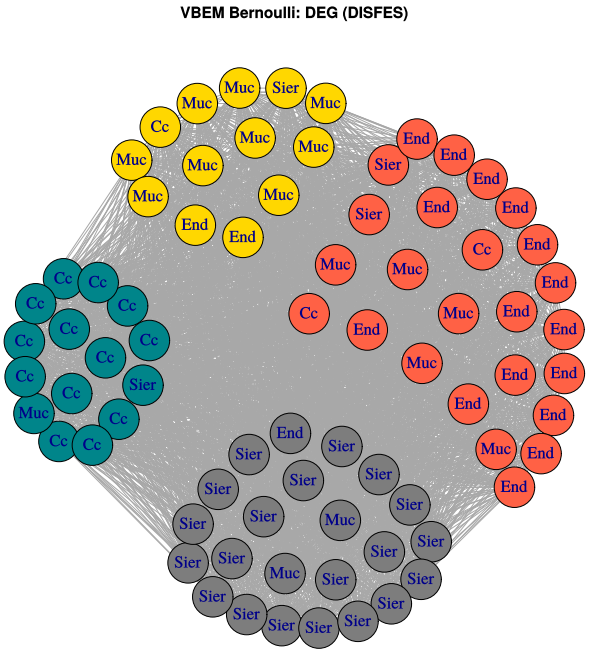


Figura B.8: Rappresentazione grafica rete di pazienti (approccio VBEM con modello di Bernoulli su matrice di adiacenza DEG(d_{FES})). I colori indicano i quattro gruppi identificati dall'algoritmo; su ogni nodo viene riportata l'abbreviazione della vera istologia.

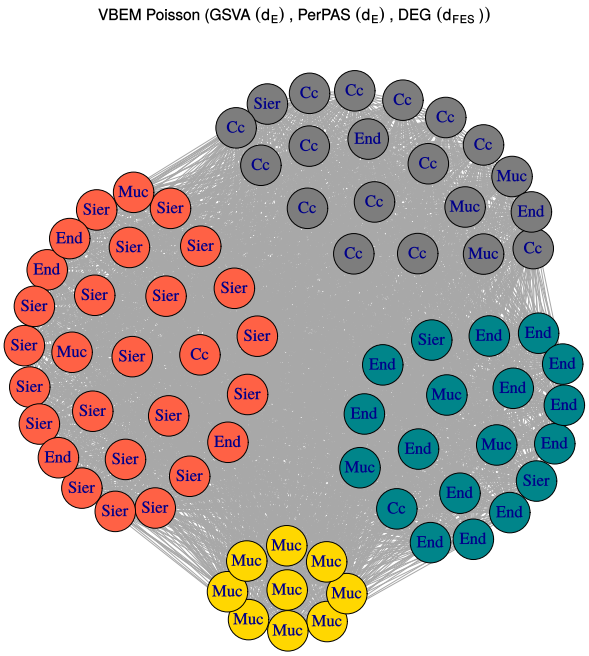


Figura B.9: Rappresentazione grafica rete di pazienti (approccio VBEM con modello di Poisson su matrice adiacenza ottenuta dall'integrazione delle matrici GSVA, PerPAS e DEG(d_{FES})). I colori indicano i quattro gruppi identificati dall'algoritmo; su ogni nodo viene riportata l'abbreviazione della vera istologia.

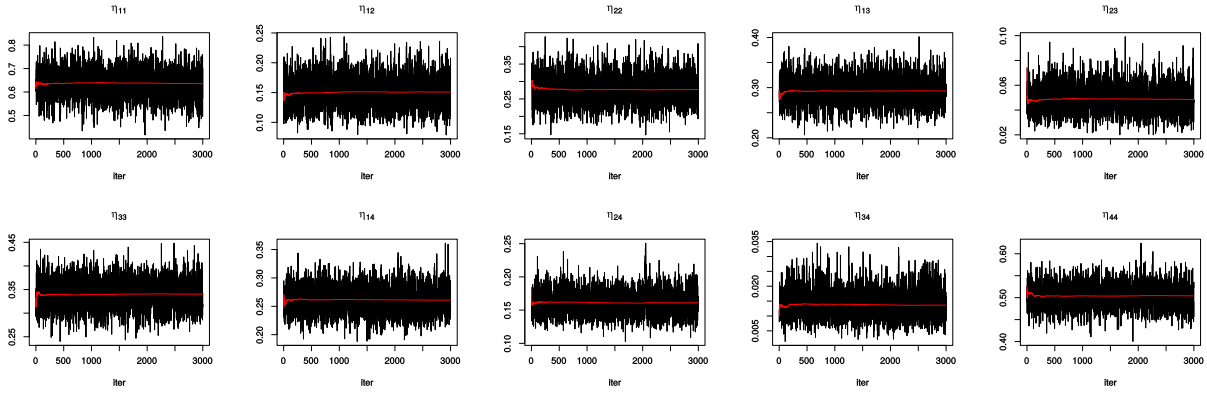


Figura B.10: Gibbs sampling GSVA: grafici valori simulati (3,000) di η. In rosso viene rappresentata la media al crescere del numero di iterazioni.

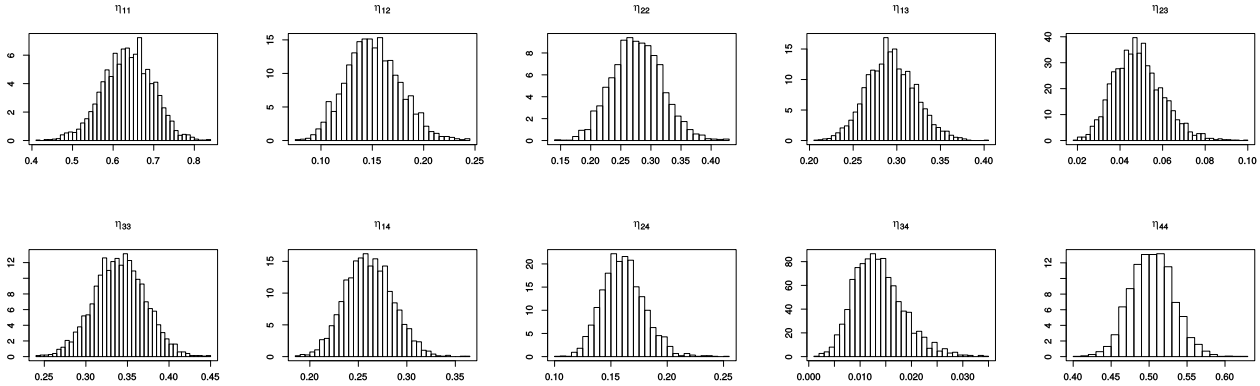


Figura B.11: Gibbs sampling GSVA: istogrammi distribuzioni a posteriori di η.

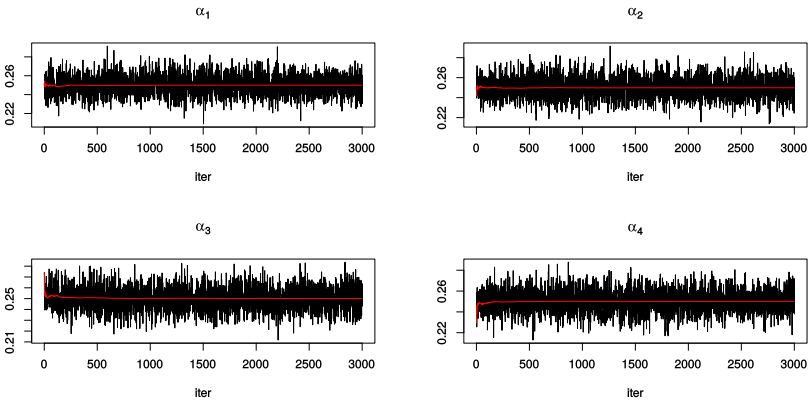


Figura B.12: Gibbs sampling GSVA: grafici valori simulati (3,000) di α. In rosso viene rappresentata la media al crescere del numero di iterazioni.

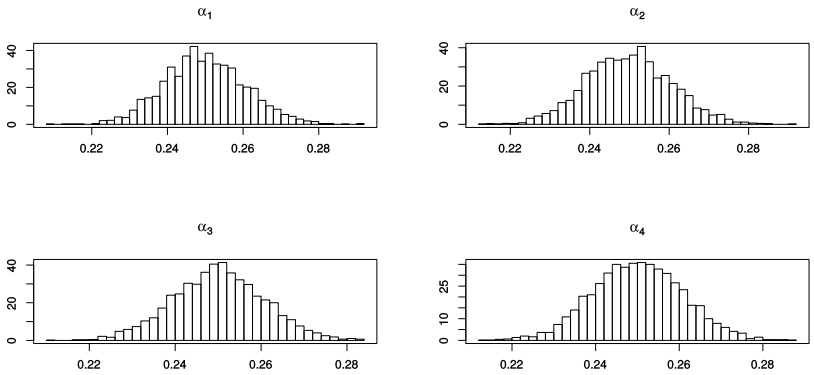


Figura B.13: Gibbs sampling GSVA: istogrammi distribuzioni a posteriori di α.

