UNIVERSITA' DEGLI STUDI DI PADOVA

FACOLTA' DI SCIENZE STATISTICHE

CORSO DI LAUREA SPECIALISTICA IN STATISTICA INFORMATICA

TESI DI LAUREA

PATTERN DI ESPRESSIONE GENICA ATTRAVERSO METODI BAYESIANI EMPIRICI PARAMETRICI.

Relatrice: Chiar.ma Prof.ssa
Monica Chiogna

Laureanda
Laura Gavagnin

ANNO ACCADEMICO 2006/07

Alla mia famiglia

Indice

Introduzione

E' ormai appurato da tempo che gran parte delle malattie derivano da alterazioni del codice genetico. La tecnologia del DNA *microarray* si è rilevato un ottimo mezzo per cinque principali obiettivi biologici:

1. l'identificazione di geni con livelli di espressione diversa sotto diverse condizioni sperimentali o tra soggetti che presentano varie forme della stessa patologia;
2. l'identificazione di gruppi di geni che con buona probabilità sono correlati tra loro;
3. la caratterizzazione gnomica della cellula malata attraverso la classificazione di campi biologici (soggetti sani vs soggetti affetti da una determinata patologia);
4. l'identificazione di geni il cui valore di espressione è biologicamente utile per determinare un particolare gruppo o fenotipo (tali geni sono detti marcatori);
5. l'identificazione di nuove classi di una specifica patologia (come nel caso delle patologie oncologiche: esisto molte classi di tumori diversi).

L'identificazione di mutazioni è fondamentale per la prevenzione delle malattie genetiche, per la diagnostica precoce dei tumori, nonché in microbiologia per la identificazione di ceppi batterici o virali. Un altro settore di applicazione è quello dell'analisi funzionale simultanea di decine di migliaia di geni e, in futuro prossimo, di tutti i geni che costituiscono il nostro patrimonio genetico. Inoltre i risultati che ci si aspetta di ottenere con questa nuova tecnologia saranno fondamentali per sviluppare nuovi farmaci, e per meglio utilizzare quelli attualmente disponibili dando al medico la possibilità di adattare la terapia sulla base delle caratteristiche genetiche di ognuno di noi.

Il livello di espressione genica contiene la chiave per affrontare problemi legati alla prevenzione e alla cura di alcune malattie, per comprendere i

meccanismi di evoluzione biologica e per scoprire adeguati trattamenti farmacologici. Il recente avvento della tecnologia del DNA *microarray* ha permesso di manipolare simultaneamente migliaia di geni, motivando lo sviluppo della classificazione di tumore con l'utilizzo dei dati d'espressione genica.

Nel presente elaborato si considera una particolare forma di leucemia: la leucemia linfatica cronica a B-cellule (B-CLL). La Leucemia Linfatica Cronica a cellule B (LLC-B) è la forma di leucemia più frequente nella popolazione adulta. Viene diagnosticata generalmente in età media o avanzata (età mediana alla diagnosi: 65 anni). Solo il 15% dei pazienti ha un'età inferiore a 50 anni. Negli ultimi anni tuttavia la percentuale di pazienti giovani è in aumento, probabilmente perché un numero maggiore di casi viene diagnosticato a seguito di esami occasionali, in assenza di qualunque sintomo. Questi soggetti hanno spesso come unica alterazione all'esame emocromocitometrico un aumento del numero di globuli bianchi (leucocitosi) con aumento percentuale dei linfociti (linfocitosi). In questi pazienti la diagnosi di certezza di Leucemia Linfatica Cronica può essere fatta agevolmente dallo specialista ematologo mediante la tipizzazione immunologica dei linfociti del sangue periferico.

La Leucemia Linfatica Cronica è una malattia clonale caratterizzata dalla proliferazione e dal progressivo accumulo di linfociti B nel sangue, nel midollo, nei linfonodi, nella milza. Il risultato è l'aumento del numero dei globuli bianchi (leucocitosi), l'aumento delle dimensioni delle ghiandole linfatiche (linfoadenomegalia), l'aumento delle dimensioni della milza (splenomegalia). Con il progredire della malattia possono comparire altri sintomi legati all' insufficienza midollare quali:anemia (riduzione del numero di globuli rossi o eritrociti), piastrinopenia (riduzione del numero delle piastrine), neutropenia (riduzione del numero dei granulociti neutrofili). L'anemia causa stanchezza e pallore cutaneo, la piastrinopenia causa manifestazioni emorragiche, la neutropenia aggrava il rischio infettivo già intrinseco alla malattia. Possono inoltre manifestarsi disordini autoimmuni quali anemia emolitica autoimmune e piastrinopenia autoimmune. Dal punto di vista clinico e della sopravvivenza la Leucemia Linfatica Cronica si

comporta in modo eterogeneo. Vi sono infatti pazienti asintomatici in cui le alterazioni ematologiche rimangono stabili per anni senza alcuna terapia ed hanno sopravvivenza non diversa da quella attesa per l'età, e pazienti che hanno invece una malattia progressiva con sopravvivenze inferiori a 3 anni. Come vedremo più avanti, studi recenti hanno dimostrato che questa variabilità clinica dipende da differenze biologiche della malattia. La definizione alla diagnosi delle caratteristiche biologiche della malattia è quindi oggi un momento importante ai fini delle successive decisioni terapeutiche.

Gli stadi avanzati (in cui gli anni di sopravvivenza media sono dai 6 ai 2), comprendono il 40-45% dei casi e comportano una significativa riduzione dell'aspettativa di vita. Negli stadi iniziali (in cui gli anni di sopravvivenza media sono oltre i 9) la sopravvivenza è nettamente migliore, ma si osserva una notevole eterogeneità di decorso. Infatti, una quota di pazienti ha una aspettativa di vita non compromessa dalla malattia, mentre una quota pari al 40% progredisce entro 2 anni ed ha una sopravvivenza globale significativamente ridotta rispetto a quanto atteso per l'età. Vi sono però altri parametri che consentano di predire più accuratamente la prognosi individuale di un paziente in stadio iniziale. I più significativi sono:

- tempo di raddoppio dei linfociti inferiore a 6 mesi o aumento della linfocitosi superiore al 50% in meno di due mesi;
- valori elevati di beta2-microglobulina e di LDH (indici di massa di malattia e di rapida proliferazione cellulare);
- livelli sierici elevati di timidin-kinasi (TK) e dell'antigene CD23 solubile;
- un assetto immunofenotipico non tipico;
- una morfologia dei linfociti "variante";
- un infiltrato linfatico di tipo diffuso alla biopsia osteomidollare.

Più recentemente sono stati individuati nuovi parametri biologici prognosticamente significativi, indipendenti dai parametri clinici convenzionali sopra citati:

- Lo stato mutazionale dei geni IgVH (regione variabile delle catene pesanti delle immunoglobuline). In base allo stato mutazionale si distinguono oggi due sottotipi di Leucemia Linfatica Cronica B: una frazione di casi (50% circa) con IgVH in stato non mutato, cioè senza mutazioni somatiche, ed una frazione con mutazioni somatiche (mutati). La situazione non mutata si associa ad una malattia più estesa (stadio più avanzato) e comporta una prognosi più sfavorevole. L'impatto prognostico negativo dello stato non mutato è evidente anche nei pazienti in stadio clinico iniziale. Inoltre, lo stato non mutato si associa più frequentemente ad alterazioni cromosomiche sfavorevoli. Al momento attuale lo stato mutazionale dei geni IgVH appare il più potente fattore prognostico.
- L'espressione della proteina ZAP70 da parte dei linfociti. I linfociti B normali non esprimono questa proteina. All'analisi in citofluorimetria a flusso circa il 40% dei pazienti con Leucemia Linfatica Cronica esprime invece ZAP70. L'espressione correla con lo stato IgVH non mutato e dal punto di vista prognostico ha un significato sfavorevole.
- L'espressione dell'antigene CD38 da parte dei linfociti. In presenza di una percentuale di linfociti CD38-positivi superiore al 30% l'andamento clinico è sfavorevole.
- La presenza di anomalie citogenetiche all'analisi citogenetica e alla FISH. La trisomia 12(16% dei casi), la delezione 17p13 (17%) e la delezione 11q23 (18%) hanno significato prognostico sfavorevole. La delezione 13q14 (55%) è invece favorevole se isolata (sopravvivenza simile a quella dei pazienti con cariotipo normale). Le alterazioni sfavorevoli si riscontrano più spesso in pazienti in stadio avanzato, ma anche in una certa quota di pazienti in stadio A (15% circa).

L'elaborato considera proprio uno di questi parametri biologici, ossia lo stato mutazionale del gene V_H3-21. Lo studio infatti inizialmente consisteva nell'analisi di 29417 *patterrns* di espressione in 65 pazienti affetti da Leucemia Linfatica Cronica B con lo scopo di evidenziare geni co-regolati con lo stato del gene in

studio: 13 pasienti con Ig mutato e 52 pazienti con Ig non mutato. La dimensione è stata poi ridotta per problemi computazioneli considerando solamente 10000 *patterns* di espressione.

Per le analisi sono stati utilizzati i metodi bayesiani empirici parametrici descritti nel Capitolo 2. Nel Capitolo 3 sono state fatte le simulazioni necessarie a valutare l'influenza del modello parametrico scelto a priori sull'individuazione di geni differenzialmente espressi, considerando che i due sottogruppi possono avere dimensioni fortemente sbilanciate.

Le prime analisi sulla casistica reale si basano sostanzialmente su una caratteristica tipica di dati raccolti con *microarray*: la presenza di un grosso numero di osservazioni mananti. Sono stati quindi valutate nel Capitolo 4 le quattro diverse tecniche di imputazione (eliminando il 10% delle osservazioni in maniera casuale) prima confrontando l'errore RMS per i vari metodi, poi con riferimento ai metodi empirici bayesiani per il controllo della differenza di espressione dei geni.

Infine, nel Capitolo 5 sono stati applicati i metodi bayesiani empirici nel *dataset* ridotto per l'identificazione di geni differenzialmente espressi nei due sottogruppi

INTRODUZIONE

Capitolo 1

La biologia molecolare e i *microarray.*

1.1 La Cellula

Introduzione

La cellula è la più piccola unità dell'organismo in grado di funzionare in modo autonomo. Tutti i viventi sono costituiti da una o più cellule: in base a questa caratteristica, possono essere suddivisi, rispettivamente, in organismi unicellulari e pluricellulari. Al primo gruppo appartengono, ad esempio, archebatteri, eubatteri, alghe azzurre; il secondo comprende le piante, gli animali e i funghi pluricellulari. Tutte le cellule sono accumunate da "orfanelli" (particolari organuli) e strutture tra cui la membrana esterna, il citoplasma e la molecola di DNA, che contiene il codice genetico dell'organismo a cui la cellula appartiene.
Negli organismi pluricellulari le cellule si coordinano e formano livelli di organizzazione superiori: i tessuti, caratterizzati da cellule specializzate a svolgere determinate funzioni; gli organi, composti da più tessuti, che effettuano anch'essi specifiche funzioni; gli apparati (o sistemi), nei quali diversi organi interagiscono per il compimento di funzioni superiori; infine, l'organismo. Ogni elemento di un livello è dotato di capacità che l'elemento al livello inferiore non possiede. Così una singola cellula nervosa è capace di trasmettere impulsi nervosi a un'altra cellula, ma non è in grado di elaborare pensieri. Strutture come i virus e i prioni non vengono considerati viventi perché mancano di una organizzazione cellulare.

CAPITOLO 1

I biologi studiano le cellule per comprendere le modalità con cui esse si formano a partire dalle molecole, e per chiarire i meccanismi con i quali poi, una volta formate, esse cooperano alla costruzione di organismi complessi come gli esseri umani. La conoscenza delle cellule è alla base, dunque, della comprensione dei processi fisiologici, delle modalità di sviluppo e dei fenomeni di invecchiamento dell'organismo; in tal modo, essa diventa di importanza fondamentale per chiarire come si instaurano i processi patologici.

1.1.1 La teoria cellulare

Le cellule furono osservate per la prima volta nel 1665 da Robert Hooke, che studiò con un microscopio rudimentale sottili fettine di sughero e vide che esse erano formate da elementi di forma regolare. Egli chiamò cellule questi elementi (dal latino cellula "piccola stanza"), perché esse avevano l'aspetto di piccole scatole.

Figura 1.1 A sinistra ritratto di Robert Hooke, a destra il microscopio usato nella prima scoperta della cellula.

Nel 1830 Theodor Schwann compì studi al microscopio sulla cartilagine di animali e vide che questa era formata da cellule simili a quelle delle piante, e ipotizzò che le cellule fossero gli elementi costitutivi fondamentali di piante e animali; analoghe conclusioni trasse nel 1839 Matthias Schleiden. Nel 1860 Rudolf Virchow affermò che le cellule dovevano essere le "unità vitali" di tutti gli organismi, e che ogni cellula deriva da un'altra cellula.

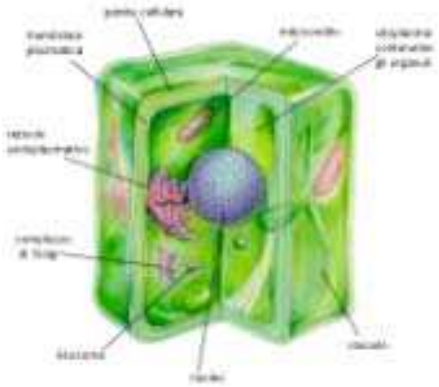
Figura 1.2 A partire da sinistra i ritratti di Theodor Schwann, Matthias Schleiden e Rudolf Virchow.

1.1.2 Le caratteristiche generali delle cellule

Le cellule possono avere dimensioni e forme molto diverse. Le cellule batteriche sono le più piccole, avendo una lunghezza dell'ordine di 1 μm (un milionesimo di metro). Le cellule dei tessuti animali hanno forma estremamente varia, a seconda del tipo e della funzione (possono essere sferiche, dai contorni irregolari, stellate, poliedriche, cubiche, cilindriche ecc.). Il diametro è compreso fra i 10 e i 20 μm e la superficie è spesso ricca di intro-ed estroflessioni. Le cellule nervose, ad esempio, hanno grossolanamente forma stellata, sono dotate di sottili

prolungamenti che possono raggiungere anche diversi metri di lunghezza (come avviene, ad esempio, nelle fibre nervose che innervano il collo delle giraffe).

(a)



(b)

Figura 1.3 (a) La cellula nervosa. (b) La cellula vegetale

Le cellule vegetali hanno solitamente forma poliedrica, con una lunghezza compresa tra i 20 e i 30 μm; la regolarità della loro forma è dovuta al fatto che esse possiedono, al contrario delle cellule animali, pareti cellulari rigide.

In tutti i viventi le cellule condividono alcune caratteristiche fondamentali. Tutte le cellule sono delimitate da una membrana (detta membrana plasmatica o plasmalemma) che racchiude il citoplasma. Questo è formato da una componente semifluida, il citosol, contenente acqua, sali minerali e molecole organiche, in cui si trovano immerse strutture dette organuli o organelli (nelle cellule eucarioti, vedi avanti), ciascuno preposto a una particolare funzione.

Le cellule sono la sede di reazioni chimiche che permettono loro di crescere, di produrre energia e di eliminare le scorie. Nel loro insieme, tutte queste reazioni sono denominate metabolismo (termine derivante da una parola greca che significa "cambiamento"). Le reazioni nella cellula avvengono in presenza di speciali catalizzatori, detti enzimi, costituiti da molecole proteiche.

Le informazioni necessarie allo svolgimento di tutte le attività metaboliche delle cellule e, in sostanza, le informazioni che rendono possibile la vita, sono

contenute negli acidi nucleici, presenti all'interno delle cellule stesse: l'acido desossiribonucleico (DNA) fa da stampo per la produzione di acido ribonucleico (RNA) il quale, interagendo con strutture proteiche dette ribosomi, determina la sintesi di molecole proteiche. In tal modo avviene la formazione degli enzimi che, a loro volta, permettono lo svolgimento di tutte le attività cellulari.

Le cellule sono capaci di riprodursi: ciascuna di esse si divide in due cellule figlie mediante un processo che prende il nome di mitosi. La capacità di dividersi delle cellule è differente in base al tipo cui esse appartengono. Si possono riconoscere tre categorie: cellule soggette al rinnovamento, che per tutta la vita dell'individuo vengono continuamente sostituite da cellule nuove (come avviene nella cute); cellule in espansione, che smettono di dividersi quando l'individuo ha completato la sua crescita, ma che possono occasionalmente riprendere a dividersi come conseguenza di ferite o traumi (come avviene nel fegato, nella tiroide, nel tessuto muscolare liscio); cellule statiche, che perdono la capacità di dividersi prima ancora che l'accrescimento dell'organismo sia completo (ad esempio, le cellule nervose). Alcune cellule nell'organismo mantengono la capacità di riprodursi per tutta la vita, e rimangono indifferenziate, potendo quindi dare luogo a diversi tipi cellulari: tali cellule sono dette staminali.

La chimica cellulare si basa prevalentemente sui composti del carbonio (detti composti organici) e quasi esclusivamente su reazioni chimiche che hanno luogo in soluzione acquosa, nello stretto intervallo di temperature normalmente riscontrabili sulla Terra.

I principali tipi di molecole organiche che compongono la cellula sono le proteine (formate dall'unione di molte subunità, dette amminoacidi), i carboidrati (sia zuccheri semplici, sia polisaccaridi, cioè lunghe catene di molecole di zuccheri), i grassi (tra i quali sono molto importanti i fosfolipidi, costituenti fondamentali della membrana plasmatica) e gli acidi nucleici (composti dall'unione di molti nucleotidi).

1.1.3 Procarioti ed eucarioti

Le cellule, in base alla loro organizzazione interna, possono essere distinte in due grandi categorie: cellule procarioti e cellule eucarioti. Il termine procariote deriva dal greco e significa “prima del nucleo”; il termine eucariote significa “vero nucleo”.

- Le cellule procaroiti

Le cellule procarioti sono tipiche degli archeobatteri, degli eubatteri e delle alghe azzurre. Esse sono relativamente piccole (con un diametro generalmente compreso fra 1 e 5 μm) e hanno una struttura interna alquanto semplice; il loro DNA si trova concentrato in una regione del citoplasma, senza essere delimitato da alcuna membrana. Sono prive di organuli, a eccezione dei ribosomi, preposti alla sintesi delle proteine. Le funzioni cellulari sono comunque effettuate da complessi enzimatici analoghi a quelli delle cellule eucarioti. Gli organismi formati da cellule procarioti sono detti procarioti

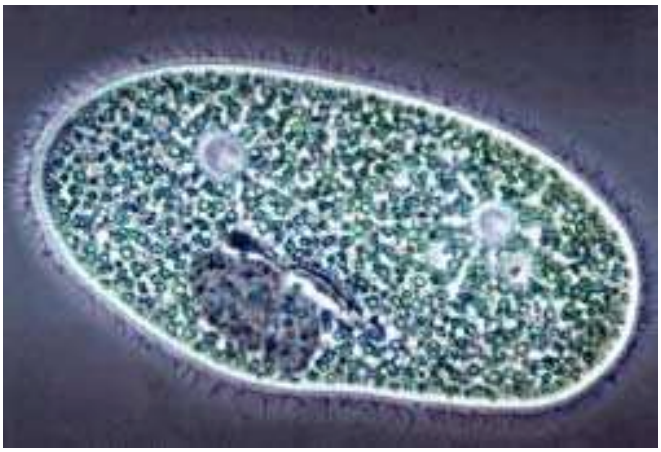


Figura 1.5 Immagine di una cellula procariota

- Le cellule eucarioti

Le cellule eucarioti costituiscono tutti gli altri organismi viventi (i protisti, le piante, i funghi e gli animali) sono molto più grandi (solitamente il loro asse maggiore è compreso fra i 10 e i 50 μm); in esse il DNA è racchiuso da una membrana, formando così un particolare organulo chiamato nucleo. Queste cellule possiedono organuli immersi nel citoplasma, ognuno deputato a svolgere una particolare funzione. Gli organismi formati da cellule eucarioti sono detti eucarioti.

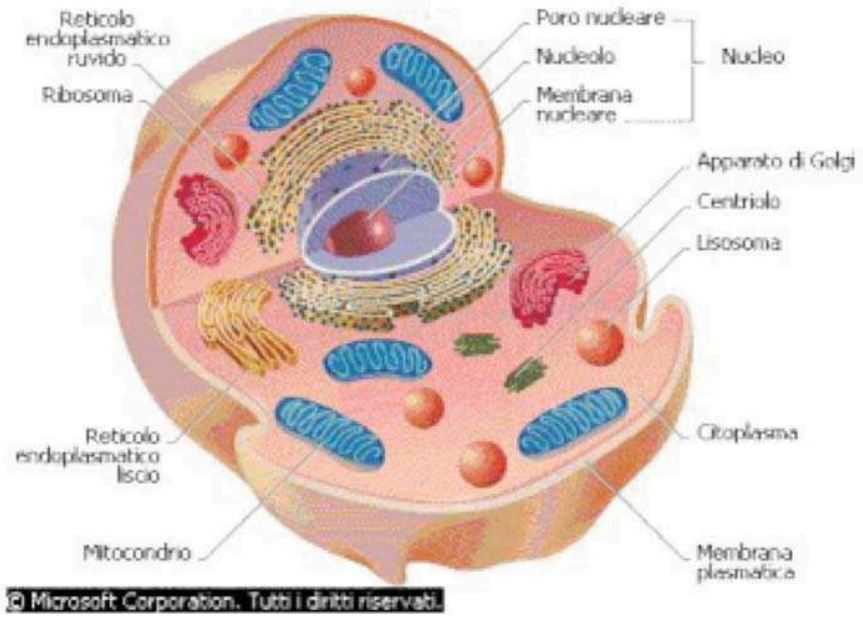


Figura 1.6 La cellula eucariote.

1.1.4 Le membrana plasmatica

La membrana plasmatica racchiude il contenuto della cellula e costituisce una barriera fra l'ambiente intracellulare (ambiente interno) e quello extracellulare (ambiente esterno). È costituita da un doppio strato continuo di molecole di fosfolipidi, dello spessore di 8-10 *nm* (un nanometro corrisponde a un miliardesimo di metro), attraversata parzialmente o completamente da numerose proteine. I fosfolipidi sono particolari grassi, formati da una "testa" di glicerolo legato a un gruppo fosfato, e da due "code" di acidi grassi.

La funzione di barriera svolta dalla membrana permette la regolazione della composizione chimica della cellula. La maggior parte degli ioni e delle molecole idrosolubili non è in grado di attraversare spontaneamente tale barriera, che è di natura lipidica; per farlo, necessita di una specifica proteina trasportatrice (detta carrier) o di una struttura, detta canale, formata da una grossa proteina infissa nello spessore della membrana e dotata di una cavità centrale. Avvalendosi di questi meccanismi di trasporto, la cellula può mantenere la concentrazione interna degli ioni e delle piccole molecole su valori diversi da quelli che caratterizzano l'ambiente esterno.

La membrana plasmatica può presentare estroflessioni a forma di dita, che prendono il nome di microvilli e hanno la funzione di aumentare la superficie di scambio tra la cellula e l'ambiente esterno. Esse sono tipiche, ad esempio, nelle cellule che rivestono la superficie dell'intestino, nelle quali il gran numero di microvilli (che formano il cosiddetto orletto a spazzola) garantisce una grande capacità di assorbimento delle sostanze nutritive.

La membrana rappresenta anche, oltre che un filtro per le sostanze in entrata e in uscita, il mezzo con cui la cellula si "fa riconoscere" dalle altre cellule. Essa contiene molecole particolari, di solito formate da zuccheri legati a proteine, che corrispondono a una sorta di "carta d'identità" in base alla quale la cellula viene

riconosciuta come facente parte del sé, ossia dell'organismo stesso, e non viene attaccata dal sistema immunitario, oppure come estranea (non sé) e come tale, da distruggere. L'insieme delle molecole che caratterizzano i diversi tipi di cellule e di tessuti dell'organismo viene chiamato complesso maggiore di istocompatibilità (MCH); esso è responsabile del fatto che i tessuti trapiantati agiscono da antigeni e vengono attaccati dall'organismo ricevente (fenomeno del rigetto). Pertanto, si sottopone il paziente a terapia immunodepressiva prima di un trapianto.

Nelle cellule animali, la membrana plasmatica non presenta generalmente strati esterni di rivestimento. Nei batteri e nei vegetali, invece, all'esterno della membrana si trova una parete rigida, alquanto spessa e robusta, costituita da polisaccaridi complessi (nel caso delle piante superiori, soprattutto da cellulosa). Tale struttura nei batteri ha soprattutto una funzione protettiva; nei vegetali, oltre a questa funzione, la parete svolge un ruolo di sostegno e serve a mantenere la forma tipica della cellula. La parete limita i movimenti della cellula, come pure l'ingresso e la fuoriuscita di materiali.

In un organismo pluricellulare, le cellule si collegano l'una all'altra mediante giunzioni intercellulari. Nelle piante superiori le cellule sono connesse mediante "ponti" di citoplasma (denominati plasmodesmi).

Nella maggior parte degli animali, le cellule sono legate fra loro mediante una rete a maglie relativamente larghe, costituita da grosse molecole organiche (la cosiddetta matrice extracellulare) e mediante punti di adesione fra le membrane plasmatiche (giunzioni cellulari).

1.1.5 Citoplasma e Citosol

L'intero volume della cellula, con esclusione del nucleo, è occupato dal citoplasma. Questo comprende una soluzione acquosa concentrata, denominata

citosol, nella quale si trovano sospesi enzimi e gli organuli cellulari. Nel citoplasma avvengono reazioni come quelle di glicolisi e quella di fermentazione, importanti per l'ottenimento di energia. In esso si trova anche un sistema di filamenti proteici, il *citoscheletro*, che è coinvolto con numerose funzioni, quali il sostegno della membrana cellulare e il movimento ameboide della cellula.

1.1.6 Organuli nella cellula eucariote

Nelle cellule eucarioti strutture membranose dette organuli sono deputate allo svolgimento di specifiche funzioni, e permettono una compartimentazione delle funzioni cellulari, proprietà che non si riscontra nelle cellule procarioti e che rende le eucarioti molto più efficienti.

1 Il Nucleo

L'organulo di maggiori dimensioni all'interno della maggior parte delle cellule vegetali e animali è il nucleo: è delimitato da una membrana e ha forma e dimensioni variabili a seconda del tipo cellulare. All'interno del nucleo si trovano il DNA, che costituisce il materiale genetico della cellula, e proteine (dette istoni) solitamente presenti in coppie, in un numero variabile e caratteristico di ciascuna specie.

2 Ciglia e flagelli

Molte cellule possiedono sulla superficie strutture flessibili, simili a "peli", denominate ciglia o flagelli, contenenti un fascio centrale di microtubuli che funziona da motore del movimento. Ciglia e flagelli si flettono dando luogo a un

battito regolare, simile a quello di una frusta, reso possibile dall'energia conservata sotto forma di molecole di adenosina trifosfato (ATP) all'interno dei microtubuli.

3 I mitocondri

I mitocondri costituiscono la sede del processo di respirazione cellulare, mediante il quale la cellula ricava energia (sotto forma di molecole di ATP) bruciando molecole di glucosio, derivanti dalla demolizione delle sostanze nutritive, in presenza di ossigeno.

4 I ribosomi

I ribosomi rappresentano la sede della sintesi delle proteine. Sono formati da due subunità di un particolare tipo di RNA (RNA ribosomiale) e possono essere associate alle membrane del reticolo endoplasmatico.

5 Reticolo endoplasmatico e apparato di Golgi

Una rete tridimensionale di sacche, dette cisterne, delimitate da membrane e tra loro comunicanti, costituisce il reticolo endoplasmatico, che rappresenta il compartimento cellulare dove avviene la sintesi di gran parte dei componenti delle membrane, e dei materiali destinati a essere esportati all'esterno della cellula.
Pile di cisterne appiattite, anch'esse delimitate da membrane, costituiscono, invece, l'apparato di Golgi, che riceve le molecole sintetizzate nel reticolo endoplasmatico, le elabora e le indirizza a diversi siti interni o esterni alla cellula.

6 Lisosomi, perossisomi e vacuoli

I lisosomi contengono enzimi responsabili della digestione di numerose molecole inutili o nocive per la cellula.
I perossisomi sono vescicole delimitate da membrana, che costituiscono un ambiente isolato e circoscritto per reazioni nel corso delle quali vengono generate e demolite forme particolarmente pericolose e reattive dei perossidi di idrogeno.
I vacuoli sono piccole cavità delimitate da una membrana, nelle quali vengono accumulate scorie del metabolismo cellulare.
Nella cellula vengono continuamente formate e distrutte piccole vescicole membranose, deputate al trasporto dei materiali da un organulo all'altro. In una tipica cellula animale, il complesso degli organuli delimitati da membrana può occupare fino a metà del volume totale della cellula. Fra il reticolo endoplasmatico, l'apparato di Golgi, i lisosomi, la membrana plasmatica e l'ambiente extracellulare esiste uno scambio continuo di sostanze, mediato da vescicole che si staccano dalla membrana di un organulo per fondersi con quella di un altro.

7 Organuli tipici della cellula vegetale

Le cellule vegetali possiedono alcune strutture tipiche: la parete, i plastidi e il vacuolo.
La parete costituisce uno strato rigido e robusto, posto all'esterno della membrana cellulare. I plastidi si possono considerare come sacche membranose, nelle quali la cellula può accumulare sostanze. I leucoplasti sono plastidi nei quali viene confinato l'amido di riserva, in attesa di utilizzazione; i cromoplasti sono plastidi nei quali si accumulano pigmenti detti carotenoidi, di colore rosso o giallo. I cloroplasti rappresentano la sede della fotosintesi clorofilliana, e contengono le molecole di clorofilla necessarie al processo. Un grosso vacuolo centrale, ossia una cavità delimitata da una membrana e piena di un liquido detto succo

vacuolare, costituisce per la cellula vegetale una sorta di idroscheletro, e svolge anche funzioni metaboliche.

1.2 Nozioni di biologia molecolare

1.2.1 Breve rassegna sulla scoperta del DNA

Ogni essere vivente possiede un programma genetico, cioè un insieme di istruzioni che specificano le sue caratteristiche e dirigono le sue attività metaboliche. Questo insieme di istruzioni costituisce l'informazione biologica, cioè è ereditaria ed è trasferita da una generazione all'altra attraverso la riproduzione. Le caratteristiche trasmesse sono dette caratteri ereditari.

L'informazione biologica è organizzata in unità fondamentali, dette geni, ciascuna delle quali interviene nella determinazione di un carattere ed è ereditata dai genitori.

Già con le prime ipotesi riguardanti l'evoluzione, si cercò di comprendere come i caratteri peculiari di un organismo venissero trasmessi e come le specie si evolvessero. Alcuni, come il noto scienziato Lamarck (1787), avevano ipotizzato che i caratteri acquisiti durante la vita fossero trasmissibili di padre in figlio: è il caso del famoso esempio della giraffa e del suo collo. Molte furono però le critiche rivolte a questa teoria, dovute dal fatto che molti caratteri acquisiti durante la vita non sono ereditari.

CAPITOLO 1

Il primo a dedicarsi con metodo scientifico allo studio dell'ereditarietà dei caratteri fu un abate austiaco Gregor Mendel, Bateson W. (1809). A quei tempi Mendel non aveva nessuna conoscenza della struttura intrinseca del DNA, tuttavia aveva intuito alcuni caratteri che ricomparivano regolarmente nelle popolazioni. Le regole fondamentali che mettevano in connessione questi eventi non erano ancora chiare. I primi esperimenti che Mendel condusse furono sul piante di piselli odorosi caratterizzate dalla capacità di effettuare l'autofecondazione e caratterizzate da cicli vitali non troppo lunghi.

Le ricerche di Mendel non furono prese immediatamente in considerazione, ma le baso della genetica erano comunque scoperte e senza avere idea di come fosse strutturato il DNA.

La prova decisiva che il depositario dell'informazione è il DNA fu fornita nel 1952 da A.D. Harshey e M.Chase, i quali dimostrarono che i batteriofagi, per introdurre nel batterio ospite il loro materiale ereditario, iniettano una molecola di DNA.

L'importante scoperta fatta sul DNA suscitò la curiosità degli scienziati sulla struttura di tale molecola. Agli inizi degli anni '50, un giovane scienziato americano, James Watson, si recò a Cambridge, in Inghilterra, con una borsa di studio per lavorare sui problemi di struttura molecolare e, al Cavendish Laboratory, incontrò il fisico Francis Crick. Entrambi si interessavano di DNA e ben presto cominciarono a lavorare insieme per cercare di capire come fosse strutturata tale molecola. Essi non eseguirono veri e propri esperimenti, ma intrapresero, piuttosto, un esame razionale dei dati allora noti sul DNA, cercando di organizzarli in modo logico. Le informazioni che essi avevano su tale molecola riguardavano le sue grosse dimensioni e la sua struttura lunga e filiforme formata da nucleotidi. Inoltre nel 1950 Linus Pauling

aveva dimostrato che le proteine sono spesso disposte in maniera elicoide e vengono mantenuti in questa disposizione da legami idrogeno idrogeno che si formano sulle spire adiacenti all'elica. Questa dimostrazione risultò utile ai fini della ricerca in quanto la molecola di DNA si comporta in modo simile alla molecola delle proteine.

Studi intrapresi da Maurice Wilkins e Rosalind Frankilng ai raggi X dimostrarono la forma a grande elica del DNA. Infine Chargaff verificò l'esattezza di due proporzioni che dimostravano l'impossibilità di legare chimicamente due basi purine (a due anelli) o due basi pirimidine (ad una unico anello), e quindi all'assunzione che la timina si può legare solamente alla adenina, e la citosina solamente alla guanina.

L'insieme di tutte queste scoperte ha condotto alla formulazione della struttura del DNA e alla chiarificazione dello due funzioni.

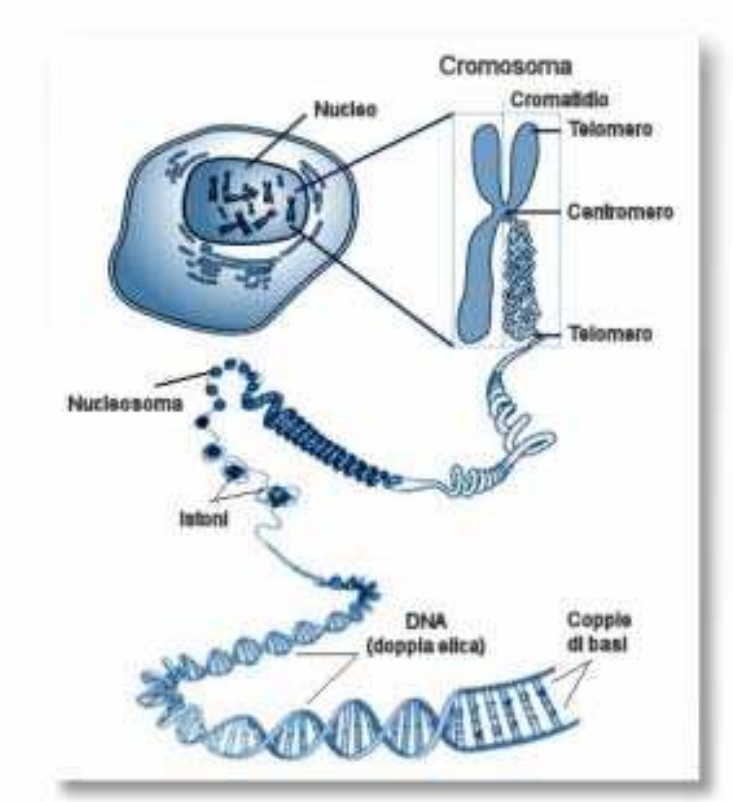
1.2.2 Il DNA

Il DNA è entrato nell'immaginario collettivo sotto forma di una lunghissima scala a chiocciola, la conformazione dedotta grazie agli studi di cristollagra_a di *Watson*, *Crick*, *Wilkins* e *Franklin*.

Altra immagine comune e familiare è quella del cromosoma, una X un po' paffuta sul quale è possibile individuare la posizione dei geni.

CAPITOLO 1

L'osservazione diretta della molecola attraverso un microscopio ottico ci offre ben altro scenario: un ammasso filamentoso colorabile con determinate sostanze, genericamente chiamato cromatina.



E' possibile osservare i cromosomi a forma di X al microscopio ottico, ma solo in una fase molto ristretta della vita di una cellula, quando inizia il processo di divisione, al termine del quale da una cellula se ne ottengono due (proliferazione cellulare).

In realtà quelle cose che chiamiamo cromosomi o cromatina non sono composti da solo DNA. Sono presenti, in termini di massa, oltre il doppio di proteine rispetto al DNA e un buon 10% di un'altra molecola particolare, l'RNA.

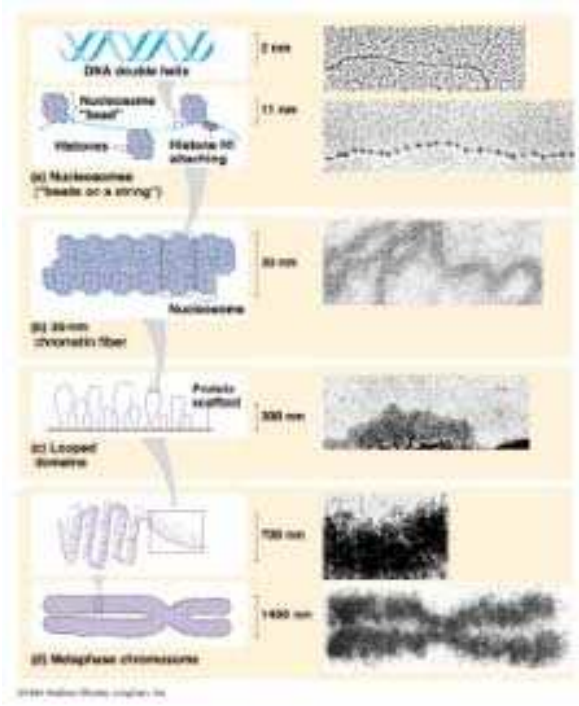
Il DNA custodisce all'interno del nucleo le informazioni per costruire i componenti della cellula, al sicuro dai danni ossidativi e di altra natura che avvengono nel resto della cellula.

La sintesi dei nuovi materiali avviene, però proprio nel resto della cellula, nel citoplasma e in alcuni organelli. Il passaggio di informazioni dal nucleo al citoplasma avviene grazie ad una molecola che viene sintetizzata ad immagine del tratto di DNA contenente l'informazione d'interesse. La molecola in questione è l'RNA messaggero, che una volta sintetizzato esce dal nucleo e porta le

informazioni alle strutture dedicate alla sintesi delle nuove componenti. Quindi il 10% di RNA di cui abbiamo parlato è costituito da nuove molecole appena sintetizzate e in partenza per il citoplasma.

Le proteine sono molto importanti in questa sede, poiché mantengono accese o spente determinate sequenze di DNA, come se agissero premendo su freni e acceleratori. Inoltre, permettono ed eseguono la sintesi di molecole di RNA messaggero e la replicazioni del DNA quando la cellula deve dividersi.

Il nostro genoma è costituito da circa 3 miliardi di paia di *basi*, che disposte in fila una dietro l'altra costituiscono un filamento lungo circa 1,8 metri. In realtà è suddiviso in 46 filamenti separati, chiamati cromosomi. Questi sono raccolti all'interno del nucleo delle cellule, una struttura rivestita da una membrana della dimensione media di circa 6 micrometri di diametro (6 millesimi di mm).



Una lunga molecola in piccolissimo spazio: nel cromosoma umano più piccolo, lungo circa 2 micrometri, sono compressi 14 mm di DNA. Per stare in queste ridotte dimensioni il DNA utilizza delle strutture proteiche attorno le quali si arrotola, un po' come del filo intorno a dei rocchetti.

Ogni struttura è formata da quattro coppie di rocchetti (proteine chiamate istoni) attorno a cui si arrotola il DNA e una specie di blocco che ferma il DNA all'ingresso e all'uscita dalla struttura (l'istone H), come un dito che ferma un nastro mentre facciamo un fiocco.

Queste strutture prendono il nome di nucleosomi, ognuna è grande 6x11 nm (1 nm è un milionesimo di mm) e su ognuna è arrotolato un filamento di DNA contenente circa 200 paia di *basi*. Tre stringe di nucleosomi, ciascuna di circa 10 nm, si avvolgono una sull'altra in una corda spessa circa 30 nm. Questa struttura si compatta ulteriormente durante la formazione dei cromosomi forma di X prima della duplicazione cellulare. I meccanismi esatti con cui avviene questa condensazione sono tutt'ora un mistero oggetto di studio.

L'RNA polimerasi ha una dimensione di circa 13x14 nm, poco più di un nucleosoma.

Il trasferimento delle informazioni dal nucleo al citoplasma avviene grazie a delle proteine (RNA polimerasi) che copiano il DNA in RNA messaggero.

Questo, una volta sintetizzato viene trasferito nel citoplasma dove viene tradotto, mediante appositi decodificatori, in proteine. Normalmente il DNA non è esposto all'azione delle RNA polimerasi, non è come un libro già aperto su una pagina sempre accessibile. Tutt'altro, non solo è come un libro chiuso, ma anche sigillato con un lucchetto. Solo quando la cellula ha bisogno di una determinata proteina viene inviata nel nucleo una proteina particolare che funziona come una chiave che apre il lucchetto.

Intervengono poi altre proteine a districare la matassa di DNA (come se cercassero di aprire il libro) per esporre il tratto d'interesse. Solo a questo punto le RNA polimerasi possono iniziare a leggere il DNA e a copiarlo in RNA messaggero.

Per questo motivo il DNA all'interno del nucleo può essere presente in due diverse modalità, una meno compatta disponibile alla lettura, definita per il suo aspetto omogeneo eucromatina, e una avvolta in strutture dense che impediscono l'accesso alle RNA polimerasi, l'eterocromatina.

Per usare un altro esempio, possiamo dire che l'eucromatina è come una cartella aperta che contiene diversi file direttamente apribili, l'eterocromatina è invece come una cartella zippata: bisogna prima de-zipparla per poter accedere alla lettura dei file.

1.3 Il codice genetico

La scoperta della struttura a doppia elica nel 1953 ha fatto immediatamente sorgere una do domanda: come l'informazione genetica può essere codificata dal DNA?

Era noto che il DNA è costituito da solo quatto tipi diversi di mattoncini, le basi *adenina*, *timina*, *guanina*, *citosina* e che nella doppia elica ad una *adenina* è sempre affacciata una *timina*, ad una *guanina* una *citosina*. In seguito si scoprì che per ogni gene viene sintetizzato, all'occorrenza, un filamento complementare di RNA, una molecola molto simile al DNA ma in grado di esistere come singolo filamento e che al posto della *timina* possiede un'altra basa, *l'uracile*.

Si sapeva che in qualche modo questa molecola di RNA veniva letta nella cellula per costruire le proteine. Queste, però, sono costituite da ben venti mattoncini diversi, gli aminoacidi.

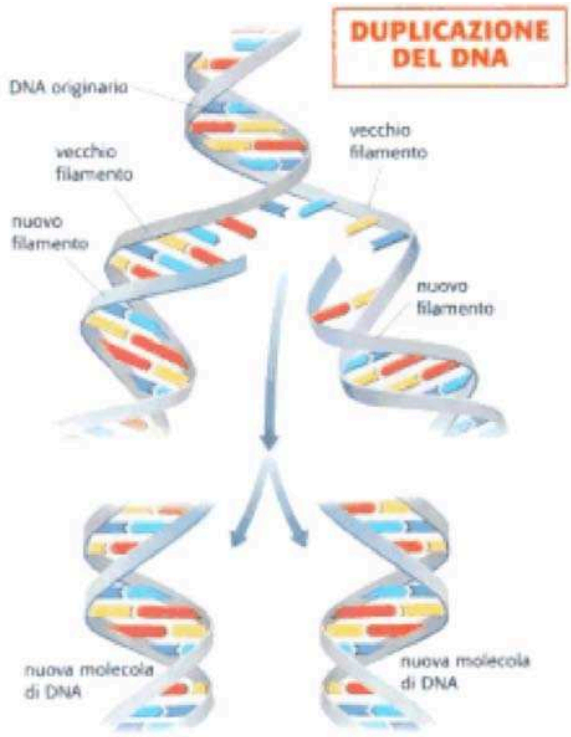
Presto il trucco venne svelato: il genoma è scritto in un linguaggio composta da parole di tre lettere, le triplette o codoni, ognuna indica un aminoacido. I conti però non tornavano: con 4 diverse lettere si possono formare ben 64 parole diverse, mentre gli aminoacidi sono solo venti. Si scoprì, che proprio come nella nostra lingua, anche il genoma aveva dei sinonimi: alcune parole indicavano lo stesso aminoacido, il quale poteva essere specificato anche con tre diverse triplette.

Inoltre, gli scienziati osservarono che determinate triplette, per la precisione tre, provocavano il termine della costruzione delle proteine e non indicavano l'aggiunta per nessun aminoacido, vennero pertanto chiamati codoni di stop. La lettura di tutte le lettere del nostro menoma ha tenuto occupati diversi scienziati coinvolti nel *Progetto Genoma Umano*, ed ora è in atto la traduzione ed interpretazione, un processo che si preannuncia più lungo e difficile ma

indubbiamente con implicazioni di enorme importanza per lo sviluppo della medicina e la comprensione della nostra biologia.

Abbiamo visto che le informazioni possono passare dal DNA, posto nel nucleo, al citoplasma dove vengono trasformate in istruzioni per creare le proteine. Questo processo avviene in tutte le cellule, di qualsiasi organismo dotato di nucleo da, ormai, miliardi di anni.

Infatti, le sequenze di DNA presenti nelle nostre cellule lo erano già nei nostri avi primitivi, e molte di queste sono ancora più antiche: le ritroviamo praticamente invariate in organismi che si sono evoluti prima della comparsa dei vertebrati sulla faccia della terra.



Il DNA, quindi, viene tramandato di generazione in generazione, in maniera altamente fedele, talvolta con piccole variazioni che, raramente ma con importanti risvolti, hanno permesso l'evoluzione delle specie. Infatti, una delle caratteristiche intrinseche della vita, quella di potersi riprodurre, è presente ed è dovuta proprio al DNA.

L'unità fondamentale della vita è la cellula, in grado di dare origine a molte copie di se stessa attraverso la riproduzione seriale di un processo noto con il nome di divisione cellulare. Prima di ogni divisione, devono essere effettuate nuove copie di ciascuna delle molecole che compongono la cellula, DNA compreso.

Esistono dei meccanismi molto sofisticati che coinvolgono un elevato numero di componenti cellulari e permettono di copiare in maniera fedele il DNA e generare

cellule che possiedono cloni identici di DNA. Questo processo è chiamato replicazione del DNA, può essere definito come il processo che rende in grado l'informazione genetica di un organismo di essere trasferita alle cellule figlie create durante la duplicazione cellulare. Quando le cellule si riproducono durante la crescita i un organismo o per rimpiazzare quelle morte, la duplicazione del DNA avviene in maniera molto fedele e secondo il meccanismo della replicazione. Quando, invece, vengono generate le cellule gameti (nell'uomo lo spermatozoo e l'ovulo), che fondendosi daranno origine ad un nuovo organismo, a questo processo se ne aggiunge uno, chiamato ricombinazione genica, che permette di mescolare un po' i geni dando origine a nuove combinazioni geniche e alla variabilità all'interno di una stessa specie.

Il modello proposto da *Watson* e *Crick* per la doppia elica di DNA descrive la presenza di due filamenti di DNA appaiati e complementari nella loro sequenza di basi nucleotidiche (cioè se in un filamento c'è la ...A... o la ...C... sull'altro c'è, rispettivamente la ...T... o la ...G...). Questo modello prevede in sé il meccanismo attraverso cui il DNA può essere duplicato: essendo i filamenti complementari, ciascuno dei due può funzionare da stampo per sintetizzare l'altro.

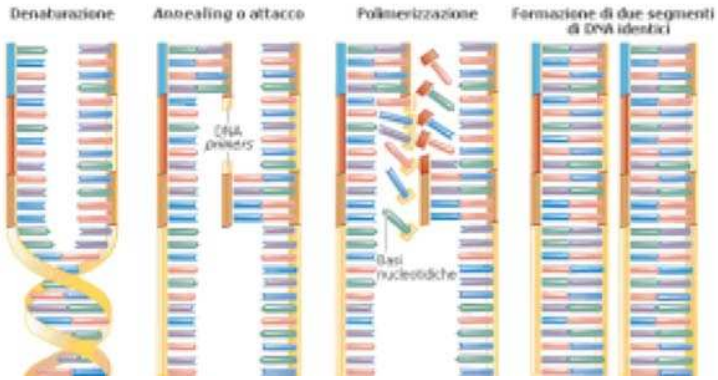
I primi esperimenti hanno dimostrato come scaldando una soluzione di DNA fino a quasi 100°C, questo andasse incontro ad un fenomeno chiamato denaturazione del DNA: in pratica, i due filamenti complementari di DNA, tenuti insieme solo da ponti idrogeno, si separano. Gli scienziati hanno poi scoperto che diverse molecole di DNA avevano una diversa temperatura di denaturazione, e che questa dipendeva dalla sequenza: più era alta la percentuale in C e G, più la temperatura doveva essere alta.

Questo dato confermava quanto previsto da *Watson* e *Crick*: le coppie TA sono tenute insieme da solo due ponti idrogeni, mentre le coppie CG ne hanno tre, quindi è naturale serva più energia per separarle. Inaspettatamente, gli scienziati hanno osservato che il DNA denaturato era in grado di rinaturarsi, cioè i filamenti separarsi erano in grado di riaccoppiarsi. Grazie a questa proprietà è stato possibile sviluppare alcune delle tecniche più importanti della biologia

molecolare: la clonazione del DNA, la PCR, il sequenziamento del DNA. Questi esperimenti, non solo hanno confermato il modello di *Watson* e *Crick*, ma hanno portato a scoprire le fasi primarie della replicazione del DNA, basate proprio sull'apertura della doppia elica, per permettere ad enzimi appositi di copiarla.

Le molecole di DNA presenti nelle nostre cellule hanno una dimensione notevole (ragionando in termini molecolari): contengono ciascuna centinaia di milioni di basi. Come può la cellula che si divide copiarle accuratamente e in relativamente breve tempo?

Due scoperte hanno permesso agli scienziati di rispondere a questa domanda. La prima riguardava lo studio della duplicazione del genoma di un batterio, che confermava il modello di apertura dei filamenti di DNA con successiva copiatura degli stessi. La seconda è relativa alla scoperta di un enzima chiamato DNA polimerasi, in grado di utilizzare un filamento di DNA per sintetizzarne il complementare. La DNA polimerasi legge il filamento e sceglie da una miscela di basi a sua disposizione quale inserire seguendo le leggi di complementarietà: aggiungerà una C ogni volta che trova una G, una T quando legge una A e viceversa. Le basi sono però un po' diverse da quelle inserite nella doppia elica del DNA, possiedono infatti un gruppo fosfato in più, per questo vengono definite *nucleotidi trifosfati*.



La DNA polimerasi per aggiungere il nucleotide trifosfato al filamento nascente di DNA ne rompe il legame con l'ultimo fosfato. Questo processo libera energia che l'enzima utilizza per legare il nucleotide a quelli già posizionati.

Affinché possa avvenire il processo di replicazione del DNA, questo deve essere svolto dalla conformazione usuale, per permettere l'accesso della polimerasi. Di

questo processo si occupa un altra proteina, la topoisomerasi, che esplica la propria azione srotolando il DNA.

Il DNA all'interno delle nostre cellule è sottoforma di lunghissimi filamenti: se la polimerasi iniziasse a copiare dall'inizio e procedesse fino al termine, l'intero processo sarebbe lunghissimo. Infatti, non è così che accade. In più punti di ogni singolo filamento di DNA si formano come degli occhielli ad opera degli enzimi elicasi: dei tratti di DNA, chiamati forcelle di replicazione, in cui i due filamenti antiparalleli si separano e all'interno dei quali la polimerasi inizia a copiare.

In questo modo più polimerasi copiano in contemporanea diversi tratti di DNA, rendendo l'intero processo estremamente rapido.

Nonostante le DNA polimerasi eseguano il loro lavoro ad una velocità straordinaria, da 500 a 1000 nucleotidi al secondo, sono molto accurate: sbagliano solamente una volta su un miliardo. Inizialmente si pensava che il DNA fosse una molecola altamente stabile, che non fossero possibili modificazioni di alcun tipo, data la sua importanza come pietra miliare delle informazioni genetiche.

Studi successivi rivelarono che il DNA era una molecola estremamente dinamica, continuamente sottoposta ad una miriade di tipi diversi di agenti in grado di danneggiarla. Contemporaneamente venne scoperto che le cellule possedevano dei meccanismi per riparare i danni subiti dal DNA.

Questi sistemi sono molto importanti nella vita quotidiana, sia per proteggere il DNA dagli agenti mutageni, sia per porre rimedio a quell'unica volta su un miliardo in cui le nostre DNA polimerasi sbagliano. A riprova di ciò, le persone che presentano questi sistemi difettivi sviluppano malattie molto importanti come lo *xeroderma pigmentoso* (che aumenta di 10000 volte il rischio di sviluppare il tumore alla pelle), il cancro al colon ereditario non poliposo e alcune forme di tumore al seno.

I primi studi risalgono agli anni Trenta, ad opera di due tedeschi, Karl Zimmer e Max Delbruck e di un russo Nokolai Timofèeff-Ressovsky e riguardavano lo studio del come agenti tipo radiazioni ionizzanti e raggi ultravioletti fossero in grado di indurre modificazioni ereditabili nel moscerino della frutta. A metà degli

anni Trenta diventò evidente come le cellule in vitro erano in grado di riparare i danni indotti da agenti mutageni. La scoperta dei meccanismi di riparazione dovette aspettare gli anni Quaranta, questa volta grazie a due studi indipendenti da parte di Albert Kelner, del gruppo di lavoro di Milislav Demerec al Cold Spring Harbor Laboratori e di Renato Dulbecco, del laboratorio di Salvador Luria alla University of Indiana. In realtà i loro studi erano indirizzati ad altre scoperte, ma durante alcuni esperimenti osservarono come le cellule, se illuminate con luce di determinate lunghezze d'onda (come quella solare), miglioravano la loro capacità di recuperare dai danni subiti in seguito all'esposizione alle radiazioni UV. Venne così scoperta la fotoriattivazione, grazie alla quale il DNA danneggiato dall'esposizione ai raggi UV viene riparato da un sistema composto da enzimi attivati dalla luce. Le radiazioni UV inducono la formazione di legami covalenti fra due timine quando queste si trovano affiancate, formando i dimeri di timina. Questa modifica impedisce la corretta funzione del DNA. Gli enzimi attivati dalla luce, quando colpiti da una luce di circa 300 nm, leggono il DNA finché non incrociano un dimero di timina e lo separano. Attualmente si conosco diversi meccanismi di riparazione, ciascuno interviene per riparare danni particolari. Per nominarne qualcuno: il sistema di riparazione mediante escissione di nucleotidi, mediante escissione di basi e di controllo degli accoppiamenti fra le basi. Tutti questi meccanismi sono in gradi di riconoscere l'errore, tagliare via il pezzo “guasto” e sostituirlo con uno nuovo e corretto. In tutti i casi si tratta di processi complessi, nel primo, ad esempio, sono coinvolte ben 30 diverse proteine.

Quando il cromosoma viene spezzato in due i radicali dell'ossigeno sono in grado di danneggiare in maniera molto grave il DNA, provocando delle rotture sui filamenti, talvolta spezzando in due un cromosoma. In questa situazione la cellula non è vitale ed è destinata a morire, a meno di non trasformarsi in cellula tumorale.

Interviene un sistema di riparazione molto particolare, la cui anomalia (un gene per la proteina BRCA1 non è funzionale) è coinvolta nell'insorgenza di buona parte dei tumori al seno ereditari.

In ogni cellula noi possediamo due cromosomi analoghi, che contengono gli stessi geni, anche se in varianti diverse. Il sistema che si occupa di aggiustare i cromosomi rotti lo fa confrontando quello spezzato con il suo ”fratello” sano e aggiungendo le parti mancanti. Non sempre questi sistemi funzionano alla perfezione, soprattutto quando i danni da riparare sono tanti, come quando ci sottoponiamo ad intense dosi di sostanze mutagene (ad es. sostanze radioattive). Quando non è possibile riparare tutti i danni le cellule vanno incontro a morte spontanea: siccome i danni subiti potrebbero aver causato delle modificazioni tali nella cellula da renderla pericolosa per il resto dell'organismo, questa si suicida.
La non funzionalità di questi sistemi, sia di riparo che di induzione al suicidio, predispone all'insorgenza del cancro: quando una mutazione porta all'attivazione di un gene che favorisce lo sviluppo del cancro questa non viene corretta, permettendo l'insorgenza del tumore. Questa elasticità del nostro genoma, di essere modificato, aggiustato e talvolta rimanere alterato, ha un significato molto importante per quello che è la vita sulla terra al giorno d'oggi: è, infatti, grazie ad essa che il genoma dei primi esseri viventi si è modificato, corretto, amplificato durante i secoli dando origine alle diverse specie e alle variabilità osservabili all'interno di queste.
Quindi il nostro DNA è continuamente in bilico in un delicato equilibrio: la sua dinamicità lo porta ad essere continuamente in sospeso tra un effetto dannoso e più frequente, lo sviluppo del cancro, e la nascita di un nuovo evento a favore della trasmissione della vita, molto raro, ma alla base dell'evoluzione delle specie.

1.4 I *microarray*

Il primo lavoro sui *microarray* è stato pubblicato nel 1995 da Mark Schena e collaboratori (Schena et al., 1995) dell’università di Stanford. L’idea ebbe origine

dalla necessità di studiare l'espressione genica delle piante attraverso la caratterizzazione dei loro fattori di trascrizione: la difficoltà dovuta all'assenza di adeguati strumenti di analisi fece avanzare la proposta di sviluppare degli appositi chip di vetro come dispositivi utili allo studio dei trascritti.

Il *Davis Laboratory* e il dipartimento di biochimica di Stanford realizzarono microscopici array (*microarray*) contenenti sequenze geniche di piante bloccate su un substrato di vetro; i *microarray* furono poi utilizzati per misurare l'espressione genica di tali piante in esperimenti di ibridazione con campioni di mRNA (RNA messaggero) marcati in fluorescenza. In condizioni sperimentali appropriate, i segnali fluorescenti sulla superficie del vetrino producono una misura dell'espressione di ogni gene rappresentato sul *microarray*: dalla quantificazione di tale fluorescenza è possibile risalire al livello di espressione di ciascun gene.

Il laboratorio di Stanford utilizzò tecniche fotolitografiche, *ink jetting* e *contact printing* per creare i microarray, mutuando tre approcci tradotti in realtà solo negli ultimi ventanni in ambiente microelettronico per la costruzione di circuiti microlavorati (MEMS): si può, quindi, comprendere il motivo di un tale divario temporale fra la comprensione del paradigma biologico alla base dei microarray e la loro effettiva realizzazione.

1.4.1 La tecnologia alla base dei *microarray*

La tecnologia dei *microarray* a DNA si basa sulla capacità di ibridizzazione degli acidi nucleici, secondo cui due filamenti di DNA ibridizzano tra di loro se sono complementari l'uno all'altro. Questa complementarità riflette la regola di Watson e Crick secondo la quale *l'adenina* si lega alla *timina* e la *citosina* si lega alla *guanina*. Uno o entrambi i filamenti di DNA ibridizzati possono essere sostituiti

con RNA che, pur differendo per la presenza *dell'uracile* al posto della *timina*, va incontro ugualmente al fenomeno dell'ibridizzazione.

L'ibridizzazione è stata per decenni utilizzata in biologia molecolare come principio base di metodiche quali il Southern blotting e il Northern blotting; i *microarray* a DNA sono una massiccia parallelizzazione di queste tecniche poiché sono in grado di analizzare migliaia di geni contemporaneamente.

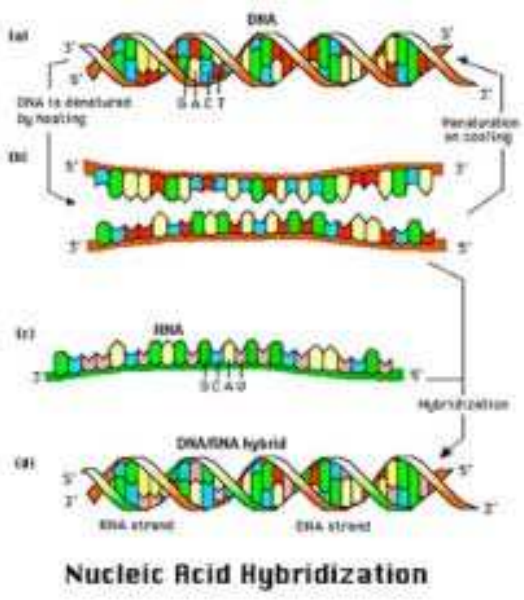
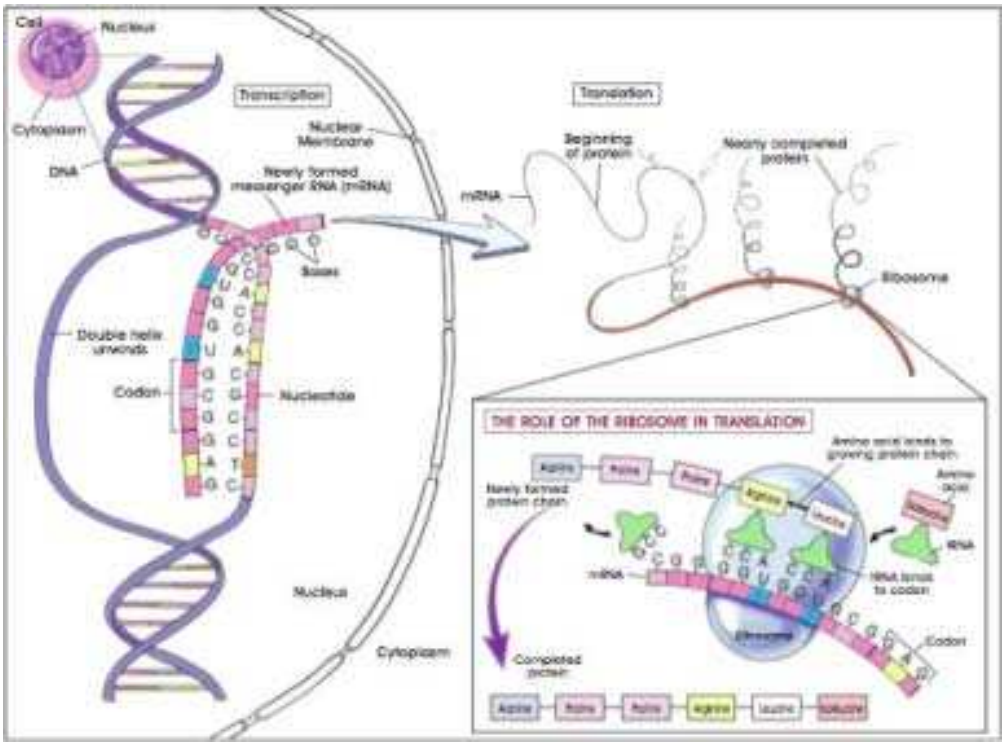


Figura 1.7 Ibridazione di DNA e RNA.

Nel caso dei *microarray* invece di distribuire le sonde oligonucleotidiche su un gel che contiene i campioni di RNA o DNA, esse vengono bloccate su una superficie di vetro. Sonde diverse possono essere posizionate alla distanza di qualche micron l'una dall'altra in modo da disporne un numero molto elevato in pochi centimetri quadrati. Il campione in studio viene marcato con fluorocromi e lasciato ibridizzare con le sonde presenti sul *microarray*. Dopo aver lavato l'eccesso di materiale non ibridizzato, i fluorocromi legati al campione ibridizzato vengono eccitati con un laser di opportuna lunghezza d'onda che scandisce la superficie del chip. Poiché la posizione delle sonde è individuabile grazie ad uno

schema a mappa cartesiana, è possibile quantificare l'ammontare di campione ibridizzato a partire all'immagine generata con lo scanner.

La concentrazione di un particolare mRNA è il risultato dell'espressione del gene da cui esso viene trascritto; per questo motivo le applicazioni che fanno uso di *microarray* a cDNA vengono spesso denominate *analisi dell'espressione genica.*



1.8 Processo di sintesi delle proteine.

Quando si vuole evidenziare la differente risposta di un gene alla sua esposizione a trattamenti diversi o osservare la sua espressione in momenti diversi si dice che si sta generando un profilo di espressione.

Un'altra applicazione tipica dei *microarray* è la rilevazione di polimorfismi in geni specifici: la peculiare struttura parallela dei *microarray* consente di rilevare simultaneamente numerosi polimorfismi genetici in più geni, permettendo in questo modo di fare una genotipizzazione.

Esistono diversi tipi di *microarray*, catalogati, a seconda del materiale che viene utilizzato come sonde, in:

- *Microarray* a cDNA, con sonde di lunghezza maggiore di 200 basi ottenute per retrotrascrizione da mRNA, frammentate, amplificate con PCR e depositate su un supporto di vetro o di nylon;
- *Microarray* ad oligonucleotidi, con sonde di lunghezza fra 25 e 80 basi ottenute da materiale biologico o per via artificiale e depositate su un supporto di vetro;
- *Microarray* ad oligonucleotidi, con sonde di lunghezza fra 25 e 30 basi sintetizzate in situ con tecniche fotolitografiche su wafer di silicio.

Per l'analisi dell'espressione sono presenti sul mercato due tecnologie dominanti: *Affymetrix*, *Inc. GeneChip* e quella degli *"spotted" array a cDNA*.

1.4.2 La tecnologia *Affymetrix GeneChip*

Affymetrix utilizza attrezzature simili a quelle che servono a realizzare i chip di silicio per i computer, che consentono di avere una produzione massiva di chip ad un costo ragionevole. Così come i chip per computer sono fatti utilizzando maschere che controllano il processo di deposizione e rimozione del silicio dalla superficie del chip, analogamente Affymertix usa maschere di controllo della sintesi degli oligonucleotidi sul *microarray*. Il risultato di questo processo è la produzione di alcune centinaia di migliaia di oligonucleotidi differenti, ciascuno dei quali presente in milioni di copie sul vetrino.

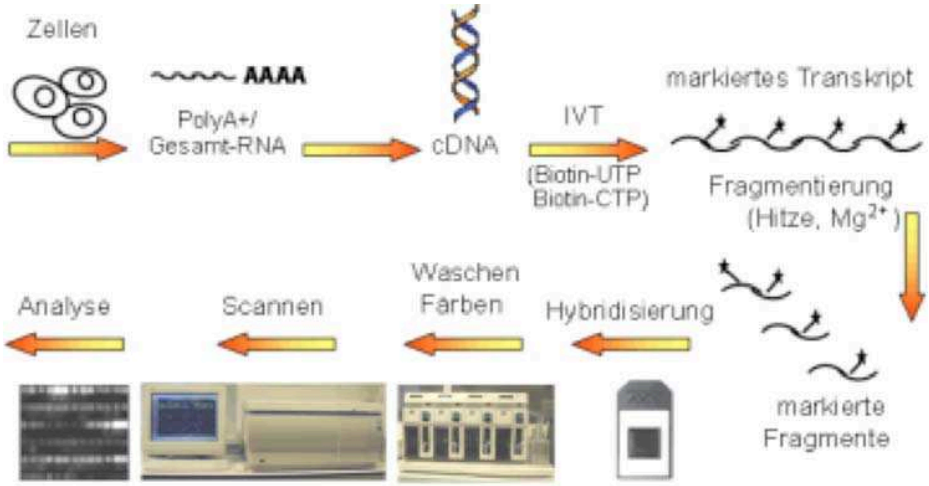


Figura 1.9 *Microarray Affymetrix*

Per le analisi di espressione sono utilizzati gruppi di sonde di almeno 40 oligonucleotidi per gene; *Affymetrix* ha selezionato, per ogni gene, una regione con la minor omologia con altri geni. A partire da questa regione vengono disegnati da 11 a 20 oligonucleotidi rappresentativi del *perfect match* (PM), cioè della perfetta complementarità con l'mRNA bersaglio, e 11-20 oligonucleotidi identici ai precedenti tranne che per il nucleotide centrale, utili per rilevare il miIsmatch (MM), cioè la non perfetta complementarità.

Affymetrix afferma che gli oligonucleotidi MM sono capaci di mettere in evidenza la presenza di segnali aspecifici permettendo di rilevare con maggiore sicurezza i segnali deboli.

L'ibridizzazione di ogni oligonucleotide con il proprio complementare dipende dalla sequenza specifica; poiché si è interessati alla misura del cambiamento di espressione di un gene è necessario ottenere un dato cumulativo da tutte le sonde che identificano quel gene. *Affymetrix* calcola questo dato cumulativo facendo una media della differenza fra sonde PM e MM dello stesso gene:

$$AvgDiff = \frac{\sum_N (PM - MM)}{N}$$

dove N è il numero di sequenze specifiche che identificano un gene. Se il numero che si ottiene da questo calcolo è negativo o molto piccolo significa che il cDNA bersaglio è assente o che si è verificata un'ibridizzazione non specifica.

Tutti gli algoritmi che riguardano la rilevazione di ibridizzazione sul chip, la generazione del dato cumulativo e la sua elaborazione sono protetti dalla tecnologia proprietaria *Affymetrix* che, per altro, si riserva di modificarli senza renderli noti.

