POLITECNICO DI TORINO
Facoltà di Ingegneria dell'Informazione
Corso di Laurea in Ingegneria Informatica

Tesi di Laurea

Classificazione di dati genetici

Relatore
prof. Elena Baralis

Candidato
Alessandro Fiori

Novembre 2006

Sommario

Capitolo 1
Introduzione

La bioinformatica è una disciplina che unisce analisi di carattere biologico con l'utilizzo come mezzo di supporto dell'informatica. Negli anni l'aspetto informatico è cresciuto ed è diventato di vitale importanza per la biologia e specialmente per affrontare analisi complesse sia in termini di complessità che di mole di dati utilizzati, analisi che introducono nuove sfide tecnologiche proprio nell'ambito dell'informatica.
In ambito genetico negli ultimi anni si è sviluppata una tecnologia che ha rivoluzionato l'approccio sperimentale : i *microarray*, vetrini con decine di migliaia di sonde costituite da materiale genico che, opportunamente trattato, produce una grande quantità di dati di espressione genetica. Dalla biologia sappiamo che questi valori determinano il comportamento dei geni e dell'organismo, come anche la predisposizione o lo sviluppo di malattie quali i tumori, anche se le relazioni tra queste informazioni sono ancora da scoprire e attualmente sono oggetto di ricerca a livello mondiale.
Si capisce quindi che un'analisi accurata e ben strutturata di questi dati è molto importante per la ricerca e può portare a notevoli benefici per la salute e la sopravvivenza dell'uomo.
Questo studio si focalizza sull'aspetto informatico dell'analisi dei dati derivanti dai microarray e in particolar modo sugli aspetti di data mining collegati alla raccolta di una quantità di dati così elevata.
I passi di analisi di dati affrontati sono il filtraggio dei dati, la costruzione di modelli di classificazione e la determinazione dei geni (features) che caratterizzano le varie categorie dei pazienti.
Per questo studio si utilizzeranno due dataset differenti tra loro per provenienza e per caratteristiche intrinseche. Infatti un dataset deriva da analisi di tessuti del colon e le tipologie di pazienti sono due: pazienti sani e pazienti con tumore. L'altro dataset, invece, si basa su osservazioni di tessuti della prostata e individua tre categorie di pazienti: quelli sani, quelli con tumore maligno e infine quelli con tumore benigno. Pertanto gli esperimenti che si svolgeranno saranno scelti in base al tipo di dataset utilizzato e al problema che viene determinato dalle analisi preliminari.
È importante sottolineare che i dati su cui sono stati svolti gli esperimenti non sono dati simulati, ma sono stati forniti da esperti del settore che avevano effettiva necessità di compiere le analisi svolte in questa tesi.

Il filtraggio dei dati è uno dei passi iniziali nell'analisi di questi dati, ma non per questo meno importante. Infatti i filtri devono essere applicati in modo molto accurato cercando di evitare l'eliminazione di dati che contengono molta informazione e che potrebbero essere utili per le analisi future.
La creazione di modelli di classificazione è uno degli aspetti più studiati e sviluppati del data mining. La classificazione è infatti un problema molto complesso sia per la difficoltà intrinseca della stessa, sia per la quantità di dati che bisogna analizzare per la creazione dei modelli. Negli anni sono stati sviluppati vari metodi di classificazione. In questo studio verranno analizzati quelli più adatti per la tipologia di dato che è oggetto di studio. I dati genetici da microarray, infatti, presentano delle caratteristiche che non permettono di utilizzare le tecniche classiche di data mining. Quest'ultime prevedono un numero limitato di *features* e numerosi *samples*. Al contrario, i dati analizzati comprendono un numero relativamente ristretto di pazienti e molte migliaia di geni per ogni paziente.
Per gli esperimenti si utilizzano dei tool che sono stati creati per lo studio di dati biologi, integrandoli dove sorge la necessità con programmi creati ad hoc. Questi tool implementano alcuni degli algoritmi di classificazione maggiormente usati allo stato attuale dell'arte per la costruzione dei modelli di classificazione, oltre a fornire la possibilità di selezionare gli attributi, quindi i geni, che secondo alcune valutazioni statistiche risultano migliori per il processo di classificazione.
Un obiettivo di questo studio è anche la determinazione dei geni che meglio rappresentano le varie classi dei campioni/pazienti. Verrà analizzata l'influenza che queste selezioni possono avere in termini di prestazioni e di complessità computazionale per gli algoritmi di classificazione.
Si determineranno quindi sia modelli di classificazione sia i relativi geni che potranno essere utilizzati nelle analisi future di campioni sugli stessi tipi di tessuto.

Capitolo 2
Bioinformatica

Il problema fondamentale che si trova ad affrontare oggi la ricerca biologica è quello di disporre di una quantità immensa e crescente di dati, che viene prodotta empiricamente dagli studi chimici, biochimici e genetico-molecolari, e che però non rappresenta una effettiva conoscenza sul modo di funzionare dei sistemi biologici [1-29].
In altre parole, le informazioni sulla localizzazione dei geni, sulle sequenze, sui tipi di proteine e le loro strutture mancano spesso di un significato funzionale; cioè non si sa come agiscono nella cellula.
Per gestire questa massa di informazioni biologiche in continuo aumento e per analizzarle in modo da trovare delle relazione che facciano emergere qualche nuovo principio organizzativo o funzionale della vita è nata la bioinformatica.
In realtà, storicamente, la bioinformatica nasce con il problema di utilizzare il computer per analizzare le sequenze dei geni e delle proteine, ma il settore si è progressivamente ampliato per comprenderne appunto la gestione, l'elaborazione, l'analisi e la visualizzazione di grandi quantità di dati prodotti dalle ricerche genomica, proteomica, controllo di farmaci e chimica combinatoria.
La bioinformatica include quindi l'integrazione e l'esplorazione delle sempre più estese banche dati di interesse biologico [5-29].
Le piattaforme bioinformatiche sono in genere costituite da un sistema di database interno, banche dati e collegamenti all'esterno (pubblici o privati), una serie di software che definisce gli obiettivi biologici di interesse per il ricercatore e degli algoritmi per esplorare e correlare le informazioni.
L’aspettativa dei bioinformatici è quella che emerga da questo lavoro il progetto della vita; cioè che vengano alla luce nuovi modelli o principi esplicativi in grado di correlare funzionalmente i geni con i percorsi metabolici, dare un significato evolutivo al confronto tra geni e i pool genici di diverse specie, catturare la logica che governa l'assemblaggio tridimensionale delle proteine, e quindi predire la funzione di geni e proteine integrando i diversi tipi di informazioni disponibili.
In questo modo si potrebbe spiegare quali forze guidano lo sviluppo di un organismo complesso come l'uomo a partire da una singola cellula fecondata; nonché rispondere a molte fondamentali domande che riguardano l'evoluzione della vita.

Uno degli obiettivi più importanti è decifrare l'impianto regolativo delle reti geniche e metaboliche che controllano i vari aspetti del fenotipo, incluse le malattie.
Infatti è chiaro che non tutta l'informazione biologica è racchiusa nelle sequenze di Dna che codificano per le proteine, né la struttura funzionale di una proteina è stabilita solo a livello della sequenza di aminoacidi.
Esistono delle dinamiche relazionali che si esprimono a diversi livelli dell'organizzazione cellulare e che ancora non sono conosciute.
L'approccio bioinformatico consiste nello sviluppo di modelli matematici e algoritmi che consentano di estrarre dai dati empirici le informazioni rilevanti, e trarne predizioni significative da sottoporre al controllo sperimentale [3].
La bioinformatica sta aprendo opportunità fantastiche non solo per i biologi, ma anche per informatici, ingegneri e fisici che lavorano nel campo della modellizzazione di sistemi complessi. Opportunità da sfruttare perché una valanga di finanziamenti sta investendo il settore, che è sempre più considerato cruciale nella ricerca genomica e post-genomica fondamentale e applicata.
Considerando come la ricerca clinica, soprattutto a livello dei sistemi di raccolta di informazioni, sta cambiando con l'accumularsi delle conoscenze genomiche, strutturali e funzionali, non è difficile prevedere che la bioinformatica assumerà presto un peso rilevante anche in medicina.
Le informazioni sulle sequenze di Dna, e le annotazioni riguardanti le loro funzioni, diventeranno sempre più frequentemente oggetto di riflessione da parte del medico alla ricerca di una diagnosi e di un trattamento. Quindi gli algoritmi sviluppati per la ricerca nell'ambito della bioinformatica presto diventeranno parte integrante dei sistemi clinici di raccolta ed elaborazione delle informazioni.
Inutile dire che l'industria farmaceutica è particolarmente interessata agli sviluppi della bioinformatica, dato che il problema di dare un senso alle sequenze e alle strutture proteiche è pregiudiziale per lo sviluppo di farmaci, vaccini, marcatori diagnostici e proteine terapeutiche sempre più efficaci.

2.1. Storia

Dal periodo di Mendel in poi, la raccolta di record genetici ha avuto un lungo percorso [2].
La comprensione della genetica ha fatto passi da giganti negli ultimi trent'anni.
Dal 1982, 579 geni umani sono stati mappati e mappare tramite l'ibridazione è diventato un metodo standard.
Marvin Carruthers e Leory Hood hanno fatto compiere un salto enorme alla bioinformatica quando inventarono un metodo per generare automaticamente la sequenza di Dna.
Nel 1998, fu fondata la Human Genome organization (HUGO). Si tratta di un'organizzazione internazionale di scienziati coinvolti nel Human Genome Project (Progetto Genoma Umano).
Nel 1989, la prima mappa genomica completa fu pubblicata per il batterio H. influentiae.

Nel 1990 si diede inizio al Progetto Genoma Umano.
Dal 1991, un totale di 1879 geni umani sono stati mappati. Nel 1993, Genethon, un centro di ricerca del genoma umano in Francia produsse una mappa fisica del menoma umano. Tre anni dopo, Genethon pubblicò la versione finale della Mappa Genetica Umana. Questo concluse la prima fase del Progetto Genoma Umano.
I bioinformatici sono stati quindi obbligati a creare basi di dati enormi: **GenBank, EMBL** e il database DNA del **Giappone** per memorizzare e confrontare le sequenze dei dati scaturiti dai progetti genoma umano e da altre sequenze genomiche.
Oggi, la bioinformatica abbraccia l'analisi della struttura della proteina, le informazioni funzionali della proteina e del gene, i dati dei pazienti, prove pre-cliniche e cliniche, e i percorsi metabolici di numerose specie.

2.1.1. Origine dei database bioinformatici e biologici

Dopo pochi anni furono costruiti i primi database bioinformatici e biologici quando le prime sequenze proteiche iniziarono ad essere disponibili [30].
La prima sequenza proteica riportata fu quella dell'insulina bovina nel 1956, consistente di 51 residui aminoacidici. Circa dieci anni più tardi, fu segnalata la prima sequenza dell'acido nucleico, quello del tRNA dell'alanina del lievito costituito da 77 basi.
Solo un anno più tardi, Dayhoff ha riunito tutte le sequenze disponibili per creare il primo database bioinformatico.
La banca dati della proteina (Protein DataBank) venne costruita nel 1972 con una collezione di dieci strutture cristallografiche della proteina ai raggi X, e nel 1987 nacque la SWISSPROT, base dati delle sequenze delle proteine.
Una varietà enorme di risorse dati divergenti, di tipi e dimensioni differenti, ora sono disponibili sia nel pubblico dominio che in ambito commerciale per conto terzi.
Tutte le basi di dati originali sono state organizzate in senso molto semplice con le data entry che sono immagazzinate in file normali, o una per entry o come un singolo grande file di testo.
Sono stati poi aggiunti degli indici per permettere la ricerca di keyword delle informazioni di intestazione.

2.1.2. Origine dei tools

Dopo la costruzione dei database, i tool iniziarono a essere disponibili per la ricerca delle sequenze nei database – inizialmente in modo molto semplice, guardando le keyword che matchavano con corte sequenze di parole, poi più sofisticate, utilizzando metodi per il matching di pattern e allineamento.

L’algoritmo veloce, ma meno rigoroso di BLAST, che è stato il sostegno della ricerca di sequenze nei database sin dalla sua introduzione un decennio fa, fu successivamente complementato dagli algoritmi più rigorosi e più lenti di FASTA e di Smith Waterman.
Le suites delle procedure di analisi, scritte dai ricercatori accademici a Stanford, CA, Cambridge, UK e Madison, WI per i loro progetti interni, hanno cominciato a diventare più ampiamente disponibili per l’analisi di sequenze basiche.
Queste procedure erano tipicamente una singola funzione black box che prendevano l'input e producevano l’output sotto forma di file formattati. I comandi stile UNIX furono usati per far operare gli algoritmi, con alcune suite che avevano centinaia di possibili comandi, ognuna con differenti opzioni di comando e formattazione dell’input.
Da questi sforzi iniziali, significativi avanzamenti sono stati fatti nell'automazione dell'accumulazione delle sequenze di informazioni.
La rapida innovazione nella biochimica e nella strumentazione ci ha portati al punto in cui l'intera sequenza genomica di almeno 20 organismi, principalmente agenti patogeni microbici, è conosciuta ed i progetti per delucidare almeno 100 altri genomi sono attualmente in corso.
Gruppi di ricerca ora possono competere per finire la sequenza dell’intero genoma umano.
Con le nuove tecnologie possiamo direttamente esaminare i cambiamenti nei livelli sia di espressione del mRNA che di trascrizione delle proteine nelle cellule viventi, sia durante uno stato morboso, sia in rapporto ad una manipolazione esterna. Possiamo continuare ad identificare i modelli di risposta delle cellule che ci conducono ad una comprensione del meccanismo di azione di un agente su un tessuto.
Il volume di dati derivante da progetti di questa natura è senza precedenti nell'industria farmaceutica ed avrà un effetto profondo sulle modalità con cui vengono usati i dati ed effettuati gli esperimenti nei progetti di ricerca e sviluppo della medicine.
Le ditte farmaceutiche, però, non possono ottenere l'accesso esclusivo a molte sequenze di geni o ai loro profili di espressione. La concorrenza fra i co-licenziatari di una base dati genomica è effettivamente una corsa per stabilire una regola meccanica o altre utility per un gene in una malattia al fine di assicurarsi un brevetto su quel gene. Molto di questo lavoro è effettuato dai tool informatici.
Malgrado il progresso enorme nelle tecnologie di ordinamento e di analisi di espressioni, e la corrispondente grandezza di molti dati che sono tenuti nei database pubblici, privati e commerciali, i tool usati per immagazzinare, recuperare, analizzare e diffondere i dati nella bioinformatica sono ancora molto simili ai sistemi originali riuniti insieme dai ricercatori 15-20 anni fa. Molti sono semplici estensioni degli originali sistemi accademici, avendo risposto alle esigenze sia degli utenti accademici che commerciali per molti anni.
Questi sistemi stanno cominciando a decadere mentre lottano per mantenere il passo di cambiamento nell'industria farmaceutica. I database sono ancora riuniti, organizzati, diffusi e esplorati usando file normali. Le basi dati relazionali sono ancora poche, e i sistemi oggetto-relazionale o completamente object-oriented sono ancora più rari nelle applicazioni tradizionali. Le interfacce ancora si basano sulle linee di comando che devono essere installate su ogni macchina, oppure su form HTML/CGI.

Quando i tool erano nelle mani degli esperti di bioinformatica, le ditte farmaceutiche sono state relativamente gelose.
Ora che i problemi si sono allargati per coprire il processo tradizionale di scoperta, sono necessarie soluzioni molto più flessibili e scalabili per soddisfare i requisiti informatici di R&D delle farmaceutiche.

2.1.3. Cronologia degli eventi

Ci sono differenti punti di vista sull'origine della bioinformatica [4].
Per Attwood il termine bioinformatica è usato per comprendere quasi tutte le applicazioni informatiche nelle scienze biologiche, ma originariamente è stato coniato alla metà degli anni 80 per l'analisi delle sequenze dei dati biologici.
Dall'articolo di M. S. Boguski il termine bioinformatica è un'invenzione relativamente recente, che non compare nella letteratura fino al 1991 e soltanto allora nel contesto dell'emergenti pubblicazioni di elettronica.
Il centro nazionale per l'informazione biotecnologica (NCBI) invece celebrò il decimo anniversario nel 1988.
Così la bioinformatica, di fatto, esiste da più di 30 anni ed è ora è di mezza età.

Di seguito si riportano gli eventi che caratterizzarono la storia della bioinformatica:

- **1951**:
  - Pauling e Corey proposero la struttura alpha-helix e beta-sheet
- **1953**:
  - Watson e Crick proposero il modello della doppia elica per il DNA basandosi sui dati ottenuti ai raggi X da Franklin e Wilkins
- **1954**:
  - Il gruppo di Perutz sviluppò i metodi dell'atomo pesante per risolvere i problemi di fase nella cristallografia delle proteine
- **1955**:
  - La sequenza della prima proteina ad essere analizzata, l'insulina del bovino, fu divulgata da F.Sanger
- **1962**:
  - Teoria di Pauling dell'evoluzione molecolare
- **1965**:
  - Atlante delle sequenze proteiche di Margaret Dayhoff
- **1970**:
  - Furono pubblicati i particolari dell'algoritmo di Needleman-Wunsch per il confronto di sequenza
- **1972**:
  - La prima molecola ricombinata di DNA è creata da Paul Berg e dal suo gruppo

- **1973**:
  - Fu creato il Brookhaven Protein DataBank.
- **1975**:
  - L'elettroforesi bidimensionale, dove la separazione delle proteine sul gel di poliacrilammide di SDS è unita con la separazione secondo i punti isoelettrici, è pubblicata da P. H. O'Farrell
- **1976**:
  - E. M. Southern pubblicò i particolari sperimentali della tecnica Southern Blot delle sequenze specifiche di DNA
- **1977**:
  - Fu pubblicata la descrizione completa di Brookhaven PDB (http://www.pdb.bnl.gov).
  - Allan Maxam, Walter Gilbert (Harvard) e Frederick Sanger (U.K. Medical Research Council) riportarono i metodi per generare le sequenze di DNA.
  - Furono sviluppati i primi software per analizzare e creare le sequenze di DNA
- **1980**:
  - Fu pubblicata la prima sequenza genetica completa di un organismo fu pubblicata. Il gene consisteva di 5,386 coppie basiche.
  - Wüthrich e altri pubblicarono articoli riguardanti l'uso dei multi-dimensionali NMR per la determinazione della struttura proteica.
  - IntelliGenetics fu fondata in California. Il loro primo prodotto fu una suite di programmi per l'analisi delle sequenze di DNA e delle proteine
- **1981**:
  - Fu pubblicato l'algoritmo Smith-Waterman per l'allineamento delle sequenze fu pubblicato.
- **1982**:
  - Il Genetics Computer Group (GCG) fu creato come un settore dell'università del Wisconsin. Il primo prodotto della compania fu una suite di tool per la biologia molecolare.
  - La release 3 della GenBank fu resa pubblica
- **1983**:
  - Furono introdotti i primi algoritmi di ricerca di sequenze nei database (Wilbur-Lipman).
  - LANL (Los Alamos National Laboratory) e LLNL (Lawrence Livermore National Laboratory) cominciarono la produzione delle librerie dei clone del DNA (cosmid) che rappresentano i singoli cromosomi.
  - L'analisi del DNA diventa possibile con la scoperta della reazione a catena della polimerasi. Ciò permette che i piccoli campioni di DNA siano moltiplicati per produrre un campione abbastanza grande da analizzare
- **1985**:
  - Fu pubblicato l'algoritmo FASTP/FASTN.

- Robert Sinsheimer tenne una lezione sulle sequenze del genoma umano all'università della California, Santa Cruz.
- Charles DeLisi e David A. Smith organizzarono il primo congresso a Santa Fe per valutare la possibilità di iniziativa del genoma umano

- **1986**:
  - Nel congresso di Santa Fe, si annunciò l'iniziativa del genoma umano. Con 5,3 milioni di dollari, i progetti pilota cominciarono nei laboratori nazionali della DAINA per sviluppare le risorse e le tecnologie critiche.
  - Il termine "Genomics" apparve per la prima volta per descrivere la disciplina scientifica di mappatura, ordinamento e di analisi delle sequenze genetiche. Il termine fu coniato da Thomas Roderick come nome per il nuovo giornale.
  - La base dati di SWISS-PROT è creata nel reparto di biochimica medica dell'università di Ginevra e nel laboratorio di biologia molecolare europeo (EMBL).
  - La reazione PCR fu descritta da Kary Mullis e dai suoi collaboratori
- **1987**:
  - Fu descritto l'uso dei cromosomi artificiali del lievito (YAC).
  - Il comitato consultivo congressuale istituito della DOE, HERAC, progettò un lavoro pluridisciplinare, scientifico e tecnologico di 15 anni, per tracciare ed ordinare il genoma umano.
  - La DOE indicò i centri pluridisciplinari del genoma umano.
  - NIH NIGMS cominciò a costituire un fondo per i progetti del genoma umano
- **1988**:
  - La NCBI (National Center for Biotechnology Information) fu creata alla NIH/NLM.
  - Fu costruita la rete EMBnet per la distribuzione dei database.
  - Fu dato il via all'iniziativa del genoma umano.
  - L'algoritmo FASTA per il confronto delle sequenze fu pubblicato da Pearson e Lupman.
  - I rapporti dai comitati congressuali di NAS NRC e di OTA suggerirono il programma di ricerca concordato del genoma.
  - HUGO fu fondato dagli scienziati per coordinare gli sforzi internazionali.
  - La prima riunione annuale dei Cold Spring Harbor Laboratory sulla mappatura del genoma umano
  - La DOE e la NIH firmarono le linee guida dei piani per la cooperazione sulla ricerca del genoma.
  - La sequenza di Telomere (estremità del cromosoma) che ha implicazioni in merito all'invecchiamento e alla ricerca sul cancro fu identificata a LANL
- **1989**:
  - La genetics Computer Group (GCG) diventò una compania privata.

- L’ Oxford Molceular Group,Ltd.(OMG) fu fondato in Inghilterra da Anthony Marchigton, David Ricketts, James Hiddleston, Anthony Rees e W.Graham Richards. I loro primi prodotti furono: Anaconda, Asp, Cameleon e altri (modellatori molecolari, design di medicinali, design di proteine).
- La DOE e la NIH stabilirono il Joint ELSI Working Group

- **1990**:
  - Fu implementato il programma BLAST.
  - Un gruppo di applicazioni molecolari fu fondato in California da Michael Levitt e Chris Lee. I loro primi prodotti furono Look e SegMod che erano usati per modellare i design molecolari e proteici.
  - InforMax fu fondata a Bethesda. I prodotti di questa compania si indirizzavano all’analisi delle sequenze, alla gestione dei database e dei dati, alla ricerca, alla pubblicazione di grafici, alla costruzione di cloni, alla mappatura e ai primi design.
  - La DOE e la NIH presentarono un piano congiunto di 5 anni al Congresso Americano. Il progetto di 15 anni incominciava formalmente e si incominciò a contrassegnare le posizioni dei geni sulla mappa dei cromosomi come posizioni dell'espressione del mRNA
- **1991**:
  - Fu descritti la creazione e l'utilizzo dei tag di sequenze espresse (ESTs).
  - La Myriad Genetics Inc. fu fondata nello Utah. L'obiettivo dell'azienda era di condurre alla scoperta dei geni delle maggiori malattie umane e dei loro progressi. L'azienda scoprì ed ordinò, con i propri collaboratori accademici, i seguenti geni in serie principali: BRCA1, BRACA1, CHD1, MMAC1, MMSC1, MMSC2, CtIP, p16, p19 e MTS2.
  - Fu costruita la banca dati della mappatura del cromosoma umano, GDB
- **1992**:
  - La mappa genetica dell’intero genoma umano fu pubblicato.
  - La guida di riferimento per i dati rilasciati e le risorse condivise furono annunciata da DOE e da NIH
- **1993**:
  - L’Affymetrix fu fondata a Santa Clara, California
  - L’International IMAGE Consortium stabilì come coordinare in modo efficiente la mappatura e la sequenza della rappresentazione genetica del cDNA
- **1994**:
  - Il database PRINTS dei motivi delle proteine fu pubblicato da Attwood e Beck.
  - Vennero completate le librerie di seconda generazione dei cloni del DNA che rappresentano ogni cromosoma umano da LLNL e LBNL

- **1995**:
  - Furono pubblicate le sequenze del genoma Haemophilus influenzae e del Mycoplasma genitalium.
  - Fu pubblicata una mappa fisica di oltre 15000 STS marker
- **1996**:
  - Furono pubblicate le sequenze del genoma Saccharomyces cerevisiae e del Methanococcus jannaschii.
  - Affymetrix produsse il primo chip di DNA commerciale
- **1997**:
  - Il genoma E.coli fu pubblicato.
  - La LION bioscience AG fu fondata come una compania genomica focalizzata sulla bioinformatica. La compania fu costruita da European Molecualr Biology Laboratory (EMBL), European Bioinformtics Institute (EBI), German Cancer Research Center (DKFZ), e University of Heidelberg
- **1998**:
  - Il Swiss Institute of Bioinformatics fu costituita come una fondazione non.profit.
  - Nacquerò varie società orientate alla genomica e alla bioinformatica
- **1999**:
  - Fu mappato il primo cromosoma umano.
  - Il 1° dicembre, i ricercatori del Progetto Genoma Umano annunciarono la mappatura completa del DNA del cromosoma 22.
- **2000**:
  - Furono pubblicati i genomi del Pseudomonas aeruginosa, del A.thaliana e del D.melanogaster.
  - I leader del HGP e il Presidente Clinton annunciarono il compimento della bozza della mappatura del DNA del genoma umano
- **2001**:
  - Fu pubblicato il genoma huam
  - Fu completato il cromosoma umano.
  - Il cromosoma 20 fu il terzo cromosoma completamente mappato dal progetto genoma umano
- **2002**:
  - Un consorzio internazionale pubblicò l'intera mappatura delle sequenza genetica del comune topo di casa.
- **2003**:
  - Fu completato nell'aprile il Progetto Genoma Umano.
  - Fu completato il cromosoma umano 14
- **2004**:
  - La sequenza genomica della cavia da laboratorio, Rattus norvegicus, fu completato dal progetto Rat Genome Sequencing

2.2. Banche dati biologiche

Numerosissime sono oggi le banche dati biologiche esistenti grazie allo sviluppo delle biotecnologie molecolari che hanno portato alla produzione diversificata di enormi quantità di dati biologici [5].
Attualmente, le banche dati delle sequenze nucleotidiche contengono 16 x 10^9 basi (che si abbrevia in 16 Gbp). Buona parte delle banche dati biologiche disponibili risulta essere ben strutturata e di notevole supporto alla moderna ricerca biomolecolare, per cui in questo sottocapitolo si intende offrire una panoramica completa, distinguendole per grandi categorie.
Prima di passare alla descrizione delle varie categorie, risulta necessario definire alcuni elementi fondamentali che sono di supporto alla comprensione dei contenuti e della organizzazione delle banche dati.
Una banca dati biologica raccoglie informazioni e dati derivati dalla letteratura e da analisi effettuate sia in laboratorio sia attraverso l'applicazione di analisi bioinformatiche.
Ogni banca dati biologica è caratterizzata da un elemento biologico centrale che costituisce l'oggetto principale intorno al quale viene costruita la entry della banca dati.
L'esempio seguente ci può aiutare a comprendere questo concetto.
Nelle banche dati di sequenze di acidi nucleici l'elemento centrale è la sequenza nucleotidica di DNA o RNA, a cui vengono associate annotazioni classificanti l'elemento stesso quali il nome della specie, la funzione e le referenze bibliografiche. Ciascuna entry raccoglie quindi le informazioni caratterizzanti l'elemento centrale.
Ogni qual volta si organizza una banca dati si definiscono i tipi di attributi da annotare nella stessa e anche il formato con cui tali informazioni vengono organizzate; si definisce cioè la struttura della banca dati.