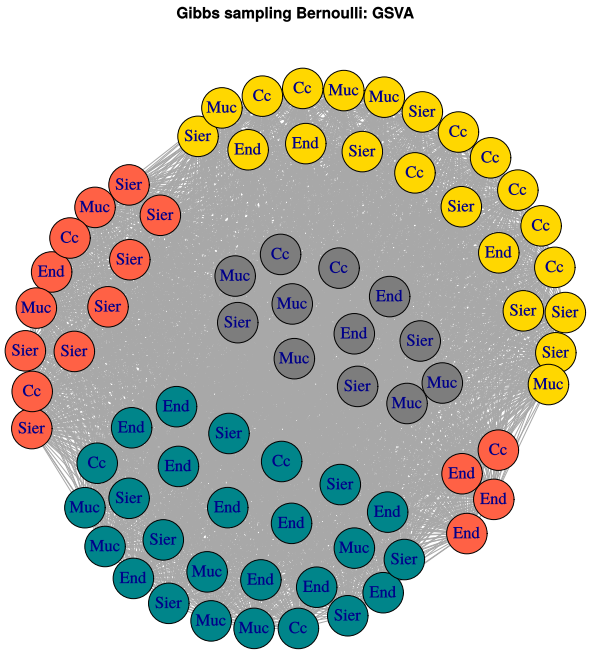


Figura B.14: Rappresentazione grafica rete di pazienti (approccio Gibbs sampling con modello di Bernoulli su matrice di adiacenza GSVA). I colori indicano i quattro gruppi identificati dall'algoritmo; su ogni nodo viene riportata l'abbreviazione della vera istologia.

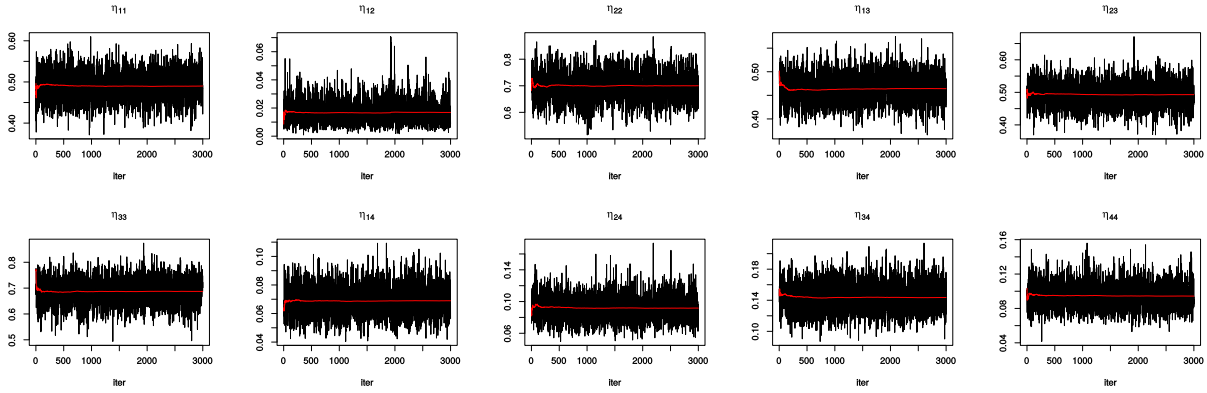


Figura B.15: Gibbs sampling PAS: grafici valori simulati (3,000) di η. In rosso viene rappresentata la media al crescere del numero di iterazioni.

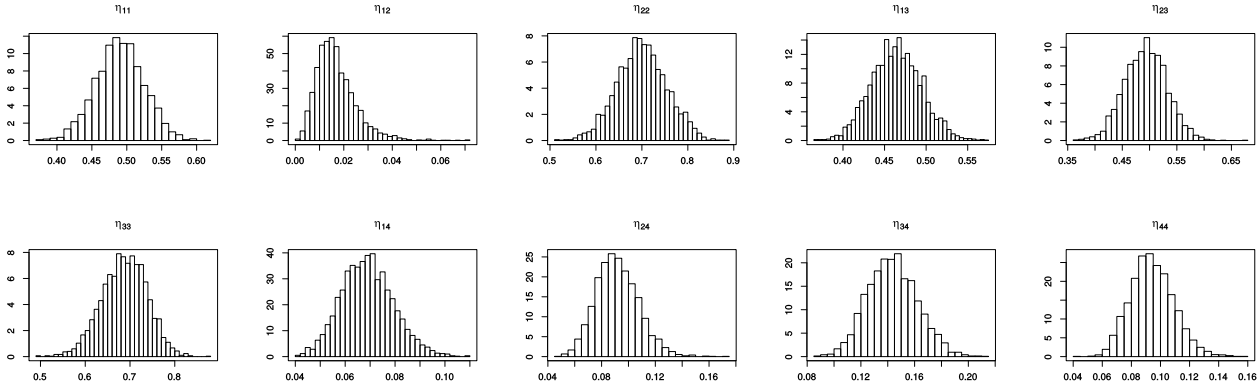


Figura B.16: Gibbs sampling PAS: istogrammi distribuzioni a posteriori di η.

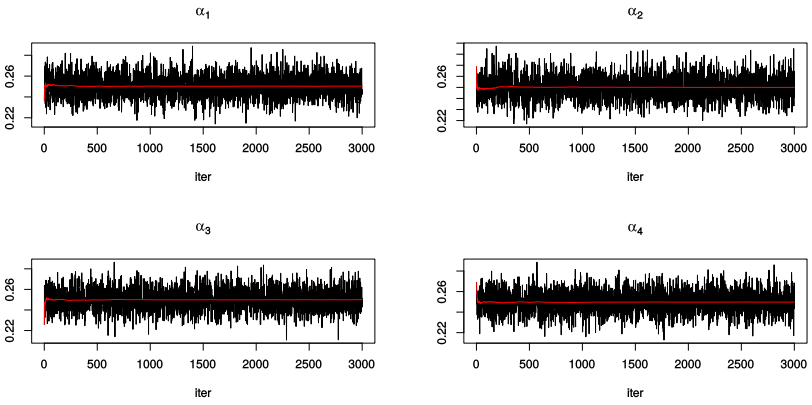


Figura B.17: Gibbs sampling PAS: grafici valori simulati (3,000) di α. In rosso viene rappresentata la media al crescere del numero di iterazioni.

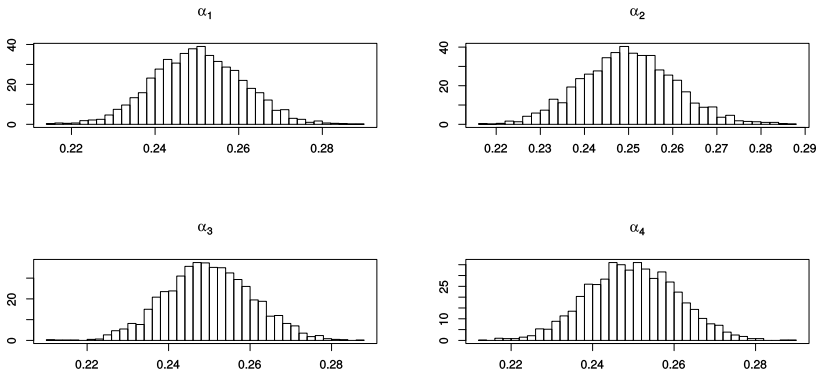


Figura B.18: Gibbs sampling PAS: istogrammi distribuzioni a posteriori di α.

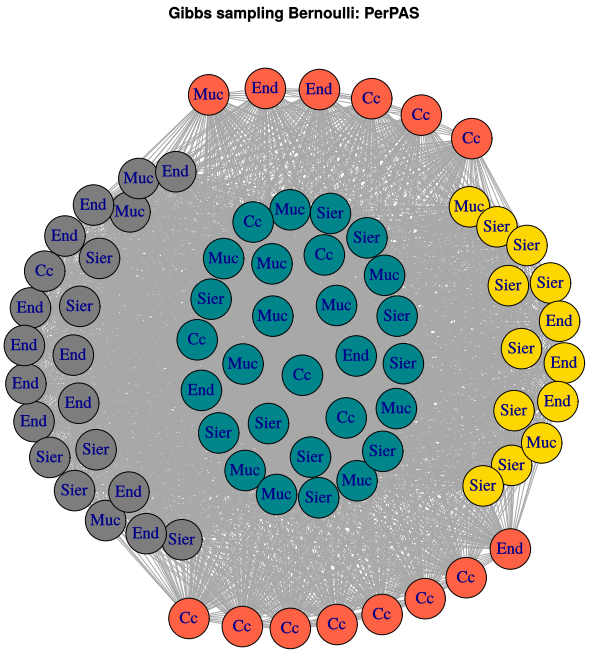


Figura B.19: Rappresentazione grafica rete di pazienti (approccio Gibbs sampling con modello di Bernoulli su matrice di adiacenza PAS). I colori indicano i quattro gruppi identificati dall'algoritmo; su ogni nodo viene riportata l'abbreviazione della vera istologia.

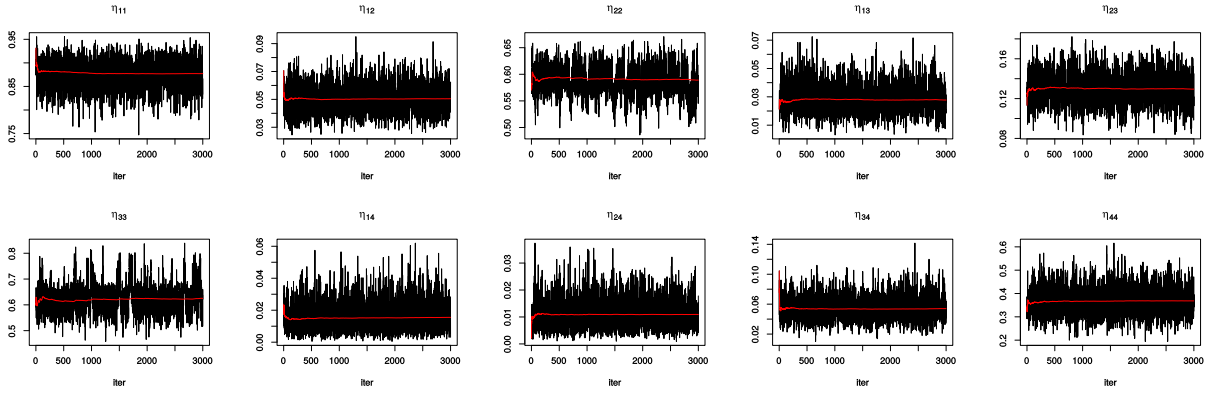


Figura B.20: Gibbs sampling DEG (d_E): grafici valori simulati (3,000) di η. In rosso viene rappresentata la media al crescere del numero di iterazioni.