Le fasi di un esperimento di analisi dell'espressione genica che fa uso di chip *Affymetrix* sono:

- Estrazione dell'RNA totale dal campione;
- Separazione dell'mRNA dall'RNA totale utilizzando colonnine con code di poly-T;
- Conversione dell'mRNA in cDNA utilizzando la trascrittasi inversa e i primer poly-T;
- Amplificazione del cDNA utilizzando T7 RNA polimerasi in presenza di biotina-UTP e biotina-CTP in modo da ottenere da 50 a 100 copie di cDNA marcato;
- Incubazione del cDNA a 94°C in un buffer di frammentazione per produrre frammenti di lunghezza tra 35 e 200 nucleotidi;
- Ibridizzazione sul chip e successivi lavaggi;
- Marcatura del cDNA ibridizzato con Streptavin-Phycoerythrin e successivi lavaggi;
- Acquisizione dell'immagine del chip con scanner laser;
- Analisi dell'immagine per l'estrapolazione dei dati.

1.4.3 La tecnologia *"Spotted" array*

L'altra tecnologia largamente utilizzata per produrre *microarray* è quella degli *"spotted" array*; in questo caso viene utilizzato un robot che preleva una piccola quantità di sonda in soluzione da una piastra da microtitolazione e la deposita sulla superficie del *microarray*. La sonda può essere cDNA, prodotto mediante PCR od oligonucleotidi; ogni sonda è complementare ad un unico gene. Esistono diversi metodi per fissare le sonde alla superficie del vetrino; il più utilizzato consiste nel ricoprire il supporto con uno strato di poli-lisina che determina la formazione di legami aspecifici con le sonde.
Il processo di "*spotting*" di questi *microarray* può essere schematizzato come segue:

- Copertura del vetrino con poli-lisina;
- Preparazione delle sonde in una piastra da microtitolazione;
- Programmazione del robot per le operazioni di "spotting" mediante pin e ugelli ink-jet;
- Deposizione delle sonde in blocchi ordinati seguendo la mappa programmata per stabilire la posizione e la concentrazione di ogni spot;
- Saturazione delle aree non stampate con anidride succinica per sfavorire legami aspecifici fra il cDNA bersaglio e il supporto;
- Denaturazione delle sonde ad alta temperatura in modo che siano a singolo filamento.

Una volta realizzato il *microarray* si può procedere alla preparazione del campione e alla sua ibridizzazione come segue:

- Estrazione dell'RNA totale dalle cellule;

- Isolamento (opzionale) dell'mRNA grazie alla presenza delle code di poly-A;
- Retrotrascrizione dell'RNA in cDNA in presenza di amino-allil-dUTP (AA-dUTP);
- Marcatura dei filamenti di cDNA con i fluorocromi Cy3 e Cy5, che si legano all'AA-dUTP;
- Ibridizzazione del cDNA marcato con le sequenze presenti sul vetrino;
- Asportazione mediante lavaggi del materiale non ibridizzato;
- Acquisizione dell'immagine del vetrino con scanner laser;
- Analisi dell'immagine per l'estrapolazione dei dati.

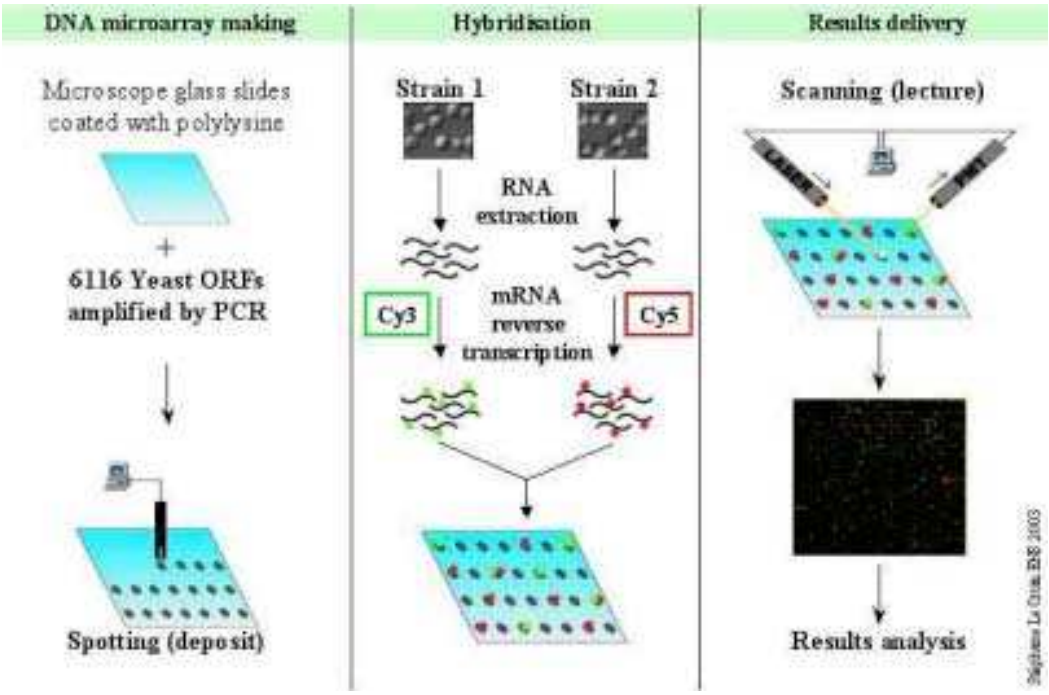


Figura 1.10 Processo di *spottinig* degli *array* e ibridazione del campione.

Rispetto alla tecnologia *Affymetrix*, negli *"spotted" array* l'irregolarità nell'operazione di deposizione delle sonde si può ripercuotere sulla corretta estrazione del dato. Inoltre, la presenza di sonde PM e MM sui vetrini *Affymetrix*,

conferisce a questi *microarray* una maggiore affidabilità nella rilevazione di segnali di ibridizzazione aspecifica.
Il vantaggio principale degli *"spotted" array*, invece, consiste nella possibilità che ogni laboratorio ha di disegnare le sonde da utilizzare nello "*spotting*" e nella maggiore flessibilità di questa tecnologia rispetto ad *Affymetrix*, i cui dati spesso non sono analizzabili con gli innumerevoli software per l'elaborazione di dati disponibili.

1.4.4 Caratteristiche di un *microarray*

Un *microarray* può essere definito come una matrice ordinata di elementi microscopici su un substrato planare che consente il legame specifico di geni o di prodotti di geni. La parola *microarray* deriva dal greco *mikro*, che significa piccolo, e dal francese *arayer*, che significa arrangiare; i *microarray*, anche conosciuti come biochip, DNA chip e gene chip, contengono, infatti, collezioni di microscopici elementi, spot, disposti in righe e colonne.
Ogni riga di elementi deve essere disposta sul substrato lungo una linea orizzontale e ogni colonna deve formare una linea verticale perpendicolare alla riga. Gli elementi ordinati devono avere uguale dimensione, uniforme spaziatura e posizione unica sul substrato
Su un singolo substrato planare possono essere combinati diversi *microarray* e ciò è utile sia dal punto di vista dell'analisi successiva, sia per i processi di realizzazione in parallelo di tali dispositivi.
L'ordinamento in righe e colonne degli elementi è un grande vantaggio per l'analisi dei *microarray*, poiché questo tipo di disposizione consente una rapida deposizione, individuazione e quantificazione degli spot.

La disposizione degli spot in righe e colonne può essere ottenuta utilizzando tecnologie standard di *motion control*, come attuatori lineari ed *encoder*, e ciò permette un abbattimento dei costi di produzione, in quanto i *microarray* possono essere stampati in modalità rapida e completamente automatizzata, con una velocità e una precisione che non sarebbero possibili con formati irregolari.
La regolarità della disposizione degli spot, inoltre, favorisce il processo di quantificazione, poiché i software di elaborazione fanno uso di griglie ordinate per l'estrazione del dato numerico e di una "mappa cartesiana" per assegnare allo spot l'identificativo del gene che rappresenta.

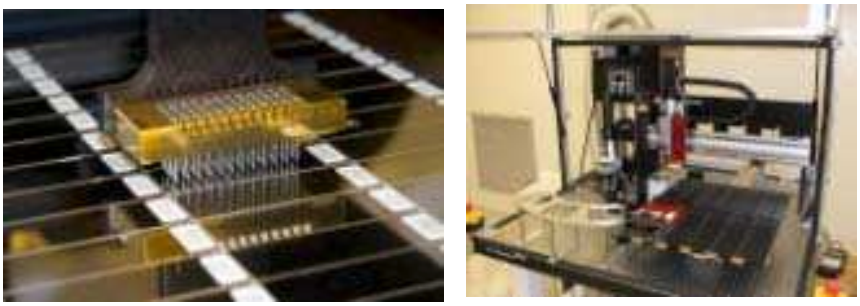


Figura 1.11 *Printer-head* di un robot per *spotting* di *microarray* e camera di *printing*.

Un tipico *spot* contiene approssimativamente 10^9 molecole bloccate sul substrato di vetro. Queste molecole sonda possono essere DNA genomico, cDNA, mRNA, proteine, tessuti o altri tipi di molecole che necessitano di un'analisi quantitativa. Oligonucleotidi sintetici, cioè piccole molecole di DNA a singolo filamento sintetizzate chimicamente possono costituire un tipo eccellente di sonda.
I vantaggi di avere elementi microscopici sono:

- alta densità degli spot (> 5000 elementi/cm2);
- rapida cinetica di reazione;
- possibilità di analizzare interi genomi su un singolo vetrino.

Gli esperimenti che esaminano tutti i geni di un genoma su un singolo substrato procurano una visione globale del fenomeno biologico, impossibile da ottenere con tecnologie limitate a sottoinsiemi di geni.
Per substrato si intende un supporto parallelo e piatto sul quale viene configurato un *microarray*. Uno dei materiali più utilizzati è il vetro per la sua capacità ideale di consentire il legame con le molecole sonda, ma possono essere utilizzati anche materiali plastici, silicio, filtri di nylon e nitrocellulosa.
Per essere utilizzato per la costruzione di un *microarray* il substrato deve essere planare: tutti i materiali planari sono solidi, ma non tutti quelli solidi sono planari. Il vantaggio di avere un substrato piatto su tutta la superficie si ripercuote sull'automatizzazione della procedura di stampa mediante pin e ugelli *ink-jet* o sulla precisione delle fotomaschere per la fotolitografia. I materiali planari consentono anche un accurato *scanning* del *microarray*, grazie alla precisa individuazione della distanza fra gli elementi ottici dello scanner e la superficie del microarray (distanza del fuoco ottico).
I materiali planari, inoltre, tendono ad essere impermeabili ai liquidi, consentono di realizzare piccoli spot e di minimizzare il volume di reazione durante l'ibridizzazione.

1.5 Applicazione dei *microarray*

I *microarray* si stanno rivelando degli strumenti efficaci in differenti campi di indagine. I primi esperimenti hanno fatto uso di questi supporti per verificare ipotesi formulate in studi precedenti.

Ultimamente la tendenza si è invertita e i *microarray* vengono utilizzati come dispositivi di indagine primaria, capaci di fornire risposte robuste, ma anche di porre nuovi quesiti al ricercatore.
Da questo punto di vista il grosso vantaggio dei *microarray*, oltre all'estesa potenza di calcolo parallelo, sta nella possibilità di poter coinvolgere nella reazione di ibridizzazione molti geni sullo stesso supporto. Questo dà un contributo alla possibilità di ricreare pathway di co-regolazione e di analizzare le inter-relazioni tra geni diversi.
Di seguito sono descritti alcuni esempi dei settori di applicazione dei *microarray*.

1.5.1 Tassonomia dei tessuti

Cellule appartenenti allo stesso organismo possiedono lo stesso genoma anche se differiscono per forma e funzione. La differenziazione di ogni cellula in un tipo o in un altro si realizza grazie ad una programmazione genetica ben definita che modifica nel corso dello sviluppo l'insieme dei geni espressi.
Esaminando il *pattern* di espressione genica su scala genomica con i *microarray* è possibile catalogare di differenti tessuti in modo da costituire un database di espressione. Uno degli scopi degli studi di questo tipo è la comprensione dei meccanismi che stanno alla base dello sviluppo e della differenziazione cellulare, che, una volta alterati, possono determinare l'insorgenza di malattie.

1.5.2 Identificazione delle basi molecolari delle malattie

Conoscere le basi molecolari di una malattia può aiutare a comprenderne la trasmissione genetica, la modalità d'insorgenza e la prognosi al fine di poter fare una diagnosi precoce o di agire con terapie mirate. Il confronto dell'espressione genica tra tessuti sani e malati mediante *microarray* può essere un valido strumento per l'identificazione di quei geni che sono coinvolti nello sviluppo di una patologia, o come geni causativi o semplicemente come fattori di rischio predisponenti.

Diversi gruppi di ricerca stanno facendo uso dei *microarray* per creare una carta d'identità dettagliata dei vari tipi di tumore al fine di costituire una vasta raccolta di profili di espressione genica da utilizzare a fini diagnostici.

Si possono ricordare in questo ambito lo studio di Ross (Ross et al., 2000) su sessanta linee cellulari differenti di cancro, denominato NCI60, o gli studi estensivi sui linfomi a cellule B giganti di Alizadeh (Alizade et al. 2000), entrambi dell'Università di Stanford.

Riuscire ad effettuare una classificazione così dettagliata si riflette sulla possibilità di riconoscere la malattia fin dai primi stadi di sviluppo, in modo da poter programmare terapie più mirate.

Capitolo 2

Metodi Bayesiani Empirici

Introduzione

La tecnologia del DNA *microarray* offre la possibilità a molti campi della ricerca biologica di analizzare simultaneamente migliaia di geni, dando luogo al problema che dal punto di vista statistico si traduce nel paradigma, largamente discusso nella letteratura, detto "*large p and small n*". I metodi Bayesiani empirici si adattano naturalmente a questa caratteristica dei dati in quanto sono in grado di ridurre in maniera significativa la dimensionalità del problema nella fase di inferenza, mediante l'utilizzo di solo alcuni parametri ignoti.

I metodi bayesiani empirici permettono di perfezionare la stima ottenuta da metodi classici della variabilità del sistema e permettono di valutare geni differenzialmente espressi attraverso l'utilizzo dell'*odds* a posteriori. In questo modo l'inferenza sulla variabilità del sistema tiene conto dell'informazione di tutti i dati.

Nella letteratura degli ultimi dieci anni si ritrovano molti lavori per l'analisi della differenza di espressione in geni sotto diverse condizioni sperimentali. Un approccio generale, presentato nel 1997 da Chen *et al.* [rif.1], consiste nell'utilizzare varianti del test *t* e del test *F* per la valutazione della variabilità del sistema. Una variante di questi test è stata presentata da Dudoit *et al.* [rif.2] nel 2002 che hanno sperimentato il test di Welsh con il calcolo del *p-value* attraverso permutazioni. L'anno precedente Tusher *et al.* [rif. 3] nel 2001 avevano utilizzato il test *t* con l'ausilio di una costante in grado di migliorare la stima dello *standard*

error attraverso una metodologia nota nella letteratura come SAM (*Significance Analysis of Microarray*) con l'ausilio dell'errore stimato e degli errori nell'identificazione dei geni differenzialmente espressi da parte dei modelli (*false discovery rate)*. Baldi e Long [rif. 4] nel 2001 avevano utilizzato la va7rianza a posteriori derivata dai metodi bayesiani senza preoccuparsi di estendere il test per problemi multidimensionali.

Uno dei primi a sviluppare metodi bayesiani empirici, che tenessero presente della possibilità di avere dati raccolti sotto diverse condizioni sperimentali, o che si riferissero a popolazioni differenti, è stato Newton *et al.* (2001) [rif. 5] con l'introduzione di un modello mistura. Stocasticamente, i geni possono essere equivalentemente espressi oppure differenzialmente espressi. I geni considerati equivalentemente espressi sono quelli che si comportano nello stesso modo in distribuzione; sono invece considerati differenzialmente espressi quei geni che presentano distribuzioni differenti. La forma specifica delle distribuzioni è data dall'espressione di media latente per ogni gene; le medie latenti non vengono trattate come parametri fissi, ma come quantità causali che seguono la forma di una distribuzione specifica. Dati i componenti appena descritti, il problema inferenziale si traduce sostanzialmente nell'individuazione della probabilità a posteriori condizionata dalle osservazioni campionarie.

In quanto segue si proporrà una carellata delle caratteristiche principali del modello mistura gerarchico, uno studio più approfondito del modello parametrico Gamma – Gamma, introdotto da Newton *et al.,* e del modello LogNormale – Normale, ampliato da Kendziorski *et al.* nel 2003 [rif.. 6]. Essendo tale modelli studiati nel caso di disegno sotto 2 condizioni sperimentali, analizzerò poi un ampliamento dei modelli in caso di un numero superiore di condizioni. Per una migliore analisi dei modelli verrà considerata l'ipotesi di costanza del coefficiente di variazione, argomento ampliamente discusso nella letteratura, come per esempio da Chen *et al.* (1997) [rif. 1].

In realtà, nel corso degli anno, i modelli gerarchici studiati sono stati di diversi tipi oltre a quello considerato nel corso dell'elaborato, ma in ognuno dei casi le

analisi condotte sono state implementate esclusivamente in casistiche specifiche: un esempio è quello portato da Efron *et al.* [rif. 7] che, nel 2004 combina l'approccio bayesiano empirico con una versione locale del *false discovery rate* per valutare le differenze di espressione genica.

2.1 La struttura del modello mistura

Un modello adatto all'analisi delle differenze genetiche sotto diverse condizioni deve essere in grado di descrivere la distribuzione di probabilità di un insieme di misure di espressioni raccolte da un gene *g*. Si assume che nella fase di raccolta dei dati, questi siano stati opportunamente normalizzati e filtrati. E' inoltre opportuno sottolineare il fatto che i geni sono raccolti solitamente da cellule sotto diverse condizioni, quindi i dati possono essere considerati replicazioni delle condizioni stesse. Il numero di *pattern* di espressione possibili dipende dal numero di condizioni sotto le quali si sono raccolti i dati. Se, per esempio, i dati sono stati raccolti sotto due condizioni (A e B) il numero di *pattern* di espressione possibili sono 2:

1. Espressione equivalente: $H_0 : \mu_A = \mu_B$;
2. Espressione differenziale: $H_1 : \mu_A \neq \mu_B$.

Se i dati sono invece raccolti sotto 3 condizioni (A, B e C) il numero di *pattern* possibili è $\binom{3}{1} + \binom{3}{2} + \binom{3}{3} = 5$:

1. *Pattern 1:* Espressione equivalente: $H_0 : \mu_A = \mu_B = \mu_C$:
2. *Pattern 2:* Espressione differenziale: $H_1 : \mu_A \neq [\mu_B = \mu_C]$;
3. *Pattern 3:* Espressione differenziale: $H_1 : [\mu_A = \mu_B] \neq \mu_C$;
4. *Pattern 4:* Espressione differenziale: $H_1 : [\mu_A = \mu_C] \neq \mu_B$;

5. *Pattern 1:* Espressione differenziale: $H_0 : \mu_A \neq \mu_B \neq \mu_C$.

Con *microarray* da cellule sotto 4 condizioni si ottengono 15 *patterns* di espressione differenti. Il numero totale dei possibili *patterns* corrispondono all'esponenzile di Bell del numero delle possibili partizioni. E' possibile comunque ridurre il numero di *patterns* dato che non tutti hanno rilevanza biologica.

2.1.1 Geni campionati sotto 2 condizioni.

Nel descrivere la struttura del modello, si prenderà ora in considerazione il caso più semplice: geni raccolti sotto 2 condizioni sperimentali, con 2 *patterns* di espressione genica.

In generale, il generico dato raccolto $x_{j,i}$ rappresenta la trasformazione logaritmica della misura originaria di espressione del *j*-esimo gene nell'*i*-esimo campione; in particolare l'espressione misurata è un rapporto di fluorescenze per ciascun canale (rosso e verde).

Nel corso della formulazione del modello si considera la singola osservazione come uno scostamento casuale da una media specifica del gene a cui si riferisce.

Nel caso di campionamento sotto 2 condizioni sperimentali, lo spazio campionario viene ripartito in 2 sottoinsiemi s_1e s_2; dove s_k contiene gli indici dei campioni nel *k*-esimo gruppo, con *k=1,2*. Nel corso dell'elaborato si sono indicate con $x_g = (x_{g,1}, x_{g,2}, ..., x_{g,n_1})$ le n_1 repliche sotto la prima condizione, e con $y_g = (y_{g,1}, y_{g,2}, ..., y_{g,n_2})$ le n_2 repliche sotto la seconda condizione. Nel caso in cui la distribuzione di probabilità dell'espressione di un gene risenta di effetti dovuti alla partizione dello spazio campionario, si tratta di differenza di espressione per il gene a cui ci si riferisce (DE_g); nel caso questo non avvenga, si è nel caso di equivalenza di espressione (EE_g). Per la formulazione del problema, si

definiscono $\mu_{g,1}$ la media calcolata nei campioni appartenenti a s_1 per il gene g, e $\mu_{g,2}$ la media in s_2 per il gene g. Si assume che tali medie derivino da una medesima distribuzione di probabilità $\pi(\mu)$: ciò costituisce l'informazione a priori. Sia p definito come la frazione di geni differenzialmente espressi (DE); quindi $1-p$ la frazione di geni equivalentemente espressi (EE).

La distribuzione marginale dei dati è:

$$p(x_g, y_g) = \Pr(DE) \cdot p(x_g, y_g \mid DE) + \Pr(EE) \cdot p(x_g, y_g \mid EE)$$
$$= p \cdot f_1(x_g, y_g) + (1-p) \cdot f_0(x_g, y_g) \quad . \tag{2.1}$$

Tale distribuzione dei dati è fornita sotto forma di modello mistura, in cui la funzione di densità di probabilità $f_0(x_g, y_g)$ descrive i dati in caso di espressione equivalente nel j-esimo gene, e la funzione di densità di probabilità $f_1(x_g, y_g)$ descrive i dati in caso di espressione differenziale. A priori non si conosce la distribuzione del gene, viene introdotto un parametro discreto di mistura p che denota la probabilità di espressione differenziale.

La conoscenza del parametro p e della forma di f_0 e f_1 permette, mediante il teorema di Bayes, di ricavare la probabilità a posteriori di espressione differenziale del j-esimo gene:

$$\Pr(DE \mid x_g, y_g) = \frac{p(x_g, y_g \mid DE)}{\Pr(x_g, y_g)}$$
$$= \frac{p \cdot f_1(x_g, y_g)}{p \cdot f_1(x_g, y_g) + (1-p) \cdot f_0(x_g, y_g)} \quad . \tag{2.2}$$

Si può considerare inoltre l'*odds* a posteriore di DE:

$$odds_g = \frac{\Pr(DE \mid x_g, y_g)}{\Pr(EE \mid x_g, y_g)}$$
$$= \frac{p}{1-p} \frac{f_1(x_g, y_g)}{f_0(x_g, y_g)}. \tag{2.3}$$

Questi due risultati permettono di fare inferenza sull'espressione differenziale di ogni singolo gene. Fondamentale a questo punto identificare la forma delle distribuzioni di probabilità f_0 e f_1 che indicano la distribuzione marginale dei dati (x_g, y_g) sotto le 2 condizioni. Nell'approccio bayesiano empirico, si suppone che x_g sia un campione casuale semplice (c.c.s) dalla distribuzione $f_{obs}(x_g|\mu_{g1})$ e che y_g sia un c.c.s dalla distribuzione $f_{obs}(y_g|\mu_{g2})$. Nel caso di equivalenza di espressione del gene, si ipotizzano le misurazioni campionarie x_g e y_g provenire da una distribuzione comune, dato che si suppone non vi siano variazioni sistematiche tra le cellule nelle 2 condizioni.

Si considerano quindi le $N=n1+n2$ misurazioni come un campione di osservazioni indipendenti e identicamente distribuite dalla distribuzione sulla singola osservazione $f_{obs}(\cdot\,|\mu_g)$. Le possibili forme parametriche che si andranno ad assumere per tale distribuzione sono la distribuzione Gamma e la distribuzione Log-Normale (vedi paragrafi 2.3.1 e 2.3.2). L'approccio consiste nel trattare μ_g non come costante, ma proveniente a sua volta da una distribuzione di probabilità $\pi(\mu_g)$, che rappresenta le variazioni nel livello di espressione medio fra tutti i geni considerati nell'esperimento di *microarray*. In questo modo si sfrutta tutta l'informazione relativa ai geni e, per fare inferenza su un dato gene, si tiene conto dei dati di espressione relativi a tutti i geni. In definitiva si può esprimere la distribuzione marginale (predittiva) dei dati, sotto ipotesi nulla *EE*, come:

$$\begin{aligned} f_0(x_g, y_g) &= \int f_0(x_g, y_g \mid \mu_g) \cdot \pi(\mu_g) \\ &= \int \left(\prod_{i=1}^{n_1} f_{obs}(x_{g,i} \mid \mu_g) \right) \left(\prod_{j=1}^{n_2} f_{obs}(y_{g,j} \mid \mu_g) \right) \pi(\mu_g) d\mu_g \end{aligned} \qquad (2.4)$$

Nel passare dalla seconda alla terza espressione è stata sfruttata l'ipotesi di indipendenza ed egual distribuzione delle osservazioni.

Sotto l'ipotesi alternativa di espressione differenziale *DE*, la media latente μ_{g1} relativa al campione $x_{g,i}$ (con $i = 1,\ldots, n_1$) è diversa dalla media μ_{g2} del campione $y_{g,i}$ (con $i = 1,\ldots, n_2$) relativo alla seconda condizione. Ognuno dei due valori della

media è generato in maniera indipendente dalla distribuzione marginale sotto ipotesi *DE*, ovvero dalla:

$$f_1(x_g, y_g) = f_0(x_g) f_0(y_g) \qquad (2.5)$$

con

$$f_0(x_g) = \int (\prod_{i=1}^{n_1} f_{obs}(x_{g,i} \mid \mu_g) \pi(\mu_g) d\mu_g \qquad (2.6)$$

e

$$f_0(y_g) = \int \prod_{i=1}^{n_2} f_{obs}(y_{g,i} \mid \mu_g) \pi(\mu_g) d\mu_g \; . \qquad (2.7)$$

Concludendo, l'*odds* dell'espressione differenziale ottenuto dalle replicazioni $x_g = (x_{g,1}, x_{g,2}, \ldots, x_{g,n1})$ per la prima condizione e $y_g = (y_{g,1}, y_{g,2}, \ldots, y_{g,n1})$ per la seconda condizione è:

$$odds_g = \frac{p}{1-p} \frac{f_0(x_g) f_0(y_g)}{f_0(x_g, y_g)} \qquad (2.8)$$

Le componenti di distribuzione sono parametrizzate nei paragrafi 2.3.1 e 2.3.2 e l'inferenza è ottenuta stimando il parametro p e i parametri della distribuzione marginale, mediante massimizzazione della verosimiglianza marginale, e infine calcolando l'*odds* a posteriori dell'espressione differenziale per ogni gene g o la probabilità a posteriori di espressione genica differenziale.

2.1.2 Geni campionati sotto più condizioni

L'estensione al caso in cui i geni siano campionati da cellule sotto più di due condizioni è immediata. Come detto nel paragrafo 2.1 sono possibili più *pattern* di espressione nel caso di condizioni multiple; il numero totale di possibili *pattern* è uguale al numero esponenziale di Bell delle possibili partizioni, ma questo

numero può essere ridotto considerando nel sistema di ipotesi i *pattern* di interesse biologico.

Si suppone che siano possibili $m + 1$ *pattern* di espressione per i di dati espressione genica del vettore $d_g = (d_{g,1}, d_{g,2}, \ldots, d_{g,N})$ relativo al gene g in N condizioni. Per ogni *pattern* k, l'insieme delle condizioni sperimentali $S = 1,\ldots,N$ è partizionato in $r(k)$ sottoinsiemi mutualmente esclusivi ed esausivi $S_{i,k}$ ($S_{i,k} \cap S_{j,k} = \phi \forall i, j = 1,2,\ldots,r(k)$ con $i \neq j$ e $\cup S_{i,j} = S$) in cui ogni misura contenuta in un sottoinsieme di $S_{i,k}$ condivide un livello medio comune di espressione. Generalizzando l'espressione (2.1), d_g è governato da un modello mistura della forma:

$$p(d_g) = \sum_{k=0}^{m} \Pr(Pattern\ k) \cdot p(d_g \mid Pattern\ k) = \sum_{k=0}^{m} p_k f_k(d_g) \quad , \tag{2.9}$$

con $k = 0,\ldots,m$ indicante i possibili $m + 1$ *pattern* distinti, p_k le probabilità a priori di appartenere ai vari *pattern* e f_k distribuzione di probabilità delle misure sotto diversi *pattern*. Conseguentemente la distribuzione di probabilità a posteriori del *pattern* di espressione k è:

$$\Pr(k \mid d_g) = \frac{p(k, d_g)}{p(d_g)} = \frac{p_k f_k(d_g)}{\sum_{j=0}^{m} p_j f_j(d_g)} \propto p_k f_k(d_g), \tag{2.10}$$

e l' *odds* a posteriori in favore del *pattern* k è:

$$odds_{g,k} = \frac{p_k}{1 - p_k} \frac{f_k(d_g)}{1 - f_k(d_g)}. \tag{2.11}$$

Generalizzando l'espressione (2.5), la densità $f_k(d_g)$, dato lo specifico *pattern* k, sarà il prodotto delle densità marginali delle misurazioni nelle varie condizioni:

$$f_k(d) = \prod_{i=1}^{r(k)} f(d_{g,S_{i,k}}) \quad , \tag{2.12}$$

dove $f(d_{g,S_{i,k}})$ è la densità di probabilità per le misurazioni del sottoinsieme $S_{i,k}$. Si assume che le misurazioni $S_{i,k}$, che condividono lo stesso livello medio di

espressione μ_g, si presentino in maniera indipendente della stessa distribuzione di probabilità $f_{obs}(\cdot|\mu_g)$. L'approccio consiste nel trattare μ_g non come fissata, ma proveniente a sua volta da una distribuzione di probabilità $\pi(\mu_g)$, che rappresenta le variazioni nel livello di espressione medio fra tutti i geni considerati nell'esperimento di *microarray*. In questo modo si sfrutta tutta l'informazione relativa ai geni e nel fare inferenza su un dato gene si tiene conto dei dati di espressione relativi a tutti i geni. In definitiva si può esprimere la distribuzione $f(d_{g,S_{i,k}})$, densità marginale (predittiva) delle misurazioni $d_{g,S_{i,k}}$ in questo modo:

$$f(d_{g,S_{i,k}}) = \int \left(\prod_{s \in S_{i,k}} f_{obs}(d_{g,s} | \mu_g) \right) \pi(\mu_g) d\mu_g \,. \qquad (2.13)$$

La probabilità a posteriori data nell'equazione 2.10 viene utilizzata per fare inferenza circa i *pattern* di espressione di ogni gene. Prima di utilizzare questo risultato deve essere indicata la forma della distribuzione da utilizzare per componenti del modello mistura di tipo gerarchico. Nei paragrafi 2.3.1 e 2.3.2 sono specificati 2 tipi di distribuzione, la famiglia Gamma-Gamma e quella LogNormale-Normale.

2.2 La distribuzione parametrica

Come visto nel paragrafo 2.2, il modello mistura è specificato dalla distribuzione sulla singola osservazione $f_{obs}(\cdot | \mu_g)$, che caratterizza la variabilità relativa alle misure ripetute di un gene avente la stessa media di espressione μ_g, e da una seconda componente $\pi(\mu_g)$ che descrive la variabilità in queste medie tra geni.

Le distribuzioni che saranno descritte nei paragrafi successivi sono di tipo parametrico, le stime dei relativi parametri dipenderanno dalla variabilità tipica di

ogni singolo esperimento di *microarray*. Ci sono invece caratteristiche tipiche dei dati ricavati da esperimenti di *mcroarray* che sono ripetutamente osservate in tali *dataset*. Una di queste riguarda il coefficiente di variazione dei dati, che risulta pressoché costante in molti *dataset*, come osservato da Chen (1997) [rif. 1] e Newton (2001) [rif. 5]. Questa considerazione permette di utilizzare il modello Gamma-Gamma e il modello LogNormale-Normale che ipotizzano il coefficiente di variazione costante per tutti i geni analizzati. Nel Capitolo 3 sono state effettuate le simulazioni per verificare la robustezza dei due modelli; in questo modo si possono analizzare i dati relativi alle leucemie (vedi Capitolo 4).

2.2.1 Il modello Gamma-Gamma

Nel modello Gamma-Gamma (GG), la distribuzione sulla singola osservazione è di tipo Gamma con parametro di forma $\alpha > 0$ e media μ_g; essendo il valore atteso dato dal rapporto tra parametro di forma e parametro di scala, il parametro di scala λ_g è pari a α / μ_g. La distribuzione sulla singola osservazione assume quindi la forma:

$$f_{obs}(z \mid \mu_g) = \frac{\lambda_g^{\alpha} z^{\alpha-1} \exp\{-\lambda_g z\}}{\Gamma(\alpha)} \qquad (2.14)$$

per $z > 0$.

Il coefficiente di variazione della variabile $Z \mid \mu_g \sim Ga(\alpha,\lambda_g)$ è pari a

$$\frac{SE(Z \mid \mu_g)}{E(Z \mid \mu_g)} = \frac{\sqrt{\alpha / \lambda^2}}{\alpha / \lambda} = \frac{1}{\sqrt{\alpha}} \qquad (2.15)$$

ed è quindi costante per tutti i geni g, dato che il parametro α non dipende dal singolo gene. Fissato α, è necessario scegliere la distribuzione marginale di μ_g,

$\pi(\mu_g)$. Se si ipotizza questa distribuzione essere l'inversa di una distribuzione Gamma, in questo modo la quantità $\lambda_g = \alpha / \mu_g$ ha distribuzione Gamma con parametro di forma α_0 e parametro di scale ν, ovvero $\lambda_g \sim Ga(\alpha_0, \nu)$. In definitiva, il modello statistico probabilistico prevede un parametro tridimensionale, $\theta = (\alpha, \alpha_0, \nu)$.

Si ottiene la densità marginale predittiva delle osservazioni (cfr. 2.13) tramite integrazione:

$$
\begin{aligned}
f(z_1, z_2, \ldots, z_n) &= \int_0^\infty f(z_1, z_2, \ldots, z_n \mid \mu_g)\pi(\mu_g)d\mu_g \\
&= \int_0^\infty \prod_{i=1}^{n}\left[f\left(z_i \mid \mu_g\right)\right]\cdot \pi(\mu_g)d\mu_g \\
&= \int_0^\infty \prod_{i=1}^{n}\left[\frac{\lambda_g^{\alpha} z_i^{\alpha-1} e^{-\lambda_g z_i}}{\Gamma(\alpha)}\right]\cdot \frac{\nu^{\alpha_0}\lambda_g^{\alpha_0-1}e^{-\nu\lambda_g}}{\Gamma(\alpha_0)}d\lambda_g \\
&= \frac{\left(\prod_{i=1}^{n} z_i\right)^{\alpha-1}\nu^{\alpha_0}}{\Gamma^n(\alpha)\Gamma(\alpha_0)}\int_0^\infty \lambda_g^{n\alpha+\alpha_0-1}e^{-\lambda_g\left(\sum_{i=1}^{n} z_i+\nu\right)}d\lambda_g \\
&= \frac{\left(\prod_{i=1}^{n} z_i\right)^{\alpha-1}\nu^{\alpha_0}}{\Gamma^n(\alpha)\Gamma(\alpha_0)}\int_0^\infty \frac{t^{n\alpha+\alpha_0-1}e^{-t}}{\left(\sum_{i=1}^{n} z_i+\nu\right)^{n\alpha+\alpha_0}}dt \\
&= \frac{\nu^{\alpha_0}\Gamma\left(n\alpha+\alpha_0\right)}{\Gamma^n(\alpha)\Gamma(\alpha_0)}\cdot\frac{\left(\prod_{i=1}^{n} z_i\right)^{\alpha-1}}{\left(\sum_{i=1}^{n} z_i+\nu\right)^{n\alpha+\alpha_0}}
\end{aligned}
\qquad (2.16)
$$

.

Per passare dalla prima all'ultima espressione delle (2.16) è stata sfruttata l'indipendenza delle variabili Z_i e la proprietà di egual distribuzione. Per l'integrazione è stato utilizzato il metodo di sostituzione ponendo $t = \lambda_g (\sum_{i=1}^{n} z_i + \nu)$. Ottenuto questo risultato importante, è possibile calcolare la

probabilità a posteriori di qualsiasi *pattern* di espressione secondo la formula 2.10 e le misure di *odds* relative. Nel caso particolare di 2 condizioni, l' *odds* a posteriori per l'espressione differenziale si semplifica come di seguito:

$$
\begin{aligned}
odds &= \frac{\Pr(DE \mid x_g, y_g)}{\Pr(EE \mid x_g, y_g)} \\
&= \frac{p}{1-p}\frac{f_0(x_g) f_0(y_g)}{f(x_g, y_g)} \\
&= \frac{p}{1-p}\cdot\frac{\dfrac{K'\left(\prod_{i=1}^{n_1} x_{g,i}\right)^{\alpha-1}}{\left(\nu+\sum_{i=1}^{n_1} x_{g,i}\right)^{n_1\alpha+\alpha_0}}\cdot\dfrac{K''\left(\prod_{i=1}^{n_2} y_{g,i}\right)^{\alpha-1}}{\left(\nu+\sum_{i=1}^{n_2} y_{g,i}\right)^{n_2\alpha+\alpha_0}}}{\dfrac{K'''\left(\prod_{i=1}^{n_1} x_{g,i}\prod_{i=1}^{n_2} y_{g,i}\right)^{\alpha-1}}{\left(\nu+\sum_{i=1}^{n_1} x_{g,i}+\sum_{i=1}^{n_2} y_{g,i}\right)^{N\alpha+\alpha_0}}} \\
&= \frac{p}{1-p}\frac{K'K''}{K'''}\cdot\frac{\left(\nu+\sum_{i=1}^{n_1} x_{g,i}+\sum_{i=1}^{n_2} y_{g,i}\right)^{N\alpha+\alpha_0}}{\left(\nu+\sum_{i=1}^{n_1} x_{g,i}\right)^{n_1\alpha+\alpha_0}\left(\nu+\sum_{i=1}^{n_2} y_{g,i}\right)^{n_2\alpha+\alpha_0}},
\end{aligned}
$$

(2.17)

con $N=n_1+n_2$.

Ora si possono semplificare alcuni termini relativi a *K', K''* e *K'''*.

$$
\begin{aligned}
\frac{K'K''}{K'''} &= \frac{\nu^{\alpha_0}\Gamma(n_1\alpha+\alpha_0)}{\Gamma^{n_1}(\alpha)\Gamma(\alpha_0)}\cdot\frac{\nu^{\alpha_0}\Gamma(n_2\alpha+\alpha_0)}{\Gamma^{n_2}(\alpha)\Gamma(\alpha_0)}\cdot\frac{\Gamma^{N}(\alpha)\Gamma(\alpha_0)}{\nu^{\alpha_0}\Gamma(N\alpha+\alpha_0)} \\
&= \frac{\nu^{\alpha_0}\Gamma(n_1\alpha+\alpha_0)\Gamma(n_2\alpha+\alpha_0)}{\Gamma(\alpha_0)\Gamma(N\alpha+\alpha_0)},
\end{aligned}
$$

(2.18)

sapendo che $\Gamma^{n_1}(\alpha)\cdot\Gamma^{n_2}(\alpha)=\Gamma^{n_1+n_2}(\alpha)=\Gamma^{N}(\alpha)$.

Dopo questa semplificazione la formula finale può essere così scritta:

$$odds_g = \frac{p}{1-p}\cdot K\cdot\frac{\left(\nu+\sum_{i=1}^{n_1}x_{g,i}+\sum_{i=1}^{n_2}y_{g,i}\right)^{N\alpha+\alpha_0}}{\left(\nu+\sum_{i=1}^{n_1}x_{g,i}\right)^{n_1\alpha+\alpha_0}\left(\nu+\sum_{i=1}^{n_2}y_{g,i}\right)^{n_2\alpha+\alpha_0}} \quad , \qquad (2.19)$$

con

$$K = \frac{\nu^{\alpha_0}\,\Gamma(n_1\alpha+\alpha_0)\Gamma(n_2\alpha+\alpha_0)}{\Gamma(\alpha_0)\Gamma(N\alpha+\alpha_0)} \quad . \qquad (2.20)$$

Nel paragrafo 2.4 viene indicato il metodo per ottenere le stime dei parametri $\theta = (\alpha\ ,\ \alpha_0\ ,\nu)$ e del quarto parametro p, indicante la probabilità a priori di geni differenzialmente espressi.

2.2.2 Il modello LogNormale-Normale

Nel modello LogNormale-Normale (LNN) si ipotizza che la distribuzione relativa alla trasformata logaritmica della singola misurazione sia normale. Si indica con $\tilde{z}_{g,i} = \log z_{g,i}$ il logaritmo naturale della misura di espressione $z_{g,i}$. La variabile $Z_{g,i}|\mu_g$ si distribuisce come una $N(\mu_g\ ,\ \sigma^2)$, con una varianza σ^2 comune per tutti i geni e con una media μ_g dipendente dal singolo gene. La distribuzione a priori di μ_g è una $N(\mu_0\ ,\ \tau_0^2)$. In definitiva, è previsto un parametro tridimensionale $\theta = (\mu\ ,\ \sigma^2,\ \tau_0)$, da cui si ricava la densità marginale predittiva delle osservazioni. Dato $Z_{g,i}=\mu_g+\varepsilon$ con $\mu_g \sim N(0\ ,\ \tau_0^2)$ e $\varepsilon_i \sim N(0\ ,\ \sigma^2)$, indipendentemente da μ_g, si ricavano le seguenti quantità:

$$
\begin{aligned}
&E\left(\tilde{Z}_{g,i}\right) = E\left(\mu_g\right) + E\left(\varepsilon_i\right) = \mu_0, \\
&Var\left(\tilde{Z}_{g,i}\right) = Var\left(\mu_g\right) + Var\left(\varepsilon_i\right) = \tau_0^2 + \sigma^2 \\
&Cov\left(\tilde{Z}_{g,i} \cdot \tilde{Z}_{g,j}\right) = E\left(\tilde{Z}_{g,i} \cdot \tilde{Z}_{g,j}\right) - E\left(\tilde{Z}_{g,i}\right) \cdot E\left(\tilde{Z}_{g,j}\right) \\
&= \left[E\left(\mu_g^2\right) + E\left(\mu_g \cdot \varepsilon_j\right) + E\left(\mu_g \cdot \varepsilon_i\right)\right] + E\left(\varepsilon_i \cdot \varepsilon_j\right) \\
&= E\left(\mu_g^2\right) - \mu_0^2 \\
&= Var(\mu_g) \\
&= \tau_0^2,
\end{aligned}
$$

(2.20)

con $i \neq j$.

In conclusione la variabile n-dimensionale $\widetilde{Z}_g$ è una normale multipla con la seguente struttura:

$$
\left(\widetilde{Z}_{g,i}\right)_{n\times 1} = \begin{pmatrix} \widetilde{Z}_{g,1} \\ \widetilde{Z}_{g,2} \\ \vdots \\ \widetilde{Z}_{g,n} \end{pmatrix} \sim N_n \left(\begin{pmatrix} \mu_0 \\ \mu_0 \\ \vdots \\ \mu_0 \end{pmatrix}, \begin{pmatrix} \sigma^2 + \tau_0^2 & \tau_0^2 & \cdots & \tau_0^2 \\ \tau_0^2 & \sigma^2 + \tau_0^2 & \cdots & \tau_0^2 \\ \vdots & \vdots & \ddots & \\ \tau_0^2 & \tau_0^2 & \cdots & \sigma^2 + \tau_0^2 \end{pmatrix} \right)
$$

In forma compatta,

$$
\left(\widetilde{Z}_{g,i}\right)_{n\times 1} \sim N_n\left(\mu_0, \Sigma_n\right),
$$

con $\underline{\mu_0} = (\mu_0, \ldots, \mu_0)^{\mathrm{T}}$ e $\Sigma_n = \sigma^2 I_n + \tau_0^2 M_n$, con I_n matrice identità $n \times n$ e M_n di tutti 1.

Con questi risultati si possono calcolare le probabilità a posteriori di qualsiasi *pattern* di espressione secondo la formula 2.10 e le misure di *odds* relative.

Nel paragrafo 2.4 viene descritto il metodo per ottenere la stima dei parametri $\theta = (\mu, \sigma^2, \tau_0)$ e del quarto parametro p, indicante la probabilità a priori di geni differenzialmente espressi.

2.3 Stima dei parametri

L'approccio utilizzato nel seguente elaborato è l'approccio Bayesiano empirico. In tale approccio, i parametri che caratterizzano il modello mistura di tipo gerarchico sono tutti stimati utilizzando i dati a disposizione. Per i modelli GG e LNN discussi in 2.3.1 e 2.3.2, il parametro θ è stimato con il metodo della massima verosimiglianza marginale. Nel modello GG, va stimato il parametro $\theta = (\alpha, \alpha_0, \nu)$ e nel modello LNN il parametro $\theta = (\mu, \sigma^2, \tau_0)$. Inoltre devono essere stimate le proporzioni di mistura del modello gerarchico, ovvero le probabilità a priori p_k di appartenenza al *k*-simo *pattern* da parte del gene *g*. Il calcolo dei parametri può essere ottenuto mediante l'algoritmo EM e la funzione di ottimizzazione numerica *optim* di R per stimare θ.

Dato il vettore di dati d_g governato dal modello mistura della forma 2.9, viene introdotto un indicatore di *pattern* $\phi_{g,l}$ definito uguale a uno se il *pattern* di espressione del gene g è l e zero altrimenti. La verosimiglianza e la log-verosimiglianza per i dati completi $(d_g, \phi_{g,l})$ sono date dalle seguenti espressioni:

$$L_c(\theta) = L_c(\theta; d_g, \underline{\phi}_g) = \prod_g \prod_{k=0}^{m} \left[p_k f_k(d_g) \right]^{\underline{\phi}_{g,k}}$$

$$l_c(\theta) = l_c(\theta; d_g, \underline{\phi}_g) = \sum_g \log \prod_{k=0}^{m} \left[p_k f_k(d_g) \right]^{\underline{\phi}_{g,k}} = \sum_g \sum_{k=0}^{m} \underline{\phi}_{g,k} \log \left[p_k f_k(d_g) \right] = \sum_g \sum_{k=0}^{m} \underline{\phi}_{g,k} \left[\log p_k + \log f_k(d_g) \right]$$

Il passo E dell'algoritmo EM prevede, fissato un valore θ_0 per il parametro θ, di calcolare il valore atteso di $l_c(\theta)$ condizionato ai dati osservati e a θ_0, ottenendo:

$$\hat{l}_c(\theta) = \sum_g \sum_{k=0}^{m} \underline{\phi}_{g,k} \left[\log p_k + \log f_k(d_g) \right]$$

$\hat{\phi}_{g,l}$ è la probabilità a posteriori del *pattern l* per il gene *g*, ovvero:

$$\Pr(l \mid d_g) = \frac{p_l f_l(d_g)}{\sum_{k=0}^{m} p_k f_k(d_g)}$$

con θ_0 che parametrizza la densità f_k.

Per la stima di p_k viene utilizzata la media aritmetica $\hat{\phi}_{g,l}$, relativa a tutti i geni. Il passo M dell'algoritmo provvede a stimare θ mediante massimizzazione numerica di $\hat{l}_c(\theta)$. Il processo è ripetuto fino ad ottenere la convergenza delle stime. E' opportuno controllare i risultati per diversi valori di θ_0.

Capitolo 3

Simulazioni Con *EBarrays*

Introduzione

Le simulazioni che vengono proposte nei paragrafi che seguono, servono per valutare il comportamento dei due modelli presi in considerazione (Gamma-Gamma e LogNormale-Normale) e la capacità di questi di identificare correttamente l'espressione genica. Il software statistico utilizzato in questo contesto è *R* ed il codice delle funzioni appositamente create è riportato in appendice A3.2. Relativamente al programma utilizzato, è stata utilizzata la libreria *EBarrays* (*Kendziorski* et al., *2006,* [rif. 8]), che implementa le principali funzioni dei metodi bayesiani empirici parametrici. Nel corso dell'elaborato sono state utilizzate le principali funzioni della libreria, ampiamente discusse nell'*help* di *R*, in particolare:

- *emfit*: stima dei parametri del modello GG o LNN attraverso l'uso dell'algoritmo EM;
- *postprob:* calcolo delle probabilità a posteriori dei diversi *pattern* di espressione;
- *plotMarginal:* produzione dei grafici della distribuzione marginale del modello scelto e della curva della densità marginale dei dati stimata con il metodo del nucleo;
- *ebPatterns:* generazione dei *pattern* di espressione da utilizzare per input alla funzione *emfit*;

- *ceckCV:* generazione dei grafici per l'analisi diagnostica relativa all'assunzione del coefficiente di variazione costante;
- *ceckModel:* produzione dei *Q-Q – plot* per l'analisi grafica diagnostica.

3.1 Campioni sbilanciati

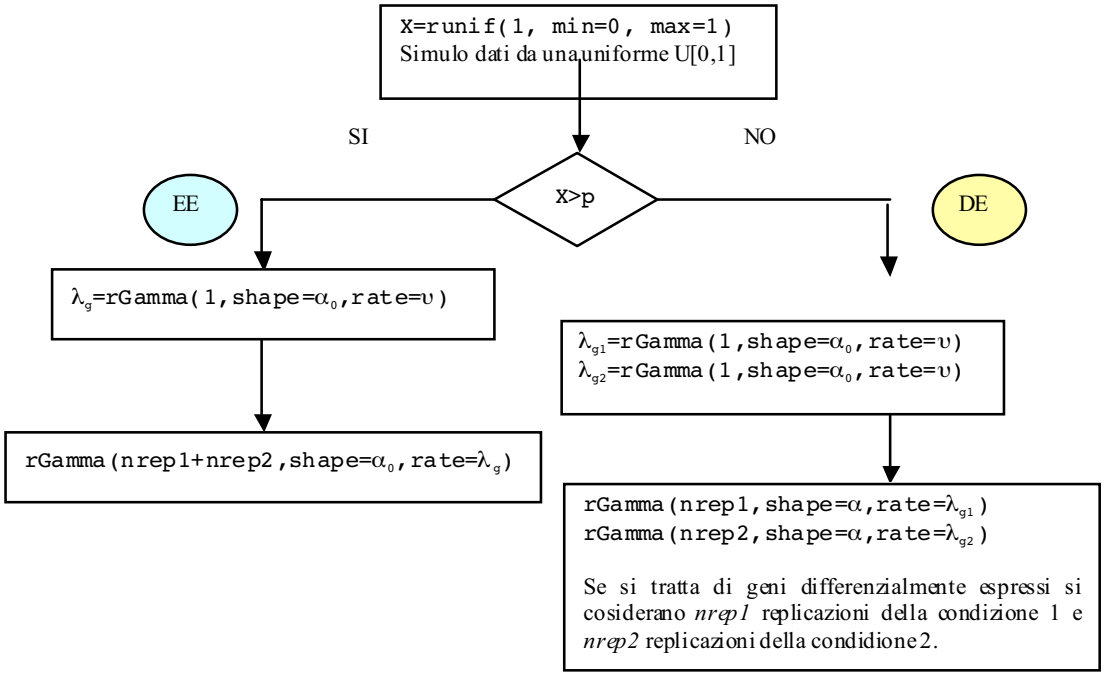
In letteratura, l'efficacia dei metodi bayesiani empirici è stata studiata nel caso di campioni bilanciati, come per esempio le simulazioni riportate nella tesi di Lise M. (2004) [rif. 9].

Con lo scopo di in due condizioni non bilanciate, quindi, sono proposte, per entrambi i modelli, due simulazioni di 10.000 geni sotto 2 condizioni, con 3 repliche per la prima condizione e 15 repliche per la seconda condizione. Nella pratica, la scelta del numero di repliche per ogni condizione deriva dalla casistica dei dati provenienti da *microarray*. Le due condizioni, infatti, si riferiscono allo stato di due gruppi di cellule: cellule sane e cellule malate, cellule trattate contro cellule non trattate, oppure caratterizzanti due tipologie della malattia oggetto di studio. Caratteristica peculiare dell'elaborato consiste nel considerare campioni sbilanciati sotto le 2 condizioni, al fine di tenere in considerazione che nella casistica reale sono rari i casi in cui i due campioni sono perfettamente bilanciati.

In un primo momento si sono impostati i parametri dei modelli per le simulazioni simili a quelli utilizzati nella letteratura da *Newton* et al. [rif. 5]: $\theta = (\alpha, \alpha_0, \upsilon) = (10, 0.9, 0.5)$ per il modello Gamma-Gamma e $\theta = (\mu_0, \sigma, \tau) = (2.3, 0.3, 1.39)$ per il modello LogNormale-Normale. I due modelli sono stati poi riparametrizzati con l'utilizzo dei valori proposti da *Chiogna* et al. [rif. 10]: $\theta = (\alpha, \alpha_0, \upsilon) = (1, 1.1, 45.4)$ per il modello Gamma-Gamma e

$\theta = (\mu_0, \sigma, \tau) = (6.58, 0.9, 1.13)$ per il modello LogNormale-Normale. I risultati verranno riportati in appendice A3.1.

Sono state quindi simulate 10.000 espressioni geniche secondo il modello GG dalla funzione *sim2GG* proposta in appendice A3.2.1.1 seguendo il diagramma riportato nello Schema 3.1.



Schema 3.1. Schema della simulazione del modello Gamma-Gamma. Tali operazioni sono ripetute 10000 volte per simulare 10000 geni sotto 2 condizioni sperimentali, con i parametri *nrep1* pari a 3 e *nrep2* a 15 ed indicano il numero di repliche per ogni condizione. Il parametro p è stato settato pari a 0.2, gli alti parameri $\theta = (\alpha, \alpha_0, \upsilon) = (10, 0.9, 0.5)$.

Per ogni singolo gene è stata calcolata la media delle 3 replicazioni sotto la prima condizione e la media delle 15 replicazioni sotto la seconda condizione. Questa

operazione ha reso possibile la costruzione del grafico in Figura 3.1 che mostra i geni equivalentemente espressi e differenzialmente espressi ottenuti via simulazione mediante il modello Gamma-Gamma attraverso le istruzioni in appendice A3.2.1.2.

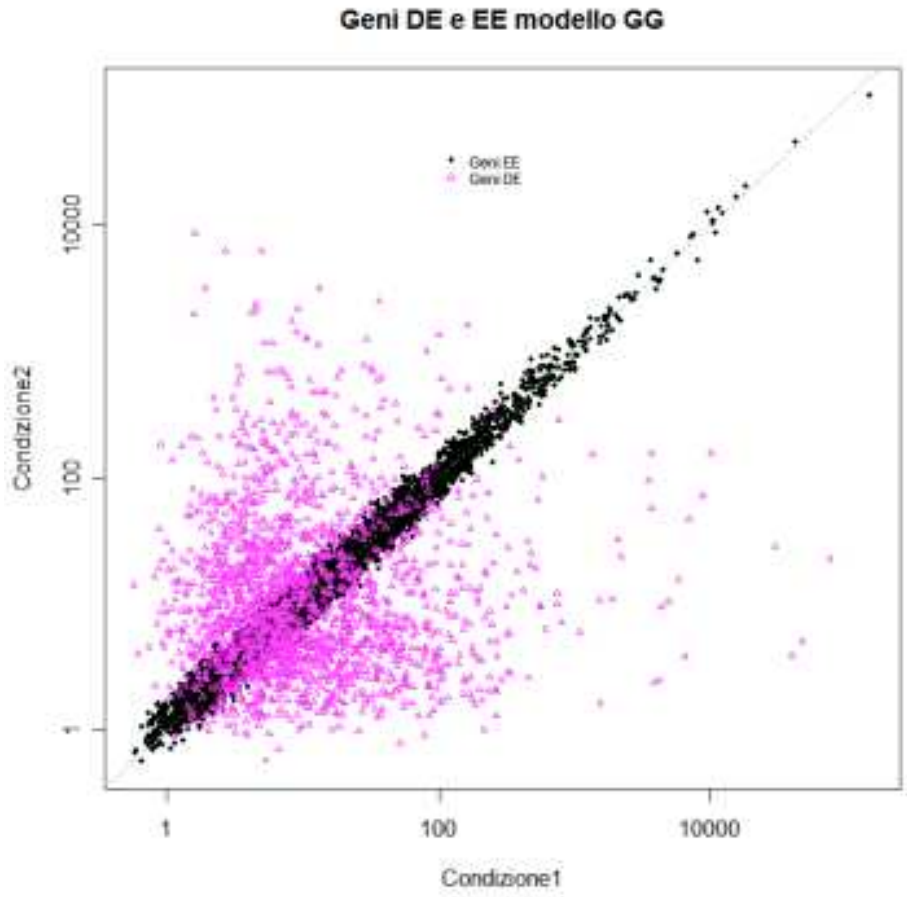
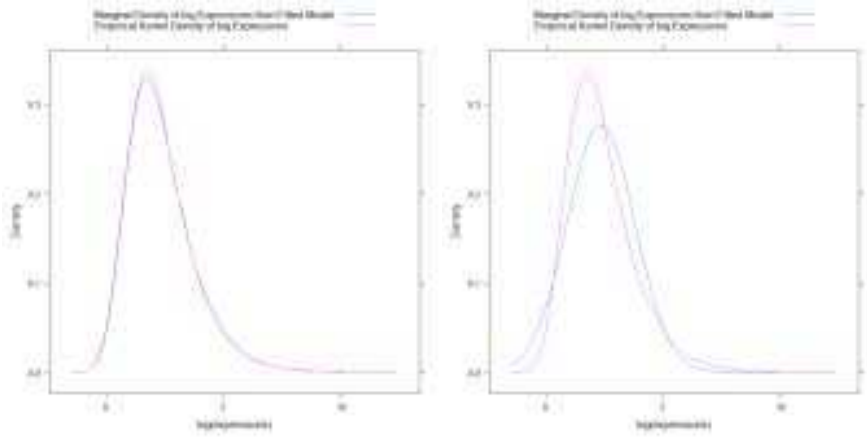


Figura 3.1. Corretta specificazione dell'espressione genica nel modello GG simulato.

Dato che lo scopo della simulazione è valutare quanto la bontà dell'inferenza sia influenzata dalla diversa numerosità dei gruppi, l'attenzione è stata focalizzata nella stima dei parametri dei modelli GG e LNN attraverso la funzione *emfit* della libreria *EBarrays* di *R* attraverso le istruzioni riportate in appendice A3.2.1.3.

Nella Figura 3.2 sono riportate le curve delle densità marginali dei logaritmi delle espressioni geniche simulate da GG ottenute mediante il metodo del nucleo sovrapposte alle curve delle densità marginali dei logaritmi delle espressioni ottenuta mediante i modelli imposti.

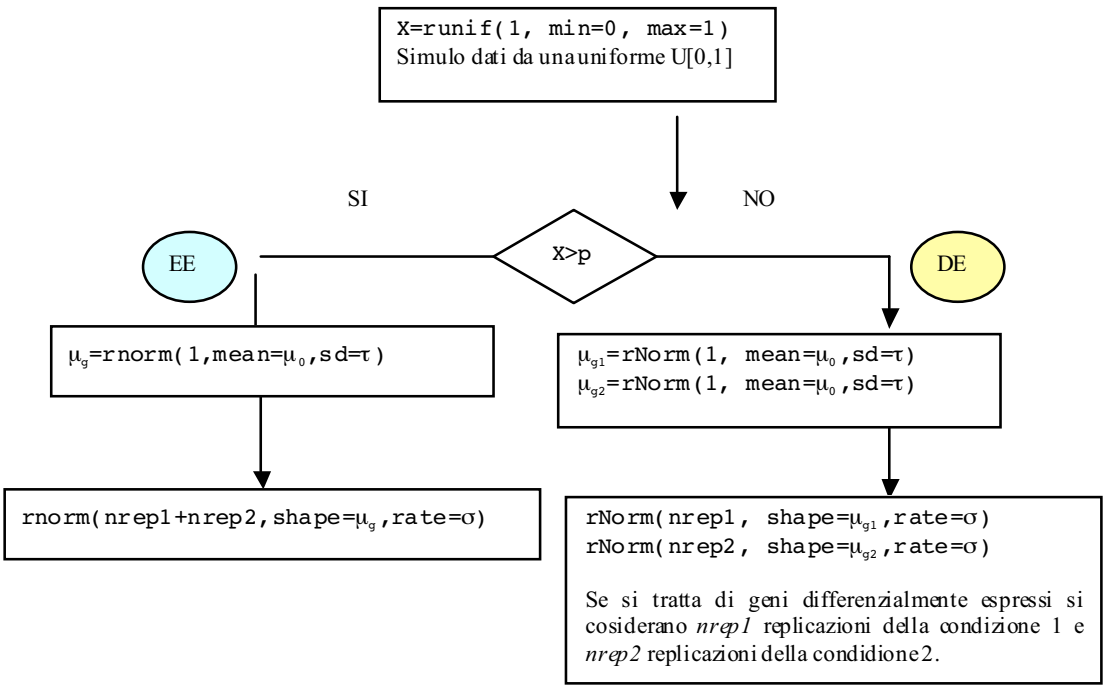


(a) (b)

Figura 3.2. (a) Dati simulati da modello GG e modellazione con GG. (b) Dati simulati con GG e modellazione con LNN.

I grafici riportati in Figura 3.2 dimostrano, com'è ovvio pensare, che dati simulati da un modello Gamma-Gamma sono descritti in maniera più accurata da una modellazione con Gamma-Gamma. Importante è valutare se l'assunzione di un modello parametrico rispetto ad un altro porta a sostanziali differenze nell'identificazione dei geni.

Per valutare l'effetto del modello parametrico scelto, è stata creata una funzione analoga alla precedente, riferita però al modello LogNormale – Normale: 10000 geni simulati sotto 2 condizioni, con 3 repliche per la prima condizione e 15 repliche per la seconda condizione con la funzione *sim2LNN*. Il diagramma nello Schema 3.2 mostra il procedimento per le simulazioni da un modello LogNormale – Normale seguito in appendice A3.2.1.4.



Schema 3.2. Schema della simulazione del modello LogNormale-Normale. Tali operazioni sono state ripetute 10000 volte per simulare 10000 geni sotto 2 condizioni sperimentali, con i parametri *nrep1* pari a 3 e *nrep2* a 15 ed indicano il numero di repliche per ogni condizione. Il parametro p è stato settato pari a 0.2, gli alti parameri $\theta = (\mu_0, \sigma, \tau) = (2.3,0.3,1.39)$.

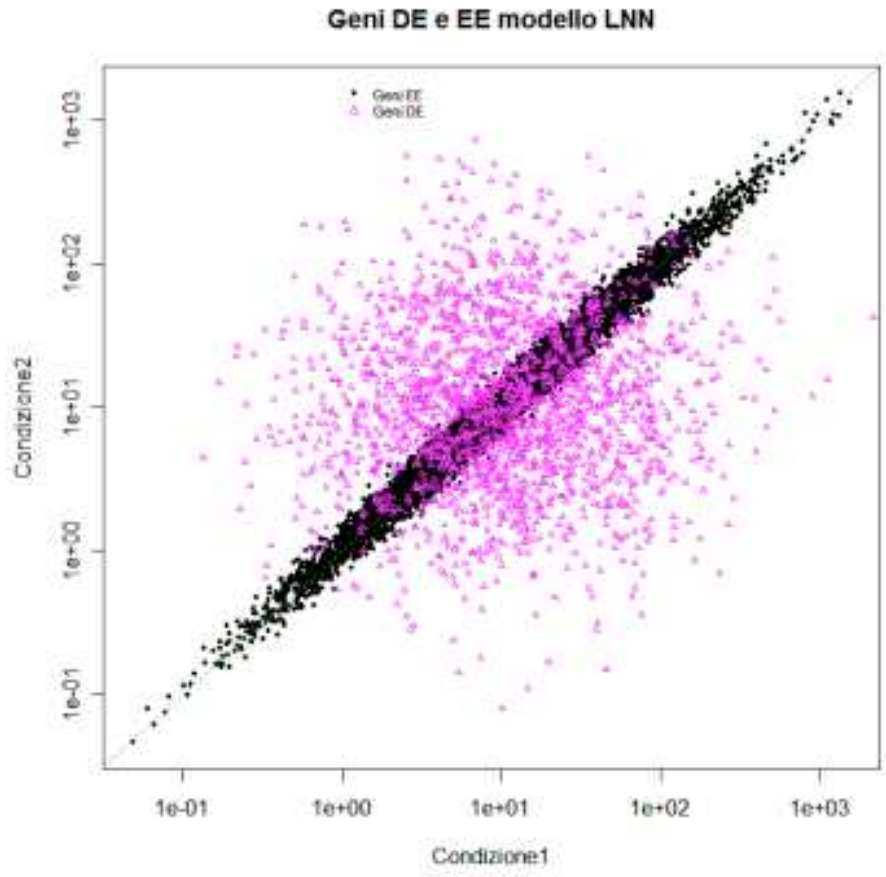
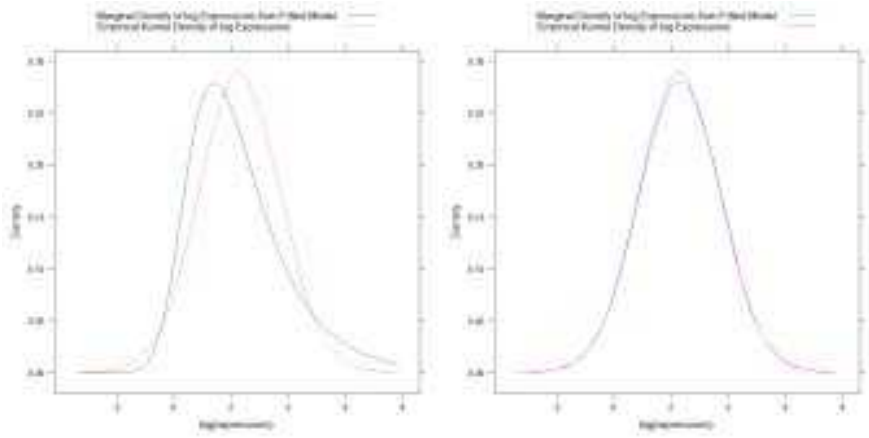


Figura 3.3. Corretta specificazione dell'espressione genica nel modello LNN simulato.

Ancora una volta è stata utilizzata la funzione *emfit* della libreria *EBarrays* per stimare i parametri dei modelli, ottenendo i grafici delle densità per valutare dell'adattabilità del modello (Figura 3.4 ottenuta attraverso l'implementazione in *R* delle operazioni in appendice A3.2.1.5). La conclusione ovvia, in questo caso, è che per dati simulati da una LogNormale, il modello che approssima meglio i dati è LNN.



(a) (b)

Figura 3.4. (a) Dati simulati da modello LNN e modellazione con LNN. (b) Dati simulati con LNN e modellazione con GG.

Per valutare quanto la bontà dell'inferenza sia influenzata dalla scelta del modello, è opportuno confrontare il comportamento delle densità empiriche rispetto alle densità teoriche in caso di non corretta specificazione del modello, quindi confrontare i grafici in Figura 3.2 (b) e 3.4 (a).
La sovrapposizione della densità empirica alla densità marginale stimata secondo il modello prefissato, indica quanto il modello teorico sia in grado di descrivere i dati simulati.
Utilizzando la probabilità a posteriori d'espressione genica differenziale, è stato considerato come differenzialmente espresso quel gene che presenta tale probabilità maggiore di 0.5.
A questo punto sono state create le Tabelle 3.1 (a) e (b) per i dati simulati da GG, le Tabelle 3.2 (a) e (b) per i dati simulati da LNN attraverso le operazioni descritte in appendice A3.2.1.9 e A3.2.1.10 che mostrano le tabelle di confusione per le due simulazioni nelle due modellazioni.

	Espressione prevista da GG	
Espressione esatta	Equivalente	Differente
Equivalente	7992	60
Differente	449	1499

(a)

	Espressione prevista da LNN	
Espressione esatta	Equivalente	Differente
Equivalente	7997	55
Differente	469	1479

(b)

Tabella 3.1. (a) Tabella di corretta e non corretta identificazione dell'espressione genica con il modello GG (dati simulati da GG). (b) Tabella di corretta e non corretta identificazione dell'espressione genica con il modello LNN (dati simulati da GG).

	Espressione prevista da GG	
Espressione esatta	Equivalente	Differente
Equivalente	7962	54
Differente	505	1480

(a)

	Espressione prevista da LNN	
Espressione esatta	Equivalente	Differente
Equivalente	7955	61
Differente	494	1491

(b)

Tabella 3.2. (a) Tabella di corretta e non corretta identificazione dell'espressione genica con il modello GG (dati simulati da LNN). (b) Tabella di corretta e non corretta identificazione dell'espressione genica con il modello LNN (dati simulati da LNN).

Ancora una volta le prime considerazioni dal confronto dei due modelli nelle due simulazioni conducono ad un risultato ovvio: è evidente una migliore adattabilità ai dati dei modelli correttamente specificati.

Su 8052 geni simulati da GG come equivalentemente espressi, ne sono stati identificati correttamente 7992 (99%) dal modello GG e 7997 (99%) dal modello LNN; su 1948 geni simulati da GG come differenzialmente espressi ne sono stati identificati correttamente 1499 (77%) dal modello GG e 1479 (76%) dal modello LNN.