2.2.1. Le banche dati primarie

Le **banche dati primarie** (o banche dati di sequenze di acidi nucleici) contengono solo informazioni molto generiche che vengono associate ad una sequenza per poterla identificare dal punto di vista specie-funzione.
Le principali banche dati primarie sono tre:

- **EMBL datalibrary**
- **GenBank**
- **DDBJ**

La EMBL datalibrary è la banca dati europea costituita nel 1980 nel laboratorio europeo di biologia molecolare di Heidelberg (Germania).
La GenBank è la corrispondente banca dati americana costituita nel 1982.
La DDBJ è la corrispondente Giapponese.

Le banche dati primarie vengono aggiornate continuamente, attraverso internet, sia da ricercatori, che producono nuove sequenze, sia da annotatori reclutati dai centri di raccolta dati, anche se quest'ultimi forniscono un apporto minore.
Precedentemente abbiamo scritto che le informazioni contenute nelle banche dati primarie sono molto generiche, quindi è possibile trovare delle ridondanze delle informazioni, ossia le sequenze vengono immesse più volte, e ciò implica che le statistiche effettuate sono poco attendibili.
Per far fronte a questo inconveniente viene applicato il software CleanUP che genera un insieme di sequenze non ridondanti.

2.2.2. Le banche dati specializzate

Oltre alle banche dati primarie vi sono numerosissime **banche dati specializzate** (o banche dati sequenze proteiche) che raccolgono sia sequenze proteiche ottenute dalla determinazione sperimentale della sequenza aminoacidica, sia sequenze proteiche derivate dalla traduzione di sequenze nucleotidiche per le quali sia stata individuata o predetta la funzione di gene codificante per una proteina.
I dati estratti dalle banche dati primarie relativi a proteine vengono accuratamente validati e arricchiti di informazioni specifiche.
Le banche dati specializzate sono tre:

- **SWISSPROT**
- **TREMBL**
- **PIR**

La SWISSPROT è sviluppata in Svizzera a Ginevra dal gruppo di Amos Bairoch che afferisce all'istituto nazionale SIB e che ha sviluppato numerose altre banche dati, tutte integrate fra loro.
Grande cura in SWISSPROT è posta all'annotazione del nome della proteina e al codice del relativo gene, anche se a tal proposito vi è il problema della nomenclatura dei geni e delle proteine. Numerosissimi sono i casi in cui a uno stesso gene, in specie diverse, viene attribuito un nome differente e anche diversi sono i casi di geni aventi lo stesso nome pur svolgendo una differente funzione.
Per ovviare a questo problema è stato costituito un consorzio da parte dei gruppi coinvolti nei progetti genomici per realizzare **Gene Ontology** (GO), un vocabolario controllato descrittivo delle funzioni molecolari, dei processi biologici e delle localizzazioni cellulari relative a ciascun gene e al suo prodotto.
La banca dati SWISSPROT è aggiornata dal gruppo svizzero in collaborazione con l'EBI dove viene sviluppata un'altra banca dati di proteine, TREMBL, risultato della traduzione automatica in aminoacidi di tutte le sequenze annotate nella banca dati EMBL come sequenze codificanti proteine.
La PIR (Protein Information Resource) è sviluppata in collaborazione fra due grossi centri: la Georgetown University negli USA e il MIPS a Monaco di Baviera. La PIR è senz'altro

una banca dati valida dal punto di vista della qualità delle annotazioni e del livello di aggiornamento, ma è poco integrata con le altre banche dati biologiche.

2.2.3. Banche dati di motivi e domini proteici

Una banca dati è un utilissimo strumento di consultazione, sia per effettuare ricerche testuali ed estrarre informazioni specifiche su un dato argomento, sia come strumento di comparazione per individuare, in nuove sequenze, caratteristiche strutturali e funzionali già riscontrate in altre ed annotate in banche dati specifiche.
Tra le due possibilità, la più interessante, dal punto di vista bioinformatico, è la comparazione, che può essere effettuata attraverso l'applicazione di tecniche di ricerca di similarità o, quando la ricerca di similarità non evidenzia sequenze simili a quelle in oggetto, attraverso l'applicazione di tecniche di ricerca di segnali (pattern recognition) basate su algoritmi più o meno complessi; questo secondo approccio consente di ritrovare segnali, motivi o domini strutturali e funzionali che si conservano nel tempo.
Numerose banche dati specializzate, che annotano informazioni relative a motivi e domini funzionali, sono state integrate in **InterPRO**, una risorsa bioinformatica, sviluppata all'EBI (centro di raccolta), che consente di ricercare contemporaneamente informazioni funzionali e strutturali relative a una proteina o ad una famiglia di proteine su più banche dati, distribuite su calcolatori diversi e strutturate in modo differente.
La ricerca dei dati in InterPRO viene effettuata attraverso un sistema di ricerca semplice basato su componenti del DBMS Oracle o attraverso il sito SRS dell'EBI.
Inoltre, attraverso il software InterPROscan, è possibile ricercare motivi strutturali e funzionali annotati nelle banche dati integrate in InterPRO al fine di caratterizzare, dal punto di vista funzionale, nuove proteine derivate da progetti di sequenziamento genomico.
Una delle banche dati integrate in InterPRO è **PROSITE** che annota patterns aminoacidici individuati in set di sequenze proteiche determinati sperimentalmente in una o più proteine e riportati in letteratura.
La banca dati PROSITE contiene motivi codificati in due modi diversi: i pattern e le matrici.
I pattern sono motivi definiti con una sintassi riconducibile ad espressioni regolari mentre le matrici sono invece definite facendo ricorso alle matrici posizionali di peso, compresi gli Hidden Markov Models.

2.2.4. Banche dati di strutture proteiche

Come si è già potuto notare nei paragrafi precedenti, la conoscenza di motivi strutturali delle proteine è fondamentale per la comprensione funzionale delle biosequenze.

Per dati strutturali di una proteina s'intende la distribuzione spaziale degli atomi componenti gli aminoacidi e quindi degli aminoacidi stessi; tali dati corrispondono alle coordinate atomiche determinate attraverso analisi cristallografiche ai raggi X.
L'unica banca dati che raccoglie tali informazioni è la **PDB** (Protein Data Bank) che nel dicembre 2002 riportava circa 19400 strutture comprendenti:

- strutture proteiche - coprono quasi il 90% dela banca dati;
- complessi di proteine con acidi nucleici (i fattori di trascrizione legati al DNA e i ribosomi) - circa il 4% della banca dati;
- strutture di acidi nucleici (DNA e RNA) – circa il 6% della banca dati;
- strutture di carboidrati, che attualmente sono solo 18 – circa lo 0,09%.

Questo database è un riferimento unico per tutti gli studi strutturali di interesse biologico.
Il PDB è una banca dati ridondante, ossia contiene molte strutture della stessa proteina o proteine simili, e purtroppo è non ideale perché i file non sono tra loro omogenei.

2.2.5. Risorse genomiche

Con l'avanzare dei progetti genomici, la bioinformatica ha avuto un forte impulso e quindi uno sviluppo di risorse genomiche accessibili più o meno liberamente in rete. Le risorse genomiche sono siti in cui è possibile reperire dati relativi al sequenziamento genomico e al mappaggio.
Le risorse possono essere:

- risorse integrate, dove sono disponibili dati relativi a tutti i genomi attualmente in fase di studio;
- risorse relative ai genomi di specifiche categorie di organismi, dove sono raccolti tutti i dati genomici dei Batteri;
- risorse organismo specifiche quali **MGD** per il topo, **RGD** per il ratto, **GDB** per l'uomo e altri ancora.
  Una caratteristica comune a tutti questi siti è la possibilità di scaricarsi sul proprio computer le sequenze dell'intero genoma o di parti di esse.
  E' sempre possibile quindi effettuare ricerche di similarità di sequenza contro l'intero genoma o parti di esso, mediante l'applicazione dei metodi FASTA e/o BLAST.

In tale contesto il sistema più rilevante è **Ensembl**, sviluppato per la raccolta dei dati relativi alle annotazioni del genoma umano. Annotare un genoma significa caratterizzare le sue funzioni attraverso la ricerca di dati già determinati, oppure applicando metodologie bioinformatiche che consentano di caratterizzare nuove funzioni e proprietà.
Un altro sistema che può essere preso a riferimento è **HumGuide**, attraverso il quale è possibile eseguire un'analisi dettagliata delle risorse riguardanti il Progetto genoma umano.

2.2.6. Banche dati del trascrittoma

Nell'evoluzione dei progetti genomici si è diffusa la tendenza a raggruppare le categorie di dati biologici in omics e in tale contesto è inserito il trascrittoma, ossia l'insieme di tutti i trascritti di un dato organismo, ottenuti attraverso il sequenziamento dei cDNA completi o delle EST (Expressed Sequenze Tags), sequenze parziali che consentono di caratterizzare i cloni di cDNA, etichettandoli con le rispettive sequenze terminali.
Si stanno così realizzando i database del trascrittoma o comunque database associati ai dati del trascrittoma.
Fra i database del trascrittoma vi è dbEST che è stato realizzato per raccogliere le EST prodotte per ciascun gene.
Nella banca dati dbEST, intorno alla metà del 2002, erano registrate più di 12 milioni di EST, appartenenti a oltre 400 organismi diversi. Soltanto le EST umane superavano i 4,5 milioni, mentre quelle di topo superavano i 2,5 milioni.
Assumendo che i geni umani siano circa 35000, ogni gene umano dovrebbe essere rappresentato in media oltre 120 volte.
In realtà alcuni geni sono rappresentati molto più di altri geni ed è probabile che alcuni geni umani non siano ancora stati identificati come EST.
Il clustering di EST ha lo scopo di raggruppare (in inglese *to cluster*) tutte le EST appartenenti allo stesso gene.
Per eseguire il clustering di EST non si possono dare delle regole generali o dei programmi pronti da eseguire. Il problema consiste nel fatto che ogni progetto di clustering è in qualche modo diverso dagli altri. Nei casi più complessi il clustering può essere realizzato a partire da tutte le sequenze note di trascritti di un determinato organismo; nei casi più semplici è limitato a piccoli progetti di EST di organismi precedentemente ignorati dal punto di vista molecolare.
Uno dei problemi generali più difficili da risolvere riguarda i criteri secondo cui due sequenze debbano essere poste in uno stesso cluster piuttosto che in due cluster diversi.
Di seguito riportiamo una breve panoramica dei metodi di clustering utilizzati su queste banche dati, alcuni dei quali verranno ripresi in modo più approfondito nel capitolo riguardante il data minino [14].
Gli algoritmi di clustering possono essere divisi in:

- **Gerarchici** - Non è necessaria alcuna informazione a priori sui dati di espressione e il risultato dell'algoritmo è una serie annidata di gruppi (o cluster);
- **Non-Gerarchici** - Cercano di raggruppare gli elementi (in questo caso i geni) in un numero predefinito *k* di gruppi, senza specificare alcuna relazione tra di essi.

Metodi gerarchici. La classificazione gerarchica è semplice e facile da interpretare. E' un metodo agglomerativo e, quindi, parte con un numero di cluster pari al numero totale di geni per raggrupparli successivamente in base al grado di similarità.
I punti principali sono:

1. calcolo di una matrice di distanze a coppie, in cui il numero di righe e di colonne è pari al numero di geni e ogni cella rappresenta la distanza tra i due rispettivi geni;

2. individuazione della coppia di geni o di cluster più simili per raggrupparli in un unico cluster; nel caso di più di una coppia di geni/cluster, con lo stesso grado di similarità delle procedure standard, scelgono la coppia da prendere;
3. aggiornamento della matrice di distanza a coppie;
4. ripetizione della procedura dal punto 2, finché non si ottiene un unico cluster (la radice dell'albero) contenente tutti gli elementi.

Metodi non-gerarchici. Se si è in grado di avere delle informazioni a priori sul numero finale di possibili cluster i metodi non-gerarchici possono essere una valida alternativa ai metodi gerarchici.
Gli algoritmi di analisi non-gerarchica cercano di raggruppare gli elementi in modo tale che siano il più possibile omogenei all'interno dei cluster e il più possibile disomogenei tra i vari cluster.
Non viene inoltre prodotto alcun albero come risultato.
I passi principali sono:

1. tutti gli elementi sono assegnati casualmente nei *k* cluster definiti a priori;
2. è calcolato un vettore di espressione media per ogni cluster e quindi è generata una matrice di distanze a coppie tra tutti i *k* cluster, sulla base di questo vettore medio, la matrice di distanza a coppie è ricalcolata e aggiornata;
3. con un processo iterativo gli elementi sono spostati tra un cluster e l'altro, ad ogni spostamento è calcolata la distanza tra l'elemento spostato e il nuovo cluster; l'elemento può rimanere nel nuovo cluster solo se la sua distanza con il vettore medio del nuovo cluster è minore di quella con il vecchio cluster;
4. ripetizione della procedura dal punto 2, finché gli spostamenti non generino ulteriori variabilità *intra*- o *inter*-cluster.

2.2.7. Altre banche dati

Finora abbiamo dedicato particolare attenzione alle banche dati più importanti, ad esse se ne aggiungono delle altre, che elenchiamo:

- **Banche dati biologiche per il sistema immunitario**
- **Banche dati di geni**
- **Banche dati di profili di espressione**
- **Banche dati di polimorfismi e mutazioni**
- **Banche dati di pathways metabolici**
- **Banche dati mitocondriali**

Non meno importanti per la loro funzione, ma non di grande rilievo dal punto di vista informatico.

Capitolo 3
Microarray

Le tecnologie dei Microarray in generale forniscono i nuovi attrezzi che trasformano il **"senso"** per cui gli esperimenti scientifici sono effettuati [30].
Il vantaggio principale delle tecnologie basate sui microarray rispetto ai metodi tradizionali è quello della scalabilità.
In luogo degli esperimenti condotti in precedenza, basati sui risultati derivanti da uno o pochi geni, i microarray tengono conto dell'interrogazione simultanea di migliaia di geni o di interi genomi.

3.1. Cosa sono i microarray

I Microarray sono vetrini microscopici che contengono una serie ordinata di campioni (DNA, RNA, proteina, tessuto) [7].
Il tipo di microarray dipende dal materiale disposto sulla slide: DNA, DNA microarray; RNA, RNA microarray; proteina, microarray proteici; tessuto, microarray tissutali.
Poiché i campioni sono organizzati nell'ordine in cui i dati sono ottenuti dal microarray, questi possono essere tracciati nuovamente per qualsiasi dei campioni analizzati.
Ciò significa che i geni sul microarray sono indirizzabili.
Il numero di campioni ordinati su un microarray può aggirasi intorno alla centinaia di migliaia.
Il classico microarray contiene parecchie migliaia dei geni accessibili.
Il microarray più largamente utilizzato è il DNA microarray.
Un **DNA microarray** (comunemente conosciuto come *gene chip*, *DNA chip*, o *biochip*) è costituito da una collezione di sequenze oligonucleotidiche di DNA attaccate ad una superficie solida come vetro, plastica, o chip siliconici formanti un array.
Tali array sono usati per determinare:

- il livello di espressione dei geni in un campione, comunemente detto profilo d'espressione

- la presenza di un gene o di una breve sequenza in miscela di migliaia (spesso anche tutto il patrimonio genetico di un individuo umano o non).

I **microarray** sfruttano una tecnica di ibridazione inversa, consistente cioè nel fissare tutti i probe su un supporto e nel marcare invece l'acido nucleico target [30].
È una tecnica che, sviluppata negli anni '90, oggi permette l'analisi dell'espressione genica monitorando in una sola volta gli RNA prodotti da migliaia di geni.
Per studiare gli mRNA, questi vengono prima estratti dalle cellule, convertiti in cDna attraverso una reazione di polymerase chain reaction (PCR), con l'uso di un enzima chiamato transcriptasi inversa, e allo stesso momento marcati con una sonda fluorescente.
Quando si fa provoca l'ibridazione fra la sonda presente sulla matrice e il cDna target, quest'ultimo rimane legato alla sonda e può essere identificato semplicemente rilevando la posizione a cui è rimasto legato.
Le principali applicazioni dei microarray sono l'analisi dei polimorfismi SNP, il confronto di popolazioni di RNA di cellule diverse, l'utilizzo per nuove metodologie di sequenziamento del Dna, nonché per lo screening di sequenze senso e antisenso nella ricerca degli oligonucleotidi usati in campo farmaceutico.
Il segmento di DNA legato al supporto solido (oligonucleotide) è noto come probe.
Migliaia di probe sono usati contemporaneamente in un array.
Questa tecnologia è nata da una tecnica semplice, nota come Southern blotting, dove frammenti di DNA attaccati ad un substrato sono testati da sonde geniche aventi sequenze conosciute.
La misura dell'espressione genica mediante microarray ha un notevole interesse sia nel campo della ricerca di base che nella diagnostica medica, in particolare di malattie a base genetica, dove l'espressione genetica di cellule sane viene comparata con quella di cellule affette dalla malattia in esame.

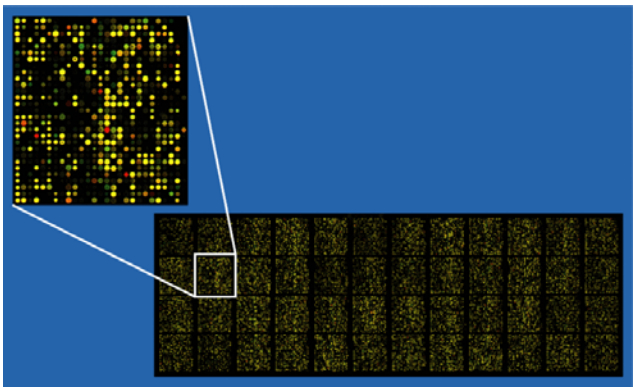

Figura 3.1 - Microarray

3.2. Produzione

La produzione di microarray non è un task triviale, ma un'arte e una scienza [8].
Questo tipo di lavoro richiede considerevole esperienza in chimica, ingegneria, programmazione, gestione di vasti progetti e biologia molecolare.
Lo scopo durante questa fase è ottenere punti riproducibili con una morfologia costante.
I microarray possono essere fabbricati usando diverse tecnologie, fra cui la stampa di micro solchi, usando un particolare microspillo appuntito su una lastrina di vetro dove verrà attaccata covalentemente la sonda (probe) di materiale genetico ottenuta per clonazione, sfruttando la tecnica PCR, (fotolitografia).
Esistono però in commercio maschere preformate prodotte da ditte specializzate.
Ogni singolo clone viene posizionato nell'esatta locazione sul vetrino da un robot.
E' evidente che questa tecnica richiede apparecchiature robotiche molto sofisticate. Il nucleo dell'apparecchiatura è costituito da una "gruppo scrivente" che preleva uno più campioni di cDna mediante l'utilizzo di pennini e li trasferisce su vetrini per microscopio; il movimento è ovviamente controllato da un computer.
Durante la deposizione del campione, il sistema di controllo del robot registra automaticamente tutte le informazioni necessarie alla caratterizzazione ed alla completa identificazione di ciascun punto della matrice.
Quando la sonda è sul vetrino si effettua il processing, passaggio in cui la sonda viene legata covalentemente al supporto attraverso una reazione innescata dall'irraggiamento con luce ultravioletta, o incubando il vetrino a 80 °C per 2 h.
Infine il cDna viene reso a singola catena attraverso una denaturazione termica o chimica.
Questa tecnica però dà la possibilità di creare solo microarray a bassa densità (ovvero con poche sonde per mm quadrati).
I DNA microarray possono essere usati per rivelare RNA che può essere o non essere tradotto in proteine.
Gli scienziati chiamano questa analisi "analisi dell'espressione" o profilo d'espressione.
Con la tecnologia dei microarray si possono avere decine di migliaia di risultati in pochissimo tempo.
Questa tecnologia ha perciò permesso notevoli accelerazioni in diversi campi di investigazione biochimici e biotecnologici.
L'uso di microarray per lo studio del profilo d'espressione genetica è stato pubblicato per la prima volta nel 1995 (*Science*); il primo genoma eucariotico completato con analisi di microarray fu quello del *Saccharomyces cerevisiae* nel 1997 (*Science*).
Di seguito verranno esaminati altri tipi di produzione, determinati essenzialmente dal tipo di microarray che si analizza.

3.2.1. Array per fotolitografia (chip-array)

In questo caso gli oligonucleotidi sono sintetizzati in situ; questa tecnica è stata utilizzata per la prima volta dall'Affymetrix, che ne detiene il brevetto.
La tecnica per la produzione di questi chip è detta fotolitografia, con la quale è possibile sintetizzare molte migliaia di differenti oligonucleotidi sulla superficie di un vetrino.
Anche se questa tecnica di sintesi è molto accurata, la massima lunghezza degli oligonucleotidi che è possibile raggiungere è di 25 nucleotidi.
Putroppo gli oligonucleotidi di queste dimensioni non sono sufficienti a dare specificità al microarray; per questo servono almeno 3 oligonucleotidi che legano un gene, e altri 3 oligonucleotidi che presentano un mismatch che serviranno da controllo negativo.
Le analisi di un singolo gene richiedono lo studio di sei spot che devono avere come risultato: i tre oligonucleotidi corretti, positivi, e i tre oligonucleotidi con il mismatch, negativi.
Ogni volta bisogna fare un chip per il controllo e uno del soggetto da analizzare, perché non si può effettuare un'ibridazione per competizione.
Sui microarray a bassa densità solitamente si usano marcatori radioattivi, che però non permettono una risoluzione sufficientemente elevata per i chip ad alta densità, con i quali è necessario utilizzare marcatori fluorescenti.
Una volta che il microarray è stato costruito o comprato, e il campione di acidi nucleici da analizzare è stato isolato, si provoca la reazione di ibridazione, che permette la formazione degli eteroduplex.
Per ottenere dei buoni microarray è essenziale difenderli dall'umidità (se l'ambiente è secco la soluzione evapora, se è umido si deposita dell'acqua) e dalla polvere (ogni spot è grande circa 50 micron, un granello di polvere è più grande di 50 micron, per cui può coprire vari spot); per questo motivo sono state create delle camere apposite per l'ibridazione dei microarray che vengono sigillate.
Dopo l'ibridazione il microarray viene lavato per rimuovere il cDna che non si è legato.
Di norma il Dna fluorescente dei campioni sperimentali è mescolato con un Dna di un soggetto di controllo marcato con un colorante fluorescente diverso.
Per i microarray si usano solitamente Cy3 (che emette una lunghezza d'onda nel campo del verde) e Cy5 (che emette nel campo del rosso).
In questo modo se la quantità di RNA espressa da un gene nelle cellule di interesse è aumentata (up regolata) rispetto a quella del campione di riferimento, lo spot che ne risulta sarà del colore del primo fluorescente.
Viceversa se l'espressione del gene è diminuita (down regolata) rispetto al campione di riferimento lo spot sarà colorato dal secondo fluorescente.
La fluorescenza è rilevata da uno scanner a laser, grazie al quale si acquisisce un'immagine per ogni fluoroforo.
Vengono poi usati dei software appositi per convertire i segnali in una gamma di colori dipendente dalla loro intensità.
Il segnale rilevato dallo scanner viene infine sottoposto ad altri algoritmi di filtrazione e di pulizia e convertito in valori numerici.