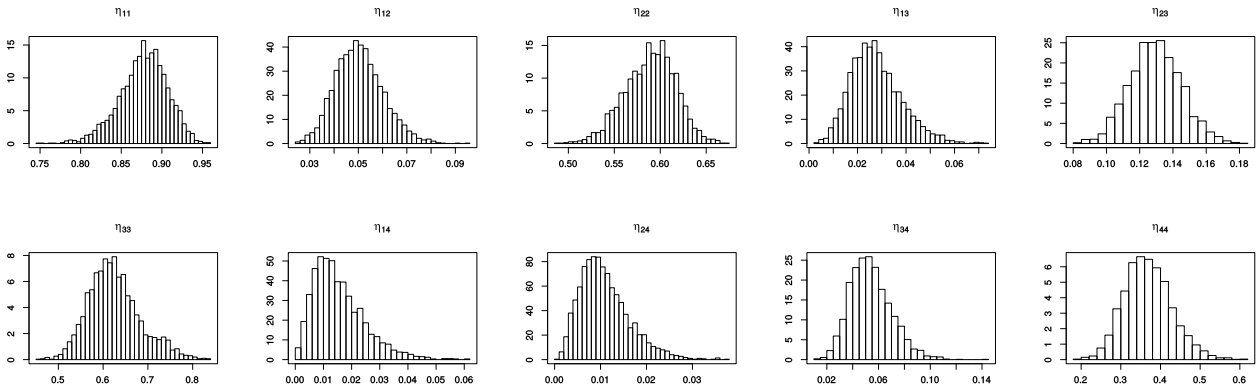


Figura B.21: Gibbs sampling DEG (d_E): istogrammi distribuzioni a posteriori di η.

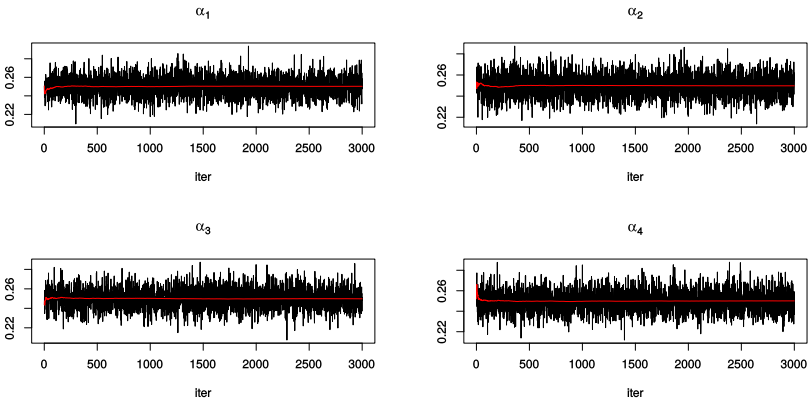


Figura B.22: Gibbs sampling DEG (d_E): grafici valori simulati (3,000) di α. In rosso viene rappresentata la media al crescere del numero di iterazioni.

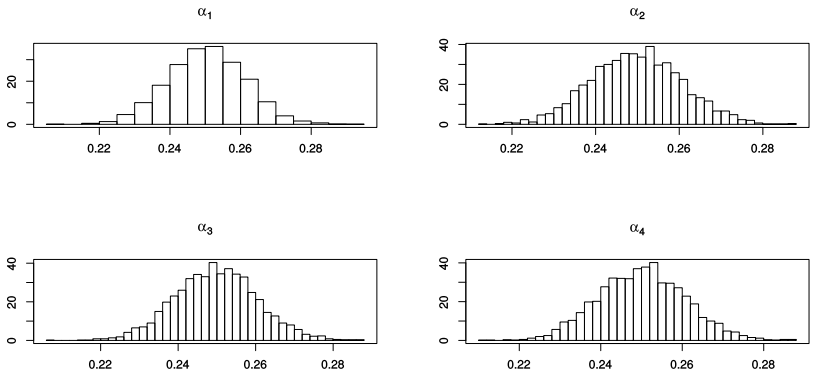


Figura B.23: Gibbs sampling DEG (d_E): istogrammi distribuzioni a posteriori di α.

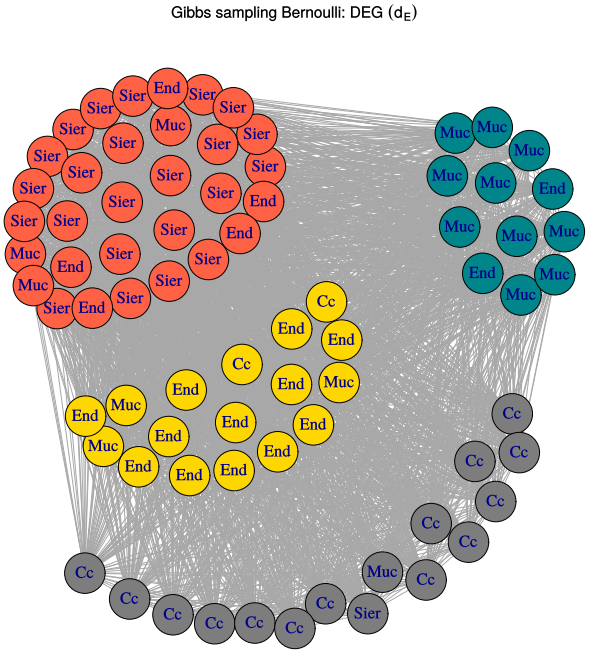


Figura B.24: Rappresentazione grafica rete di pazienti (approccio Gibbs sampling con modello di Bernoulli su matrice di adiacenza DEG (d_E)). I colori indicano i quattro gruppi identificati dall'algoritmo; su ogni nodo viene riportata l'abbreviazione della vera istologia.

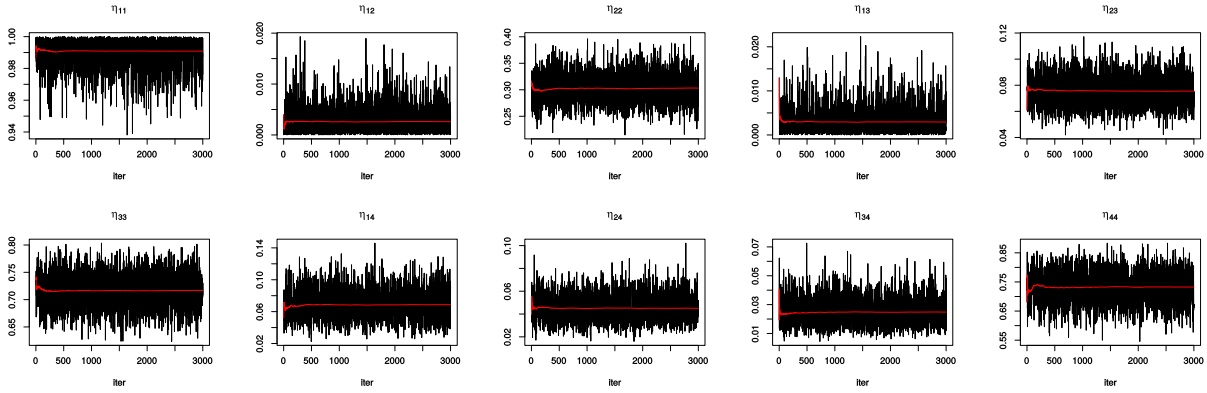


Figura B.25: Gibbs sampling DEG (d_{FES}): grafici valori simulati (3,000) di η. In rosso viene rappresentata la media al crescere del numero di iterazioni.

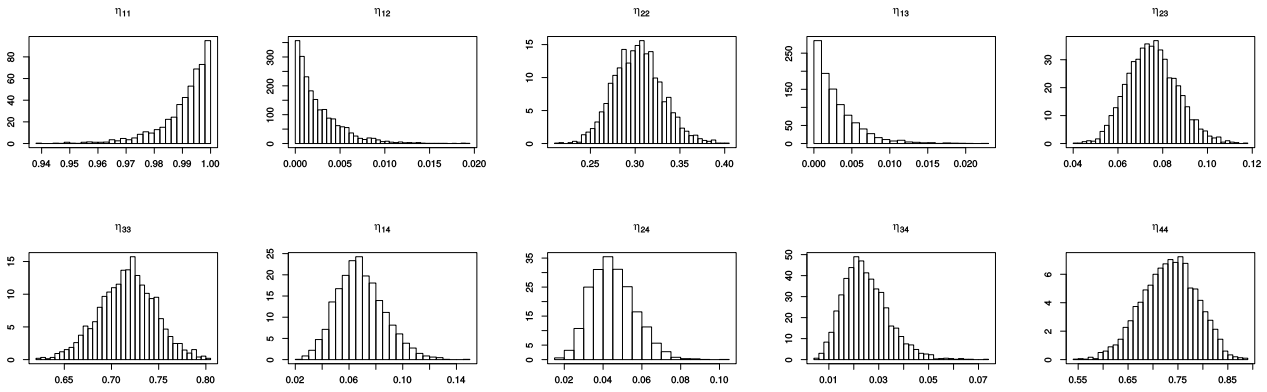


Figura B.26: Gibbs sampling DEG (d_{FES}): istogrammi distribuzioni a posteriori di η.