Su 8016 geni simulati da LNN come equivalentemente espressi ne sono stati identificati correttamente 7962 (99%) dal modello GG e 7955 (98%) dal modello LNN; su 1985 geni simulati da LNN come differenzialmente espressi ne sono stati identificati correttamente 1480 (75%) dal modello GG e 1491 (75%) dal modello LNN.

Per non giungere ad un risultato ovvio, l'interesse è volto a confrontare i casi di non corretta specificazione dei modelli. A questo proposito sono state confrontate le percentuali di errata identificazione nelle due simulazioni. La percentuale di errata identificazione sul totale dei geni indicati come differenzialmente espressi, è pari a 24% per geni simulati da GG e 25% per geni simulati da LNN. La percentuale di errata identificazione sul totale dei geni indicati come equivalentemente espressi, è pari a 0.68% per geni simulati da GG e 0.67% per geni simulati da LNN.

E' da sottolineare la leggera inferiorità percentuale di errata identificazione sui geni simulati da GG e modellati con LNN. Si conferma quindi il risultato dell'analisi grafica preliminare: l'inferenza su dati simulati da LNN risente leggermente meno della non corretta specificazione del modello.

Nelle Figure 3.5 e 3.6 sono riportati i grafici che mostrano, con diversa colorazione, i geni indicati come differenzialmente espressi dai modelli variando il valore di probabilità a posteriori prima per i geni simulati da GG e poi dai geni simulati da LNN. Le istruzioni per ottenere tali grafici sono riportate in appendice A3.2.1.6.

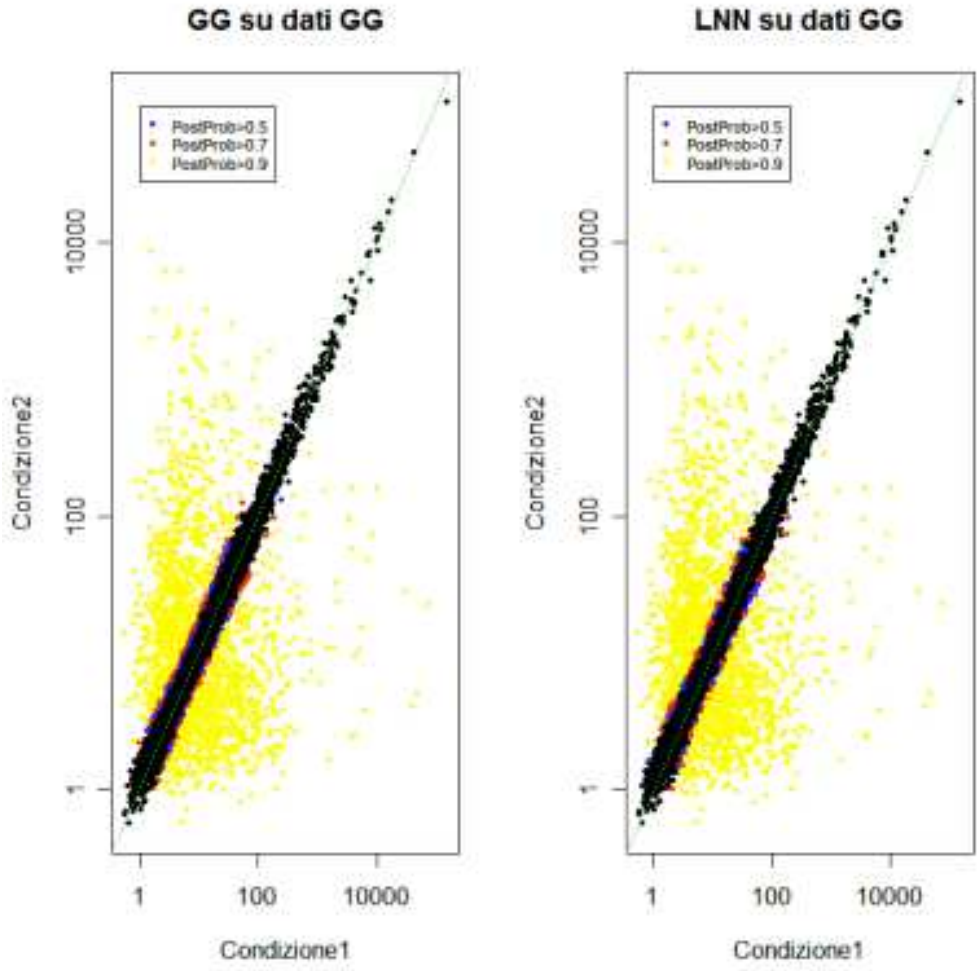


Figura 3.5 Geni identificati come differenzialmente espressi nei due modelli con diversa probabilità a posteriori (dati simulati da GG).

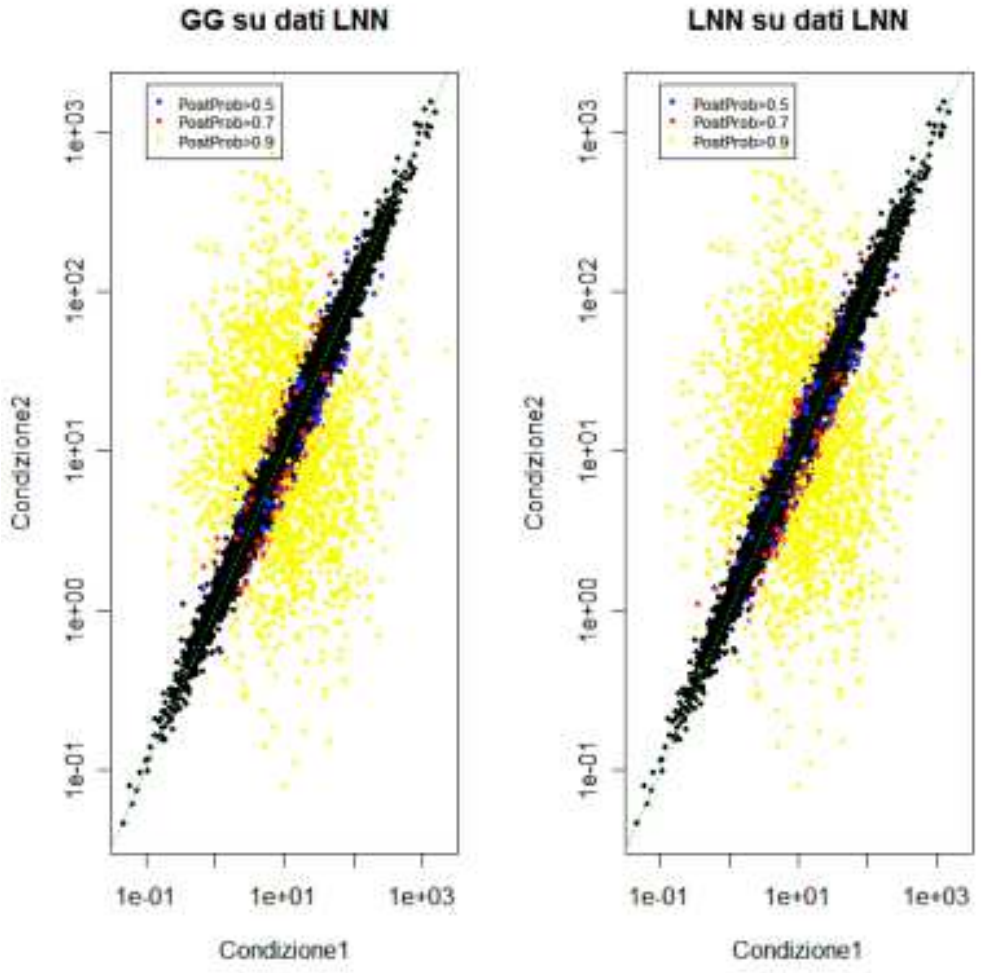
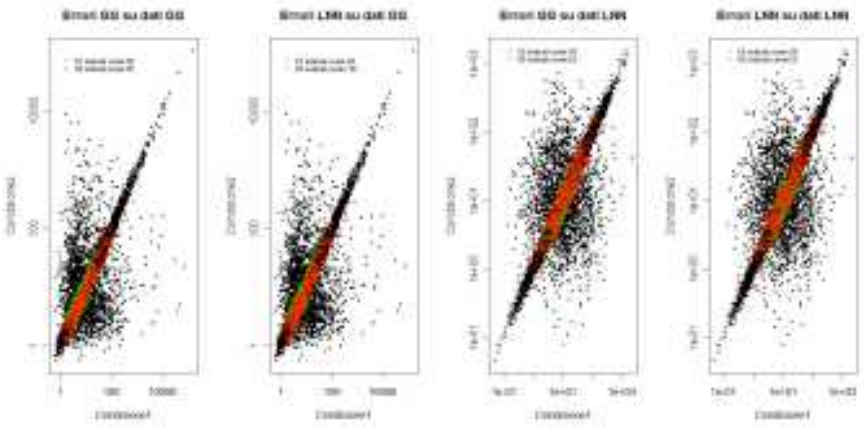


Figura 3.6 Geni identificati come differenzialmente espressi nei due modelli con diversa probabilità a posteriori (dati simulati da LNN).

Attraverso i grafici di Figura 3.5 e di Figura 3.6 si può notare che sia nel caso di geni simulati da GG, sia in quello di geni simulati da LNN, non vi è una sostanziale differenza tra i geni identificati come differenzialmente espressi nei due modelli in corrispondenza di pari probabilità a priori. Tale evidenza è mostrata dalle nuvole di punti colorati nello stesso modo nelle due simulazioni.

Una verifica analoga è stata fatta con la costruzione dei grafici di scorretta identificazione nelle due simulazioni per i due modelli presentati nella Figura 3.7 (a) e (b) con l'ausilio delle istruzioni riportate in appendice A3.2.1.12.



(a) (b)

Figura 3.7. (a) Geni non correttamente identificati dai due modelli (dati simulati da GG). (b) Geni non correttamente identificati dai due modelli (dati simulati da LNN).

Dai grafici in Figura 3.6 si giunge a conclusioni analoghe alle precedenti: non vi è una differenza molto significativa di errore di identificazione nel caso di corretta e di errata specificazione del modello; i geni non correttamente identificati sono gli stessi sia nel caso di corretta specificazione del modello sia nel caso di errata specificazione del modello.

Per quanto riguarda la simulazione di dati GG, è stato costruito il grafico degli *odds* a posteriori, ossia al grafico già presentato in Figura 3.1 che raffigura la corretta specificazione dell'espressione genica nel modello GG simulato, si sono sovrapposte le curve di livello degli *odds* pari a 1, 10 e 100. I livelli indicano rispettivamente che geni esterni a tali curve presentano una probabilità a posteriori

di espressione differenziale di 1 ,10 o 100 volte la probabilità a posteriori di espressione equivalente. Com'è possibile notare dal grafico in Figura 3.7, ottenuto attraverso le istruzioni in appendice A3.2.1.13, le curve degli *odds* non sono rette ma le regioni comprese tra le curve sono più ampie per livelli di intensità bassa e alta nelle due condizioni. Questa caratteristica deriva da una proprietà peculiare della distribuzione Gamma che fa in modo che l'*odds* sia funzione del valore complessivo dell'espressione genica nelle 2 condizioni.

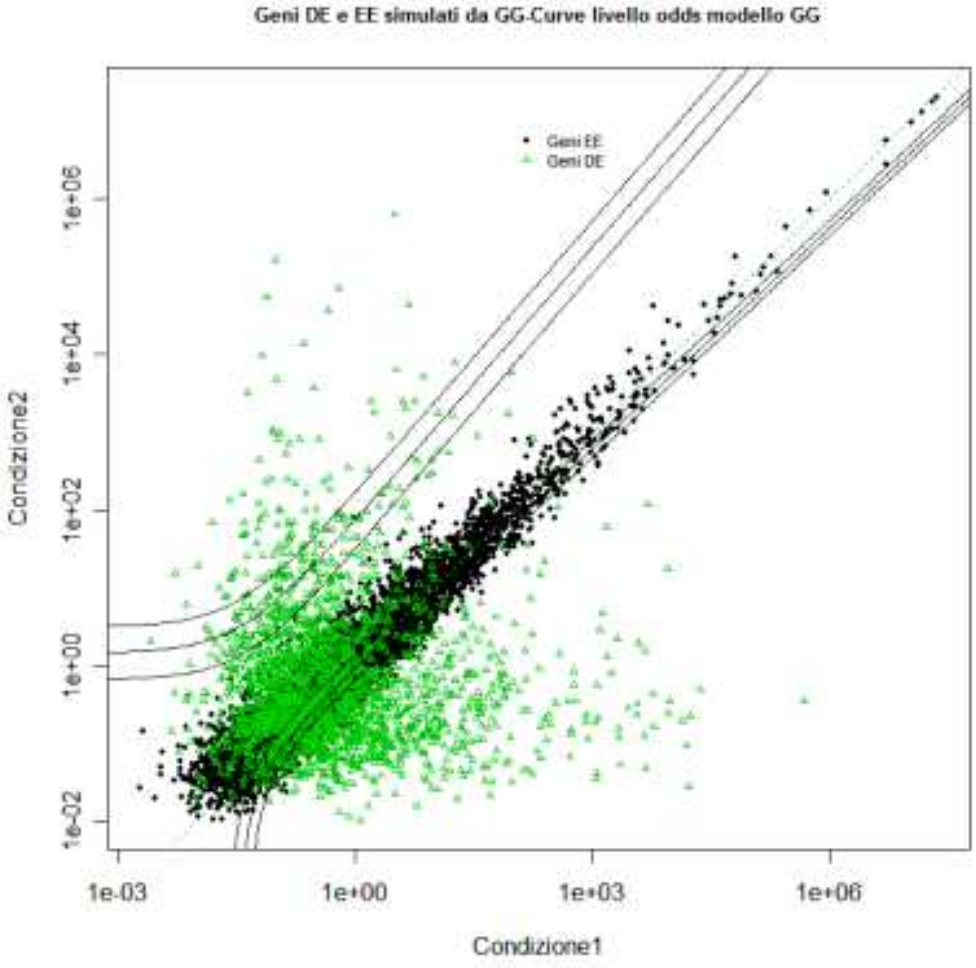


Figura 3.8. Curve di livello degli *odds* calcolate mediante il modello GG. Corrispondono agli *odds* di 1, 10 e 100 rispettivamente dalla curva più interna alla curva più esterna. I punti compresi tra le due curve più interne rappresentano i

geni identificati come equivalentemente espressi dal modello GG. I punti indicati con i triangolini rappresentano i geni simulati come differenzialmente espressi.

3.2 Indicatori di bontà dei modelli e stima dei parametri.

I modelli GG e LNN utilizzati per l'identificazione dell'espressione genica riescono ad individuare correttamente una parte dei geni equivalentemente o differenzialmente espressi, commettendo degli errori rispetto all'effettiva espressione genica. Sono stati utilizzati alcuni indici presenti nella letteratura atti a valutare la bontà attraverso la capacità del modello stesso.
Sono state fatte quindi delle simulazioni per il calcolo degli indici di sensitività (*sensitivity*), specificità (*specificity)*, previsione positiva dei valori (*PPV: positive predictive value*), previsione negativa dei valori (*NPV: negative predictive value*) e indice corrispondente all'errore di primo tipo (*FDR: false discovery rate*).

	Espressione prevista da modello	
Espressione esatta	Equivalente	Differente
Equivalente	a	b
Differente	c	d

Tabella 3.3. Tabella che indica come vengono identificati i geni simulati.

Utilizzando la dicitura della Tabella 3.3 cerchiamo di comprendere in dettaglio gli indicatori di qualità elencati sopra.

- *Sensitivity (sens) = d/(c+d)*: corrisponde alla frazione dei geni differenzialmente espressi correttamente specificati dal modello.
- *Specificity (spec) = a/(a+b)*: è la frazione di geni equivalentemente espressi correttamente specificati.

- *Positive predictive value (PPV) = d/(b+d)*: è la frazione di geni identificati come differenzialmente espressi ed effettivamente tali.
- *Negative predictive value (NPV) = a/(a+c)*: è la frazione di geni identificati come equivalentemente espressi ed effettivamente tali.
- *False discovery rate (FDR) = b/(b+d)*; rapporto tra il numero di falsi positivi identificati dal modello come differenzialmente espressi essendo equivalentemente espressi, e il numero di geni identificati come differenzialmente espressi dal modello.

La simulazione è stata fatta variando la probabilità a priori *p*, relativa ai geni differenzialmente espressi, da 0.1 a 0.5 con incrementi di 0.1. Per ogni proporzione sono stati simulati 10 *dataset*, ognuno con 1000 geni in 2 condizioni, con 3 repliche per la prima condizione e 4 repliche per la seconda. I dati simulati da GG e da LNN sono stati modellati con entrambe le strutture parametriche e gli indicatori sono stati calcolati facendo la media dei valori degli indici per le 10 ripetizioni effettuate. Nelle tabelle 3.4, 3.5, 3.6 e 3.7 sono riportati gli indici di sensibilità calcolati come detto in precedenza. Inoltre, tra parentesi, sono riportate gli *standard error* nelle 10 ripetizioni. Le operazioni per la costruzione delle tabelle sono riportate in appendice A3.2.2.1 e A3.2.2.2.

p	0.1	0.2	0.3	0.4	0.5
Sens	0.235(0.044)	0.371(0.037)	0.448(0.045)	0.503(0.043)	0.540(0.069)
Spec	0.993(0.006)	0.981(0.005)	0.954(0.011)	0.922(0.025)	0.821(0.065)
PPV	0.818(0.116)	0.834(0.032)	0.804(0.029)	0.816(0.038)	0.763(0.049)
NPV	0.922(0.006)	0.859(0.012)	0.806(0.014)	0.737(0.013)	0.638(0.018)
FDR	0.182(0.116)	0.166(0.032)	0.196(0.029)	0.184(0.038)	0.237(0.049)

Tabella 3.4. Indici di bontà del modello GG su dati simulati dal modello GG; le stime sono ottenute dalla media delle 10 ripetizioni, tra parentesi indicato lo *standard error*.

p	0.1	0.2	0.3	0.4	0.5
Sens	0.320(0.072)	0.388(0.058)	0.555(0.051)	0.482(0.103)	0.632(0.036)
Spec	0.976(0.012)	0.949(0.011)	0.913(0.025)	0.925(0.052)	0.769(0.040)
PPV	0.605(0.099)	0.703(0.034)	0.739(0.059)	0.826(0.067)	0.729(0.025)
NPV	0.932(0.014)	0.883(0.011)	0.827(0.016)	0.733(0.032)	0.682(0.018)
FDR	0.395(0.099)	0.297(0.034)	0.261(0.059)	0.174(0.066)	0.271(0.025)

Tabella 3.5. Indici di bontà del modello GG su dati simulati dal modello LNN; le stime sono ottenute dalla media delle 10 ripetizioni, tra parentesi indicato lo *standard error*.

p	0.1	0.2	0.3	0.4	0.5
Sens	0.320(0.072)	0.388(0.058)	0.555(0.051)	0.482(0.103)	0.632(0.036)
Spec	0.976(0.012)	0.949(0.011)	0.913(0.025)	0.925(0.052)	0.769(0.040)
PPV	0.605(0.099)	0.703(0.034)	0.739(0.059)	0.826(0.067)	0.729(0.025)
NPV	0.932(0.014)	0.883(0.011)	0.827(0.016)	0.733(0.032)	0.682(0.018)
FDR	0.395(0.099)	0.297(0.034)	0.261(0.059)	0.174(0.066)	0.271(0.025)

Tabella 3.6. Indici di bontà del modello LNN su dati simulati dal modello GG; le stime sono ottenute dalla media delle 10 ripetizioni, tra parentesi indicato lo *standard error*.

p	0.1	0.2	0.3	0.4	0.5
Sens	0.224(0.029)	0.308(0.029)	0.395(0.042)	0.441(0.038)	0.509(0.064)
Spec	0.983(0.008)	0.977(0.006)	0.951(0.010)	0.921(0.019)	0.803(0.056)
PPV	0.608(0.103)	0.774(0.044)	0.771(0.039)	0.790(0.029)	0.728(0.038)
NPV	0.920(0.006)	0.846(0.011)	0.791(0.015)	0.713(0.015)	0.617(0.016)
FDR	0.391(0.103)	0.226(0.044)	0.229(0.039)	0.210(0.029)	0.272(0.038)

Tabella 3.7. Indici di bontà del modello LNN su dati simulati dal modello LNN; le stime sono ottenute dalla media delle 10 ripetizioni, tra parentesi indicato lo *standard error*.

Grazie alle tabelle così costituite si è in grado di procedere con un'analisi più approfondita della *performance* dei due modelli. Per mettere a confronto in maniera più immediata i 2 modelli attraverso gli indici sopra proposti, si sono costruiti i grafici in Figura 3.8 e Figura 3.9 (A3.2.2.3) che rappresentano rispettivamente gli indici di bontà del modello GG e del modello LNN sui dati simulati al variare della probabilità a priori d'espressione genica differenziale. Si sono utilizzati due colori differenti per distinguere le simulazioni: il colore più scuro è stato utilizzato nel caso di corretta specificazione del modello mentre quello più chiaro in caso contrario.

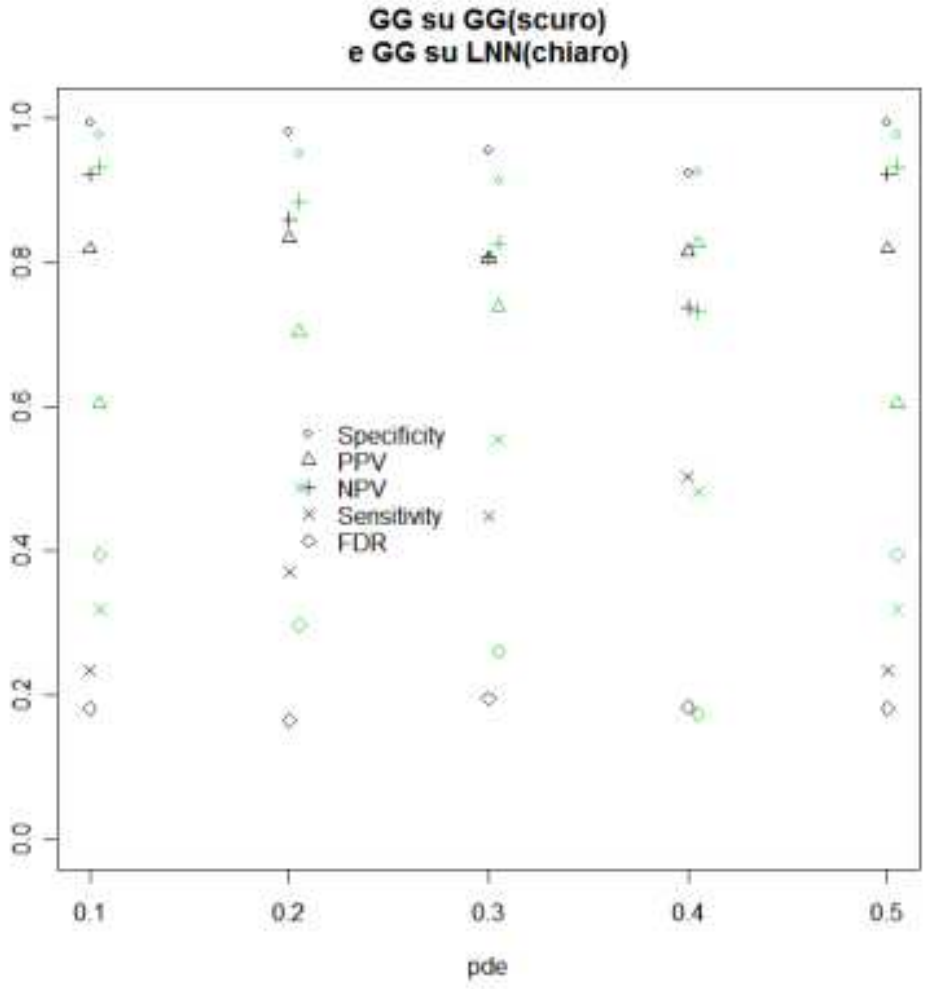


Figura 3.9. Indicatori di bontà del modello GG su dati simulati al variare della probabilità a priori di espressione genica differenziale. I caratteri più scuri indicano gli indicatori del modello GG su dati simulati da GG, mentre caratteri chiari si riferiscono al modello GG su dati simulati da LNN.

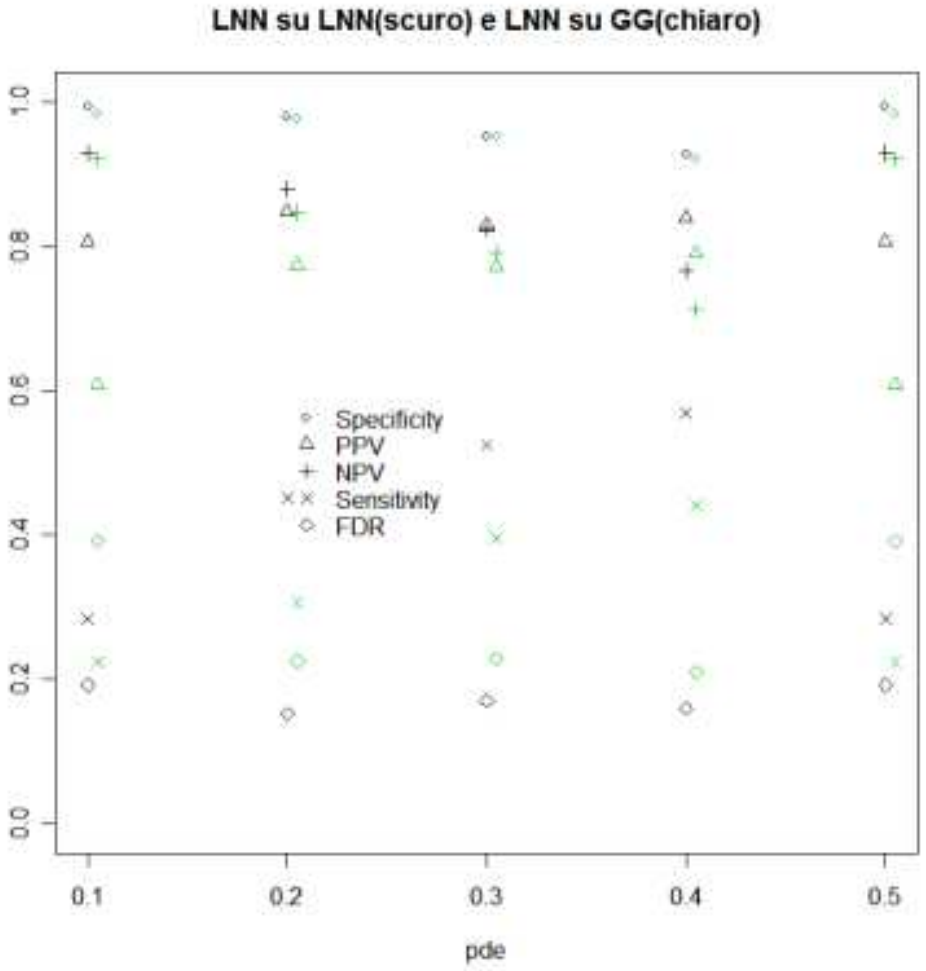


Figura 3.10. Indicatori di bontà del modello LNN su dati simulati al variare della probabilità a priori di espressione genica differenziale. I caratteri più scuri indicano gli indicatori del modello LNN su dati simulati da LNN, mentre caratteri chiari si riferiscono al modello LNN su dati simulati da GG.

Si può notare che le *performance* dei due modelli sono ottimali nel caso di corretta specificazione del modello iniziale. Una caratteristica peculiare del modello LNN consiste nell'aumento della *performance* all'aumentare della probabilità a priori di geni differenzialmente espressi, in caso di non corretta specificazione. Indipendentemente dal modello generatore dei dati, le analisi

proposte mettono in evidenza che la scelta iniziale del modello Gamma-Gamma piuttosto che LogNormale-Normale non influisce in maniera significativa sulla capacità di corretta identificazione dei geni.

Sono state inoltre ripetute 10 simulazioni con 10000 geni per ogni simulazione in 2 condizioni con 3 repliche per la prima condizione e 15 repliche per la seconda condizione, utilizzando il modello GG con la funzione *sim4GG*, e il modello LNN con la funzione *sim4LNN*. I parametri relativi sono stati stimati mediante la funzione *emfit* della libreria *EBarrays* e nelle tabelle 3.8 e 3.9 sono riportate le stime con lo *standard error* relativo alle stime nelle 10 ripetizioni. Tutte le operazioni effettuate sono state riportate in appendice A3.2.2.4.

p	0.1	0.2	0.3	0.4	0.5
$\hat{\alpha}$	10.011(0.025)	10.010(0.001)	9.998(0.027)	10.003(0.040)	9.994(0.040)
$\hat{\alpha}_0$	0.899(0.014)	1.109(0.012)	0.903(0.012)	0.900(0.011)	0.899(0.008)
$\hat{\nu}$	0.500(0.009)	0.452(0.007)	0.403(0.006)	0.500(0.008)	0.499(0.008)
$\hat{p}$	0.101(0.004)	0.197(0.019)	0.301(0.007)	0.399(0.006)	0.500(0.008)

Tabella 3.8. Stime dei parametri del modello GG sui dati simulati dal modello GG con $(\alpha, \alpha_0, \upsilon) = (10, 0.9, 0.5)$ al variare di p. Tra parentesi è riportato lo *standard error*.

p	0.1	0.2	0.3	0.4	0.5
$\hat{\mu}_0$	2.293(0.013)	2.294(0.011)	2.292(0.014)	2.304(0.013)	2.302(0.012)
$\hat{\sigma}$	0.299(0.001)	0.300(0.001)	0.300(0.001)	0.300(0.001)	0.300(0.001)
$\hat{\tau}$	1.392(0.008)	1.388(0.010)	1.385(0.009)	1.389(0.008)	1.390(0.005)
$\hat{p}$	0.100(0.005)	0.203(0.004)	0.300(0.006)	0.397(0.006)	0.501(0.004)

Tabella 3.9. Stime dei parametri del modello LNN sui dati simulati dal modello LNN con (μ_0, σ, τ) = (2.3, 0.3, 1.39) al variare di *p*. Tra parentesi è riportato lo *standard error*.

3.3 Il Coefficiente di Variazione

Nella letteratura su analisi di espressioni geniche, si è riscontrato un argomento comune che riguarda il coefficiente di variazione. Molti studiosi sostengono che tale coefficiente possa essere ipotizzato costante. I modelli Gamma-Gamma e LogNormale-Normale presentati nei precedenti paragrafi ipotizzano un coefficiente di variazione costante per tutti i geni. Ciò che nel seguente paragrafo ci si propone di fare è di verificare tale ipotesi di costanza.

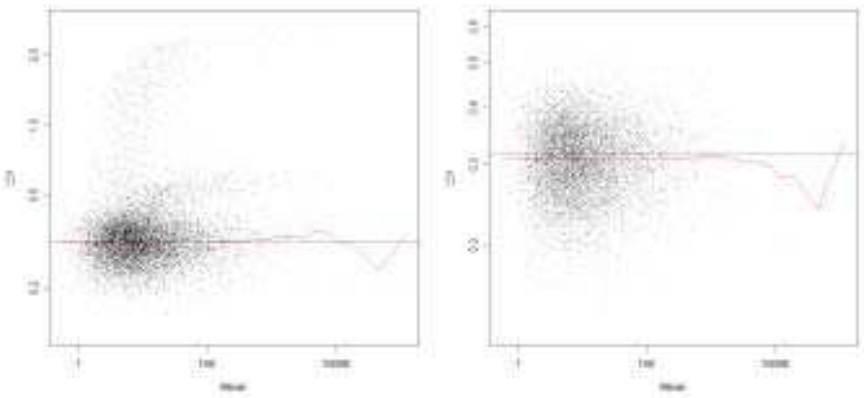
Per costruire i grafici in Figura 3.10 e 3.11 è stata utilizzata la funzione *checkCCV* della libreria *EBarrays* (in grado di costruire il grafico dei coefficienti di variazione relativo a 10000 simulazioni di geni secondo i modelli descritti nei paragrafi precedenti), in funzione della media relativa alle misure delle 3 replicazioni della prima condizione e delle 4 replicazioni della seconda

condizione. Nel grafico fornito dalla funzione, c'è una linea tratteggiata che rappresenta il CV teorico: per i dati simulati dal modello GG è pari a:

$$\frac{1}{\sqrt{\alpha}} = \frac{1}{10} \cong 0.32$$

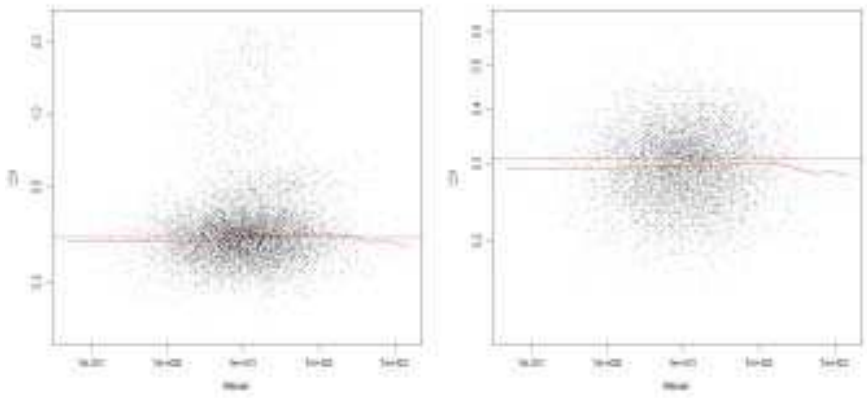
mentre nel modello per i dati simulati da LNN risulta:

$$\sqrt{\exp\{\sigma^2\} - 1} = \sqrt{\exp\{0.3^2\} - 1} \cong 0.31$$



(a) (b)

Figura 3.11. (a) Coefficiente di variazione della media per tutti i geni simulati da GG. (b) Coefficiente di variazione della media per i geni equivalentemente espressi simulati da GG.

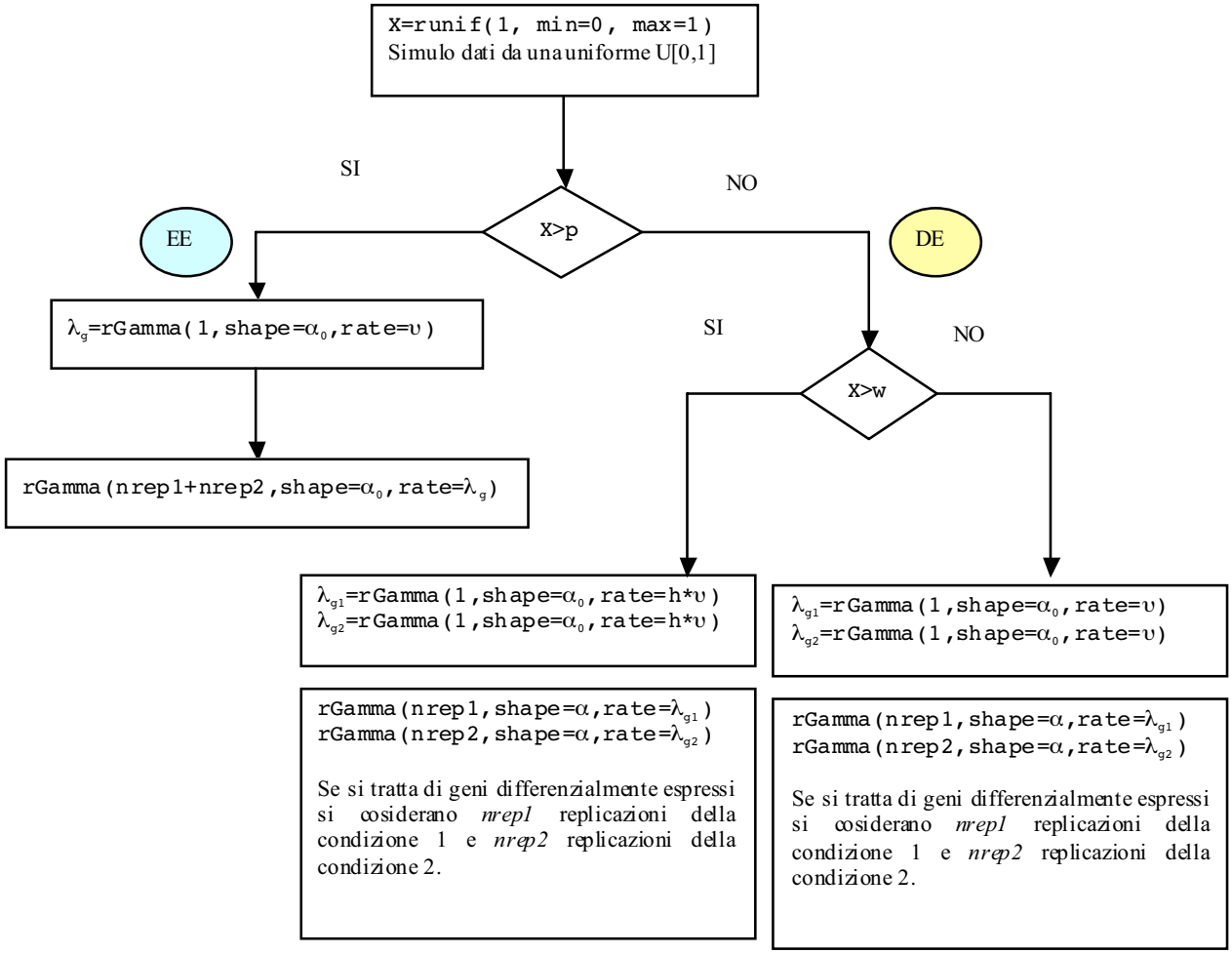


(a) (b)

Figura 3.12. (a) Coefficiente di variazione della media per tutti i geni simulati da LNN. (b) Coefficiente di variazione della media per i geni equivalentemente espressi simulati da LNN.

La curva continua inserita nei grafici in rosso è stata calcolata mediante un metodo di regressione non parametrica ed evidenzia che il coefficiente di variazione è mediamente costante al variare della media. Nonostante la notevole variabilità, il coefficiente varia tra 0.2 e 0.4 quindi si può ipotizzare costante per i dati di espressione genica. Le operazioni fatte per ottenere i grafici sono riportate in appendice A3.2.3.1.

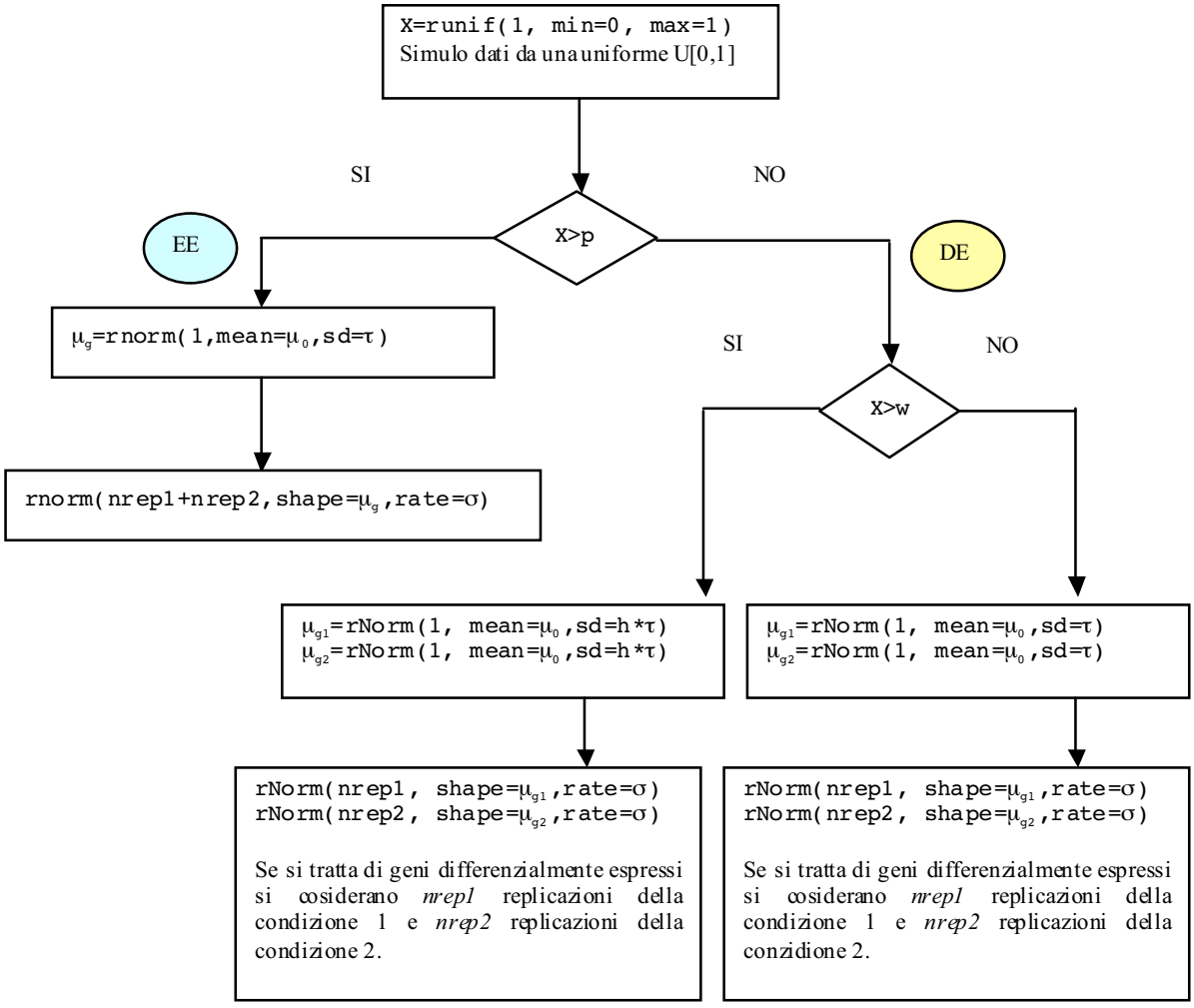
A questo punto proponiamo di valutare la robustezza dei 2 modelli in mancanza dell'assunzione di coefficiente di variazione costante, ossia stabilire come variano le *performance* dei due modelli negando tale ipotesi. Lo Schema 3.3 rappresenta la simulazione del modello Gamma-Gamma, implementato in *R* dalla funzione *simGGnotCostantCV* in appendice A3.2.3.2.



Schema 3.3. Schema della simulazione del modello Gamma-Gamma con coefficiente di variazione non costante. Tali operazioni sono ripetute 10000 volte per simulare 10000 geni sotto 2 condizioni sperimentali, con i parametri *nrep1* pari a 3 e *nrep2* a 15 ed indicano il numero di repliche per ogni condizione. Il parametro p è stato assegnato pari a 0.2, gli alti parameri $\theta = (\alpha, \alpha_0, \upsilon) = (10,0.9,0.5)$. I parametri w e h fanno sì che ci siano due distribuzioni distinte che descrivono i geni differenzialmente espressi. La funzione *simGGnotCostantCV* implementa la funzione i *R*.

In modo analogo sono state fatte le simulazioni di dati simulati da modello LogNormale-Normale con probabilità a priori *p* pari a 0.2, gli altri parametri sono stati fissati μ_0=2.3, σ=0.3, τ=1.39. Si è quindi costruito il grafico che mostra i geni equivalentemente e differenzialmente espressi.

Lo Schema 3.4 presenta le simulazioni implementate dalla funzione *simLNNnotConstantCV* in appendice A3.2.3.3.



Schema 3.4. Schema della simulazione del modello LogNormale-Normale. Tali operazioni sono ripetute 10000 volte per simulare 10000 geni sotto 2 condizioni

sperimentali, con i parametri *nrep1* pari a 3 e *nrep2* a 15 ed indicano il numero di repliche per ogni condizione. Il parametro *p* è stato assegnato pari a 0.2, gli alti parameri $\theta = (\mu_0, \sigma, \tau) = (2.3,0.3,1.39)$. I parametri *w* e *h* fanno sì che ci siano due distribuzioni distinte che descrivono i geni differenzialmente espressi. La funzione *simLNNnotCostantCV* implementa la funzione i *R*.

Variando i parametri *w* e *h,* è possibile costruire i grafici in Figura 3.13 e 3.14 che mostrano i geni simulati come equivalentemente e differenzialmente espressi rispettivamente attraverso il modello GG e LNN, l'istogramma del logaritmo dei dati d'espressione genica simulata con GG e LNN, il grafico di verifica dell'ipotesi di costanza del coefficiente di variazione dei dati in funzione della media impostando *h=1/100* e *w=1/10*. Le operazioni per ottenere tali grafici sono riportate in appendice A3.2.3.4.

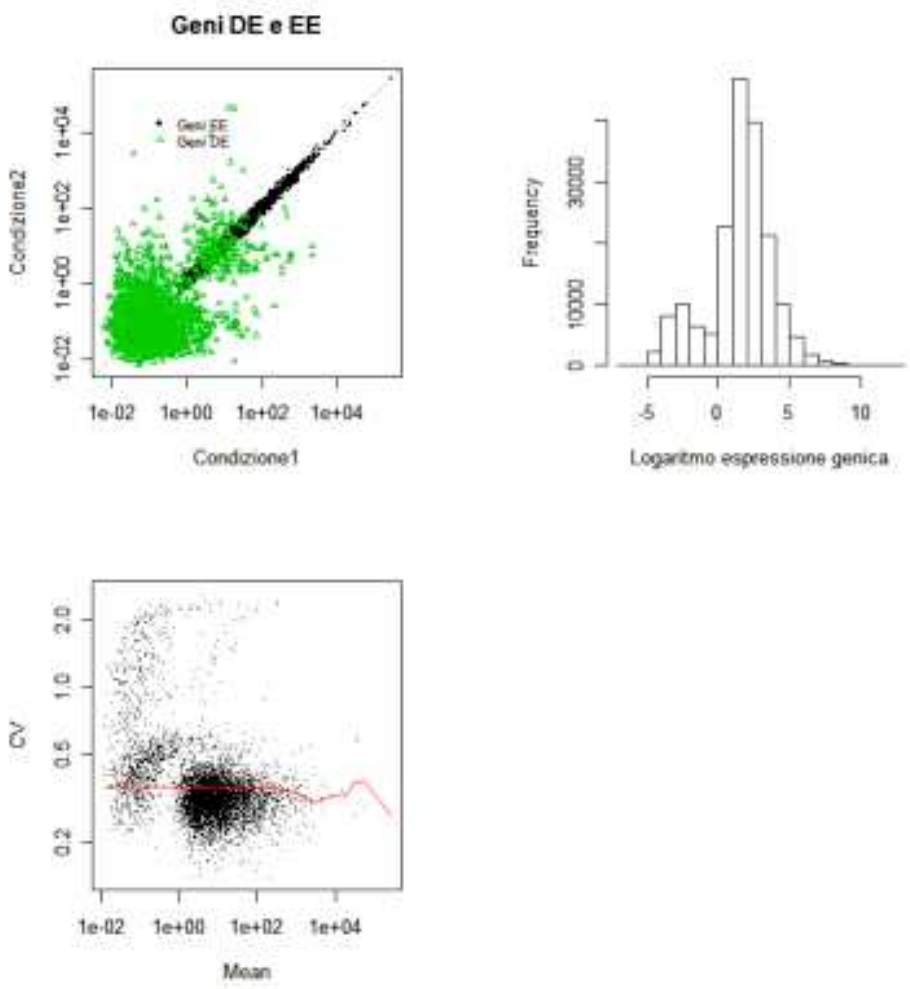


Figura 3.13. In alto a sinistra il grafico che rappresenta le simulazioni di geni dal modello Gamma-Gamma come equivalentemente e differenzialmente espressi con parametri *w* e *h* posti uguale rispettivamente a 1/10 e 1/100; in alto a destra l'istogramma del logaritmo dei dati di espressione genica simulati con GG. In basso il coefficiente di variazione dei dati in funzione della media.

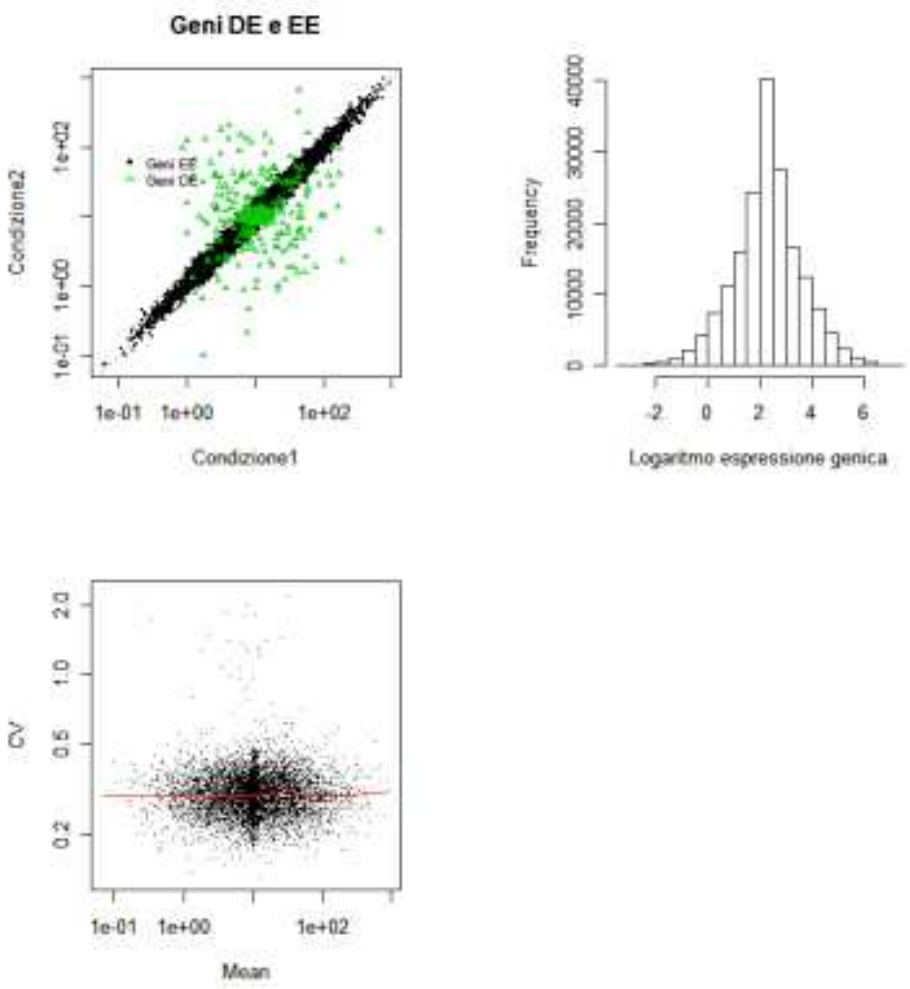
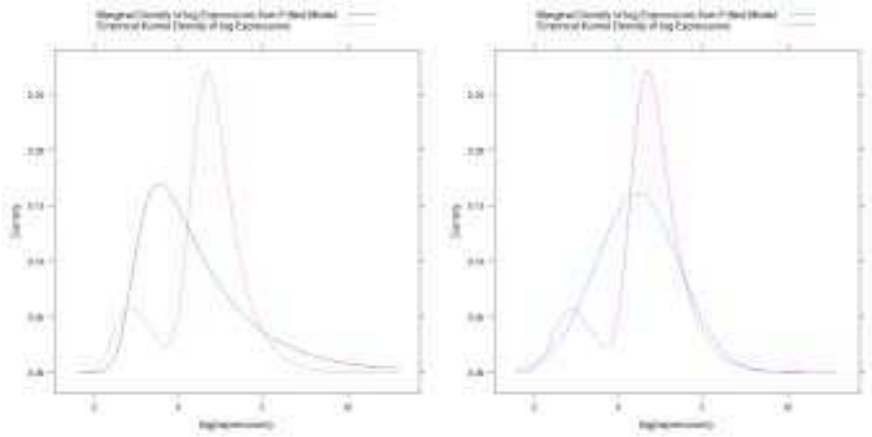


Figura 3.14. In alto a sinistra il grafico che rappresenta le simulazioni di geni dal modello LogNormale-Normale come equivalentemente e differenzialmente espressi con parametri *w* e *h* posti uguale rispettivamente a 1/10 e 1/100; in alto a destra l'istogramma del logaritmo dei dati di espressione genica simulati con LNN. In basso il coefficiente di variazione dei dati in funzione della media.

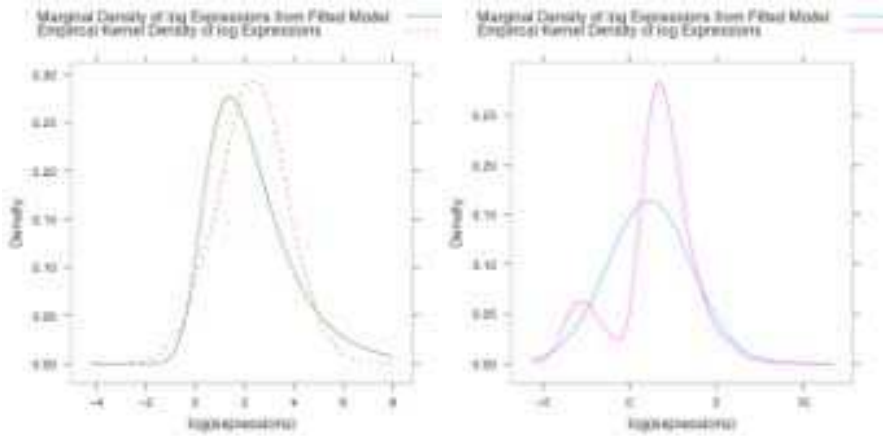
Gli istogrammi delle simulazioni in Figura 3.13 e 3.14 (A3.2.3.5) fanno ipotizzare che un possibile modello adatto ai dati sia di tipo bimodale; i grafici per la verifica dell'ipotesi di costanza del coefficiente di variazione portano a rifiutare tale ipotesi.

Per verificare l'adeguatezza dei modelli per i dati simulati sono stati costruiti i grafici in Figura 3.20 e 3.21, in cui la densità marginale secondo il modello imposto è confrontata con la densità stimata mediante il metodo del nucleo.



(a) (b)

Figura 3.15. (a) Dati simulati dal modello GG con CV non costante (*h=1/100, w=1/10*) e modellazione con GG. (b) Dati simulati dal modello GG con CV non costante (*h=1/100, w=1/10*) e modellazione con LNN.



(a) (b)

Figura 3.16. (a) Dati simulati dal modello LNN con CV non costante (*h=1/100, w=1/10*) e modellazione con GG. (b) Dati simulati dal modello LNN con CV non costante (*h=1/100, w=1/10*) e modellazione con LNN.

Nei grafici presentati in precedenza si evidenzia l'incapacità dei modelli di descrivere i dati, che si presentano secondo una distribuzione bimodale.
Sono state create le Tabelle 3.10 (a) e (b) per valutare le simulazioni fatte con GG sotto le due modellazioni, e le Tabelle 3.11 (a) e (b) per valutare le simulazioni fatte con LNN sotto le due modellazioni; con i parametri *h* e *w* fissati. I calcoli per la compilazione delle tabelle sono riportati in appendice A3.2.3.6.
Nei grafici presentati in precedenza si evidenzia l'incapacità dei modelli di descrivere i dati, che si presentano secondo una distribuzione bimodale.
Sono state create le Tabelle 3.10 (a) e (b) per valutare le simulazioni fatte con GG sotto le due modellazioni, e le
Tabelle 3.11 (a) e (b) per valutare le simulazioni fatte con LNN sotto le due modellazioni; con i parametri *h* e *w* fissati

	Espressione prevista dal modello GG	
Espressione esatta	Equivalente	Differente
Equivalente	7964	17
Differente	570	1449

(a)

	Espressione prevista dal modello LNN	
Espressione esatta	Equivalente	Differente
Equivalente	7964	17
Differente	570	1449

(b)

Tabella 3.10. (a) Corretta e non corretta identificazione dell'espressione genica con il modello GG su dati simulati da GG. (b) Corretta e non corretta identificazione dell'espressione genica con il modello LNN su dati simulati da GG.

	Espressione prevista dal modello LNN	
Espressione esatta	Equivalente	Differente
Equivalente	7843	20
Differente	1884	151

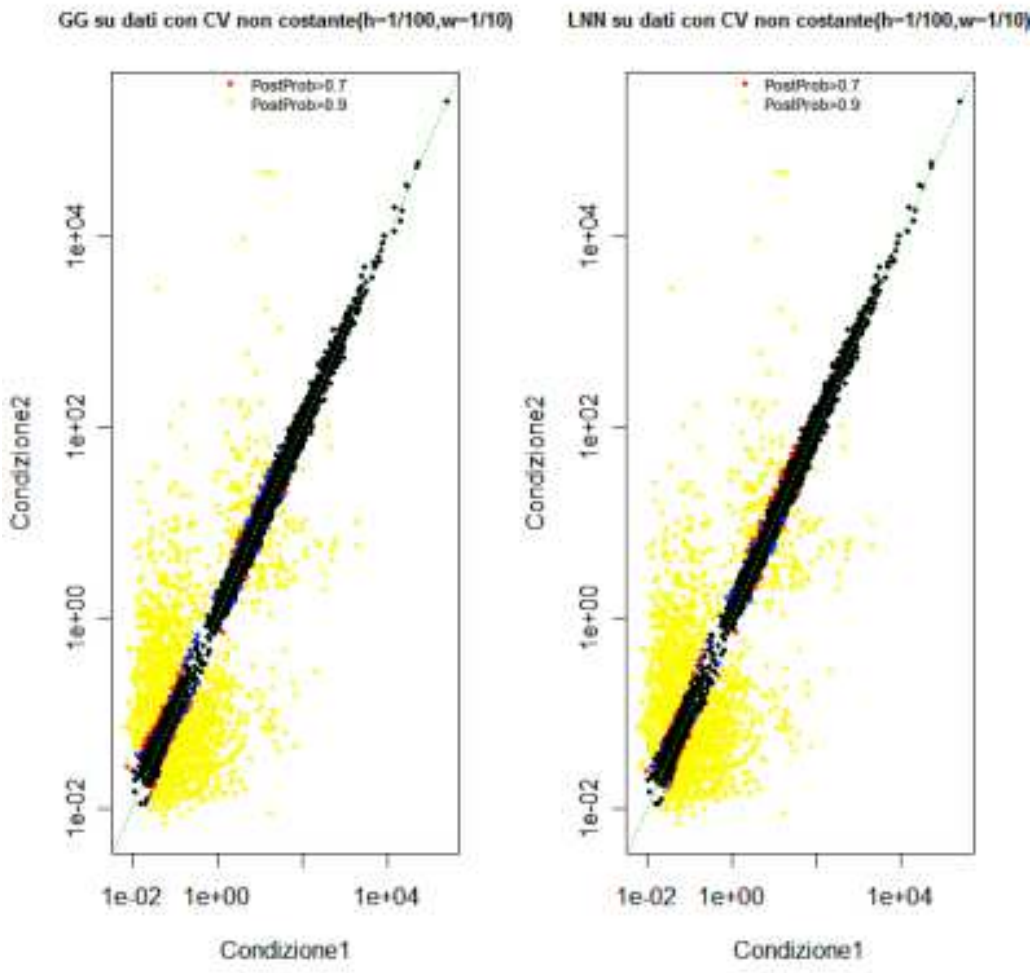
(a)

	Espressione prevista dal modello GG	
Espressione esatta	Equivalente	Differente
Equivalente	7953	7
Differente	1892	143

(b)

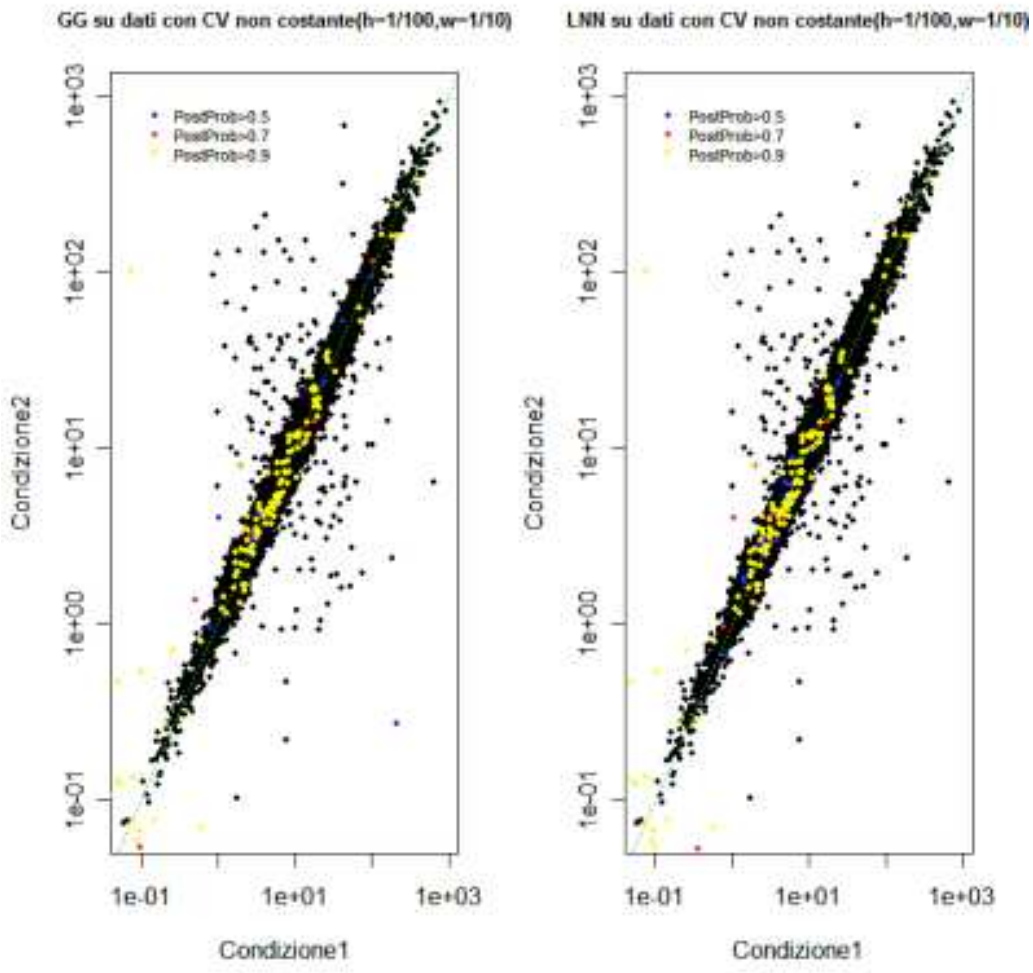
Tabella 3.11. (a) Corretta e non corretta identificazione dell'espressione genica con il modello GG su dati simulati da LNN. (b) Corretta e non corretta identificazione dell'espressione genica con il modello LNN su dati simulati da LNN.

Dalle tabelle sopra descritte si può notare che si ha una corretta identificazione di geni differenzialmente espressi pari al 71% per il modello GG e del 69% per il modello LNN nel caso di simulazioni da GG con coefficiente di variazione non costante; la stessa percentuale è pari al 33% per il modello GG e del 79% per il modello LNN. Tali percentuali evidenziano che nel caso di simulazioni da dati GG la non corretta specificazione del modello non influenza la corretta identificazione dei geni differenzialmente espressi, nel caso invece di simulazioni da LNN la situazione peggiora di molto: nel caso di non corretta specificazione del modello la percentuale decresce al 33%.
Si è voluto procedere con la costruzione dei grafici in Figura 3.17 e 3.18 che mostrano con diversa colorazione i geni che sono indicati come differenzialmente espressi dai modello con diverso valore della probabilità a posteriori (A3.2.3.7).



(a) (b)

Figura 3.17. (a) Geni indicati dal modello GG con CV non costante come differenzialmente espressi con diversa probabilità a priori per dati simulati da GG. (b) Geni indicati dal modello LNN con CV non costante come differenzialmente espressi con diversa probabilità a priori per dati simulati da GG.



(a) (b)

Figura 3.18 (a) Geni indicati dal modello GG con CV non costante come differenzialmente espressi con diversa probabilità a priori per dati simulati da LNN. (b) Geni indicati dal modello LNN con CV non costante come differenzialmente espressi con diversa probabilità a priori per dati simulati da LNN.

Anche questa volta non si notano sostanziali differenze da parte dei due modelli nell'indicare i geni in cui punti si discostano dalla bisettrice. Essi corrispondono a geni con espressione media differente nelle 2 condizioni.

Per un'ulteriore valutazione delle *performance* dei modelli sotto ipotesi di coefficiente di variazione non costante, si sono calcolati gli indici di bontà dei due modelli, riportati nelle tabelle 3.12, 3.13, 3.14 e 3.15. Sono stati, quindi, costruiti i grafici in Figura 3.19 e 3.20 che mettono a confronto tali indici.

p	0.1	0.2	0.3	0.4	0.5
Sens	0.667(0.042)	0.709(0.024)	0.718(0.028)	0.727(0.030)	0.750(0.021)
Spec	0.999(0.008)	0.998(0.002)	0.998(0.002)	0.996(0.003)	0.995(0.005)
PPV	0.992(0.009)	0.989(0.012)	0.993(0.006)	0.991(0.005)	0.993(0.006)
NPV	0.965(0.003)	0.931(0.005)	0.887(0.011)	0.843(0.019)	0.800(0.008)
FDR	0.008(0.009)	0.010(0.011)	0.007(0.006)	0.009(0.005)	0.007(0.006)

Tabella 3.12 Indicatori di bontà del modello GG su dati simulati dal modello GG con CV non costante (*h=1/100*, *w=1/10*); le stime sono ottenute dalle medie delle 10 ripetizioni, tra parentesi è indicato lo *standard error*.

p	0.1	0.2	0.3	0.4	0.5
Sens	0.675(0.033)	0.712(0.022)	0.721(0.026)	0.732(0.030)	0.758(0.020)
Spec	0.998(0.001)	0.995(0.002)	0.996(0.002)	0.992(0.004)	0.988(0.004)
PPV	0.978(0.018)	0.976(0.012)	0.987(0.006)	0.984(0.007)	0.984(0.005)
NPV	0.965(0.004)	0.932(0.004)	0.888(0.009)	0.845(0.019)	0.804(0.008)
FDR	0.022(0.018)	0.024(0.012)	0.013(0.006)	0.016(0.007)	0.016(0.005)

Tabella 3.13. Indicatori di bontà del modello LNN su dati simulati dal modello GG con CV non costante (*h=1/100*, *w=1/10*); le stime sono ottenute dalle medie delle 10 ripetizioni, tra parentesi è indicato lo *standard error*.

p	0.1	0.2	0.3	0.4	0.5
Sens	0.078(0.005)	0.071(0.007)	0.075(0.006)	0.074(0.007)	0.074(0.006)
Spec	1.000(0.001)	1.000(0.001)	0.999(0.001)	0.999(0.001)	0.999(0.001)
PPV	0.975(0.010)	0.974(0.018)	0.964(0.008)	0.972(0.013)	0.971(0.008)
NPV	0.813(0.002)	0.812(0.004)	0.814(0.004)	0.813(0.003)	0.812(0.004)
FDR	0.025(0.009)	0.026(0.018)	0.036(0.008)	0.028(0.013)	0.029(0.008)

Tabella 3.14. Indicatori di bontà del modello GG su dati simulati dal modello LNN con CV non costante (*h=1000*, *w=9/10*); le stime sono ottenute dalle medie delle 10 ripetizioni, tra parentesi è indicato lo *standard error*.

p	0.1	0.2	0.3	0.4	0.5
Sens	0.078(0.005)	0.071(0.006)	0.075(0.006)	0.074(0.006)	0.075(0.006)
Spec	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)	1.000(0.001)
PPV	0.981(0.009)	0.979(0.013)	0.977(0.008)	0.980(0.009)	0.982(0.007)
NPV	0.813(0.002)	0.812(0.004)	0.814(0.004)	0.813(0.003)	0.812(0.004)
FDR	0.019(0.001)	0.021(0.013)	0.023(0.008)	0.020(0.009)	0.018(0.007)

Tabella 3.15. Indicatori di bontà del modello LNN su dati simulati dal modello LNN con CV non costante (*h=1/100*, *w=1/10*); le stime sono ottenute dalle medie delle 10 ripetizioni, tra parentesi è indicato lo *standard error*.