Il principale problema dei microarray è la mancanza di standardizzazione, che causa difficoltà nel confronto di dati; inoltre, se oggi con questa tecnica è possibile analizzare i livelli di espressione di un singolo gene ottenendo degli ottimi risultati, la combinazione dello studio di molte migliaia di geni risulta molto complicato e può portare spesso a dei falsi positivi.
Questo accade anche perchè alcuni cDna possono cross-ibridare altre sonde, che invece avrebbero dovuto rilevare altri geni.
Un altro problema è presentato dai fluorofori, che, nonostante siano molto simili fra loro, presentano delle differenze problematiche.
Esiste una diversa efficienza di fluorescenza tra Cy3 e Cy5 che deve essere standardizzata dai software di rilevazione.
Poiché Cy3 è più piccolo di Cy5, esiste un diverso livello di incorporazione del due fluorofori, in quanto la polimerasi presenta più difficoltà a inserire il nucleotide marcato con Cy5 a causa dell'ingombro sterico.
Come se non bastasse Cy5 si presenta più labile di Cy3, quindi una prima scansione di Cy3 con il laser potrebbe ridurre la fluorescenza di Cy5.
Per ovviare a tutte questa problematiche e per creare un minimo di standardizzazione si effettua il Die swap: consiste nell'effettuare un secondo microarray scambiando l'uso dei fluorofori.
Se nel primo microarray Cy3 è stato usato per marcare il cDna sperimentale, nel secondo microarray si userà Cy3 per marcare il cDna del soggetto di controllo, e viceversa per Cy5.

3.2.2. Spotted microarrays

Negli **spotted microarrays** (o **microarrays a doppio canale**), i probe sono oligonucleotidi, cDNA o piccoli frammenti prodotti con la tecnologia PCR corrispondenti a mRNA.
Questo tipo di microarray sfrutta l'ibridazione di DNA con cDNA da due campioni comparati (es. paziente e controllo), che sono marcate con due differenti fluorofori.
I campioni possono essere miscelati e ibridizzati in un singolo microarray e quindi analizzati, permettendo la visualizzazione dei geni up-regolati e down-regolati contemporaneamente.
Con questa tecnica il livello assoluto dell'espressione genica non può essere apprezzata a pieno, ma il costo dell'analisi è ridotto della metà.

3.2.3. Microarrays di oligonucleotidi

Nei **Microarrays di oligonucleotidi** (o **single-channel microarrays**), i probe sono progettati per riconoscere parti di sequenze di mRNA conosciute o predette.

Vi sono matrici microarray di tal specie commercializzate da numerose ditte specializzate come GE Healthcare, Affymetrix, or Agilent.
Queste matrici contengono oligonucleotidi importanti per alcune analisi rutinarie o addirittura grosse parti di genomi di vari esseri viventi.
Possono inoltre essere prodotte matrici ad hoc al fine di soddisfare qualsiasi bisogno, sia per la ricerca che per la diagnostica. Arrays oligonucleotidici possono essere prodotti o per deposizione piezoelettrica dell'intera lunghezza dell'oligo, o per sintesi in situ (fotolitografia).
Arrays di lunghi oligonucleotidici sono composti da 60-meri (oligo costituiti da 60 basi) e sono prodotti con la tecnologia ink-jet printing su substrati di silicio.

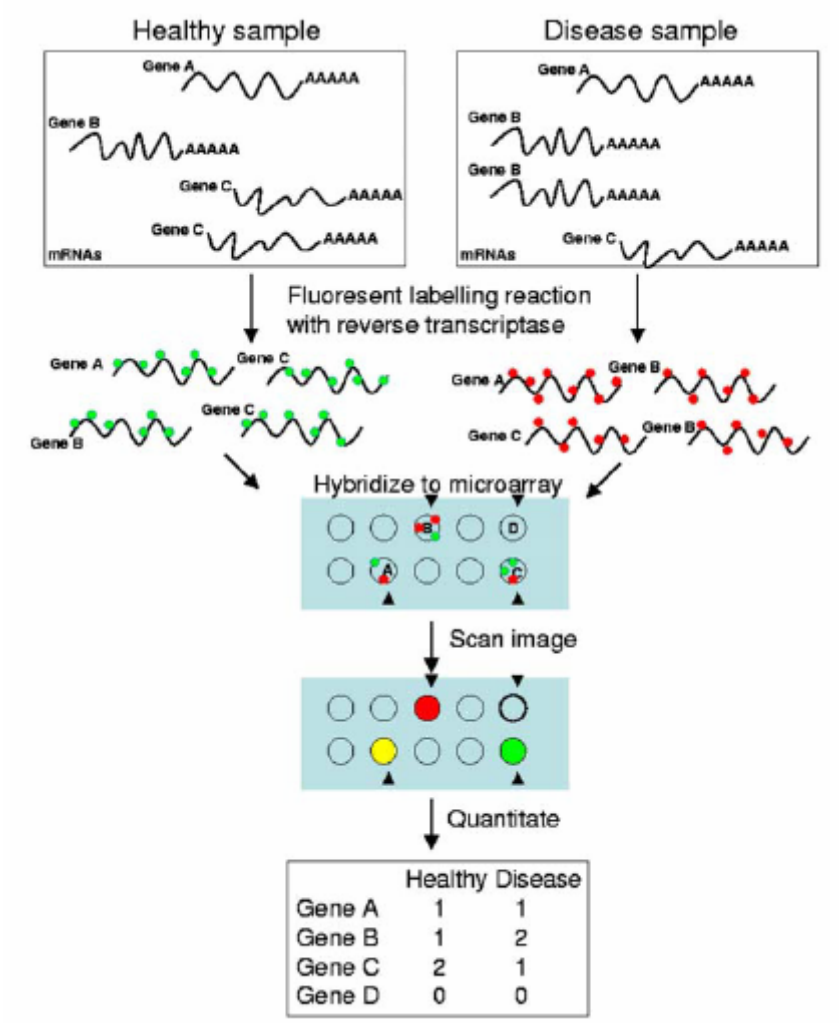

Figura 3.2 - Work flow di un tipico esperimento di espressione di microarray

Arrays di oligonucleotidi corti sono composti da 25-meri o 30-meri e sono prodotti per sintesi fotolitografica su substrati di silicio (Affymetrix) o per deposizione piezoelettrica su matrici acrilammidiche (GE Healthcare).
La NimbleGen Systems, recentemente, ha sintetizzato nuove matrici dette Maskless Array che possono essere utilizzate in modo flessibile con numerosissimi oligonucleotidi test (probe).
I nucleotidi che formeranno gli oligonucleotidi sono dei nucleotidi modificati che presentano un gruppo protettore fotolabile, il quale finché è presente ne impedisce il legame all'oligonucleotide in crescita.
Questo gruppo può essere allontanato con una fonte luminosa che permette ai nucleotidi di reagire.
Si usano delle "maschere" per determinare quali nucleotidi e in quale posizione devono essere attivati dalla luce.
In questo modo sequenze oligonucleotidiche specifiche possono essere costruite in posizioni predeterminate.
Questa tecnica permette di preparare microarray ad alta densità.
Un array standard può contenere più di 390000 pozzetti test (spots).
Nuovi array sono in studio per la ricerca in campo biochimico (vie metaboliche) o per la diagnosi e la prevenzione in campo medico.
In particolare questa tecnica è importante per l'analisi del genoma di soggetti con malattie genetiche o che sono soggetti a potenziali malattie familiari, come il diabete, malattie cardiovascolari o tumori familiari.

3.3. Microarray di genotipi

DNA microarrays possono essere usati per lo studio di genotipi.
Gli **SNP microarrays** sono particolari DNA microarrays che sono usati per identificare i così detti tratti ipervariabili, ovvero quelle sequenze che variano da individuo ad individuo nell'ambito della stessa specie o in sotto popolazioni isolate geograficamente o socialmente.
Arrays di oligonucleotidi corti sono usati per identificare il polimorfismo di un singolo nucleotide (single nucleotide polymorphisms) (SNPs), che si pensano responsabili della variazione genetica e della suscettibilità individuale a manifestare determinate malattie.
I DNA microarrays possono essere usati anche per la genotipizzazione (genotyping) che trova impiego nella medicina forense (esame del DNA), nella diagnostica e in una nuova branca della farmacologia la farmacogenomica che si propone di studiare la relazione tra diversità genetica e risposta ai farmaci, intendendo per risposta sia gli effetti terapeutici che quelli collaterali o avversi.

Gli SNP microarrays sono usati per tracciare i profili di mutazione somatica nelle cellule tumorali. L'amplificazione e la delezione a cui vanno soggette queste cellule possono essere investigate contemporaneamente ai microarrays l'ibridazione genomica comparativa.

3.4. Protein Microarray

Si ottengono utilizzando differenti proteine, fissate su microarray, come sonde.
I protein microarray sono usati per identificare le interazioni proteina-proteina o, ad esempio, per identificare i substrati delle proteine chinasi o ancora per identificare gli obiettivi di piccole molecole biologicamente attive.
Le proteine più comunemente usate durante un protein microarray sono gli anticorpi monoclonali, dove gli anticorpi sono stampati sul vetrino e usati come sonde per rilevare le proteine del lisato cellulare.
L'uso di anticorpi monoclonali però crea alcuni problemi, principalmente derivanti dal fatto che non esistono anticorpi per la maggior parte delle proteine.
Più recentemente ci si sta spingendo verso altri tipi di molecole da usare come sonde, quali peptidi di piccole, medie e grandi dimensioni.
Gli anticorpi, comunque, rappresentano ancora ad oggi, la sonda più efficace per i protein microarray.
I protein microarray (detti anche biochip, proteinchip) sono utilizzati nelle applicazioni biomediche per determinare la presenza e/o la quantità di proteine in campioni biologici, ad esempio nel sangue.
Anche se i protein microarray usano metodi di rilevazione simili ai DNA microarray, i protein microarray presentano un altro problema: la concentrazione di diverse proteine in un campione biologico può presentare molti ordini di grandezza di differenza da quello degli mRNA.
Di conseguenza, i metodi di rilevazione dei protein microarray devono avere una gamma molto più vasta di rilevazione.
Il metodo preferito di rilevazione rimane comunque sempre la rilevazione per fluorescenza, poiché è sicuro, sensibile e può dare alte risoluzioni

3.5. Microarrays e bioinformatica

3.5.1. Standardizazzione

La mancanza di standardizazione negli arrays presenta un problema interoperativo nella bioinformatica, che non può far prescindere dallo scambio di dati ottenuti con tale tecnica. Diversi progetti open-source si prefiggono di facilitare l'interscambio di dati ottenuti da arrays [9].
Il "Minimum Information About a Microarray Experiment" (MIAME) XML standard base per la descrizione di esperimenti di microarray è stato adottato da molte pubblicazioni scientifiche come standard richiesto per l'accettazione di lavori che contengono risultati ottenuti attraverso analisi di microarray.

3.5.2. Analisi statistica

L'analisi di DNA microarray propone numerosi problemi di carattere statistico, compresa la normalizzazione dei dati.
Analizzando il metodo dei microarray pare evidente che, il grande numero di geni presenti in un singolo array pone lo sperimentatore davanti ad un problema di test multiplo: anche se è estremamente raro e casuale ogni gene può dare un risultato falso positivo, mentre un test effettuato su più geni mostra sicuramente un andamento scientificamente più probante.
Una delle differenze fondamentali tra i microarray e gli altri metodi di analisi biomedici tradizionali sta nella dimensione dei dati.
Studi che contengono 100 analisi per paziente per 1000 pazienti possono essere considerati vasti studi clinici.
Uno studio microarray di media vastità comprende diverse migliaia di dati per campione su centinaia di campioni diversi.
Diventa quindi di rilevante importanza l'utilizzo della statistica nell'analisi dei microarray.
I metodi principali utilizzati sono due: numerico e grafico.
Il metodo grafico è il migliore per identificare pattern nei dati. L'approccio numerico, invece, è più preciso e obiettivo. Le due metodologie si completano a vicenda, ed entrambe sono necessarie.

3.5.3. Relazione tra gene e probe

La relazione tra probe e mRNA è molto semplice ma nello stesso tempo complessa. Il probe ha alta affinità con una singola sequenza (quella complementare), ma può legare altre sequenze non prettamente complementari. Ciò potrebbe portare a dati errati.

Capitolo 4
Data mining

I dati derivanti dai microarray, come tutti i dati delle basi dati sparse nel mondo, sono affetti da rumore e valori mancanti, che ne rendono difficile l'analisi successiva e tutti i relativi processi di data mining.
In questo capitolo si analizzeranno inizialmente le procedure di preprocessing e di normalizzazione dei dati derivanti dai microarray [12]; nella seconda parte la teoria alla base delle tecniche utilizzate successivamente negli esperimenti.
Vedremo quindi la classificazione, con i suoi vari metodi, e il clustering.

4.1. Preprocessing

I dati che si ottengono dall'analisi dei microarray possono essere affetti da rumore oppure essere incompleti [9].
Le ragioni di questi errori nei dati possono essere dovute a molte cause.
Nei microarray molto spesso mancano dei dati. Infatti bisogna ricordarsi che i valori nei microarray derivano da osservazioni dell'intensità, che possono quindi mancare in alcuni casi.
Si definiscono pertanto in questo contesto valori mancanti, quelli che non ci sono perchè:

- lo spot è vuoto
- l'intensità di background è maggiore dell'intensità dello spot

Questi valori mancati possono produrre errori nell'analisi dei dati, perché interferiscono facilmente con i calcoli statistici e il clustering.
Il rumore nei dati dei microarray è dovuto essenzialmente alla procedura di calcolo del valore di espressione.
Poiché la procedura di assimilazione dei valori si basa su un metodo visivo e sulla differenza di intensità del background con il foreground, può succedere che intervengano dei fenomeni per i quali le misurazioni vengano falsate.

L'importanza quindi di preprocessare i dati diventa chiara.
Nel data mining esistono varie procedure di "pulizia" dei dati, non sempre applicabili nell'ambito dell'analisi dei microarray [11].
I passi principali sono comunque:

- **data cleaning**, ossia una pulizia dei dati eliminando o riempiendo i valori mancanti, togliendo il più possibile il rumore, identificando o rimuovendo gli outlier e risolvendo le inconsistenze
- **data transformation**, come normalizzazioni e aggregazioni
- **data reduction**, che ottiene una rappresentazione ridotta del data set

4.1.1. Data cleaning

Esistono vari metodi per il data cleaning.
Si analizzeranno i metodi esistenti, focalizzando l'attenzione su quelli applicabili al contesto dei microarray.
Uno dei primi problemi che bisogna risolvere per analizzare i dati è il trattamento dei valori mancanti.
Esistono varie soluzioni nel data mining, ma purtroppo non tutte applicabili ai microarray.
Le uniche soluzioni perseguibili sono:

- rimpiazzare il dato mancante con un valore stimato
- cancellare il dato mancante in modo definitivo, anche per le future analisi

Nel primo caso il valore stimato può essere:

- la media tra tutte le misurazioni effettuate
- la media per quella data classe, ossia la media tra tutte le misurazioni derivanti dai campioni che appartengono alla stessa classe, ad esempio malato/sano
- il valore più probabile, che può essere determinato con la regressione, alberi decisionali o tool basati sul formalismo Bayesiano

Quest'ultimo metodo potrebbe non essere del tutto corretto, anche se è una strategia diffusa.
In confronto agli altri metodi, utilizza più informazione dai dati presenti per predirre il valore mancante.
Considerando i valori degli altri attributi nella sua stima del valore mancante, esiste una grossa probabilità che le relazioni tra il valore che verrà immesso e gli attributi siano preservate.
Per quanto riguarda la cancellazione del dato, che nel caso del microarray significa eliminare dalle successive analisi un gene, molto dipende dalla percentuale dei valori mancanti; infatti se la percentuale diventa troppo alta si rischia di perdere delle informazioni importanti. Pertanto bisogna prestare molta attenzione quando si applica questa procedura.

4.1.2. Data transformation

In questa fase i dati vengono trasformati o consolidati in forme appropriate per l'analisi. Le tecniche utilizzate comunemente sono:

- **Smoothing**: lavora per rimuovere il rumore dai dati. Alcune tra le tecniche utilizzate sono binning, clustering e regressione
- **Aggregazione**: dove operazioni di aggregazione o di sommarizzazione sono applicate ai dati. Questo tipo di operazione viene di solito scartata per i microarray poiché i dati interessanti non possono essere aggregati
- **Generalizzazione**: i livelli bassi o primitivi dei dati sono rimpiazzati da concetti di alto livello attraverso l'uso di gerarchie. Anche questa tecnica è di difficile utilizzo per i dati dei microarray
- **Normalizzazione**: dove gli attributi sono scalati in modo da ricadere in un intervallo piccolo specificato, come per esempio tra -1.0 e 1.0 o tra 0.0 e 1.0
- **Costruzione di un attributo**: detto anche feature construction, dove nuovi attributi sono costruiti o inseriti nel set degli attributi per aiutare il processo di data mining

La tecnica maggiormente utilizzata per i microarray è la normalizzazione che verrà analizzata più avanti.

4.1.3. Data reduction

Nell'analisi dei microarray la tecnica della riduzione è di difficile applicazione, se non addirittura impraticabile.
Riportiamo comunque di seguito le principali tecniche conosciute, dal momento che non si può affermare a priori se queste verranno applicate in futuro da alcuni algoritmi di analisi dei microarray.
Questa procedura può essere applicata per ottenere una rappresentazione ridotta del data set, che è molto ridotta in volume, ma mantiene l'integrità dei dati originali.
In questo modo i risultati analitici potrebbero essere più efficienti che sull'intera mole di dati.
Le strategie utilizzate comunemente sono:

- **Data cube reduction**: le operazioni di aggregazione sono applicate ai dati per la costruzione di un data cube
- **Dimension reduction**: gli attributi o le dimensioni irrilevanti, poco rilevanti o ridondanti vengono individuati e rimossi
- **Data compression**: meccanismi di codifica vengono usati per ridurre la dimensione dei dati
- **Numerosity reduction**: i dati sono rimpiazzati da alternative rappresentazioni più piccole come modelli parametrici o non parametrici come il clustering, il campionamento o l'utilizzo di istogrammi

- **Discretizzazione e generazione di concetti gerarchici**: i valori dei dati vengono sostituiti da range di valori o da livelli concettuali più alti. I concetti gerarchici permettono l'analisi a livelli multipli di astrazione e sono un potente mezzo per il data mining

4.2. Normalizzazione

Ci sono molte sorgenti di rumore che possono intervenire nelle misure dei valori di espressione dei geni. La normalizzazione serve proprio per ridurre il rumore presente nei dati misurati [13].
Un attributo è normalizzato quando si scala il suo valore in modo che ricada in un intervallo piccolo specificato, come 0.0-1.0.
La normalizzazione è particolarmente utile per gli algoritmi di classificazione che coinvolgono le reti neurali, o le misure d'istanze come i metodi di classificazione KNN (K nearest neighbor) e il clustering.
Per esempio se si utilizzano reti neurali per la classificazione con algoritmo di back-propagation, la normalizzazione dei valori di input, per ogni attributo misurato nei campioni di training, aiuterà a velocizzare la fase di apprendimento.
Esistono vari metodi per la normalizzazione dei dati:

- Min-max normalization
- Z-score normalization
- Normalization by decimal scaling
- Log-transformation

Di solito nel campo dei microarray si lavora meglio con una normalizzazione di log-transformation.
Questa trasformazione del rapporto delle intensità è buona per diversi motivi.
Il più rilevante è che il semplice rapporto fluttua per tutti i geni down regolati tra 0 e 1. La trasformazione logaritmica rimuove questo bias.
In termini statistici, questa trasformazione dà ai dati un senso di variazione più realistico, rendendo la variazione delle intensità e del rapporto delle intensità molto più indipendente dalle magnitudini assolute. Stabilizza inoltre la varianza degli spot ad alta intensità.

4.2.1. Trasformazione logaritmica

I dati dei microarray sono di solito valutati attraverso un rapporto.
Questo può essere un rapporto tra due condizioni dello stesso array, o il rapporto tra i valori assoluti derivanti da un esperimento single-dye.

Tuttavia, anche se i rapporti forniscono una misura intuitiva dei cambiamenti di espressione, presentano lo svantaggio di trattare diversamente i geni up o down regolati.
I geni up regolati di un fattore di 2 hanno un rapporto di espressione di 2, mentre quelli down regolati dallo stesso fattore hanno un rapporto di espressione di 0.5.
Ciò risulta in un grafico dove i geni up regolati hanno un range di valori molto più ampio di quelli down regolati (grafico di sinistra).

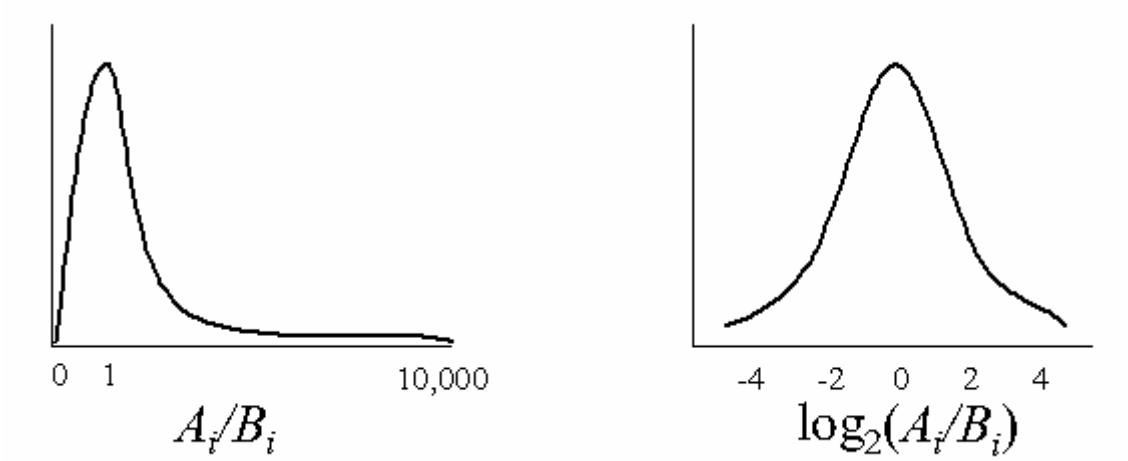

Figura 4.1 - Rapporto dei dati di microarray prima (sinistra) e dopo (destra) della trasfomazione logaritmica. Ai e Bi sono le misure di intensità per l'i-esimo elemento dell'array

Si ricordi che i logaritmi trattano simmetricamente i numeri ed i loro reciproci: $\log_2(1) = 0$, $\log_2(2) = 1$ e $\log_2(1/2) = -1$, $\log_2(4) = 2$ e $\log_2(1/4) = -2$ e così via.
Il logaritmo dei rapporti di espressione inoltre è trattato simmetricamente in modo che un gene sovraespresso di un fattore 2 abbia un $\log_2$(ratio) di 1, un gene sottoespresso di un fattore di 2 abbia un $\log_2$(ratio) di -1 e un gene espresso ad un livello costante (con un rapporto pari a 1) abbia $\log_2$(ratio) uguale a zero.
Il risultato di una trasformazione logaritmica consiste nel fatto che, i dati con una distribuzione asimmetrica positiva, sono trasformati in una distribuzione più simmetrica intorno allo 0 (solitamente generando una divisione normale).
Ciò significa che un grafico è generato là dove i geni up e down regolati sono trattati in modo simile, entrambi usando parti simile del grafico (grafico di destra).
Un altro risultato della trasformazione logaritmica è la diminuzione dell'influenza di valori molto alti sul valore medio o mediano, poiché, con questo tipo di trasformazione, si otterranno valori più piccoli.
I piccoli valori saranno maggiormente sparsi e avranno maggiore influenza.
La scelta di $\log_{10}$, di $\log_e$ o di $\log_2$ dipendente dall'utente, anche se, per convenienza d'interpretazione della scala, i biologi spesso preferiscono scale $\log_2$.

4.3. Classificazione

La classificazione è una forma di analisi dei dati che può essere utilizzata per estrarre modelli che descrivono le classi dei dati; quindi la classificazione predice le etichette di categorie che verranno applicate ai dati [10].
Nel caso specifico, ad esempio, la classificazione determina se un paziente è malato o è sano, oppure se ha un tipo di malattia piuttosto che un altro.
La classificazione dei dati avviene in un processo a due passi.
Nel primo passo, un modello è costruito per descrivere un set di dati in classi o concetti. Il modello è costruito analizzando le tuple del database descritte dagli attributi. Ogni tupla è assunta appartenente a una classe predefinita, determinata da un attributo specifico, che è chiamato attributo di etichetta di classe.
Nel contesto della classificazione, le tuple di dato sono anche chiamate campioni, esempi od oggetti.
L'insieme di tuple utilizzate per costruire il modello, formano il training data set.
Le tuple singole che costituiscono il training set sono definite come training sample e sono selezionate in modo casuale dalla popolazione dei campioni.
Fino a che l'etichetta di classe di ogni training sample è fornita, questo passo è anche chiamato supervised learning.
Ciò si pone in contrasto con l'unsupervised learning (o clustering), nel quale, a priori, non si conoscono le etichette di classe di ogni training sample e il numero o il set di classi da apprendere.
Tipicamente, il modello appreso è rappresentato sotto forma di regole di associazione, alberi decisionali o formule matematiche.
Nel secondo passo il modello è usato per la classificazione.
Si stima, innanzitutto, l'accuratezza predittiva del modello o classificatore.
L'accuratezza di un modello su un dato test set è la percentuale di campioni del test set che sono correttamente classificati dal modello stesso.
Per ogni campione di test, l'etichetta di classe conosciuta è comparata con la predizione della classe per quel campione, fornita dal modello.
Si noti che, se l'accuratezza del modello è stimata basandosi sul training data set, questa stima potrebbe essere ottimistica, fino a che il modello costruito tende a fare overfit dei dati (ciò significa che potrebbe avere incorporato alcune particolari anomalie del training set che non sono presenti in tutta la popolazione). E' consigliabile quindi un test set.
Se l'accuratezza del modello è considerata accettabile, il modello può essere utilizzato per classificare future tuple di dato o oggetti per i quali la classe non è conosciuta.