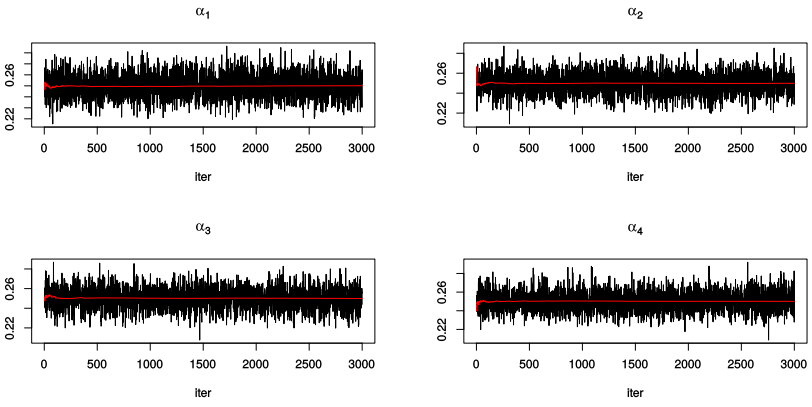


Figura B.27: Gibbs sampling DEG (d_{FES}): grafici valori simulati (3,000) di α. In rosso viene rappresentata la media al crescere del numero di iterazioni.

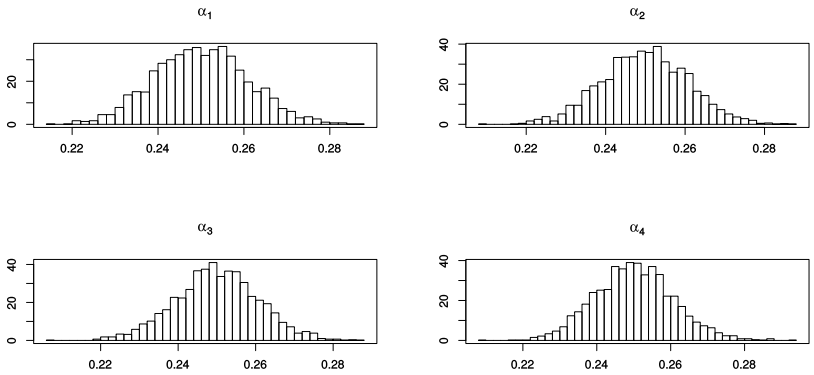


Figura B.28: Gibbs sampling DEG (d_{FES}): istogrammi distribuzioni a posteriori di α.

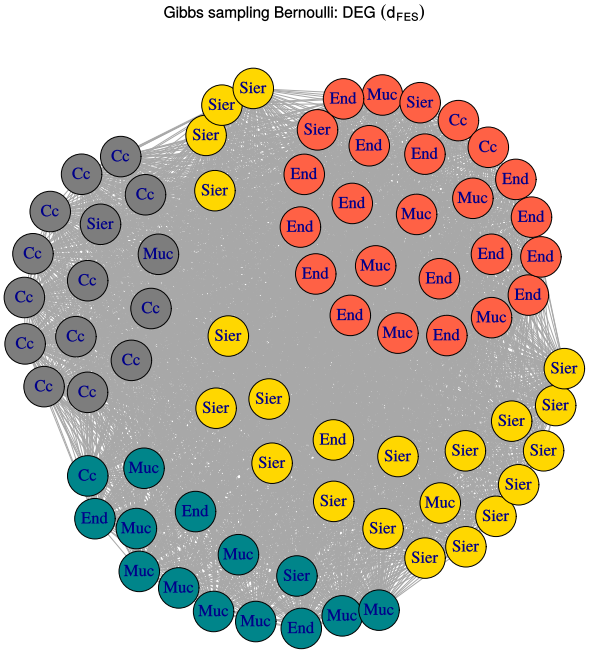


Figura B.29: Rappresentazione grafica rete di pazienti (approccio Gibbs sampling con modello di Bernoulli su matrice di adiacenza DEG (d_{FES})). I colori indicano i quattro gruppi identificati dall'algoritmo; su ogni nodo viene riportata l'abbreviazione della vera istologia.

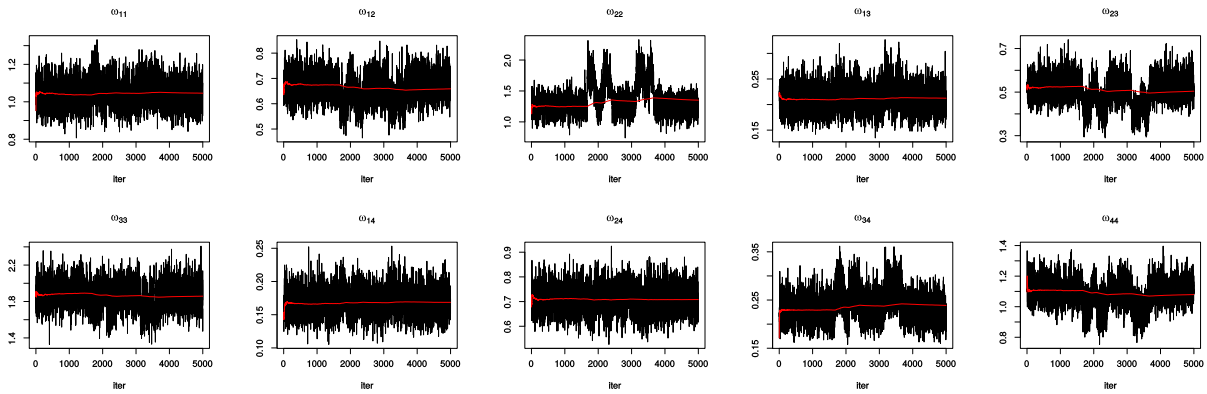


Figura B.30: Gibbs sampling modello Poisson: grafici valori simulati (5,000) di ω. In rosso viene rappresentata la media al crescere del numero di iterazioni.

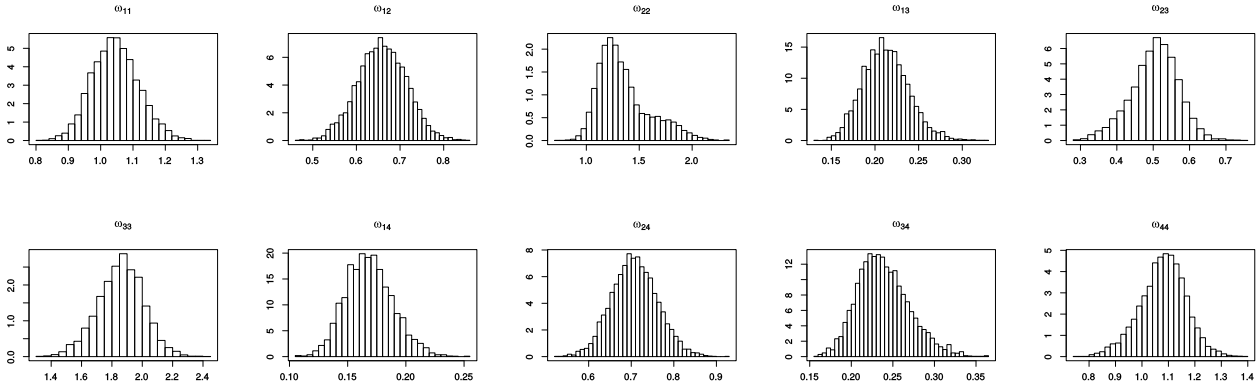


Figura B.31: Gibbs sampling modello di Poisson: istogrammi distribuzioni a posteriori di ω.

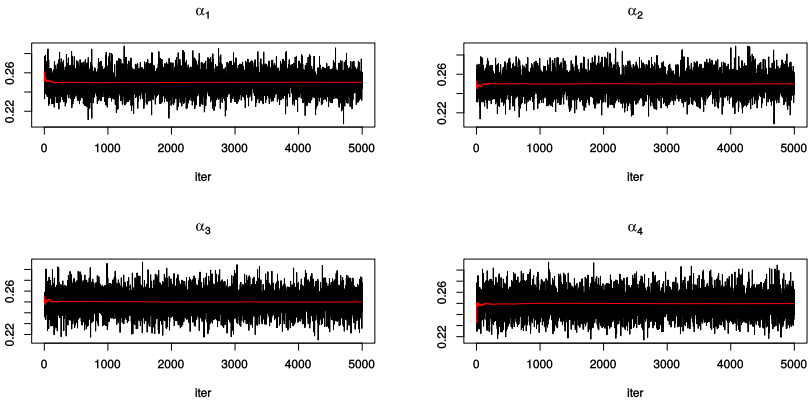


Figura B.32: Gibbs sampling modello di Poisson: grafici valori simulati (5,000) di α. In rosso viene rappresentata la media al crescere del numero di iterazioni.

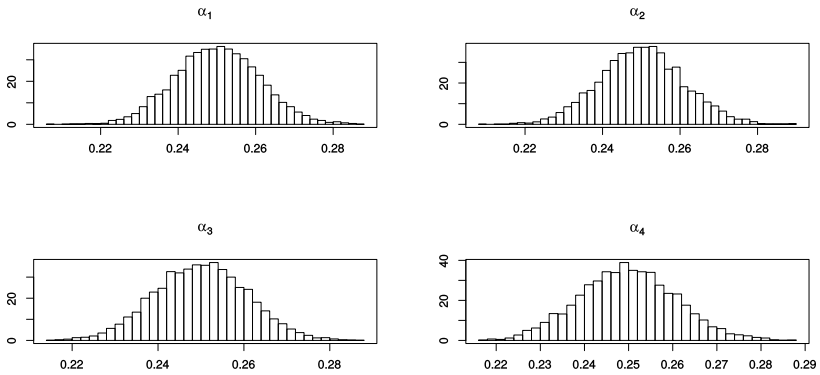


Figura B.33: Gibbs sampling modello di Poisson: istogrammi distribuzioni a posteriori di α.