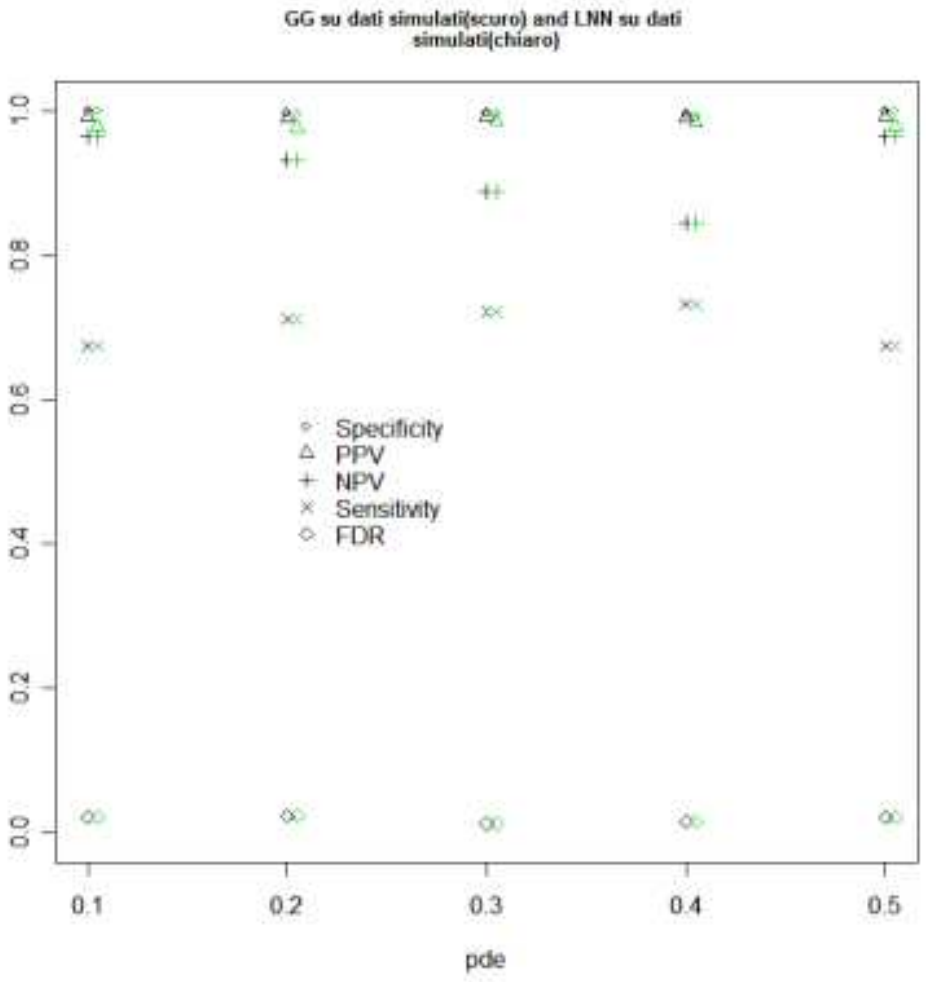


Figura 3.19. Indicatori di bontà del modello GG e LNN su dati simulati da GG con CV non costante (*h=1/100, w=1/10*), al variare della probabilità a priori di espressione genica differenziale. I caratteri più scuri mostrano gli indicatori del modello GG mentre quelli più chiari quelli del modello LNN.

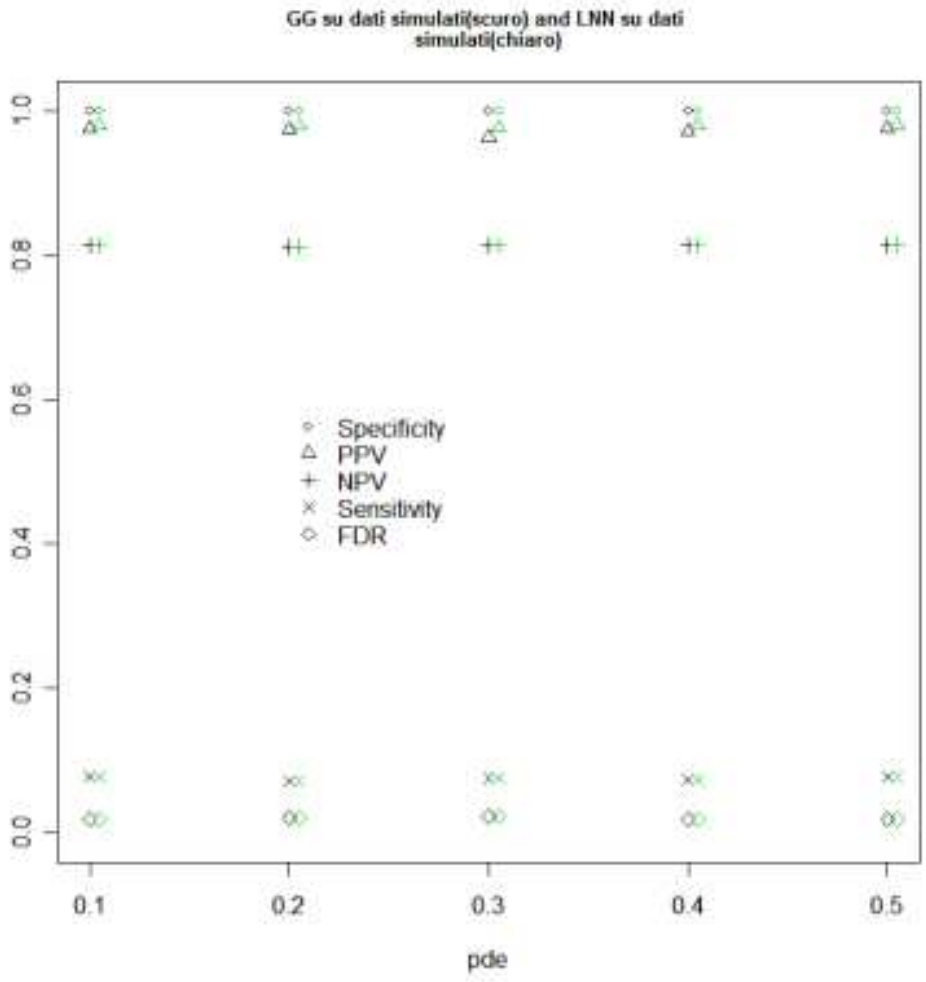


Figura 3.20. Indicatori di bontà del modello GG e LNN su dati simulati da LNN con CV non costante (*h=1/100, w=1/10*), al variare della probabilità a priori di espressione genica differenziale. I caratteri più scuri mostrano gli indicatori del modello GG mentre quelli più chiari quelli del modello LNN.

L'analisi degli indici attraverso i grafici fa a notare che, nonostante il rilevante peggioramento dell'indice *sentitivity*, in particolar modo per quanto riguarda il modello LNN, gli altri indicatori non presentano notevoli differenze. Si può quindi concludere che i due modelli, sotto l'ipotesi di coefficiente di variazione non costante, riescono a discriminare adeguatamente i geni che si discostano dalla

bisettrice. Per la costruzione delle tabelle sono state utilizzate le *routine* in appendice A3.2.3.8, per i grafici di confronto delle *performance* le *routine* in appendice A3.2.3.9.

Per concludere, sono state effettuate 10 simulazioni con 10000 geni per ogni simulazione in 2 condizioni con 3 repliche per la prima condizione e 15 repliche per la seconda, utilizzando il modello GG e il modello LNN con CV non costante, *h=1/100* e *w=1/10*. I parametri relativi sono stati stimati con l'ausilio della funzione *emfit* e nelle tabelle 3.16 e 3.17 sono riportate le stime con l'errore standard relativo alle stime nelle 10 simulazioni. Si può vedere come le stime siano vicine al vero valore del parametro con uno standard error relativamente basso.

p	0.1	0.2	0.3	0.4	0.5
$\hat{\alpha}$	9.927(0.028)	9.726(0.026)	9.505(0.046)	9.320(0.035)	9.156(0.042)
$\hat{\alpha}_0$	0.265(0.003)	0.302(0.001)	0.347(0.001)	0.397(0.003)	0.459(0.005)
$\hat{\upsilon}$	0.010(0.001)	0.010(0.001)	0.010(0.001)	0.010(0.001)	0.010(0.001)
$\hat{p}$	0.078(0.002)	0.158(0.004)	0.240(0.004)	0.319(0.003)	0.405(0.005)

Tabella 3.16. Stime dei parametri del modello GG sui dati simulati dal modello GG con (α, α_0, υ) = (10, 0.9, 0.5) con CV non costante (*h=1/100* e *w=1/*10), al variare di *p*. Tra parentesi è riportato lo *standard error*.

p	0.1	0.2	0.3	0.4	0.5
$\hat{\mu}_0$	2.302(0.010)	2.299(0.013)	2.303(0.009)	2.296(0.009)	2.297(0.011)
$\hat{\sigma}$	0.300(0.001)	0.300(0.001)	0.300(0.001)	0.300(0.001)	0.300(0.001)
$\hat{\tau}$	1.329(0.013)	1.259(0.012)	1.182(0.016)	1.127(0.015)	1.041(0.012)
$\hat{p}$	0.010(0.001)	0.021(0.002)	0.033(0.001)	0.044(0.002)	0.058(0.002)

Tabella 3.17. Stime dei parametri del modello LNN sui dati simulati dal modello LNN con (μ_0, σ, τ) = (2.3, 0.3, 1.39) con CV non costante (*h=1/100* e *w=1/*10), al variare di *p*. Tra parentesi è riportato lo *standard error*.

Si sono riportati in appendice i risultati delle stesse simulazioni riparametrizzando i due modelli, utilizzando valori proposti nella letteratura da Chiogna *et. al.* [rif. 10]: per il modello GG i nuovi parametri utilizzati sono (α, α_0, υ) = (1, 1.1, 45.4), per il modello LNN (μ_0, σ, τ) = (6.58, 0.9, 1.13). In termini di *performance* il risultato migliora confermando le considerazioni proposte nel capitolo.

CAPITOLO 3

Appendice A3

A3.1 Risultati relativi alle simulazioni dei due modelli riparametrizzati GG con (α, α_0, υ) = (1, 1.1, 45.4) e LNN con (μ_0, σ, τ) = (6.58, 0.9, 1.13).

3.1.1 Identificazione del modello

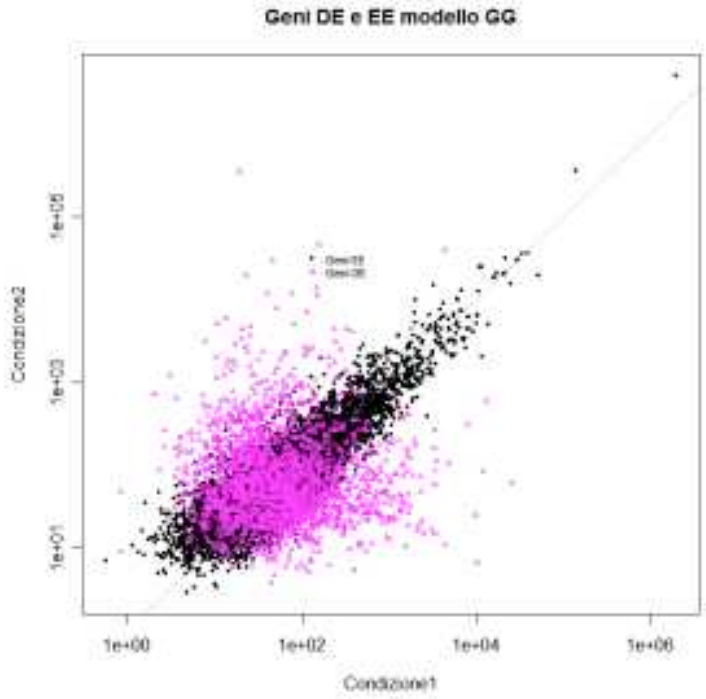


Figura 3.20 Corretta specificazione dell'espressione genica nel modello GG simulato.

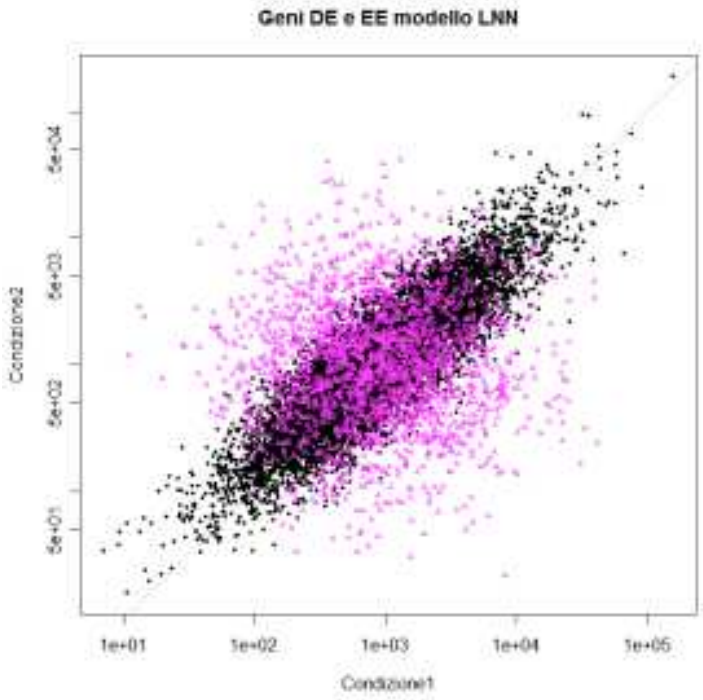
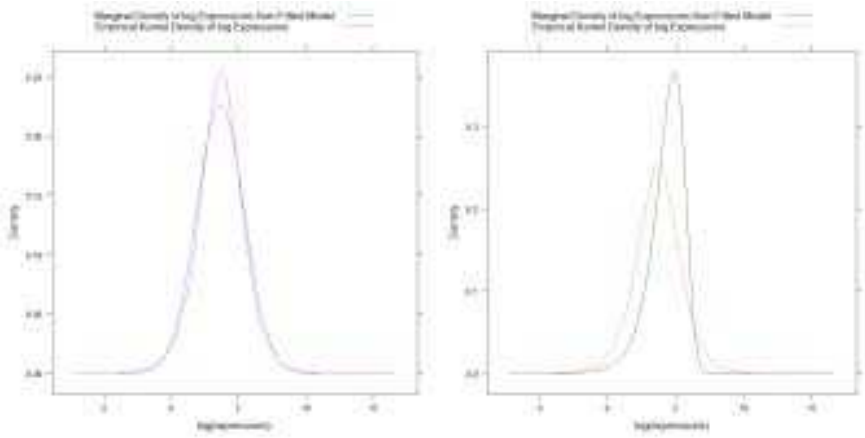
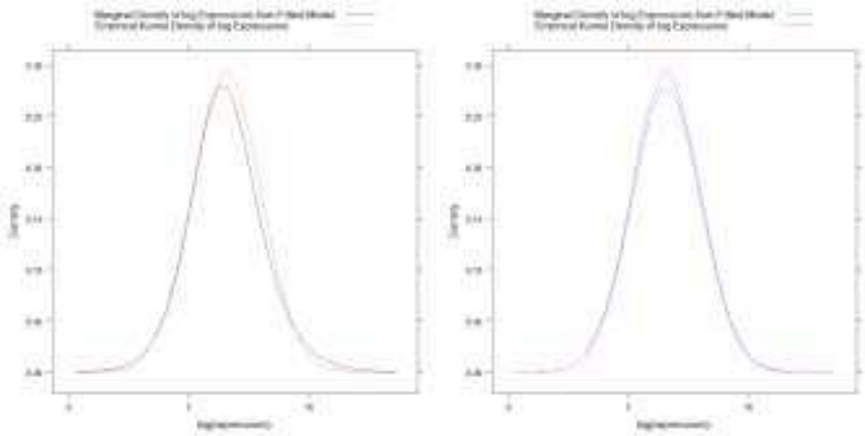


Figura 3.21 Corretta specificazione dell'espressione genica nel modello LNN simulato.



(a) (b)

Figura 3.22. (a) Dati simulati da modello GG e modellazione con GG. (b) Dati simulati con GG e modellazione con LNN.



(a) (b)

Figura 3.23. (a) Dati simulati da modello LNN e modellazione con GG. (b) Dati simulati con LNN e modellazione con LNN.

	Espressione prevista dal modello GG	
Espressione esatta	Equivalente	Differente
Equivalente	7977	0
Differente	1981	42

(b)

	Espressione prevista dal modello LNN	
Espressione esatta	Equivalente	Differente
Equivalente	7794	183
Differente	1381	642

(b)

Tabella 3.18. (a) Corretta e non corretta identificazione dell'espressione genica con il modello GG su dati simulati da GG. (b) Corretta e non corretta identificazione dell'espressione genica con il modello LNN su dati simulati da GG.

	Espressione prevista dal modello GG	
Espressione esatta	Equivalente	Differente
Equivalente	7597	419
Differente	1020	964

(a)

	Espressione prevista dal modello LNN	
Espressione esatta	Equivalente	Differente
Equivalente	7874	142
Differente	1103	881

(b)

Tabella 3.19. (a) Corretta e non corretta identificazione dell'espressione genica con il modello GG su dati simulati da LNN. (b) Corretta e non corretta identificazione dell'espressione genica con il modello LNN su dati simulati da LNN.

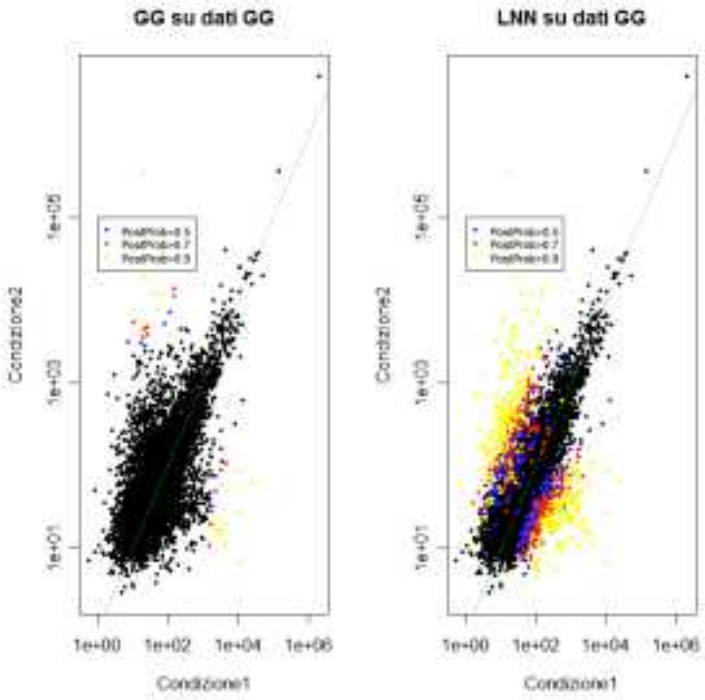


Figura 3.24 Geni identificati come differenzialmente espressi nei due modelli con diversa probabilità a posteriori (dati simulati da GG).

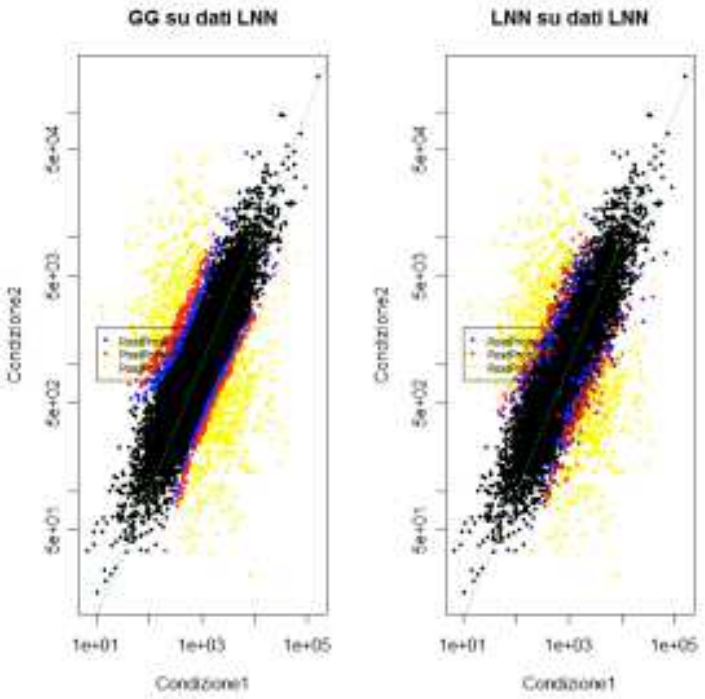


Figura 3.25 Geni identificati come differenzialmente espressi nei due modelli con diversa probabilità a posteriori (dati simulati da LNN).

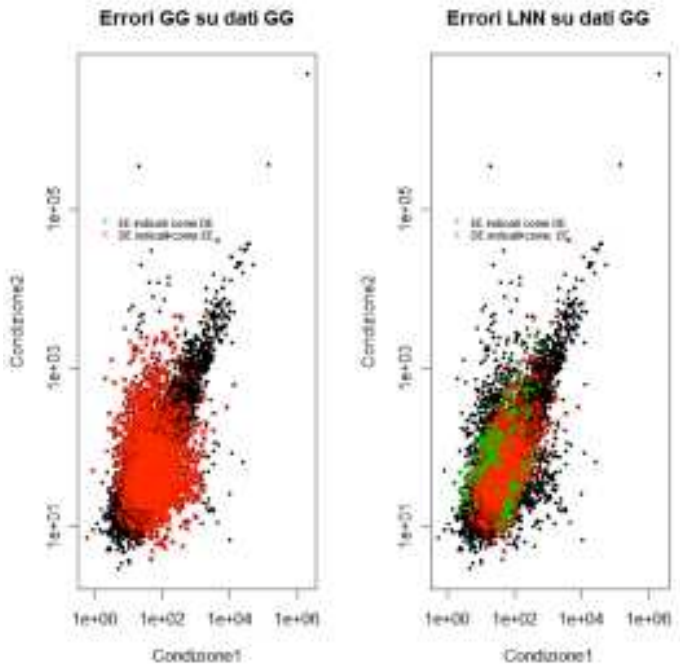


Figura 3.26. Geni non correttamente identificati dai due modelli (dati simulati da GG).

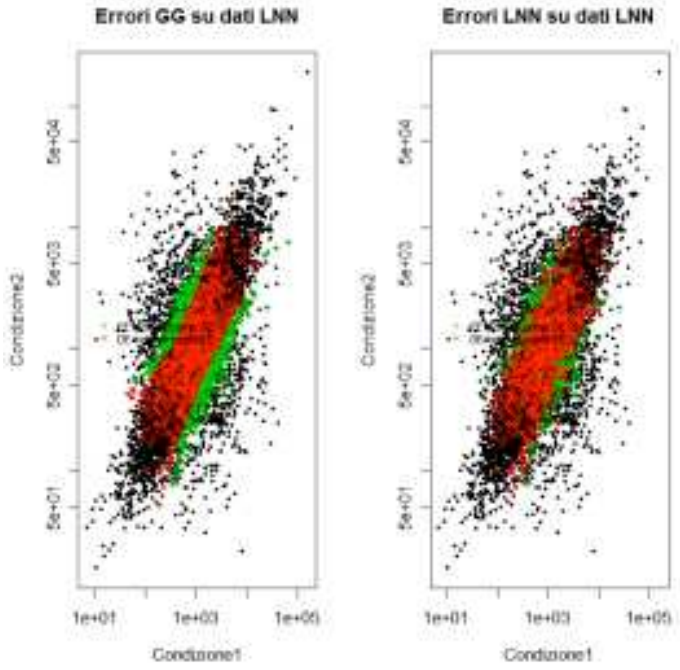


Figura 3.27. (a) Geni non correttamente identificati dai due modelli (dati simulati da LNN).

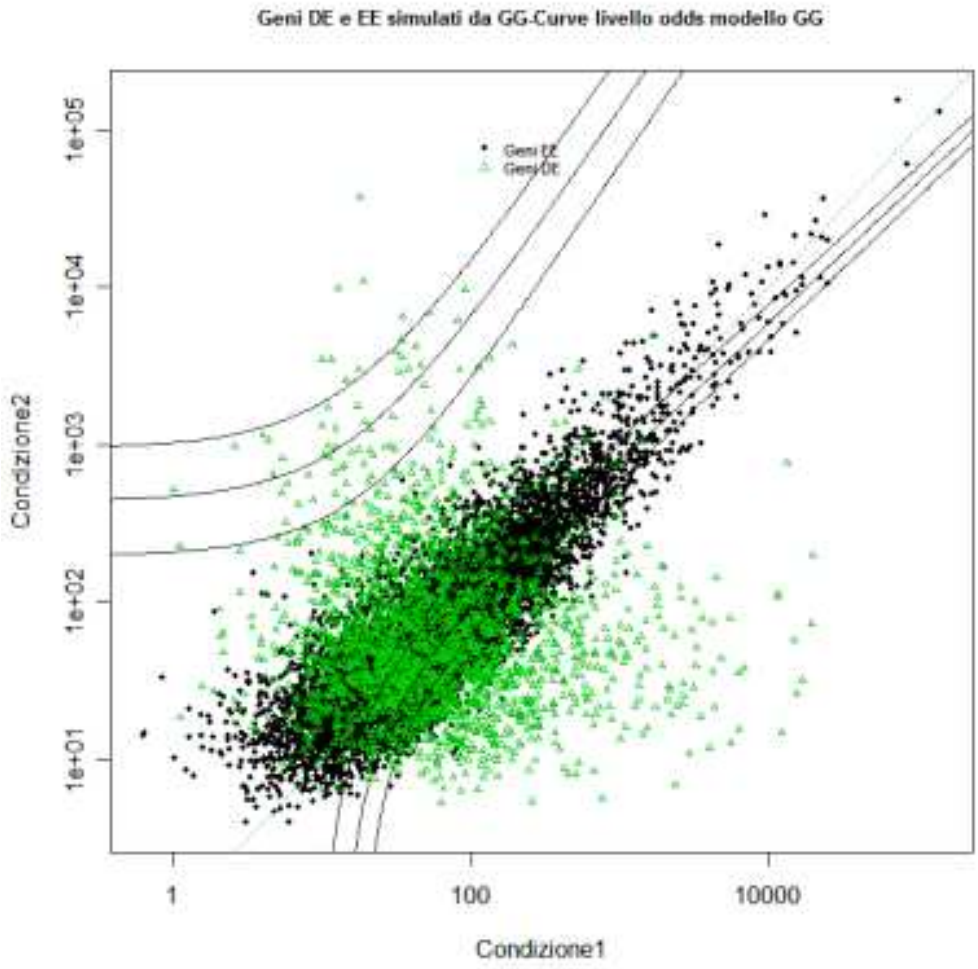


Figura 3.28. Curve di livello degli *odds* calcolate mediante il modello GG. Corrispondono agli *odds* di 1, 10 e 100 rispettivamente dalla curva più interna alla curva più esterna. I punti compresi tra le due curve più interne rappresentano i geni identificati come equivalentemente espressi dal modello GG. I punti indicati con i triangolini rappresentano i geni simulati come differenzialmente espressi.

A3.1.2 Indicatori di bontà e stima dei parametri

p	0.1	0.2	0.3	0.4	0.5
Sens	0.282(0.061)	0.397(0.035)	0.449(0.047)	0.516(0.035)	0.600(0.040)
Spec	0.992(0.004)	0.979(0.006)	0.961(0.013)	0.912(0.018)	0.852(0.031)
PPV	0.801(0.061)	0.829(0.034)	0.831(0.036)	0.804(0.025)	0.804(0.023)
NPV	0.926(0.011)	0.864(0.011)	0.805(0.013)	0.733(0.013)	0.680(0.020)
FDR	0.199(0.061)	0.171(0.034)	0.169(0.036)	0.196(0.025)	0.196(0.023)

Tabella 3.20. Indici di bontà del modello GG su dati simulati dal modello GG; le stime sono ottenute dalla media delle 10 ripetizioni, tra parentesi indicato lo *standard error*.

p	0.1	0.2	0.3	0.4	0.5
Sens	0.341(0.094)	0.482(0.081)	0.532(0.028)	0.599(0.039)	0.617(0.823)
Spec	0.983(0.011)	0.952(0.018)	0.920(0.015)	0.865(0.020)	0.826(0.046)
PPV	0.732(0.138)	0.724(0.076)	0.742(0.036)	0.751(0.023)	0.783(0.039)
NPV	0.930(0.009)	0.884(0.015)	0.820(0.010)	0.760(0.013)	0.686(0.042)
FDR	0.268(0.138)	0.276(0.076)	0.258(0.036)	0.249(0.023)	0.217(0.039)

Tabella 3.21. Indici di bontà del modello GG su dati simulati dal modello LNN; le stime sono ottenute dalla media delle 10 ripetizioni, tra parentesi indicato lo *standard error*.

p	0.1	0.2	0.3	0.4	0.5
Sens	0.234(0.051)	0.318(0.031)	0.392(0.049)	0.471(0.046)	0.547(0.041)
Spec	0.987(0.004)	0.975(0.010)	0.954(0.015)	0.906(0.022)	0.855(0.038)
PPV	0.672(0.085)	0.771(0.051)	0.789(0.041)	0.777(0.030)	0.794(0.034)
NPV	0.921(0.010)	0.848(0.009)	0.788(0.011)	0.714(0.013)	0.653(0.017)
FDR	0.328(0.085)	0.229(0.051)	0.211(0.041)	0.223(0.030)	0.206(0.034)

Tabella 3.22. Indici di bontà del modello LNN su dati simulati dal modello GG; le stime sono ottenute dalla media delle 10 ripetizioni, tra parentesi indicato lo *standard error*.

p	0.1	0.2	0.3	0.4	0.5
Sens	0.370(0.049)	0.459(0.043)	0.505(0.031)	0.560(0.045)	0.623(0.040)
Spec	0.994(0.002)	0.982(0.005)	0.960(0.010)	0.925(0.015)	0.875(0.021)
PPV	0.879(0.054)	0.863(0.036)	0.848(0.028)	0.836(0.019)	0.833(0.015)
NPV	0.934(0.005)	0.882(0.009)	0.818(0.010)	0.756(0.016)	0.700(0.021)
FDR	0.121(0.054)	0.137(0.036)	0.152(0.029)	0.164(0.019)	0.167(0.015)

Tabella 3.23. Indici di bontà del modello LNN su dati simulati dal modello LNN; le stime sono ottenute dalla media delle 10 ripetizioni, tra parentesi indicato lo *standard error*.

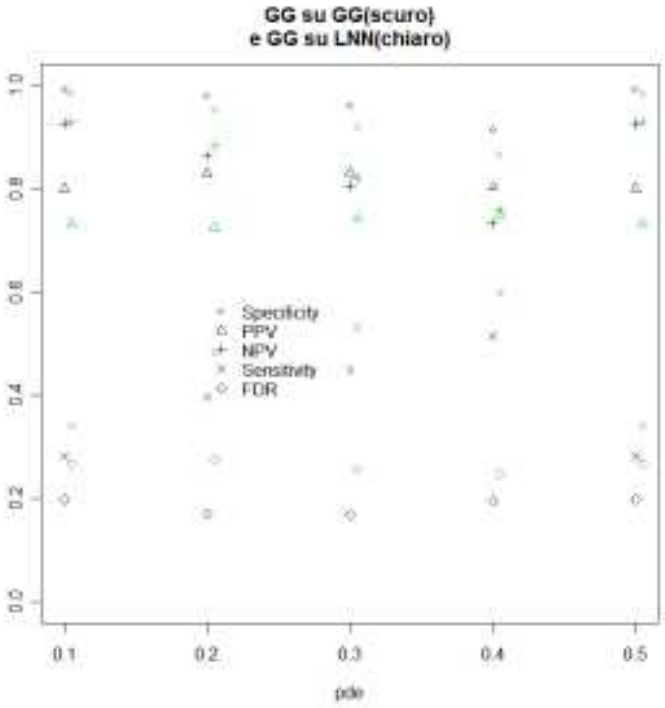


Figura 3.29. Indicatori di bontà del modello GG su dati simulati al variare della probabilità a priori di espressione genica differenziale. I caratteri più scuri indicano gli indicatori del modello GG su dati simulati da GG, mentre caratteri chiari si riferiscono al modello GG su dati simulati da LNN.

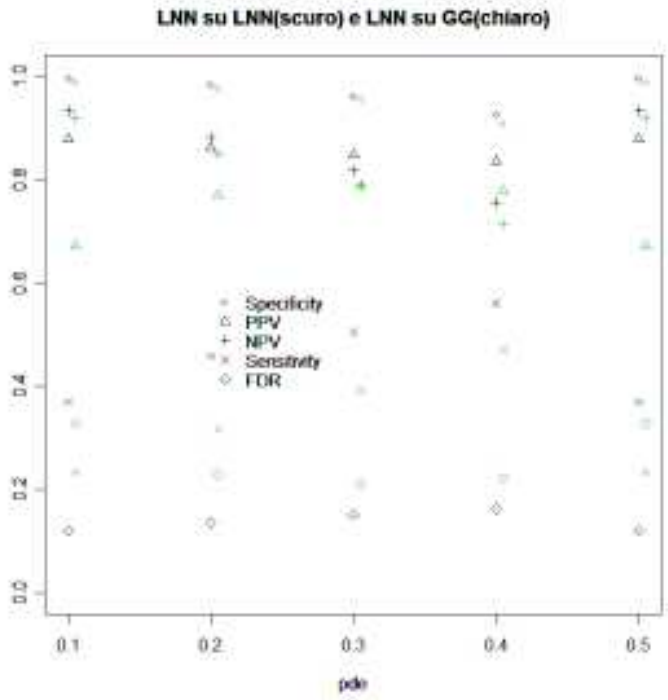


Figura 3.30. Indicatori di bontà del modello LNN su dati simulati al variare della probabilità a priori di espressione genica differenziale. I caratteri più scuri indicano gli indicatori del modello LNN su dati simulati da LNN, mentre caratteri chiari si riferiscono al modello LNN su dati simulati da GG.

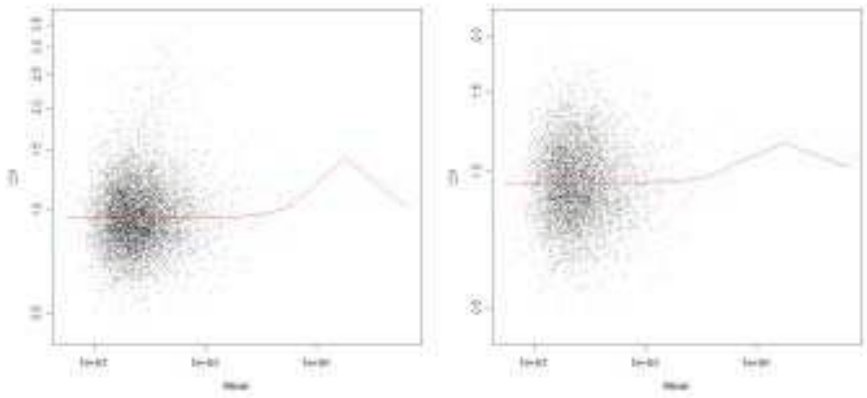
p	0.1	0.2	0.3	0.4	0.5
$\hat{\alpha}$	1.010(0.001)	1.010(0.001)	1.010(0.0019	1.010(0.001)	1.010(0.001)
$\hat{\alpha}_0$	1.197(0.015)	1.096(0.021)	0.097(0.021)	1.085(0.006)	1.090(0.014)
$\hat{\nu}$	44.715(0.918)	44.724(0.839)	44.605(0.927)	44.287(0.441)	44.532(0.717)
$\hat{p}$	0.096(0.006)	0.187(0.010)	0.292(0.005)	0.385(0.016)	0.583(0.010)

Tabella 3.24. Stime dei parametri del modello GG sui dati simulati dal modello GG con (α, α_0, υ) = (1, 1.1, 45.4) al variare di *p*. Tra parentesi è riportato lo *standard error*.

p	0.1	0.2	0.3	0.4	0.5
$\hat{\mu}_0$	6.587(0.014)	6.577(0.011)	6.579(0.010)	6.578(0.008)	6.584(0.017)
$\hat{\sigma}$	0.900(0.002)	0.900(0.010)	0.900(0.002)	0.900(0.001)	0.900(0.001)
$\hat{\tau}$	1.129(0.010)	1.126(0.006)	1.129(0.006)	1.131(0.004)	1.131(0.009)
$\hat{p}$	0.106(0.006)	0.206(0.007)	0.302(0.006)	0.405(0.008)	0.499(0.010)

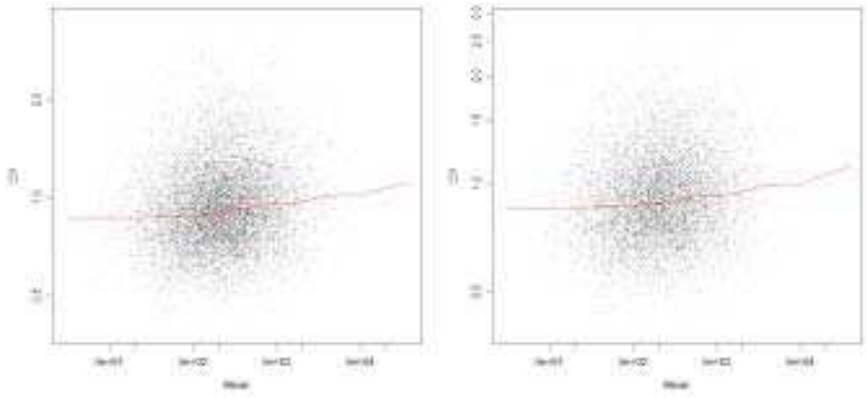
Tabella 3.25. Stime dei parametri del modello GG sui dati simulati dal modello GG con (μ_0, σ, τ) = (6.58, 0.9, 1.13) al variare di *p*. Tra parentesi è riportato lo *standard error*.

A3.1.3 Ipotesi coefficiente di variazione costante



(a) (b)

Figura 3.31. (a) Coefficiente di variazione della media per tutti i geni simulati da GG. (b) Coefficiente di variazione della media per i geni equivalentemente espressi simulati da GG.



(a) (b)

Figura 3.32. (a) Coefficiente di variazione della media per tutti i geni simulati da LNN. (b) Coefficiente di variazione della media per i geni equivalentemente espressi simulati da LNN.

A3.1.4 Dati simulati da modelli GG e LNN con CV non costante (*h=1/100, w=1/10*).

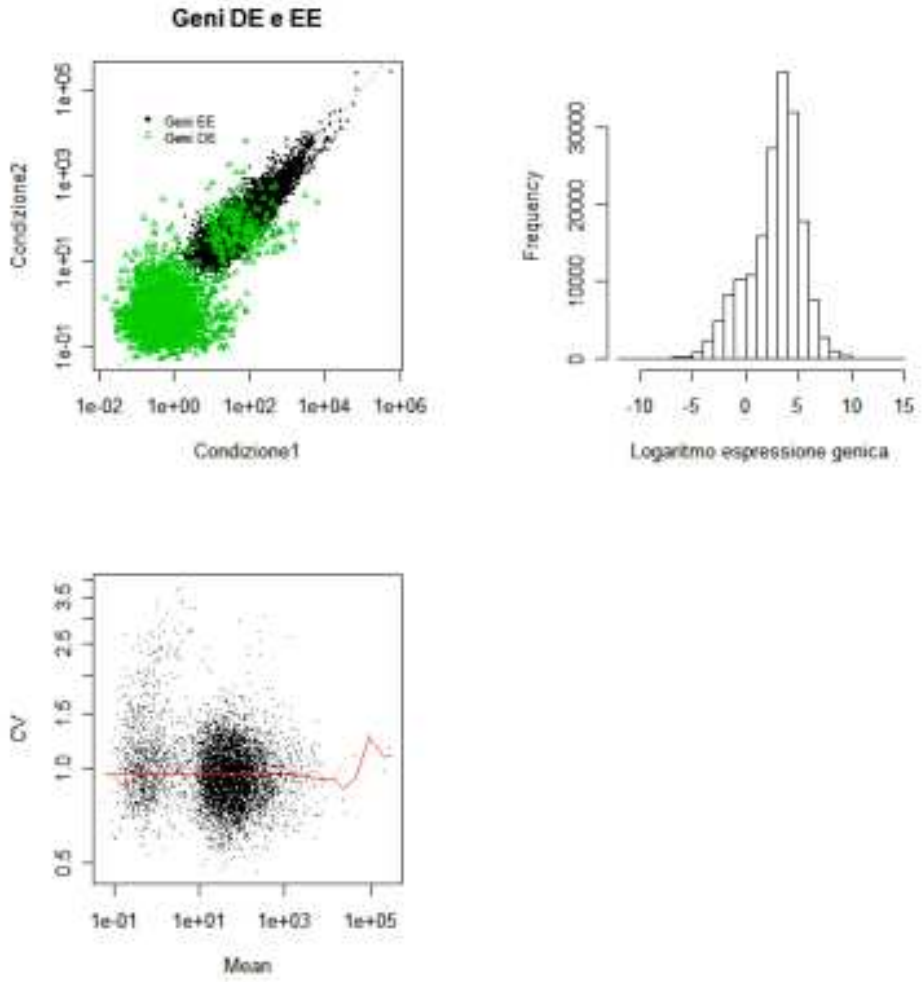


Figura 3.33. In alto a sinistra il grafico che rappresenta le simulazioni di geni dal modello Gamma-Gamma come equivalentemente e differenzialmente espressi con parametri *w* e *h* posti uguale rispettivamente a 1/10 e 1/100; in alto a destra l'istogramma del logaritmo dei dati di espressione genica simulati con GG. In basso il coefficiente di variazione dei dati in funzione della media.

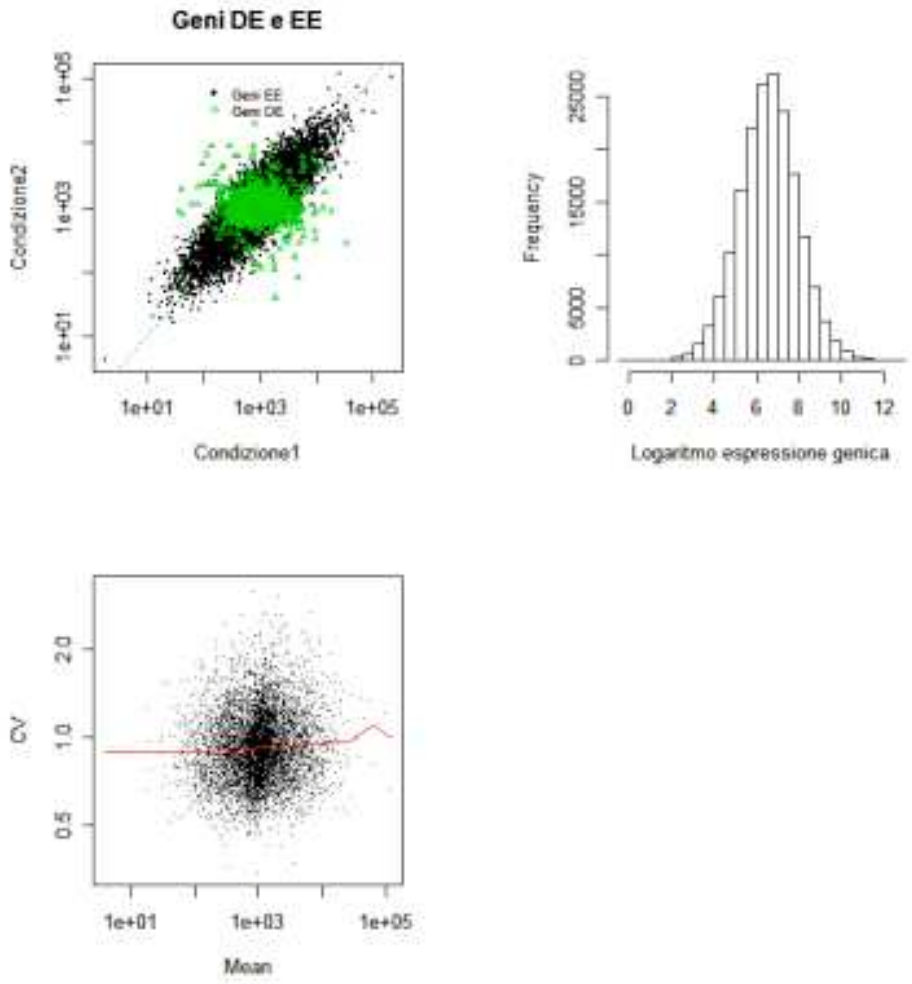
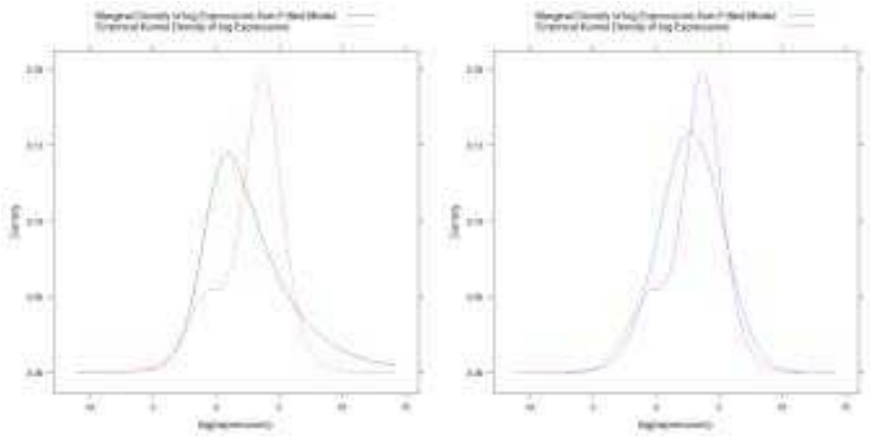
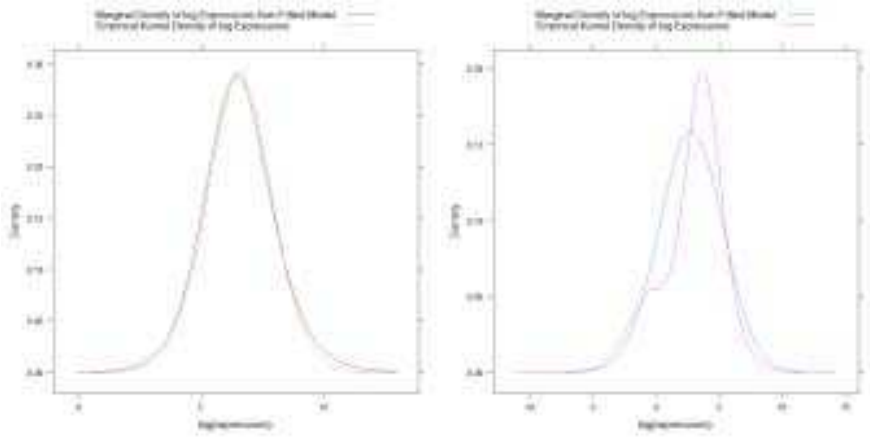


Figura 3.34. In alto a sinistra il grafico che rappresenta le simulazioni di geni dal modello LogNormale-Normale come equivalentemente e differenzialmente espressi con parametri *w* e *h* posti uguale rispettivamente a 1/10 e 1/100; in alto a destra l'istogramma del logaritmo dei dati di espressione genica simulati con LNN. In basso il coefficiente di variazione dei dati in funzione della media.



(a) (b)

Figura 3.35. (a) Dati simulati dal modello GG con CV non costante e parametri *h=1/100* e *w=1/10* modellazione con GG. (b) Dati simulati dal modello GG con CV non costante e parametri *h=1/100* e *w=1/10* modellazione con LNN.



(a) (b)

Figura 3.35. (a) Dati simulati dal modello LNN con CV non costante e parametri *h=1/100* e *w=1/10* modellazione con GG. (b) Dati simulati dal modello LNN con CV non costante e parametri *h=1/100* e *w=1/10* modellazione con LNN.

	Espressione prevista dal modello GG	
Espressione esatta	Equivalente	Differente
Equivalente	7967	35
Differente	1603	395

(a)

	Espressione prevista dal modello LNN	
Espressione esatta	Equivalente	Differente
Equivalente	7923	79
Differente	1697	301

(b)

Tabella 3.26. (a) Corretta e non corretta identificazione dell'espressione genica con il modello GG su dati simulati da GG. (b) Corretta e non corretta identificazione dell'espressione genica con il modello LNN su dati simulati da GG.

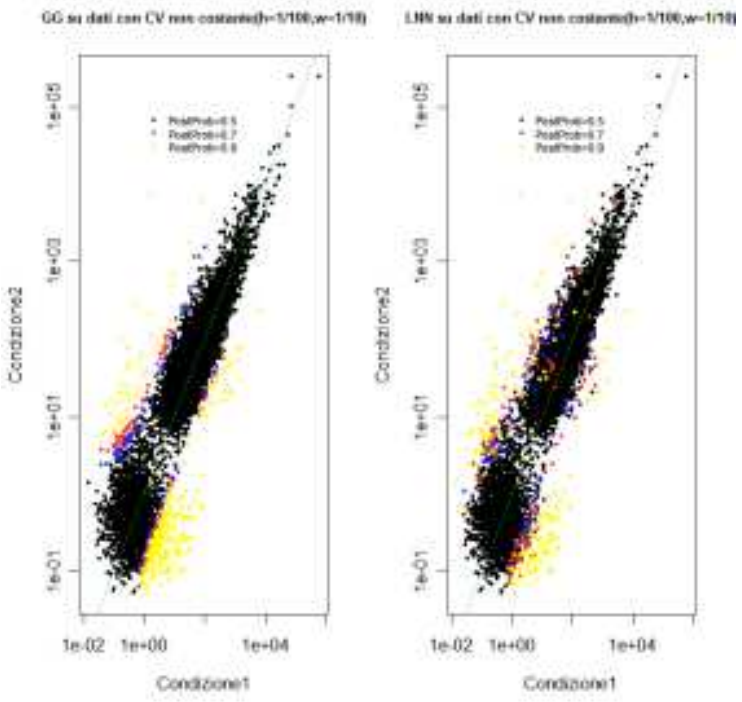
	Espressione prevista dal modello GG	
Espressione esatta	Equivalente	Differente
Equivalente	7871	129
Differente	1894	106

(a)

	Espressione prevista dal modello LNN	
Espressione esatta	Equivalente	Differente
Equivalente	7979	0
Differente	1973	27

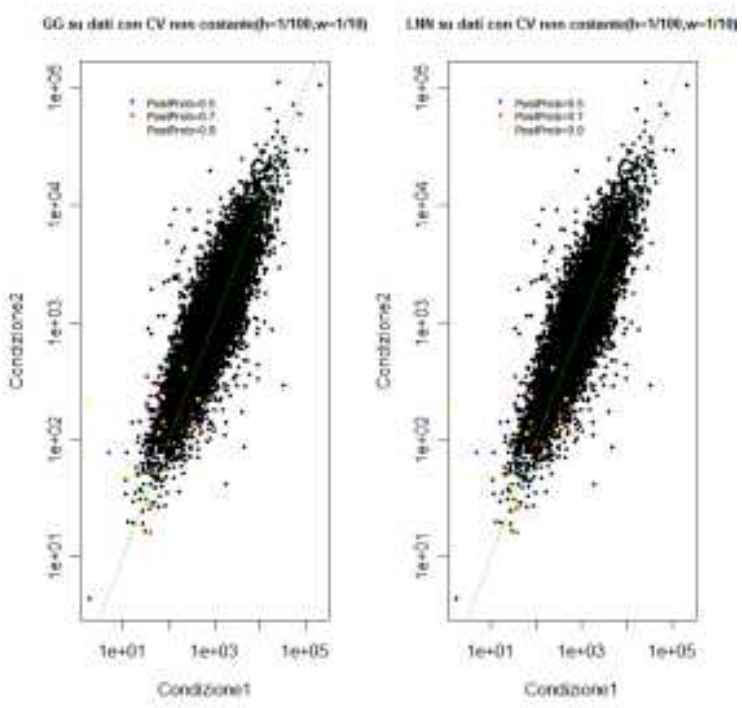
(b)

Tabella 3.27. (a) Corretta e non corretta identificazione dell'espressione genica con il modello GG su dati simulati da LNN. (b) Corretta e non corretta identificazione dell'espressione genica con il modello LNN su dati simulati da LNN.



(a) (b)

Figura 3.36. (a) Geni indicati dal modello GG con CV non costante come differenzialmente espressi con diversa probabilità a priori per dati simulati da GG. (b) Geni indicati dal modello LNN con CV non costante come differenzialmente espressi con diversa probabilità a priori per dati simulati da GG



(a) (b)

Figura 3.37. (a) Geni indicati dal modello GG con CV non costante come differenzialmente espressi con diversa probabilità a priori per dati simulati da LNN. (b) Geni indicati dal modello LNN con CV non costante come differenzialmente espressi con diversa probabilità a priori per dati simulati da LNN.

p	0.1	0.2	0.3	0.4	0.5
Sens	0.159(0.038)	0.218(0.032)	0.260(0.332)	0.306(0.037)	0.346(0.018)
Spec	0.998(0.001)	0.995(0.003)	0.991(0.005)	0.990(0.004)	0.986(0.006)
PPV	0.903(0.056)	0.921(0.040)	0.927(0.040)	0.951(0.016)	0.960(0.017)
NPV	0.914(0.007)	0.843(0.012)	0.757(0.007)	0.684(0.018)	0.604(0.016)
FDR	0.097(0.056)	0.079(0.040)	0.073(0.040)	0.048(0.016)	0.040(0.017)

Tabella 3.28. Indicatori di bontà del modello GG su dati simulati dal modello GG con CV non costante (*h=1/100* e *w=1/10*); le stime sono ottenute dalle medie delle 10 ripetizioni, tra parentesi è indicato lo *standard error*.

p	0.1	0.2	0.3	0.4	0.5
Sens	0.152(0.034)	0.165(0.036)	0.200(0.026)	0.231(0.029)	0.255(0.018)
Spec	0.994(0.004)	0.988(0.003)	0.987(0.005)	0.986(0.004)	0.977(0.006)
PPV	0.783(0.124)	0.770(0.067)	0.864(0.052)	0.915(0.026)	0.916(0.020)
NPV	0.913(0.008)	0.833(0.014)	0.741(0.008)	0.661(0.017)	0.570(0.019)
FDR	0.217(0.124)	0.230(0.067)	0.136(0.052)	0.085(0.026)	0.084(0.020)

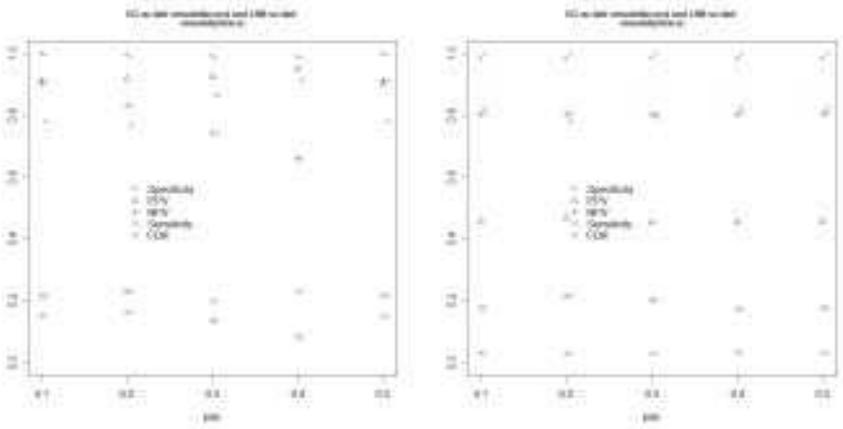
Tabella 3.29. Indicatori di bontà del modello LNN su dati simulati dal modello GG con CV non costante (*h=1/100* e *w=1/10*); le stime sono ottenute dalle medie delle 10 ripetizioni, tra parentesi è indicato lo *standard error*.

p	0.1	0.2	0.3	0.4	0.5
Sens	0.052(0.005)	0.052(0.008)	0.048(0.010)	0.054(0.006)	0.042(0.008)
Spec	0.984(0.003)	0.985(0.004)	0.985(0.005)	0.984(0.001)	0.988(0.004)
PPV	0.461(0.042)	0.468(0.054)	0.453(0.062)	0.456(0.030)	0.497(0.054)
NPV	0.805(0.002)	0.807(0.002)	0.804(0.004)	0.806(0.003)	0.805(0.002)
FDR	0.539(0.042)	0.532(0.054)	0.547(0.062)	0.544(0.030)	0.502(0.054)

Tabella 3.30. Indicatori di bontà del modello GG su dati simulati dal modello LNN con CV non costante (*h=1/100* e *w=1/10*); le stime sono ottenute dalle medie delle 10 ripetizioni, tra parentesi è indicato lo *standard error*.

p	0.1	0.2	0.3	0.4	0.5
Sens	0.033(0.005)	0.031(0.004)	0.031(0.005)	0.035(0.006)	0.030(0.003)
Spec	0.998(0.001)	0.998/9.001)	0.998(0.001)	0.998(0.001)	0.998(0.001)
PPV	0.821(0.045)	0.783(0.054)	0.797(0.048)	0.826(0.045)	0.812(0.052)
NPV	0.804(0.002)	0.805(0.002)	0.804(0.004)	0.805(0.002)	0.805(0.001)
FDR	0.178(0.045)	0.217(0.054)	0.203(0.048)	0.174(0.045)	0.188(0.053)

Tabella 3.31. Indicatori di bontà del modello LNN su dati simulati dal modello LNN con CV non costante (*h=1/100* e *w=1/10*); le stime sono ottenute dalle medie delle 10 ripetizioni, tra parentesi è indicato lo *standard error*.



(a) (b)

Figura 3.38. (a) Indicatori di bontà del modello GG e LNN su dati simulati da GG con CV non costante (*h=1/100* e *w=1/10*), al variare della probabilità a priori di espressione genica differenziale. I caratteri più scuri mostrano gli indicatori del modello GG mentre quelli più chiari quelli del modello LNN. (b) Indicatori di bontà del modello GG e LNN su dati simulati da LNN con CV non costante (*h=1/100* e *w=1/10*), al variare della probabilità a priori di espressione genica differenziale. I caratteri più scuri mostrano gli indicatori del modello GG mentre quelli più chiari quelli del modello LNN.

p	0.1	0.2	0.3	0.4	0.5
$\hat{\alpha}$	1.010(0.001)	1.010(0.001)	1.010(0.001)	1.010(0.001)	1.010(0.0019
$\hat{\alpha}_0$	0.317(0.005)	0.286(0.002)	0.284(0.001)	0.300(0.001)	0.322(0.003)
$\hat{\nu}$	1.785(0.099)	0.693(0.032)	0.415(0.009)	0.319(0.004)	0.268(0.004)
$\hat{p}$	0.031(0.003)	0.069(0.005)	0.121(0.004)	0.175(0.007)	0.236(0.004)

Tabella 3.32. Stime dei parametri del modello GG sui dati simulati dal modello GG con (α, α_0, υ) = (1, 1.1, 45.5) al variare di *p*. Tra parentesi è riportato lo *standard error*.

p	0.1	0.2	0.3	0.4	0.5
$\hat{\mu}_0$	6.581(0.012)	6.580(0.015)	6.584(0.005)	6.584(0.009)	6.578(0.007)
$\hat{\sigma}$	0.899(0.001)	0.899(0.001)	0.899(0.001)	0.899(0.001)	0.898(0.001)
$\hat{\tau}$	1.074(0.008)	1.023(0.014)	0.967(0.007)	0.900(0.012)	0.834(0.008)
$\hat{p}$	0.022(0.002)	0.038(0.004)	0.056(0.003)	0.082(0.004)	0.116(0.005)

Tabella 3.33. Stime dei parametri del modello GG sui dati simulati dal modello GG con (μ_0, σ, τ) = (6.58, 0.9, 1.13) al variare di *p*. Tra parentesi è riportato lo *standard error*.

A3.2 Codice *R* relativo alle simulazioni effettuate.

A3.2.1 Identificazione del modello.

A3.2.1 Funzione *sim2GG* per la simulazione di espressioni geniche dal modello Gamma-Gamma descritta nello Schema 3.1.

```
function(ngeni=10000,nrep1,nrep2,alpha,alpha0,nu,p=0.2)
{     ncond<-2
      matrice<-
matrix(rep(NA,ngeni*(nrep1+nrep2)),ncol=nrep1+nrep2,byrow=T)
      #vettore che indica se un gene µe differenzialmente espresso
      DE<-rep(FALSE,ngeni)
      for(i in 1:ngeni)
            { if(runif(1)>p)
            #Nel caso di espressione equivalente
                  { lambda<-rgamma(1,shape=alpha0,rate=nu)
                  matrice[i,]<-rgamma(nrep1+nrep2,shape=alpha,
                  rate=lambda)
                  }
            else
            #Nel caso di espressione differenziale
            { lambda1<-rgamma(1,shape=alpha0,rate=nu)
            lambda2<-rgamma(1,shape=alpha0,rate=nu)
            cond1<-rgamma(nrep1,shape=alpha,rate=lambda1)
            cond2<-rgamma(nrep2,shape=alpha,rate=lambda2)
            matrice[i,]<-c(cond1,cond2)
            DE[i]<-TRUE
                  }
            }
      list(matrice=matrice,DE=DE)
}
```

A3.2.1.2 Utilizzo della funzione *sim2GG* per la costruzione del grafico in Figura 3.1 e 3.20.

```
nrep1=3
nrep2=15
# Per il grafico in figura 3.1
alpha=10
alpha0=0.9
nu=0.5
simulazioneGG=sim2GG(p=0.2)
#medie per tutti i geni
n<-10000
medieTutti<-matrix(rep(0,10000*2),ncol=2)
for(i in 1:n)
      { medieTutti[i,1]<-mean(simulazioneGG$matrice[i,1:3])
        medieTutti[i,2]<-mean(simulazioneGG$matrice[i,4:7])
      }
plot(medieTutti[!simulazioneGG$DE,1],medieTutti[!simulazioneGG$DE,
2],pch=16,cex=.5,log="xy",
xlab="Condizione1",ylab="Condizione2",main="Geni DE e EE modello
GG")
points(medieTutti[simulazioneGG$DE,1],medieTutti[simulazioneGG$DE,
2],pch=2,cex=.5,col=6)
legend(100,45000,legend=c("Geni EE","Geni
DE"),col=c("black",6),cex=.7,pch=c(16,2),bty="n")
abline(0,1,lty=3,col=3)



# Per il grafico in figura 3.20 riparametrizzo e ripeto il codice
per la costruzione del grafico
alpha=1
alpha0=1.1
nu=45.4
```

A3.2.1.3 Utilizzo della funzione *sim2GG* per la stima dei parametri dei modelli GG e LNN.

```
library(EBarrays)
matrice<-simulazioneGG$matrice
pattern<-ebPatterns(c("1,1,1,1,1,1,1","1,1,1,2,2,2,2"))
gg.fit <- emfit(data = matrice, family = "GG",hypotheses =
pattern,num.iter=10)
gg.post.out <- postprob(gg.fit,matrice)
lnn.fit <- emfit(data = matrice, family = "LNN",hypotheses =
pattern,num.iter=10)
lnn.post.out <- postprob(lnn.fit,matrice)
```

A3.2.1.4 Costruzione dei grafici delle densità marginali dei geni simulati da *sim2GG* di Figura 3.2 (a) e (b) e 3.22 (a) e (b).

```
trellis.device(theme= col.whitebg())
print(plotMarginal(gg.fit,matrice))
print(plotMarginal(lnn.fit,matrice))
```

A3.2.1.5 Funzione *sim2LNN* per la simulazione di espressioni geniche dal modello LogNormale-Normale descritta nello Schema 3.2.

```
sim2LNN<-function (ngeni=10000,nrep1,nrep2,mu0,sigma,tau,p=0.2)
{ ncond=2
      matrice<-
matrix(rep(NA,ngeni*(nrep1+nrep2)),ncol=nrep1+nrep2,byrow=T)
      #vettore che indica se un gene µe differenzialmente espresso
      DE<-rep(FALSE,ngeni)
      for(i in 1:ngeni)
      {     if(runif(1)>p)
            #Equivalent Expression
            {     mu.g<-rnorm(1,mu0,tau)
                  matrice[i,]<-exp(rnorm(nrep1+nrep2,mu.g,sigma))
            }
            else
            #Different Expression
            {     mu.g1<-rnorm(1,mu0,tau)
```

```
                    mu.g2<-rnorm(1,mu0,tau)
                    cond1<-exp(rnorm(nrep1,mu.g1,sigma))
                    cond2<-exp(rnorm(nrep2,mu.g2,sigma))
                    matrice[i,]<-c(cond1,cond2)
                    DE[i]<-TRUE
                }
        }
list(matrice=matrice,DE=DE)
}
```

A3.2.1.6 Utilizzo della funzione *sim2LNN* per la costruzione del grafico in Figura 3.3 e 3.21.

```
nrep1=3
nrep2=4
# Per il grafico in figura 3.2
mu0=2.3
sigma=0.3
tau=1.39

simulazioneLNN=sim2LNN(p=0.2)
#medie per tutti i geni
n<-10000
medieTutti<-matrix(rep(0,10000*2),ncol=2)
for(i in 1:n)
{     medieTutti[i,1]<-mean(simulazioneLNN$matrice[i,1:3])
      medieTutti[i,2]<-mean(simulazioneLNN$matrice[i,4:7])
}
plot(medieTutti[!simulazioneLNN$DE,1],medieTutti[!simulazioneLNN$D
E,2],pch=16,
cex=.5,log="xy",xlab="Condizione1",ylab="Condizione2",main="Geni
DE e EE modello LNN")
abline(0,1,lty=3,col=3)
points(medieTutti[simulazioneLNN$DE,1],
medieTutti[simulazioneLNN$DE,2],pch=2,cex=.5,col=6)
legend(1,2000,legend=c("Geni EE","Geni DE"),
```

```
col=c("black",6),cex=.7,pch=c(16,2),bty="n")

# Per il grafico in figura 3.21 riparametrizzo e ripeto il codice
per la costruzione del grafico
mu0=6.58
sigma=0.9
tau=1.13
```

A3.2.1.7 Utilizzo della funzione *sim2LNN* per la stima dei parametri dei modelli GG e LNN.

```
library(EBarrays)
matrice<-simulazioneLNN$matrice
pattern<-ebPatterns(c( paste(rep(1,nrep1+nrep2),collapse=","),
paste(c(rep(1,nrep1),rep(2,nrep2)),collapse=",")))
gg.fit <- emfit(data = matrice, family = "GG",
hypotheses = pattern,num.iter=10)
gg.post.out <- postprob(gg.fit,matrice)
lnn.fit <- emfit(data = matrice, family = "LNN",
hypotheses = pattern,num.iter=10)
lnn.post.out <- postprob(lnn.fit,matrice)
```

A3.2.1.8 Costruzione dei grafici delle densità marginali dei geni simulati da *sim2LNN* di Figura 3.4 (a) e (b) e 3.23 (a) e (b).

```
trellis.device(theme= col.whitebg())
print(plotMarginal(gg.fit,matrice))
print(plotMarginal(lnn.fit,matrice))
```

A3.2.1.9 Calcoli relativi alla corretta o scorretta identificazione da parte dei modelli su simulazioni GG riportate nella Tabella 3.1 (a) e (b) e Tabella 3.18 (a) e (b).

```
#Geni differenzialmente espressi
sum(simulazioneGG$DE)
#Geni considerati differenzialmente espressi dai 2 modelli
sum(gg.post.out[, 2] > 0.5)
```

```
sum(lnn.post.out[, 2] > 0.5)
#Numero di geni correttamente identificati
#come differenzialmente espressi dai 2 modelli
sum(((gg.post.out[, 2] > 0.5) ==TRUE)&(simulazioneGG$DE ==TRUE))
sum(((lnn.post.out[, 2] > 0.5) ==TRUE)&(simulazioneGG$DE ==TRUE))
#Numero di geni correttamente identificati
#come equivalentemente espressi dai 2 modelli
sum(((gg.post.out[, 2] > 0.5)==FALSE)&(simulazioneGG$DE==FALSE))
sum(((lnn.post.out[, 2] > 0.5)==FALSE)&(simulazioneGG$DE==FALSE))
#Numero di geni identificati non correttamente
#come equivalentemente espressi dai 2 modelli
sum(((gg.post.out[, 2] > 0.5)==FALSE)&(simulazioneGG$DE==TRUE))
sum(((lnn.post.out[, 2] > 0.5)==FALSE)&(simulazioneGG$DE==TRUE))
#Numero di geni identificati non correttamente
#come differenzialmente espressi dai 2 modelli
sum(((gg.post.out[, 2] > 0.5)==TRUE)&(simulazioneGG$DE==FALSE))
sum(((lnn.post.out[, 2] > 0.5)==TRUE)&(simulazioneGG$DE==FALSE))
```

A3.2.1.10 Calcoli relativi alla corretta o scorretta identificazione da parte dei modelli su simulazioni LNN riportate nella Tabella 3.2 (a) e (b) e Tabella 3.19 (a) e (b).

```
#Geni differenzialmente espressi
sum(simulazioneLNN$DE)
#Geni considerati differenzialmente espressi dai 2 modelli
sum(gg.post.out[, 2] > 0.5)
sum(lnn.post.out[, 2] > 0.5)
#Numero di geni correttamente identificati
#come differenzialmente espressi dai 2 modelli
sum(((gg.post.out[, 2] > 0.5) ==TRUE)&(simulazioneLNN$DE ==TRUE))
sum(((lnn.post.out[, 2] > 0.5) ==TRUE)&(simulazioneLNN$DE ==TRUE))
#Numero di geni correttamente identificati
#come equivalentemente espressi dai 2 modelli
sum(((gg.post.out[, 2] > 0.5)==FALSE)&(simulazioneLNN$DE==FALSE))
sum(((lnn.post.out[, 2] > 0.5)==FALSE)&(simulazioneLNN$DE==FALSE))
#Numero di geni identificati non correttamente
```

```
#come equivalentemente espressi dai 2 modelli
sum(((gg.post.out[, 2] > 0.5)==FALSE)&(simulazioneLNN$DE==TRUE))
sum(((lnn.post.out[, 2] > 0.5)==FALSE)&(simulazioneLNN$DE==TRUE))
#Numero di geni identificati non correttamente
#come differenzialmente espressi dai 2 modelli
sum(((gg.post.out[, 2] > 0.5)==TRUE)&(simulazioneLNN$DE==FALSE))
sum(((lnn.post.out[, 2] > 0.5)==TRUE)&(simulazioneLNN$DE==FALSE))
```