4.3.1. Comparare i metodi di classificazione

I metodi di classificazione possono essere comparati e valutai in accordo con i seguenti criteri:

- **Accuratezza**: si riferisce all'abilità del modello di predire correttamente l'etichetta di classe di un nuovo dato
- **Velocità**: si riferisce ai costi computazionali coinvolti nella generazione e nell'utilizzo del modello
- **Robustezza**: l'abilità del modello a produrre predizioni corrette su dati rumorosi o con valori mancanti
- **Scalabilità**: si riferisce all'abilità nel costruire un modello efficiente data una gran quantità di dati
- **Interpretabilità**: si riferisce al livello di comprensione che il modello fornisce

4.3.2. Decision tree

Un decision tree è un flow-chart con struttura ad albero, dove ogni nodo rappresenta un test su un attributo, ogni ramo rappresenta un'uscita del test, e le foglie rappresentano le classi o le distribuzioni delle classi.
Per classificare un campione sconosciuto, i valori degli attributi del campione vengono testati in base al decision tree. Si traccia quindi un percorso dalla radice a una foglia che indica la predizione di classe per quel campione.
Gli alberi decisionali possono essere facilmente convertiti in regole di classificazione.

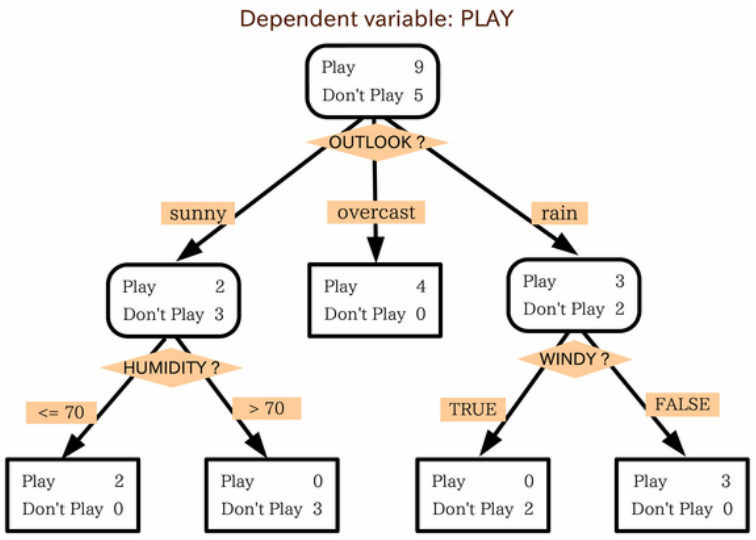

Figura 4.2 - Esempio di decision tree

La conoscenza rappresentata in un albero decisionale può essere estratta e rappresentata nella forma della regola di classificazione IF-THEN. Una regola è creata per ogni percorso dalla radice a una foglia. Ogni arco lungo il percorso forma una congiunzione nella regola. Ogni foglia indica la predizione di classe, formando così la conseguenza della regola (parte del THEN).
Le regole IF-THEN possono essere di più facile comprensione per l'essere umano, in particolar modo se l'albero decisionale è grande.
Quando gli alberi decisionali vengono costruiti, molte delle ramificazioni possono rappresentare rumore o outliers nei dati di training.
Il tree pruning tenta di identificare e rimuovere questi rami con l'obbiettivo di migliorare l'accuratezza di classificazione sui dati futuri.
I metodi tipicamente utilizzati sono misure statistiche.

4.3.3. Classificazione Bayesiana

I classificatori Bayesiani sono dei classificatori statistici.
Possono predire la probabilità di appartenenza ad una classe, come la probabilità che un dato campione appartenga a una particolare classe.
Questi classificatori sono basati sul teorema di Bayes, descritto di seguito.
Alcuni studi comparando algoritmi di classificazione hanno trovato che i classificatori Bayesiani, conosciuti come naive Bayesian, possono essere comparabili nelle performance con gli alberi decisionali e le reti neurali. Inoltre essi hanno la capacità di avere un'alta accuratezza e velocità quando vengono applicate a grandi database.
Questa tipologia di classificatori presume che l'effetto di un valore di un attributo su una data classe è indipendente dai valori degli altri attributi. Ciò è fatto per semplificare i calcoli coinvolti e, in questo senso, è considerato "naive".
Il teorema di Bayes è il seguente:

$$P(H \mid X) = \frac{P(X \mid H)P(H)}{P(X)}$$

dove P(H|X) è la probabilità a posteriori di H condizionato da X, mentre P(X) e P(H) sono le probabilità che si verifichino gli eventi H e X.

4.3.4. Classificatori k-Nearest Neighbor

Questi classificatori sono basati sull'apprendimento per analogia.
I campioni di training sono descritti da attributi numerici n-dimensionali. Ogni campione rappresenta un punto in uno spazio n-dimensionale.

In questo modo tutti i campioni del training set sono immagazzinati in uno spazio pattern n-dimensionale. Quando si ha un campione sconosciuto, un classificatore k-nearest neighbor cerca lo spazio pattern per k campioni di training che sia il più vicino al campione esaminato.
La vicinanza è definita in termini di distanza Euclidea, dove per distanza euclidea tra due punti, $X = (x_1, x_2, \ldots, x_n)$ e $Y = (y_1, y_2, \ldots, y_n)$ si intende

$$d(X,Y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$$

Il campione sconosciuto è assegnato alla classe maggiormente comune tra i k vicini. Quando k vale 1, il campione sconosciuto è assegnato alla classe del campione del training set più vicino.
Questi classificatori sono chiamati anche istance-based o lazy learners poiché immagazzinano tutti i campioni del training set e non costruiscono un classificatore fino a che un nuovo campione non deve essere classificato.
Ciò contrasta con il metodo eager learning, quale il decision tree, che costruisce un modello generalizzato prima di ricevere i nuovi campioni da classificare.
I lazy learners possono incorrere in costi computazionali eccessivi quando il numero di potenziali vicini, con i quali bisogna compare il campione dato non etichettato, è grande. Richiedono quindi una tecnica efficiente di indicizzazione.
Ovviamente i metodi lazy learning sono più veloci nella fase di apprendimento rispetto agli eager, ma più lenti nella classificazione a causa del tempo di tutti i calcoli effettuati.
Il nearest neighbor, a differenza dell'albero decisionale, assegna un peso uguale a ogni attributo. Questo può causare confusione quando ci sono molti attributi irrilevanti nei dati.

4.4. Clustering

Il clustering è il processo di raggruppamento dei dati in classi o cluster, in modo che gli oggetti in un cluster abbiano una grande similarità nel confronto con gli altri, ma abbiano una grande dissimilarità con gli oggetti degli altri cluster.
Le differenze sono valutate sulla base dei valori degli attributi che descrivono l'oggetto. Spesso sono utilizzate le misure di distanze.
Il clustering ha le sue radici in vari settori, dal data mining alla statistica, fino ad arrivare alla biologia.
Nel data mining l'obbiettivo principale è di trovare metodi che rendano efficiente e effettiva l'analisi di cluster in database molto grandi.

Nel data mining i requisiti tipici per il clustering sono:

- **Scalabilità**: molti algoritmi di clustering lavorano su data set molto piccoli. Sono necessari perciò algoritmi che abbiano una grande scalabilità, ossia che possano trattare con milioni di oggetti
- **Abilità di trattare con differenti tipi di attributi**: molti algoritmi attuali sono progettati su clustering basati su dati numerici. Nonostante ciò alcune applicazioni hanno bisogno di metodi che trattano anche attributi binari, nominali e ordinali.
- **Scoprire cluster con forme arbitrarie**: molti algoritmi determinano cluster basandosi sulle distanze euclidee o di Manhattan. Questi tipi di algoritmi tentano quindi di trovare cluster sferici con dimensione e densità simili.
- **Requisiti minimi di conoscenza del dominio per determinare i parametri di input**: molti algoritmi di clustering richiedono all'utente di immettere alcuni parametri per l'analisi. I risultati risultano quindi sensibili alla variazione di questi parametri.
- **Abilità di lavorare con dati rumorosi**: alcuni degli algoritmi presenti sono sensibili ai dati rumorosi e agli outlier presenti nei database, influenzando così la qualità del risultato.
- **Insensibilità all'ordine dei record in input**: alcuni algoritmi sono sensibili all'ordine di immissione dei dati. E' importante che ciò non accada onde non ottenere risultati differenti a seconda dell'ordine dei dati
- **Alta dimensionalità**: gli algoritmi di clustering funzionano bene di solito con dati che contengono poche dimensioni (due o tre al massimo).
- **Clustering sotto requisiti**: alcune volte è importante imporre dei requisiti sull'analisi dei dati.
- **Interpretabilità e utilizzabilità**: gli utenti si aspettano che i risultati di un clustering siano interpretabili, comprensibili e utilizzabili.

In letteratura esistono un gran numero di algoritmi di clustering. La scelta dell'algoritmo dipende sia dal tipo di dato disponibile, sia dall'obbiettivo da raggiungere e dall'applicazione dello stesso.
I maggiori metodi di clustering possono essere classificati nelle seguenti categorie:

- **Metodi di partizionamento**: Dato un database di n oggetti o tuple, questo metodo costruisce k partizioni di dati, dove ogni partizione rappresenta un cluster e $k \leq n$. Quindi classifica i dati in k gruppi, che soddisfano tutti i seguenti requisiti
  - ogni gruppo deve contenere almeno un oggetto
  - ogni oggetto deve appartenere solamente a un gruppo

  Dato k, il numero di partizioni da costituire, un metodo di partizionamento crea un partizionamento iniziale; utilizza una tecnica di rilocazione iterativa che tenta di migliorare il partizionamento muovendo gli oggetti da un gruppo a un altro.
  Il criterio generale per un partizionamento è che gli oggetti nello stesso cluster siano vicini o relazionati tra loro, mentre gli oggetti di cluster differenti siano distanti o molto differenti.

- **Metodi gerarchici**: Questa tipologia di clustering crea una decomposizione gerarchica degli oggetti del data set.
  Un metodo gerarchico può essere classificato come agglomerativo o divisivo, a seconda di come la decomposizione è fatta.
  Un approccio agglomerativo, chiamato anche bottom-up, parte dal presupposto che ogni singolo oggetto forma un gruppo. Successivamente unisce gli oggetti o i gruppi vicini, fino a che tutti i gruppi siano uniti in uno unico o fino a che non si raggiunga una condizione di terminazione.
  L'approccio divisivo, chiamato anche top-down, parte con tutti gli oggetti all'interno dello stesso cluster. A ogni successiva iterazione, un cluster viene diviso in cluster più piccoli, fino a che eventualmente ogni oggetto sia un cluster o fino a che non si raggiunga una condizione di terminazione
- **Metodi basati sulla densità**: Molti metodi di partizionamento sono basati sulla distanza degli oggetti. Metodi del genere possono trovare solo cluster con forma sferica e trovano difficoltà a scoprire clustering di forma arbitraria.
  Si sono quindi costruiti algoritmi basati sul concetto di densità.
  Il metodo è di continuare a far crescere un dato cluster fino a che la densità (numero di oggetti o punti) nel "vicinato" ecceda una data soglia. In questo modo, per ogni punto all'interno di un dato cluster, il vicinato di un dato raggio deve contenere almeno un numero di punti minimo.
- **Metodi basati sulla griglia**: Questi metodi quantizzano lo spazio degli oggetti in un numero finito di celle che formano una struttura a griglia. Tutte le operazioni di clustering sono fatte sulla struttura a griglia.
  Il principale vantaggio di questo approccio è la sua velocità di processamento, che è tipicamente indipendente dal numero di oggetti e dipendente solamente dal numero di celle in ogni dimensione nello spazio quantizzato.
- **Metodi basati su modelli**: Questi algoritmi ipotizzano un modello per ognuno dei cluster e trovano il migliore per rappresentare i dati di un dato modello.
  Di solito localizza i cluster costruendo una funzione di densità che riflette la distribuzione spaziale dei dati.

Alcuni algoritmi di clustering integrano queste idee, in modo tale che risulta difficile classificarli sotto un'unica categoria.
D'altra parte alcune applicazioni hanno dei criteri di clustering che richiedono l'integrazione di molte tecniche.

Capitolo 5
Soluzioni scelte sviluppate

In questo capitolo vengono illustrati i vari algoritmi e programmi che sono stati utilizzati per le analisi dei dati che verranno descritti nel Capitolo 6.
Siccome i dati analizzati appartengono a due categorie diverse e hanno caratteristiche intrinseche differenti, alcuni algoritmi sono stati applicati a una categoria di dati, mentre altri all'altra categoria.
Le prove effettuate e i risultati ottenuti verranno esposti rispettivamente nel Capitolo 6 e nel Capitolo 7.

5.1. Pvclust

Pvclust è un pacchetto aggiuntivo per il software statistico R che permette di valutare l'incertezza nell'analisi gerarchica di cluster [28].
Può essere usato facilmente per problemi statistici generali, quale analisi di dati da microarray, per effettuare l'analisi del bootstrap di clustering, molto diffuso nell'analisi filogenetica.

5.1.1. Algoritmo

Per ogni cluster nel clustering gerarchico, vengono calcolate le quantità denominate p-*value* attraverso il ricampionamento multiscala del bootstrap.
Il *p-value* di un cluster è un valore fra 0 e 1, che indica quanto fortemente il cluster è sostenuto dai dati.
Il pvclust fornisce due tipi di *p-value*: *p-value dell'*AU (approssimativamente imparziale) e il valore di BP (probabilità del bootstrap).

Il *p-value dell'*AU, che è calcolato dal ricampionamento del bootstrap multiscala, è un'approssimazione migliore al *p-value* imparziale, che non il valore di BP calcolato dal ricampionamento del bootstrap normale.
Il pvclust effettua l'analisi gerarchica dei cluster attraverso la funzione "hclust" e automaticamente calcola i *p-value* per tutti i cluster contenuti nel clustering dei dati originali. Fornisce inoltre tool grafici, quali la funzione "plot" e "pvrect". Quest'ultima funzione evidenzia le serie di cluster con p-value relativamente alti o bassi.
Il tempo di calcolo dell'intero algoritmo può essere enormemente diminuito grazie all'opzione di calcolo in parallelo.

5.2. PAMR

Il metodo PAMR è un approccio per la predizione delle classi dai valori di espressione dei geni, basati su un classificatore dei centroidi più vicini. Il metodo PAMR identifica un subset di geni che caratterizzano meglio ogni classe.
La tecnica è generica e può essere usata in molti altri problemi di classificazione.
Questo metodo calcola un centroide standardizzato per ogni classe [15]. Il centroide standardizzato è la media dei valori di espressione per ogni gene in ogni classe, divisa per la deviazione standard interna della classe per quel gene.
Ai geni dei centroidi standardizzati è sottratto un offset detto soglia, il cui valore è definito dall'utente. Geni il cui valore originale fosse minore della soglia sono impostati a zero.
Di seguito viene illustrato l'algoritmo passo dopo passo.

5.2.1. Metodo

Sia x_{ij} l'espressione per i geni i = 1, 2, …p e i campioni j = 1,2,…n. Abbiamo classi 1, 2, …K, e siano C_k gli indici dei n_k campioni nella classe k. L'i-esimo componente del centroide per la classe k è $\bar{x}_{ik} = \sum_{j \in C_k} x_{ij} / n_k$, la media del valore di espressione nella classe k per il gene i; l'i-esimo componente di tutti i centroidi è $\bar{x}_i = \sum_{j=1}^{n} x_{ij} / n$.

In parole, restringiamo i centroidi di classe verso il centroide generale dopo aver standardizzato con la deviazione standard della classe per ogni gene. Questa standardizzazione ha l'effetto di dare un alto peso ai geni la cui espressione è stabile nei campioni della stessa classe. La standardizzazione è inerente ad altri comuni metodi statistici come l'analisi lineare discriminante.

Sia

$$d_{ik} = \frac{\bar{x}_{ik} - \bar{x}_i}{m_k \cdot (s_i + s_0)}, \qquad (1)$$

dove si è la deviazione standard della classe per il gene i:

$$s_i^2 = \frac{1}{n-K} \sum_k \sum_{j \in C_k} (x_{ij} - \bar{x}_{ik})^2, \qquad (2)$$

e $m_k = \sqrt{1/n_k + 1/n}$ fa $m_k \cdot s_i$ uguale allo standard error stimato del numeratore in d_{ik}.
Al denominatore, il valore s_0 è una costante positiva (con lo stesso valore per tutti i geni). Incluso per salvaguardare dalla possibilità di larghi valori di d_{ik} cambiando tra geni con livelli di espressione basse. Settiamo s_0 uguale al valore mediano di si sul set di geni.
Una strategia simile fu usata in SAM.
Questo d_{ik} è una statistica t per il gene i, comparando la classe k con il centroide totale. Riscriviamo l'equazione (1) come:

$$\bar{x}_{ik} = \bar{x}_i + m_k (s_i + s_0) d_{ik} \qquad (3)$$

Il nostro metodo restringe ogni d_{ik} verso lo 0

$$\bar{x}'_{ik} = \bar{x}_i + m_k (s_i + s_0) d'_{ik} \qquad (4)$$

Il restringimento che si utilizza è chiamato *soft thresholding*: ogni d_{ik} è ridotto di una quantità Δ in valore assoluto ed è settato a zero se il suo valore assoluto è inferiore a zero. Algebricamente il soft thresholding è definito da:

$$d'_{ik} = sign(d_{ik})(|d_{ik}| - \Delta)_+, \qquad (5)$$

dove il + significa la parte positiva ($t_+ = t$ se $t>0$ e zero altrimenti).
Poiché molti dei valori $\bar{x}_{ik}$ possono essere rumorosi o vicini alla media generale $\bar{x}_i$, il soft thresholding di solito produce valutazioni più certe delle medie reali.
Questo metodo ha la proprietà di far sì che molti dei componenti (geni) sono eliminati dalla previsione di classe quando il parametro di restringimento Δ viene aumentato. Specificatamente, se per un gene i, d_{ik} è ristretto a zero per tutte le classi k, allora il centroide per il gene i è $\bar{x}_i$, lo stesso per tutte le classi. Questo gene i non contribuisce al calcolo del centroide più vicino.

5.3. GEMS

GEMS (Gene Expression Model Selector) è un programma che automatizza la costruzione e la valutazione di diversi classificatori basati su SVM (Support Vector Machine) e offre la possibilità di valutare le performance dei modelli costruiti [17].
Gli SVM sono, discutibilmente, lo sviluppo più importante nella classificazione supervisionata degli ultimi anni.
Gli SVM offrono spesso prestazioni superiori nella classificazione in confronto ad altre procedure di learning in molti domini e compiti; sono ragionevolmente insensibili ai problemi sulle dimensioni e sono abbastanza efficienti nel gestire la classificazione su una grande quantità, sia di campioni che di variabili.
Nella bioinformatica clinica, gli SVM hanno permesso la costruzione di potenti modelli sperimentali diagnostici del cancro basati su valori di espressione dei geni con migliaia di variabili e con poche dozzine di campioni.
Inoltre, parecchie implementazioni efficienti e di alta qualità dei SVM facilitano l'applicazione di queste tecniche nella pratica quotidiana.
La prima generazione dei SVM poteva essere applicata soltanto a compiti di classificazione binari. Tuttavia, la maggior parte dei compiti diagnostici nel mondo reale non sono binari. Inoltre, a parità di condizioni, la classificazione multicategoria è significativamente più difficile della classificazione binaria.
Negli ultimi anni sono emersi molti algoritmi che permettono la classificazione multicategoria con gli SVM.
Di seguito vengono illustrati vari algoritmi che sono implementati da questo programma. Nelle descrizioni seguenti k è il numero di classi o di categorie di diagnosi e n è il numero di campioni o pazienti presenti nel training data set.

5.3.1. SVM binario

L'idea principale del SVM binario è di mappare implicitamente i dati, in uno spazio dimensionale, attraverso una funzione kernel e poi risolvere un problema di ottimizzazione per identificare l'iperpiano del massimo margine che separa le istanze di learning.
L'iperpiano è basato su un insieme delle istanze di training di contorno, denominati vettori di sostegno.
I nuovi casi sono classificati a seconda della parte dell'iperpiano nella quale cadono (Figura 5.1).
Il problema di ottimizzazione è, molto spesso, formulato in modo da tener conto dei dati non separabili, penalizzando le classificazioni errate.

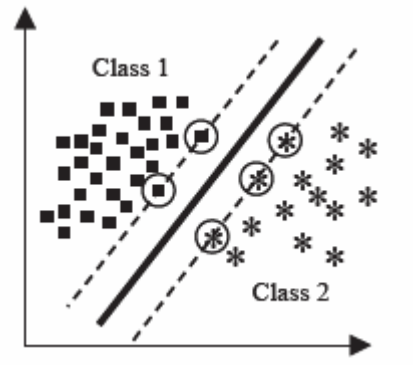

Figura 5.1 - Iperpiano generato dal SVM binario

5.3.2. SVM multiclasse: one-versus-rest (OVR)

Questo è concettualmente il metodo più semplice degli SVM multiclassi. Qui, costruiamo i k classificatori binari SVM: la classe 1 (positive) contro tutti le altre classi (negativi), la classe 2 contro tutte le altre, …, la classe K contro tutte le altre (Figura 5.2a).
La funzione combinata della decisione OVR sceglie la classe di un campione che corrisponde al massimo valore delle k funzioni di decisione binaria, specificate dall'iperpiano positivo. Così facendo, gli iperpiani di decisione calcolati tramite k SVM shift, mettono in discussione l'ottimizzazione della classificazione multicategoria.
Questo metodo è computazionalmente costoso, poiché bisogna risolvere k problemi di ottimizzazione di complessità quadratica (QP) di dimensione n.
Questa tecnica, inoltre, attualmente non ha una giustificazione teorica quale l'analisi di generalizzazione, che è una proprietà rilevante di un robusto algoritmo di learning.

5.3.3. SVM multiclasse: one-versus-one (OVO)

Questo metodo coinvolge la costruzione di classificatori binari SVM per tutti gli accoppiamenti di classi; in totale ci sono $\binom{k}{2} = [k(k-1)]/2$ accoppiamenti (Figura 5.2b). In altre parole, per ogni coppia di classi, un problema binario SVM è risolto (con il problema di ottimizzazione per massimizzare il margine tra due classi). La funzione di decisione assegna un'istanza a una classe che ha il più grande numero di voti, chiamata anche Max Wins strategy. Se i vincoli ancora persistono, ad ogni campione sarà assegnata un'etichetta basata sulla classificazione fornita da un ulteriore iperpiano.

Uno dei benefici di questo metodo è che per ogni accoppiamento delle classi ci si occupa di un problema molto più piccolo di ottimizzazione, e in totale bisogna risolvere k(k-1)/2 problemi di complessità quadratica di dimensione più piccola di n.
Poiché gli algoritmi di ottimizzazione di complessità quadratica usati per le SVM sono problemi di dimensione polinomiale, una tal riduzione può rendere un risparmio notevole nel tempo totale di calcolo.
Inoltre, alcuni ricercatori postulano che, anche se l'intero problema multicategoria è inseparabile, alcuni dei problemi binari secondari sono separabili, quindi il metodo OVO può condurre a un miglioramento della classificazione rispetto al metodo OVR.
Diversamente dal metodo OVR, in questo caso il rompere i legami svolge soltanto un ruolo minore e non influisce significativamente i contorni di decisione.