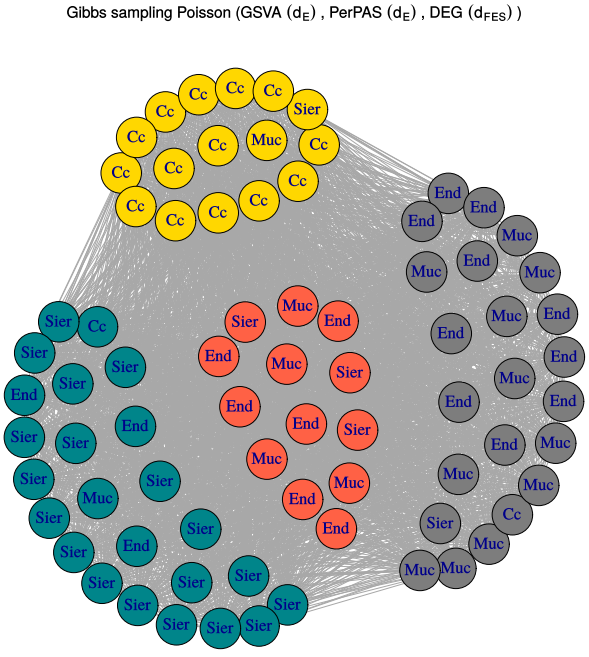


Figura B.34: Rappresentazione grafica rete di pazienti (approccio Gibbs sampling con modello di Poisson. I colori indicano i quattro gruppi identificati dall'algoritmo; su ogni nodo viene riportata l'abbreviazione della vera istologia.

Appendice C

Codici

C.1 VBEM: Poisson

```
# Variational Inference mediante ELBO e CAVI Algorithm (modello di Poisson)
#numero di possibili relazioni intra e fra gruppi
nkl.fun <- function(nk)
        {
        m1 <- nk%*%t(nk)
        diag(m1) <- sapply(nk,function(x) choose(x,2))
        m1[which(m1==0)] <- 1
        m1
        }
#numero di relazioni osservate intra e fra gruppi
ekl.fun <- function(A,X,K)
        {
        X <- factor(X,levels=1:K)
        A[lower.tri(A,diag=TRUE)] <- 0
        m1 <- apply(apply(A,1,function(y) tapply(y,X,sum)),
                1,function(y)tapply(y,X,sum))
        m1 <- m1[upper.tri(m1,diag=TRUE)]
        m1[which(is.na(m1)==TRUE)] <- 0
        m01 <- matrix(NA,nrow=K,ncol=K)
        m01[upper.tri(m01,diag=T)] <- m1
        m01[lower.tri(m01,diag=F)] <- t(m01)[lower.tri(m01,diag=F)]
        m01[is.na(m01)] <- 0
        m01
        }
#funzione per creare matrice di coppie indici (triangolare superiore, i<j)
aij.fun <- function(n)
        {
        x <- NULL
        for(i in 1:n)
                        {
                        j=i
                        while(j<n)
                        {
```

```
                        j <- j+1
                        x <- rbind(x,c(i,j))
                         }
                         }
        return(x)
        }
#funzione per creare matrice con tutte le coppie tranne diagonale (j!=i)
aij.fun2 <- function(n)
        {
        x <- expand.grid(1:n,1:n)
        cond <- which(apply(x,1,function(y) y[1]==y[2]))
        x <- x[-cond,]
        return(x)
        }
#matrice nxK classificazione
class.fun <- function(n,X,K)
        {
        x<-matrix(0,nrow=n,ncol=K)
        for(i in 1:n)
                {
                c <- X[i]
                x[i,c] <- 1
                }
        return(x)
        }
#inizializzazione parametri con metodo dei momenti (MM)
init.param <- function(A,X,K)
  {
  omega <- ekl.fun(A=A,X=X,K=K)
  nkl <- nkl.fun(as.vector(table(X)))
  m <- omega/nkl

  omega2 <- ekl.fun(A=A^2,X=X,K=K)
  m2 <- omega2/nkl

  sigma2 <- m2-m^2

  a <- (m^2/sigma2)
  b <- (m/sigma2)

  return(list(a=a,b=b))
  }

#####################################################################
#CAVI ALGORITHM: POISSON MODEL, INTEGRATION OF ADJACENCY MATRICES#
#####################################################################
CAVI_POISSON <-  function(A,K,eps=1e-06,eps0=1e-04,X)
        {
        #A = matrice di adiacenza
        #K = numero di comunita'
        #eps = epsilon per la convergenza delle probabilita' di classificazione
```

```
85          #eps = epsilon per la convergenza dell'intero algoritmo
86          #X = vettore per inizializzazione algoritmo
87          n <- dim(A)[1]
88          #coppie indici i<j
89          phiaij <- aij.fun(n)
90          #coppie indici i!=j
91          phiaij2 <- aij.fun2(n)
92
93          #inizializzazione parametri variazionali:
94          #assegnazione ai K gruppi
95          phi <- matrix(rep(rep(1/K,K),n),
96                  nrow=n,ncol=K,byrow=TRUE)
97          #parametro nq
98          nq <- rep(1/2,K)
99
100         #parametro nq0
101         nq0 <- rep(1/2,K)
102         #parametri a[k,l] e b[k,l]
103         initials <- init.param(A=A,X=X,K=K)
104
105         a0 <- a01 <- initials$a
106         b0 <- b01 <- initials$b
107         #controllo parametri
108         a01[which(a01 == 0)] <- 1
109         b01[which(b01 == 0)] <- 1
110         a0[which(a0 == 0)] <- 1
111         b0[which(b0 == 0)] <- 1
112
113         #inizializzazione parametri convergenza
114         ELBO <- 0  #valore iniziale per il Lower Bound
115         epsilon <- epsilon0 <- 10^3 #epsilon iniziale per i while
116         iter <- 0 #iterazioni
117
118         #inizio algoritmo
119         while(epsilon > eps0)
120                 {
121         #aggiornameto classificazione
122                 exp.1 <- digamma(a0)-log(b0)
123                 exp.2 <- a0/b0
124
125                 epsilon0 <- 10^3
126                 while(epsilon0 > eps)
127                 {
128                 phi.old <- phi
129                 for(i in 1:n)
130                 {
131                 adj <- A[i,]
132                 for(k in 1:K)
133                 {
134                 somma <- 0
135                 seque1 <- 1:n
```

```
136                 seque1 <- seque1[-i]
137                 exp0 <- digamma(nq[k])-digamma(sum(nq))
138                 for(j in seque1)
139                 {
140                 somma <- somma +
141                  sum(phi[j,]*(adj[j]*exp.1[k,]-exp.2[k,]))
142                 }
143                 phi[i,k] <- exp(somma + exp0)
144                 }
145                  #normalizzazione probabilit
146                 phi[i,] <- phi[i,]/sum(phi[i,])
147                 }
148                 epsilon0 <- sum(abs(phi.old-phi))
149                 }
150
151                 nq <- nq0
152                 nq <- nq + apply(phi,2,sum)
153
154                 a0 <- a01
155                 b0 <- b01
156                 #aggiornamento parametri variazionali a[k,l] e b[k,l]
157                 for(k in 1:K)
158                 {
159                 for(l in 1:K)
160                 {
161                 if(k==l)
162                         {
163                         phi.a <- as.vector(apply(phiaij,1,
164                         function(y)
165                          A[y[1],y[2]]*phi[y[1],k]*phi[y[2],l]))
166
167                         phi.b <- as.vector(apply(phiaij,1,
168                         function(y)
169                          phi[y[1],k]*phi[y[2],l]))
170
171                         a0[k,l] <- a0[k,l] + sum(phi.a)
172                         b0[k,l] <- b0[k,l] + sum(phi.b)
173                         }
174                 else
175                         {
176                         phi.a <- as.vector(apply(phiaij2,1,
177                         function(y)
178                          A[y[1],y[2]]*phi[y[1],k]*phi[y[2],l]))
179
180                         phi.b <- as.vector(apply(phiaij2,1,
181                         function(y)
182                         phi[y[1],k]*phi[y[2],l]))
183
184                         a0[k,l] <- a0[k,l] + sum(phi.a)
185                         b0[k,l] <- b0[k,l] + sum(phi.b)
186                         }
```

```
187
188                 }
189                 }
190                 #calcolo ELBO (Lower Bound)
191                 ua01 <- a01[upper.tri(a01,diag=T)]
192                 ub01 <- b01[upper.tri(b01,diag=T)]
193                 ua0 <- a0[upper.tri(a0,diag=T)]
194                 ub0 <- b0[upper.tri(b0,diag=T)]
195
196                 ELBO0  <- lgamma(sum(nq0))+sum(lgamma(nq)) -
197                           lgamma(sum(nq))-sum(lgamma(nq0)) +
198                           sum(-ua0*log(ub0)-lgamma(ua01)+
199                           lgamma(ua0)+ua01*log(ub01) ) -
200                           sum(phi*log(phi))
201
202
203
204
205
206                 #aggiornamento parametri convergenza
207                 ELBO <- c(ELBO,ELBO0)
208                 epsilon <- abs(ELBO[iter] - ELBO[iter-1])
209                 iter <- iter + 1
210
211                 #stampa
212                 print(c(ELBO0,iter,epsilon))
213                 }
214                 #fine algoritmo
215
216         #output funzione
217         return(list(phi=phi,nq=nq,a0=a0,b0=b0,lunghezza=length(ELBO),elbo=ELBO))
218         }
```