A3.2.1.11 Costruzione dei grafici in Figura 3.5, 3.6, 3.24 e 3.25 che mostrano i geni simulati indicati come differenzialmente espressi dai due modelli.

```
# per i geni simulati da GG
indice<-1:10000
# modellazione con GG
posteriorProb=0.5
indice05<-indice[gg.post.out[,2]>posteriorProb]
n=length(indice05)
medie05<-matrix(rep(0,n*2),ncol=2)
for(i in 1:n)
{     medie05[i,1]<-mean(simulazioneGG$matrice[indice05[i],1:3])
      medie05[i,2]<-mean(simulazioneGG$matrice[indice05[i],4:7])
}
posteriorProb=0.7
indice07<-indice[gg.post.out[,2]>posteriorProb]
n=length(indice07)
medie07<-matrix(rep(0,n*2),ncol=2)
for(i in 1:n)
{     medie07[i,1]<-mean(simulazioneGG$matrice[indice07[i],1:3])
      medie07[i,2]<-mean(simulazioneGG$matrice[indice07[i],4:7])
}
posteriorProb=0.9
indice09<-indice[gg.post.out[,2]>posteriorProb]
n=length(indice09)
```

```
medie09<-matrix(rep(0,n*2),ncol=2)
for(i in 1:n)
{     medie09[i,1]<-mean(simulazioneGG$matrice[indice09[i],1:3])
      medie09[i,2]<-mean(simulazioneGG$matrice[indice09[i],4:7])
}
par(mfrow=c(1,2))
plot(medieTutti[,1],medieTutti[,2],pch=16,cex=.5,log="xy",xlab="Co
ndizione1",
ylab="Condizione2",main="GG su dati GG")
abline(0,1,lty=3,col=3)
points(medie05[,1],medie05[,2],pch=16,cex=.5,col="blue")
points(medie07[,1],medie07[,2],pch=16,cex=.5,col="red")
points(medie09[,1],medie09[,2],pch=16,cex=.5,col="yellow")
legend(1,100000,legend=c("PostProb>0.5","PostProb>0.7","PostProb>0
.9"),
col=c("blue","red","yellow"),cex=.7,pch=c(16,16,16))

# modellazione con LNN
posteriorProb=0.5
indice05<-indice[lnn.post.out[,2]>posteriorProb]
n=length(indice05)
medie05<-matrix(rep(0,n*2),ncol=2)
for(i in 1:n)
{ medie05[i,1]<-mean(simulazioneGG$matrice[indice05[i],1:3])
medie05[i,2]<-mean(simulazioneGG$matrice[indice05[i],4:7])
}

posteriorProb=0.7
indice07<-indice[lnn.post.out[,2]>posteriorProb]
n=length(indice07)
medie07<-matrix(rep(0,n*2),ncol=2)
for(i in 1:n)
{ medie07[i,1]<-mean(simulazioneGG$matrice[indice07[i],1:3])
medie07[i,2]<-mean(simulazioneGG$matrice[indice07[i],4:7])
}
posteriorProb=0.9
```

```
indice09<-indice[lnn.post.out[,2]>posteriorProb]
n=length(indice09)
medie09<-matrix(rep(0,n*2),ncol=2)
for(i in 1:n)
{ medie09[i,1]<-mean(simulazioneGG$matrice[indice09[i],1:3])
medie09[i,2]<-mean(simulazioneGG$matrice[indice09[i],4:7])
}
plot(medieTutti[,1],medieTutti[,2],pch=16,cex=.5,log="xy",
xlab="Condizione1",ylab="Condizione2",main="LNN su dati GG")
abline(0,1,lty=3,col=3)
points(medie05[,1],medie05[,2],pch=16,cex=.5,col="blue")
points(medie07[,1],medie07[,2],pch=16,cex=.5,col="red")
points(medie09[,1],medie09[,2],pch=16,cex=.5,col="yellow")
legend(1,100000,legend=c("PostProb>0.5","PostProb>0.7",
"PostProb>0.9"),col=c("blue","red","yellow"),cex=.7,
pch=c(16,16,16))
```

 # per i geni simulati da LNN

```
indice<-1:10000
# modellazione con GG
posteriorProb=0.5
indice05<-indice[gg.post.out[,2]>posteriorProb]
n=length(indice05)
medie05<-matrix(rep(0,n*2),ncol=2)
for(i in 1:n)
{     medie05[i,1]<-mean(simulazioneLNN$matrice[indice05[i],1:3])
      medie05[i,2]<-mean(simulazioneLNN$matrice[indice05[i],4:7])
}
posteriorProb=0.7
indice07<-indice[gg.post.out[,2]>posteriorProb]
n=length(indice07)
medie07<-matrix(rep(0,n*2),ncol=2)
for(i in 1:n)
{     medie07[i,1]<-mean(simulazioneLNN$matrice[indice07[i],1:3])
      medie07[i,2]<-mean(simulazioneLNN$matrice[indice07[i],4:7])
```

```
}
posteriorProb=0.9
indice09<-indice[gg.post.out[,2]>posteriorProb]
n=length(indice09)
medie09<-matrix(rep(0,n*2),ncol=2)
for(i in 1:n)
{     medie09[i,1]<-mean(simulazioneLNN$matrice[indice09[i],1:3])
      medie09[i,2]<-mean(simulazioneLNN$matrice[indice09[i],4:7])
}
par(mfrow=c(1,2))
plot(medieTutti[,1],medieTutti[,2],pch=16,cex=.5,log="xy",xlab="Co
ndizione1",
ylab="Condizione2",main="GG su dati LNN")
abline(0,1,lty=3,col=3)
points(medie05[,1],medie05[,2],pch=16,cex=.5,col="blue")
points(medie07[,1],medie07[,2],pch=16,cex=.5,col="red")
points(medie09[,1],medie09[,2],pch=16,cex=.5,col="yellow")
legend(.1,3000,legend=c("PostProb>0.5","PostProb>0.7","PostProb>0.
9"),
col=c("blue","red","yellow"),cex=.7,pch=c(16,16,16))

#modellazioe con LNN
posteriorProb=0.5
indice05<-indice[lnn.post.out[,2]>posteriorProb]
n=length(indice05)
medie05<-matrix(rep(0,n*2),ncol=2)
for(i in 1:n)
{     medie05[i,1]<-mean(simulazioneLNN$matrice[indice05[i],1:3])
      medie05[i,2]<-mean(simulazioneLNN$matrice[indice05[i],4:7])
}
posteriorProb=0.7
indice07<-indice[lnn.post.out[,2]>posteriorProb]
n=length(indice07)
medie07<-matrix(rep(0,n*2),ncol=2)
for(i in 1:n)
{     medie07[i,1]<-mean(simulazioneLNN$matrice[indice07[i],1:3])
```

```
      medie07[i,2]<-mean(simulazioneLNN$matrice[indice07[i],4:7])
}
posteriorProb=0.9
indice09<-indice[lnn.post.out[,2]>posteriorProb]
n=length(indice09)
medie09<-matrix(rep(0,n*2),ncol=2)
for(i in 1:n)
{     medie09[i,1]<-mean(simulazioneLNN$matrice[indice09[i],1:3])
      medie09[i,2]<-mean(simulazioneLNN$matrice[indice09[i],4:7])
}
plot(medieTutti[,1],medieTutti[,2],pch=16,cex=.5,log="xy",
xlab="Condizione1",ylab="Condizione2",main="LNN su dati LNN")
abline(0,1,lty=3,col=3)
points(medie05[,1],medie05[,2],pch=16,cex=.5,col="blue")
points(medie07[,1],medie07[,2],pch=16,cex=.5,col="red")
points(medie09[,1],medie09[,2],pch=16,cex=.5,col="yellow")
legend(.1,3000,legend=c("PostProb>0.5","PostProb>0.7","PostProb>0.
9"),
col=c("blue","red","yellow"),cex=.7,pch=c(16,16,16))
```

A3.2.1.12 Costruzione dei grafici in Figura 3.7 (a) e (b) e nelle Figure 3.26 e 3.27 che mostrano i geni simulati non correttamente identificati dai modelli.

```
# per geni simulati da GG
plot(medieTutti[,1],medieTutti[,2],pch=16,cex=.5,log="xy",xlab="Co
ndizione1",
ylab="Condizione2",main="Errori GG su dati GG")

points(medieTutti[gg.post.out[,1]>0.5 & simulazioneGG$DE),1],
medieTutti[gg.post.out[,1]>0.5 &
(simulazioneGG$DE),2],col=2,pch=4,cex=.7)
```

```
points(medieTutti[gg.post.out[,2]>0.5 & !(simulazioneGG$DE),1],
medieTutti[gg.post.out[,2]>0.5 &
!(simulazioneGG$DE),2],col=3,pch=3,cex=.7)

legend(1,100000,legend=c("EE indicati come DE","DE indicati come
EE"),col=c("green","red"),cex=.7,pch=c(3,4),bty="n")

# per geni simulati da LNN
plot(medieTutti[,1],medieTutti[,2],pch=16,cex=.5,log="xy",xlab="Co
ndizione1",
ylab="Condizione2",main="Errori LNN su dati GG")

points(medieTutti[lnn.post.out[,1]>0.5 & (simulazioneGG$DE),1],
medieTutti[lnn.post.out[,1]>0.5 &
(simulazioneGG$DE),2],col=2,pch=4,cex=.7)

points(medieTutti[lnn.post.out[,2]>0.5 & !(simulazioneGG$DE),1],
medieTutti[lnn.post.out[,2]>0.5 &
!(simulazioneGG$DE),2],col=3,pch=3,cex=.7)

legend(1,100000,legend=c("EE indicati come DE",
"DE indicati come
EE"),col=c("green","red"),cex=.7,pch=c(3,4),bty="n")
```

A3.2.1.13 Costruzione del grafico con curve di *odds* di Figura 3.8 e 3.28.

```
#costruzione della funzione contouOdds che calcola le curve di
livello degli #odds per la funzione Gamma.
contourOdds<-function(theta,daticond1,daticond2)
{     a<-theta[1]
      a0<-theta[2]
      nu<-theta[3]
      p<-theta[4]
      n1<-nrep1
      n2<-nrep2
      N<-n1+n2
      daticond1<-exp(daticond1)
```

```
      daticond2<-exp(daticond2)
      k<-
((nu^a0)*gamma(n1*a+a0)*gamma(n2*a+a0))/(gamma(a0)*gamma(N*a+a0))
      p/(1-p)*k*(sum(3*daticond1)+sum(3*daticond2)+nu)^(N*a+a0)/
      ((sum(3*daticond1)+nu)^(n1*a+a0)*(sum(3*daticond2)+nu)^(n2*a
+a0))
}
#stima dei parametri con EM
theta<-c(gg.fit@ thetaEst,gg.fit@probEst[2])
theta<-as.vector(theta)
minimo<-log(min(simulazione$matrice))
massimo<-log(max(simulazione$matrice))
#variabile lunghezza per definire il dettaglio delle curve di
livello
lunghezza=500
sequenza<-seq(minimo,massimo,le=lunghezza)
matrice<-matrix(rep(0,lunghezza*lunghezza),ncol=lunghezza)
for(i in 1:lunghezza)
{     for(j in 1:lunghezza)
      {
      matrice[i,j]=contourOdds(theta,sequenza[i],sequenza[j])
      }
}
#utilizzo della funzione contour per la creazione della mappa
delle curve di #livello degli odds.
contour(exp(sequenza),exp(sequenza),matrice,levels=c(1,10,100),add
=T,drawlabels=FALSE)
```

A3.2.2 Indicatori di bontà e stima dei parametri

A3.2.2.1 Funzione *simIndicatori* per il calcolo degli indicatori di bontà dei modello GG e LNN su dati simulati da GG e LNN.

```
simIndicatori<- function
(nsim=100,ngeni=10000,nrep1,nrep2,alpha,alpha0,nu,
mu0,sigma,tau,p=0.2)
```

```
{     ncond=2
#simulazione di dati da modello GG e modello LNN e costruzione di
indici di #bontà del modello.
#costruisco i vettori che contengono i parametri stimati e gli
indicatori nelle #nsim simulazioni
      vsensGGonGG<-rep(0,nsim)
      vspecGGonGG<-rep(0,nsim)
      vppvGGonGG<-rep(0,nsim)
      vnpvGGonGG<-rep(0,nsim)
      vfdrGGonGG<-rep(0,nsim)
      vsensLNNonGG<-rep(0,nsim)
      vspecLNNonGG<-rep(0,nsim)
      vppvLNNonGG<-rep(0,nsim)
      vnpvLNNonGG<-rep(0,nsim)
      vfdrLNNonGG<-rep(0,nsim)
      vsensGGonLNN<-rep(0,nsim)
      vspecGGonLNN<-rep(0,nsim)
      vppvGGonLNN<-rep(0,nsim)
      vnpvGGonLNN<-rep(0,nsim)
      vfdrGGonLNN<-rep(0,nsim)
      vsensLNNonLNN<-rep(0,nsim)
      vspecLNNonLNN<-rep(0,nsim)
      vppvLNNonLNN<-rep(0,nsim)
      vnpvLNNonLNN<-rep(0,nsim)
      vfdrLNNonLNN<-rep(0,nsim)
      for(i in 1:nsim)
      { #DATI SIMULATI DA GG
            simulazioneGG<-
            sim2GG(ngeni=ngeni,nrep1=nrep1,nrep2=nrep2,alpha=alpha
            ,alpha0=alpha0,nu=nu,p=p)
            matrice<-simulazioneGG$matrice
            pattern<-ebPatterns(c(
      paste(rep(1,nrep1+nrep2),collapse=","),
            paste(c(rep(1,nrep1),rep(2,nrep2)),collapse=",")))
      #MODELLAZIONE CON GG
            fit <- emfit(data = matrice, family = "GG",
```

```
      hypotheses = pattern,num.iter=10)
#calcolo indici di bontµa del modello
      post.out <- postprob(fit,matrice)
#tabella per il calcolo degli indici
#(EE:Equivalent Expression,DE:Different Expression)
# modello
# EE DE
#dati EE a b
# DE c d
#
      a<-sum(((post.out[, 2] < 0.5) ==
TRUE)&(simulazioneGG$DE == FALSE))
      b<-sum(((post.out[, 2] > 0.5) ==
TRUE)&(simulazioneGG$DE == FALSE))
      c<-sum(((post.out[, 2] < 0.5) ==
TRUE)&(simulazioneGG$DE == TRUE))
      d<-sum(((post.out[, 2] > 0.5) ==
TRUE)&(simulazioneGG$DE == TRUE))
      vsensGGonGG[i]<-d/(c+d)
      vspecGGonGG[i]<-a/(a+b)
      vppvGGonGG[i]<-d/(b+d)
      vnpvGGonGG[i]<-a/(a+c)
      vfdrGGonGG[i]<-b/(b+d)
#MODELLAZIONE CON LNN
      fit <- emfit(data = matrice, family = "LNN",
      hypotheses = pattern,num.iter=10)
#calcolo indici di bontµa del modello
      post.out <- postprob(fit,matrice)
      a<-sum(((post.out[, 2] < 0.5) ==
TRUE)&(simulazioneGG$DE == FALSE))
      b<-sum(((post.out[, 2] > 0.5) ==
TRUE)&(simulazioneGG$DE == FALSE))
      c<-sum(((post.out[, 2] < 0.5) ==
TRUE)&(simulazioneGG$DE == TRUE))
      d<-sum(((post.out[, 2] > 0.5) ==
TRUE)&(simulazioneGG$DE == TRUE))
```

```
      vsensLNNonGG[i]<-d/(c+d)
      vspecLNNonGG[i]<-a/(a+b)
      vppvLNNonGG[i]<-d/(b+d)
      vnpvLNNonGG[i]<-a/(a+c)
      vfdrLNNonGG[i]<-b/(b+d)
#DATI SIMULATI LNN
      simulazioneLNN<-
sim2LNN(ngeni=ngeni,nrep1=nrep1,nrep2=nrep2,
      mu0=mu0,sigma=sigma,tau=tau,p=p)
      matrice<-simulazioneLNN$matrice
      pattern<-ebPatterns(c(
paste(rep(1,nrep1+nrep2),collapse=","),
      paste(c(rep(1,nrep1),rep(2,nrep2)),collapse=",")))
#modellazione con GG
      fit <- emfit(data = matrice, family = "GG",
      hypotheses = pattern,num.iter=10)
#calcolo indici di bontµa del modello
      post.out <- postprob(fit,matrice)
      a<-sum(((post.out[, 2] < 0.5) ==
TRUE)&(simulazioneLNN$DE == FALSE))
      b<-sum(((post.out[, 2] > 0.5) ==
TRUE)&(simulazioneLNN$DE == FALSE))
      c<-sum(((post.out[, 2] < 0.5) ==
TRUE)&(simulazioneLNN$DE == TRUE))
      d<-sum(((post.out[, 2] > 0.5) ==
TRUE)&(simulazioneLNN$DE == TRUE))
      vsensGGonLNN[i]<-d/(c+d)
      vspecGGonLNN[i]<-a/(a+b)
      vppvGGonLNN[i]<-d/(b+d)
      vnpvGGonLNN[i]<-a/(a+c)
      vfdrGGonLNN[i]<-b/(b+d)
#modellazione con LNN
      fit <- emfit(data = matrice, family = "LNN",
      hypotheses = pattern,num.iter=10)
#calcolo indici di bonta del modello
      post.out <- postprob(fit,matrice)
```

```
      a<-sum(((post.out[, 2] < 0.5) ==
TRUE)&(simulazioneLNN$DE == FALSE))
      b<-sum(((post.out[, 2] > 0.5) ==
TRUE)&(simulazioneLNN$DE == FALSE))
      c<-sum(((post.out[, 2] < 0.5) ==
TRUE)&(simulazioneLNN$DE == TRUE))
      d<-sum(((post.out[, 2] > 0.5) ==
TRUE)&(simulazioneLNN$DE == TRUE))
      vsensLNNonLNN[i]<-d/(c+d)
      vspecLNNonLNN[i]<-a/(a+b)
      vppvLNNonLNN[i]<-d/(b+d)
      vnpvLNNonLNN[i]<-a/(a+c)
      vfdrLNNonLNN[i]<-b/(b+d)
      }
list(indicatoriGGonGG=
list(sens=mean(vsensGGonGG),se.sens=sqrt(var(vsensGGonGG)),
spec=mean(vspecGGonGG),se.spec=sqrt(var(vspecGGonGG)),
PPV=mean(vppvGGonGG),se.PPV=sqrt(var(vppvGGonGG)),
NPV=mean(vnpvGGonGG),se.NPV=sqrt(var(vnpvGGonGG)),
FDR=mean(vfdrGGonGG),se.FDR=sqrt(var(vfdrGGonGG))),
indicatoriGGonLNN=
list(sens=mean(vsensGGonLNN),se.sens=sqrt(var(vsensGGonLNN))
,
spec=mean(vspecGGonLNN),se.spec=sqrt(var(vspecGGonLNN)),
PPV=mean(vppvGGonLNN),se.PPV=sqrt(var(vppvGGonLNN)),
NPV=mean(vnpvGGonLNN),se.NPV=sqrt(var(vnpvGGonLNN)),
FDR=mean(vfdrGGonLNN),se.FDR=sqrt(var(vfdrGGonLNN))),
indicatoriLNNonGG=
list(sens=mean(vsensLNNonGG),se.sens=sqrt(var(vsensLNNonGG))
,
spec=mean(vspecLNNonGG),se.spec=sqrt(var(vspecLNNonGG)),
PPV=mean(vppvLNNonGG),se.PPV=sqrt(var(vppvLNNonGG)),
NPV=mean(vnpvLNNonGG),se.NPV=sqrt(var(vnpvLNNonGG)),
FDR=mean(vfdrLNNonGG),se.FDR=sqrt(var(vfdrLNNonGG))),
indicatoriLNNonLNN=
```

```
      list(sens=mean(vsensLNNonLNN),se.sens=sqrt(var(vsensLNNonLNN
)),
      spec=mean(vspecLNNonLNN),se.spec=sqrt(var(vspecLNNonLNN)),
      PPV=mean(vppvLNNonLNN),se.PPV=sqrt(var(vppvLNNonLNN)),
      NPV=mean(vnpvLNNonLNN),se.NPV=sqrt(var(vnpvLNNonLNN)),
      FDR=mean(vfdrLNNonLNN),se.FDR=sqrt(var(vfdrLNNonLNN))))
}
```

A3.2.2.2 Utilizzo della funzione *simIndicatori* per ottenere le Tabelle 3.4, 3.5, 3.6, 3.7, 3.20, 3.21, 3.22 e 3.23.

```
nsim<-10
ngeni<-1000
nrep1=3
nrep2=15
#Per la prima parametrizzazione dei modelli
alpha=10
alpha0=0.9
nu=0.5,
mu0=2.3
sigma=0.3
tau=1.39
simIndicatori(nsim,ngeni,p=0.1)
simIndicatori(nsim,ngeni,p=0.2)
simIndicatori(nsim,ngeni,p=0.3)
simIndicatori(nsim,ngeni,p=0.4)
simIndicatori(nsim,ngeni,p=0.5)

#Per la prima parametrizzazione dei modelli
alpha=1
alpha0=1.1
nu=45.4,
mu0=6.58
sigma=0.9
tau=1.13
simIndicatori(nsim,ngeni,p=0.1)
```

```
simIndicatori(nsim,ngeni,p=0.2)
simIndicatori(nsim,ngeni,p=0.3)
simIndicatori(nsim,ngeni,p=0.4)
simIndicatori(nsim,ngeni,p=0.5)
```

A3.2.2.3 Utilizzo della funzione *simIndicatori* per la costruzione dei grafici in Figura 3.9, 3.10, 3.29 e 3.30 che mettono a confronto gli indici di bontà dei modelli.

```
#PER GENI SIMULATI DA GG
plot(seq(0.1,0.5,by=0.1),c(p1$indicatoriGGonGG$spec,p2$indicatoriG
GonGG$spec,p3$indicatoriGGonGG$spec,p4$indicatoriGGonGG$spec,p1$in
dicatoriGGonGG$spec),
xlab="pde",ylab="",ylim=c(0,1),pch=1,main="GG su GG(scuro) e GG su
LNN(chiaro)")
points(seq(0.1+0.005,0.5+0.005,by=0.1),c(p1$indicatoriGGonLNN$spec
,p2$indicatoriGGonLNN$spec,p3$indicatoriGGonLNN$spec,p4$indicatori
GGonLNN$spec,p1$indicatoriGGonLNN$spec),pch=1,col=3)

points(seq(0.1,0.5,by=0.1),c(p1$indicatoriGGonGG$PPV,p2$indicatori
GGonGG$PPV,p3$indicatoriGGonGG$PPV,p4$indicatoriGGonGG$PPV,p1$indi
catoriGGonGG$PPV),pch=2)

points(seq(0.1+0.005,0.5+0.005,by=0.1),c(p1$indicatoriGGonLNN$PPV,
p2$indicatoriGGonLNN$PPV,p3$indicatoriGGonLNN$PPV,p4$indicatoriGGo
nLNN$PPV,p1$indicatoriGGonLNN$PPV),pch=2,col=3)

points(seq(0.1,0.5,by=0.1),c(p1$indicatoriGGonGG$NPV,p2$indicatori
GGonGG$NPV,p3$indicatoriGGonGG$NPV,p4$indicatoriGGonGG$NPV,p1$indi
catoriGGonGG$NPV),pch=3)points(seq(0.1+0.005,0.5+0.005,by=0.1),c(p
1$indicatoriGGonLNN$NPV,p2$indicatoriGGonLNN$NPV,p3$indicatoriGGon
LNN$NPV,p4$indicatoriGGonLNN$NPV,p1$indicatoriGGonLNN$NPV),pch=3,c
ol=3)
```

```
points(seq(0.1,0.5,by=0.1),c(p1$indicatoriGGonGG$sens,p2$indicator
iGGonGG$sens,p3$indicatoriGGonGG$sens,p4$indicatoriGGonGG$sens,p1$
indicatoriGGonGG$sens),pch=4)

points(seq(0.1+0.005,0.5+0.005,by=0.1),c(p1$indicatoriGGonLNN$sens
,p2$indicatoriGGonLNN$sens,p3$indicatoriGGonLNN$sens,p4$indicatori
GGonLNN$sens,p1$indicatoriGGonLNN$sens),pch=4,col=3)

points(seq(0.1,0.5,by=0.1),c(p1$indicatoriGGonGG$FDR,p2$indicatori
GGonGG$FDR,p3$indicatoriGGonGG$FDR,p4$indicatoriGGonGG$FDR,p1$indi
catoriGGonGG$FDR),pch=5)

points(seq(0.1+0.005,0.5+0.005,by=0.1),c(p1$indicatoriGGonLNN$FDR,
p2$indicatoriGGonLNN$FDR,p3$indicatoriGGonLNN$FDR,p4$indicatoriGGo
nLNN$FDR,p1$indicatoriGGonLNN$FDR),pch=5,col=3)

legend(0.2, 0.6, c("Specificity", "PPV",
"NPV","Sensitivity","FDR"),pch = seq(1:5,by=1),bty="n")


#PER GENI SIMULATI DA LNN

plot(seq(0.1,0.5,by=0.1),c(p1$indicatoriLNNonLNN$spec,p2$indicator
iLNNonLNN$spec,p3$indicatoriLNNonLNN$spec,p4$indicatoriLNNonLNN$sp
ec,p1$indicatoriLNNonLNN$spec),xlab="pde",ylab="",ylim=c(0,1),pch=
1,main="LNN su LNN(scuro) e LNN su GG(chiaro)")

points(seq(0.1+0.005,0.5+0.005,by=0.1),c(p1$indicatoriLNNonGG$spec
,p2$indicatoriLNNonGG$spec,p3$indicatoriLNNonGG$spec,p4$indicatori
LNNonGG$spec,p1$indicatoriLNNonGG$spec),pch=1,col=3)

points(seq(0.1,0.5,by=0.1),c(p1$indicatoriLNNonLNN$PPV,p2$indicato
riLNNonLNN$PPV,p3$indicatoriLNNonLNN$PPV,p4$indicatoriLNNonLNN$PPV
,p1$indicatoriLNNonLNN$PPV),pch=2)
```

```
points(seq(0.1+0.005,0.5+0.005,by=0.1),c(p1$indicatoriLNNonGG$PPV,
p2$indicatoriLNNonGG$PPV,p3$indicatoriLNNonGG$PPV,p4$indicatoriLNN
onGG$PPV,p1$indicatoriLNNonGG$PPV),pch=2,col=3)

points(seq(0.1,0.5,by=0.1),c(p1$indicatoriLNNonLNN$NPV,p2$indicato
riLNNonLNN$NPV,p3$indicatoriLNNonLNN$NPV,p4$indicatoriLNNonLNN$NPV
,p1$indicatoriLNNonLNN$NPV),pch=3)

points(seq(0.1+0.005,0.5+0.005,by=0.1),c(p1$indicatoriLNNonGG$NPV,
p2$indicatoriLNNonGG$NPV,p3$indicatoriLNNonGG$NPV,p4$indicatoriLNN
onGG$NPV,p1$indicatoriLNNonGG$NPV),pch=3,col=3)
points(seq(0.1,0.5,by=0.1),c(p1$indicatoriLNNonLNN$sens,p2$indicat
oriLNNonLNN$sens,p3$indicatoriLNNonLNN$sens,p4$indicatoriLNNonLNN$
sens,p1$indicatoriLNNonLNN$sens),pch=4)

points(seq(0.1+0.005,0.5+0.005,by=0.1),c(p1$indicatoriLNNonGG$sens
,p2$indicatoriLNNonGG$sens,p3$indicatoriLNNonGG$sens,p4$indicatori
LNNonGG$sens,p1$indicatoriLNNonGG$sens),pch=4,col=3)

points(seq(0.1,0.5,by=0.1),c(p1$indicatoriLNNonLNN$FDR,p2$indicato
riLNNonLNN$FDR,p3$indicatoriLNNonLNN$FDR,p4$indicatoriLNNonLNN$FDR
,p1$indicatoriLNNonLNN$FDR),pch=5)

points(seq(0.1+0.005,0.5+0.005,by=0.1),c(p1$indicatoriLNNonGG$FDR,
p2$indicatoriLNNonGG$FDR,p3$indicatoriLNNonGG$FDR,p4$indicatoriLNN
onGG$FDR,p1$indicatoriLNNonGG$FDR),pch=5,col=3)

legend(0.2, 0.6, c("Specificity", "PPV",
"NPV","Sensitivity","FDR"),pch = seq(1:5,by=1),bty="n")
```

A3.2.2.4 Utilizzo della funzione *sim4GG* e *sim4LNN* per la stima dei parametri relativi alle Tabelle 3.8, 3.9, 3.24 e 3.25.

```
#sim4GG
function (nsim=100,ngeni=10000,nrep1,nrep2,alpha,alpha0,nu,p=0.2)
{     ncond=2
      #simulazione di dati da modello GG e stima dei parametri
```

```
      #del modello GG
      #vettori che contengono i parametri stimati
      valpha<-rep(0,nsim)
      valpha0<-rep(0,nsim)
      vnu<-rep(0,nsim)
      vp<-rep(0,nsim)
      for(i in 1:nsim)
      {     #dati simulati da GG
            simulazioneGG<-
            sim2GG(ngeni=ngeni,nrep1=nrep1,nrep2=nrep2,alpha=alpha
            ,alpha0=alpha0,nu=nu,p=p)
            matrice<-simulazioneGG$matrice
            pattern<-
      ebPatterns(c(paste(rep(1,nrep1+nrep2),collapse=","),
            paste(c(rep(1,nrep1),rep(2,nrep2)),collapse=",")))
            #modellazione e stime dei parametri
            gg.fit <- emfit(data = matrice, family =
            "GG",hypotheses = pattern,num.iter=10)
            valpha[i]<-gg.fit@thetaEst[1]
            valpha0[i]<-gg.fit@thetaEst[2]
            vnu[i]<- gg.fit@thetaEst[3]
            vp[i]<-gg.fit@probEst[2]
      }
list(alpha=mean(valpha),se.alpha=sqrt(var(valpha)),alpha0=mean(val
pha0),se.alpha0=sqrt(var(valpha0)),nu=mean(vnu),se.nu=sqrt(var(vnu
)),p=mean(vp),se.p=sqrt(var(vp)))
}


nrep1=3
nrep2=15


#per la prima parametrizzazione
alpha=10
alpha0=0.9
nu=0.5
```

```
sim4GG(nsim=10,ngeni=10000,p=0.1)
sim4GG(nsim=10,ngeni=10000,p=0.2)
sim4GG(nsim=10,ngeni=10000,p=0.3)
sim4GG(nsim=10,ngeni=10000,p=0.4)
sim4GG(nsim=10,ngeni=10000,p=0.5)


#per la seconda parametrizzazione
alpha=1
alpha0=1.1
nu=45.5

sim4GG(nsim=10,ngeni=10000,p=0.1)
sim4GG(nsim=10,ngeni=10000,p=0.2)
sim4GG(nsim=10,ngeni=10000,p=0.3)
sim4GG(nsim=10,ngeni=10000,p=0.4)
sim4GG(nsim=10,ngeni=10000,p=0.5)



#sim4LNN
function (nsim=100,ngeni=10000,nrep1,nrep2,mu0,sigma,tau,p=0.2)
{     ncond=2
      #simulazione di dati da modello LNN e
      #stima dei parametri del modello LNN
      #vettori che contengono i parametri stimati
      vmu0<-rep(0,nsim)
      vsigma<-rep(0,nsim)
      vtau<-rep(0,nsim)
      vp<-rep(0,nsim)
      for(i in 1:nsim)
      {     #dati simulati da LNN
            simulazioneLNN<-
            sim2LNN(ngeni=ngeni,nrep1=nrep1,nrep2=nrep2,mu0,sigma,
            tau,p=p)
            matrice<-simulazioneLNN$matrice
```

```
            pattern<-ebPatterns(c(
            paste(rep(1,nrep1+nrep2),collapse=","),
            paste(c(rep(1,nrep1),rep(2,nrep2)),collapse=",")))
            #modellazione e stime dei parametri
            lnn.fit <- emfit(data = matrice, family =
            "LNN",hypotheses=pattern,num.iter=10)
            vmu0[i]<-lnn.fit@family@invlink(lnn.fit@thetaEst)[1]
            vsigma[i]=sqrt(lnn.fit@family@invlink
            (lnn.fit@thetaEst)[2])
            vtau[i]=sqrt(lnn.fit@family@invlink(lnn.fit@thetaEst)[
            3])
            vp[i]<-lnn.fit@probEst[2]
      }
list(mu0=mean(vmu0),se.mu0=sqrt(var(vmu0)),sigma=mean(vsigma)
,se.sigma=sqrt(var(vsigma)),tau=mean(vtau),
se.tau=sqrt(var(vtau)),p=mean(vp),se.p=sqrt(var(vp)))
}

nrep1=3
nrep2=15

#per la prima parametrizzazione
mu0=2.3
sigma=0.3
tau=1.39

sim4LNN(nsim=10,ngeni=10000,p=0.1)
sim4LNN(nsim=10,ngeni=10000,p=0.2)
sim4LNN(nsim=10,ngeni=10000,p=0.3)
sim4LNN(nsim=10,ngeni=10000,p=0.4)
sim4LNN(nsim=10,ngeni=10000,p=0.5)


#per la seconda parametrizzazione
mu0=6.58
sigma=0.9
```

```
tau=1.13

sim4LNN(nsim=10,ngeni=10000,p=0.1)
sim4LNN(nsim=10,ngeni=10000,p=0.2)
sim4LNN(nsim=10,ngeni=10000,p=0.3)
sim4LNN(nsim=10,ngeni=10000,p=0.4)
sim4LNN(nsim=10,ngeni=10000,p=0.5)
```

A3.2.3.9 Istruzioni per la costruzione dei grafici in Figura 3.18 (a) e (b) che mettono a confronto le *performance* dei modelli nelle due simulazioni con *h=1000* e *w=9/10*.

```
#per geni simulati da GG
plot(seq(0.1,0.5,by=0.1),c(p1$indicatoriGG$spec,p2$indicatoriGG$sp
ec,
p3$indicatoriGG$spec,p4$indicatoriGG$spec,p1$indicatoriGG$spec),xl
ab="pde",
ylab="",ylim=c(0,1),pch=1,main="GG su dati simulati(scuro) and LNN
su dati simulati(chiaro)",cex.main=.8)

points(seq(0.1+0.005,0.5+0.005,by=0.1),c(p1$indicatoriLNN$spec,
p2$indicatoriLNN$spec,p3$indicatoriLNN$spec,p4$indicatoriLNN$spec,
p1$indicatoriLNN$spec),pch=1,col=3)

points(seq(0.1,0.5,by=0.1),c(p1$indicatoriGG$PPV,
p2$indicatoriGG$PPV,
p3$indicatoriGG$PPV,p4$indicatoriGG$PPV,
p1$indicatoriGG$PPV),pch=2)

points(seq(0.1+0.005,0.5+0.005,by=0.1),c(p1$indicatoriLNN$PPV,
p2$indicatoriLNN$PPV,p3$indicatoriLNN$PPV,p4$indicatoriLNN$PPV,
p1$indicatoriLNN$PPV),pch=2,col=3)
```

```
points(seq(0.1,0.5,by=0.1),c(p1$indicatoriLNN$NPV,p2$indicatoriLNN
$NPV,
p3$indicatoriLNN$NPV,p4$indicatoriLNN$NPV,
p1$indicatoriLNN$NPV),pch=3)

points(seq(0.1+0.005,0.5+0.005,by=0.1),c(p1$indicatoriLNN$NPV,
p2$indicatoriLNN$NPV,p3$indicatoriLNN$NPV,p4$indicatoriLNN$NPV,
p1$indicatoriLNN$NPV),pch=3,col=3)

points(seq(0.1,0.5,by=0.1),c(p1$indicatoriLNN$sens,p2$indicatoriLN
N$sens,
p3$indicatoriLNN$sens,p4$indicatoriLNN$sens
,p1$indicatoriLNN$sens),pch=4)

points(seq(0.1+0.005,0.5+0.005,by=0.1),c(p1$indicatoriLNN$sens,
p2$indicatoriLNN$sens,p3$indicatoriLNN$sens,p4$indicatoriLNN$sens,
p1$indicatoriLNN$sens),pch=4,col=3)

points(seq(0.1,0.5,by=0.1),c(p1$indicatoriLNN$FDR,p2$indicatoriLNN
$FDR,
p3$indicatoriLNN$FDR,p4$indicatoriLNN$FDR,
p1$indicatoriLNN$FDR),pch=5)

points(seq(0.1+0.005,0.5+0.005,by=0.1),c(p1$indicatoriLNN$FDR,
p2$indicatoriLNN$FDR,p3$indicatoriLNN$FDR,p4$indicatoriLNN$FDR,
p1$indicatoriLNN$FDR),pch=5,col=3)

legend(0.2, 0.6, c("Specificity", "PPV", "NPV","Sensitivity",
"FDR"),
pch = seq(1:5,by=1),bty="n")



#per geni simulati da LNN
plot(seq(0.1,0.5,by=0.1),c(p1$indicatoriGG$spec,p2$indicatoriGG$sp
ec,
```

```
p3$indicatoriGG$spec,p4$indicatoriGG$spec,p1$indicatoriGG$spec),xl
ab="pde",
ylab="",ylim=c(0,1),pch=1, main="GG su dati simulati(scuro) and
LNN su dati
simulati(chiaro)",cex.main=.8)

points(seq(0.1+0.005,0.5+0.005,by=0.1),c(p1$indicatoriLNN$spec,
p2$indicatoriLNN$spec,p3$indicatoriLNN$spec,p4$indicatoriLNN$spec,
p1$indicatoriLNN$spec),pch=1,col=3)

points(seq(0.1,0.5,by=0.1),c(p1$indicatoriGG$PPV,p2$indicatoriGG$P
PV,
p3$indicatoriGG$PPV,p4$indicatoriGG$PPV,p1$indicatoriGG$PPV),pch=2
)

points(seq(0.1+0.005,0.5+0.005,by=0.1),c(p1$indicatoriLNN$PPV,
p2$indicatoriLNN$PPV,p3$indicatoriLNN$PPV,p4$indicatoriLNN$PPV,
p1$indicatoriLNN$PPV),pch=2,col=3)

points(seq(0.1,0.5,by=0.1),c(p1$indicatoriLNN$NPV,p2$indicatoriLNN
$NPV,
p3$indicatoriLNN$NPV,p4$indicatoriLNN$NPV,
p1$indicatoriLNN$NPV),pch=3)

points(seq(0.1+0.005,0.5+0.005,by=0.1),c(p1$indicatoriLNN$NPV,
p2$indicatoriLNN$NPV,p3$indicatoriLNN$NPV,p4$indicatoriLNN$NPV,
p1$indicatoriLNN$NPV),pch=3,col=3)

points(seq(0.1,0.5,by=0.1),c(p1$indicatoriLNN$sens,p2$indicatoriLN
N$sens,
p3$indicatoriLNN$sens,p4$indicatoriLNN$sens,p1$indicatoriLNN$sens)
,pch=4)
points(seq(0.1+0.005,0.5+0.005,by=0.1),c(p1$indicatoriLNN$sens,
p2$indicatoriLNN$sens,p3$indicatoriLNN$sens,p4$indicatoriLNN$sens,
p1$indicatoriLNN$sens),pch=4,col=3)
```

```
points(seq(0.1,0.5,by=0.1),c(p1$indicatoriLNN$FDR,p2$indicatoriLNN
$FDR,
p3$indicatoriLNN$FDR,p4$indicatoriLNN$FDR,
p1$indicatoriLNN$FDR),pch=5)

points(seq(0.1+0.005,0.5+0.005,by=0.1),c(p1$indicatoriLNN$FDR,
p2$indicatoriLNN$FDR,p3$indicatoriLNN$FDR,p4$indicatoriLNN$FDR,
p1$indicatoriLNN$FDR),pch=5,col=3)

legend(0.2, 0.6, c("Specificity", "PPV", "NPV","Sensitivity",
"FDR"),
pch = seq(1:5,by=1),bty="n")
```

A3.2.3 Il di Coefficiente di variazione costante.

A3.2.3.1 Grafici di Figura 3.11, 3.12, 3.31 e 3.32.

```
#utilizzo della funzioe ceckCCV della libreria EBarrays
#su dati simulati da GG
checkCCV(simulazioneGG$matrice)
abline(h=1/sqrt(10),lty=2)
#su dati simulati da GG come differenzialmente espressi
checkCCV((simulazioneGG$matrice)[!simulazioneGG$DE,])
abline(h=1/sqrt(10),lty=2)

#su dati simulati da LNN
checkCCV(simulazioneLNN$matrice)
abline(h= sqrt(exp(0.3^2)-1),lty=2)
#su dati simulati da LNN come differenzialmente espressi
checkCCV((simulazioneLNN$matrice)[!simulazioneLNN$DE,])
abline(h= sqrt(exp(0.3^2)-1),lty=2)
```

A3.2.3.2 Funzione *simGGnotCostantCV* per la simulazione dal modello GG con coefficiente di variazione non costante, sintetizzata nello Schema 3.3.

```
function (ngeni=10000,nrep1,nrep2,alpha,alpha0,nu,p=0.2,w,h)
{     ncond=2
      matrice<-
      matrix(rep(NA,ngeni*(nrep1+nrep2)),ncol=nrep1+nrep2,byrow=T)
      #vettore che indica se un gene µe differenzialmente espresso
      DE<-rep(FALSE,ngeni)
      for(i in 1:ngeni)
      {     if(runif(1)>p)
            #Equivalent Expression
            {     lambda<-rgamma(1,shape=alpha0,rate=nu)
                  matrice[i,]<-
            rgamma(nrep1+nrep2,shape=alpha,rate=lambda)
            }
            else
```

```
            #Different Expression
            {     if(runif(1)<w)
                  {     lambda1<-rgamma(1,shape=alpha0,rate=nu)
                        lambda2<-rgamma(1,shape=alpha0,rate=nu)
                        cond1<-
                        rgamma(nrep1,shape=alpha,rate=lambda1)
                        cond2<-
                        rgamma(nrep2,shape=alpha,rate=lambda2)
                        matrice[i,]<-c(cond1,cond2)
                        DE[i]<-TRUE
                  }
                  else
                  {     lambda1<-rgamma(1,shape=alpha0,rate=nu*h)
                        lambda2<-rgamma(1,shape=alpha0,rate=nu*h)
                        cond1<-
                        rgamma(nrep1,shape=alpha,rate=lambda1)
                        cond2<-
                        rgamma(nrep2,shape=alpha,rate=lambda2)
                        matrice[i,]<-c(cond1,cond2)
                        DE[i]<-TRUE
                  }
            }
      }
list(matrice=matrice,DE=DE)
}
```

A3.2.3.3 Funzione *simLNNnotCostantCV* per la simulazione dal modello LNN con coefficiente di variazione non costante, sintetizzata nello Schema 3.4.

```
function (ngeni=10000,nrep1,nrep2,mu0,sigma,tau,p=0.2,h,w)
{     ncond=2
      matrice<-
      matrix(rep(NA,ngeni*(nrep1+nrep2)),ncol=nrep1+nrep2,byrow=T)
      #vettore che indica se un gene µe differenzialmente espresso
      DE<-rep(FALSE,ngeni)
      for(i in 1:ngeni)
      {     if(runif(1)>p)
      #Equivalent Expression
            {     mu.g<-rnorm(1,mu0,tau)
                  matrice[i,]<-exp(rnorm(nrep1+nrep2,mu.g,sigma))
            }
            else
            #Different Expression
            {     if(runif(1)<w)
                  {     mu.g1<-rnorm(1,mu0,tau)
                        mu.g2<-rnorm(1,mu0,tau)
                        cond1<-exp(rnorm(nrep1,mu.g1,sigma))
                        cond2<-exp(rnorm(nrep2,mu.g2,sigma))
                        matrice[i,]<-c(cond1,cond2)
                        DE[i]<-TRUE
                  }
                  else
                  {     mu.g1<-rnorm(1,mu0,tau)
                        mu.g2<-rnorm(1,mu0,tau)
                        cond1<-exp(rnorm(nrep1,mu.g1,h*sigma))
                        cond2<-exp(rnorm(nrep2,mu.g2,h*sigma))
                        matrice[i,]<-c(cond1,cond2)
                        DE[i]<-TRUE
                  }
            }
      }
```

```
list(matrice=matrice,DE=DE)
}
```

A3.2.3.4 Utilizzo delle funzioni *simGGnotCostantCV* e *simLNNnotCostantCV* con *h=1/100* e *w=1/10* per la costruzione dei grafici relativi alle simulazioni in Figura 3.13, 3.14, 3.33 e 3.34.

```
nrep1=3
nrep2=15

#per la prima parametrizzazione
mu0=2.3
sigma=0.3
tau=1.39

simulazione=simGGnotCostantCV(h=1000,w=9/10)
simulazioneGG=simulazione
n<-10000
medieTutti<-matrix(rep(0,10000*2),ncol=2)
for(i in 1:n)
      {     medieTutti[i,1]<-mean(simulazioneGG$matrice[i,1:3])
            medieTutti[i,2]<-mean(simulazioneGG$matrice[i,4:7])
      }
par(mfrow=c(2,2),pty="s")

plot(medieTutti[,1],medieTutti[,2],log="xy",xlab="Condizione1",yla
b="Condizione2",main="Geni DE e EE",type="n")
abline(0,1,lty=3,col=3)
points(medieTutti[!simulazioneGG$DE,1],medieTutti[!simulazioneGG$D
E,2],pch=16,
cex=.5)
points(medieTutti[simulazioneGG$DE,1],medieTutti[simulazioneGG$DE,
2],pch=2,cex=.5,col=3)
legend(0.01,45000,legend=c("Geni EE","Geni
DE"),col=c("black","green"),cex=.7,
pch=c(16,2),bty="n")
```

```
hist(log(simulazione$matrice),xlab="Logaritmo espressione
genica",main="")
checkCCV(simulazione$matrice)


simulazioneLNN=simLNNnotCostantCV(h=100,w=9/10)
simulazione1=simulazioneLNN
n<-10000
medieTutti<-matrix(rep(0,10000*2),ncol=2)
for(i in 1:n)
{ medieTutti[i,1]<-mean(simulazioneLNN$matrice[i,1:3])
medieTutti[i,2]<-mean(simulazioneLNN$matrice[i,4:7])
}
par(mfrow=c(2,2),pty="s")

plot(medieTutti[,1],medieTutti[,2],log="xy",xlab="Condizione1"
,ylab="Condizione2",main="Geni DE e EE",type="n")
abline(0,1,lty=3,col=3)
points(medieTutti[!simulazioneLNN$DE,1],
medieTutti[!simulazioneLNN$DE,2],
pch=16,cex=.5)
points(medieTutti[simulazioneLNN$DE,1],
medieTutti[simulazioneLNN$DE,2],pch=2,cex=. 5,col=3)
legend(0.01,45000,legend=c("Geni EE","Geni DE"),
col=c("black","green"),cex=.7,
pch=c(16,2),bty="n")
hist(log(simulazione1$matrice),
xlab="Logaritmo espressione genica",main="")
checkCCV(simulazione1$matrice)

#per la seconda parametrizzazione si reimpostano i parametri e si
utlizza il codice per la costruzione dei grafici sopra proposto.

mu0=6.58
sigma=0.9
tau=1.13
```

A3.2.3.5 Stima dei modelli GG e LNN e costruzione dei grafici delle densità per le simulazioni con GG in Figura 3.15 e 3.35, per le simulazioni con LNN in Figura 3.16 e 3.36 con *h=1/100* e *w=1/10*.

```
#per geni simulati da GG
nrep1=3
nrep2=15
pattern<-ebPatterns(c( paste(rep(1,nrep1+nrep2),collapse=","),
paste(c(rep(1,nrep1),rep(2,nrep2)),collapse=",")))
#per geni simulati da GG e modellati con GG
gg.fit.noCV <- emfit(data = simulazione$matrice, family = "GG",
hypotheses = pattern,num.iter=10)
trellis.device(theme= col.whitebg())
print(plotMarginal(gg.fit.noCV,simulazione$matrice))
#per geni simulati da GG e modellati con LNN
lnn.fit.noCV <- emfit(data = simulazione$matrice, family = "LNN",
hypotheses = pattern,num.iter=10)
print(plotMarginal(lnn.fit.noCV,simulazione$matrice))

gg.post.out.noCV <- postprob(gg.fit.noCV,simulazione$matrice)
lnn.post.out.noCV <- postprob(lnn.fit.noCV,simulazione$matrice)

#per geni simulati da LNN
nrep1=3
nrep2=15
pattern<-ebPatterns(c( paste(rep(1,nrep1+nrep2),collapse=","),
paste(c(rep(1,nrep1),rep(2,nrep2)),collapse=",")))

#per geni simulati da LNN e modellati con GG
gg.fit.noCV <- emfit(data = simulazione1$matrice, family = "GG",
hypotheses = pattern,num.iter=10)
trellis.device(theme= col.whitebg())
print(plotMarginal(gg.fit.noCV,simulazione1$matrice))
#per geni simulati da LNN e modellati con LNN
lnn.fit.noCV <- emfit(data = simulazione$matrice, family = "LNN",
```

```
hypotheses = pattern,num.iter=10)
print(plotMarginal(lnn.fit.noCV,simulazione$matrice))

gg.post.out.noCV <- postprob(gg.fit.noCV,simulazione1$matrice)
lnn.post.out.noCV <- postprob(lnn.fit.noCV,simulazione1$matrice)


#procedo nello stesso modo nelle 2 parametrizzazioni.
```

A3.2.3.6 Calcoli relativi la corretta o scorretta identificazione da parte dei modelli nelle simulazioni. Le Tabelle 3.10 (a) e (b) e 3.26 (a) e (b) si riferiscono a simulazioni da GG, le Tabelle 3.11 (a) e (b) e 3.27 (a) e (b) si riferiscono a simulazioni da LNN con *h=1/100* e *w=1/10*.

```
#Geni differenzialmente espressi simulati da GG
sum(simulazione$DE)

#Geni considerati differenzialmente espressi dai 2 modelli
sum(gg.post.out.noCV[, 2] > 0.5)
sum(lnn.post.out.noCV[, 2] > 0.5)

#Numero di geni correttamente identificati come differenzialmente
espressi dai 2 #modelli
sum(((gg.post.out.noCV[, 2] > 0.5) == TRUE)&(simulazione$DE ==
TRUE))
sum(((lnn.post.out.noCV[, 2] > 0.5) == TRUE)&(simulazione$DE ==
TRUE))

#Numero di geni correttamente identificati come equivalentemente
espressi dai 2 #modelli
sum(((gg.post.out.noCV[, 2] > 0.5) == FALSE)&(simulazione$DE ==
FALSE))
sum(((lnn.post.out.noCV[, 2] > 0.5) == FALSE)&(simulazione$DE ==
FALSE))
```

```
#Numero di geni identificati non correttamente come
equivalentemente espressi #dai 2 modelli
sum(((gg.post.out.noCV[, 2] > 0.5) == FALSE)&(simulazione$DE ==
TRUE))
sum(((lnn.post.out.noCV[, 2] > 0.5) == FALSE)&(simulazione$DE ==
TRUE))

#Numero di geni identificati non correttamente come
differenzialmente espressi #dai 2 modelli
sum(((gg.post.out.noCV[, 2] > 0.5) == TRUE)&
(simulazione$DE == FALSE))
sum(((lnn.post.out.noCV[, 2] > 0.5) == TRUE)&
(simulazione$DE == FALSE))


#Geni differenzialmente espressi simulati da LNN
sum(simulazione1$DE)

#Geni considerati differenzialmente espressi dai 2 modelli
sum(gg.post.out.noCV[, 2] > 0.5)
sum(lnn.post.out.noCV[, 2] > 0.5)



#Numero di geni correttamente identificati come differenzialmente
espressi dai 2 #modelli
sum(((gg.post.out.noCV[, 2] > 0.5) == TRUE)&(simulazione$DE ==
TRUE))
sum(((lnn.post.out.noCV[, 2] > 0.5) == TRUE)&(simulazione$DE ==
TRUE))

#Numero di geni correttamente identificati come equivalentemente
espressi dai 2 #modelli
sum(((gg.post.out.noCV[, 2] > 0.5) == FALSE)&(simulazione$DE ==
FALSE))
```

```
sum(((lnn.post.out.noCV[, 2] > 0.5) == FALSE)&(simulazione$DE ==
FALSE))

#Numero di geni identificati non correttamente come
equivalentemente espressi #dai 2 modelli
sum(((gg.post.out.noCV[, 2] > 0.5) == FALSE)&(simulazione$DE ==
TRUE))
sum(((lnn.post.out.noCV[, 2] > 0.5) == FALSE)&(simulazione$DE ==
TRUE))

#Numero di geni identificati non correttamente come
differenzialmente espressi #dai 2 modelli
sum(((gg.post.out.noCV[, 2] > 0.5) == TRUE)&(simulazione$DE ==
FALSE))
sum(((lnn.post.out.noCV[, 2] > 0.5) == TRUE)&(simulazione$DE ==
FALSE))
```

A3.2.3.7 Costruzione dei grafici in Figura 3.17, 3.18, 3.36 e 3.37.

```
#per le simulazioni da GG
indice<-1:10000
#modellazione con GG
posteriorProb=0.5
indice05<-indice[gg.post.out.noCV[,2]>posteriorProb]
n=length(indice05)
medie05<-matrix(rep(0,n*2),ncol=2)
for(i in 1:n)
      {     medie05[i,1]<-
mean(simulazioneGG$matrice[indice05[i],1:3])
            medie05[i,2]<-
      mean(simulazioneGG$matrice[indice05[i],4:7])
      }

posteriorProb=0.7
indice07<-indice[gg.post.out.noCV[,2]>posteriorProb]
n=length(indice07)
medie07<-matrix(rep(0,n*2),ncol=2)
```

```
for(i in 1:n)
{ medie07[i,1]<-mean(simulazioneGG$matrice[indice07[i],1:3])
medie07[i,2]<-mean(simulazioneGG$matrice[indice07[i],4:7])
}

posteriorProb=0.9
indice09<-indice[gg.post.out.noCV[,2]>posteriorProb]
n=length(indice09)
medie09<-matrix(rep(0,n*2),ncol=2)
for(i in 1:n)
{ medie09[i,1]<-mean(simulazioneGG$matrice[indice09[i],1:3])
medie09[i,2]<-mean(simulazioneGG$matrice[indice09[i],4:7])
}

plot(medieTutti[,1],medieTutti[,2],pch=16,cex=.
5,log="xy",xlab="Condizione1",
ylab="Condizione2",main="GG su dati con CV non costante
(h=100,w=9/10)", cex.main=.8)
abline(0,1,lty=3,col=3)
points(medie05[,1],medie05[,2],pch=16,cex=.5,col="blue")
points(medie07[,1],medie07[,2],pch=16,cex=.5,col="red")
points(medie09[,1],medie09[,2],pch=16,cex=.5,col="yellow")
legend(1e-
08,1e+35,legend=c("PostProb>0.5","PostProb>0.7","PostProb>0.9"),
col=c("blue","red","yellow"),cex=.7,pch=c(16,16,16),bty="n")

#modellazione con LNN
posteriorProb=0. 5
indice05<-indice[lnn.post.out.noCV[,2]>posteriorProb]
n=length(indice05)
medie05<-matrix(rep(0,n*2),ncol=2)
for(i in 1:n)
{     medie05[i,1]<-mean(simulazioneGG$matrice[indice05[i],1:3])
      medie05[i,2]<-mean(simulazioneGG$matrice[indice05[i],4:7])
}
```

```
posteriorProb=0.7
indice07<-indice[lnn.post.out.noCV[,2]>posteriorProb]
n=length(indice07)
medie07<-matrix(rep(0,n*2),ncol=2)
for(i in 1:n)
{     medie07[i,1]<-mean(simulazioneGG$matrice[indice07[i],1:3])
      medie07[i,2]<-mean(simulazioneGG$matrice[indice07[i],4:7])
}

posteriorProb=0.9
indice09<-indice[lnn.post.out.noCV[,2]>posteriorProb]
n=length(indice09)
medie09<-matrix(rep(0,n*2),ncol=2)
for(i in 1:n)
{     medie09[i,1]<-mean(simulazioneGG$matrice[indice09[i],1:3])
      medie09[i,2]<-mean(simulazioneGG$matrice[indice09[i],4:7])
}

plot(medieTutti[,1],medieTutti[,2],pch=16,cex=.
5,log="xy",xlab="Condizione1",
ylab="Condizione2",main="LNN su dati con CV non
costante(h=100,w=9/10)",
cex.main=.8)
abline(0,1,lty=3,col=3)
points(medie05[,1],medie05[,2],pch=16,cex=.5,col="blue")
points(medie07[,1],medie07[,2],pch=16,cex=.5,col="red")
points(medie09[,1],medie09[,2],pch=16,cex=.5,col="yellow")
legend(1e-
08,1e+35,legend=c("PostProb>0.5","PostProb>0.7","PostProb>0.9"),
col=c("blue","red","yellow"),cex=.7,pch=c(16,16,16),bty="n")

#per le simulazioni da LNN
indice<-1:10000
#modellazione con GG
posteriorProb=0.5
indice05<-indice[gg.post.out.noCV[,2]>posteriorProb]
```

```
n=length(indice05)
medie05<-matrix(rep(0,n*2),ncol=2)
for(i in 1:n)
{     medie05[i,1]<-mean(simulazioneGG$matrice[indice05[i],1:3])
      medie05[i,2]<-mean(simulazioneGG$matrice[indice05[i],4:7])
}

posteriorProb=0.7
indice07<-indice[gg.post.out.noCV[,2]>posteriorProb]
n=length(indice07)
medie07<-matrix(rep(0,n*2),ncol=2)
for(i in 1:n)
{     medie07[i,1]<-mean(simulazioneGG$matrice[indice07[i],1:3])
      medie07[i,2]<-mean(simulazioneGG$matrice[indice07[i],4:7])
}

posteriorProb=0.9
indice09<-indice[gg.post.out.noCV[,2]>posteriorProb]
n=length(indice09)
medie09<-matrix(rep(0,n*2),ncol=2)
for(i in 1:n)
{     medie09[i,1]<-mean(simulazioneGG$matrice[indice09[i],1:3])
      medie09[i,2]<-mean(simulazioneGG$matrice[indice09[i],4:7])
}

plot(medieTutti[,1],medieTutti[,2],pch=16,cex=.
5,log="xy",xlab="Condizione1",
ylab="Condizione2",main="GG su dati con CV non
costante(h=100,w=9/10)", cex.main=.8)
abline(0,1,lty=3,col=3)
points(medie05[,1],medie05[,2],pch=16,cex=.5,col="blue")
points(medie07[,1],medie07[,2],pch=16,cex=.5,col="red")
points(medie09[,1],medie09[,2],pch=16,cex=.5,col="yellow")
legend(1e-
08,1e+35,legend=c("PostProb>0.5","PostProb>0.7","PostProb>0.9"),
col=c("blue","red","yellow"),cex=.7,pch=c(16,16,16),bty="n")
```

```
#modellazione con LNN
posteriorProb=0.5
indice05<-indice[lnn.post.out.noCV[,2]>posteriorProb]
n=length(indice05)
medie05<-matrix(rep(0,n*2),ncol=2)
for(i in 1:n)
{     medie05[i,1]<-mean(simulazioneGG$matrice[indice05[i],1:3])
      medie05[i,2]<-mean(simulazioneGG$matrice[indice05[i],4:7])
}

posteriorProb=0.7
indice07<-indice[lnn.post.out.noCV[,2]>posteriorProb]
n=length(indice07)
medie07<-matrix(rep(0,n*2),ncol=2)
for(i in 1:n)
{     medie07[i,1]<-mean(simulazioneGG$matrice[indice07[i],1:3])
      medie07[i,2]<-mean(simulazioneGG$matrice[indice07[i],4:7])
}

posteriorProb=0.9
indice09<-indice[lnn.post.out.noCV[,2]>posteriorProb]
n=length(indice09)
medie09<-matrix(rep(0,n*2),ncol=2)
for(i in 1:n)
{     medie09[i,1]<-mean(simulazioneGG$matrice[indice09[i],1:3])
      medie09[i,2]<-mean(simulazioneGG$matrice[indice09[i],4:7])
}

plot(medieTutti[,1],medieTutti[,2],pch=16,cex=.
5,log="xy",xlab="Condizione1",
ylab="Condizione2",main="LNN su dati con CV non
costante(h=100,w=9/10)",
cex.main=.8)
abline(0,1,lty=3,col=3)
points(medie05[,1],medie05[,2],pch=16,cex=.5,col="blue")
```

```
points(medie07[,1],medie07[,2],pch=16,cex=.5,col="red")
points(medie09[,1],medie09[,2],pch=16,cex=.5,col="yellow")
legend(1e-
08,1e+35,legend=c("PostProb>0.5","PostProb>0.7","PostProb>0.9"),
col=c("blue","red","yellow"),cex=.7,pch=c(16,16,16),bty="n")
```

A3.2.3.8 Funzione *indicatoriNotCostantCV* per il calcolo degli indici di bontà dei modelli GG e LNN sui dati simulati con coefficiente di variazione non costante con *h=1000* e *w=9/10* e istruzioni per ottenere le tabelle 3.12, 3.13, 3.14, 3.15, 3.28, 3.29, 3.30 e 3.31.

```
#per la simulazione di geni da GG
function
(nsim=100,ngeni=10000,nrep1,nrep2,alpha,alpha0,nu,p=0.2,w,h)
{     ncond=2
      #simulazione di dati con coefficiente di variazione
      #non costante e costruzione di indici di bontà del modello.
      #vettori che contengono i parametri stimati
      #e gli indicatori nelle nsim simulazione
      vsensGG<-rep(0,nsim)
      vspecGG<-rep(0,nsim)
      vppvGG<-rep(0,nsim)
      vnpvGG<-rep(0,nsim)
      vfdrGG<-rep(0,nsim)
      vsensLNN<-rep(0,nsim)
      vspecLNN<-rep(0,nsim)
      vppvLNN<-rep(0,nsim)
      vnpvLNN<-rep(0,nsim)
      vfdrLNN<-rep(0,nsim)
      for(i in 1:nsim)
      {     #dati simulati
            simulazione<-
            simGGnotCostantCV(ngeni=ngeni,nrep1=nrep1,
            nrep2=nrep2,alpha=alpha,alpha0=alpha0,nu=nu,p=p,w=w,h=
            h)
```

```
matrice<-simulazione$matrice
pattern<-ebPatterns(c(
paste(rep(1,nrep1+nrep2),collapse=","),
paste(c(rep(1,nrep1),rep(2,nrep2)),collapse=",")))
#modellazione con GG
gg.fit <- emfit(data = matrice, family = "GG",
hypotheses = pattern,num.iter=10)
#calcolo indici di bontà del modello
gg.post.out <- postprob(gg.fit,matrice)
#tabella per il calcolo degli indici
#(EE:Equivalent Expression,DE:Different Expression)
# modello
# EE DE
#dati EE a b
# DE c d
a<-sum(((gg.post.out[, 2] < 0.5))&(!simulazione$DE))
b<-sum(((gg.post.out[, 2] > 0.5))&(!simulazione$DE))
c<-sum(((gg.post.out[, 2] < 0.5))&(simulazione$DE))
d<-sum(((gg.post.out[, 2] > 0.5))&(simulazione$DE))
vsensGG[i]<-d/(c+d)
vspecGG[i]<-a/(a+b)
vppvGG[i]<-d/(b+d)
vnpvGG[i]<-a/(a+c)
vfdrGG[i]<-b/(b+d)
#modellazione con LNN
lnn.fit <- emfit(data = matrice, family = "LNN",
hypotheses = pattern,num.iter=10)
#calcolo indici di bont`a del modello
lnn.post.out <- postprob(lnn.fit,matrice)
a<-sum(((lnn.post.out[, 2] < 0.5))&(!simulazione$DE))
b<-sum(((lnn.post.out[, 2] > 0.5))&(!simulazione$DE))
c<-sum(((lnn.post.out[, 2] < 0.5))&(simulazione$DE))
d<-sum(((lnn.post.out[, 2] > 0.5))&(simulazione$DE))
vsensLNN[i]<-d/(c+d)
vspecLNN[i]<-a/(a+b)
vppvLNN[i]<-d/(b+d)
```

```
            vnpvLNN[i]<-a/(a+c)
            vfdrLNN[i]<-b/(b+d)
      }
list(indicatoriGG=list(sens=mean(vsensGG),
se.sens=sqrt(var(vsensGG)),spec=mean(vspecGG),
se.spec=sqrt(var(vspecGG)),PPV=mean(vppvGG),
se.PPV=sqrt(var(vppvGG)),NPV=mean(vnpvGG),
se.NPV=sqrt(var(vnpvGG)),FDR=mean(vfdrGG),
se.FDR=sqrt(var(vfdrGG))),
indicatoriLNN=list(sens=mean(vsensLNN),
se.sens=sqrt(var(vsensLNN)),spec=mean(vspecLNN),
se.spec=sqrt(var(vspecLNN)),PPV=mean(vppvLNN),
se.PPV=sqrt(var(vppvLNN)),NPV=mean(vnpvLNN),
se.NPV=sqrt(var(vnpvLNN)),FDR=mean(vfdrLNN),
se.FDR=sqrt(var(vfdrLNN))))
}

nsim<-10
ngeni<-1000
h=1/100
w=1/10
nrep1=3
nrep2=15

#per la prima parametrizzazione
alpha=10
alpha0=0.9
nu=0.5

indicatoriNotConstantCV (nsim=nsim,ngeni=ngeni,p=0. 1,h=h,w=w)
indicatoriNotConstantCV (nsim,ngeni,p=0. 2,h=h,w=w)
indicatoriNotConstantCV (nsim,ngeni,p=0. 3,h=h,w=w)
indicatoriNotConstantCV (nsim,ngeni,p=0. 4,h=h,w=w)
indicatoriNotConstantCV (nsim,ngeni,p=0. 5,h=h,w=w)

#per la seconda parametrizzazione si reimpostano i parametri
```

```
mu0=6.58
sigma=0.9
tau=1.13
indicatoriNotConstantCV (nsim=nsim,ngeni=ngeni,p=0. 1,h=h,w=w)
indicatoriNotConstantCV (nsim,ngeni,p=0. 2,h=h,w=w)
indicatoriNotConstantCV (nsim,ngeni,p=0. 3,h=h,w=w)
indicatoriNotConstantCV (nsim,ngeni,p=0. 4,h=h,w=w)
indicatoriNotConstantCV (nsim,ngeni,p=0. 5,h=h,w=w)
```

#per la simulazione di geni da LNN

```
# Funzione indicatoriNotConstantCV
function (nsim=100,ngeni=10000,nrep1,nrep2, mu0,sigma,tau,p=0.2
,w,h)
{     ncond=2
      #simulazione di dati con coefficiente di variazione
      #non costante e costruzione di indici di bontà del modello.
      #vettori che contengono i parametri stimati
      #e gli indicatori nelle nsim simulazione
      vsensGG<-rep(0,nsim)
      vspecGG<-rep(0,nsim)
      vppvGG<-rep(0,nsim)
      vnpvGG<-rep(0,nsim)
      vfdrGG<-rep(0,nsim)
      vsensLNN<-rep(0,nsim)
      vspecLNN<-rep(0,nsim)
      vppvLNN<-rep(0,nsim)
      vnpvLNN<-rep(0,nsim)
      vfdrLNN<-rep(0,nsim)
      for(i in 1:nsim)
      {     #dati simulati
            simulazione<- simLNNnotCostantCV(h=1/100,w=1/10)
            matrice<-simulazione$matrice
            pattern<-
            ebPatterns(c(paste(rep(1,nrep1+nrep2),collapse=","),
```

```
        paste(c(rep(1,nrep1),rep(2,nrep2)),collapse=",")))
        #modellazione con GG
        gg.fit <- emfit(data = matrice, family = "GG",
        hypotheses = pattern,num.iter=10)
        #calcolo indici di bontà del modello
        gg.post.out <- postprob(gg.fit,matrice)
        #tabella per il calcolo degli indici
        #(EE:Equivalent Expression,DE:Different Expression)
        # modello
        # EE DE
        #dati EE a b
        # DE c d
        a<-sum(((gg.post.out[, 2] < 0.5))&(!simulazione$DE))
        b<-sum(((gg.post.out[, 2] > 0.5))&(!simulazione$DE))
        c<-sum(((gg.post.out[, 2] < 0.5))&(simulazione$DE))
        d<-sum(((gg.post.out[, 2] > 0.5))&(simulazione$DE))
        vsensGG[i]<-d/(c+d)
        vspecGG[i]<-a/(a+b)
        vppvGG[i]<-d/(b+d)
        vnpvGG[i]<-a/(a+c)
        vfdrGG[i]<-b/(b+d)
        #modellazione con LNN
        lnn.fit <- emfit(data = matrice, family = "LNN",
        hypotheses = pattern,num.iter=10)
        #calcolo indici di bont`a del modello
        lnn.post.out <- postprob(lnn.fit,matrice)
        a<-sum(((lnn.post.out[, 2] < 0.5))&(!simulazione$DE))
        b<-sum(((lnn.post.out[, 2] > 0.5))&(!simulazione$DE))
        c<-sum(((lnn.post.out[, 2] < 0.5))&(simulazione$DE))
        d<-sum(((lnn.post.out[, 2] > 0.5))&(simulazione$DE))
        vsensLNN[i]<-d/(c+d)
        vspecLNN[i]<-a/(a+b)
        vppvLNN[i]<-d/(b+d)
        vnpvLNN[i]<-a/(a+c)
        vfdrLNN[i]<-b/(b+d)
}
```

```
list(indicatoriGG=list(sens=mean(vsensGG),
se.sens=sqrt(var(vsensGG)),spec=mean(vspecGG),
se.spec=sqrt(var(vspecGG)),PPV=mean(vppvGG),
se.PPV=sqrt(var(vppvGG)),NPV=mean(vnpvGG),
se.NPV=sqrt(var(vnpvGG)),FDR=mean(vfdrGG),
se.FDR=sqrt(var(vfdrGG))),
indicatoriLNN=list(sens=mean(vsensLNN),
se.sens=sqrt(var(vsensLNN)),spec=mean(vspecLNN),
se.spec=sqrt(var(vspecLNN)),PPV=mean(vppvLNN),
se.PPV=sqrt(var(vppvLNN)),NPV=mean(vnpvLNN),
se.NPV=sqrt(var(vnpvLNN)),FDR=mean(vfdrLNN),
se.FDR=sqrt(var(vfdrLNN))))
}

nsim<-10
ngeni<-1000
h=1/100
w=1/10
nrep1=3
nrep2=15

#per la prima parametrizzazione
mu0=2.3
sigma=0.3
tau=1.39

indicatoriNotConstantCV (nsim=nsim,ngeni=ngeni,p=0. 1,h=h,w=w)
indicatoriNotConstantCV (nsim,ngeni,p=0. 2,h=h,w=w)
indicatoriNotConstantCV (nsim,ngeni,p=0. 3,h=h,w=w)
indicatoriNotConstantCV (nsim,ngeni,p=0. 4,h=h,w=w)
indicatoriNotConstantCV (nsim,ngeni,p=0. 5,h=h,w=w)

#per la seconda parametrizzazione si reimpostano i parametri
mu0=6.58
sigma=0.9
tau=1.13
```

```
indicatoriNotConstantCV (nsim=nsim,ngeni=ngeni,p=0. 1,h=h,w=w)
indicatoriNotConstantCV (nsim,ngeni,p=0. 2,h=h,w=w)
indicatoriNotConstantCV (nsim,ngeni,p=0. 3,h=h,w=w)
indicatoriNotConstantCV (nsim,ngeni,p=0. 4,h=h,w=w)
indicatoriNotConstantCV (nsim,ngeni,p=0. 5,h=h,w=w)
```