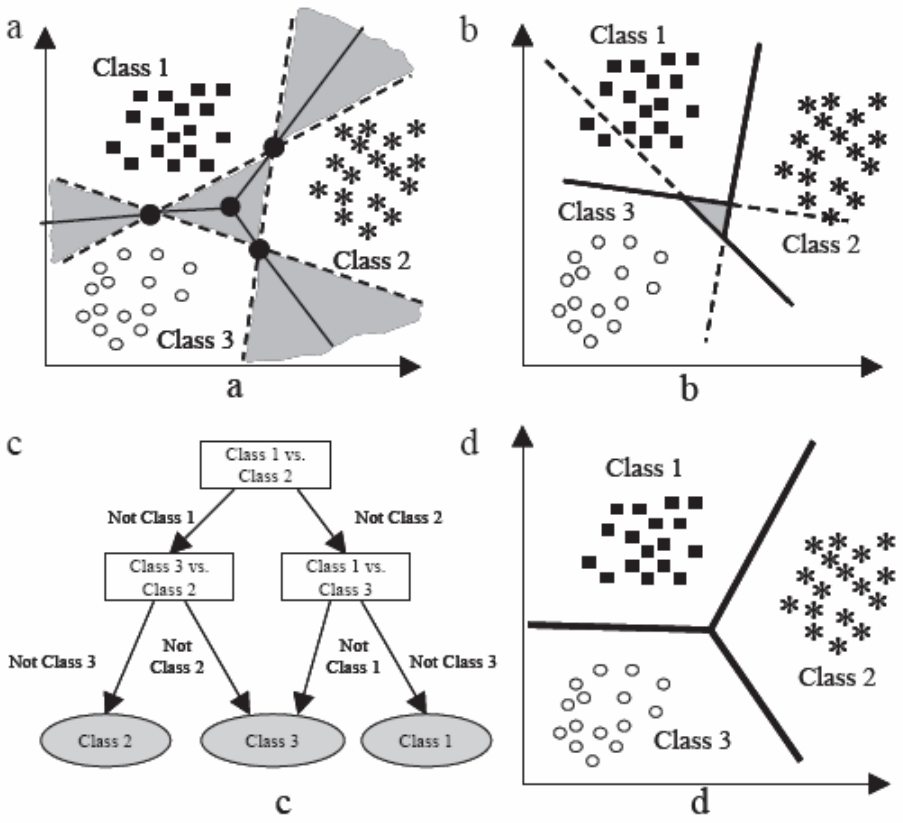

Figura 5.2 - Algoritmi SVM multiclasse applicati a un problema diagnostico su tre classi. (a) OVR costruisce 3 classificatori: (1) classe 1 verso classi 2 e 3; (2) classe 2 verso classi 1 e 3; (3) classe 3 verso classi 1 e 2. (b) OVO costruisce 3 classificatori: (1) classe 1 verso classe 2; (2) classe 2 verso classe 3; (3) classe 1 verso classe 3. (c) DAGSVM costruisce un albero decisionale sulla base di classificatori OVO. (d) WW e CS costruiscono un classificatore singolo massimizzando il margine tra tutte le classi simultaneamente

5.3.4. SVM multiclasse: DAGSVM

La fase di learning di questo algoritmo è simile all'approccio OVO utilizzando classificatori SVM binari multipli; tuttavia, la fase di testing del DAGSVM richiede la costruzione di un grafico di decisione diretta aciclico (DDAG) usando $\binom{k}{2}$ classificatori (Figura 5.2c).
Ogni nodo di questo grafo è un SVM binario per una coppia di classi, ad esempio (p, q).
Al livello topologico più basso ci sono k foglie che corrispondono alle k decisioni di classificazione. Ogni nodo non-foglia (p, q) ha due archi: l'arco di sinistra corrisponde alla decisione 'non p' e quello di destra corrisponde al 'non q'.
La scelta dell'ordine delle classi nella lista del DDAG può essere arbitraria.
Oltre ai vantaggi ereditati dal metodo OVO, DAGSVM è caratterizzato da un limite sull'errore di generalizzazione.

5.3.5. SVM multiclasse: metodo di Weston e Watkins (WW)

Questo approccio al SVM multiclasse è visto da alcuni ricercatori come estensione naturale del problema binario di classificazione SVM (Figura 5.2d).
Qui, nel k-esimo caso di classe bisogna risolvere un singolo problema quadratico di ottimizzazione di dimensione *(k-1)n* che è identico al SVM binario per il caso k = 2.
In una formulazione leggermente differente del problema di complessità quadratica, una formulazione limitata, tecniche di decomposizione possono fornire un significativo speed-up nella soluzione del problema di ottimizzazione.
Questo metodo non ha un limite stabilito sull'errore di generalizzazione e la sua ottimizzazione attualmente non è dimostrata.

5.3.6. SVM multiclasse: metodo di Crammer e Singer (CS)

Questa tecnica è simile alla WW (Figura 5.2d). Richiede la soluzione di un singolo problema di complessità quadratica di dimensione *(k-1)n*, comunque utilizza meno variabili nei vincoli del problema di ottimizzazione e quindi è computazionalmente più economico.
Simile al WW, l'uso di decomposizioni può fornire un significativo speed-up nella soluzione del problema di ottimizzazione. Purtroppo, l'ottimizzazione del CS, come pure i limiti sulla generalizzazione non sono ancora stati dimostrati.

5.3.7. Parametri per gli algoritmi di classificazione

I parametri per gli algoritmi di classificazione sono stati scelti da procedure annidate di cross-validation per ottimizzare le prestazioni evitando overfitting.
Per tutti e cinque i metodi di multiclasse SVM si è utilizzato un kernel polinomiale $K(x,y) = (\gamma \cdot x^T y + r)^p$, dove x e y sono i campioni con i valori di espressione dei geni e p, γ, r sono parametri del kernel.
Si calcola l'ottimizzazione del classificatore su un set di valori di costo C (il parametro di penalità degli SVM) = {0.0001, 0.01, 1, 100} e p = {1, 2, 3}.
I parametri del kernel γ e r sono stati settati a valori di default:

- γ = 1/numero delle variabili
- r = 0

5.3.8. Selezione dei geni

Per studiare come la riduzione delle dimensioni può migliorare le performance della classificazione, si possono applicare diverse funzioni di selezione.
I geni possono essere selezionati in base a quattro metodi:

- rapporto delle somme dei quadrati dei geni in categorie diverse con quelli della stessa categoria, BW
- valore del rapporto segnale rumore applicato in OVR (S2N_OVR)
- valore del rapporto segnale rumore applicato in OVO (S2N_OVO)
- ANOVA (KW)

Il ranking dei geni è eseguito sui campioni del training set anziché sull'intero data set per evitare overfitting.
Di seguito illustriamo in breve il calcolo che ogni algoritmo effettua per determinare il ranking di tutti i geni.
Il metodo denominato BW [23] basa il ranking sul calcolo del seguente rapporto:

$$\frac{BSS(j)}{WSS(j)} = \frac{\sum_i \sum_k I(y_i = k)(\bar{x}_{kj} - \bar{x}_{.j})^2}{\sum_i \sum_k I(y_i = k)(\bar{x}_{ij} - \bar{x}_{kj})^2}$$

dove $\bar{x}_{.j}$ denota il livello medio di espressione del gene j per tutti i campioni e $\bar{x}_{kj}$ denota la media del livello di espressione del gene j per tutti i campioni appartenenti alla classe k.
Il rank è costruito in modo tale che i geni con il rapporto BSS/WSS massimo sono i primi selezionati.

I metodi signal-to-noise (S2N) [22-24] calcolano la seguente statistica per ogni gene j:

$$S(j) = \frac{\mu_+(j) - \mu_-(j)}{\sigma_+(j) - \sigma_-(j)}$$

dove μ_+ e μ_- sono le medie delle classi +1 e -1 per il j-esimo gene. In modo analogo σ_+ e σ_- sono le deviazioni standard per le due classi per il j-esimo gene. I geni che danno maggiori valori positvi sono correlati con la classe +1, mentre i geni che danno maggiori valori negativi sono correlati con la classe -1. Si selezionano quindi m/2 geni positivi e m/2 negativi.
L'ultimo metodo utilizzato da GEMS, denominato ANOVA [16-30], si basa sul metodo Kruskal-Wallis one-way per l'analisi della varianza in base al rank. Questo metodo è definito non-parametrico, il che lo differenzia da quelli parametrici poiché la struttura del modello non è definita a priori, ma è determinata dai dati stessi. Ciò implica che la natura e il numero di parametri per il metodo sono flessibili e non determinati prima. Questi modelli vengono per questo motivo chiamati anche a distribuzione libera.
Intuitivamente si può capire che questo metodo è uguale al one-way ANOVA con la sola differenza che i dati sono rimpiazzati dal loro rank.
L'algoritmo si basa su tre passi:

1. determinare il rank di tutti i dati provenienti dai vari gruppi
2. Il test statistico è dato da:

$$K = (N-1)\frac{\sum_{i=1}^{g} n_i(\bar{r}_{i.} - \bar{r})^2}{\sum_{i=1}^{g}\sum_{j=1}^{n_i}(\bar{r}_{ij} - \bar{r})^2} \text{ , dove :}$$

   - n_g è il numero di osservazioni nel gruppo g
   - r_{ij} è il rank dell'osservazione j per il gruppo i
   - N è il numero totale delle osservazioni
   - $\bar{r}_{i.} = \frac{\sum_{j=1}^{n_i} r_{ij}}{n_i}$
   - $\bar{r}$ è la media di tutte le r_{ij}, uguale a (N+1)/2

3. infine il p-value è approssimato e calcolato il rank totale per i geni.

5.3.9. Misurazione delle performance

Si utilizzano due metriche per le performance della classificazione.
La prima unità di misura è l'accuratezza, per poter confrontare i risultati ottenuti sui dati degli sperimenti svolti rispetto a quelli pubblicati in precedenza. L'accuratezza è facile da interpretare e semplifica il testing statistico.
D'altra parte, l'accuratezza è sensibile alle probabilità a priori di classe e non descrive pienamente la difficoltà reale del problema di decisione per distribuzioni altamente sbilanciate.
Per esempio, è più difficile da raggiungere un'accuratezza del 50% per un dataset di 26 classi e di 14 tumori con la probabilità a priori della classe principale pari al 9,7% in confronto ad un'accuratezza del 75% per un dataset binario con probabilità a priori della classe principale uguale al 75,3%.
La seconda unità di misura è la *Relative Classifier Information* (RCI), che corregge le differenze nelle probabilità a priori delle categorie diagnostiche, così come il numero di categorie.
RCI è basata sull'entropia che misura quanto l'incertezza di un problema di decisione è ridotta da un classificatore rispetto alla classificazione basata solamente sulle probabilità a priori.

5.3.10. Relative Classifier Information

Il *Relative Classifier Information* è una valutazione delle performance di un classificatore basato sulla misura dell'entropia [26].
Consideriamo un classificatore costruito su un training set, l'RCI può essere calcolato matematicamente considerando:

- Matrice di confusione Q: le performance di un classificatore su un test dataset è memoriizata negli elementi di questa matrice. L'elemento q_{ij} è il numero di volte che un input del test set originariamente etichettato C_i viene etichettato C_j dal classificatore.
- In assenza del classificatore, con solo la conoscenza della distribuzione delle classi in input, si ha una determinata quantità di incertezza nel poter etichettare un nuovo dato. Ciò può essere usato come la misura della difficoltà di un problema di decisione. La probabilità che un input sconosciuto *(I)* appartenga alla classe con etichetta C_i è:
-

$$P(I \in C_i) = \frac{\sum_j q_{ij}}{\sum_{ij} q_{ij}}$$

L’incertezza associata quindi è:

$$H_d(I) = \sum_i - P(I \in C_i) \log P(I \in C_i)$$

- Dato che un input è stato etichettato dal classificatore Cj (variabilie di output O), la probabilità cha la sua originaria etichetta sia C_i è:

$$P(I \in C_i \mid O \in C_j) = p_{ij} = \frac{q_{ij}}{\sum_i q_{ij}}$$

L’incertezza sulla classificazione del campione in input dopo aver osservato l’output C_j è quindi:

$$H_{Oj}(I \mid O \in C_j) = \sum_i - p_{ij} \log p_{ij}$$

- La probabilità che O appartenga a C_j è:

$$P(O \in C_j) = p_j^{out} = \frac{\sum_i q_{ij}}{\sum_{ij} q_{ij}}$$

L'incertezza prevista nella classe di un input etichettato dato dalla classificazione fatta dalla macchina su un test set, può essere calcolata scomponendola nella probabilità relativa di appartenenza dei vari indicatori di classe:

$$H_O(I \mid O) = \sum_{ji} P(O \in C_j) H_{Oj}(I \mid O \in C_j)$$

- La quantità di incertezza nell'etichettare un campione in input risolto osservando la classificazione fatta dalla macchina è quindi:

$$H_{classifier} = H_d - H_O$$

L’RCI quindi può essere calcolato semplicemente come

$$RCI = H_{classifier} / H_d * 100$$

5.3.11. Utilizzo dell'interfaccia grafica

L'interfaccia grafica del sistema GEMS consiste di un singolo form con una barra menu nella parte superiore della finestra. Di seguito viene descritto brevemente il significato di ogni sezione indicata nel form in Figura 5.3 [18].

Sezione A. Questa sezione è utilizzata per specificare i file di input:

- Dataset (il dataset dei valori di espressione dei geni è in formato ASCII separato da tabulazioni o spazi, con le colonne che corrispondono ai geni/variabili ed le righe alle osservazioni, la prima colonna è la variabile target che è codificata con numeri interi a partire da 0. La prima riga indica il numero di campioni, di classi e di attributi)
- Nomi dei geni (file ASCII con la lista dei nomi dei geni - una linea per gene, la prima linea non è usata, la linea indica corrisponde alle colonne nel dataset)
- Numeri di accesso dei geni (file ASCII con la lista dei numeri di accesso dei geni - una linea per gene, la prima linea non è usata, l'indice di linea corrisponde alle colonne nel dataset)

L'utente deve specificare un dataset. I nomi dei geni ed i numeri di accesso (campi facoltativi) saranno utilizzati soltanto per la generazione del rapporto sperimentale in formato HTML.

Sezione B. Questa sezione è usata per selezionare il disegno sperimentale:

(1) N-fold cross-validation
(2) Leave-One-Out Cross-Validation (LOOCV).

Nel caso sia utilizzato N-fold cross-validation, è necessario immettere il numero di fold.

Sezione C. Questa sezione è usata soltanto quando il task sperimentale è di valutare le prestazioni del modello migliore (si veda la sezione M) ed è utilizzato il disegno di LOOCV. In questo caso, l'utente deve immettere il numero di fold della cross-validation per il ciclo interno del disegno di LOOCV.

Sezione D. Utilizzando questa sezione l'utente può

(1) generare a caso split stratificati di campioni per N-fold cross-validation e scartarli dopo gli esperimenti o conservarli in un file
(2) caricare split di campioni già generati

Gli split dei campioni sono salvate in un file ASCII, in cui ogni linea contiene gli indici dei campioni che partecipano ad un singolo fold (gli indici dei campioni sono delimitati da spazi).

Sezione E. Questa sezione è usata per selezionare i metodi di classificazione MC-SVM. Se l'utente seleziona gli algoritmi di classificazione multipla, il sistema ottimizzerà gli algoritmi per la cross-validation e deriverà una singolo algoritmo che rende più elevata l'accuratezza della cross-validation.

Sezione F. Questa sezione permette agli utenti di specificare una sequenza di valori di normalizzazione dei dati per ogni gene x del dataset. La normalizzazione è sempre realizzata basandosi solamente sul training dataset, in modo che i risultati finali non siano overfitted. Viene suggerito di utilizzare la normalizzazione "B" ($x \rightarrow [a, b]$ con $a = 0$ e $b = 1$) per accelerare la fase di training degli algoritmi MC-SVM. Si noti che si può avere la necessità di applicare la normalizzazione di $|x|$ prima di applicare il $\log(x)$ per assicurarsi che il dataset non contenga valori negativi.

Sezione G. Questa sezione è usata per specificare gli algoritmi di selezione dei geni. Se l'utente indica più procedure di selezione dei geni, il sistema effettuerà l'ottimizzazione degli algoritmi per la cross-validation e deriverà un singolo algoritmo che rende maggiori l'accuratezza della cross-validation.

Sezione H. Se sono impiegate tecniche di selezione dei geni, questa sezione permette di selezionare le cardinalità dei subset dei geni. Si può

(1) usare un subset di geni in formato fisso, o
(2) considerare subset multipli dei geni

ed il sistema deriverà un singolo subset di geni che rende maggiore l'aacuratezza della cross-validation.

Sezione I. Questa sezione è usata per selezionare una classe delle funzioni kernel per l'algoritmo SVM - funzioni di base polinomiale o radiale.

Sezione J. Questa sezione indica se è necessario ottimizzare i parametri del SVM per la cross-validation. Se l'ottimizzazione non è desiderata, l'utente deve immettere i valori del parametro di costo e i parametri di grado o gamma. Altrimenti, si deve immettere il range di input per l'ottimizzazione nella sezione K

Sezione K. Questa sezione contiene i range per l'ottimizzazione dei parametri di SVM per la cross-validation. Il sistema selezionerà una singola istanza dei parametri di costo e di grado o gamma che rende maggiore l'accuratezza della cross-validation.

Sezione L. Questa sezione è usata per specificare se il log è visualizzato a schermo o è salvato in un file.

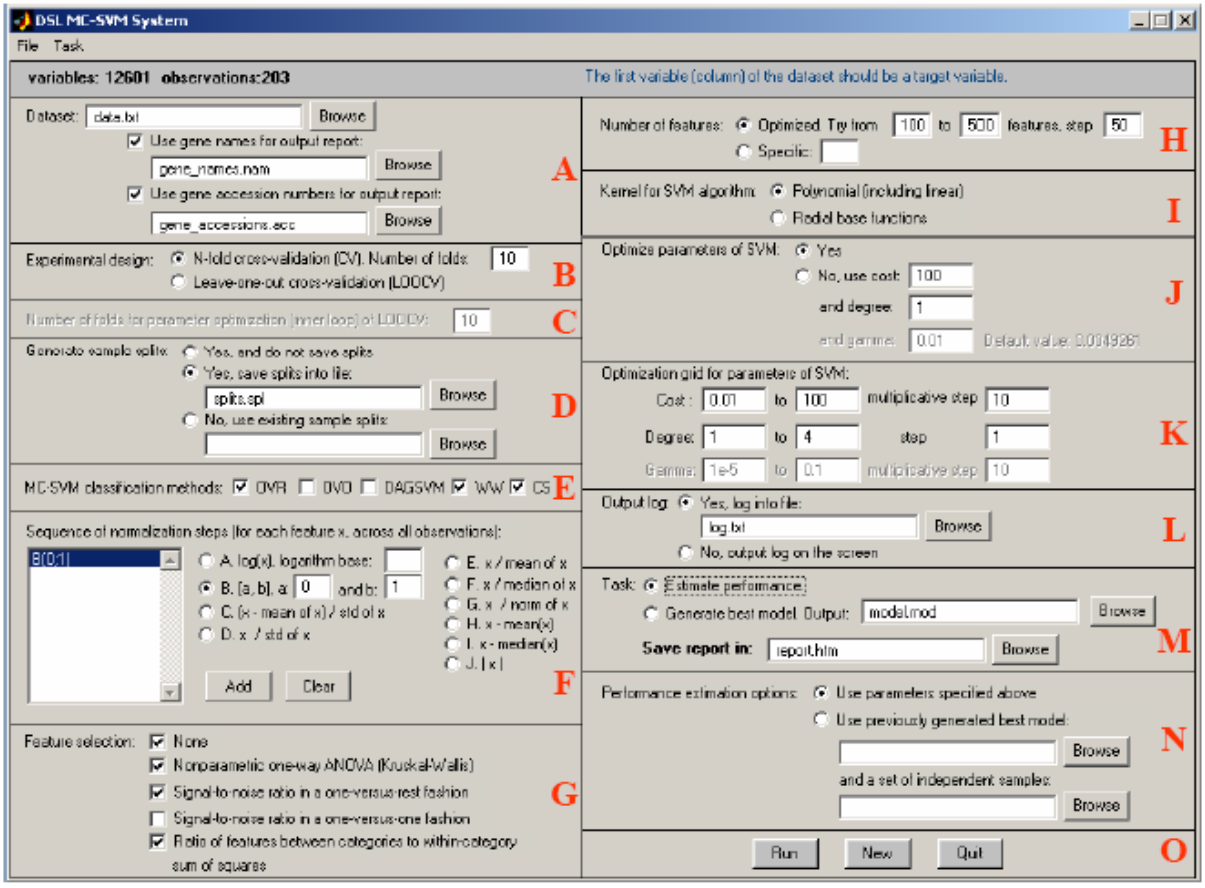

Figura 5.3 - Finestra del programma GEMS

Sezione M. Questa sezione è usata per selezionare un task sperimentale e specificare il file di output del rapporto. L'utente ha due opzioni:

(1) stimare le prestazioni, o
(2) generare il modello migliore e salvarlo.

Questa sezione inoltre contiene un campo per il file HTML di output per il rapporto.

Sezione N. Nel caso l'utente desideri valutare le prestazioni, questa sezione permette di:

(1) far funzionare l'intero esperimento con la selezione del modello, cioè valutare le prestazioni, o
(2) usare il modello migliore già generato e applicarlo ai nuovi campioni.

Nel secondo caso si deve specificare un modello e un dataset per il testing. Il testing dataset dovrebbe avere lo stesso formato (cioè dovrebbe contenere le stesse variabili nello stesso ordine) del dataset della sezione A. Se l'utente non desidera che il sistema calcoli

l'accuratezza finale, i valori della prima colonna (target) nel testing dataset devono essere sostituiti con un numero intero arbitrario.

Sezione O. Questa è la sezione di controllo dell'interfaccia di utente. Contiene tre bottoni:

- Run (stima la complessità dell'esperimento e lo esegue)
- New (resetta il form ai valori di default)
- Quit

Oltre alle sezioni descritte precedentemente, si può usare la barra del menu per aprire e salvare i file di progetto che possono anche essere creati o generati con un editor di testo.

5.4. mcSVM

Questo software, prodotto dall'Università di Genova, implementa l'apprendimento e la selezione di un modello di un Support Vector Machine (SVM) per un task multiclasse (tre classi o più).
E' stato sviluppato in Fortran95 ed è disponibile sia per la piattaforma Windows che per Unix.
Dal momento che SVM è un problema intrisicamente biclasse, questo tool utilizza un algoritmo particolare che viene presentato di seguito [19].

5.4.1. Metodo binario modificato per classificazioni multiclasse

Consideriamo un dataset composto da l pattern $(x_1,y_1), \ldots, (x_l,y_l)$ dove $x_i \in R^n$ e $y_i \in 1,2,\ldots,k$. Trasformiamo il problema di classificazione multiclasse in uno binario [20], replicando ogni pattern k volte e aumentando ogni replica con un vettore v di dimensione k:

$$(x_i, y_i) \rightarrow \begin{cases} (x_i \mid v^1, y_i^1) \\ (x_i \mid v^2, y_i^2) \\ \vdots \\ (x_i \mid v^k, y_i^k) \end{cases}$$

dove il simbolo "|" indica una concatenazione di un vettore, e dove l'i-esimo componente del vettore v_j è

$$(v^j)_i = \begin{cases} +1 \text{ if } i = j \\ -1 \text{ if } i \neq j \end{cases}$$

e

$$y_i^j = \begin{cases} +1 \text{ if } j = y_i \\ -1 \text{ if } j \neq y_i \end{cases}$$

In altre parole, il dataset originale multiclasse è stato trasformato in uno binario con $m = kl$ pattern: $(x'_1,y'_1),\ldots,(x'_m,y'_m)$ con $x'_i \in R^{n+k}$ e $y'_i \in \{-1,+1\}$.
Per esempio, un pattern derivante da un dataste tri-classe (x_i,y_i), appartenente alla classe 2 ($y_i = 2$), è espanso nel seguente modo:

$$(x_i,2) \rightarrow \begin{cases} (x_i \mid (+1,-1,-1),-1) \\ (x_i \mid (-1,+1,-1),+1) \\ (x_i \mid (-1,-1,+1),-1) \end{cases}$$

Questa espansione assomiglia alla costruzione di Kesler, che fu introdotta da Nilson nel 1965 ed è stata recentemente usata per trasformare un problema di una o due classi nella struttura di apprendimento di un SVM. Sfortunatamente, la costruzione di Kesler aumenta la dimensione dei dati da n a nk. Purtroppo l'SVM è molto sensibile all'aumento delle dimensioni, rispetto a altri metodi di apprendimento, e per questo motivo si reputa che questo è un serio svantaggio.
D'altra parte, è facile capire che questo metodo può trasformare un problema separabile lineare (nel senso OVA), in uno separabile non-lineare (binario). Come ultima osservazione su questa trasformazione, è utile notare che vengono usati i valori ±1 per costruire il vettore *v*. Questo è una scelta ovvia nella pratica comune per normalizzare ogni feature di dato nel range [-1,+1], seppur altre opzioni sono possibili.
Dopo aver trasformato il dataset, è possibile risolvere il problema di ottimizzazione associato al SVM:

$$\begin{array}{ll} \min_{w,\xi,b} & \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{m}\xi_i \\ \text{soggetto a} & y'_i\,(w \cdot x'_i + b) \geq 1 - \xi_i \\ & \xi_i \geq 0 \end{array}$$

o in forma duale

$$\begin{array}{ll} \min_{\alpha} & \frac{1}{2}\alpha^T Q\alpha - u^T\alpha \\ \text{soggetto a} & 0 \leq \alpha_i \leq C \\ & \sum_{i=1}^{m}\alpha_i y'_i = 0 \end{array}$$

dove $\boldsymbol{u}$ è il vettore di 1, Q è la matrice m x m simmetrica positiva semidefinita

$$q_{i,j} = y'_i \, y'_j \, K(x'_i, x'_j)$$

e $K(.,.)$ è una funzione kernel.
Le funzioni kernel possibili per un dataset formato da due pattern sono riassunte nella Tabella 5.1.