C.2 Gibbs: Bernoulli

```
1
2 SBM_GIBBS_BERNOULLI <- function(Nsim,init,K,A,Ti,a,b)
3         {
4         #Nsim = numero di simulazioni (oltre le simulazioni del burn-in)
5         #init = vettore classificazione per l'inizializzazione dell'algoritmo
6         #A = matrice di adiacenza
7         #K = numero di comunita'
8         #Ti = parametro a priori per Dirichlet
9         #a = parametro a priori per Gamma
10        #b = parametro a priori per Gamma
11
12        n <- dim(A)[1]
13        require(gtools) #per la distribuzione Dirichlet
14        x <- init
15        Ti <- Ti*K
```

```
16          nk<-as.vector(table(factor(x,levels=1:K)))
17
18          #classificazione secondo  'init'
19          Z <- class.fun(n=n,X=init,K=K)
20          #matrice probabilit  di assegnazione
21          phi <- matrix(rep(rep(1/K,K),n),
22                  nrow=n,ncol=K,byrow=TRUE)
23
24          #triangolare superiore k<l
25          kl <- fun2(K)
26          #diagonale k==l
27          kk <- cbind(1:K,1:K)
28
29          #coppie indici triangolare superiore i<j
30          phiaij <- aij.fun(n)
31
32          #coppie indici i!=j
33          phiaij2 <- aij.fun2(n)
34
35          #inizializziamo a[k,l] e b[k,l] della Beta
36          akl <- apply(kl,1,function(y)
37          sum(apply(phiaij2,1,function(w) A[w[1],w[2]]*Z[w[1],
38          y[1]]*Z[w[2],y[2]]  )) )
39
40          bkl <- apply(kl,1,function(y)
41          sum(apply(phiaij2,1,function(w) (1-A[w[1],w[2]])*Z[w[1],
42          y[1]]*Z[w[2],y[2]]  )) )
43
44          akk <- apply(kk,1,function(y)
45          sum(apply(phiaij,1,function(w) A[w[1],w[2]]*Z[w[1],
46          y[1]]*Z[w[2],y[2]]  )) )
47
48          bkk <- apply(kk,1,function(y)
49          sum(apply(phiaij,1,function(w) (1-A[w[1],w[2]])*Z[w[1],
50          y[1]]*Z[w[2],y[2]]  )) )
51
52          a0 <- b0 <- matrix(0,nrow=K,ncol=K)
53          diag(a0) <- akk
54          diag(b0) <- bkk
55          a0[upper.tri(a0,diag=F)] <- akl
56          b0[upper.tri(b0,diag=F)] <- bkl
57
58          #Parametri di Beta
59          sh1 <- a0[upper.tri(a0,diag=T)] + a
60          sh2 <- b0[upper.tri(b0,diag=T)] + b
61
62
63          #Parametri Burn-in
64          M0 <- 5000 #numero simulazioni prima parte burnin
65          T.vec <- seq(10*76,Ti,length=M0)
66          w.vec <- seq(1/n,1,length=M0)
```

```
67
68          #BURN-IN : PRIMA FASE (INIZIO)
69          #M0 = 5,000 simulazioni
70          for(i in 1:M0)
71                  {
72                  theta <- as.vector(rdirichlet(1,nk+T.vec[i])) #theta
73                  eta <- rbeta(length(sh1),shape1=sh1*w.vec[i],shape2=sh2*w.vec[i]) #eta
74
75                  #matrice eta
76                  hat1 <- matrix(0,nrow=K,ncol=K)
77                  hat1[upper.tri(hat1,diag=TRUE)] <- eta
78                  hat1[lower.tri(hat1)] <- t(hat1)[lower.tri(hat1)]
79
80                  #inizio predizione gruppi (classificazione Z)
81                  for(l in 1:n)
82                          {
83                          for(j in 1:K)
84                                  {
85                          seque1 <- 1:n
86                          seque1 <- seque1[-l]
87                          si <- sapply(1:K, function(y) sum(sapply(seque1,function(w)
88                          A[l,w]*Z[w,y])) )
89                          no <- sapply(1:K, function(y) sum(sapply(seque1,function(w)
90                          (1-A[l,w])*Z[w,y])) )
91                          phi[l,j] <- theta[j]*prod( (hat1[j,]^si)*(1-hat1[j,])^no)
92                                  }
93                          Z[l,] <- as.vector(class.fun(1,which.max(phi[l,]),K))
94                          }
95                  #fine predizione gruppi (classificazione Z)
96
97          #calcoli per aggiornamento parametri Beta
98          akl <- apply(kl,1,function(y)
99          sum(apply(phiaij2,1,function(w) A[w[1],w[2]]*Z[w[1],
100         y[1]]*Z[w[2],y[2]]  )) )
101
102         bkl <- apply(kl,1,function(y)
103         sum(apply(phiaij2,1,function(w) (1-A[w[1],w[2]])*Z[w[1],
104         y[1]]*Z[w[2],y[2]]  )) )
105
106         akk <- apply(kk,1,function(y)
107         sum(apply(phiaij,1,function(w) A[w[1],w[2]]*Z[w[1],
108         y[1]]*Z[w[2],y[2]]  )) )
109
110         bkk <- apply(kk,1,function(y)
111         sum(apply(phiaij,1,function(w) (1-A[w[1],w[2]])*Z[w[1],
112         y[1]]*Z[w[2],y[2]]  )) )
113
114         a0 <- b0 <- matrix(0,nrow=K,ncol=K)
115         diag(a0) <- akk
116         diag(b0) <- bkk
117         a0[upper.tri(a0,diag=F)] <- akl
```

```
118         b0[upper.tri(b0,diag=F)] <- bkl
119
120         #calcolo per aggiornamento parametri Dirichlet
121         nk<-as.vector(table(factor(apply(Z,2,sum),levels=1:K)))
122
123         #aggiornamento parametri distribuzioni
124         #a posteriori (Dirichlet e Beta)
125         sh1 <- a0[upper.tri(a0,diag=T)] + a
126         sh2 <- b0[upper.tri(b0,diag=T)] + b
127         alpha <- nk + Ti
128
129         #stampa avanzamento burn-in
130         print(paste("burn-in step:",i,collapse=""))
131         }
132         #BURN-IN : PRIMA FASE (FINE)
133
134         #salvataggio informazioni fine burn-in
135         Z0 <- Z
136         phi0 <- phi
137
138         #calcoli per aggiornamento parametri Beta
139         akl <- apply(kl,1,function(y)
140         sum(apply(phiaij2,1,function(w) A[w[1],w[2]]*Z[w[1],
141         y[1]]*Z[w[2],y[2]]  )) )
142
143         bkl <- apply(kl,1,function(y)
144         sum(apply(phiaij2,1,function(w) (1-A[w[1],w[2]])*Z[w[1],
145         y[1]]*Z[w[2],y[2]]  )) )
146
147         akk <- apply(kk,1,function(y)
148         sum(apply(phiaij,1,function(w) A[w[1],w[2]]*Z[w[1],
149         y[1]]*Z[w[2],y[2]]  )) )
150
151         bkk <- apply(kk,1,function(y)
152         sum(apply(phiaij,1,function(w) (1-A[w[1],w[2]])*Z[w[1],
153         y[1]]*Z[w[2],y[2]]  )) )
154
155         a0 <- b0 <- matrix(0,nrow=K,ncol=K)
156         diag(a0) <- akk
157         diag(b0) <- bkk
158         a0[upper.tri(a0,diag=F)] <- akl
159         b0[upper.tri(b0,diag=F)] <- bkl
160
161         #calcolo per aggiornamento parametri Dirichlet
162         nk<-as.vector(table(factor(apply(Z,2,sum),levels=1:K)))
163
164         #aggiornamento parametri distribuzioni
165         #a posteriori (Dirichlet e Beta)
166         sh1 <- a0[upper.tri(a0,diag=T)] + a
167         sh2 <- b0[upper.tri(b0,diag=T)] + b
168         alpha <- nk + Ti
```

```
169
170         #creazione matrici per il salvataggio delle informazioni
171         theta <- matrix(NA,nrow=Nsim+M0,ncol=length(alpha))
172         eta <- matrix(NA, nrow=Nsim+M0, ncol=length(sh1))
173         Z10 <- phi10 <- array(NA,c(n,K,Nsim+M0))
174
175         #CONTINUO BURN-IN + SIMULAZIONE (INIZIO)
176         for(i in 1:(M0+Nsim))
177                 {
178                 theta[i,] <- as.vector(rdirichlet(1,alpha))  #theta
179                 eta[i,] <- rbeta(length(sh1),shape1=sh1,shape2=sh2) #eta
180
181                 #matrice eta
182                 hat1 <- matrix(0,nrow=K,ncol=K)
183                 hat1[upper.tri(hat1,diag=TRUE)] <- eta[i,]
184                 hat1[lower.tri(hat1)] <- t(hat1)[lower.tri(hat1)]
185
186         #inizio predizione gruppi (classificazione Z)
187         for(l in 1:n)
188                         {
189                         for(j in 1:K)
190                                 {
191                         seque1 <- 1:n
192                         seque1 <- seque1[-l]
193                         si <- sapply(1:K, function(y) sum(sapply(seque1,function(w)
194                         A[l,w]*Z[w,y])) )
195                         no <- sapply(1:K, function(y) sum(sapply(seque1,function(w)
196                         (1-A[l,w])*Z[w,y])) )
197                         phi[l,j] <- theta[i,j]*prod( (hat1[j,]^si)*(1-hat1[j,])^no)
198                                 }
199                         Z[l,] <- as.vector(class.fun(1,which.max(phi[l,]),K))
200                         }
201         #fine predizione gruppi (classificazione Z)
202         phi10[,,i] <- phi
203         Z10[,,i] <- Z
204
205         #calcoli per aggiornamento parametri Beta
206         akl <- apply(kl,1,function(y)
207         sum(apply(phiaij2,1,function(w) A[w[1],w[2]]*Z[w[1],
208         y[1]]*Z[w[2],y[2]]  )) )
209
210         bkl <- apply(kl,1,function(y)
211         sum(apply(phiaij2,1,function(w) (1-A[w[1],w[2]])*Z[w[1],
212         y[1]]*Z[w[2],y[2]]  )) )
213
214         akk <- apply(kk,1,function(y)
215         sum(apply(phiaij,1,function(w) A[w[1],w[2]]*Z[w[1],
216         y[1]]*Z[w[2],y[2]]  )) )
217
218         bkk <- apply(kk,1,function(y)
219         sum(apply(phiaij,1,function(w) (1-A[w[1],w[2]])*Z[w[1],
```

```
220         y[1]]*Z[w[2],y[2]]  )) )
221
222         a0 <- b0 <- matrix(0,nrow=K,ncol=K)
223         diag(a0) <- akk
224         diag(b0) <- bkk
225         a0[upper.tri(a0,diag=F)] <- akl
226         b0[upper.tri(b0,diag=F)] <- bkl
227
228         #calcolo per aggiornamento parametri Dirichlet
229         nk<-as.vector(table(factor(apply(Z,2,sum),levels=1:K)))
230
231         #aggiornamento parametri distribuzioni
232         #a posteriori (Dirichlet e Beta)
233         sh1 <- a0[upper.tri(a0,diag=T)] + a
234         sh2 <- b0[upper.tri(b0,diag=T)] + b
235         alpha <- nk + Ti
236
237         #stampa avanzamento simulazione
238         print(paste("algorithm step:",i,collapse=""))
239                 }
240         #CONTINUO BURN-IN + SIMULAZIONE (FINE)
241
242         return(list(init0=init,Z0=Z,phi0=phi,phi=phi10,theta=theta,eta=eta,Z10=Z10,phi_last
      =phi))
243 #Z0 e phi0 sono i risultati dopo le M0 iterazioni della prima fase di burn-in
244 #Z10 e phi sono i risultati sulle M0+Nsim simulazioni
245         }
```