A3. 2.3.9 Istruzioni per la costruzione dei grafici in Figura 3.19, 3.20 e 3.38 (a) e (b) che mettono a confronto le *performance* dei modelli nelle due simulazioni con *h=1/100* e *w=1/10.*

```
#per geni simulati da GG
plot(seq(0.1,0.5,by=0.1),c(p1$indicatoriGG$spec,p2$indicatoriGG$sp
ec,
p3$indicatoriGG$spec,p4$indicatoriGG$spec,p1$indicatoriGG$spec),xl
ab="pde",
ylab="",ylim=c(0,1),pch=1,main="GG su dati simulati(scuro) and LNN
su dati simulati(chiaro)",cex.main=.8)

points(seq(0.1+0.005,0.5+0.005,by=0.1),c(p1$indicatoriLNN$spec,
p2$indicatoriLNN$spec,p3$indicatoriLNN$spec,p4$indicatoriLNN$spec,
p1$indicatoriLNN$spec),pch=1,col=3)

points(seq(0.1,0.5,by=0.1),c(p1$indicatoriGG$PPV,
p2$indicatoriGG$PPV,
p3$indicatoriGG$PPV,p4$indicatoriGG$PPV,
p1$indicatoriGG$PPV),pch=2)

points(seq(0.1+0.005,0.5+0.005,by=0.1),c(p1$indicatoriLNN$PPV,
p2$indicatoriLNN$PPV,p3$indicatoriLNN$PPV,p4$indicatoriLNN$PPV,
p1$indicatoriLNN$PPV),pch=2,col=3)

points(seq(0.1,0.5,by=0.1),c(p1$indicatoriLNN$NPV,p2$indicatoriLNN
$NPV,
```

```
p3$indicatoriLNN$NPV,p4$indicatoriLNN$NPV,
p1$indicatoriLNN$NPV),pch=3)

points(seq(0.1+0.005,0.5+0.005,by=0.1),c(p1$indicatoriLNN$NPV,
p2$indicatoriLNN$NPV,p3$indicatoriLNN$NPV,p4$indicatoriLNN$NPV,
p1$indicatoriLNN$NPV),pch=3,col=3)

points(seq(0.1,0.5,by=0.1),c(p1$indicatoriLNN$sens,p2$indicatoriLN
N$sens,
p3$indicatoriLNN$sens,p4$indicatoriLNN$sens
,p1$indicatoriLNN$sens),pch=4)

points(seq(0.1+0.005,0.5+0.005,by=0.1),c(p1$indicatoriLNN$sens,
p2$indicatoriLNN$sens,p3$indicatoriLNN$sens,p4$indicatoriLNN$sens,
p1$indicatoriLNN$sens),pch=4,col=3)

points(seq(0.1,0.5,by=0.1),c(p1$indicatoriLNN$FDR,p2$indicatoriLNN
$FDR,
p3$indicatoriLNN$FDR,p4$indicatoriLNN$FDR,
p1$indicatoriLNN$FDR),pch=5)

points(seq(0.1+0.005,0.5+0.005,by=0.1),c(p1$indicatoriLNN$FDR,
p2$indicatoriLNN$FDR,p3$indicatoriLNN$FDR,p4$indicatoriLNN$FDR,
p1$indicatoriLNN$FDR),pch=5,col=3)

legend(0.2, 0.6, c("Specificity", "PPV", "NPV","Sensitivity",
"FDR"),
pch = seq(1:5,by=1),bty="n")



#per geni simulati da LNN
plot(seq(0.1,0.5,by=0.1),c(p1$indicatoriGG$spec,p2$indicatoriGG$sp
ec,
p3$indicatoriGG$spec,p4$indicatoriGG$spec,p1$indicatoriGG$spec),xl
ab="pde",
```

```
ylab="",ylim=c(0,1),pch=1, main="GG su dati simulati(scuro) and
LNN su dati
simulati(chiaro)",cex.main=.8)

points(seq(0.1+0.005,0.5+0.005,by=0.1),c(p1$indicatoriLNN$spec,
p2$indicatoriLNN$spec,p3$indicatoriLNN$spec,p4$indicatoriLNN$spec,
p1$indicatoriLNN$spec),pch=1,col=3)

points(seq(0.1,0.5,by=0.1),c(p1$indicatoriGG$PPV,p2$indicatoriGG$P
PV,
p3$indicatoriGG$PPV,p4$indicatoriGG$PPV,p1$indicatoriGG$PPV),pch=2
)

points(seq(0.1+0.005,0.5+0.005,by=0.1),c(p1$indicatoriLNN$PPV,
p2$indicatoriLNN$PPV,p3$indicatoriLNN$PPV,p4$indicatoriLNN$PPV,
p1$indicatoriLNN$PPV),pch=2,col=3)

points(seq(0.1,0.5,by=0.1),c(p1$indicatoriLNN$NPV,p2$indicatoriLNN
$NPV,
p3$indicatoriLNN$NPV,p4$indicatoriLNN$NPV,
p1$indicatoriLNN$NPV),pch=3)

points(seq(0.1+0.005,0.5+0.005,by=0.1),c(p1$indicatoriLNN$NPV,
p2$indicatoriLNN$NPV,p3$indicatoriLNN$NPV,p4$indicatoriLNN$NPV,
p1$indicatoriLNN$NPV),pch=3,col=3)

points(seq(0.1,0.5,by=0.1),c(p1$indicatoriLNN$sens,p2$indicatoriLN
N$sens,
p3$indicatoriLNN$sens,p4$indicatoriLNN$sens,p1$indicatoriLNN$sens)
,pch=4)
points(seq(0.1+0.005,0.5+0.005,by=0.1),c(p1$indicatoriLNN$sens,
p2$indicatoriLNN$sens,p3$indicatoriLNN$sens,p4$indicatoriLNN$sens,
p1$indicatoriLNN$sens),pch=4,col=3)

points(seq(0.1,0.5,by=0.1),c(p1$indicatoriLNN$FDR,p2$indicatoriLNN
$FDR,
```

```
p3$indicatoriLNN$FDR,p4$indicatoriLNN$FDR,
p1$indicatoriLNN$FDR),pch=5)

points(seq(0.1+0.005,0.5+0.005,by=0.1),c(p1$indicatoriLNN$FDR,
p2$indicatoriLNN$FDR,p3$indicatoriLNN$FDR,p4$indicatoriLNN$FDR,
p1$indicatoriLNN$FDR),pch=5,col=3)

legend(0.2, 0.6, c("Specificity", "PPV", "NPV","Sensitivity",
"FDR"),
pch = seq(1:5,by=1),bty="n")
```

A3.2.3.10 Stima dei parametri dei modelli con CV non costante, *h=1/100* e *w=1/10*, riportati nelle Tabelle 3.17, 3.18, 3.32 e 3.33.

```
function
(nsim=100,ngeni=10000,nrep1,nrep2,alpha,alpha0,nu,h,w,p=0.2)
{ ncond=2
#simulazione di dati da modello GG e stima dei parametri
#del modello GG
#vettori che contengono i parametri stimati
valpha<-rep(0,nsim)
valpha0<-rep(0,nsim)
vnu<-rep(0,nsim)
vp<-rep(0,nsim)
for(i in 1:nsim)
{ #dati simulati da GG
simulazioneGG<-
simGGnotCostantCV(ngeni=ngeni,nrep1=nrep1,nrep2=nrep2,alpha=alpha,
alpha0=alpha0,nu=nu,p=p,h=h,w=w)
matrice<-simulazioneGG$matrice
pattern<-ebPatterns(c(paste(rep(1,nrep1+nrep2),collapse=","),
paste(c(rep(1,nrep1),rep(2,nrep2)),collapse=",")))
#modellazione e stime dei parametri
```

```
gg.fit <- emfit(data = matrice, family = "GG",hypotheses =
pattern,num.iter=10)
valpha[i]<-gg.fit@thetaEst[1]
valpha0[i]<-gg.fit@thetaEst[2]
vnu[i]<- gg.fit@thetaEst[3]
vp[i]<-gg.fit@probEst[2]
}
list(alpha=mean(valpha),se.alpha=sqrt(var(valpha)),
alpha0=mean(valpha0),se.alpha0=sqrt(var(valpha0)),
nu=mean(vnu),se.nu=sqrt(var(vnu)),p=mean(vp),
se.p=sqrt(var(vp)))
}


function
(nsim=100,ngeni=10000,nrep1,nrep2,mu0,sigma,tau,p=0.2,h,w)
{ ncond=2
#simulazione di dati da modello LNN e
#stima dei parametri del modello LNN
#vettori che contengono i parametri stimati
vmu0<-rep(0,nsim)
vsigma<-rep(0,nsim)
vtau<-rep(0,nsim)
vp<-rep(0,nsim)
for(i in 1:nsim)
{ #dati simulati da LNN
simulazioneLNN<-
simLNNnotCostantCV(ngeni=ngeni,nrep1=nrep1,nrep2=nrep2,mu0,sigma,t
au,p=p,h=h,w=w)
matrice<-simulazioneLNN$matrice
pattern<-ebPatterns(c( paste(rep(1,nrep1+nrep2),collapse=","),
paste(c(rep(1,nrep1),rep(2,nrep2)),collapse=",")))
#modellazione e stime dei parametri
lnn.fit <- emfit(data = matrice, family =
"LNN",hypotheses=pattern,num.iter=10)
vmu0[i]<-lnn.fit@family@invlink(lnn.fit@thetaEst)[1]
```

```
vsigma[i]=sqrt(lnn.fit@family@invlink
(lnn.fit@thetaEst)[2])
vtau[i]=sqrt(lnn.fit@family@invlink(lnn.fit@thetaEst)[3])
vp[i]<-lnn.fit@probEst[2]
}
list(mu0=mean(vmu0),se.mu0=sqrt(var(vmu0)),sigma=mean(vsigma)
,se.sigma=sqrt(var(vsigma)),tau=mean(vtau),
se.tau=sqrt(var(vtau)),p=mean(vp),se.p=sqrt(var(vp)))
}
#ripeto lo stesso codice con le due parametrizzazioni
h=1/100
w=1/10
s1<-sim4GG(nsim=10,ngeni=10000,p=0.1)
s2<-sim4GG(nsim=10,ngeni=10000,p=0.2)
s3<-sim4GG(nsim=10,ngeni=10000,p=0.3)
s4<-sim4GG(nsim=10,ngeni=10000,p=0.4)
s5<-sim4GG(nsim=10,ngeni=10000,p=0.5)
s6<-sim4LNN(nsim=10,ngeni=10000,p=0.1)
s7<-sim4LNN(nsim=10,ngeni=10000,p=0.2)
s8<-sim4LNN(nsim=10,ngeni=10000,p=0.3)
s9<-sim4LNN(nsim=10,ngeni=10000,p=0.4)
s10<-sim4LNN(nsim=10,ngeni=10000,p=0.5)
```

CAPITOLO 3

Capitolo 4

Metodi di imputazione di valori mancanti su una casistica reale.

Introduzione

Le leucemie sono malattie gravi del tessuto emopoietico che produce i globuli e le piastrine del sangue. Si tratta di un vero e proprio cancro del sangue. Leucemia, termine di origine greca, significa letteralmente "sangue bianco". Il nome si spiega col fatto che in questo genere di malattie il numero dei globuli bianchi (leucociti) può essere elevato. I globuli bianchi si formano nel midollo osseo come i globuli rossi (eritrociti). I leucociti hanno un'importante funzione nel sistema immunitario dell'organismo. Essi sono suddivisi, in base al grado di maturazione, nei cosiddetti granulociti, linfociti e monociti. Una parte dei linfociti si sviluppa nel tessuto linfatico (linfonodi, milza, tonsille, timo). Una volta maturi i linfociti entrano nel circolo sanguigno per svolgervi la propria funzione.

Le leucemie si suddividono nelle forme mieloidi e linfatiche. Nelle leucemie mieloidi sono i granulociti e le loro forme primitive ad essere colpiti. Le cellule cancerogene si moltiplicano in modo incontrollato impedendo la formazione di cellule normali nel midollo osseo. Nel caso della leucemia linfatica oltre al midollo osseo è colpito il sistema linfatico. In entrambe le patologie possono riversarsi nel sangue enormi quantità di "globuli bianchi" degenerati.

In base al decorso della malattia i medici distinguono forme acute e croniche di leucemia. Le leucemie croniche hanno un decorso più lento, mentre le forme acute

hanno gravi conseguenze: se non curate provocano la morte del paziente nel giro di pochi mesi. In base al decorso della malattia e al tipo di cellule colpite distinguiamo quattro grandi tipi di leucemia:

1. leucemie mieloidi croniche (LMC)
2. leucemie linfatiche croniche (LLC)
3. leucemie mieloidi acute (LMA)
4. leucemie linfatiche acute (LLA)

La leucemia linfatica cronica (LLC) è una patologia neoplastica del sistema linfatico caratterizzata da un accumulo di linfociti nel sange periferico, nel midollo osseo e negli organi linfatici (linfonodi e milza). Rappresenta la forma di leucemia di più frequente osservazione nel mondo occidentale ed è tipicamente una malattia dell'adulto anziano in quanto l'età media di insorgenza è intorno ai 65 anni.

Nella maggior parte dei casi (60%) il sospetto diagnostico viene in maniera occasionale: sono pazienti in pieno benessere che presentano un aumento di globuli bianchi (leucocitosi) all'esame emocromocitometrico: questo riscontro è al giorno d'oggi sempre più frequente per la diffusa abitudine di effettuare esami routinari di controllo. Nei restanti casi la malattia viene diagnosticata per la comparsa di un aumento volumetrico di uno o più linfonodi delle stazioni linfoghiandolari superficiali (collo, ascelle, inguine).

Circa nei 2/3 dei casi la diagnosi di leucemia linfatica cronica è casuale e, pertanto, in questi pazienti la malattia decorre in maniera asintomatica. Questa quota di pazienti presenta una sopravvivenza molto più lunga, e pertanto negli stadi iniziali spesso non è indicato iniziare un trattamento citoriduttivo. Nei pazienti sintomatici, la più frequente modalità di presentazione è la comparsa di una adenopatia generalizzata: sono interessate preferenzialmente le stazioni linfonodali del collo, delle ascelle o dell'inguine e i linfononodi sono quasi sempre di consistenza non dura, non dolenti e senza tendenza a confluire in pacchetti. Frequenti sono anche l'aumento di dimensioni della milza (splenomegalia) e del fegato (epatomegalia).

Con il progredire della malattia possono comparire altri sintomi, che non sono caratteristici della leucemia linfatica cronica sono comuni al altre patologie linfoproliferative, e sono conseguenti all'invasione del midollo osseo da parte dei linfociti neoplastici: la stanchezza, associata al pallore cutaneo e palpitazioni sono una conseguenza dell'anemia, mentre le manifestazioni emorragiche sono secondarie alla riduzione delle piastrine.

Inoltre l'accumulo dei linfociti patologici ostacola la normale produzione da parte del midollo osseo di linfociti e granulociti neutrofili, che sono le cellule deputate a difendere l'organismi dai microrganismi patogeni; in questo modo si crea uno stato di immunodeficienza che predispone l'individuo malato all'insorgenza di infezioni.

Infine in un 5% di pazienti la malattie si può manifestare sotto forma di fenomeni autoimmuni, cioè la produzione di antigeni propri, in particolare antigeni di globuli rossi e piastrine, dando origine a patologie quali l'anemia emolitica autoimmune, la piastrinopenia autoimmune o più raramente l'associazione di entrambi (sindrome di *Fisher-Evans*).

Pertanto l'approccio clinico ai pazienti con leucemia linfatica cronica deve mirare sia alla valutazione dei segni che esprimono l'accumulo dei linfociti neoplastici, così come degli effetti correlati allo squilibrio immunologico che tale terapia determina e che sono appunto rappresentanti delle infezioni e delle complicanze autoimmuni.

Nonostante negli ultimi anni siano stati fatti enormi passi avanti nella comprensione dei meccanismi di funzionamento della cellula tumorale, sono ancora ignoti i fattori che portano all'isorgenza dei tumori. La ricerca ha mostrato che le probabilità di ammalarsi di tumore è legata principalmente a due fattori:

- fattori ambientali;
- caratteristiche genetiche dell'individuo.

Per quanto riguarda la LLC, a differenza di altri tipi di leucemia, la sua insorgenza non è influenzata dall'esposizione a radiazioni ionizzanti e ad agenti chimici.

Viceversa, diversi studi hanno dimostrato la possibilità che fattori genetici o familiari possono predisporre ad un più elevato rischio di sviluppare la malattia. Infatti, nei parenti di primo grado di pazienti affetti da LLC è stata osservata un'incidenza di casi da 2 a 7 volte maggiore rispetto a quella osservata in una popolazione normale di controllo.

Anche per la LLC, nonostante numerosi progressi compiuti negli ultimi anni, non sono ancora stati definiti con chiarezza i meccanismi responsabili della trasformazione di un linfocita normale in senso neoplastico. I linfociti sono cellule fondamentali del sistema immunitario di un individuo; essi rappresentano le cosiddette "sentinelle" che in condizioni normali sorvegliano costantemente l'organismo e sono pronti in ogni momento ad attivare la risposta immune nei confronti di agenti patogeni, siano essi microrganismi o cellule tumorali; si distinguono in B o T a seconda che la risposta immunitaria sia prevalentemente anticorpo-mediata o cellulo-mediata. Nel caso della leucemia linfatica cronica si verifica che uno di questi linfociti (nella maggior parte dei casi un linfocita B) subisce una trasformazione in senso neoplastico e da' origine ad un clone linfocitario, cioè una popolazione di cellule tutte uguali tra loro e presentano 2 caratteristiche peculiari:

1. non rispondono più agli stimoli fisiologici;
2. hanno perso la capacità di andare incontro ad apotosi, cioè la morte programmata della cellula.

In questo modo i linfociti neoplastici continuano a dividersi e ad accumularsi nel sangue periferico, nel midollo osseo, negli organismi linfatici (linfonodi e milza) e talora negli organismi extralinfatici.

4.1. I dati

I dati a nostra disposizione sono stati forniti dal Nucleo di ricerca clinica e laboratoristica in ematologia (NRCLE) del C.R.O di Aviano diretto dal Dott. Gattei V. Il database completo, su cui lo stesso Nucleo di ricerca ha pubblicato un lavoro in un numero della rivista *Blood* del 2007, era formato da un campione di 1076 pazienti affetti da LLC provenienti da uno studio multicentrico. L'attenzione è stata focalizzata sul gene IGHV3-21: il *dataset* a nostra disposizione contiene i profili genetici di 65 soggetti attraverso la misurazione di 28513 geni. Le misure di ciascun gene rappresentano la fluorescenza relativa a ciascun canale. In particolare si hanno 13 soggetti con IGHV3-21 + e 52 soggetti con IGHV3-21-.
Una caratteristica insolita dei dati a nostra disposizione è la totale assenza di valori mancanti. Nelle casistiche presentate in letteratura, una delle caratteristiche basi dei dati di *microarray* è la presenza di moltissimi valori mancanti dovuti a errori, problemi che si possono incontrare in fase di sperimentazione o di computazione ed elaborazione dell'immagine ed infine nel processo di traduzione da tinte a valori numerici. L'ipotesi è che i dati forniti fossero stati precedentemente completati attraverso una delle tecniche di imputazione proposte nel paragrafo 4.2. A questo proposito è stato considerato un *dataset* ridotto di 65 soggetti e 10000 espressioni geniche, per la difficoltà computazionale che avrebbe comportato analizzare un numero superiore di geni, ed è valutata l'influenza delle diverse tecniche di imputazione sui metodi empirici bayesiani nello studio di differenza di espressione. Quindi sono state imposte alcune osservazioni come valori mancanti attraverso la *routine R* presentata in A4.3.1, in particolare si sono eliminate il 10% delle osservazione considerando le osservazioni riportate in letteratura.

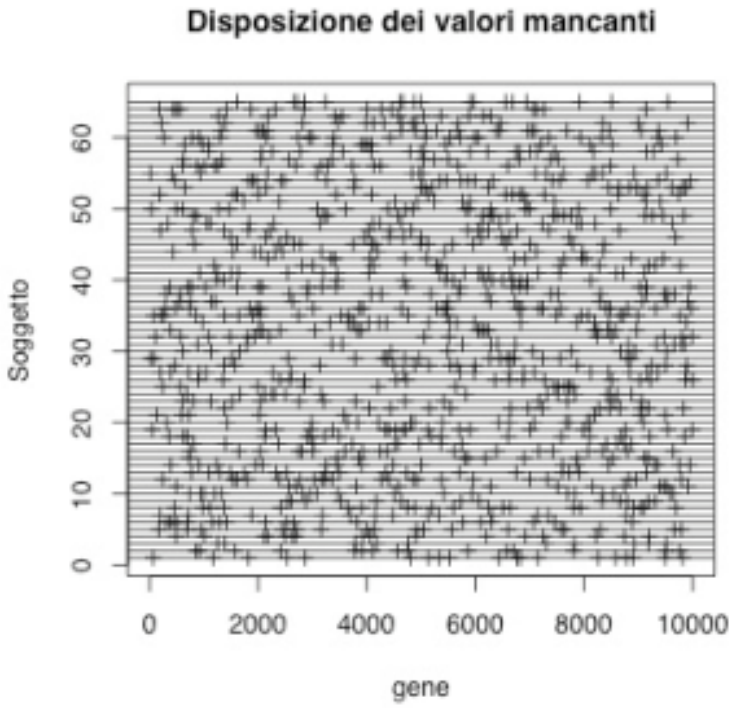


Figura 4.1 Disposizione dei valori mancanti all'interno della matrice di dati: ciascuna riga rappresenta un paziente, i segmenti verticali indicano invece il gene per il quale è stata eliminato il valore relativo all'espressione. La figura è stata realizzata attraverso la *routine R* riportata in appendice A4.

4.2. Imputazione dei valori mancanti

Il primo problema che si incontra nell'analisi dei *microarray* è l'esistenza dei valori mancanti nella matrice dei dati. Le ragioni per cui vengono a mancare le misure di un certo gene sono diverse e possono essere connesse agli strumenti utilizzati (presenza di polvere o graffi nei vetrini, …), oppure al trattamento computazionale dell'immagine e al processo di trasformazione del segnale

luminoso in dato numerico (insufficiente risoluzione, alterazione dell'immagine, …).

Lo scopo di questa fase dell'elaborato consiste nel valutare l'influenza di vari metodi di imputazione dei valori mancanti, dato che non è prevista la presenza degli stessi nei metodi empirici bayesiani. Di seguito vengono studiati quattro tecniche di imputazione, in un momento successivo verranno messi a confronto per verificare quale di essi sia il più adeguato per il *dataset* che si sta studiando.

4.2.1. Tecniche di imputazione

Le tecniche di imputazione che verranno studiate sono quattro:

- **Metodo di imputazione con la media generale:**

  Il metodo prevede la stima del valore mancante con la media dell'espressione assunta dallo stesso gene su tutti i soggetti (la media generale per l'appunto). Questa tecnica non è ottimale in quanto non tiene in considerazione la struttura di correlazione dei dati, tale stima non usa informazioni che potrebbero provenire da geni correlati con quelli mancanti e che consentirebbero una stima più accurata. La procedura di imputazione è implementata dalla funzione *sost.mean* in appendice A4.3.1.3.

- **Metodo di imputazione con la media di gruppo:**

  Il metodo stima il valore mancante con la media dell'espressione assunta dallo stesso gene solo sui soggetti che appartengono al medesimo gruppo (con la stessa variazione del gene IGHV3-21). Un problema nel quale ci si può imbattere con questa tecnica è l'espressione di geni per i quali non si è rilevata alcuna misura su un intero gruppo. In tal caso si può procedere

sostituendo il valore mancante con la media delle espressioni dello stesso gene sui restanti gruppi. La procedura di imputazione è implementata dalla funzione *sost.mg* in appendice A4.3.1.3.

- **Metodo di imputazione con la decomposizione in valori singolari (o metodo SVD):**

  Tale metodo (Tibshirani *et. al* [rif. 11]) si serve della decomposizione in valori singolari (SVD) e crea un insieme di vettori mutuamente ortogonali che vengono poi combinati linearmente per approssimare le espressioni di tutti i geni del *dataset X*. Sia dunque:

$$X_{p\times n} = U_{p\times p} D_{p\times n} V^T_{n\times n}$$

  la decomposizione in valori singolari di *X*, dove *D* è una matrice *pseudodiagonale* che riporta nella diagonale gli autovalori di *X*, mentre *U* e V^T sono matrici contenenti i corrispondenti autovettori. Si considera ora la decomposizione troncata:

$${}_J\hat{X} = {}_J U_{p\times p} \quad {}_J D_{p\times n} \quad {}_J V^T_{n\times n},$$

  dove ${}_J D_{p\times n}$ è la matrice *pseudodiagonale* contenente i primi *J=n* autovalori di *X*, e ${}_J U_{p\times p}$ e ${}_J V^T_{n\times n}$ le corrispondenti matrici di autovettori. Questa combinazione fornisce la matrice ${}_J\hat{X}$ di rango *J* che meglio approssima il minimo vincolato:

$$\min_{rk(M)=J} \|X - M\|^2$$

  dove *M* è una matrice di rango *J*.

  La soluzione può essere interpretata anche dal punto di vista della regressione dei minimi quadrati ordinari (MQO). Sia infatti *x* un qualsiasi vettore riga di *X* e si consideri la regressione MQO degli *n* elementi di *x* sui *J* vettori di ${}_J V^T_{n\times n}$, ciascuno di lunghezza *n*. Regredire *x* su ${}_J V^T_{n\times n}$ equivale a risolvere il problema di minimo

$$\min_{\beta}\left(x - {}_{J}V\beta\right)^{2}$$

che ha come soluzione $\hat{\beta} = \left({}_{J}V^{T}\,{}_{J}V\right)^{-1}{}_{J}V^{T}x$ che equivale a $\hat{\beta} = {}_{J}V^{T}x$ essendo ${}_{J}V^{T}_{n\times n}$ ortogonale. I valori previsti saranno pertanto $\hat{x} = {}_{J}V\hat{\beta}$. Generalizzando per tutte le righe di X si ha che $X\,{}_{J}V = {}_{J}U\,{}_{J}D$ fornisce tutti i coefficienti di ogni riga di X mentre $\hat{X} = {}_{J}U\,{}_{J}D\,{}_{J}V^{T}$ dà tutti i valori stimati. Così, una volta indicati i J autovalori più significativi e definita ${}_{J}V^{T}_{n\times n}$, la decomposizione in valori singolari approssima ciascuna riga di X. Si noti che la decomposizione in valori singolari può essere condotta solo su una matrice priva di valori mancanti. Pertanto si è partizionato il *dataset* X in due matrici X^{c} ed X^{m}; la prima (9188 geni per 65 osservazioni) contiene tutti e soli i geni che non presentano valori mancanti, la seconda (812 geni per 65 osservazioni) raccoglie invece i geni che possiedono almeno un valore mancante. Indicando con x^{*} un qualsiasi vettore appartenente a X^{m}, i suoi valori mancanti possono essere imputati con un procedimento di regressione simile che sfrutta la decomposizione in valori singolari di X^{c}. Una volta individuati i J autovalori più significativi, si stima il dato mancante k nel gene i, prima regredendo questo gene sui J autovalori più significativi, e poi usando il coefficiente di regressione per ricostruire k da una combinazione lineare dei J autovettori. Naturalmente questa operazione prevede che il k-esimo valore del geni i e gli elementi k-esimi dei J autovettori non vengano utilizzati nei calcoli per la stima del coefficiente di regressione che servirà per la stima del k-esimo valore mancante. In particolare si cercherà il valore del parametro β che risolve il problema del minimo vincolato

$$\min_{\beta}\sum_{l\ nonmancanti}\left(x_{i}^{*} - \sum_{j=1}^{J} v_{ij}\beta_{j}\right)$$

Dopo aver applicato alla matrice X^c la decomposizione in valori singolari, si definisce ${}_JV^{*T}$ la matrice ottenuta rimuovendo da ${}_JV^T$ le colonne che corrispondo agli elementi mancanti di x^* e ${}_JV^{(*)T}$ la matrice ad essa complementare. La soluzione del problema di minimo sarà pertanto $\hat{\beta}={}_JV^{*T}x^*$, ed i valori potranno essere stimati con ${}_JV^{(*)T}\hat{\beta}$. Questa procedura è stata implementa dalle *routine* in appendice A4.3.1.3.

- **Metodo di imputazione *k-nearest neighbors* (metodo *knn*):**
  Tale metodo prevede l'utilizzo, nella stima dei dati mancanti, dei geni più somiglianti a quello che presenta valore mancante. Ideata da Tibshirani *et al.* [rif. 11], questa tecnica prevede ancora la partizione del *dataset* in X^c ed X^m. Indicato x^* un qualsiasi valore appartenente alla matrice X^m, l'algoritmo consiste nel calcolare la distanza tra x^* e tutti i geni contenuti nella matrice X^c usando solo le coordinate dei valori di x^* che non risultano mancanti. Successivamente il dato da imputare viene calcolato come una media pesata dei k geni che risultano più prossimi (o "vicini") ad x^*. In questa media, il contributo di ciascun gene viene pesato sulla base della somiglianza della sua espressione con quella del gene x^*. In particolare si sono usati pesi inversamente proporzionali alla distanza e tali da sommare a uno. La matrice che viene suggerita per il calcolo della distanza è quella Euclidea che risulta sufficientemente accurata nonostante sia sensibile agli *outliers* che possono presentarsi nei dati da *microarray*. Questa procedura è stata implementa dalle *routine* in appendice A4.3.1.3.

Per quanto concerne la scelta dei parametri J e k che regolano il numero di autovalori nel caso dell' imputazione SVD ed il numero dei geni "vicini" da considerare nel caso del calcolo della media pesata nel caso *knn*, il *dataset* originale è stato suddiviso in 5 sottoinsiemi ognuno con 2000 espressioni geniche. In ogni *subset* sono stati imputati il 10% di valori

mancanti, come proposto in precedenza e, per diversi valori di J e di k sono stati applicati i due algoritmi di imputazione sotto osservazione, ed è stato calcolato un indice di bontà di imputazione (RMS error, Root Mean Squared error) definito come la media dei quadrati tra il vero valore e il valore stimato, tutto sotto radice:

$$RMS\ error = \sqrt{\frac{1}{m}\sum_{i=1}^{m}\left(x_i - \hat{x}_i\right)^2},$$

dove m è il numero di valori mancanti, x_i e $\hat{x}_i$ sono rispettivamente il vero valore stimato del dato mancante.

Questa operazione ha lo scopo di individuare i valori J e k che minimizzano l' RMS. Le figure 4.2 e 4.3 riportano l' RMS error in funzione rispettivamente di J e di k; la linea rappresenta la media dell' RMS error sui cinque campioni per i quali si è applicato l'algoritmo.

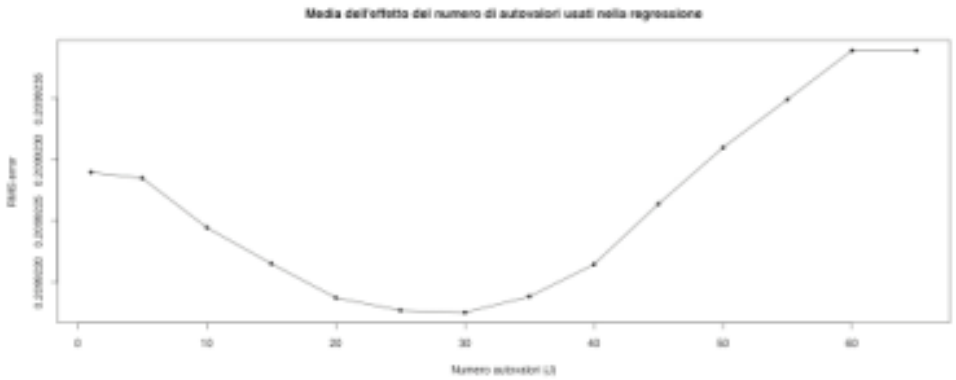


Figura 4.2 Metodo SVD: effetto del parametro J che regola il numero di autovettori usati per la stima del coefficiente di regressione β sull' RMS error.

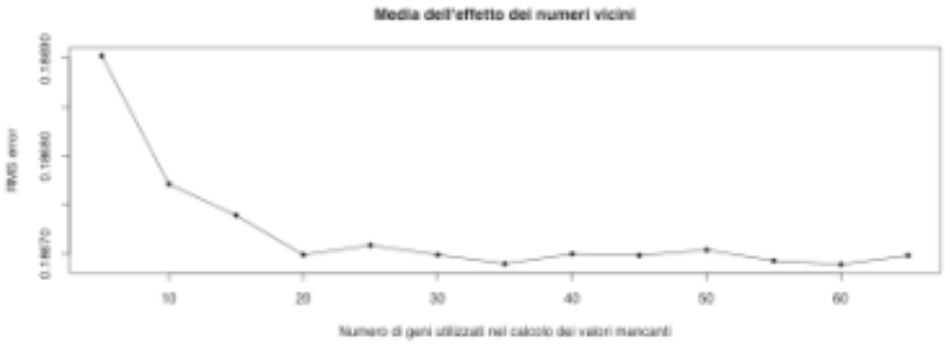


Figura 4.3 Metodo *knn*: andamento dell' RMS error in funzione di *k* che regola il numero di geni su cui si calcola la media pesata per l'imputazione dei dati mancanti.

Per quanto riguarda la scelta di *J*, il grafico della figura 4.2 mette in luce che RMS error viene minimizzato con *J* pari a 30, si è deciso quindi di adottare *J=30* che corrisponde alla scelta di usare all'incirca il 40% degli autovettori.

L'RMS error in funzione di *k* ha un minimo per *k* compreso tra 10 e 15 all'incirca. Pertanto è parso sensato adottare *k*=12.

Il grafico in Figura 4.3 mette in evidenza come la prestazione dell'algoritmo declini quando si usano pochi valori di stima, e ciò è dovuto all'eccessiva enfasi di alcuni modelli di espressione. Il peggioramento delle *performance* per valori crescenti di *k* può essere spiegata in ragione a due motivi. Innanzitutto l'inclusione di modelli di espressione significativamente diversi (o lontani) dal gene di interesse può far perdere in accuratezza in quanto il gruppo di geni su cui si calcola la media pesata è troppo grande e non rilevante per la stima; secondariamente è importante tener presente che nei dati derivati da *microarray* può esserci molto

rumore, e con l'aumentare di *k* il contributo del disturbo alla stima supera il contributo del segnale, causando una diminuzione di accuratezza.

Nella figura 4.4 si riportano infine gli errori RMS relativi ai quattro metodi di imputazione ciascuno dei quali è stato applicato a cinque diversi campioni.

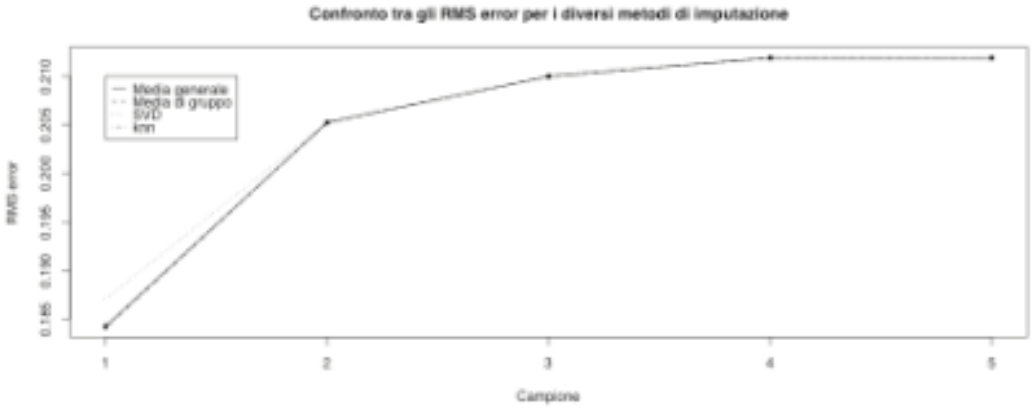


Figura 4.4 Confronto tra le *performance* dei quattro metodi d'imputazione che sono stati applicati sugli stessi cinque campioni. Ciascuna curva rappresenta un singolo metodo. Per gli algoritmi *knn* e SVD, i parametri *J* e *k* sono stati fissati rispettivamente 17 e 12.

La figura 4.4 evidenzia la superiore accuratezza dei metodi *knn* e SVD mentre stupisce la scarsa differenza di *performance* tra il metodo della media generale e il metodo della media di gruppo.

4.2.2.Efficacia delle tecniche di imputazione

Nel paragrafo precedente, sono state presentate le diverse tecniche di imputazione. Nel seguito, l'efficacia di tali tecniche è valutata con riferimento ai metodi empirici bayesiani per il controllo della differenza di espressione dei geni.

L'obiettivo è quello di verificare se i quattro metodi di imputazione possono avere un impatto sull'identificazione di geni differenzialmente espressi.

L'analisi preliminare consiste di valutare la possibilità dell'uso dei modelli Gamma-Gamma e LogNormale-Normale descritti nel capitolo 2 attraverso l'analisi visiva di alcuni grafici che si possono ottenere con i metodi *checkCCV* e *checkModel* della libreria *EBarrays*. Per valutare l'assunzione della costanza del coefficiente di variazione per i diversi metodi di imputazione, sono stati costruiti i grafici in Figura 4.5. La figura mostra una forte variabilità del CV indipendentemente dall'algoritmo utilizzato: esso sembra dipendere fortemente dalla media.

Indipendentemente dalla tecnica di imputazione utilizzata, l'ipotesi di costanza del coefficiente di variazione non è verificata: i grafici in Figura 4.5 evidenziano una significativa dipendenza del coefficiente di variazione dalla media.

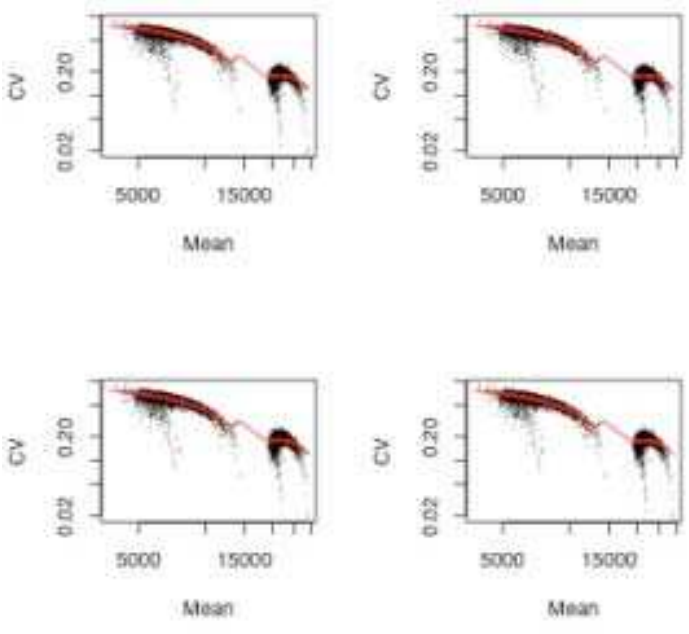


Figura 4.5 Coefficiente di variazione in funzione della media per i dati di espressione genica di soggetti leucemici avendo imputato i dati mancanti attraverso i 4 algoritmi di imputazione. Rispettivamente, in alto a sinistra i dati sono imputati con la media generale, in alto a destra con la media di gruppo, in basso a sinistra con l'algoritmo SVD ed infine, in basso a destra, con il metodo *knn*.

Per valutare l'impatto dei metodi d'imputazione dei dati mancanti nei metodi empirici bayesiani, si considera il problema del confronto tra i due gruppi in studio valutando il seguente sistema d'ipotesi:

$$H_0 : \mu_{IGHV\,3-21+} = \mu_{IGHV\,3-21-} \Rightarrow Espressione\,genica\,equivalente$$
$$H_1 : \mu_{IGHV\,3-21+} \neq \mu_{IGHV\,3-21-} \Rightarrow Espressione\,genica\,differenziale$$

dove con μ_g indichiamo la media di espressione per i dati appartenenti al gruppo g (g= {IGVH3-21+, IGVH3-21-}).
Mediante la funzione *emfit* di *EBarrays* sono stati stimati i parametri dei modelli Gamma-Gamma e LogNormale-Normale sui quattro *dataset* che corrispondono alle quattro tecniche di imputazione dei dati mancanti.

Metodo di imputazione	Media generale	Media di gruppo	Metodo SVD	Metodo *knn*
$\hat{\alpha}$	3.408	3.406	3.406	3.407
$\hat{\alpha}_0$	3.185	3.180	3.180	3.185
$\hat{v}$	9999.99	9999.99	999.99	9999.99
$\hat{p}_1$	0.999	0.999	0.999	0.999

Tabella 3.8. Stime dei parametri del modello GG sui dati simulati dal modello GG con (α, α_0, υ) = (10, 0.9, 0.5) al variare della tecnica.

Metodo di imputazione	Media generale	Media di gruppo	Metodo SVD	Metodo *knn*
$\hat{\mu}_0$	9.272	9.273	9.273	9.272
$\hat{\sigma}$	0.490	0.490	0.490	0.490
$\hat{\tau}$	0.518	0.517	0.517	0.518
$\hat{p}$	0.972	0.972	0.972	0.972

Tabella 3.9. Stime dei parametri del modello GG sui dati simulati dal modello GG con (μ_0, σ, τ) = (2.3, 0.3, 1.39) al variare della tecnica di imputazione.

La tecnica permette di calcolare le probabilità a posteriori (forniti i dati di espressione genica con le quattro tecniche di imputazione dei valori mancanti) dei geni di essere differenzialmente espressi. Considerando differenzialmente espressi i geni con probabilità a posteriori maggiore di 0.5, si sono confrontati i risultati dei due modelli nei quattro casi:

	Espressione prevista da LNN	
Espressione prevista da GG	Equivalente	Differente
Equivalente	9854	139
Differente	0	7

(a)

	Espressione prevista da LNN	
Espressione prevista da GG	Equivalente	Differente
Equivalente	9852	141
Differente	0	7

(b)

	Espressione prevista da LNN	
Espressione prevista da GG	Equivalente	Differente
Equivalente	9848	146
Differente	0	6

(c)

	Espressione prevista da LNN	
Espressione prevista da GG	Equivalente	Differente
Equivalente	9854	140
Differente	0	6

(d)

Tabella 4.1 Confronto circa l'espressione genica secondo i modelli GG e LNN in cui si testano le diverse tecniche di imputazione dei dati mancanti: (a) con la media generale, (b) con la media di gruppo, (c) con l'algoritmo SVD e (d) con il metodo *knn*.

Dal confronto delle tecniche attraverso la Tabella 4.1, è opportuno evidenziare che attraverso la modellazione con GG sono stati identificati differenzialmente espressi 6 geni su 10000, pari allo 0.06%, quando sono stati utilizzati gli algoritmi SVD e *knn*; mentre attraverso la sostituzione dei valori mancanti con la media generale o la media di gruppo ne sono stati identificati 7 (0.07%). Le differenze più significative per i diversi metodi di imputazioni, si evidenziano con la modellazione attraverso LNN, infatti, i geni differenzialmente espressi sono 146 (1.46%) per il metodo della media generale e per l'algoritmo *knn*, 148 (1.48%) per il metodo della media di gruppo e 152 (1.52%) per l'algoritmo SVD. Si può quindi teorizzare una diversa *performance* dell'algoritmo SVD in quanto un maggior numero di geni, rispetto agli altri tre metodi di imputazione, viene identificati come differenzialmente espressi solo dal modello LNN e non dal modello GG.

Per valutare ulteriormente le tecniche di imputazione, sono stati applicati i metodi empirici bayesiani applicati ai dati imputati e ai dati reali (quelli fornitici da C.R.O. di Aviano). Sono stati costruiti 5 *training set* costituito di 100 geni e 65 osservazioni, 4 relativi ai dati imputati e 1 relativo ai dati reali; ed i corrispettivi *test set* di 9900 geni e 65 osservazioni su cui testare i modelli .

Sono state costruite le tabelle 4.2 e 2.3 che riportano matrici di confusione che confrontano le identificazioni fatte sui dati reali e sui dati imputati modellati con GG e LNN.

	Espressione prevista dal modello GG sui dati i cui valori mancanti sono stati sostituiti dalla media di gruppo.	
Espressione prevista dal modello GG sui dati reali	Equivalente	Differente
Equivalente	9900	0
Differente	0	0

(a)

	Espressione prevista dal modello GG sui dati i cui valori mancanti sono stati sostituiti dalla media generale.	
Espressione prevista dal modello GG sui dati reali	Equivalente	Differente
Equivalente	9900	0
Differente	0	0

(b)

	Espressione prevista dal modello GG sui dati i cui valori mancanti sono stati sostituiti dall'algoritmo SVD.	
Espressione prevista dal modello GG sui dati reali	Equivalente	Differente
Equivalente	9900	0
Differente	0	0

(c)

	Espressione prevista dal modello GG sui dati i cui valori mancanti sono stati sostituiti dall'algoritmo *knn*.	
Espressione prevista dal modello GG sui dati reali	Equivalente	Differente
Equivalente	9900	0
Differente	0	0

(d)

Tabella 4.2. Matrici di confusione sui dati modellati con GG utilizzando (a) l'imputazione della media generale, (b) l'imputazione della media di gruppo, (c) l'utilizzo dell'algoritmo SVD e (d) dell'algoritmo *knn*.

	Espressione prevista dal modello LNN sui dati i cui valori mancanti sono stati sostituiti dalla media generale.	
Espressione prevista dal modello LNN sui dati reali	Equivalente	Differente
Equivalente	9783	1
Differente	6	110

(a)

	Espressione prevista dal modello LNN sui dati i cui valori mancanti sono stati sostituiti dalla media di gruppo.	
Espressione prevista dal modello LNN sui dati reali	Equivalente	Differente
Equivalente	9782	2
Differente	4	112

(b)

	Espressione prevista dal modello LNN sui dati i cui valori mancanti sono stati sostituiti dall'algoritmo SVD.	
Espressione prevista dal modello LNN sui dati reali	Equivalente	Differente
Equivalente	9781	3
Differente	9	107

(c)

	Espressione prevista dal modello LNN sui dati i cui valori mancanti sono stati sostituiti dall'algoritmo *knn*.	
Espressione prevista dal modello LNN sui dati reali	Equivalente	Differente
Equivalente	9873	0
Differente	0	116

(d)

Tabella 4.3 Matrici di confusione sui dati modellati con LNN utilizzando (a) l'imputazione della media generale, (b) l'imputazione della media di gruppo, (c) l'utilizzo dell'algoritmo SVD e (d) dell'algoritmo *knn*.

Come parametro di confronto delle *performance* dei 4 metodi d'imputazione, è sembrato opportuno utilizzare l'errore di identificazione ossia, è importante valutare quanti geni sono stati identificati come differenzialmente espressi nei dati imputati quando nei dati reali sono stati identificati di equivalente espressione; viceversa, di fondamentale importanza è valutare la frazione di geni che nella casistica reale sono stati identificati come differenzialmente espressi ma non ugualmente nei dati imputati.

Considerando la modellazione con GG, le matrici di confusione in Tabella 4.1 mostrano una totale assenza di geni identificati come differenzialmente espressi sia nei dati reali, che nei dati imputati con i 4 algoritmi. Per il confronto delle *performance* degli algoritmi d'imputazione dei dati mancanti, questa situazione non porta alcuna informazione. Si passa quindi al confronto delle matrici di confusione sui dati modellati con LNN riportate in Tabella 4.2. Come prima osservazione è utile sottolineare la totale assenza di errore d'identificazione quando i dati mancanti vengono imputati attraverso l'algoritmo *knn.* Utilizzando gli altri metodi d'imputazione dei dati mancanti sono stati rilevati alcuni errori: la frazione di geni che nella casistica reale sono stati identificati come differenzialmente espressi ma non ugualmente nei dati imputati è pari a 6 su 116 nel caso si utilizzi la media generale, tale frazione diminuisce a 4 su 116 con l'utilizzo della media di gruppo, mentre peggiora, con 9 casi su 116 nel caso si utilizzi l'algoritmo SVD. Viceversa, i geni che nella casistica reale vengono identificati come equivalenemente espressi e non ugualmente nei dati imputati, sono 3 su 9783 nel caso dell'algoritmo SVD, che si conferma il metodo di imputazione di dati mancanti con maggiore impatto, diminuisce a 2 su 9783 con l'utilizzo della media di gruppo, e nonostante ciò che ci si aspetta, diminuisce ancora a 1 su 9783 con l'utilizzo della media generale. E' utile comunque sottolineare che i 4 metodi di imputazione non hanno *performance* di molto differenti.

In appendice A4.1 vengono costruite le tabelle che indicano le posizioni dei geni identificati come differenzialmente espressi sia sui dati reali che sui dati imputati, dei geni identificati come equivalentemente espressi sui dati reali e differenzialmente sui dati imputati e l'elenco dei geni identificati come differenzialmente espressi sui dati reali mentre equivalentemente espressi sui dati imputati. In appendice A4.2 vengono riportati i grafici relativi alla modellazione con GG e LNN per l'identificazione di geni differenzialmente espressi. Infine, in appendice A4.3 vengono riportate le *routine R* con cui si sono condotte le analisi.

CAPITOLO 4

Appendice 4

A4.1 I geni

Metodo di imputazione con la media generale		
Geni identificati DE sia nei dati reali che nei dati imputati	Geni identificati DE nei dati reali e EE nei dati imputati	Geni identificati EE nei dati reali e DE nei dati imputati
10, 34, 187, 219, 282, 330, 349, 380, 488, 528, 559, 585, 609, 653, 672, 720, 725, 796, 847, 850 , 865, 871, 873, 1032, 1180, 1296, 1368, 1519, 1577, 1601, 1640, 1648, 1742, 1786, 1848 1849, 1877, 2013, 2110, 2116, 2144, 2194, 2260, 2315, 2329, 2371, 2392, 2436, 2494, 2552, 2598, 2677, 2683, 2694, 2696, 2775, 2778, 2838, 2904, 2928, 2969, 3016, 3061, 3093, 3197, 3287, 3299, 3340, 3344, 3355, 3361, 3362, 3410, 3415, 3436, 3461, 3493, 3531, 3573, 3642, 3665, 3688, 3774, 3823, 3871, 4047, 4072, 4104, 4116, 4120, 4179, 4187, 4358, 4376, 4427, 4513, 4520, 4758, 4807, 4826, 4842, 4890.	825.	233, 304, 1677, 1802, 1805, 3990.

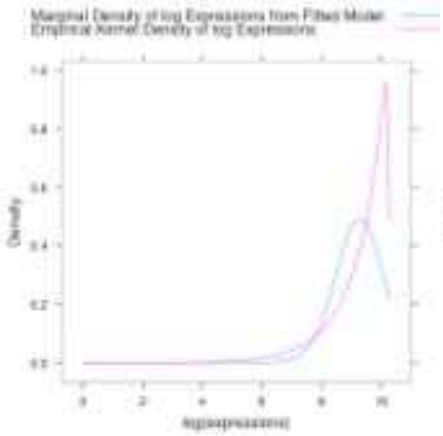
Metodo di imputazione con la media di gruppo		
Geni identificati DE sia nei dati reali che nei dati imputati	Geni identificati DE nei dati reali e EE nei dati imputati	Geni identificati EE nei dati reali e DE nei dati imputati
10, 34, 187, 219, 282, 302, 304, 330, 349, 380, 559, 585, 609, 653, 672, 720, 725, 796, 847, 850, 865, 871, 873, 930, 1022, 1032, 1180, 1296, 1368, 1519, 1577, 1601, 1640, 1648, 1684, 1742, 1763, 1786, 1848, 1849, 1877, 2013, 2110, 2116, 2144, 2194, 2260, 2315, 2329, 2371, 2384, 2392, 2436, 2494, 2552, 2598, 2677, 2683, 2689, 2694, 2696, 2775, 2778, 2838, 2904, 2928, 2969, 3016, 3061, 3093, 3197, 3287, 3299, 3340, 3344, 3355, 3361, 3362, 3410, 3415, 3436, 3461, 3493, 3531, 3573, 3642, 3665, 3688, 3774, 3823, 3871, 3990, 4047, 4072, 4104, 4116, 4120, 4179, 4187, 4358, 4427, 4513, 4520, 4758, 4807, 4826, 4842, 4890.	825, 4080.	233, 1677, 1802, 1805.

Metodo di imputazione con l'algoritmo SVD.		
Geni identificati DE sia nei dati reali che nei dati imputati	Geni identificati DE nei dati reali e EE nei dati imputati	Geni identificati EE nei dati reali e DE nei dati imputati
10, 34, 219, 282, 302, 330, 349, 380, 488, 528, 559, 585, 609, 653, 672, 720, 725, 796, 850 , 865, 871, 873, 930, 1022, 1032, 1180, 1296, 1368, 1515, 1519, 1577, 1601, 1640, 1648, 1742, 1763, 1786, 1848 1849, 1877, 2013, 2110, 2116, 2144, 2194, 2260, 2329, 2371, 2384, 2392, 2436, 2494, 2552, 2598, 2677, 2683, 2689, 2694, 2696, 2775, 2778, 2838, 2904, 2928, 2969, 3016, 3061, 3093, 3197, 3287, 3299, 3340, 3344, 3355, 3361, 3362, 3410, 3415, 3436, 3461, 3493, 3531, 3573, 3642, 3665, 3688, 3774, 3823, 3871, 3990, 4047, 4104, 4116, 4120, 4179, 4187, 4358, 4376, 4427, 4513, 4520, 4758, 4807, 4826, 4842, 4890.	730, 1550, 4080	187, 233, 304, 847, 1677, 1802, 1805, 2315, 4072.

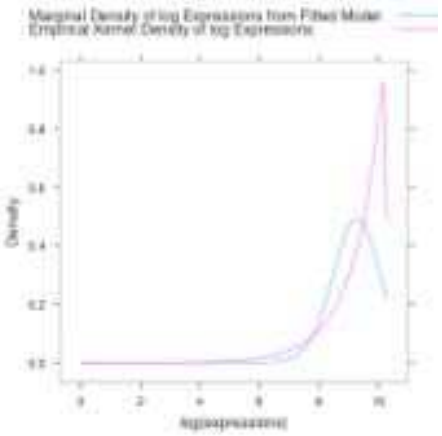
Metodo di imputazione con l'algoritmo *knn*.		
Geni identificati DE sia nei dati reali che nei dati imputati	Geni identificati DE nei dati reali e EE nei dati imputati	Geni identificati EE nei dati reali e DE nei dati imputati
10, 34, 187, 219, 233, 282, 304, 330, 349, 380, 488, 528, 559, 585, 609, 653, 672, 720, 725, 796, 847, 850 , 865, 871, 873, 1032, 1180, 1296, 1368, 1519, 1577, 1601, 1640, 1648, 1677, 1742, 1786, 1802, 18,05, 1848 1849, 1877, 2013, 2110, 2116, 2144, 2194, 2260, 2315, 2329, 2371, 2392, 2436, 2494, 2552, 2598, 2677, 2683, 2694, 2696, 2775, 2778, 2838, 2904, 2928, 2969, 3016, 3061, 3093, 3197, 3287, 3299, 3340, 3344, 3355, 3361, 3362, 3410, 3415, 3436, 3461, 3493, 3531, 3573, 3642, 3665, 3688, 3774, 3823, 3871, 3990, 4047, 4072, 4104, 4116, 4120, 4179, 4187, 4358, 4376, 4427, 4513, 4520, 4758, 4807, 4826, 4842, 4890.		

A4.2 Grafici delle analisi con i metodi bayesiani empirici sui dati imputati.

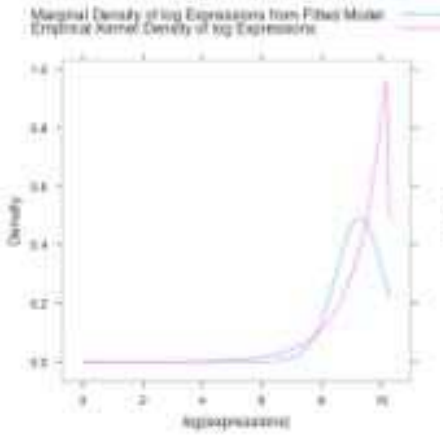
A42.2.1 Distribuzione marginale del modello GG sovrapposta alla densità di dati di espressione genica, calcolata con il metodo del nucleo.



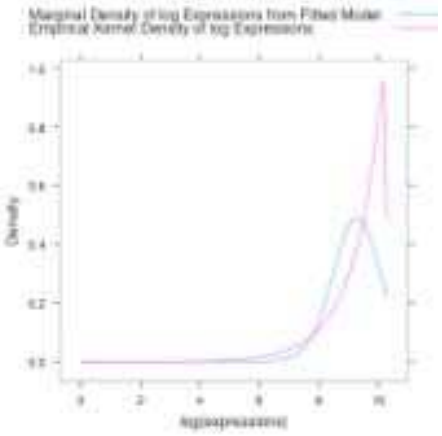
(a) imputazione con la media generale.



(b) imputazione con la media di gruppo.

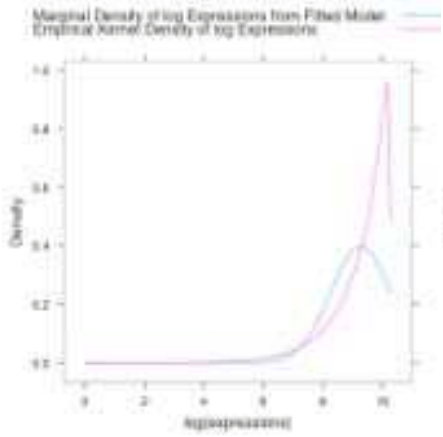


(c) imputazione con l'algoritmo SVD.

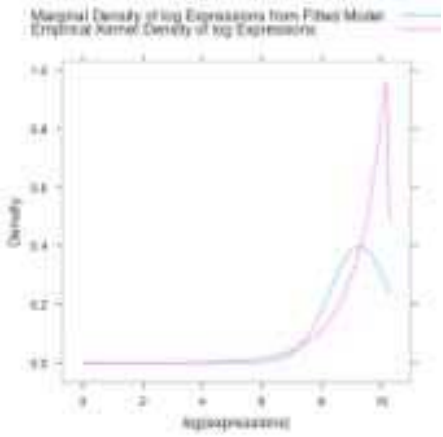


(d) imputazione con l'algoritmo *knn*.

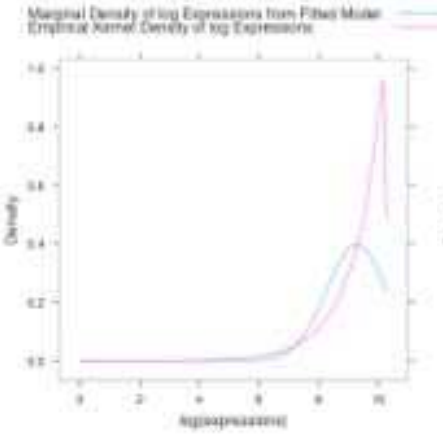
A42.2.2 Distribuzione marginale del modello LNN sovrapposta alla densità di dati di espressione genica, calcolata con il metodo del nucleo.



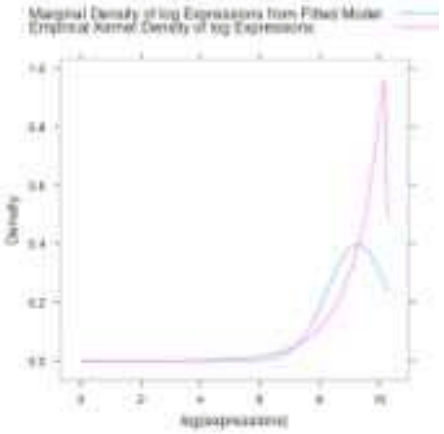
(a) imputazione con la media generale.



(b) imputazione con la media di gruppo.

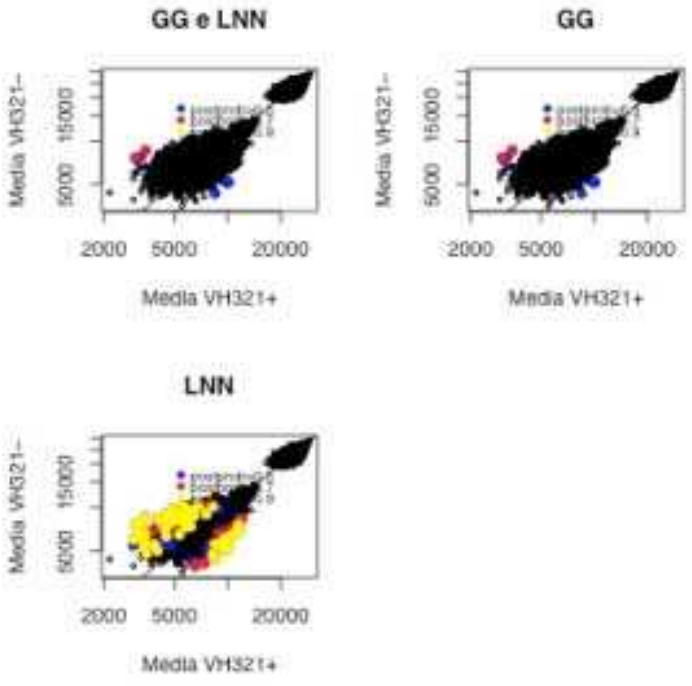


(c) imputazione con l'algoritmo SVD.

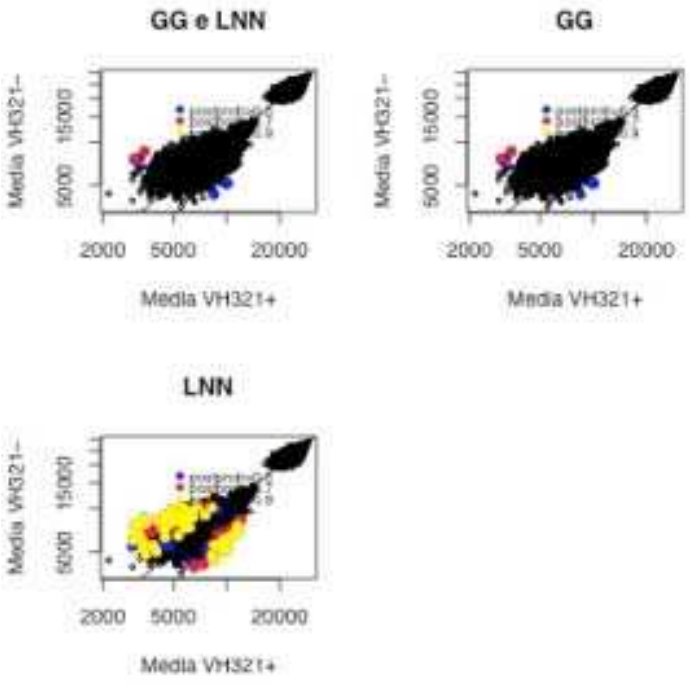


(d) imputazione con l'algoritmo *knn*.

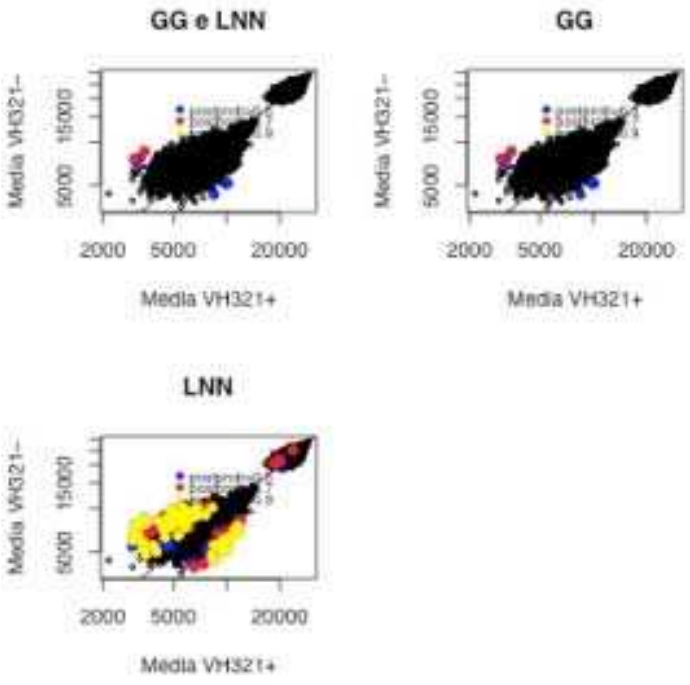
A42.2.3 Diagrammi di dispersione del rapporto tra 2 tinte: sull'asse delle ascisse la media di espressione dei geni negli individui appartenenti al gruppo VH321+, sulle ordinate quella relativa al gruppo VH321-. Con colorazione diversa sono colorati i geni con probabilità a posteriori superiore a 0.5 di essere equivalentemente espressi secondo i vari livelli. In alto a sinistra i geni indicati da entrambe i modelli, negli altri 2 i grafici dei geni indicati rispettivamente da GG e LNN.



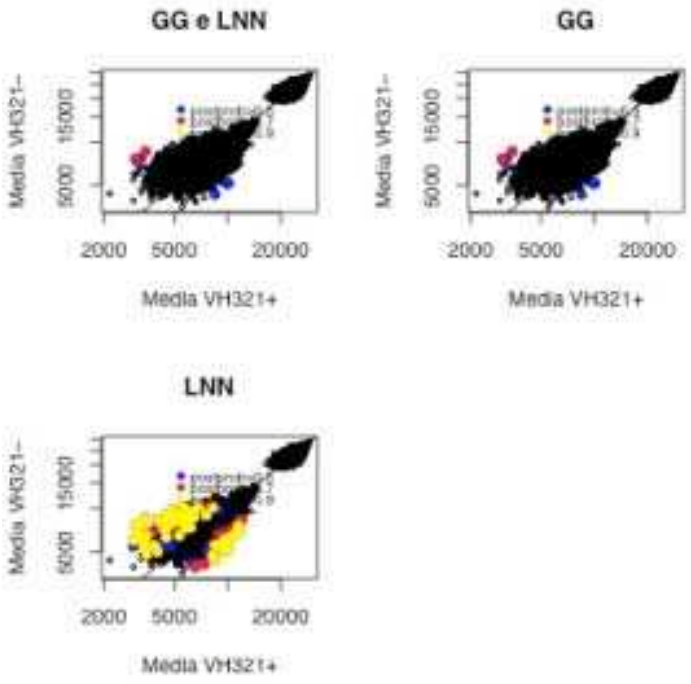
(a) dati imputati dalla media generale



(b) dati imputati dalla media di gruppo

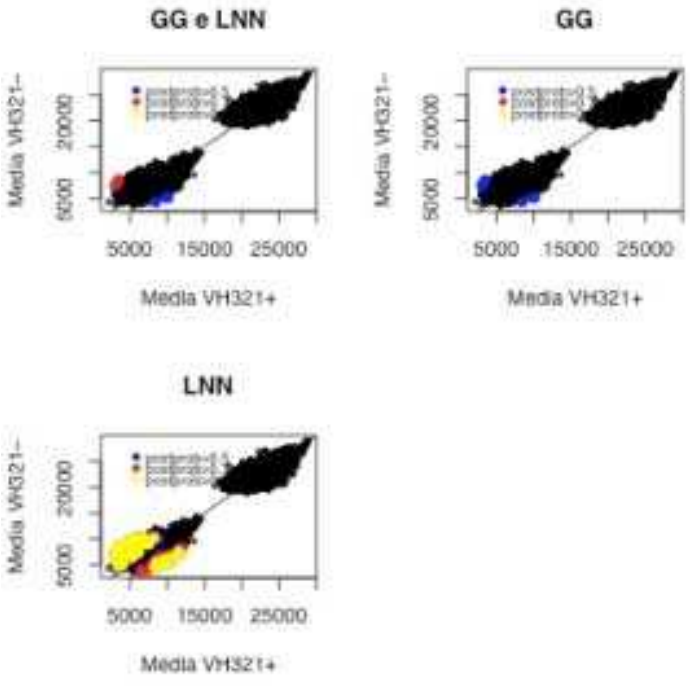


(c) dati imputati con l'algoritmo SVD

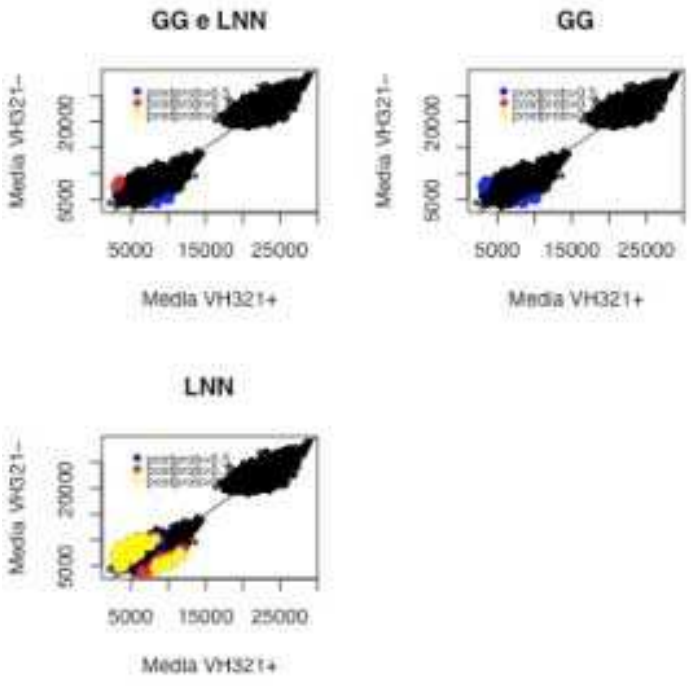


(d) dati imputati con l'algoritmo *knn*.