Nome	Kernel
Lineare	$K(x_1, x_2) = x_1 \cdot x_2$
Gaussiano normalizzato	$K(x_1, x_2) = e^{-\frac{\gamma}{n}\|x_1 - x_2\|}$
Polinomiale normalizzato	$K(x_1, x_2) = \frac{(x_1 \cdot x_2 + n)^P}{\sqrt{(x_1 \cdot x_1 + n)^P}\sqrt{(x_2 \cdot x_2 + n)^P}}$

Tabella 5.1 - Kernel utilizzati da mcSVM

Per effettuare la classificazione di un nuovo pattern z, bisogna applicare la stessa procedura descritta in precedenza, replicando il pattern e aumentandolo:

$$z \rightarrow \begin{cases} z \mid v^1 = z'_1 \\ z \mid v^2 = z'_2 \\ \quad \vdots \\ z \mid v^k = z'_k \end{cases}$$

poi ogni pattern aumentato è dato in input al SVM, e messo in fase di training in accordo alle equazioni del problema di ottimizzazione.
La decisione è presa con l'arbitro Winner-Take-All (WTA). In altre parole,

$$\hat{y} = \arg\max[F(z'_1), ..., F(z'_k)]$$

con

$$F(z'_i) = \sum_{j=1}^{m} \alpha_j y'_j \, K(x'_j, z'_i) + b$$

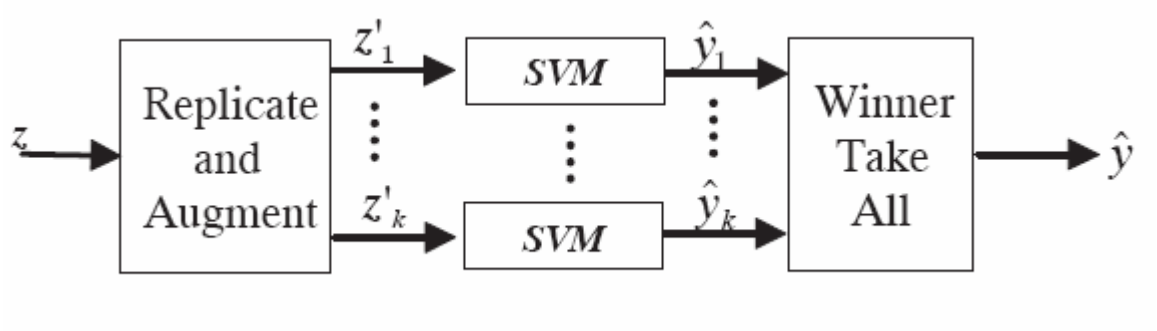

Figura 5.4 - Metodo binario aumentato

Notare che i k SVM sono, in effetti, lo stesso: questo fatto può essere sfruttato durante la fase di feedforward per ridurre il costo computazionale.

5.4.2. Stima dell'errore di generalizzazione

La stima dell'errore di generalizzazione è uno dei più importanti compiti delle macchine di apprendimento: in parole povere si è interessati a stimare la probabilità che la macchina costruita possa misclassificare nuovi pattern, assumendo che i nuovi dati derivino dalla stessa distribuzione (sconosciuta) dei dati del training set.
Di seguito verrà utilizzato π per indicare l'errore generalizzato sconosciuto e $\hat{\pi}$ per la sua stima. In particolare si è interessati al caso probabilistico peggiore: si vuole trovare il limite superiore dell'errore reale di generalizzazione

$$\pi \leq \hat{\pi}$$

che abbia probabilità 1-δ, dove δ è un valore definito dall'utente (tipicamente δ=0,05 o minore, a seconda delle applicazioni). Notare che questo approccio è leggermente differente dagli approci tradizionali statistici, dove l'obbiettivo è stimare un approssimazione dell'errore $\pi \approx \hat{\pi} \pm \Delta_\pi$ dove Δ_π è l'intervallo di confidenza.
L'errore empirico verrà indicato da

$$\nu = \frac{1}{l}\sum_{i=1}^{l} I(y_i, \hat{y}_i)$$

dove $I(y_i, \hat{y}_i)$=1 se $y_i \neq \hat{y}_i$ e zero altrimenti.
Se possibile, $\hat{\pi}$ sarà indicato come la somma di tre termini:

(1) l'errore empirico ν

(2) un termine di correzione, che tiene in considerazione il fatto che ν è ovviamente una sottostima dell'errore vero, e
(3) un termine di confidenza, che deriva dal fatto che il numero di pattern del training è finito

La generalizzazione della stima è fortemente relazionata alla scelta del modello, che è la procedura che permette di scegliere gli iper-parametri ottimali per l'SVM (*C,γ, o p*).
Il valore $\hat{\pi}$ è calcolato per vari valori degli iper-parametri e l'SVM ottimo è scelto come quello per cui si è ottenuto il minimo.

Metodo	TRAIN	CORR	CONF
Training set	ν		$\frac{\hat{\sigma}}{\sqrt{l}} F^{-1}(1-\delta)$
Test set		ν_{test}	$\sqrt{\frac{-\ln\delta}{2m}}$
K-fold cross validation		ν_{kcv}	$\sqrt{\frac{-k\ln\delta}{2l}}$
Leave-one-out		ν_{loo}	$\frac{\hat{\sigma}}{\sqrt{l}} F^{-1}(1-\delta)$
Bootstrap		ν_{boot}	$\frac{\hat{\sigma}_{boot}}{\sqrt{N_B}} F^{-1}(1-\delta)$

Tabella 5.2 - Stima del limite superiore dell'errore di generalizzazione

In Tabella 5.2 vengono indicati come sono composti i limiti superiori per l'errore di generalizzazione. I termini vengono qui indicati con TRAIN, CORR e CONF: l'entrata vuota indica che il termine corrispondente non è usato per il calcolo.

Nella tabella vengono riportati alcuni termini statistici, che di seguito vengono illustrati brevemente:

- $\hat{\sigma}$ indica la stima della deviazione standard di una variabile casuale X (il campione in esame) e viene calcolato nel seguente modo:

$$\hat{\sigma} = \sqrt{\frac{1}{l-1}\sum_{i=1}^{l}\left(X_i - \frac{1}{l}\sum_{j=1}^{l} X_j\right)}$$

- $F^{-1}(.)$ è la funzione inversa normale cumulativa di distribuzione.

- l è il numero di pattern del dataset originale
- m è il numero di pattern dopo la trasformazione del problema multiclasse in quello binario aumentato
- N_B è il numero di bootstrap replicati

5.4.3. File di configurazione

Il file di configurazione è un file di testo con estensione .cfg attraverso il quale vengono passate informazioni al tool per la generazione del SVM sotto forma:

VARIABILE = VALORE

Dove VARIABILE indica uno dei parametri che devono essere passati a mcSVM e VALORE una stringa alfanumerica. Solo un'informazione per linea è permessa e le linee che iniziano per '#' vengono ignorate [21].
I principali parametri sono:

- fDATA = file: indica il file con relativo percorso del dataset per il training. Il file deve essere un file di testo con un pattern per riga. Ogni pattern è composto da valori numerici separati da spazi o tabulazioni (feature) e dall'identificatore della classe.
- nPATTERN = N: dove N è un intero che indica il numero di pattern
- nFEATURE = N: dove N è un intero che indica il numero di feature
- nCLASS = N: dove N è un intero che indica il numero di classi
- tVAL = {T,F}: indica se un file per la validazione è disponibile i cui parametri sono indicati dai campi seguenti
- fVALDATA = file: indica il file con relativo percorso del dataset per il testing. Il file ha lo stesso formato di quello del training
- nVALPATTERN = N: dove N è un intero che indica il numero di pattern
- tVALTARGET = {T,F}: indica se i pattern di validazione contengono la classe di appartenenza

N	Range di normalizzazione $[x_-, x_+]$
0	nessuno
1	$[x_{\min}, x_{\max}]$
2	$[\bar{x} - c\sigma, \bar{x} + c\sigma]$
3	2 + saturazione

Tabella 5.3 – Normalizzazioni

- tNORM = {0,1,2,3}: indica il tipo di normalizzazione possibile sui dati. Le normalizzazioni possibili sono indicati nella Tabella 5.3
- KERNEL = {0,1,2}: determina il kernel in accordo alla Tabella 5.4

KERNEL	Tipo di kernel
0	Lineare
1	Gaussiano
2	Polinomiale normalizzato

Tabella 5.4- Tipi di kernel

- GAMMA = X: dove X è un valore che indica la profondità del kernel gaussiano
- POLY = N: dove N è un intero che indica il grado del kernel polinomiale
- tGEN = {0,1,2,3,4}: indica quale metodo è utilizzato per stimare l'errore di generalizzazione del SVM su nuovi dati, secondo la Tabella 5.5

tGEN	Stima dell'errore di generalizzazione
0	Nessuna stima
1	Test set
2	Leave-one-out
3	Bootstrap
4	K-fold validation

Tabella 5.5 - Metodi di stima dell'errore

- nTEST = N: indica il numero di pattern per test set.
- nBOOT = N: dove N è un intero che indica il numero di bootstrap replicati
- nKFOLD = N: dove N è un valore intero che indica il numero di fold. Il numero di campioni del training set deve essere esattamente divisibile per N

5.5. Weka

Il tool Weka fornisce un ambiente general-purpose per classificazioni automatiche, regressione, clustering e feature selection, problemi di data mining molto comuni nella bioinformatica [25].
Contiene al suo interno un'ampia collezione di algoritmi e metodi di pre-processamento dei dati, complementati da un'interfaccia grafica per l'esplorazione dei dati e la comparazione sperimentale di differenti macchine di apprendimento sullo stesso problema.

I dati vengono forniti sotto forma di una singola tabella relazionale.
Il suo obiettivo è di assistere l'utente nell'estrapolare informazioni utili dai dati e identificare in modo semplice un algoritmo efficiente per la creazione di modelli.
L'interfaccia principale di Weka è Explorer (Figura 5.5). È composta da un set di pannelli, ognuno dei quali è adibito ad un task preciso.
Il panello Preprocess legge i dati da un file, database SQL o URL. Una limitazione significativa è che tutti i dati sono tenuti in memoria principale, perciò deve essere utilizzato un sottocampionamento per i dataset troppo grandi.
I dati letti da file possono essere poi pre-processati utilizzando uno dei filtri di Weka. Per esempio si possono eliminare tutte le istanze (righe) per le quali un certo attributo (colonna) ha un determinato valore. Un'operazione di *undo* è fornita per riportare i dati nella forma originale.
Il panello di Preprocess mostra inoltre un istogramma dell'attributo che è correntemente selezionato e alcune statistiche su di esso.
Dopo aver caricato un dataset, l'analisi può essere effettuata utlizzando gli strumenti disponibili negli altri panelli.

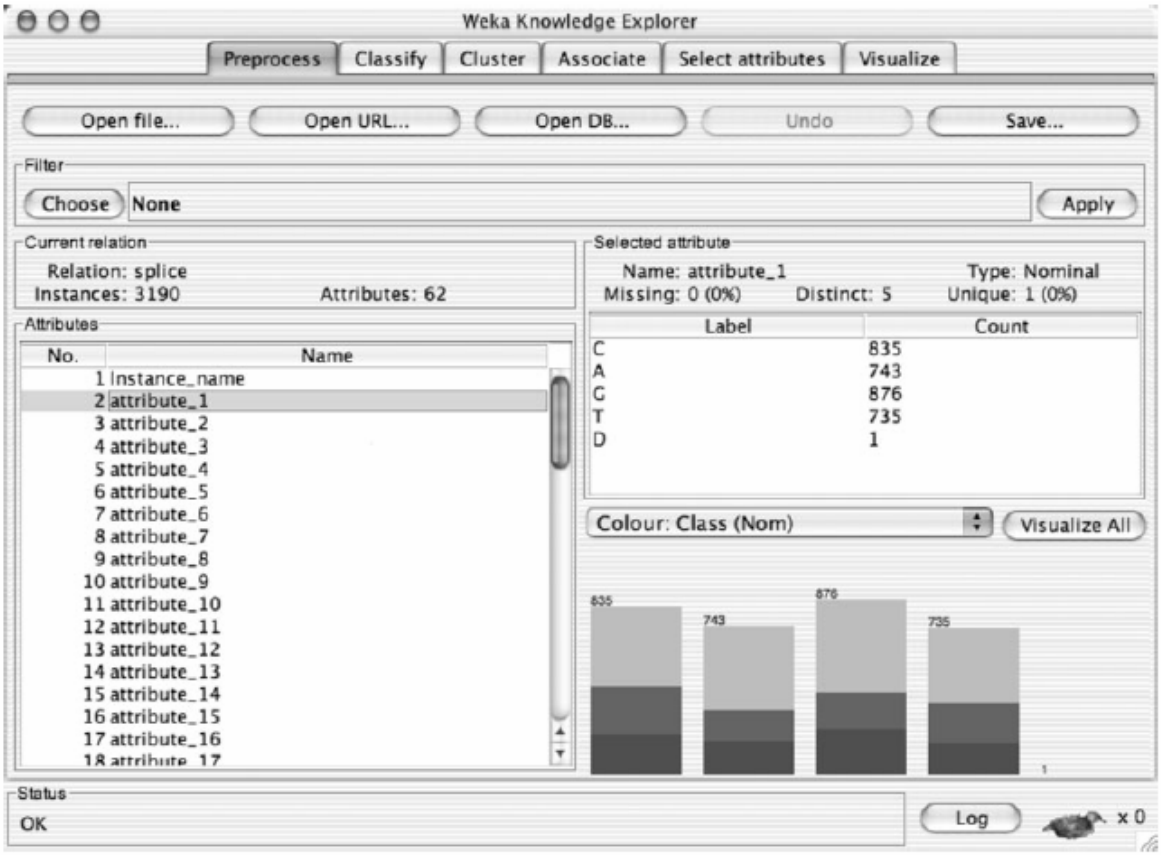

Figura 5.5 - Interfaccia di Weka

Se i dati richiedono un problema di classificazione o di regressione, si può usare il pannello Classify. Questo fornisce un'interfaccia per gli algoritmi di learning per modelli di classificazione e di regressione e tool di valutazione per analizzare i risultati del processo di learning.
Weka dispone di molte tecniche di learning per classificazione e regressione: alberi decisionali, set di regole, classificatori Bayesiani, Support vector machine, regressioni logistiche e lineari e mtodi nearest-neighbor. Gli algoritmi di apprendimento possono essere valutati utilizzando la cross-validation o un set esterno. Il tool fornisce misure numeriche standard per le performance (per esempio accuratezza, errore quadratico medio) come grafici per per visualizzare le performance del classificatore (per esempio curve di precision-recall).
Il terzo pannello, Cluster, permette l'utilizzo degli algoritmi di cluster. Questi includono k-means, mix di distribuzioni normali con matrici diagonali di co-varianza stimate utilizzando EM, e uno schema euristico incrementale gerarchico. L'assegnazione dei cluster può essere visualizzata e comparata con i cluster originali definiti da uno degli atributi dei dati.
Weka contiene anche algoritmi per la generazione di regole di associazione che possono essere usate per identificare relazioni tra gruppi di attributi nei dati. Questi sono disponibili tramite il panello Associate.
Comunque, molto più interessante nell'ambito della bioinformatica è il pannello Select attribute, che offre metodi per identificare quali subset di attributi sono predittivi per un altro attributo (target), come per esempio la classe di appartenenza. Weka contiene molti metodi per la ricerca nello spazio dei subset degli attributi. I metodi di ricerca includono correlazione e criteri basati sull'entropia così come le performance di un determinato schema di apprendimento (per esempio un albero decisionale) per un particolare subset di attributi. Differenti metodi di ricerca e valutazione possono essere combinati, rendendo il sistema molto flessibile.
L'ultimo pannello, Visualization, mostra una matrice di *scatter plot* per tutte le coppie di attributi nei dati. Ogni elemento della matrice può essere selezionato e allargato in una finestra a parte, dove si possono ricavare le informazioni sui punti di dato.

Capitolo 6
Esperimenti svolti

In questo capitolo sono illustrati gli esperimenti che sono svolti sui dati descritti di seguito. Siccome i dati sono di diversa natura, il capitolo è stato diviso in due sezioni.
Nella prima sezione saranno illustrati gli esperimenti eseguiti su dati inerenti i valori di espressione dei geni per i tessuti prelevati dal colon. Si tenterà quindi di distinguere tra pazienti malati e sani.
Nella seconda parte invece si analizzeranno dati provenienti da analisi alla prostata. In questo caso le classi saranno tre: paziente sano, con tumore benigno e con tumore maligno.

6.1. Dati del colon

I dati analizzati sono relativi ai valori di espressione di 22175 geni, per ognuno dei 19 pazienti da cui sono stati estratti campioni di tessuto del colon. Dei 19 pazienti, 11 sono classificati come sani e 8 sono affetti da tumore.
I risultati preliminari descritti in questo sottocapitolo sono stati ottenuti partendo dai dati già filtrati, forniti insieme ai dati originali. Il filtraggio, ossia l'eliminazione di dati considerati superflui per le analisi future, è basato sul guadagno d'informazione di ognuno dei geni. I dati così filtrati restringono l'analisi a 8466 geni dei 22175 originali.
I risultati presentati sono stati paragonati a quelli ottenuti applicando PAMR, forniti insieme ai dati originali.

6.1.1. Analisi preliminare

Analizzando i valori di espressione degli 8466 geni filtrati, separatamente per pazienti sani e malati, e confrontando gli intervalli in cui essi ricadono, si ottiene che 3269 geni hanno valori di espressione che ricadono in intervalli completamente disgiunti.

La classificazione delle due categorie di pazienti risulta immediata in quanto è sufficiente uno qualunque dei 3269 geni "disgiunti" per ottenere un valore di soglia grazie al quale costruire un classificatore perfetto sui dati forniti. Ne consegue che non è possibile valutare quali geni siano più adatti allo scopo basandosi solo sulle prestazioni di classificazione, in quanto l'accuratezza sarebbe massima in ogni caso.
Anche eseguendo il clustering dei pazienti in base ai valori di espressione dei geni si individuano facilmente i due gruppi di pazienti (sani/malati). A questo proposito è stato applicato il metodo Pvclust, che applica il clustering gerarchico e fornisce per ogni cluster due *p-value*: AU (*Approximate Unbiased*) e BP (*Bootstrap Probability*) , rispettivamente indicati con i numeri in rosso e verde.

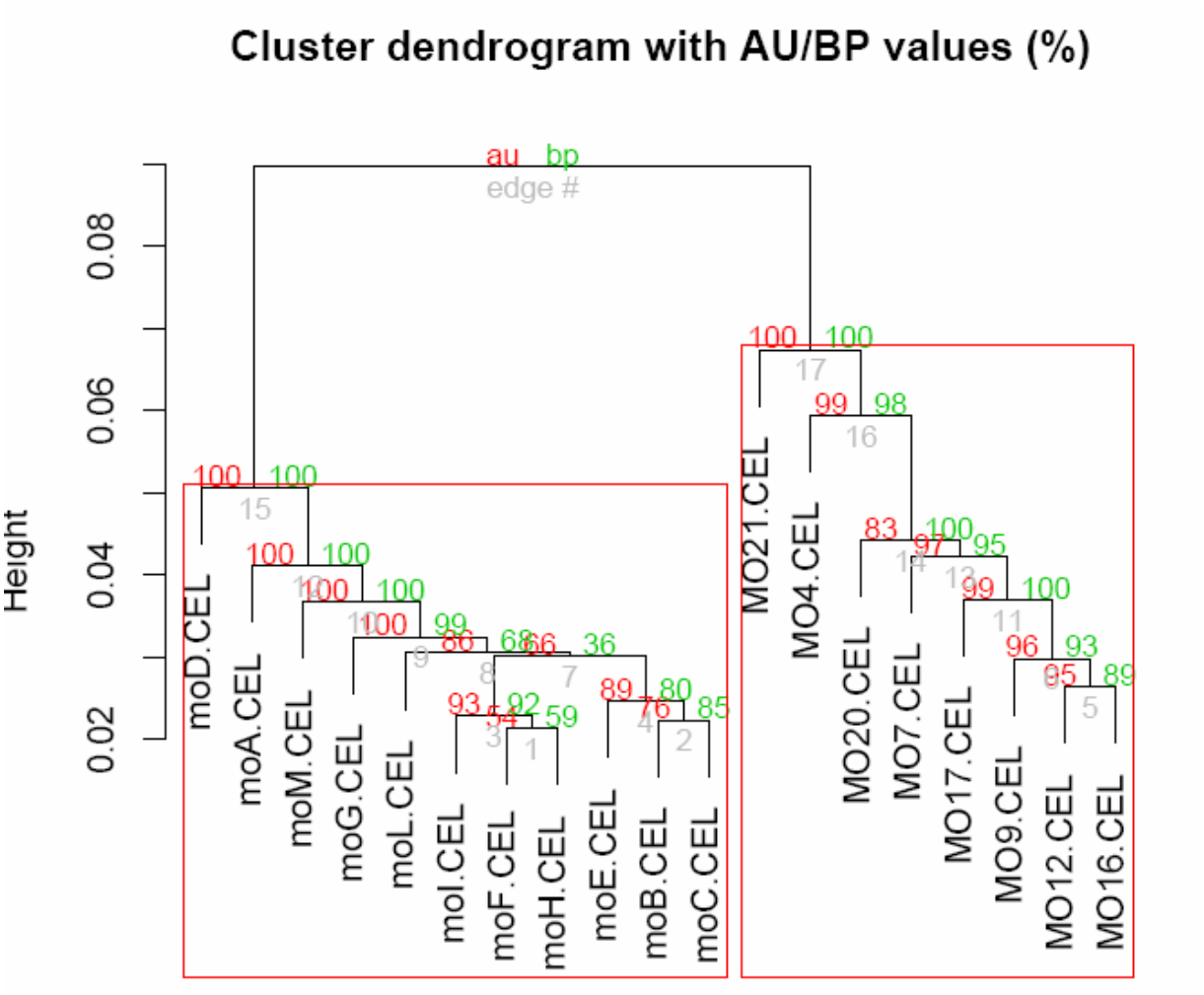

Figura 6.1 - Dendrogramma ottenuto dall'applicazione di PVClust ai pazienti

In Figura 6.1 è riportato il dendrogramma ottenuto applicando PVClust ai pazienti analizzati. I rettangoli rossi evidenziano i cluster che dividono perfettamente i pazienti classificati come malati da quelli sani e hanno *p-value* massimo, a conferma del fatto che le due categorie hanno valori di espressione genetica molto diversi tra loro, ma molto simili tra pazienti dello stesso gruppo.
I numeri in grigio sui rami del dendrogramma indicano il numero di campioni (pazienti) che ogni cluster raggruppa.
Alla luce di tali considerazioni, il problema si concentra sulla selezione dei migliori 100 geni a partire dai 3269 che permettono di distinguere le due categorie di pazienti.
Sono stati applicati diversi criteri di ordinamento ai valori di espressione dei geni in modo da stimare la capacità di classificazione dei geni stessi, come spiegato in seguito.
L'ordinamento ottenuto è stato paragonato con i risultati di riferimento descritti nel prossimo paragrafo.

6.1.2. Risultati di riferimento

L'analisi di riferimento svolta con PAMR ha selezionato 221 geni impostando la soglia al valore 5,228. I relativi risultati sono illustrati in Figura 6.2 e Figura 6.3.

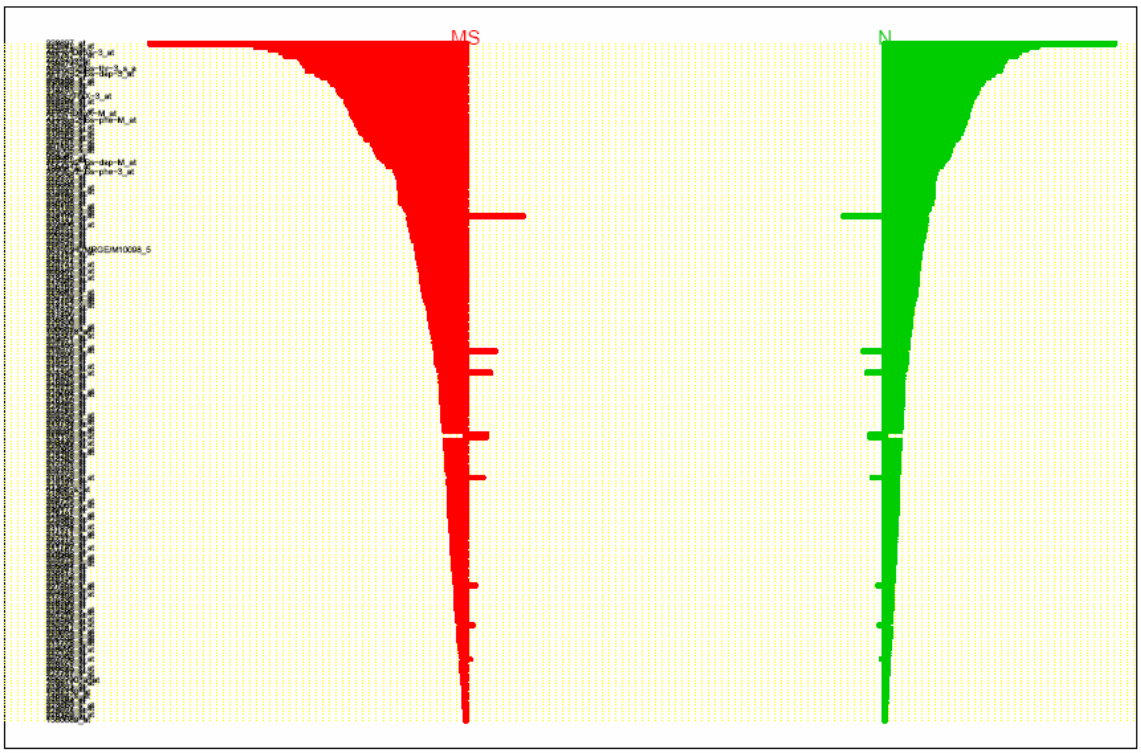

Figura 6.2 - Centroidi dei 221 geni selezionati da PAMR

- In Figura 6.2 sono elencati i geni che l'algoritmo seleziona per distinguere le due classi di pazienti e le relative componenti dei centroidi; in rosso a sinistra sono rappresentati i malati (MS), in verde a destra i sani (N). I geni nella prima parte dell'elenco hanno valori molto diversi tra loro, a differenza dei geni della seconda metà dell'elenco.
- In Figura 6.3 è illustrato il risultato della classificazione dei pazienti usando i geni selezionati; le due categorie vengono separate in modo netto e senza errori.