C.3 Gibbs: Poisson

```
1  SBM_GIBBS_POISSON <- function(Nsim,X,K,A,Ti,a,b)
2          {
3          #Nsim = numero di simulazioni (oltre le simulazioni del burn-in)
4          #init = vettore classificazione per l'inizializzazione dell'algoritmo
5          #A = matrice di adiacenza
6          #K = numero di comunita'
7          #Ti = parametro a priori per Dirichlet
8          #a = parametro a priori per Beta
9          #b = parametro a priori per Beta
10
11         require(gtools) #per la distribuzione Dirichlet
12         Ti <- Ti*K
13         n <- dim(A)[1]
14         nk <- as.vector(table(factor(X,levels=1:K)))
15         init <- X
16
17         #coppie indici triangolare superiore i<j
18         phiaij <- aij.fun(n)
19
20         #coppie indici i!=j
```

```
21          phiaij2 <- aij.fun2(n)
22
23          #classificazione secondo  'X'
24          Z0 <- class.fun(n=n,X=X,K=K)
25          #matrice probabilit  di assegnazione
26          phi <- matrix(rep(rep(1/K,K),n),
27                  nrow=n,ncol=K,byrow=TRUE)
28
29          #triangolare superiore k<l
30          kl <- fun2(K)
31          #diagonale k==l
32          kk <- cbind(1:K,1:K)
33
34          #inizializziamo a[k,l] e b[k,l] della Gamma
35          akl <- apply(kl,1,function (y)
36          sum(apply(phiaij2,1,function(x)
37          A[x[1],x[2]]*Z0[x[1],
38          y[1]]*Z0[x[2],y[2]])) )
39
40          bkl <- apply(kl,1,function (y)
41          sum(apply(phiaij2,1,function(x)
42          Z0[x[1],y[1]]*Z0[x[2],y[2]])) )
43
44          akk <- apply(kk,1,function (y)
45          sum(apply(phiaij,1,function(x)
46          A[x[1],x[2]]*Z0[x[1],
47          y[1]]*Z0[x[2],y[2]])) )
48
49          bkk <- apply(kk,1,function (y)
50          sum(apply(phiaij,1,function(x)
51          Z0[x[1],y[1]]*Z0[x[2],y[2]])) )
52
53          a0 <- b0 <- matrix(0,nrow=K,ncol=K)
54          diag(a0) <- akk
55          diag(b0) <- bkk
56          a0[upper.tri(a0,diag=F)] <- akl
57          b0[upper.tri(b0,diag=F)] <- bkl
58
59          #Parametri di Gamma
60          sh1 <- a0[upper.tri(a0,diag=T)] + a
61          sh2 <- b0[upper.tri(b0,diag=T)] + b
62
63          #Parametri Burn-in
64          M0 <- 5000 #numero simulazioni prima parte burnin
65          T.vec <- seq(10*76,Ti,length=M0)
66          w.vec <- seq(1/n,1,length=M0)
67
68          #BURN-IN : PRIMA FASE (INIZIO)
69          #M0 = 5,000 simulazioni
70          for(i in 1:M0)
71                  {
```

```
72                theta <- as.vector(rdirichlet(1,nk+T.vec[i])) #theta
73                omega <- rgamma(length(sh1),shape=sh1*w.vec[i],rate=sh2*w.vec[i]) #omega
74
75                #matrice omega
76                OX <- matrix(0,nrow=K,ncol=K)
77                OX[upper.tri(OX,diag=TRUE)] <- omega
78                OX[lower.tri(OX)] <- t(OX)[lower.tri(OX)]
79
80                #inizio predizione gruppi (classificazione Z)
81                for(l in 1:n)
82                        {
83                        for(j in 1:K)
84                                {
85                                seque1 <- 1:n
86                                seque1 <- seque1[-l]
87                                om1 <- sapply(1:K, function(w)
88                                sum(sapply(seque1,function(y) A[l,y]*Z0[y,w])) )
89                                om2 <- sapply(1:K, function(w)
90                                sum(sapply(seque1,function(y) Z0[y,w])) )
91                                phi[l,j] <- theta[j]* prod( (OX[j,]^om1)*exp(-OX[j,]*om2) )
92                                }
93                                Z0[l,] <- as.vector(class.fun(1,which.max(phi[l,]),K))
94                        }
95                #fine predizione gruppi (classificazione Z)
96
97        #calcoli per aggiornamento parametri Gamma
98        akl <- apply(kl,1,function (y)
99        sum(apply(phiaij2,1,function(x)
100       A[x[1],x[2]]*Z0[x[1],
101       y[1]]*Z0[x[2],y[2]])) )
102
103       bkl <- apply(kl,1,function (y)
104       sum(apply(phiaij2,1,function(x)
105       Z0[x[1],y[1]]*Z0[x[2],y[2]])) )
106
107       akk <- apply(kk,1,function (y)
108       sum(apply(phiaij,1,function(x)
109       A[x[1],x[2]]*Z0[x[1],
110       y[1]]*Z0[x[2],y[2]])) )
111
112       bkk <- apply(kk,1,function (y)
113       sum(apply(phiaij,1,function(x)
114       Z0[x[1],y[1]]*Z0[x[2],y[2]])) )
115
116       a0 <- b0 <- matrix(0,nrow=K,ncol=K)
117       diag(a0) <- akk
118       diag(b0) <- bkk
119       a0[upper.tri(a0,diag=F)] <- akl
120       b0[upper.tri(b0,diag=F)] <- bkl
121
122       #calcolo per aggiornamento parametri Dirichlet
```

```
123         nk<-as.vector(table(factor(apply(Z0,2,sum),levels=1:K)))
124
125         #aggiornamento parametri distribuzioni
126         #a posteriori (Dirichlet e Poisson)
127         sh1 <- a0[upper.tri(a0,diag=T)] + a
128         sh2 <- b0[upper.tri(b0,diag=T)] + b
129
130         #stampa avanzamento burn-in
131         print(paste("burn-in step:",i,collapse=""))
132         }
133         #BURN-IN : PRIMA FASE (FINE)
134
135         #salvataggio informazioni fine burn-in
136         Z0 <- Z0
137         phi0 <- phi
138
139         #calcoli per aggiornamento parametri Gamma
140         akl <- apply(kl,1,function (y)
141         sum(apply(phiaij2,1,function(x)
142         A[x[1],x[2]]*Z0[x[1],
143         y[1]]*Z0[x[2],y[2]])) )
144
145         bkl <- apply(kl,1,function (y)
146         sum(apply(phiaij2,1,function(x)
147         Z0[x[1],y[1]]*Z0[x[2],y[2]])) )
148
149         akk <- apply(kk,1,function (y)
150         sum(apply(phiaij,1,function(x)
151         A[x[1],x[2]]*Z0[x[1],
152         y[1]]*Z0[x[2],y[2]])) )
153
154         bkk <- apply(kk,1,function (y)
155         sum(apply(phiaij,1,function(x)
156         Z0[x[1],y[1]]*Z0[x[2],y[2]])) )
157
158         #calcolo per aggiornamento parametri Dirichlet
159         nk<-as.vector(table(factor(apply(Z0,2,sum),levels=1:K)))
160
161         a0 <- b0 <- matrix(0,nrow=K,ncol=K)
162         diag(a0) <- akk
163         diag(b0) <- bkk
164         a0[upper.tri(a0,diag=F)] <- akl
165         b0[upper.tri(b0,diag=F)] <- bkl
166
167         #aggiornamento parametri distribuzioni
168         #a posteriori (Dirichlet e Gamma)
169         sh1 <- a0[upper.tri(a0,diag=T)] + a
170         sh2 <- b0[upper.tri(b0,diag=T)] + b
171         nk<-as.vector(table(factor(apply(Z0,2,sum),levels=1:K)))
172         alpha <- nk + Ti
173
```

```
174         #creazione matrici per il salvataggio delle informazioni
175         theta <- matrix(NA,nrow=Nsim+M0,ncol=length(alpha))
176         omega <- matrix(NA, nrow=Nsim+M0, ncol=length(sh1))
177         Z10 <- phi10 <- array(NA,c(n,K,Nsim+M0))
178 
179         #CONTINUO BURN-IN + SIMULAZIONE (INIZIO)
180         for(i in 1:(M0+Nsim))
181                 {
182                 theta[i,] <- as.vector(rdirichlet(1,alpha)) #theta
183                 omega[i,] <-  rgamma(dim(omega)[2],shape=sh1,rate=sh2) #omega
184 
185                 #matrice omega
186                 OX <- matrix(NA,nrow=K,ncol=K)
187                 OX[upper.tri(OX,diag=TRUE)] <- omega[i,]
188                 OX[lower.tri(OX)] <- t(OX)[lower.tri(OX)]
189 
190 
191         #inizio predizione gruppi (classificazione Z)
192         for(l in 1:n)
193                 {
194                 for(j in 1:K)
195                                 {
196                                 seque1 <- 1:n
197                                 seque1 <- seque1[-l]
198                                 om1 <- sapply(1:K, function(w) sum(sapply(seque1,function(y
    ) A[l,y]*Z0[y,w])) )
199                                 om2 <- sapply(1:K, function(w) sum(sapply(seque1,function(y
    ) Z0[y,w])) )
200                                 phi[l,j] <- theta[i,j]* prod( (OX[j,]^om1)*exp(-OX[j,]*om2)
     )
201                                 }
202                                 Z0[l,] <- as.vector(class.fun(1,which.max(phi[l,]),K))
203                 }
204 
205         #fine predizione gruppi (classificazione Z)
206         phi10[,,i] <- phi
207         Z10[,,i] <- Z0
208 
209         #calcoli per aggiornamento parametri Poisson
210         akl <- apply(kl,1,function (y)
211         sum(apply(phiaij2,1,function(x)
212         A[x[1],x[2]]*Z0[x[1],
213         y[1]]*Z0[x[2],y[2]])) )
214 
215         bkl <- apply(kl,1,function (y)
216         sum(apply(phiaij2,1,function(x)
217         Z0[x[1],y[1]]*Z0[x[2],y[2]])) )
218 
219         akk <- apply(kk,1,function (y)
220         sum(apply(phiaij,1,function(x)
221         A[x[1],x[2]]*Z0[x[1],
```

```
222         y[1]]*Z0[x[2],y[2]])) )
223
224         bkk <- apply(kk,1,function (y)
225         sum(apply(phiaij,1,function(x)
226         Z0[x[1],y[1]]*Z0[x[2],y[2]])) )
227
228         a0 <- b0 <- matrix(0,nrow=K,ncol=K)
229         diag(a0) <- akk
230         diag(b0) <- bkk
231         a0[upper.tri(a0,diag=F)] <- akl
232         b0[upper.tri(b0,diag=F)] <- bkl
233
234         #calcolo per aggiornamento parametri Dirichlet
235         nk<-as.vector(table(factor(apply(Z0,2,sum),levels=1:K)))
236
237         #aggiornamento parametri distribuzioni
238         #a posteriori (Dirichlet e Poisson)
239         sh1 <- a0[upper.tri(a0,diag=T)] + a
240         sh2 <- b0[upper.tri(b0,diag=T)] + b
241         alpha <- nk + Ti
242
243         #stampa avanzamento simulazione
244         print(paste("algorithm step:",i,collapse=""))
245                 }
246         #CONTINUO BURN-IN + SIMULAZIONE (FINE)
247
248         return(list(init0=init,Z0=Z0,phi0=phi0,phi=phi10,Z10=Z10,theta=theta,omega=omega))
249 #Z0 e phi0 sono i risultati dopo le M0 iterazioni della prima fase di burn-in
250 #Z10 e phi sono i risultati sulle M0+Nsim simulazioni
251         }
252 #CODICE MODELLO (FINE)
```