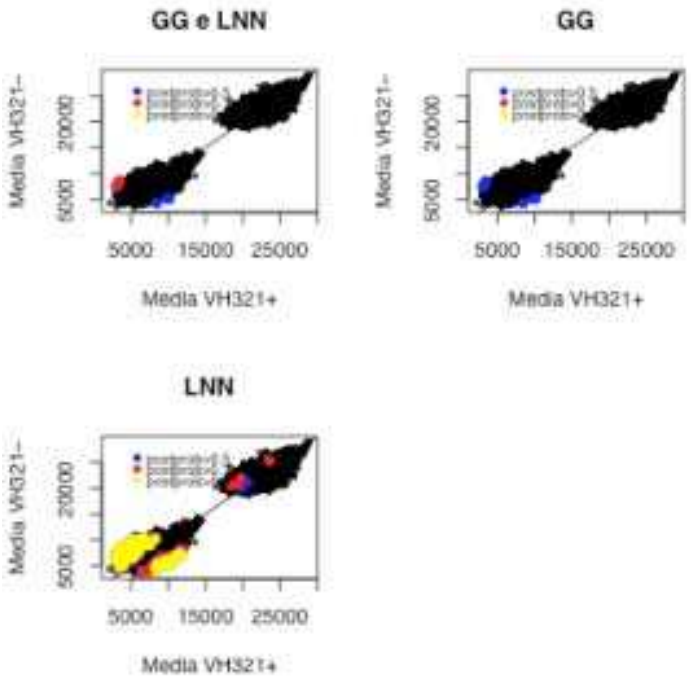
A42.2.4 Diagrammi di dispersione del rapporto tra 2 tinte: sull'asse delle ascisse la media di espressione dei geni negli individui appartenenti al gruppo VH321+, sulle ordinate quella relativa al gruppo VH321-. Con colorazione diversa sono colorati i geni con probabilità a posteriori superiore a 0.5 di essere differenzialmente espressi secondo i vari livelli. In alto a sinistra i geni indicati da entrambe i modelli, negli altri 2 i grafici dei geni indicati rispettivamente da GG e LNN.



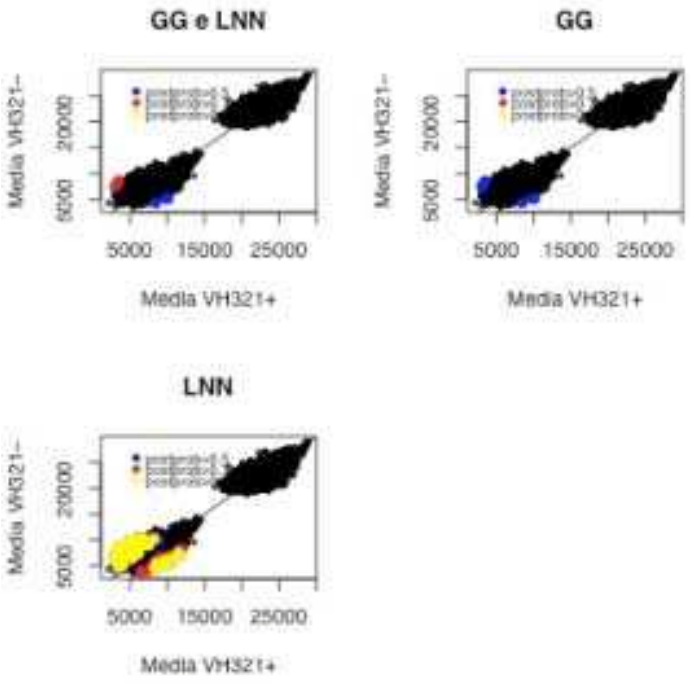
(a) dati imputati dalla media generale



(b) dati imputati dalla media di gruppo



(c) dati imputati con l'algoritmo SVD



(d) dati imputati con l'algoritmo *knn*.

A4.3 Codice *R* relativo alle anelisi effettuate:

A4.3.1 I valori mancanti

A4.3.1.1 *Routine R* per l'imputazione di valori mancanti in maniera casuale utilizzando la funzione *sample* di *R*.

```
ngeni<-10000
ncasi<-65
dati<-as.vector(data)
id<-sample(1:ngeni*ncasi,1000,replace=FALSE)
dati[id]<-NA
dati<-matrix(dati,ngeni,ncasi)
```

A4.3.1.2 Costuzione del grafico che plotta i valori mancanti in Figura 4.1.

```
plot. a<-function(data)
      {
      x<-0
      plot(1,1,ylim=c(1,nrow(data)),xlim=c(1,ncol(data)),type="n",
      main="Disposizione dei valori
      mancanti",xlab="gene",ylab="Soggetto")
      for(i in 1:nrow(data)){
            y<-0
            x<-x+1
            abline(h=x)
            for(j in 1:ncol(data)){
            y<-y+1
            if(is.na(data[i,j])){
            points(y,x,pch=3)
            }
      #else{
```

```
            #points(y,x,col=2,pch=3)}
            }
      }
}
```

A4.3.1.3 Funzioni per la costruzione dei 4 *dataset* attraverso i metodi di imputazione dei valori mancanti studiati.

```
#sostituzione dei valori mancanti con la media generale
sost.mean<-function(data)
      {
      dati<-data
      m<-apply(data,1,mean,na.rm=T)
      for(j in 1:ncol(data)){
            for(i in 1:nrow(data)){
                  if(is.na(data[i,j]))
                  dati[i,j]<-m[i]
                  }
            }
dati
}

data.mean<-sost.mean(dati)


#sostituzione dei valori mancanti con la media di gruppo
sost.mg<-function(data)
      {
      dati<-data
      Ana<-data[,1:13]
      Bna<-data[,14:65]
      medA<-apply(Ana,1,mean,na.rm=T)
      mA<-matrix(medA,nrow=1)
      medB<-apply(Bna,1,mean,na.rm=T)
```

```
      mB<-matrix(medB,nrow=1)
      for(j in 1:13){
            for(i in 1:nrow(data)){
                  if(is.na(data[i,j]))
                  dati[i,j]<-mA[i]
                  }
            }
            for(j in 14:65){
                  for(i in 1:nrow(data)){
                        if(is.na(data[i,j]))
                              dati[i,j]<-mB[i]
                              }
                        }
      dati
}

data.mg<-sost.mg(dati)


#crea la matrice Xc
togli<-function(data,mat){
      x<-rep(0,65)
      for(i in 1:nrow(mat)){
            if(mat[i,1]==0){
                  x<-rbind(x,data[i,])
            }
      }
x<-x[-1,]
x
}

#crea la matrice Xm
restante<-function(data,mat)
      {
      x<-rep(0,65)
      for(i in 1:nrow(mat)){
```

```
            if(mat[i,1]!=0){
                  x<-rbind(x,data[i,])
            }
      }
x<-x[-1,]
x
}
```

```
#l'otput di contag è mat
contag<-function(data)
      {
      count<-rep(0,nrow(data))
      for(i in 1:nrow(data)){
            for(j in 1:ncol(data)){
                  if(is.na(data[i,j])){
                        count[i]<-count[i]+1}
                  }
            }
      count
}

#Algoritmo SVD
svd.imput<-function(dat,xm,xmc,mat,j) #dat sarebbe xc
      {
      mat<-matrix(mat,ncol=1)
      data<-as.matrix(dat)
      y<-1
      x<-0
      el<-NULL
      for(i in 1:j){
            x<-x+1
            el[i]<-ncol(dat)-j+x
```

```
            }
      dat<-as.matrix(dat)
      dati<-as.matrix(dat)
      dati. svd<-svd(dati)
      VT<-t(as.matrix(dati.svd$v))
      V<-as.matrix(dati.svd$v)
      for(h in 1:nrow(xm)){
      missing<-NULL
      m<-0
      ind<-NULL
      l<-0
      for(k in 1:ncol(xm)){
            m<-m+1
            if(is.na(xm[h,k])){
                  l<-l+1
                  ind[l]<-m
            }
      }
      x.min<-matrix(xm[h,-ind],ncol=1)
      VT.min<-VT[-el,-ind]
      VT.compl<-VT[-el,ind]
      beta<-(VT.min%*%x.min)
      missing<-t(VT.compl)%*%beta
      f<-0
      for(z in 1:ncol(xm)){
            if(is.na(xm[h,z])){
                  f<-f+1
                  xm[h,z]<-missing[f]
            }
      }
}
da.def<-riordina(xm,dat,mat)
da.def
}

riordina<-function(da,xc,mat)
```

```
      {
      x<-rep(0,ncol(xc))
      j<-0
      k<-0
      for(i in 1:nrow(mat)){
            if(mat[i,]==0){
            j<-j+1
            x<-rbind(x,xc[j,])
            }
      if(mat[i,]!=0){
            k<-k+1
            x<-rbind(x,da[k,])
            }
      }
x<-x[-1,]
rownames(x)<-c(1:nrow(mat))
x
}

m<-contag(dati)
m<-matrix(m,ncol=1)
xc<-togli(dati,m)
xm<-restante(dati,m)
xmc<-rbind(xc,xm)

data.svd<-svd.imput(xc,xm,xmc,m,30)
```

A4.3.1.4 *Routine* R per la costruzione dei grafici in Figura 4.5.

```
library(EBarrays)
par(mfrow=c(2,2))
checkCCV(data.mean)
checkCCV(data.mg)
```

```
checkCCV(data.SVD)
checkCCV(data.knn)
```

A4.3.1.5 *Routine* R per la costruzione delle tabelle dei parametri per il modello GG e per il modello LNN rispettivamente in Tabella 4.8 e 4.9.

```
# Le funzioni qui riportate vanno applicate ai 4 dataset costruiti
con le imputazioni.

pattern <- ebPatterns(c("1, 1, 1, 1, 1, 1, 1, 1, 1,1,1,
1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,
1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1",
"1, 1, 1, 1, 1, 1, 1, 1, 1,
1,1,1,1,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,
2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2"))
gg <- emfit(data, family = "GG", pattern,num.iter = 15)
gg.post<- postprob(gg, data)
lnn<- emfit(data, family = "LNN", pattern,num.iter = 15)
lnn.post<- postprob(lnn, data)
```

A4.3.1.6 *Routine R* per la costruzione delle tabelle dei parametri per il modello GG e per il modello LNN rispettivamente in Tabella 4.8 e 4.9.

```
#geni indicati DE da entrambi i modelli
sum(gg.post[,2]>0.5 & lnn.post[,2]>0.5)
#geni indicati EE da modello GG e DE da LNN
sum(gg.post [,1]>0.5 & lnn.post[,2]>0.5)
#geni indicati EE da entrambi i modelli
sum(gg.post [,1]>0.5 & lnn.post[,1]>0.5)
#geni indicati DE da modello GG e EE da LNN
sum(gg.post[,2]>0.5 & lnn.post[,1]>0.5)
```

A4.3.1.7 *Routine R* per la valutazione delle *performance* dei metodi di imputazione per i metodo empirici bayesiani.

```
trainig.set.real<-data[1:100,]
test.set.real<-data[101:10000,]
trainig.set.mean<-data.mean[1:100,]
test.set.mean<-data.mean[101:10000,]
trainig.set.mg<-data.mean[1:100,]
test.set.mg<-data.mean[101:10000,]
trainig.set.mg<-data.mg[1:100,]
test.set.mg<-data.mg[101:10000,]
trainig.set.svd<-data.svd[1:100,]
test.set.svd<-data.svd[101:10000,]
trainig.set.knn<-data.knn[1:100,]
test.set.knn<-data.knn[101:10000,]
```

```
gg.mean<- emfit(trainig.set.mean, family = "GG", pattern,num.iter
= 15)
gg.post.mean<- postprob(gg.mean, test.set.mean)
lnn.mean<- emfit(trainig.set.mean, family = "LNN",
pattern,num.iter = 15)
lnn.post.mean<- postprob(lnn.mean, test.set.mean)
gg.real<- emfit(trainig.set.real, family = "GG", pattern,num.iter
= 15)
gg.post.real<- postprob(gg.real,test.set.real)
lnn.real<-emfit(trainig.set.real, family = "LNN", pattern,num.iter
= 15)
lnn.post.real<- postprob(lnn.real,test.set.real)

gg.mg<- emfit(trainig.set.mg, family = "GG", pattern,num.iter =
15)
gg.post.mg<- postprob(gg.mg, test.set.mg)
lnn.mg<- emfit(trainig.set.mg, family = "LNN", pattern,num.iter =
15)
lnn.post.mg<- postprob(lnn.mg, test.set.mg)
gg.real<- emfit(trainig.set.real, family = "GG", pattern,num.iter
= 15)
gg.post.real<- postprob(gg.real,test.set.real)
lnn.real<-emfit(trainig.set.real, family = "LNN", pattern,num.iter
= 15)
lnn.post.real<- postprob(lnn.real,test.set.real)

gg.svd<- emfit(trainig.set.svd, family = "GG", pattern,num.iter =
15)
gg.post.svd<- postprob(gg.svd, test.set.svd)
lnn.svd<- emfit(trainig.set.svd, family = "LNN", pattern,num.iter
= 15)
lnn.post.svd<- postprob(lnn.svd, test.set.svd)
gg.real<- emfit(trainig.set.real, family = "GG", pattern,num.iter
= 15)
gg.post.real<- postprob(gg.real,test.set.real)
```

```
lnn.real<-emfit(trainig.set.real, family = "LNN", pattern,num.iter
= 15)
lnn.post.real<- postprob(lnn.real,test.set.real)


gg.knn<- emfit(trainig.set.knn, family = "GG", pattern,num.iter =
15)
gg.post.knn<- postprob(gg.knn, test.set.knn)
lnn.knn<- emfit(trainig.set.knn, family = "LNN", pattern,num.iter
= 15)
lnn.post.knn<- postprob(lnn.knn, test.set.knn)
gg.real<- emfit(trainig.set.real, family = "GG", pattern,num.iter
= 15)
gg.post.real<- postprob(gg.real,test.set.real)
lnn.real<-emfit(trainig.set.real, family = "LNN", pattern,num.iter
= 15)
lnn.post.real<- postprob(lnn.real,test.set.real)


#MODELLAZIONE CON GG
#geni indicati DE sia nei dati reali che nei dati sostituiti con
la media gen
sum(gg.post.real[,2]>0.5 & gg.post.mean[,2]>0.5)
#geni indicati EE nei dati reali e DE nei dati sostituiti
sum(gg.post.real [,1]>0.5 & gg.post.mean[,2]>0.5)
#geni indicati EE sia nei dati reali che nei dati sostituiti con
la media gen
sum(gg.post.real [,1]>0.5 & gg.post.mean[,1]>0.5)
#geni indicati DE nei dati reali e EE nei dati sostituiti con la
media gen
sum(gg.post.real[,2]>0.5 & gg.post.mean[,1]>0.5)

#MODELLAZIONE CON LNN
#geni indicati DE sia nei dati reali che nei dati sostituiti con
la media gen
sum(lnn.post.real[,2]>0.5 & lnn.post.mean[,2]>0.5)
#geni indicati EE nei dati reali e DE nei dati sostituiti
sum(lnn.post.real [,1]>0.5 & lnn.post.mean[,2]>0.5)
```

```
#geni indicati EE sia nei dati reali che nei dati sostituiti con
la media gen
sum(lnn.post.real [,1]>0.5 & lnn.post.mean[,1]>0.5)
#geni indicati DE nei dati reali e EE nei dati sostituiti con la
media gen
sum(lnn.post.real[,2]>0.5 & lnn.post.mean[,1]>0.5)

#MODELLAZIONE CON GG
#geni indicati DE sia nei dati reali che nei dati sostituiti con
la media gen
sum(gg.post.real[,2]>0.5 & gg.post.mg[,2]>0.5)
#geni indicati EE nei dati reali e DE nei dati sostituiti
sum(gg.post.real [,1]>0.5 & gg.post.mg[,2]>0.5)
#geni indicati EE sia nei dati reali che nei dati sostituiti con
la media gen
sum(gg.post.real [,1]>0.5 & gg.post.mg[,1]>0.5)
#geni indicati DE nei dati reali e EE nei dati sostituiti con la
media gen
sum(gg.post.real[,2]>0.5 & gg.post.mg[,1]>0.5)

#MODELLAZIONE CON LNN
#geni indicati DE sia nei dati reali che nei dati sostituiti con
la media gen
sum(lnn.post.real[,2]>0.5 & lnn.post.mg[,2]>0.5)
#geni indicati EE nei dati reali e DE nei dati sostituiti
sum(lnn.post.real [,1]>0.5 & lnn.post.mg[,2]>0.5)
#geni indicati EE sia nei dati reali che nei dati sostituiti con
la media gen
sum(lnn.post.real [,1]>0.5 & lnn.post.mg[,1]>0.5)
#geni indicati DE nei dati reali e EE nei dati sostituiti con la
media gen
sum(lnn.post.real[,2]>0.5 & lnn.post.mg[,1]>0.5)

#MODELLAZIONE CON GG
#geni indicati DE sia nei dati reali che nei dati sostituiti con
la media gen
```

```
sum(gg.post.real[,2]>0.5 & gg.post.svd[,2]>0.5)
#geni indicati EE nei dati reali e DE nei dati sostituiti
sum(gg.post.real [,1]>0.5 & gg.post.svd[,2]>0.5)
#geni indicati EE sia nei dati reali che nei dati sostituiti con
la media gen
sum(gg.post.real [,1]>0.5 & gg.post.svd[,1]>0.5)
#geni indicati DE nei dati reali e EE nei dati sostituiti con la
media gen
sum(gg.post.real[,2]>0.5 & gg.post.svd[,1]>0.5)

#MODELLAZIONE CON LNN
#geni indicati DE sia nei dati reali che nei dati sostituiti con
la media gen
sum(lnn.post.real[,2]>0.5 & lnn.post.svd[,2]>0.5)
#geni indicati EE nei dati reali e DE nei dati sostituiti
sum(lnn.post.real [,1]>0.5 & lnn.post.svd[,2]>0.5)
#geni indicati EE sia nei dati reali che nei dati sostituiti con
la media gen
sum(lnn.post.real [,1]>0.5 & lnn.post.svd[,1]>0.5)
#geni indicati DE nei dati reali e EE nei dati sostituiti con la
media gen
sum(lnn.post.real[,2]>0.5 & lnn.post.svd[,1]>0.5)

#MODELLAZIONE CON GG
#geni indicati DE sia nei dati reali che nei dati sostituiti con
la media gen
sum(gg.post.real[,2]>0.5 & gg.post.knn[,2]>0.5)
#geni indicati EE nei dati reali e DE nei dati sostituiti
sum(gg.post.real [,1]>0.5 & gg.post.knn[,2]>0.5)
#geni indicati EE sia nei dati reali che nei dati sostituiti con
la media gen
sum(gg.post.real [,1]>0.5 & gg.post.knn[,1]>0.5)
#geni indicati DE nei dati reali e EE nei dati sostituiti con la
media gen
sum(gg.post.real[,2]>0.5 & gg.post.knn[,1]>0.5)
```

```
#MODELLAZIONE CON LNN
#geni indicati DE sia nei dati reali che nei dati sostituiti con
la media gen
sum(lnn.post.real[,2]>0.5 & lnn.post.knn[,2]>0.5)
#geni indicati EE nei dati reali e DE nei dati sostituiti
sum(lnn.post.real [,1]>0.5 & lnn.post.knn[,2]>0.5)
#geni indicati EE sia nei dati reali che nei dati sostituiti con
la media gen
sum(lnn.post.real [,1]>0.5 & lnn.post.knn[,1]>0.5)
#geni indicati DE nei dati reali e EE nei dati sostituiti con la
media gen
sum(lnn.post.real[,2]>0.5 & lnn.post.knn[,1]>0.5)
```

A4.3.1.7 *Routine R* per la costruzione dei grafici delle densità in appendice A4.3.

```
print(plotMarginal(gg,data.mean))
print(plotMarginal(gg,data.mg))
print(plotMarginal(gg,data.svd))
print(plotMarginal(gg,data.knn))


print(plotMarginal(lnn,data.mean))
print(plotMarginal(lnn,data.mg))
print(plotMarginal(lnn,data.svd))
print(plotMarginal(lnn,data.knn))
```

A4.3.1.8 Funzioni per la creazione dei diagrammi di dispersione per il *pattern 0:* espressione equivalente.

```
n<-10000
indici<-seq(1,n)
medieTutti<-matrix(rep(0,n*2),ncol=2)
for(i in 1:n)
```

```
{ medieTutti[i,1]<-mean(data.svd [i,1:13])
medieTutti[i,2]<-mean(data.svd [i,14:65])
}

posteriorProb=0.5
indice05<-indici [gg.post [,2]>posteriorProb
& lnn.post [,2]>posteriorProb]
n<-length(indice05)

medie05<-matrix(rep(0,n*2),ncol=2)
for(i in 1:n)
{ medie05[i,1]<-mean(data.svd [indice05[i],1:13])
medie05[i,2]<-mean(data.svd [indice05[i],14:65])
}

posteriorProb=0.7
indice07<-indici[gg.post [,2]>posteriorProb & lnn.post
[,2]>posteriorProb]
n<-length(indice07)
medie07<-matrix(rep(0,n*2),ncol=2)
for(i in 1:n)
{ medie07[i,1]<-mean(data.svd [indice07[i],1:13])
medie07[i,2]<-mean(data.svd [indice07[i],14:65])
}
posteriorProb=0.9
indice09<-indici [gg.post [,2]>posteriorProb &
lnn.post[,2]>posteriorProb]
n<-length(indice09)
medie09<-matrix(rep(0,n*2),ncol=2)
for(i in 1:n)
{ medie09[i,1]<-mean(data.svd [indice09[i],1:13])
medie09[i,2]<-mean(data.svd [indice09[i],14:65])
}
par(mfrow=c(2,2))
plot(medieTutti[,1],medieTutti[,2],xlab="Media VH321+",
ylab="Media VH321-",type="n",log="xy",main="GG e LNN")
```

```
points(medieTutti[-c(indice05,indice07,indice09),1],
medieTutti[-c(indice05,indice07,indice09),2],cex=.5)

abline(0,1)
points(medie05[,1],medie05[,2],col="blue",pch=16)
points(medie07[,1],medie07[,2],col="red",pch=16)
points(medie09[,1],medie09[,2],col="yellow",pch=16)
legend(5000,20000,legend=c("postprob>0.5","postprob>0.7",
"postprob>0.9"),col=c("blue","red","yellow"),cex=.7,
pch=c(16,16,16),bty="n")



#geni indicati dal modello GG
posteriorProb=0.5
indice05<-indici[gg.post[,2]>posteriorProb ]
n<-length(indice05)
medie05<-matrix(rep(0,n*2),ncol=2)
for(i in 1:n)
{ medie05[i,1]<-mean(data.svd[indice05[i],1:13])
medie05[i,2]<-mean(data.svd[indice05[i],14:65])
}
posteriorProb=0.7
indice07<-indici[gg.post[,2]>posteriorProb]
n<-length(indice07)
medie07<-matrix(rep(0,n*2),ncol=2)
for(i in 1:n)
{ medie07[i,1]<-mean(data.svd [indice07[i],1:13])
medie07[i,2]<-mean(data.svd [indice07[i],14:65])
}
posteriorProb=0.9
indice09<-indici[gg.post [,2]>posteriorProb ]
n<-length(indice09)
medie09<-matrix(rep(0,n*2),ncol=2)

for(i in 1:n)
```

```
{ medie09[i,1]<-mean(data.svd[indice09[i],1:13])
medie09[i,2]<-mean(data.svd[indice09[i],14:65])
}
plot(medieTutti[,1],medieTutti[,2],xlab="Media VH321+",
ylab="Media VH321-",type="n",log="xy",main="GG")
points(medieTutti[-c(indice05,indice07,indice09),1],
medieTutti[-c(indice05,indice07,indice09),2],cex=.5)
abline(0,1)
points(medie05[,1],medie05[,2],col="blue",pch=16)
points(medie07[,1],medie07[,2],col="red",pch=16)
points(medie09[,1],medie09[,2],col="yellow",pch=16)
legend(5000,20000,legend=c("postprob>0.5","postprob>0.7",
"postprob>0.9"),col=c("blue","red","yellow"),cex=.7,
pch=c(16,16,16),bty="n")


#geni indicati dal modello LNN
posteriorProb=0.5
indice05<-indici[lnn.post [,2]>posteriorProb]
n<-length(indice05)
medie05<-matrix(rep(0,n*2),ncol=2)
for(i in 1:n)
{ medie05[i,1]<-mean(data.svd[indice05[i],1:13])
medie05[i,2]<-mean(data.svd[indice05[i],14:65])
}
posteriorProb=0.7
indice07<-indici[lnn.post[,2]>posteriorProb]
n<-length(indice07)
medie07<-matrix(rep(0,n*2),ncol=2)

for(i in 1:n)
{ medie07[i,1]<-mean(data.svd[indice07[i],1:13])
medie07[i,2]<-mean(data.svd[indice07[i],14:65])
}
posteriorProb=0.9
indice09<-indici[lnn.post [,2]>posteriorProb ]
```

```
n<-length(indice09)
medie09<-matrix(rep(0,n*2),ncol=2)
for(i in 1:n)
{ medie09[i,1]<-mean(data.svd[indice09[i],1:13])
medie09[i,2]<-mean(data.svd[indice09[i],14:65])
}
plot(medieTutti[,1],medieTutti[,2],xlab="Media VH321+",
ylab="Media VH321-",type="n",log="xy",main="LNN")
points(medieTutti[-c(indice05,indice07,indice09),1],
medieTutti[-c(indice05,indice07,indice09),2],cex=.5)
abline(0,1)
points(medie05[,1],medie05[,2],col="blue",pch=16)
points(medie07[,1],medie07[,2],col="red",pch=16)
points(medie09[,1],medie09[,2],col="yellow",pch=16)
legend(5000,20000,legend=c("postprob>0.5","postprob>0.7",
"postprob>0.9"),col=c("blue","red","yellow"),cex=.7,
pch=c(16,16,16),bty="n")
```

A4.3.1.9 Funzioni per la creazione dei diagrammi di dispersione per il *pattern 1:* espressione differente.

```
#MEDIE SU DATASET DATA(log(R/G))
#medie per tutti i geni
n<-10000
medieTutti<-matrix(rep(0,n*2),ncol=2)
for(i in 1:n)
{ medieTutti[i,1]<-mean(data.svd[i,1:13])
medieTutti[i,2]<-mean(data.svd[i,14:65])
}
#geni indicati da entrambi i modelli GG e LNN
posteriorProb=0.5
indice05<-indici[gg.post[,2]>posteriorProb
& lnn.post[,2]>posteriorProb]
```

```
n<-5000
medie05<-matrix(rep(0,n*2),ncol=2)
for(i in 1:n)
{ medie05[i,1]<-mean(data.svd[indice05[i],1:13])
medie05[i,2]<-mean(data.svd[indice05[i],14:65])
}
posteriorProb=0.7
indice07<-indici[gg.post[,2]>posteriorProb
& lnn.post[,2]>posteriorProb]
medie07<-matrix(rep(0,n*2),ncol=2)
for(i in 1:n)
{ medie07[i,1]<-mean(data.svd[indice07[i],1:13])
medie07[i,2]<-mean(data.svd[indice07[i],14:65])
}
posteriorProb=0.9
indice09<-indici [gg.post[,2]>posteriorProb
& lnn.post[,2]>posteriorProb]
medie09<-matrix(rep(0,n*2),ncol=2)
for(i in 1:n)
{ medie09[i,1]<-mean(data.svd[indice09[i],1:13])
medie09[i,2]<-mean(data.svd[indice09[i],14:65])
}
par(mfrow=c(2,2))
plot(medieTutti[,1],medieTutti[,2],xlab="Media VH321+",
ylab="Media VH321-",type="n",main="GG e LNN")
points(medieTutti[-c(indice05,indice07,indice09),1],
medieTutti[-c(indice05,indice07,indice09),2],cex=.5)
abline(0,1)
points(medie05[,1],medie05[,2],col="blue",pch=16)
points(medie07[,1],medie07[,2],col="red",pch=16)
points(medie09[,1],medie09[,2],col="yellow",pch=16)
legend(5000,28000,legend=c("postprob>0.5","postprob>0.7",
"postprob>0.9"),col=c("blue","red","yellow"),cex=.7,
pch=c(16,16,16),bty="n")
#geni indicati dal modello GG
posteriorProb=0.5
```

```
indice05<-indici[gg.post[,2]>posteriorProb]
medie05<-matrix(rep(0,n*2),ncol=2)
for(i in 1:n)
{ medie05[i,1]<-mean(data.svd[indice05[i],1:13])
medie05[i,2]<-mean(data.svd[indice05[i],14:65])
}
posteriorProb=0.7
indice07<-indici$V3[gg.post [,2]>posteriorProb]
medie07<-matrix(rep(0,n*2),ncol=2)
for(i in 1:n)
{ medie07[i,1]<-mean(data.svd[indice07[i],1:13])
medie07[i,2]<-mean(data.svd[indice07[i],14:65])
}
posteriorProb=0.9
indice09<-indici[gg.post[,2]>posteriorProb]
medie09<-matrix(rep(0,n*2),ncol=2)
for(i in 1:n)
{ medie09[i,1]<-mean(data.svd[indice09[i],1:13])
medie09[i,2]<-mean(data.svd[indice09[i],14:65])
}
plot(medieTutti[,1],medieTutti[,2],xlab="Media VH321+",
ylab="Media VH321-",type="n",main="GG")
points(medieTutti[-c(indice05,indice07,indice09),1],
medieTutti[-c(indice05,indice07,indice09),2],cex=.5)
abline(0,1)
points(medie05[,1],medie05[,2],col="blue",pch=16)
points(medie07[,1],medie07[,2],col="red",pch=16)
points(medie09[,1],medie09[,2],col="yellow",pch=16)
legend(5000,28000,legend=c("postprob>0.5","postprob>0.7",
"postprob>0.9"),col=c("blue","red","yellow"),cex=.7,
pch=c(16,16,16),bty="n")
#geni indicati dal modello LNN
posteriorProb=0.5
indice05<-indici[lnn.post[,2]>posteriorProb]
medie05<-matrix(rep(0,n*2),ncol=2)
for(i in 1:n)
```

```
{ medie05[i,1]<-mean(data.svd[indice05[i],1:13])
medie05[i,2]<-mean(data.svd[indice05[i],14:65])
}
posteriorProb=0.7
indice07<-indici[lnn.post[,2]>posteriorProb]
medie07<-matrix(rep(0,n*2),ncol=2)
for(i in 1:n)
{ medie07[i,1]<-mean(data.svd[indice07[i],1:13])
medie07[i,2]<-mean(data.svd[indice07[i],14:65])
}
posteriorProb=0.9
indice09<-indici[lnn.post[,2]>posteriorProb]
medie09<-matrix(rep(0,n*2),ncol=2)
for(i in 1:n)
{ medie09[i,1]<-mean(data.svd[indice09[i],1:13])
medie09[i,2]<-mean(data.svd[indice09[i],14:65])
}
plot(medieTutti[,1],medieTutti[,2],xlab="Media VH321+",
ylab="Media VH321-",type="n",main="LNN")
points(medieTutti[-c(indice05,indice07,indice09),1],
medieTutti[-c(indice05,indice07,indice09),2],cex=.5)
abline(0,1)
points(medie05[,1],medie05[,2],col="blue",pch=16)
points(medie07[,1],medie07[,2],col="red",pch=16)
points(medie09[,1],medie09[,2],col="yellow",pch=16)
legend(5000,28000,legend=c("postprob>0.5","postprob>0.7",
"postprob>0.9"),col=c("blue","red","yellow"),cex=.7,
pch=c(16,16,16),bty="n")
```

Capitolo 5

Metodi Bayesiani empirici sui dati

5.1 Analisi esplorativa

Introduzione

Nel seguito verranno illustrate alcune tecniche di analisi condotte al fine di esplorare le potenzialità di alcune tecnologie di analisi multivariata nell'estrapolare informazioni di interesse biologico sui geni disponibili nel *dataset*. Con *informazioni di interesse biologico* si intendono informazioni sia sul livello di espressione anomala (sovraespressione/sottoespressione) di alcuni singoli geni sia sulla co-espressione di gruppi di geni specifici di un certo gruppo. Si è pertanto fatto il primo tentativo di isolare geni differenzialmente espressi.

5.1 Le Curve di Andrews

Uno strumento sofisticato per la visualizzazione di dati multidimensionali sono i diagrammi di Andrews (*Andrews plot*). Essi sono particolarmente indicati per individuare unità statistiche simili e/o aberranti (*outliers*). Il metodo consiste nel trasformare ogni osservazione in una serie di Fourier, ovvero mappare una

osservazione *p*-dimensionale in uno spazio bidimensionale mediante la trasformazione:

$$f_x(t) = \frac{x_1}{\sqrt{2}} + x_2 sent + x_3 \cos t + x_4 sen2t + x_5 \cos 2t.....$$

al variare del parametro t tra $[-\pi, \pi]$.

Le caratteristiche dei dati sono preservate dalle curve di Andrews grazie ad alcune caratteristiche di cui gode la serie sopra presentata. Essa, infatti:

1. *Preserva la media.*

   Si indichi con $\underline{\bar{x}}$ il vettore delle medie delle *n* osservazioni:

   $\underline{\bar{x}} = (\bar{x}_1, \bar{x}_2, ..., \bar{x}_p)^T$

   con

   $\bar{x}_j = \frac{\sum_{i=1}^{n} x_{ij}}{n} \quad con\ j = 1, ..., p$

   allora la trasformazione relativa alla media $\bar{x}$, coincide con la media delle trasformazioni corrispondenti alle *n* osservazioni.

   Si ha quindi:

   $$f_x(t) = \frac{\sum_{i=1}^{n} f_{x_i}(t)}{n}.$$

2. *Preserva le distanze.*

   La distanza tra due osservazioni è preservata nella trasformazione in serie:

   $$\pi \| x - y \|^2 = \pi \sum_{i=1}^{p} (x_i - y_i)^2 = \| f_x(t) - f_y(t) \| = \| f_x(t) - f_y(t) \|_{L_2} = \int_{-\pi}^{\pi} \left[f_x(t) - f_y(t) \right] dt.$$

   In questo modo si ha proporzionalità tra la distanza che separa le due funzioni calcolate nei due punti *x* e *y* e la distanza Euclidea tra gli stessi punti.

3. *Preserva l'ordinamento.*

   Se un punto *y* è collocato sulla linea che unisce *x* e *z*, allora per qualsiasi valore di *t*, la $f_y(t)$ è posizionata tra $f_x(t)$ e $f_z(t)$.

4. *Preserva la varianza.*

   Se le osservazioni sono incorrelate e con varianza σ^2, allora la varianza della funzione in *t* è espressa da:

$$\text{var}[f_x(t)] = \sigma^2\left(\frac{1}{2} + sen^2 t + \cos^2 t + sen^2 2t + \cos^2 2t ...\right)$$

   Si possono distinguere ora due casi:

   - *p* pari:

     la varianza si riduce ad una costante, $\frac{1}{2}\sigma^2 p$;

   - *p* dispari:

     la varianza varia tra $\sigma^2(p-1)$ e $\sigma^2(p+1)$.

   Si noti che nel primo caso la varianza non dipende da *t* e nel secondo caso, l'influenza di *t* si fa più debole all'aumentare di *p*. In questo modo la variabilità della funzione è quasi costante su tutto l'intervallo [-π, π], cosa che facilita l'interpretazione del grafico.

5. *Fornisce una proiezione nello spazio unidimensionale.*

   Per un particolare valore di $t=t_0$ il valore della funzione è proporzionale alla lunghezza della proiezione del vettore x'=(x_1, x_2, ..., x_p) sul vettore

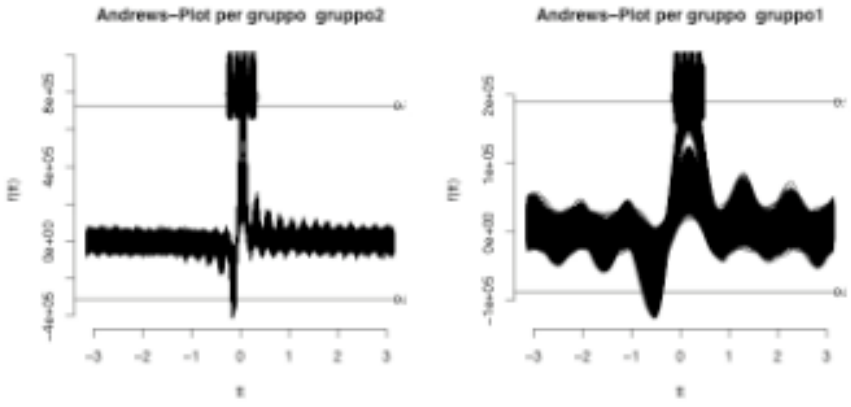
$$f_1(t_0) = (1/2, sent_0, \cos t_0, sen2t_0, \cos 2t_0, ...)$$

   Poiché

$$f_x(t_0) = \left\{\frac{x' f_1(t_0)}{f_1'(t_0)}\right\} \times \left[f_1'(t_0) f_1(t_0)\right]$$

Anche in questo caso, la proiezione nello spazio unidimensionale può rilevare *cluster* di osservazione, *outliers* ed altre peculiarità che diventano evidenti in questo spazio, ma, in altre circostanze risultano oscurate dalle dimensioni.

La prima domanda cui rispondere è se le curve di Andrews siano in grado di evidenziare geni con comportamenti anomali.
Anzitutto i profili di Andrews sono stati tracciati per tutti i geni, distinguendo i due gruppi, denominando *gruppo 1* tutti i geni appartenenti a VH321+ e con *gruppo 2* tutti i geni appartenenti a VH321-. I grafici della Figura 5.1 mettono in luce un'effettiva diversità tra i profili genetici dei due gruppi. Si noti anche una significativa differenza tra il campo di variazione delle curve nei due gruppi.



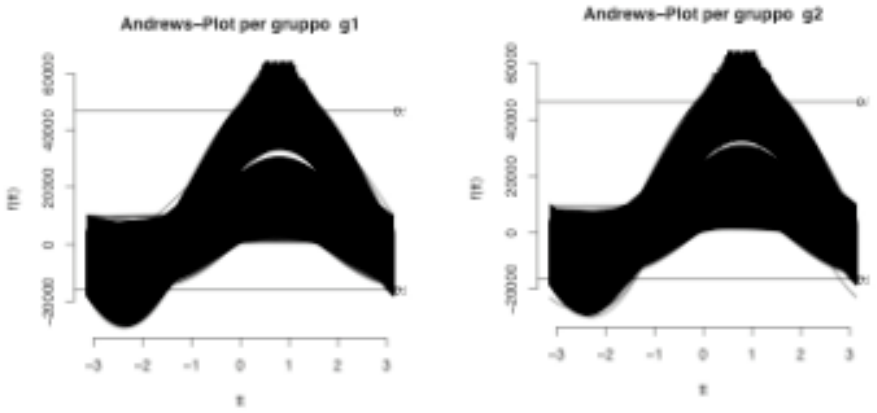
(a) (b)

Figura 5.1 Curve di Andrews per i 10000 geni distinti per i due gruppi. (a) Profili dei geni calcolati sui soggetti appartenenti al *gruppo 1*. (b) Profili dei geni calcolati sui soggetti appartenenti al *gruppo 2*.

Bisogna comunque tener presente che la forma delle curve di Andrews dipende anche dal numero di osservazioni di cui si dispone. Dal momento che i due gruppi

hanno numerosità diverse, è possibile che le discrepanze tra le curve siano imputabili a questo più che ad una effettiva differenza dei livelli di espressione dei geni.

Per verificare questa ipotesi si sono selezionati tre soggetti a caso in ciascun gruppo in modo da avere la stessa numerosità in ogni classe e si sono nuovamente tracciati i profili, proposti in Figura 5.2.



(a) (b)

Figura 5.1 Curve di Andrews per i 10000 geni distinti per i due gruppi. (a) Profili dei geni calcolati su 3 soggetti appartenenti al *gruppo 1*. (b) Profili dei geni calcolati su 3 soggetti appartenenti al *gruppo 2*.

Effettivamente le differenze sono praticamente scomparse, la differenza tra le curve di Andrews in Figura 5.2 sono quindi dovute, quasi completamente, alla grossa differenza tra le numerosità campionarie dei due gruppi.

La funzione *andrews.plot* (in appendice A5.), con l'ausilio della quale si sono tracciati i profili di Andrews, fornisce in *output* l'elenco dei geni le cui curve oltrepassano le due bande fissate in modo da lasciar uscire il 25% dei profili sia verso l'alto che verso il basso.

Dall'analisi grafica, non si possono trarre conclusioni su differenze di espressioni di geniche tra i due gruppi, ma si può ipotizzare la presenza di alcuni particolari geni che caratterizzano uno dei gruppi. L'utilizzo delle curve di Andrews sarà utile per valutare il profilo genico dei geni che saranno identificati come differenzialmente espressi attraverso i metodi empirici bayesiani proposti nei paragrafi successivi.

5.2 Analisi preliminare

L'analisi preliminare consiste nel valutare la possibilità d'uso dei modelli GG e LNN descritti nel capitolo 2 attraverso l'analisi visiva di alcuni grafici che si possono ottenere con i metodi *checkC*CV e *checkModel* della libreria *EBarrays*. Per valutare l'assunzione relativa al coefficiente di variazione costante, è stato costruito il grafico di Figura 5.3 che mostra una forte variabilità del CV per i dati con media di espressione prossima a 1. In questo caso, il CV dipende chiaramente dalla media e l'assunzione non sembra adeguata. I grafici di Figura 5.4 e 5.5 mostrano rispettivamente i Gamma QQ− *plots* e i LogNormale QQ − *plots* rappresentati per sottoinsiemi di dati che condividono la stessa media empirica. Si possono notare in tutti i QQ −*plots* notevoli difficoltà, soprattutto nelle code, dei modelli Gamma e LogNormale di aderire ai dati relativi alle misure di espressione. Nonostante questo, le simulazioni nel Capitolo 3 hanno evidenziato che le tecniche sono robuste rispetto la mancata assunzione relativa al CV costante e a distribuzioni marginali empiriche dei dati che male si approssimano ai modelli da utilizzare.

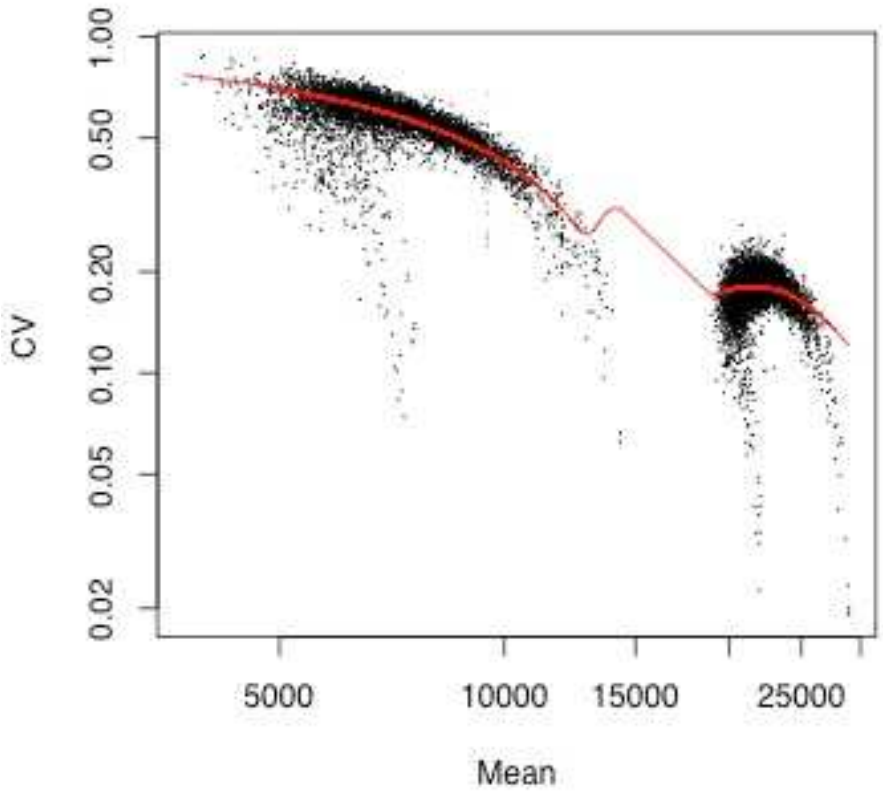


Figura 5.3 Coefficiente di variazione in funzione della media per i dati di espressione genica in soggetti leucemici.

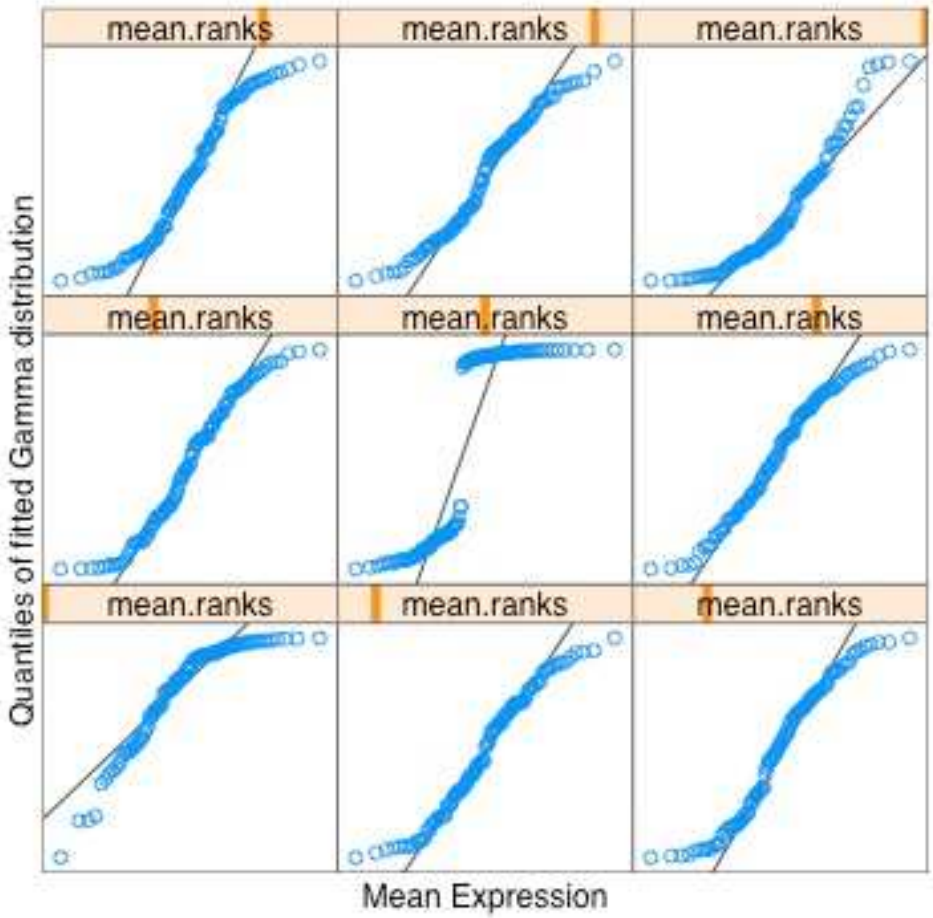


Figura 5.4 Dati di espressione genica dei soggetti leucemici: Gamma QQ – *plots* per sottoinsiemi di dati che condividono la stessa media empirica.

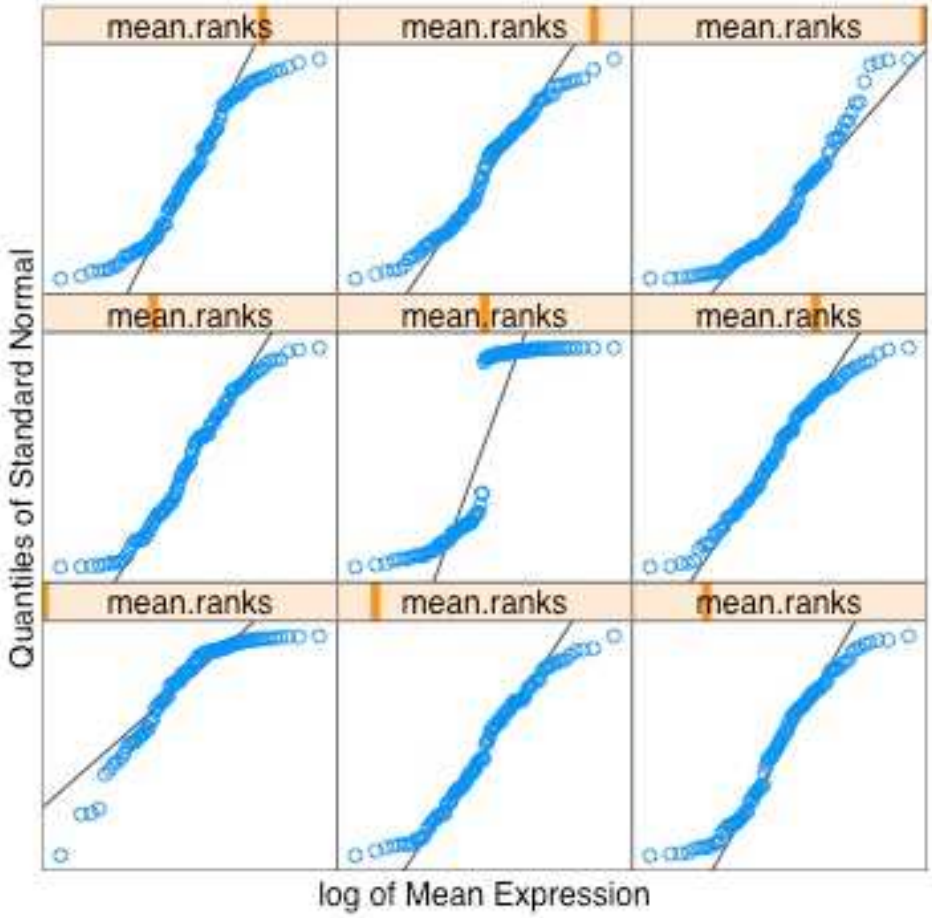


Figura 5.5 Dati di espressione genica dei soggetti leucemici: LogNormale QQ – *plots* per sottoinsiemi di dati che condividono la stessa media empirica.

5.3 Confronto tra VH321+ e VH32-

Lo scopo del paragrafo è valutare le differenze tra due sottoinsiemi del campione, attraverso l'analisi delle differenze di espressioni tra i geni dei due sottogruppi.
In termini statistici, il problema si traduce nella valutazione del sistema di ipotesi:

$$\begin{cases} H_0 : \mu_{VH321+} = \mu_{VH321-} \Rightarrow Espressione\ genica\ equivalente\ (EE) \\ H_1 : \mu_{VH321+} \neq \mu_{VH321-} \Rightarrow Espressione\ genica\ differenziale\ (DE) \end{cases}$$

per ogni singolo gene, dove μ_g indica la media di espressione per i soggetti appartenenti al gruppo g (g=$\{$VH321+, VH321-$\}$).
Mediante la funzione *emfit* di *EBarrays* sono stati stimati i parametri dei modelli Gamma-Gamma e LogNormale-Normale che sono risultati rispettivamente:

$$GG : (\hat{\alpha}, \hat{\alpha}_0, \hat{\nu}, \hat{p}) = (3.38, 3.12, 10000, 0.999)$$
$$LNN : (\hat{\mu}_0, \hat{\sigma}, \hat{\tau}, \hat{p}) = (9.27, 0.49, 0.55, 0.967)$$

La tecnica utilizzata permette di calcolare la probabilità a posteriori (forniti i dati di espressione genica) dei geni di essere differenzialmente espressi. Considerando differenzialmente espressi i geni con probabilità a posteriori maggiore di 0.5 il modello LNN identifica 139 geni differenzialmente espressi su 10000 pari a 1.39% del totale, mentre il modello GG ne individua solo 7 pari allo 0.07% del totale. Come si può vedere dalla Tabella 5.1, i 7 geni identificati come differenzialmente espressi da GG, fanno parte dei geni differenzialmente espressi dal modello LNN; mentre 139 sono stati identificati come differenzialmente espressi solo da LNN.

	Espressione prevista da LNN	
Espressione prevista da GG	Equivalente	Differente
Equivalente	9854	139
Differente	0	7

Tabella 5.1 Confronto circa l'identificazione dell'espressione genica secondo i modelli GG e LNN.

Non sembra esserci un modello preferibile all'altro, entrambi mostrano grosse difficoltà nel descrivere la distribuzione marginale dei dati, come è possibile vedere nei grafici in Figura 5.6 e 5.7 che confermano le assunzioni fatte nell'analisi preliminare proposta nel paragrafo precedente. Essi mostrano le densità dei modelli GG e LNN stimati, sovrapposta alla densità marginale dei dati stimata attraverso il metodo del nucleo.

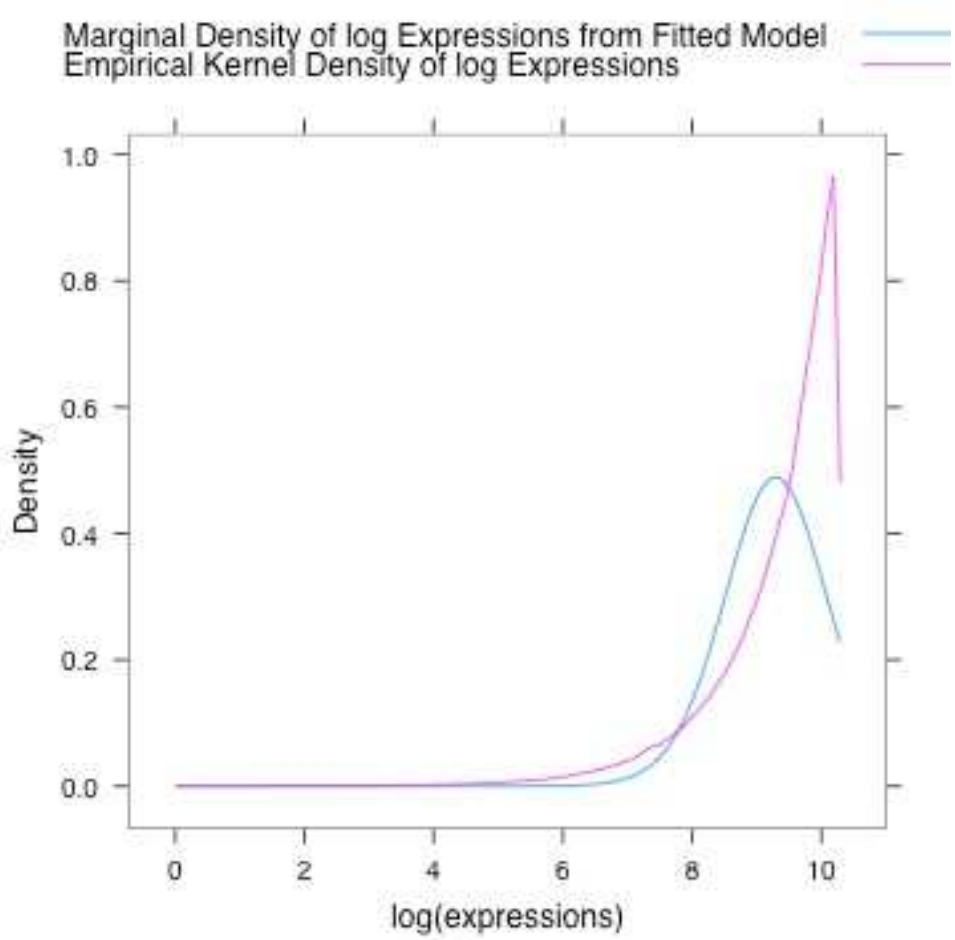


Figura 5.6 Distribuzione marginale del modello GG sovrapposta alla densità marginale dei dati di espressione genica calcolata attraverso il metodo del nucleo.

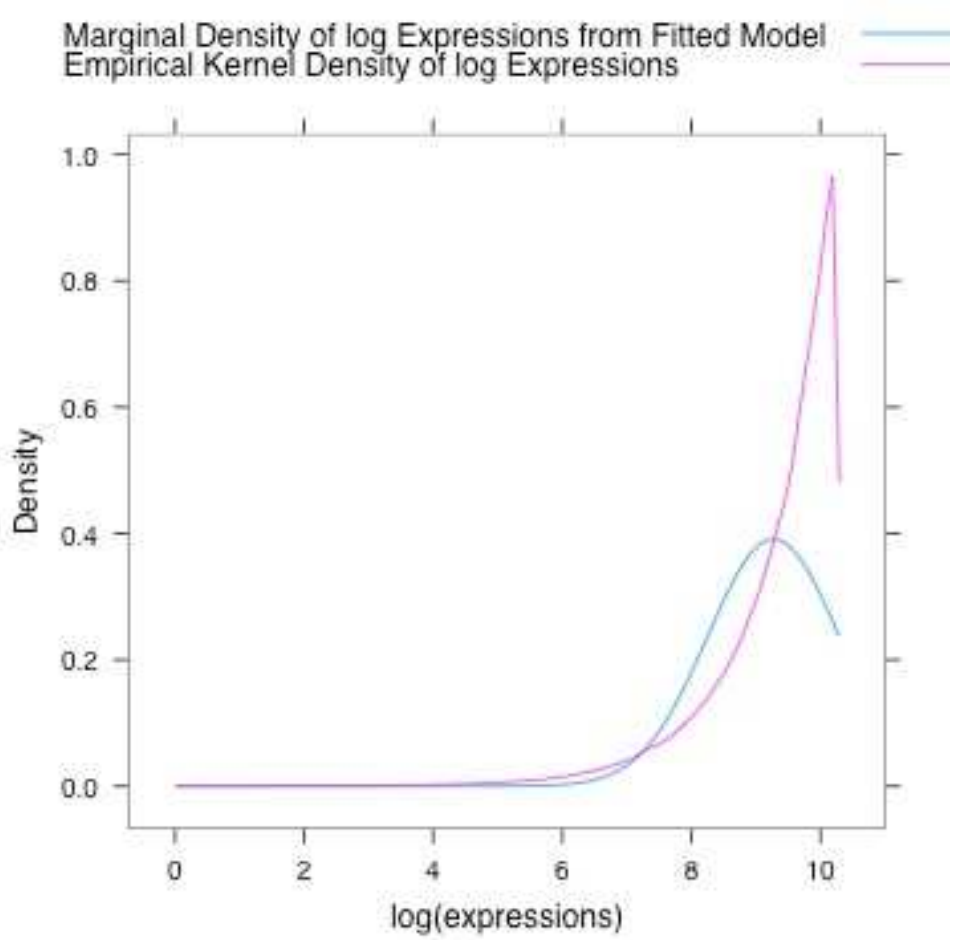


Figura 5.7 Distribuzione marginale del modello LNN sovrapposta alla densità marginale dei dati di espressione genica calcolata attraverso il metodo del nucleo.

Considerando la probabilità a posteriori di diversa espressione secondo i due modelli utilizzati, si possono ricavare i grafici in Figura 5.8 e 5.9 che evidenziano con diversa colorazione i geni differenzialmente espressi con probabilità a posteriori superiore a 0.5, 0.7, 0.9.

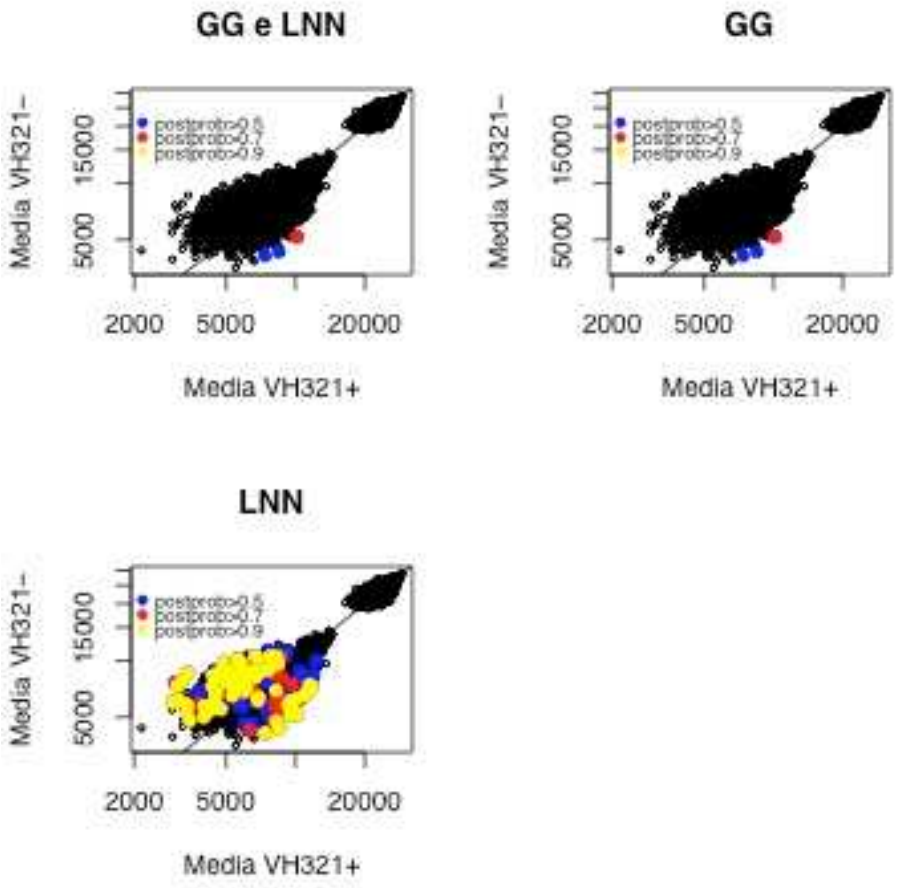


Figura 5.8 Diagrammi di dispersione del rapporto tra le 2 tinte; sull'asse delle ascisse la media di espressione dei geni negli individui appartenenti al gruppo VH321+, sulle ordinate quella relative al gruppo VH321-. Con colorazione diversa sono indicati i geni con probabilità a posteriori superiore a 0.5 di essere differenzialmente espressi secondo i vari modelli. In alto a sinistra i geni indicati da entrambe i modelli, negli altri 2 grafici i geni indicati rispettivamente dal modello GG e LNN.

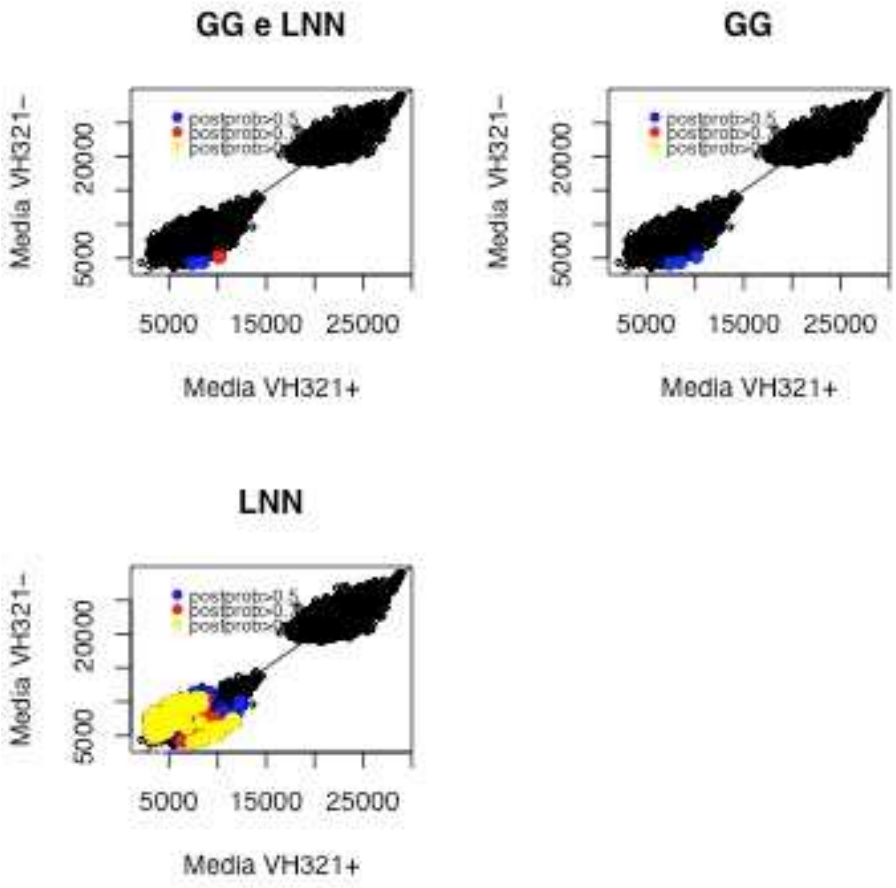


Figura 5.9 Diagrammi di dispersione del rapporto tra le 2 tinte; sull'asse delle ascisse la media di espressione dei geni negli individui appartenenti al gruppo VH321+, sulle ordinate quella relative al gruppo VH321-. Con colorazione diversa sono indicati i geni con probabilità a posteriori superiore a 0.5 di essere differenzialmente espressi secondo i vari modelli. In alto a sinistra i geni indicati da entrambe i modelli, negli altri 2 grafici i geni indicati rispettivamente dal modello GG e LNN.

I grafici in Figura 5.8 e 5.9 mettono in luce la scarsa capacità del modello GG di identificare geni differenzialmente espressi nei due gruppi. Ciò nonostante si nota la capacità, nei casi di probabilità a posteriori pari a 0.05 o 0.07, del modello GG di identificare geni la cui espressione media dei 2 gruppi si discosta dalla retta bisettrice. Il modello LNN sembra comportarsi peggio, ma questo è dovuto al fatto che esso lavora sul logaritmo naturale dei dati. Infatti, nei grafici di Figura 5.9, dove la media di espressione nelle replicazioni corrisponde al logaritmo in base 2 del rapporto delle 2 tinte, il modello LNN identifica leggermente meglio i geni la cui espressione media nei 2 gruppi si discosta dalla retta bisettrice.

Infine interessante è notare il comportamento delle curve di livello degli *odds* a posteriori dell'espressione differenziale come definita nell'espressione 2.3, che nel caso di imposizione del modello Gamma-Gamma assume la forma 2.24.

Sono stati rappresentati i geni identificati come equivalentemente espressi e differenzialmente espressi dal modello Gamma-Gamma. Sono state infine sovrapposte, come già proposto nel Capitolo 3 relativo alle simulazioni, le curve degli *odds* a posteriori di espressione differenziale superiore a 1, 10 e 100 che indicano rispettivamente che i geni esterni a tali curve presentano una probabilità a posteriori di espressione differenziale superiore di 1, 10, 1000 volte rispetto alla probabilità di espressione equivalente.

Come per le simulazioni, è ancora possibile notare la forma curvilinea degli *odds* che fanno in modo che la regione compresa tra le due curve dello stesso livello per valori di intensità bassi e alti nelle 2 condizioni sia più ampie. Questo, grazie alla già citata proprietà del modello Gamma-Gamma che fa in modo che l'*odds* sia funzione del valore complessivo dell'espressione genica nelle 2 condizioni.