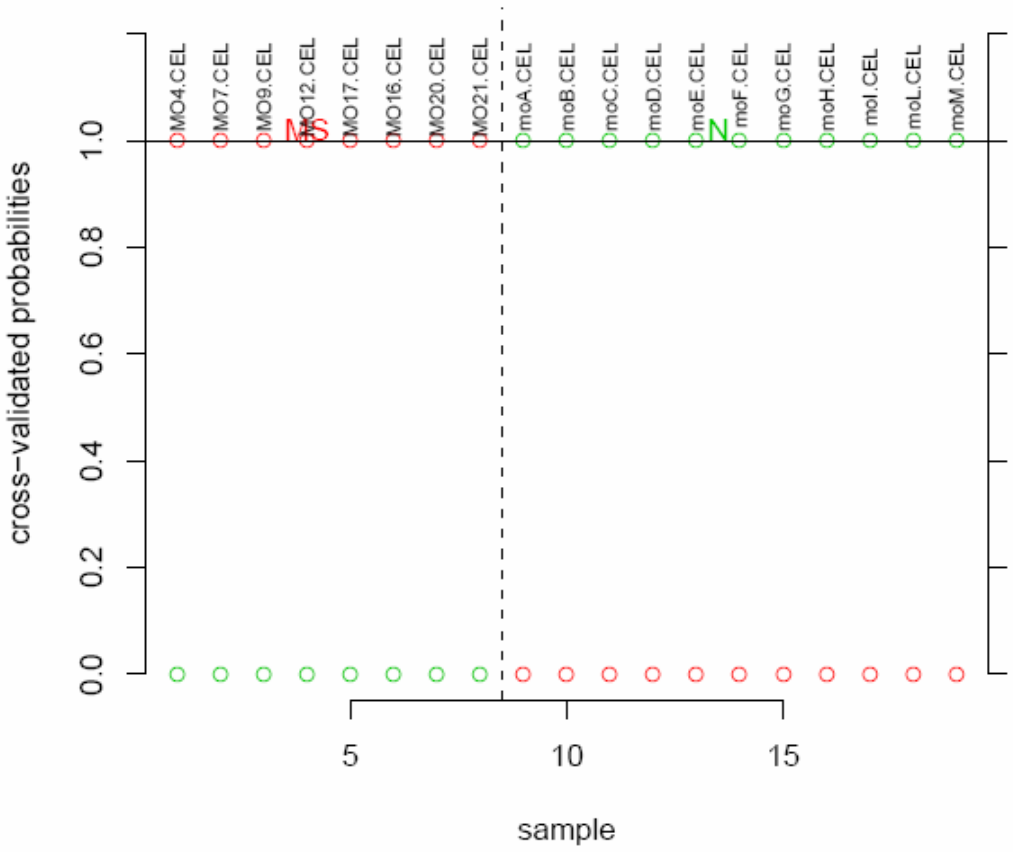

Figura 6.3 - Cross-validation usando i geni selezionati

6.1.3. Ordinamento

L'ordinamento dei geni è stato effettuato tenendo conto delle seguenti caratteristiche dei valori di espressione.

- **Ampiezza** degli intervalli dei valori di espressione di ogni gene per ciascuna delle due categorie (sani e malati). Un intervallo stretto è considerato più selettivo di un intervallo ampio, quindi il relativo gene ha una capacità migliore di classificazione. Solitamente l'ampiezza dell'intervallo per i pazienti sani è diversa dall'ampiezza

dell'intervallo per i pazienti malati, quindi è possibile assegnare al gene considerato una capacità di classificazione determinata dalla:

- media delle ampiezze dei due intervalli,
- ampiezza massima dei due intervalli,
- ampiezza minima dei due intervalli.

- **Distanza** tra i due intervalli di valori di espressione, uno relativo ai pazienti sani, l'altro a quelli malati. Maggiore è la distanza tra i due intervalli e maggiore sarà la capacità di distinguere le due categorie. Ad ogni gene può essere associata:
  - la distanza tra le medie dei due intervalli,
  - la distanza minima tra gli intervalli,
  - la distanza massima tra gli intervalli.

L'ordinamento è stato effettuato considerando sia i valori di distanza e ampiezza separatamente, sia calcolando il rapporto tra la distanza e l'ampiezza degli intervalli dei valori di espressione per ogni gene.
Gli esperimenti sono stati svolti calcolando la distanza tra gli intervalli e l'ampiezza degli stessi con tutte le combinazioni dei metodi elencati (massimo, media, minimo).
In Figura 6.4 è riportato un esempio di valutazione delle ampiezze e delle distanze relative a un singolo gene: supponendo di avere 4 valori di espressione di pazienti malati (MS#) e tre valori per i sani (N#), la figura visualizza le distanze (massima e minima) tra gli intervalli e la loro ampiezza. La distanza media è ottenuta come differenza tra i valori medi dei due intervalli e l'ampiezza media è ottenuta come media tra l'ampiezza massima e quella minima.

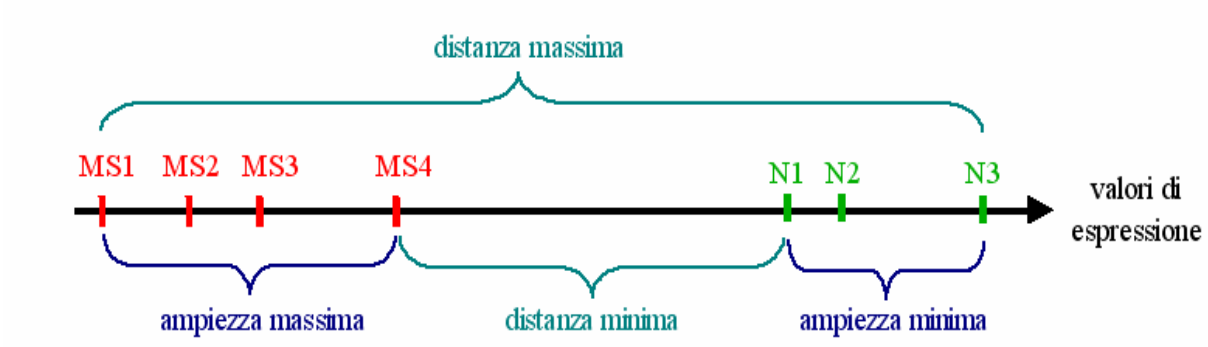

Figura 6.4 - Rappresentazione grafica delle ampiezze degli intervalli e delle relative distanze

6.1.4. Valutazioni

Prima di valutare i risultati sono state analizzate le posizioni e le distribuzioni dei valori ottenuti dai vari tipi di ordinamento e dai geni scelti da PAMR, in modo da evidenziare il loro andamento e identificare le caratteristiche dei geni selezionati. In Figura 6.5 e Figura 6.6 sono illustrati due grafici esemplificativi dei risultati ottenuti.

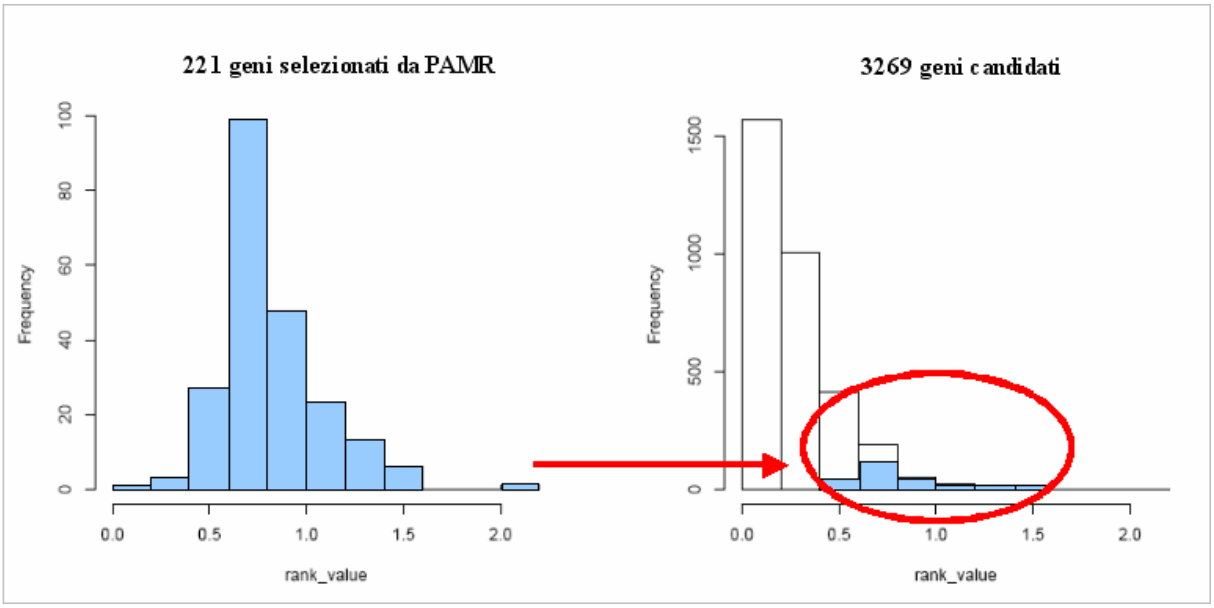

Figura 6.5 - Distribuzioni dei valori di distanza minima

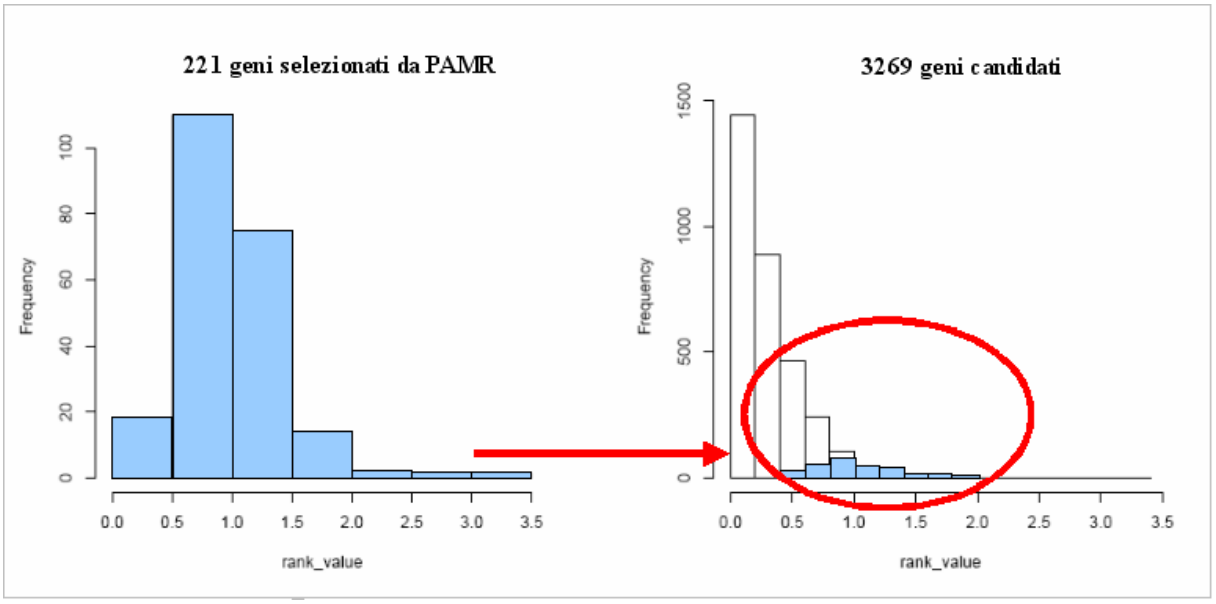

Figura 6.6 - Distribuzioni dei rapporti distanza minima / ampiezza massima

In Figura 6.5 è riportato il confronto tra le distribuzioni dei valori di distanza minima per i geni selezionati da PAMR (a sinistra) e di tutti i 3269 geni candidati (a destra).
In base a questo criterio di ordinamento, geni con valori elevati di distanza (rank_value) sono candidati migliori. La distribuzione evidenzia che, tra tutti i geni candidati, quelli con valori più elevati hanno rank_value superiore a 0,5 mentre la maggior parte dei geni ha valori inferiori. La distribuzione dei valori dei geni selezionati da PAMR è significativamente diversa da quella dei geni candidati e indica che sono correttamente scartati molti geni con bassi valori di distanza.
Questo comportamento denota che il criterio della distanza minima permette di individuare un sottoinsieme di geni coerente con quello scelto da PAMR. Tale criterio sembra appropriato per determinare i geni che differenziano meglio le due categorie di pazienti. In particolare la distanza minima è più adatta degli altri criteri di distanza (massima e media) in quanto misura la distanza di caso peggiore, che è quella tra i due valori più prossimi appartenenti a classi diverse (cfr. Figura 6.4).
È possibile giungere alle stesse valutazioni considerando anche il criterio dell'ampiezza massima degli intervalli, oltre che la sola distanza minima. Siccome si ritiene più selettivo un gene i cui valori di espressione hanno una distribuzione con bassa varianza, l'ipotesi di caso peggiore prevede di considerare l'ampiezza massima tra i due intervalli di valori associati ai sani e ai malati. I grafici delle distribuzioni dei valori ottenuti considerando il rapporto tra distanza minima e ampiezza massima sono riportati in Figura 6.6.
Complessivamente i due criteri di ordinamento più adeguati risultano essere:
- distanza minima;
- distanza minima / ampiezza massima.

6.2. Dati della prostata

I dati pervenutici riguardanti la diagnostica del tumore alla prostata sono relativi ai valori di espressione di 19664 geni, per ognuno dei 95 campioni.
Purtroppo non si hanno tutti i valori di espressione per ogni gene e per ogni campione. Inoltre i campioni inviatici sono in riferimento a prelievi di tessuto in istanti diversi per alcuni pazienti. Per esempio, il tessuto della prostata di un paziente è stato prelevato una prima volta quando è stato diagnosticato il tumore benigno, e poi una seconda volta, in seguito all'operazione, per verificare che fosse guarito.
Assieme ai dati di espressione sono stati forniti anche i *p-value* associati ad ogni dato. Tale valore rappresenta la probabilità che il dato di espressione associato misurato sia errato. Quindi, valori bassi di *p-value* indicano misurazioni affidabili, mentre valori alti denotano misure inaffidabili dovute al rumore.

I pazienti sono divisi in 3 classi:

- con tumore maligno, 67
- con tumore benigno, 9
- sani, 19.

Tale distribuzione è evidenziata dal grafico a settore circolare in Figura 6.7.

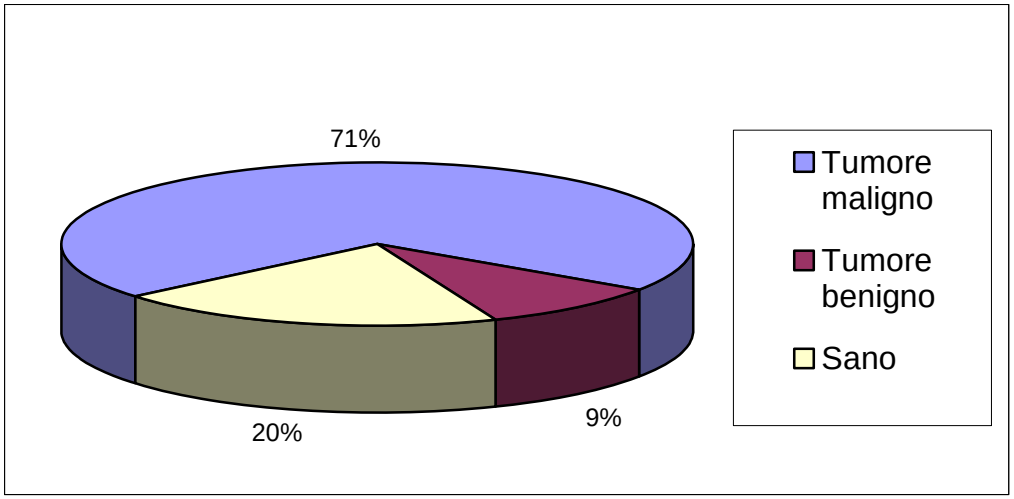

Figura 6.7 - Distrubuzione delle classi sui dati della prostata

6.2.1. Filtraggio dei dati

Come si è detto in precedenza, per alcuni geni e per alcuni pazienti non si avevano i dati relativi ai valori di espressione. Dal momento che i software utilizzati per l'analisi di questi dati non ammettono l'omissione di valori nei file forniti come input, si è deciso di eliminare i geni che non avessero anche solo un valore per un campione.
In questo modo si è passati da un totale di 19664 geni a un totale di 17695 geni, con una perdita del 10,01% rispetto ai dati iniziali dovuta alla mancanza di misurazioni durante il prelevamento e il trattamento dei campioni.
Si è valutata la possibilità di inserire valori pre-calcolati al posto di quelli mancanti, ma ci si è trovati in difficoltà nel scegliere il valore più adatto. I valori possibili per la sostituzione potevano essere la media tra tutti i valori di espressione già presenti per quel gene, oppure la media del valore di espressione per quel gene per i campioni appartenenti alla stessa classe.
Entrambe le soluzioni però avevano qualche problema di applicazione. In alcuni casi mancavano proprio i valori per tutti i campioni. Difficile quindi poter calcolare un valore medio non avendo neanche un valore su cui basarsi. Un altro problema era la possibilità di influenzare le analisi future inserendo un valore che nella realtà sarebbe stato differente.

Dal momento che la mole di geni a disposizione permetteva l'eliminazione di quegli attributi, i geni, per i quali non si avevano dei valori, e inoltre permetteva ai programmi di creare modelli e di analizzare i dati, si è deciso per non considerare questi attributi eliminandoli da tutte le analisi future.
Per avere la possibilità di valutare la bontà dei geni selezionati dai vari algoritmi, i dati sono stati ordinati in modo decrescente per valori di p-value massimo. I geni migliori avevano quindi indice di colonna maggiore, rispetto a quelli con p-value massimo alto.
I software utilizzati per l'analisi sono stati GEMS e mcSVM. Tali strumenti non prevedeno l'utilizzo di informazioni sui valori di *p-value*, che quindi non sono stati considerati nelle analisi successive. Di seguito vengono illustrati i risultati ottenuti .

6.2.2. Utilizzo di GEMS

Per creare modelli di classificazione si è utilizzato il programma GEMS, che permette di creare modelli di classificazione in base a vari tipi di metodi di classificazione MC-SVM e di integrare questi modelli con algoritmi di *feature selection*.
Per avere una visione complessiva di questi metodi e delle loro prestazioni sui dati forniti, si è deciso di svolgere sperimenti su tutte le possibili combinazioni tra i metodi SVM, gli algoritmi di *feature selection* disponibili e le tipologie di *kernel*.
I metodi SVM disponibili sono:

- OVR
- OVO
- DAGSVM
- WW
- CS

Gli algoritmi di *feature selection* messi a disposizione dal programma sono:

- Non parametric one-way ANOVA: Kruskal-Wallis (KW)
- Signal-to-noise ratio in one-versus-rest fashion (S2N_OVR)
- Signal-to-noise ratio in one-versus-one fashion (S2N_OVO)
- Ratio of variables between categories to within categories sum of square (BW)

I tipi di *kernel* disponibili sono:

- gaussiano
- polinomiale

Per calcolare l'influenza che gli algoritmi di selezione hanno sulle performance dei metodi SVM, si sono svolti anche esperimenti senza l'utilizzo della *feature selection*, opzione disponibile in GEMS.

Per avere un confronto con il metodo utilizzato sui dati del colon, si è aggiunto questo metodo, chiamato degli intervalli di espressione (IE), a quelli di *feature selection* forniti dal tool.
L'unica osservazione che si può fare da un'analisi dei risultati ottenuti dai ricercatori utilizzando queste metodologie di classificazione è che l'unione della feature selection con qualsiasi metodo di classificazione aumenta l'accuratezza dell'intero modello rispetto alla classificazione senza feature selection.

6.2.3. Utilizzo di mcSVM

Il tool mcSVM permette la costruzioni di modelli basati su problemi multiclasse con un algoritmo biclasse modificato di SVM.
Il software non prevede un'operazione preliminare di *feature selection*. Per questo motivo, e per avere la possibilità di confrontare le performance ottenute con gli algoritmi forniti da GEMS, si è integrato in modo manuale la selezione dei geni migliori, determinati dagli algoritmi descritti precedentemente.
Per valutare anche le prestazioni dell'algoritmo al variare del dataset di *training*, sia in numero che in distribuzione delle classi, i 95 pazienti iniziali sono stati suddivisi in modo tale da costruire diversi tipi di training e test set.
Si sono ottenuti così quattro coppie di *training* set e test set, riportati in Tabella 6.1.
Per ogni coppia training e test est si sono applicate le combinazioni possibili tra i kernel forniti da mcSVM e comuni a GEMS (gaussiano e polinomiale), e i cinque metodi di selezione considerati negli altri esperimenti (KW, BW, S2N_OVO, S2N_OVR, IE).

Training set		Test set	
80 campioni	Classe 0: 55 Classe 1: 9 Classe 2: 16	15 campioni	Classe 0: 12 Classe 1: 0 Classe 2: 3
70 campioni	Classe 0: 48 Classe 1: 7 Classe 2: 15	25 campioni	Classe 0: 19 Classe 1: 2 Classe 2: 4
60 campioni	Classe 0: 42 Classe 1: 6 Classe 2: 12	31 campioni	Classe 0: 21 Classe 1: 3 Classe 2: 7
21 campioni	Classe 0: 7 Classe 1: 7 Classe 2: 7	74 campioni	Classe 0: 60 Classe 1: 2 Classe 2: 12

Tabella 6.1 - Divisione in training e testing set

Purtroppo il report che il tool fornisce dopo l'esecuzione dell'algoritmo e del test sui dati forniti, non specifica quali campioni di una determinata classe abbiano portato all'errore.

Per valutare quindi le performance del modello costruito in base anche alle classi sbagliate durante la fase di testing, si è dovuto dividere tutti i test set a seconda delle classi di appartenenza di ogni singolo campione.

6.2.4. Utilizzo di Weka

Con Weka si possono costruire vari modelli di classificatori in base alle esigenze e ai dati da analizzare.
Si utilizza questo tool per effettuare un'analisi delle prestazioni degli alberi decisionali, che sono classificatori ad alta interpretabilità. Siccome una limitazione del programma è l'utilizzo della memoria per la conservazione dei dati, i dati analizzati con Weka comprendono solamente su un subset dei dati iniziali.
Questo subset viene definito utilizzando i geni in comune a tutti i metodi di *feature selection* utilizzati con i tool precedenti. In questo modo si riduce il numero di dati che deve essere immagazzinato in memoria e si considerano solamente i geni che sono stati considerati i migliori dalle analisi precedenti.
Dopo aver costruito il subset dei dati per le analisi, si procede con la costruzione di un albero decisionale utilizzando uno degli algoritmi messi a disposizione da Weka. Dall'albero di decisione costruito da Weka è possibile risalire direttamente ai geni effettivamente utilizzati per la classificazione dei campioni analizzati.

Capitolo 7
Risultati

In questo capitolo verranno mostrati i risultati ottenuti dagli esperimenti illustrati nel capitolo precedente.
Siccome i dati sono di diversa natura, il capitolo è stato diviso in due sezioni differenti.
La prima illustra i risultati ottenuti e l'elenco dei geni selezionati come i migliori per la distinzione tra le classi dei dati prelevati dal tessuto del colon.
La seconda parte del capitolo, invece, mostra ciò che si è ottenuto dall'analisi dei dati per i tumori alla prostata.

7.1. Dati del colon

L'analisi delle posizioni dei geni selezionati da PAMR con quelli dei vari ordinamenti evidenzia un'elevata affinità tra PAMR e il criterio della distanza minima. Infatti i primi 62 geni ottenuti ordinando secondo questo criterio coincidono con quelli individuati da PAMR e complessivamente 77 geni dei primi 100 sono presenti anche nei primi 100 di PAMR.
Altri criteri di ordinamento producono risultati poco coerenti con quelli di PAMR. I risultati di tali confronti sono riportati nella Tabella 7.2, dove è indicato il numero di geni (in percentuale) individuati sia da PAMR sia da ciascun metodo di ordinamento, considerando le prime 100 e le prime 221 posizioni.
I criteri di ordinamento indicati nella prima riga della Tabella 7.2 sono descritti nella Tabella 7.1.
Esempio. Considerando il metodo di ordinamento Rank1 (distanza media / ampiezza media), solo l'8,6% dei primi 221 geni è presente anche nei 221 geni selezionati da PAMR, e nei primi 100 geni solo 7 sono selezionati anche da PAMR.
Il metodo di ordinamento che ha fornito i risultati più coerenti con quelli di PAMR è il Rank12 che considera solo la distanza minima: quasi l'80% dei geni individuati è in comune. I criteri di ordinamento che considerano la distanza minima hanno risultati simili a quelli di PAMR. In generale però non tutti i criteri confermano i geni scelti da PAMR: i

risultati più discordanti sono quelli che considerano la distanza media come criterio di ordinamento (ad es. Rank3 e Rank10).