Bibliografia e Sitografia

[1] Bioconductor. *Bioconductor. Open Source Software for Bioinformatics*. 2017. URL: http://bioconductor.org/.

[2] NCBI (National Center for Biotechnology Information). *CTH cystathionine gamma-lyase [Homo sapiens (human)]*. 2017. URL: https://www.ncbi.nlm.nih.gov/gene/1491.

[3] NCBI (National Center for Biotechnology Information). *F2 coagulation factor II, thrombin [Homo sapiens (human)]*. 2017. URL: https://www.ncbi.nlm.nih.gov/gene/2147.

[4] NCBI (National Center for Biotechnology Information). *HNF1B HNF1 homeobox B [Homo sapiens (human)]*. 2017. URL: https://www.ncbi.nlm.nih.gov/gene/6928.

[5] NCBI (National Center for Biotechnology Information). *PTH1R parathyroid hormone 1 receptor [Homo sapiens (human)]*. 2017. URL: https://www.ncbi.nlm.nih.gov/gene/5745.

[6] NCBI (National Center for Biotechnology Information). *PTHLH parathyroid hormone like hormone [Homo sapiens (human)]*. 2017. URL: https://www.ncbi.nlm.nih.gov/gene/5744.

[7] NCBI (National Center for Biotechnology Information). *RASD1 ras related dexamethasone induced 1 [Homo sapiens (human)]*. 2017. URL: https://www.ncbi.nlm.nih.gov/gene/51655.

[8] David M. Blei, Alp Kucukelbir e Jon D. McAuliffe. "Variational Inference: A Review for Statisticians". In: *Journal of the American Statistical Association* 112.518 (2017), pp. 859–877. DOI: http://dx.doi.org/10.1080/01621459.2017.1285773.

[9] AIRC (Associazione Italiana per la Ricerca sul Cancro). *Ciste/i*. 2017. URL: http://www.airc.it/cancro/glossario/c/cistei/.

[10] AIRC (Associazione Italiana per la Ricerca sul Cancro). *Tumore ovaie*. 2017. URL: http://www.airc.it/tumori/tumore-all-ovaio.asp.

[11] Gabor Csardi e Tamas Nepusz. "The igraph software package for complex network research". In: *InterJournal* Complex Systems (2006).

[12] Justin Guinney e Robert Castelo. *Gene Set Variation Analysis for microarray and RNA-seq data: R package*. 2013. URL: https://doi.org/doi:10.18129/B9.bioc.GSVA.

[13] Sonja Hänzelmann, Justin Guinney e Robert Castelo. "GSVA: gene set variation analysis for microarray and RNA-Seq data". In: *BMC Bioinformatics* 14 (2013), p. 7. DOI: https://doi.org/10.1186/1471-2105-14-7.

[14] William H. Kruskal e W. Allen Wallis. "Use of Ranks in One-Criterion Variance Analysis". In: *Journal of the American Statistical Association* 47.260 (1952), pp. 583–621. DOI: 10.1080/01621459.1952.10483441.

[15] P. Latouche, E. Birmelé e C. Ambroise. "Variational Bayesian inference and complexity control for stochastic block models". In: *Statistical Modelling: An International Journal* 12.1 (2012), pp. 93–115. DOI: https://doi.org/10.1177/1471082X1001200105.

[16] Chengyu Liu e Sampsa Hautaniemi. *PerPAS (Personalized Pathway Alteration analysiS) : R package*. 2017.

[17] Chengyu Liu, Rainer Lehtonen e Sampsa Hautaniemi. "PerPAS: Topology-Based Single Sample Pathway Analysis Method". In: *IEEE/ACM Transactions on Computational Biology and Bioinformatics* PP.99 (2017), pp. 1–1. DOI: http://doi.org/10.1109/TCBB.2017.2679745.

[18] Ulrike von Luxburg. "A tutorial on spectral clustering". In: *Statistics and Computing* 17.4 (2007), pp. 395–416. DOI: https://doi.org/10.1007/s11222-007-9033-z.

[19] A. James O'Malley e Peter V. Marsden. "The analysis of social networks". In: *Health Services and Outcomes Research Methodology* 8.4 (2008), pp. 222–269. DOI: https://dx.doi.org/10.1007%2Fs10742-008-0041-z.

[20] Gabriele Sales, Enrica Calura e Chiara Romualdi. *graphite: GRAPH Interaction from pathway Topological Environment*. 2017. DOI: https://doi.org/doi:10.18129/B9.bioc.graphite.

[21] Fabrizio Serra, Chiara Romualdi e Federico Fogolari. "Similarity Measures Based on the Overlap of Ranked Genes Are Effective for Comparison and Classification of Microarray Data". In: *Journal of Computational Biology* 23.7 (2016), pp. 1–12. DOI: https://doi.org/10.1089/cmb.2015.0057.

[22] Jeremy D. Silver, Matthew E. Ritchie e Gordon K. Smyth. "Microarray background correction: maximum likelihood estimation for the normal–exponential convolution". In: *Biostatistics* 10.2 (2008), pp. 352–363. DOI: https://doi.org/10.1093/biostatistics/kxn042.

[23] Tom A. B. Snijders e Krzysztof Nowicki. "Estimation and Prediction for Stochastic Blockmodels for Graphs with Latent Block Structure". In: *Journal of Classification* 14.1 (1997), pp. 75–100. DOI: https://doi.org/10.1007/s003579900004.

[24] Tom A. B. Snijders e Krzysztof Nowicki. "Estimation and Prediction for Stochastic Blockstructures". In: *Journal of the American Statistical Association* 96.455 (2001), pp. 1077–1087.

[25] Fattaneh A. Tavassoli e Peter Devilee. *World Health Organization Classification of Tumours. Pathology and Genetics. Tumours of the Breast and Female Genital Organs (Chapter 2 : Tumours of the Ovary and Peritoneum)*. Lyon: IARC Press, 2003.

[26] R Core Team. *R: A Language and Environment for Statistical Computing*. 2017. URL: https://www.r-project.org/.

[27] Hadley Wickham. *ggplot2: Elegant Graphics for Data Analysis*. New York: Springer-Verlag, 2009. ISBN: 978-0-387-98140-6.

[28] Wikipedia. *Nephron*. 2017. URL: https://en.wikipedia.org/wiki/Nephron.

[29] Wikipedia. *Scale-free network*. 2017. URL: https://en.wikipedia.org/wiki/Scale-free_network.

[30] Wikipedia. *Stochastic block model*. 2017. DOI: https://en.wikipedia.org/wiki/Stochastic_block_model.