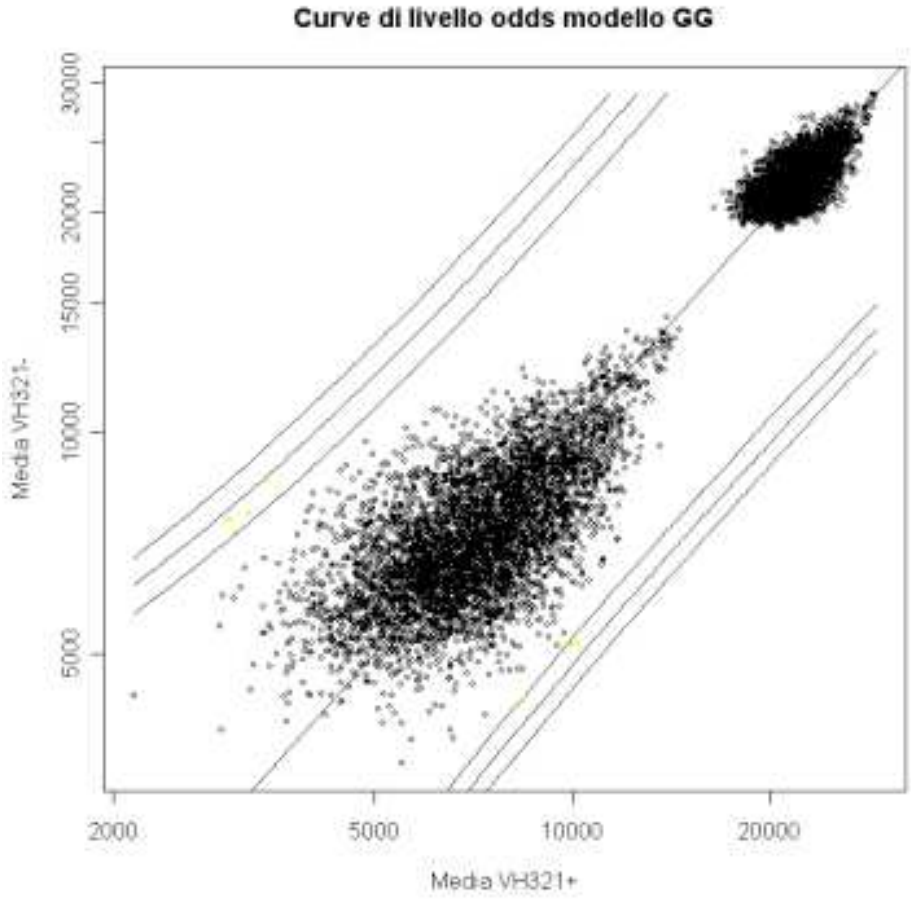


Figura 5.10. Curve degli *odds* calcolate mediante il modello GG sui dati relativi alle medie di espressione dei soggetti del gruppo VH321+ (sulle ascisse) e VH321- (sulle ordinate). Corrispondono agli *odds* di 1, 10, 100 rispettivamente dalla curva più interna alla curva più esterna. I punti compresi tra le 2 curve più interne rappresentano i geni identificati come equivalentemente espressi dal modello GG.

Sono state infine tracciate le curve di Andrews per quei geni che sono identificati come differenzialmente espressi. In prima analisi considero i geni identificati da entrambe i modelli:

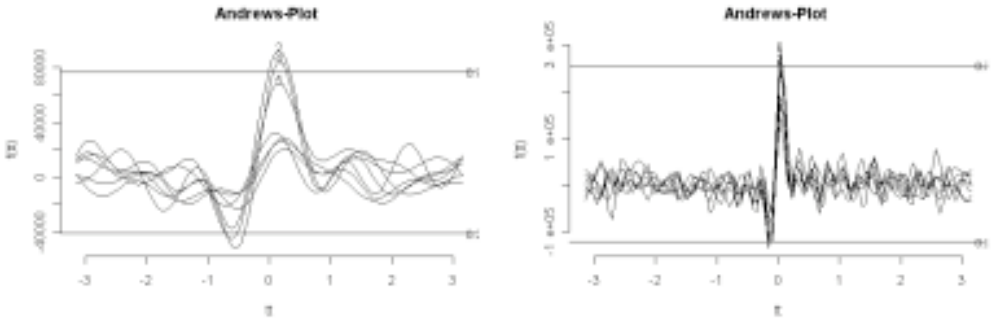


Figura 5.11. Curve di Andrews per i geni identificati differenzialmente espressi al 95% sia dal modello GG che dal modello LNN. A sinistra il grafico che si riferisce al gruppo di 13 soggetti appartenenti a VH321+, il secondo ai 52 soggetti del gruppo VH321-.

E' evidente dal grafico una diversità dei profili di espressione nei due gruppi, sia per quanto riguarda l'andamento, sia per quanto riguarda la variabilità. Ma, come precedentemente dimostrato, tali differenze possono dipendere dalla diversa numerosità campionaria nei due gruppi. Ancora una volta, quindi, sono stati considerati 3 soggetti per ciascun gruppo e confrontate così le curve di Andrews in campioni con la stessa numerosità.

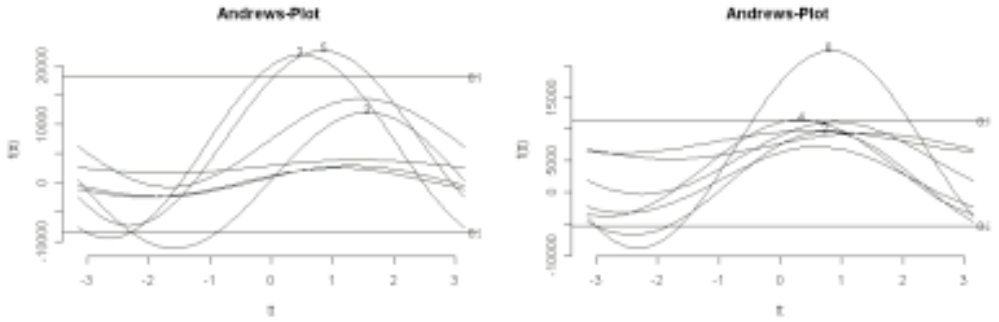


Figura 5.12. Curve di Andrews per i geni identificati come differenzialmente espressi sia dal modello GG che dal modello LNN. A sinistra il grafico che si riferisce a 3 dei 13 soggetti appartenenti a VH321+, il secondo a 3 dei 52 soggetti del gruppo VH321-.

Le differenze, questa volta, non sembrano appianarsi, soprattutto se si considera il *range* di variabilità.

La stessa operazione è stata fatta sui geni identificati come differenzialmente espressi solo dal modello LNN.

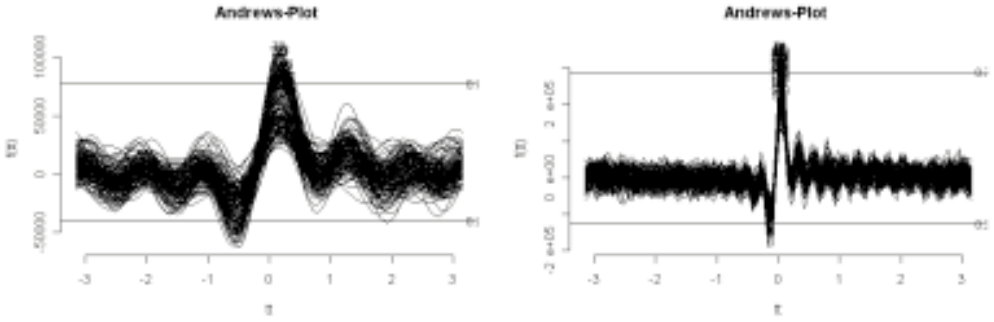


Figura 5.13. Curve di Andrews per i geni identificati differenzialmente espressi al 95% solamente dal modello LNN. A sinistra il grafico che si riferisce al gruppo di 13 soggetti appartenenti a VH321+, il secondo ai 52 soggetti del gruppo VH321-.

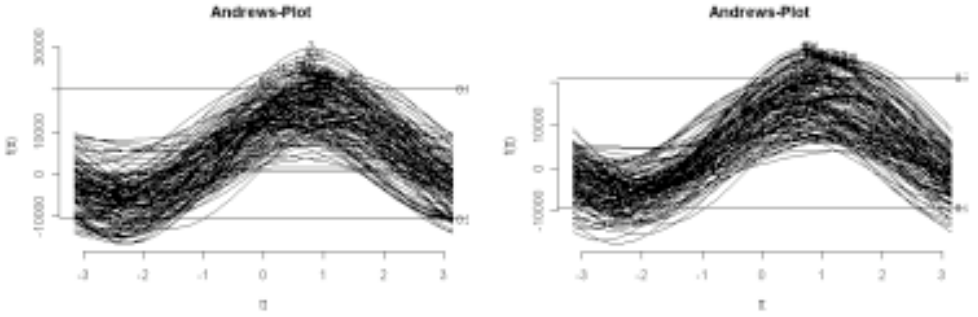


Figura 5.12. Curve di Andrews per i geni identificati come differenzialmente espressi solamente dal modello LNN. A sinistra il grafico che si riferisce a 3 dei 13 soggetti appartenenti a VH321+, il secondo a 3 dei 52 soggetti del gruppo VH321-.

In questo caso le differenze si appianano quasi totalmente. Da questa analisi conclusiva è possibile ipotizzare che il modello GG identifica con più difficoltà geni differenzialmente espressi rispetto al modello LNN. Ciò nonostante, i geni che GG identifica sono caratterizzati da profili molto diversi nei due gruppi, a differenza di quelli identificati da LNN.

I grafici forniti nel corso del capitolo, volti a modellare e fare previsioni sui dati attraverso metodi bayesiani empirici, hanno evidenziato la presenza di *cluster* intrinsechi nei dati. In particolar modo, se si considera il grafico prodotto dalla funzione *checkCCV* della libreria *EBarrays*, proposto in Figura 5.3, evidenzia la presenza di due sottoclassi che influenzano la particolare andatura della curva del coefficiente di variazione, caratterizzata da una dipendenza del coefficiente di variazione dalla media in maniera opposta di quello che ci si aspettava. Come priva valutazione si è voluta verificare l'ipotesi che i due *cluster* intrinsechi dei dati siano proprio le due classi considerate nel problema. Suddiviso quindi il *dataset* in due, sono state condotte le analisi preliminari.

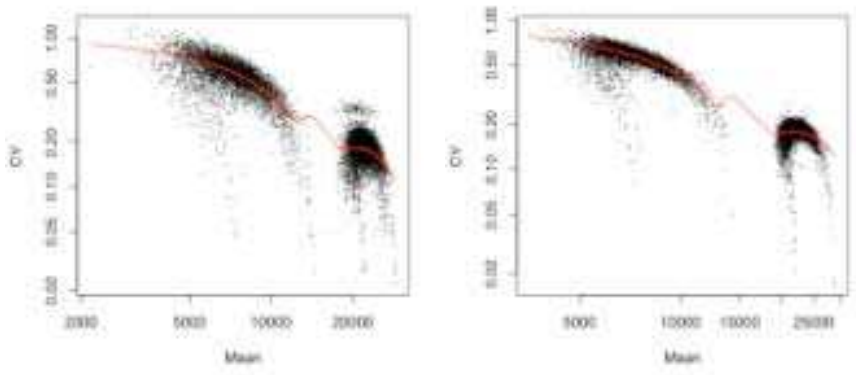


Figura 5.13 Coefficiente di variazione in funzione della media per i dati di espressione genica in soggetti leucemici, a sinistra sui soggetti del gruppo VH321+, a destra quelli del gruppo VH321-.

I grafici in Figura 5.13 smentiscono l'ipotesi di partenza: è evidente, ancora, la presenza di due *cluster* naturali dei dati, in entrambi i sottogruppi. Quindi il *dataset* è stato suddiviso sulla base della misura della media per ogni gene. Se la media di espressione genica dell'*i*-esimo gene è inferiore a 13000 l'*i-esimo* gene appartiene al primo gruppo, se l'*i*-esimo gene ha media di espressione superiore a

20000 esso appartiene al secondo gruppo. Per i 248 geni non considerati nei due gruppi le analisi con i metodi bayesiani empirici non hanno individuato geni differenzialmente espressi né con il modello Gamma-Gamma, né con il modello LogNormale-Normale. Il primo *dataset* contiene i dati di espressione di 4975 geni su 65 pazienti, mentre il secondo ne contiene 4777 su 65 pazienti.
Per i due sottoinsiemi di dati sono state condotte le analisi viste in precedenza con i metodi bayesiani empirici, evidenziando in ogni passaggio il miglior adattamento da parte di entrambe i modelli ai due sottoinsiemi presi singolarmente piuttosto che al *dataset* completo di dati.

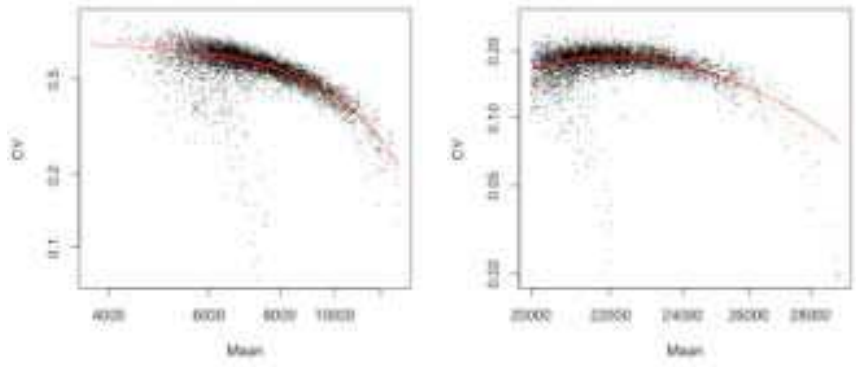


Figura 5.14 Coefficiente di variazione in funzione della media per i dati di espressione genica in soggetti leucemici, a sinistra sui soggetti del gruppo con media inferiore a 13000, a destra quelli con media superiore a 20000.

L'ipotesi del coefficiente di variazione costante non è verificata, ma l'andamento della curva del coefficiente di variazione rispetto la media è nella direzione ipotizzata: all'aumentate della media il coefficiente di variazione tende a diminuire.
Per i due *dataset* si sono poi condotte le analisi dei modelli.

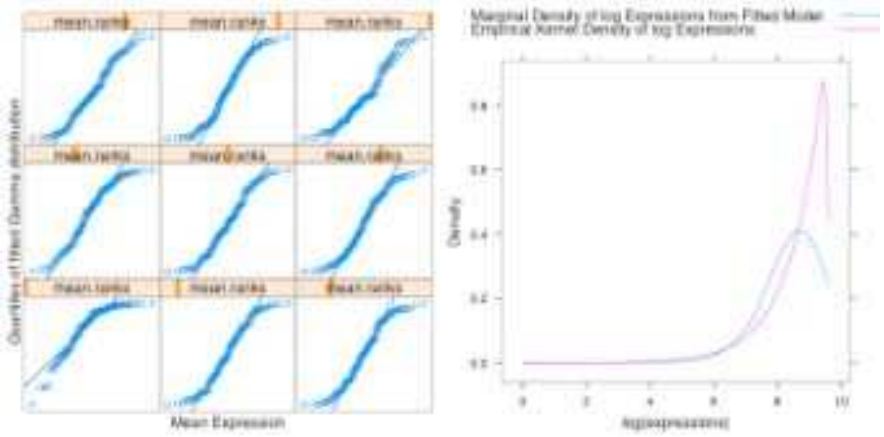


Figura 5.15 Adattamento del modello GG ai dati di espressione genica con media inferiore a 13000.

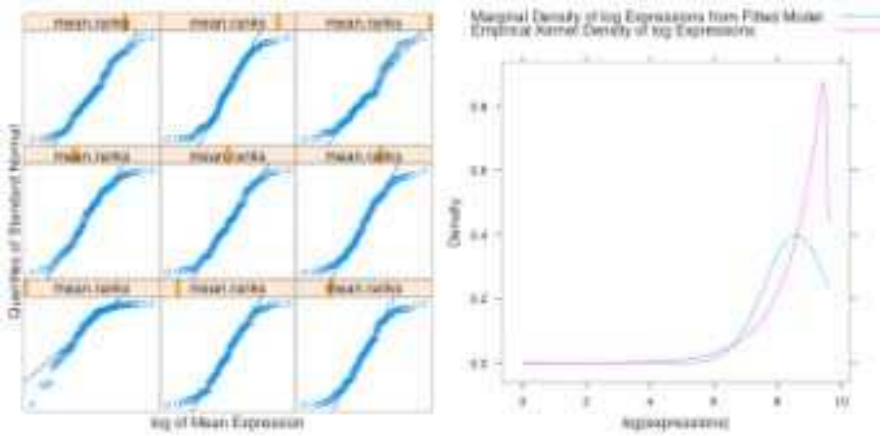


Figura 5.16 Adattamento del modello LNN ai dati di espressione genica con media inferiore a 13000.

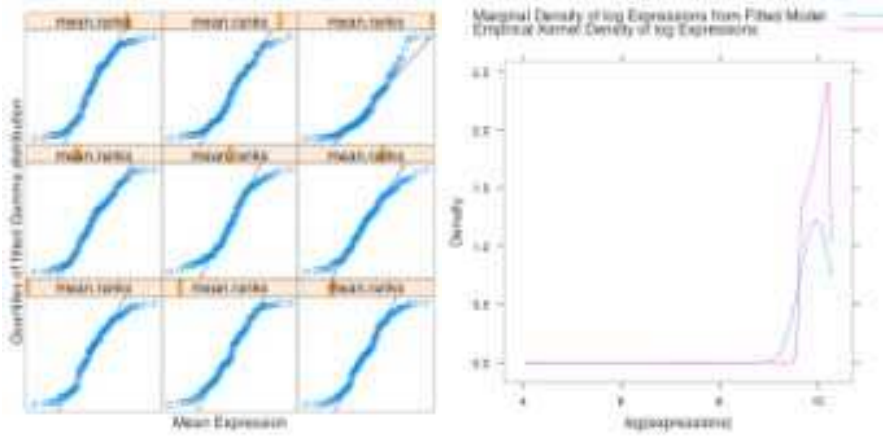


Figura 5.17 Adattamento del modello GG ai dati di espressione genica con media superiore a 20000.

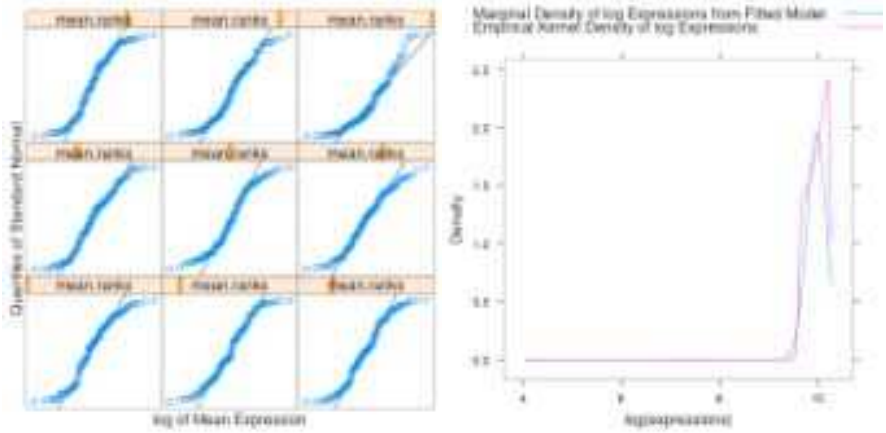


Figura 5.18 Adattamento del modello LNN ai dati di espressione genica con media superiore a 20000.

	Espressione prevista da LNN	
Espressione prevista da GG	Equivalente	Differente
Equivalente	4731	243
Differente	0	1

	Espressione prevista da LNN	
Espressione prevista da GG	Equivalente	Differente
Equivalente	4411	336
Differente	1	29

Tabella 5.2 Confronto circa l'identificazione dell'espressione genica secondo i modelli GG e LNN nei due sottoinsiemi di dati, a sinistra i dati di espressione dei geni con media inferiore a 13000, a destra i geni la cui media è superiore a 20000.

Le due tabelle sopra descritte mostrano i risultati della maggior efficacia da parte dei due modelli parametrici nel descrivere i dati. L'insieme dei geni identificati come differenzialmente espressi da entrambe i modelli sono 30 (1 per i geni con media inferiore a 13000 e 29 per i geni con media superiore a 20000) contro i 7 identificati precedentemente (Tabella 5.1). Si evidenzia un miglioramento per entrambe i modelli: il modello LNN identifica differenzialmente espresso 579 geni che non vengono identificati tali dal modello GG. In totale il modello GG riesce ad identificare 31 geni differenzialmente espressi, mentre il modello LNN ne riesce ad identificare 609 contro i 146 precedentemente identificati.

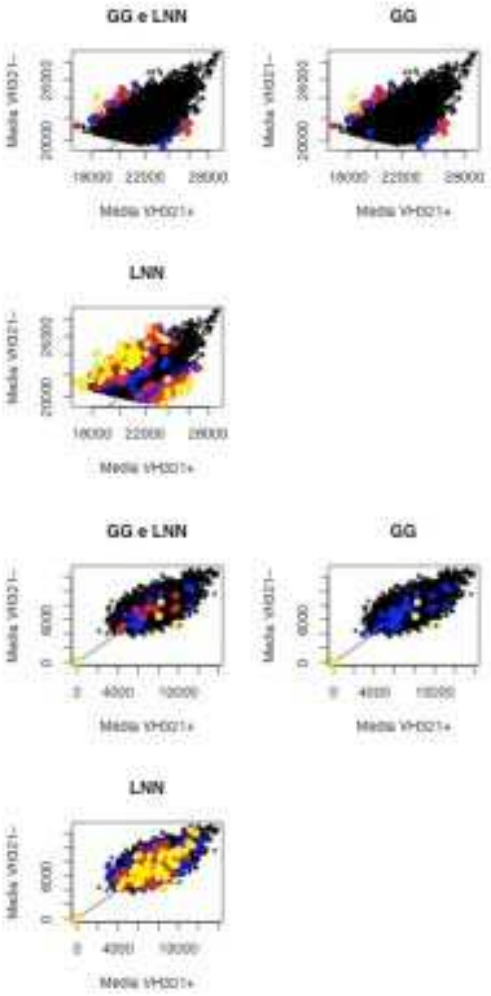


Figura 5.19 Diagrammi di dispersione del rapporto tra le 2 tinte dei geni con media inferiore a 13000; sull'asse delle ascisse la media di espressione dei geni negli individui appartenenti al gruppo VH321+, sulle ordinate quella relative al gruppo VH321-. Con colorazione diversa sono indicati i geni con probabilità a posteriori superiore a 0.5 di essere differenzialmente espressi secondo i vari modelli. In alto a sinistra i geni indicati da entrambe i modelli, negli altri 2 grafici i geni indicati rispettivamente dal modello GG e LNN. In basso i diagrammi sono stati costruiti calcolando la media attraverso la trasformazione logaritmica in base 2 del rapporto delle 2 tinte.

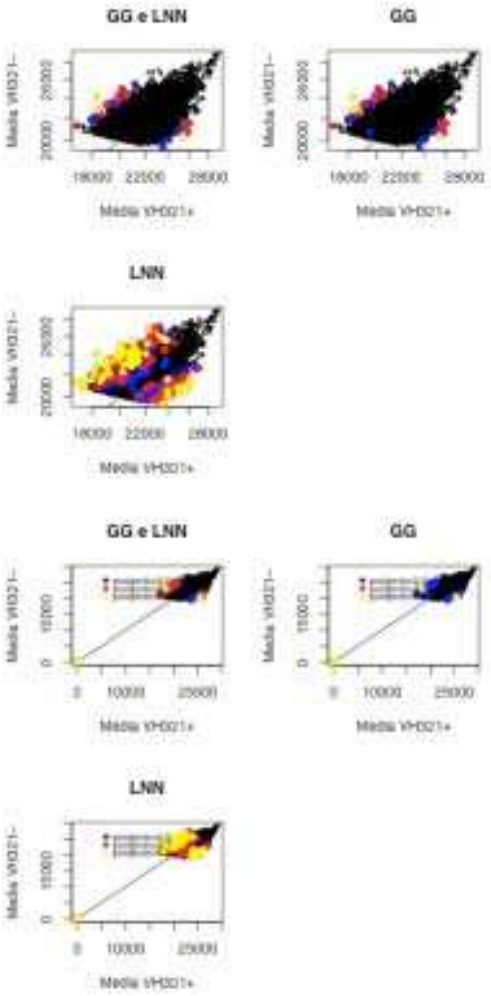


Figura 5.20 Diagrammi di dispersione del rapporto tra le 2 tinte dei geni con media superiore a 20000; sull'asse delle ascisse la media di espressione dei geni negli individui appartenenti al gruppo VH321+, sulle ordinate quella relative al gruppo VH321-. Con colorazione diversa sono indicati i geni con probabilità a posteriori superiore a 0.5 di essere differenzialmente espressi secondo i vari modelli. In alto a sinistra i geni indicati da entrambe i modelli, negli altri 2 grafici i geni indicati rispettivamente dal modello GG e LNN. In basso i diagrammi sono stati costruiti calcolando la media attraverso la trasformazione logaritmica in base 2 del rapporto delle 2 tinte.

CAPITOLO 5

Appendice 5

A5.1 Elenco dei geni identificati come differenzialmente espressi nei due gruppi.

***Geni DE da GG* (*probabilità del* 95%)**

965, 2471, 2643, 3297, 3387, 4147, 4876.

***Geni DE da LNN* (*probabilità del* 95%)**

98, 110, 134, 140, 207, 223, 244, 319, 382, 402, 430, 480, 528, 533, 588, 628, 633, 653, 659, 672, 685, 709, 744, 753, 772, 805, 825, 879, 888, 896, 914, 947, 950, 965, 971, 973, 1011, 1030, 1063, 1065, 1106, 1111, 1118, 1122, 1132, 1208, 1280, 1396, 1468, 1507, 1615, 1619, 1658, 1677, 1689, 1701, 1716, 1740, 1748, 1779, 1815, 1842, 1863, 1886, 1905, 1948, 1949, 1977, 2071, 2113, 2175, 2203, 2210, 2216, 2244, 2260, 2294, 2360, 2429, 2471, 2484, 2492, 2643, 2698, 2777, 2783, 2789, 2794, 2796, 2875, 2878, 2917, 2938, 2991, 3004, 3028, 3069, 3116, 3161, 3193, 3297, 3312, 3387, 3440, 3444, 3455, 3461, 3510, 3515, 3536, 3593, 3631, 3673, 3742, 3765, 3788, 3874, 3923, 3971, 4070, 4090, 4143, 4147, 4165, 4172, 4220, 4279, 4287, 4387, 4407, 4451, 4458, 4476, 4527, 4613, 4620, 4626, 4664, 4724, 4784, 4812, 4858, 4876, 4926, 4942, 4990.

***Geni DE da GG e da LNN contemporaneamente* (*probabilità del* 95%)**

965, 2471, 2643, 3297, 3387, 4147, 4876.

CAPITOLO 5

***Geni DE solo da LNN* (*probabilità del* 95%)**

98, 110, 134, 140, 207, 223, 244, 319, 382, 402, 430, 480, 528, 533, 588, 628, 633, 653, 659, 672, 685, 709, 744, 753, 772, 805, 825, 879, 888, 896, 914, 947, 950, 971, 973, 1011, 1030, 1063, 1065, 1106, 1111, 1118, 1122, 1132, 1208, 1280, 1396, 1468, 1507, 1615, 1619, 1658, 1677, 1689, 1701, 1716, 1740, 1748, 1779, 1815, 1842, 1863, 1886, 1905, 1948, 1949, 1977, 2071, 2113, 2175, 2203, 2210, 2216, 2244, 2260, 2294, 2360, 2429, 2484, 2492, 2698, 2777, 2783, 2789, 2794, 2796, 2875, 2878, 2917, 2938, 2991, 3004, 3028, 3069, 3116, 3161, 3193, 3312, 3440, 3444, 3455, 3461, 3510, 3515, 3536, 3593, 3631, 3673, 3742, 3765, 3788, 3874, 3923, 3971, 4070, 4090, 4143, 4165, 4172, 4220, 4279, 4287, 4387, 4407, 4451, 4458, 4476, 4527, 4613, 4620, 4626, 4664, 4724, 4784, 4812, 4858, 4926, 4942, 4990.

***Geni DE da GG sui geni con media inferiore a 13000* (*probabilità del* 95%)**
4128.

***Geni DE da GG sui geni con media inferiore a 13000* (*probabilità del* 95%)**

328, 345, 376, 525, 1062, 1229, 1231, 1366, 1484, 1488, 1650, 1950, 2041, 2316, 2496, 2629, 2669, 2712, 2772, 3214, 3273, 3460, 3610, 3680, 3803, 3933, 4058, 4126, 4270

***Geni DE da LNN sui geni con media superiore a 20000* (*probabilità del* 95%)**

12, 51, 62, 96, 98, 110, 134, 207, 238, 315, 318, 332, 353, 381, 401, 402, 403, 429, 448, 453, 469, 479, 523, 527, 541, 586, 626, 635, 639, 651, 657, 667, 670, 683, 707, 745, 751, 770, 796, 803, 805, 806, 818, 823, 834, 870, 877, 886, 888, 894, 912, 943, 946, 953, 961, 967, 969, 986, 1026, 1095, 1107, 1114, 1115, 1118, 1128, 1227, 1274, 1294, 1307, 1320, 1348, 1390, 1404,

1411, 1416, 1462, 1550, 1563, 1608, 1612, 1637, 1643, 1651, 1694, 1724, 1733, 1741, 1769, 1775, 1776, 1799, 1807, 1834, 1855, 1877, 1893, 1896, 1902, 1939, 1940, 1966, 1968, 1982, 2054 , 2061, 2098, 2103, 2105, 2109, 2121, 2193, 2200, 2205, 2206, 2234, 2284, 2349, 2402, 2404, 2411, 2418, 2460, 2464, 2473, 2481, 2525, 2562, 2567, 2583, 2592, 2621, 2641, 2687, 2711, 2714, 2719, 2766, 2772, 2776, 2778, 2779, 2783, 2785, 2821, 2822, 2842, 2864, 2867, 2873, 2927, 2980, 2993, 3017, 3105, 3106, 3150, 3182, 3217, 3248, 3251, 3286, 3301, 3346, 3376, 3388, 3429, 3433, 3444, 3450, 3451, 3499, 3504, 3525, 3547, 3559, 3565, 3572, 3582, 3590, 3620, 3629, 3662, 3670, 3731, 3750, 3754, 3756, 3777, 3859, 3882, 3905, 3908, 3909, 3914, 3956, 3967, 4055, 4075, 4124, 4146, 4153, 4161, 4173, 4185, 4197, 4201, 4260, 4267 4268, 4316, 4368, 4381, 4432, 4438, 4456, 4463, 4507, 4519, 4554, 4599, 4605, 4643, 4698 4699, 4703, 4726, 4762, 4790, 4797, 4836, 4854, 4880, 4884, 4888, 4903, 4919, 4924, 4931, 4967, 4971

***Geni DE da LNN sui geni con media superiore a 20000* (*probabilità del* 95%)**

3, 4, 13, 23, 42, 58, 77, 78, 83, 84, 100, 108, 119, 121, 135, 149, 156, 157, 164, 192, 203, 217, 221, 227, 244, 253, 255, 258, 261, 262, 264, 282, 303, 325, 347, 378, 382, 395, 402, 414 , 446, 450, 459, 498 , 500, 514, 531, 538, 550, 602, 612, 623, 643, 666, 699, 705, 719, 748, 774, 799, 801, 820, 844, 851, 912, 944, 954, 990, 1005, 1014, 1034, 1083, 1087, 1091, 1116, 1120, 1142, 1145, 1152, 1170, 1176, 1213, 1237, 1283, 1286, 1297, 1299, 1325, 1441, 1485, 1497, 1532, 1568, 1569, 1596, 1611, 1642, 1643, 1664, 1672, 1673, 1708, 1715, 1718, 1722, 1723, 1734, 1776, 1794, 1798, 1808, 1831, 1853, 1856, 1885, 1889, 1890, 1913, 1922, 1942, 1957, 1960, 1965, 1971, 1982, 1994, 2054, 2086, 2114, 2119, 2144, 2145, 2148, 2174, 2181, 2184, 2211, 2217, 2232, 2272, 2278, 2315, 2324, 2331, 2349, 2389, 2414, 2427, 2428, 2436, 2437, 2450, 2453, 2464, 2468, 2527, 2560, 2561, 2574, 2599, 2603, 2614, 2663, 2665, 2671, 2678, 2679, 2680, 2711, 2741, 2765, 2778, 2782, 2808, 2852, 2858, 2861, 2866, 2882, 2884,

2922, 2935, 2937, 2940, 2965, 2966, 3009, 3027, 3032, 3042, 3055, 3061, 3070, 3111, 3116, 3134, 3144, 3154, 3205, 3240, 3246, 3256, 3322, 3346, 3349, 3362, 3364, 3365, 3369, 3376, 3380, 3394, 3403, 3404, 3409, 3414, 3433, 3441, 3452, 3462, 3473, 3492, 3508, 3509, 3512, 3518, 3520, 3524, 3526, 3534, 3537, 3539, 3541, 3557, 3561, 3579. 3583, 3584, 3603, 3604, 3606, 3608, 3619, 3635, 3658, 3664, 3671, 3672, 3675, 3676, 3690, 3693, 3701, 3711, 3730, 3733, 3743, 3745, 3751, 3758, 3760, 3771, 3778, 3779, 3794, 3797, 3802, 3804, 3805, 3808, 3821, 3828, 3830, 3831, 3845, 3848, 3876, 3877, 3878, 3892, 3900, 3926, 3935, 3950 , 3965, 3971, 3976, 4030, 4049, 4071, 4087, 4109, 4134, 4140, 4153, 4202, 4208, 4252, 4302, 4303, 4325,4334, 4336, 4348, 4349, 4351, 4359, 4372, 4390, 4400, 4404, 4423, 4424, 4433, 4451, 4482, 4489, 4532, 4547, 4552, 4558, 4568, 4596, 4598, 4650, 4656, 4669, 4677, 4688, 4709, 4715, 4718, 4747, 4759, 4762, 4765

A5.2 Codice *R* relativo alle analisi effettuate

A5.2.1 Creazione delle Curve di Andrews

```
tab1<-read.table("tab1A.txt",header=T)
tab2<-read.table("tab1B.txt")
tab<-rbind(tab1,tab2)
t1<-tab[10001:20000,]
data<-
cbind(t1$X1645,t1$X304,t1$X305,t1$X338,t1$X110,t1$X392,t1$X412,t1$
X415,
t1$X415,t1$X419,t1$DeSimone,t1$Fortina,t1$X15,t1$Fasanelli,t1$X13,
t1$X49,t1$X67,t1$X68,t1$X79,t1$X80,t1$X98,t1$X107,t1$X127,t1$X129,
t1$X181,t1$X185,t1$X186,t1$X188,t1$X257,t1$X258,t1$X286,t1$X287,t1
$X288,t1$X313,t1$X316,t1$X318,t1$X320,t1$X323,t1$X325,t1$X326,t1$X
329,t1$X332,t1$X334,t1$X337,t1$X340,t1$X346,t1$X369,t1$X378,t1$X38
2,t1$X18,t1$X21,t1$X22,t1$X32,t1$X39,t1$X47,t1$X119,t1$X128,t1$X14
9,t1$X191,t1$X195,t1$X245,t1$X328,t1$X331,t1$X344,t1$X345,t1$X354)

trig.fn <- function(x)
        {
                n <- length(x)
                ergebnis <- numeric(n)
                gerade <- (1:n)[(1:n) %% 2 == 0]
                ungerade <- (1:n)[(1:n) %% 2 != 0]
                ergebnis[ungerade] <- sin(x[ungerade])
                ergebnis[gerade] <- cos(x[gerade])
                ergebnis
        }

andrews<-function(x, tt, n.col)
{
        n.tt <- length(tt)
        andrews.x <- rep(x[1]/sqrt(2), n.tt)
        koef <- rep(1:(n.col - 1), rep(2, n.col - 1))[1:(n.col -
1)]
```

```
        tt.m<-as.vector(koef)*matrix(tt,nrow=n.col-
1,ncol=n.tt,byrow=T)
        tt.m<-apply(tt.m,2,FUN=trig.fn)
        andrews.x<-andrews.x+x[2:n.col]%*%tt.m
#        for(i in 1:n.tt) {
#               andrews.x[i] <- andrews.x[i] +
#                       sum(x[2:n.col] * trig.fn(tt[i] * koef))
#       }
        andrews.x
}

andrews.plot<-
function(m, stand = T,ug=0.25,og=.75,titel=deparse(substitute(m)))
{
        n.row <- dim(m)[1]
        n.col <- dim(m)[2]
        tt <- seq( - pi, pi, length = 100)
        if(!stand)
                m <- standard(m)
        max.andrews <- 0
        min.andrews <- 0
        result <- matrix(0, ncol = length(tt), nrow = n.row)
        for(i in 1:n.row) {
                result[i,  ] <- andrews(m[i,  ], tt, n.col)
        }
#       result<-matrix(apply(m,1,FUN=andrews,tt,n.col),
#               ncol=length(tt),nrow=n.row,byrow=T)
        range.andrews <- range(result)
        plot(0, xlim = c( - pi, pi), ylim = range.andrews,
                axes = F, ylab = "f(tt)",xlab = "tt", type = "n")
        axis(1)
        axis(2)
        for(i in 1:n.row)
                lines(tt, result[i,  ])
        result.range<-apply(result,1,range)
        range.ug<-quantile(result.range[1,],probs=ug)
```

```
        range.og<-quantile(result.range[2,],probs=og)

        vgl<-function(x,y) any(x<y[1] | x>y[2])
        ausserhalb<-apply(result,1,vgl,c(range.ug,range.og))
        abline(h=c(range.ug,range.og))
        text(par()$usr[2],range.ug,ug,cex=.8)
        text(par()$usr[2],range.og,og,cex=.8)
        print("Geni fuori dalle bande:")
        if(length(dimnames(m)[[1]])==0) names.m<-seq(n.row) else
                                names.m<-dimnames(m)[[1]]
        print(names.m[ausserhalb])
        for(i in (1:n.row)[ausserhalb]){
                wo.y<-max(
                        abs(
                        c(max(result[i,]),min(result[i,]))
                        ))
                if(wo.y==abs(min(result[i,])))
                        wo.y<-wo.y*sign(min(result[i,]))
                wo.x<-tt[result[i,]==wo.y]
                text(
                        wo.x,
                        wo.y+sign(wo.y)*0.02*diff(range.andrews),
                        names.m[i],cex=.9
                        )
        }
        title(paste("Andrews-Plot "))
        seq(n.row)[ausserhalb]
}
```

A5.2.2 Confronto tra VH321+ e VH321-

A5.2.2 Costruzione dei grafici di Figura 5.3, 5.4 e 5.5

```
Library(EBarrays)
checkCCV(data)
print(checkModel(data1, model = "gamma",nb = 100))
print(checkModel(data1, model = "lognormal",nb = 100))
```

A5.2.3 Stima dei parametri dei modelli GG e LNN e costruzione dei grafici in Figura 5.6 e 5.7.

```
pattern<-ebPatterns(c("1, 1, 1, 1, 1, 1, 1, 1, 1,1,1,
1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,
1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1",
"1, 1, 1, 1, 1, 1, 1, 1, 1,
1,1,1,1,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,
2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2"))
gg<- emfit(data1, family = "GG", pattern,num.iter = 15)
gg.post<- postprob(gg, data)
lnn<- emfit(data1, family = "LNN", pattern,num.iter = 15)
lnn.post<- postprob(lnn, data)
print(plotMarginal(gg,data1))
print(plotMarginal(lnn,data))
```

A5.2.4 Calcolo dei valori della Tabella 5.1.

```
#geni indicati DE da entrambi i modelli
sum(gg.post[,2]>0.5 & lnn.post[,2]>0.5)
#geni indicati EE da modello GG e DE da LNN
sum(gg.post [,1]>0.5 & lnn.post[,2]>0.5)
#geni indicati EE da entrambi i modelli
sum(gg.post [,1]>0.5 & lnn.post[,1]>0.5)
#geni indicati DE da modello GG e EE da LNN
sum(gg.post[,2]>0.5 & lnn.post[,1]>0.5)
```

A5.2.5 Costruzione del grafico in Figura 5.8.

```
#MEDIE SU DATASET DATA.RAW(R/G)
#medie per tutti i geni
data.raw<-data
n<-10000
medieTutti<-matrix(rep(0,n*2),ncol=2)
for(i in 1:n)
      { medieTutti[i,1]<-mean(data.raw[i,1:13])
      medieTutti[i,2]<-mean(data.raw[i,14:65])
      }
#geni indicati da entrambi i modelli GG e LNN
posteriorProb=0.5
indice05<-indici[gg.post [,2]>posteriorProb&lnn.post
[,2]>posteriorProb]
n<-length(indice05)
medie05<-matrix(rep(0,n*2),ncol=2)
for(i in 1:n)
      { medie05[i,1]<-mean(data.raw[indice05[i],1:13])
      medie05[i,2]<-mean(data.raw[indice05[i],14:65])
      }
posteriorProb=0.7
indice07<-indici [gg.post [,2]>posteriorProb&lnn.post
[,2]>posteriorProb]
n<-length(indice07)
medie07<-matrix(rep(0,n*2),ncol=2)
for(i in 1:n)
      { medie07[i,1]<-mean(data.raw[indice07[i],1:13])
      medie07[i,2]<-mean(data.raw[indice07[i],14:65])
      }
posteriorProb=0.9
indice09<-indici[gg.post[,2]>posteriorProb & lnn.post
[,2]>posteriorProb]
n<-length(indice09)
medie09<-matrix(rep(0,n*2),ncol=2)
      for(i in 1:n)
      { medie09[i,1]<-mean(data.raw[indice09[i],1:13])
```

```
      medie09[i,2]<-mean(data.raw[indice09[i],14:65])
      }
par(mfrow=c(2,2))
plot(medieTutti[,1],medieTutti[,2],xlab="Media VH321-",
ylab="Media VH321+",type="n",log="xy",main="GG e LNN")
points(medieTutti[-c(indice05,indice07,indice09),1],
medieTutti[-c(indice05,indice07,indice09),2],cex=.5)
abline(0,1)
points(medie05[,1],medie05[,2],col="blue",pch=16)
points(medie07[,1],medie07[,2],col="red",pch=16)
points(medie09[,1],medie09[,2],col="yellow",pch=16)
legend(10,14,legend=c("postprob>0.5","postprob>0.7",
"postprob>0.9"),col=c("blue","red","yellow"),cex=.7,
pch=c(16,16,16),bty="n")
#geni indicati dal modello GG
posteriorProb=0.5
indice05<-indici[gg. post [,2]>posteriorProb ]
n<-length(indice05)
medie05<-matrix(rep(0,n*2),ncol=2)
for(i in 1:n)
      { medie05[i,1]<-mean(data.raw[indice05[i],1:13)
      medie05[i,2]<-mean(data.raw[indice05[i],14:65])
      }
posteriorProb=0.7
indice07<-indici [gg.post[,2]>posteriorProb]
n<-length(indice07)
medie07<-matrix(rep(0,n*2),ncol=2)
for(i in 1:n)
      { medie07[i,1]<-mean(data.raw[indice07[i],1:13])
      medie07[i,2]<-mean(data.raw[indice07[i],14:65])
      }
posteriorProb=0.9
indice09<-indici [gg.post [,2]>posteriorProb ]
n<-length(indice09)
medie09<-matrix(rep(0,n*2),ncol=2)
      for(i in 1:n)
```

```
      { medie09[i,1]<-mean(data.raw[indice09[i],1:13])
      medie09[i,2]<-mean(data.raw[indice09[i],14:65])
      }
plot(medieTutti[,1],medieTutti[,2],xlab="Media VH321-",
ylab="Media VH321+",type="n",log="xy",main="GG")
points(medieTutti[-c(indice05,indice07,indice09),1],
medieTutti[-c(indice05,indice07,indice09),2],cex=.5)
abline(0,1)
points(medie05[,1],medie05[,2],col="blue",pch=16)
points(medie07[,1],medie07[,2],col="red",pch=16)
points(medie09[,1],medie09[,2],col="yellow",pch=16)
legend(10,14,legend=c("postprob>0.5","postprob>0.7",
"postprob>0.9"),col=c("blue","red","yellow"),cex=.7,
pch=c(16,16,16),bty="n")
#geni indicati dal modello LNN
posteriorProb=0.5
indice05<-indici[lnn.post[,2]>posteriorProb]
n<-length(indice05)
medie05<-matrix(rep(0,n*2),ncol=2)
for(i in 1:n)
      { medie05[i,1]<-mean(data.raw[indice05[i],1:13])
      medie05[i,2]<-mean(data.raw[indice05[i],14:65])
      }
posteriorProb=0.7
indice07<-indici$V3[lnn.post [,2]>posteriorProb]
n<-length(indice07)
medie07<-matrix(rep(0,n*2),ncol=2)
for(i in 1:n)
      { medie07[i,1]<-mean(data.raw[indice07[i],1:13])
      medie07[i,2]<-mean(data.raw[indice07[i],14:65])
      }
posteriorProb=0.9
indice09<-indici [lnn.raw.post[,2]>posteriorProb ]
n<-length(indice09)
medie09<-matrix(rep(0,n*2),ncol=2)
for(i in 1:n)
```

```
      { medie09[i,1]<-mean(data.raw[indice09[i],1:13])
      medie09[i,2]<-mean(data.raw[indice09[i],14:65])
      }
plot(medieTutti[,1],medieTutti[,2],xlab="Media VH321-",
ylab="Media VH321+",type="n",log="xy",main="LNN")
points(medieTutti[-c(indice05,indice07,indice09),1],
medieTutti[-c(indice05,indice07,indice09),2],cex=.5)
abline(0,1)
points(medie05[,1],medie05[,2],col="blue",pch=16)
points(medie07[,1],medie07[,2],col="red",pch=16)
points(medie09[,1],medie09[,2],col="yellow",pch=16)
legend(10,14,legend=c("postprob>0.5","postprob>0.7",
"postprob>0.9"),col=c("blue","red","yellow"),cex=.7,
pch=c(16,16,16),bty="n")
```

A5.2.6 Costruzione del grafico in Figura 5.9.

```
#MEDIE SU DATASET DATA(log(R/G))
#medie per tutti i geni
n<-10000
medieTutti<-matrix(rep(0,n*2),ncol=2)
for(i in 1:n)
{ medieTutti[i,1]<-mean(data[i,1:13])
medieTutti[i,2]<-mean(data[i,14:65])
}
#geni indicati da entrambi i modelli GG e LNN
posteriorProb=0.5
indice05<-indici[gg.post[,2]>posteriorProb
& lnn.post[,2]>posteriorProb]
n<-10000
medie05<-matrix(rep(0,n*2),ncol=2)
for(i in 1:n)
{ medie05[i,1]<-mean(data[indice05[i],1:13])
medie05[i,2]<-mean(data[indice05[i],14:65])
}
```

```
posteriorProb=0.7
indice07<-indici[gg.post[,2]>posteriorProb
& lnn.post[,2]>posteriorProb]
medie07<-matrix(rep(0,n*2),ncol=2)
for(i in 1:n)
{ medie07[i,1]<-mean(data[indice07[i],1:13])
medie07[i,2]<-mean(data[indice07[i],14:65])
}
posteriorProb=0.9
indice09<-indici[gg.post[,2]>posteriorProb
& lnn.post[,2]>posteriorProb]
medie09<-matrix(rep(0,n*2),ncol=2)
for(i in 1:n)
{ medie09[i,1]<-mean(data[indice09[i],1:13])
medie09[i,2]<-mean(data[indice09[i],14:65])
}
par(mfrow=c(2,2))
plot(medieTutti[,1],medieTutti[,2],xlab="Media VH321-",
ylab="Media VH321+",type="n",main="GG e LNN")
points(medieTutti[-c(indice05,indice07,indice09),1],
medieTutti[-c(indice05,indice07,indice09),2],cex=.5)
abline(0,1)
points(medie05[,1],medie05[,2],col="blue",pch=16)
points(medie07[,1],medie07[,2],col="red",pch=16)
points(medie09[,1],medie09[,2],col="yellow",pch=16)
legend(5000,25000,legend=c("postprob>0.5","postprob>0.7",
"postprob>0.9"),col=c("blue","red","yellow"),cex=.7,
pch=c(16,16,16),bty="n")
#geni indicati dal modello GG
posteriorProb=0.5
indice05<-indici[gg.post[,2]>posteriorProb]
medie05<-matrix(rep(0,n*2),ncol=2)
for(i in 1:n)
{ medie05[i,1]<-mean(data[indice05[i],1:13])
medie05[i,2]<-mean(data[indice05[i],14:65])
}
```

```
posteriorProb=0.7
indice07<-indici[gg.post[,2]>posteriorProb]
medie07<-matrix(rep(0,n*2),ncol=2)
for(i in 1:n)
{ medie07[i,1]<-mean(data[indice07[i],1:13])
medie07[i,2]<-mean(data[indice07[i],14:65])
}
posteriorProb=0.9
indice09<-indici[gg.post[,2]>posteriorProb]
medie09<-matrix(rep(0,n*2),ncol=2)
for(i in 1:n)
{ medie09[i,1]<-mean(data[indice09[i],1:13])
medie09[i,2]<-mean(data[indice09[i],14:65])
}
plot(medieTutti[,1],medieTutti[,2],xlab="Media VH321-",
ylab="Media VH321+",type="n",main="GG")
points(medieTutti[-c(indice05,indice07,indice09),1],
medieTutti[-c(indice05,indice07,indice09),2],cex=.5)
abline(0,1)
points(medie05[,1],medie05[,2],col="blue",pch=16)
points(medie07[,1],medie07[,2],col="red",pch=16)
points(medie09[,1],medie09[,2],col="yellow",pch=16)
legend(5000,25000,legend=c("postprob>0.5","postprob>0.7",
"postprob>0.9"),col=c("blue","red","yellow"),cex=.7,
pch=c(16,16,16),bty="n")
#geni indicati dal modello LNN
posteriorProb=0.5
indice05<-indici[lnn.post[,2]>posteriorProb]
medie05<-matrix(rep(0,n*2),ncol=2)
for(i in 1:n)
{ medie05[i,1]<-mean(data[indice05[i],1:13])
medie05[i,2]<-mean(data[indice05[i],14:65])
}
posteriorProb=0.7
indice07<-indici[lnn.post[,2]>posteriorProb]
medie07<-matrix(rep(0,n*2),ncol=2)
```

```
for(i in 1:n)
{ medie07[i,1]<-mean(data[indice07[i],1:13])
medie07[i,2]<-mean(data[indice07[i],14:65])
}
posteriorProb=0.9
indice09<-indici[lnn.post[,2]>posteriorProb]
medie09<-matrix(rep(0,n*2),ncol=2)
for(i in 1:n)
{ medie09[i,1]<-mean(data[indice09[i],1:13])
medie09[i,2]<-mean(data[indice09[i],14:65])
}
plot(medieTutti[,1],medieTutti[,2],xlab="Media VH321-",
ylab="Media VH321+",type="n",main="LNN")
points(medieTutti[-c(indice05,indice07,indice09),1],
medieTutti[-c(indice05,indice07,indice09),2],cex=.5)
abline(0,1)
points(medie05[,1],medie05[,2],col="blue",pch=16)
points(medie07[,1],medie07[,2],col="red",pch=16)
points(medie09[,1],medie09[,2],col="yellow",pch=16)
legend(5000,25000,legend=c("postprob>0.5","postprob>0.7",
"postprob>0.9"),col=c("blue","red","yellow"),cex=.7,
pch=c(16,16,16),bty="n")
```

A5.2.7 Costruzione del grafico degli *odds* in Figura 5.10.

```
#Costruzione curve di livello odds del modello Gamma-Gamma
#su diagramma di dispersione tra medie di espressioni geni nelle
#leucemie ALL e medie di espressione geni nelle leucemie AML
#MEDIE SU DATASET DATA
#medie per tutti i geni
ngeni<-10000
data1<-log(data)
medieTutti<-matrix(rep(0,ngeni*2),ncol=2)
for(i in 1:ngeni)
{ medieTutti[i,1]<-mean(data[i,1:13])
```

```
medieTutti[i,2]<-mean(data[i,14:65])
}
par(mfrow=c(1,1))
plot(medieTutti[,1],medieTutti[,2],xlab="Media VH321+",
ylab="Media VH321-",type="n",log="xy",
main="Curve di livello odds modello GG")
points(medieTutti[,1],medieTutti[,2],cex=.5)
abline(0,1)
minimo<-log(min(medieTutti))
massimo<-log(max(medieTutti))
#variabile lunghezza
#per definire il dettaglio delle curve di livello
lunghezza=100
sequenza<-seq(minimo,massimo,le=lunghezza)
matrice<-matrix(rep(0,lunghezza*lunghezza),ncol=lunghezza)
ncond1=13
ncond2=52
for(i in 1:lunghezza)
{ for(j in 1:lunghezza)
{ p=postprob(gg,
matrix(c(rep(exp(sequenza[i]),ncond1),
rep(exp(sequenza[j]),ncond2)),nrow=1))
matrice[i,j]=p[2]/p[1]
}
}
contour(exp(sequenza),exp(sequenza),matrice,levels=c(1,10,100),
add=T,drawlabels=FALSE)
#punti che indicano geni il cui odds(probabilitµa a posteriori DE
#diviso probabilitµa a posteriori EE) µe maggiore di 1
odds.grande<-rep(0,ngeni)
for(i in 1:ngeni)
{ odds.grande[i]<-(gg.post)[i,2]/(gg.post)[i,1]
}
points((medieTutti[odds.grande>1,1]),
(medieTutti[odds.grande>1,2]),pch=19,cex=.7,col="yellow")
```

Capitolo 6

Considerazioni conclusive

Nel corso dell'elaborato sono state analizzate le espressioni geniche di 65 pazienti stratificati in base alla variazione del gene VH321. L'analisi era finalizzata alla valutazione dei metodi bayesiani empirici parametrici nell'identificare *pattern* di espressione genica in due gruppi sbilanciati. Nella fase preliminare è stata considerata una caratteristica peculiare dei dati di *microarray*, ossia la forte presenza di valori mancanti. Nel caso particolare, il *dataset* a nostra disposizione non possedeva tale caratteristica, ma si può ipotizzare che i dati mancanti fossero già stati imputati in precedenza dal C.R.O, che ci ha fornito i dati. Per valutare 4 tecniche di imputazione di valori mancanti diverse, si sono eliminate il 10% delle osservazioni in maniera casuale, e formati quindi i 4 insiemi di dati corrispondenti alle tecniche.

I *dataset* da analizzare, a questo punto dell'elaborato, sono 5: l'insieme di dati in cui i valori mancanti sono stati imputati con la media generale, l'insieme di dati imputati con la media di gruppo, l'insieme di dati in cui si è applicato l'algoritmo di imputazione SVD, l'insieme di dati in cui si è applicato l'algoritmo *knn* ed, infine, l'insieme di dati reali. Le tecniche di imputazione sono state d'dapprima confrontate attraverso il calcolo degli errore RMS per ciascuna tecnica. Da questo primo confronto, non sono state rilevate differenze significative tra i diversi RMS. L'attenzione è stata poi spostata nel valutare l'influenza delle tecniche di imputazione nella creazione di *pattern* di espressione genica. Per fare questo si sono utilizzati i metodi empirici bayesiani, prima per stimare i parametri dei modelli GG e LNN di 5 *trainig set* corrispondenti ai 5 insiemi di dati, per poi testare i modelli nei 5 *test set* corrispondenti. Sono state create quindi le 4 matrici

di confusione che confrontano le previsioni sui dati reali con quelle sui dati imputati. Dal confronto delle 4 matrici di confusione, è stata evidenziata una precisione estrema da parte dell'algoritmo *knn*. I risultato ottenuta risulta soddisfacente, in quanto la tecnica di imputazione *knn* risulta la migliore sia per problemi di classificazione, sia per problemi discriminanti sia per la creazione di *pattern* di espressione.

Sono stati applicati poi i metodi empirici bayesiani sul *dataset* reale per l'individuazione di geni differenzialmente espressi nei due gruppi non bilanciati oggetto di studio. Si evidenzia da questa analisi la migliore accuratezza con cui il modello GG individua dati differenzialmente espressi. Tale accuratezza è comprovata dal confronto dei profili genetici dei geni identificati come differenzialmente espressi attraverso le curve di Andrews, precedentemente utilizzate nell'analisi esplorativa. I geni differenzialmente espressi individuati da GG hanno effettivamente profili genetici molto differenti nei due gruppi.

Il modello LNN identifica un maggior numero di geni differenzialmente espressi rispetto a GG, ma, dal confronto dei loro profili, le differenze nei due gruppi non sono così evidenti.

I grafici creati per le analisi appena descritte, hanno evidenziato la presenza di una clusterizzazione naturale dei dati. In particolare, si nota che in ogni grafico ci sono 2 sottogruppi di geni. La prima ipotesi da valutare, a questo punto, è la corrispondenza della clusterizzazione naturale con la suddivisione nei due sottoruppi in analisi. Dai grafici per la valutazione della costanza del coefficiente di variazione per i soggetti del gruppo VH321+ e VH321-, mantengono le stesse caratteristiche del grafico su tutti i dati: esiste ancora la clusterizzazione che comporta un andamento anomalo della curva del coefficiente di variazione.

Il *dataset* è stato quindi suddiviso sulla base della media di espressione di ciascun gene: sono stati quindi applicati i metodi empirici bayesiani per l'individuazione di geni differenzialmente espressi nelle due partizioni del *dataset*. Questa suddivisione ha prodotto migliori risultati rispetto ai precedenti: l'ipotesi del coefficiente di variazione costante non è ancora soddisfatta, ma il suo andamento

è quello previsto. Il numero di geni totali identificati come differenzialmente espressi aumenta, facendoci ipotizzare una precedente perdita di informazioni dovuta alla clusterizzazione naturale.

Tutte le analisi condotte nell'elaborato sono state compiute grazie alla libreria di R *EBarrays* che offre la possibilità di modellare dati attraverso metodi bayesiani empirici parametrici, con i modelli Gamma-Gamma e LogNormale-Normale.

Tutte le analisi proposte hanno evidenziato le buone capacità dei metodi empirici bayesiani nell'identificare *pattern* di espressione genica in gruppi altamente sbilanciati.

CAPITOLO 6

Ringraziamenti

A conclusione del mio corso di studi desidero esprimere la mia gratitudine a quanti, in questi anni, mi hanno aiutato e sostenuto nella realizzazione di quello che all'inizio sembrava essere un desiderio lontano.

In particolare ringrazio la mia relatrice, professoressa Monica Chiogna, per la disponibilità e cortesia dimostratami in questo periodo di tesi e per aver messo a mio servizio la sua professionalità e le sue competenze.

Ringrazio i miei genitori e le mie sorelle che in questi anni non hanno mancato di incoraggiarmi, sostenermi e consigliarmi, oltre che di assumersi gli oneri della mia istruzione.

Ringrazio il Dr. Gattei e la Dr.ssa Marconi per avermi reso disponibile i dati di una casistica reale già precedentemente analizzata dal C.R.O dia Aviano.

Ringrazio il Dott. Toffoli, che nel corso dell'ultimo periodo universitario e nel periodo di tesi, mi ha dato l'opportunità, con una borsa di studio, di entrare a far parte di uno staff di ricerca nel reparto di farmacologia sperimentale e clinica del C.R.O di Aviano, offrendomi così l'opportunità di crescita professionale e, nello stesso tempo di approfondimento degli argomenti relativi alla tesi.

Ringrazio di cuore Emiliano che in tutti questi anni ha saputo sopportare e capire i miei cambiamenti d'umore, permettendomi di dedicarmi allo studio con la consapevolezza di avere accanto una persona in grado di comprendere e rispettare i miei tempi.

Ringrazio la famiglia Marra per avermi appoggiata in ogni momento ed per avermi offerto la loro ospitalità nel primo periodo della mia attività di borsista al C.R.O e al loro continuo sostegno nel corso di questi anni.

Ringrazio l'Associazione Culturale "Mu" di avermi reso disponibile l'apparecchiatura informatica per la realizzazione dell'elaborato rendendo più snello e veloce la sua realizzazione.

RINGRAZIAMENTI

Ringrazio tutte le mie colleghe del reparto di farmacologia sperimentale e clinica per essersi rese disponibile a correggere e rivedere la parte biologica dell'elaborato.

Ricordo infine con tanto affetto e gratitudine tutti coloro che in questi anni mi hanno fatto dono della loro amicizia rendendo gradevole e felice la mia permanenza a Padova ed ogni ricordo ad essa legato. Con alcune di queste persone ho avuto modo di studiare, di apprendere e di approfondire le mie conoscenze, ricevendo gli stimoli per crescere e maturare anche intellettualmente: a tutti loro va il ringraziamento più sentito.

Laura Gavagnin

Bibliografia

[1] Chen Y., Dougherty E.R., Bittner M.L. (1997). *Ratio-based decision and the quantitative analysis od cDNA microarray images*. J. Biomedical Optics 2(4), 364-374.

[2] Dudoit S., Yang T.H., Callow M.J., Speed T.P. (2000). *Statistical methods for identifyng differentially expressed genes in replicated cDNA microarray experiments*. Statistica Sinica 12, 111-139.

[3] Tusher V.G, Tibshirani R., Chu G. (2001) *Significance analysis of microarray appried ionizing radiation response.* PNAS, 9 (98): 5116-5121.

[4] Baldi P., Long A.D. (2001). *A Bayesian framework for the anallysis of microarray ecpression data: regularized t-test and statistical inferences of gene changes*. Bioinformatics, 17, 509-519.

[5] Newton M.A., Kendziorski C.M., Richmond C.S., Blattner F.R., Tsui K.W. (2001). *On differential variability of expression ratios: improving statisticalinference about gene expression changes from microarray data.* Journal of Computational Biology 8, 37-52.

[6] Kendziorski C.M., Newtyon M.A., Lan H., Gould M.N. (2003). *On parametric empirical Bayes methods for comparing multiple groups using replicated gene expression profiles*. Statistics in medicine 22, 3899-3914.

[7] Efron B., Tibshirani R. (2002). *Microarray, Empirical Bayes Methods and False Discovery Rates*. Genet. Epidemiol. 2002 Jun; 23 (1): 70-86.

BIBLIOGRAFIA

[8] Kendzioeski C., Sarkar D., Chen M, Newton M. (2005). *ParametricEmpirical Bayes Methods for Microarray Data.* Documentazione Vignette della libreria *Ebarrays* di R.

[9] Lise M. (2005): *Metodi Bayesiani empirici per l'dentificazione di geni differenzialmente espressi.* Tesi di Laurea (Facoltà di Statistica, Universit`a di Padova), relatore prof.Chiogna M.

[10] Chiogna M., Massa M.S., Romualdi C. *Effect of normalization on defecting differentially expressed genes with cDNA microarray experiments.*

[11] Hastie T., Tibshirani R., Sherlock G., Eisen M.Alter O., Botstein D. Browm P. *Imputation missing data for gene expression arrays.* (2000)

[12] Garlet M. (2003). *Analisi dei livelli di espressione genica per lo studio delleleucemie.* Tesi di Laurea (Facoltà di Statistica, Universit`a di Padova), relatore prof.Chiogna M.

[13] Newton M.A., Wang P., Kendziorski C.M. (2006). *Hierargical mixture model for expression profiles*.

[14] Parmigiani G., Garrett E.S., Irizarry R.A., Zeger S.L. (2003). *The Analysisof Gene Expression Data: Methods and Software.* Springer.

[15] Speed T. (2003). *Statistical analysis of gene expression microarray data.*Chapman & Hall/CRC.

[16] Tseng G.C. et al. [2001]. *Issues in cDNA microarray analysis:quality filtering,channel normalization, models of variations and assesment of gene effects.* Nucleic Acids Res., 29(12):2549-2557.

[17] Alberts B., Bray D., Hopkin K., Johnson A., Lewis J., Raff M., Roberts K, Walter P. (2005). *L'essenziale di biologia molecolare della cellula.* Zanichelli, seconda edizione.

[18] Benjamini Y., Hicheberg Y. (1995). *Controlling the false discovery rate in multiple hypothesis testing under dependency*. Annals of Statistics.

[19] Broet P, Richardson S., Radvanyi F. (2002). *Bayesian hierarchical model for identifying changes in gene expression from microarray experiments*. Journal of Computational Biology, 9, 619-683.

[20] Buck M.J., Kieb J.D. (2004). *ChIP-chip: considerations for the design, analysis, and application of genime-wide chromatin immunoprecipitation experiment*. Genomics, 83, 349-360.

[21] Cobb G.W. (1998). *Introduction to Design of Experiments.* Springer, New York.

[22] Ishwaran H., Rao J.S. (2003). *Detecting differential expression in microarray using Bayesian model selection.* JASA, 98, 438-455.

[23] Kerr M.K., Churchill G.A. (2001). *Experimental design for gene expression microarrays*. Biostatistics, 2, 183-201.

[24] Kerr M.K., Martin M., Churchill G.A. (2000). *Analysis of variance for gene expression microarray data.* Journal of Computational Biology, 7, 819-837.

BIBLIOGRAFIA

[25] Long A.D., Mangalan H.J. Chab B.Y., Trolleri L., Hatfield G.W., Badi P. (2001). *Global gene expression profiling in Escherichia coli K 12: improved statistical inference from DNA microarray data using analysis of variance and Bayesian statistical framework.* Journal of Biol. Hem., 276, 19937-19944.

[26] Lonnstedt I., Britton T. (2005). *Two hierarchical Bayes models for cDNA microarray gene expression.* Biostatistics, 1, 1, 1-23.

[27] Lonnstedt I., Speed T.P. (2002). *Replicated microarray data.* Statistica sinica, 12, 31-46.

[28] Parmigiani G., Garret E.S., Irizarry R.A., Zeger S.L. (2003). *The Analysis of Gene expression data: method and software.* Spinger, New York.

[29] Qui X., Klebanov L., Yakovlev A. (2005). *Correlation between gene expression levels and limitations af the empirical Bayes methodology for finding differentially expressed genes.* Statistical Applications in Genetics and Molecolar Biology, 3(1), article 3.

[30] Smyth G.K (2004). *Linear model and emoirical Bayes methods for assessing differential expression in microarray experiments.* Statistical Applications in Genetics and Molecular Biology, 3 (1).

[31] van der Lann M.J., Dudoit S., Pollard K.S. (2004). *Multiple testing. Part II. Step-down procedures for control af the family-wise error rate.* Statistical Applications un Genetics and Molecular Biology, 3(1), article 14.

[32] van der Lann M.J., Dudoit S., Pollard K.S. (2004). *Augmentations procedures for control of generalized family-wise error rate and tail probabilities*

for the proportions of false positive. Statistical Applications on Genetics and Molecular Biology, 3(1), article 15.

[33] Gottardo R. (2003). *Statistical analysis of micrarray data: a Bayesian approach*. Biostatistics, 4 (4), 597-620.

[34] Lonnstedt I.M., Rimini R., Nilson P. (2004). *Bayesian microarray one-way anova and grouping cell lines by equal expression levels*. Department of Mathematics, Uppsala Unicersity.

[35] Medvedovic M , Yeung K.Y., Bumgerner R.E (2004). *Bayesian mixture model based clustering of replicated microarray data*. Bioinformatics, 20 (8), 1222-1232.

[36] Wang, P. and Newton, M.A. (2005). *Robustness of EBarrays to one form of dependence.* Technical Report #1114, UW Department of Statistics.

[37] Yuan, M. and Kendziorski C. *A unified approach for simultaneous gene clustering and differential expression.* Biometrics.

[38] Erron B., Tibshirani R., Storey J.D., Tusher V. (2001). *Empirical Bayes Analysis of a Microarray experiment.* Journal pf th American Statistical Association, 96 (456), 1151-1160.

[39] Bomben R, Dal Bo M, Capello D, Benedetti D, Marconi D, Zucchetto A, Forconi F, Maffei R, Ghia EM, Laurenti L, Bulian P, Del Principe MI, Palermo G, Thorselius M, Degan M, Campanini R, Guarini A, Del Poeta G, Rosenquist R, Efremov DG, Marasca R, Foa R, Gaidano G, Gattei V. *Comprehensive characterization of IGHV3-21-expressing B-cell chronic lymphocytic leukemia: an Italian multicenter study.* Blood. 2007 Apr 1;109(7):2989-98.

BIBLIOGRAFIA

[40] Zucchetto A, Sonego P, Degan M, Bomben R, Dal Bo M, Russo S, Attadia V, Rupolo M, Buccisano F, Del Principe MI, Del Poeta G, Pucillo C, Colombatti A, Campanini R, Gattei V. *Signature of B-CLL with different prognosis by Shrunken centroids of surface antigen expression profiling*. J Cell Physiol. 2005 Jul;204(1):113-23.

[41] Bomben R, Dal Bo M, Zucchetto A, Zaina E, Nanni P, Sonego P, Del Poeta G, Degan M, Gattei V. *Mutational status of IgV(H) genes in B-cell chronic lymphocytic leukemia and prognosis: percent mutations or antigen-driven selection?* Leukemia. 2005 Aug;19(8):1490-2.

[42] Degan M, Rupolo M, Bo MD, Stefanon A, Bomben R, Zucchetto A, Canton E, Berretta M, Nanni P, Steffan A, Ballerini PF, Damiani D, Pucillo C, Attadia V, Colombatti A, Gattei V. *Mutational status of IgVH genes consistent with antigen-driven selection but not percent of mutations has prognostic impact in B-cell chronic lymphocytic leukemia.* Clin Lymphoma. 2004 Sep;5(2):123-6.

[43] Gottardo R., Raftery A.E., Yeung K.Y., Bumgerner R.E. (2005) *Bayesian Robust Inference for Differentian Gene Expression in microarray multiple samples*.

[44] Galfrè S.G. (2004). *SNP e Microarray. Strategie di Disegno Sperimentale e di Analisi dei dati.*

[45] Alberto R. (2004). *Studio delle proprietà di sequenze di DNA.*

Siti utili

http://50annidna.scienze.unipd.it

http://www.bioinformatica.unito.it

www.darwinweb.it

www.ecplanet.com

www.bioconductor.org

www.ematologia.unito.it

http://ihome.cuhk.edu.hk/~b400559/arraysoft_rpackages.html

www.leukaemia.org.au

www.science.ngfn.de

http://www.swisscancer.ch/broschueren

www.biomedicaltechnologies.com

http://italiasalute.leonardo.it/Centro_Malattie.asp?Sezione=Leucemia

www.katamed.it