Rank1	distanza media / ampiezza media
Rank2	distanza media / ampiezza massima
Rank3	distanza media / ampiezza minima
Rank4	distanza massima / ampiezza media
Rank5	distanza massima / ampiezza massima
Rank6	distanza massima / ampiezza minima
Rank7	distanza minima / ampiezza media
Rank8	**distanza minima / ampiezza massima**
Rank9	distanza minima / ampiezza minima
Rank10	distanza media
Rank11	distanza massima
Rank12	**distanza minima**

Tabella 7.1 - Corrispondenze tra indice di ordinamento e criterio usato

#geni	Rank 1	Rank 2	Rank 3	Rank 4	Rank 5	Rank 6	Rank 7	**Rank 8**	Rank 9	Rank 10	Rank 11	**Rank 12**
221	8,6	10,9	3,2	61,5	52,9	11,3	61,5	**62,4**	48,9	2,7	30,3	**79,6**
100	7	6	1	53	49	9	53	**55**	37	0	26	**77**

Tabella 7.2 - Percentuali di geni selezionati con i vari criteri di ordinamento in comune con quelli scelti da PAMR

Concentrando l'attenzione sui due criteri di ordinamento più promettenti, si è verificato che il 62% di geni scelti dal Rank8 è coerente con l'80% di geni scelti dal Rank12. Infatti la percentuale di geni di riferimento comuni sia al Rank8 sia al Rank12 è pari al 58%, molto prossimo al 62% iniziale.
Nel prossimo paragrafo sono elencati i geni risultanti da tale verifica.

7.1.1. Geni selezionati

Considerando entrambi i criteri di ordinamento che ottengono i risultati migliori (distanza minima e rapporto tra distanza minima e ampiezza massima), determiniamo l'elenco di geni in comune con quelli di riferimento. Tali geni, essendo stati selezionati dai criteri dimostratisi analiticamente più coerenti, sono quelli che con maggior probabilità permettono di distinguere le due classi di pazienti anche con nuovi dati, diversi da quelli attualmente analizzati. I nomi dei 128 geni individuati dall'intersezione dei criteri:

- distanza minima,
- distanza minima / ampiezza massima,
- PAMR,

sono riportati nella Tabella 7.3 ordinati per capacità di classificazione decrescente.

1. 221042_s_at	33. 228282_at	65. 224779_s_at	97. 228084_at
2. 226214_at	34. 1554241_at	66. 221494_x_at	98. 202868_s_at
3. 235767_x_at	35. 48808_at	67. 218157_x_at	99. 212493_s_at
4. 227621_at	36. 225636_at	68. 221879_at	100. 226298_at
5. 1553103_at	37. 223370_at	69. 237400_at	101. 213393_at
6. 238974_at	38. 225534_at	70. 236814_at	102. 205780_at
7. AFFX-r2-Bs-thr-3_s_at	39. 225080_at	71. 226300_at	103. 217501_at
8. 229145_at	40. 223320_s_at	72. 214512_s_at	104. 209303_at
9. 204248_at	41. 212981_s_at	73. 200682_s_at	105. 201675_at
10. 204559_s_at	42. 239346_at	74. 212051_at	106. 213616_at
11. 230466_s_at	43. 203166_at	75. 203971_at	107. 218053_at
12. 242648_at	44. 235125_x_at	76. 227466_at	108. 228341_at
13. 212781_at	45. 210908_s_at	77. 209377_s_at	109. 204779_s_at
14. 228670_at	46. 224189_x_at	78. 227600_at	110. 244177_at
15. AFFX-ThrX-3_at	47. 238121_at	79. 218291_at	111. 211025_x_at
16. 229850_at	48. 224972_at	80. 239252_at	112. 229889_at
17. 222294_s_at	49. 226231_at	81. 217877_s_at	113. 225951_s_at
18. 225945_at	50. 200826_at	82. 212204_at	114. 227319_at
19. 226642_s_at	51. 225334_at	83. 201340_s_at	115. 228760_at
20. 212911_at	52. 229842_at	84. 218258_at	116. 202646_s_at
21. AFFX-r2-Bs-phe-M_at	53. 226242_at	85. 214801_at	117. 203941_at
22. 243386_at	54. 232164_s_at	86. 226213_at	118. 238813_at
23. 235766_x_at	55. 222415_at	87. 214766_s_at	119. 225312_at
24. 236225_at	56. 234974_at	88. 220094_s_at	120. 226617_at
25. 219598_s_at	57. 220741_s_at	89. 219192_at	121. 228283_at
26. 230263_s_at	58. 229126_at	90. 209350_s_at	122. 220079_s_at
27. 227356_at	59. 209422_at	91. 226780_s_at	123. 204078_at
28. 220761_s_at	60. 218428_s_at	92. 202034_x_at	124. 208635_x_at
29. 201855_s_at	61. 217962_at	93. 222035_s_at	125. 209911_x_at
30. 207132_x_at	62. 226165_at	94. 229982_at	126. 204274_at
31. 204005_s_at	63. 225523_at	95. 203430_at	127. 226329_s_at
32. 564_at	64. 230329_s_at	96. 203449_s_at	128. 218459_at

Tabella 7.3 - Elenco ordinato dei 128 geni comuni ai criteri di selezione prescelti

Considerando il solo criterio che calcola il rapporto tra distanza minima e ampiezza massima (Rank8), oltre a quelli indicati nella Tabella 7.3, sono selezionati anche i 10 geni riportati nella Tabella 7.4.

1. 221847_at	6. 227126_at
2. 1553978_at	7. 202031_s_at
3. 218946_at	8. 218603_at
4. 212148_at	9. 226024_at
5. 227698_s_at	10. 1560089_at

Tabella 7.4 - Elenco ordinato dei 10 geni aggiuntivi selezionati considerando solo il criterio Rank8

Infine, considerando il solo criterio della distanza minima (Rank12), oltre a quelli indicati nella Tabella 7.3, sono selezionati anche i 48 geni riportati nella Tabella 7.5.

1. 228697_at	13. 202906_s_at	25. 208645_s_at	37. 226106_at
2. AFFX-DapX-3_at	14. 227522_at	26. 213734_at	38. 202625_at
3. AFFX-r2-Bs-dap-3_at	15. 244187_at	27. 232103_at	39. 203403_s_at
4. 239355_at	16. 229235_at	28. 64408_s_at	40. 219356_s_at
5. AFFX-DapX-M_at	17. 223681_s_at	29. 230350_at	41. 224586_x_at
6. 228067_at	18. 217812_at	30. 211711_s_at	42. 201219_at
7. AFFX-r2-Bs-dap-M_at	19. 212079_s_at	31. 203415_at	43. 226287_at
8. AFFX-r2-Bs-phe-3_at	20. 225939_at	32. 211747_s_at	44. 222006_at
9. 224604_at	21. 226510_at	33. 204655_at	45. 212519_at
10. 230265_at	22. 218495_at	34. 229568_at	46. 227787_s_at
11. 229120_s_at	23. 224953_at	35. 210466_s_at	47. 235384_at
12. 218334_at	24. 231513_at	36. 205664_at	48. 212857_x_at

Tabella 7.5 - Elenco ordinato dei 48 geni aggiuntivi selezionati considerando solo il criterio Rank12

I 58 geni aggiuntivi (Tabella 7.4 e Tabella 7.5) possono essere valutati avendo appurato la maggiore adeguatezza di uno dei due rispettivi criteri di ordinamento o qualora ulteriori esperimenti biologici richiedessero un maggior numero di geni da analizzare rispetto ai primi 128 geni individuati. Infine, è possibile estendere l'analisi ai geni che ciascun criterio singolarmente considera promettenti, ma che attualmente sono scartati in quanto non selezionati dall'intersezione con gli altri criteri.

7.2. Dati della prostata

Come illustrato nel Capitolo 6, si sono svolti esperimenti per ogni combinazione di algoritmo SVM, metodo di *feature selection* e funzione di *kernel*.
Per poter effettuare un confronto tra le varie tipologie, si è deciso di impostare a 500 geni il numero di *feature* selezionate da ciascun algoritmo e di utilizzare i parametri ottimizzati del *kernel* forniti dal tool stesso.
I risultati così ottenuti sono mostrati nei grafici seguenti.
Ciascun grafico rappresenta i risultati per un dato algoritmo SVM e per un determinato *kernel*. Sull'asse delle ascisse sono indicati i metodi di *feature selection* utilizzati per l'esperimento considerato, mentre sulle ordinate la percentuale ottenuta come risultato.
Le misure riportate sui grafici rappresentano la *"final accuracy"* calcolata come media delle accuratezze ottenute per ogni split, e la *"final relative classifier information (RCI)"* ottenuta sull'intero training set.

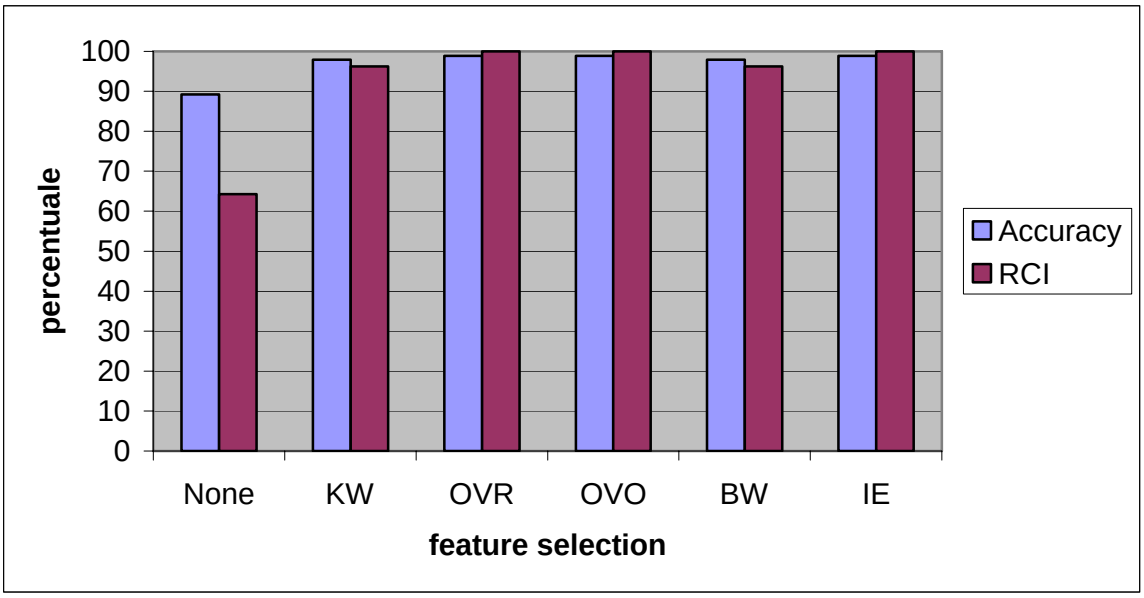

Figura 7.1 - Performance del classificatore CS con *kernel* gaussiano

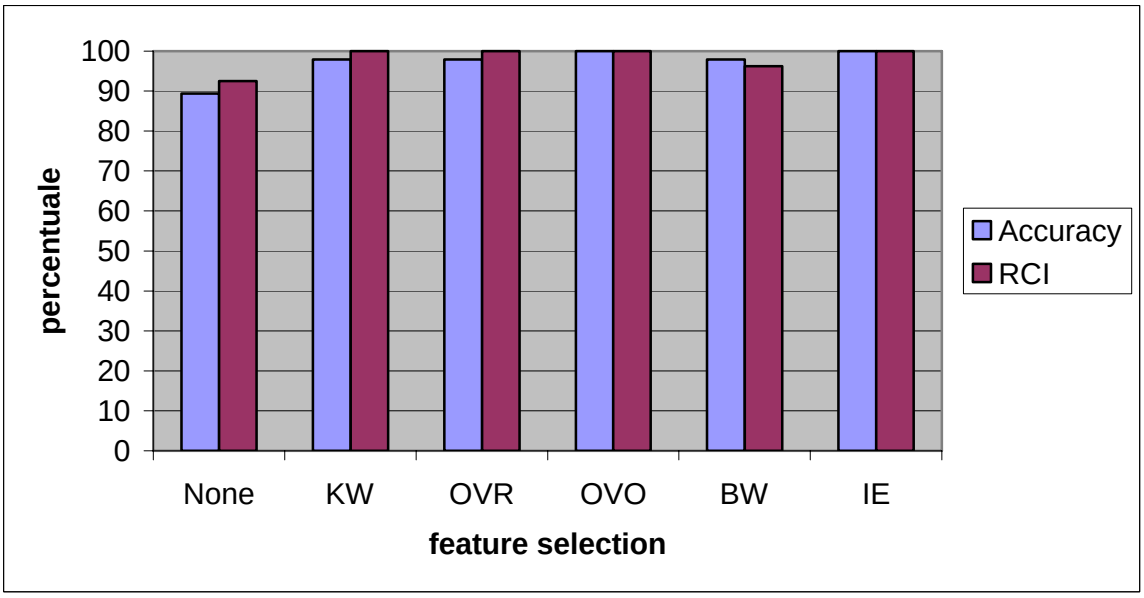

Figura 7.2 - Performance del classificatore CS con *kernel* polinomiale

Le performance ottenute dai due tipi di *kernel* con il metodo CS sotto il profilo dell'accuratezza sono abbastanza simili tra loro.
Per quanto riguarda invece le prestazioni misurate attraverso l'RCI, l'SVM CS con *kernel* polinomiale risulta migliore con qualsiasi algoritmo di *feature selection* abbinato.
La *feature selection* risulta comunque più importante per gli SVM con *kernel* gaussiano. Infatti se per l'accuratezza si ha un miglioramento del 8-10% per entrambe le tipologie di *kernel*, il miglioramento del RCI è rilevante per il *kernel* gaussiano, 30-35%, mentre più contenuto per quello polinomiale, 5-8%.
I metodi migliori di selezione sono OVO e IE con un'*accuracy* pari al 98,9% per il *kernel* gaussiano e al 100% per quello polinomiale, e con un RCI del 100% per entrambi. Il metodo peggiore, invece, risulta BW che ottiene un'accuratezza del 97,9% e un RCI del 96,2%, comunque migliore del modello costruito senza *feature selection*.
Per gli algoritmi di selezione mediamente l'accuratezza è:

- per *kernel* gaussiano pari a 98,5%
- per *kernel* polinomiale pari a 98,7%

mentre l'RCI è:

- per *kernel* gaussiano pari a 98,5%
- per *kernel* polinomiale pari a 99,2%

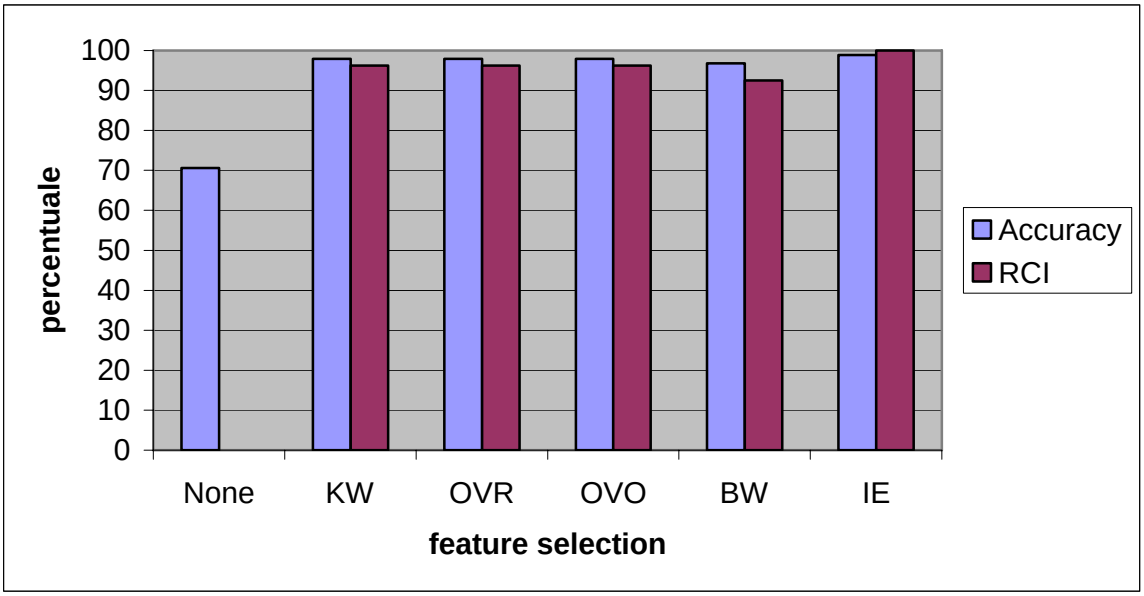

Figura 7.3 - Performance del classificatore DAG con *kernel* gaussiano

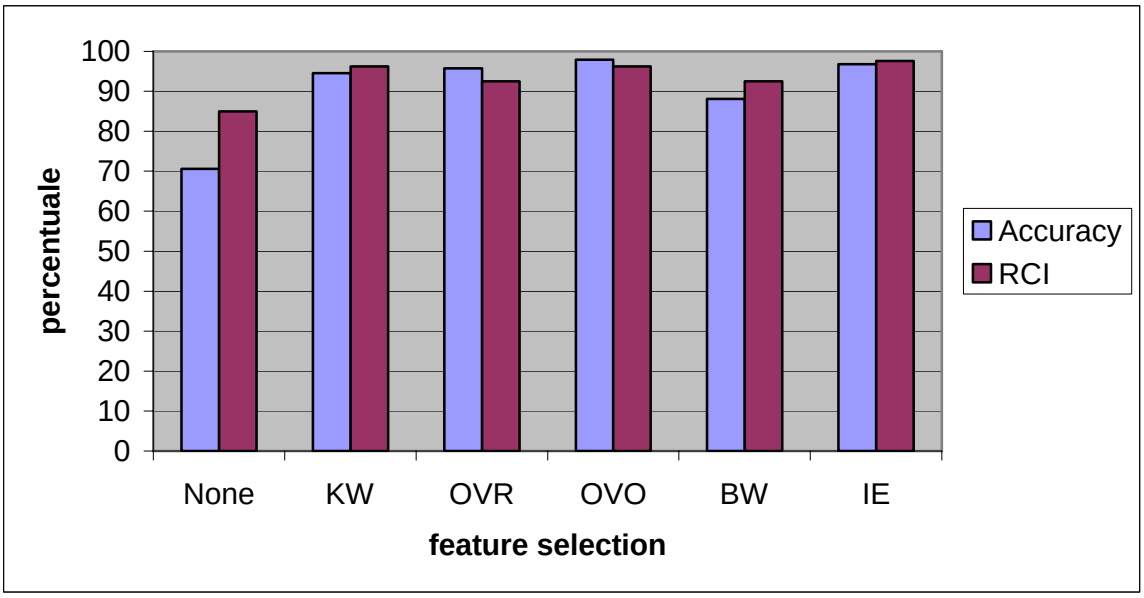

Figura 7.4 - Performance del classificatore DAG con *kernel* polinomiale

I classificatori costruiti utilizzando il metodo DAG ottengono accuratezze uguali o migliori con l'utilizzo di un *kernel* gaussiano rispetto ad uno polinomiale.
Inoltre l'RCI dei modelli con *kernel* gaussiani ottiene dei notevoli miglioramenti con l'utilizzo dei metodi di *feature selection* Si passa infatti dallo 0% al 92-100%.
L'algoritmo di selezione migliore, considerando le performance per entrambi i tipi di *kernel*, risulta quello degli intervalli di espressione, poiché raggiunge un'accuratezza del 98,9% e un RCI del 100%, prestazioni non eguagliate da nessun altro algoritmo.
Il metodo peggiore risulta il BW che ottiene un'accuratezza nel caso di *kernel* polinomiale pari al 88,1%, più basso del 6-8% rispetto agli altri.
Mediamente i metodi di *feature selection* hanno un'accuratezza del:

- 97,9% per *kernel* gaussiano
- 94,6% per *kernel* polinomiale

mentre l'RCI pari al:

- 96,2% per *kernel* gaussiano
- 95% per *kernel* polinomiale

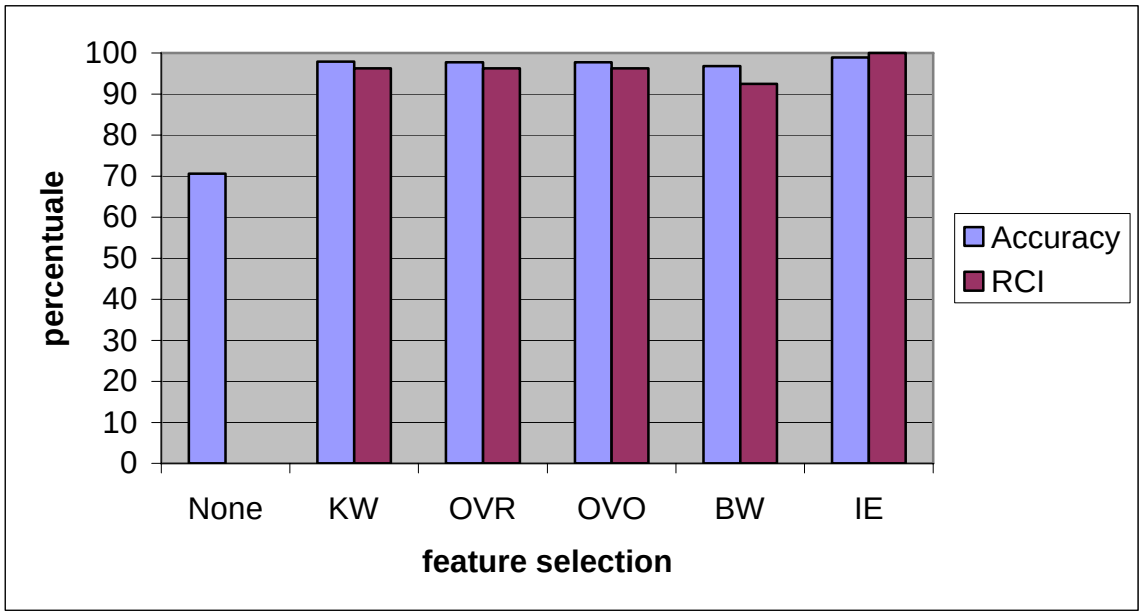

Figura 7.5 - Performance del classificatore OVO con *kernel* gaussiano

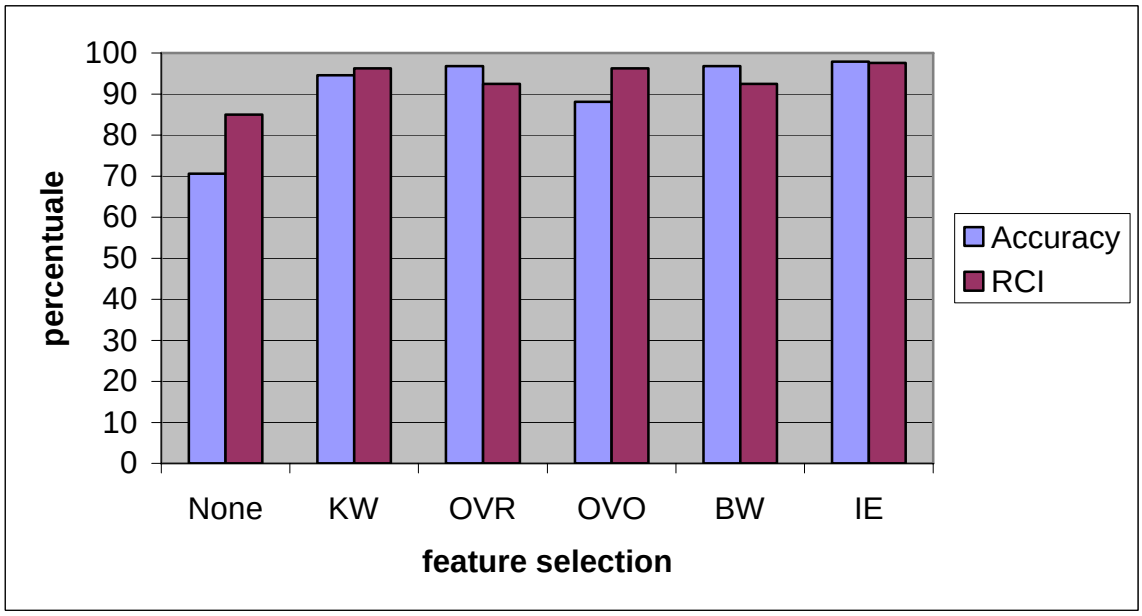

Figura 7.6 - Performance del classificatore OVO con *kernel* polinomiale

I modelli costruiti con SVM di tipo OVO risultano migliori in fatto di accuratezza se utilizzano un *kernel* di tipo gaussiano, oltre all'impiego di algoritmi di *feature selection*.

Infatti si ha un miglioramento del 26-28% per il kernel gaussiano rispetto al 18-27% per quello polinomiale.
Per quanto riguarda le performance calcolate tramite RCI utilizzando la *feature selction*, si ottiene una prestazione pari al 100% solo nel caso dell'utilizzo del metodi degli intervalli di espressione. Questo metodo risulta pertanto il migliore, in quanto anche per l'accuratezza ottiene dei risultati pari al 98-99% con le due tipologie di *kernel*.
Il metodo peggiore, invece, può essere identificato nell'algoritmo BW che ottiene un 96,8% di *accuracy* e un 92,5% di RCI per entrambi i *kernel*, anche se il metodo di selezione OVO per il *kernel* polinomiale ha una performance di 88,1% di accuratezza, inferiore al metodo BW.
La selezione dei geni, comunque, porta dei miglioramenti con tutte le tipologie di *kernel*, anche se più evidente per quello gaussiano. Infatti per quest'ultimo si passa dallo 0% senza selezione al 92,5-100%, mentre per l'altra tipologia si ha un incremento delle prestazioni del 7-12%.
Mediamente la *feature selection* ottiene risultati di accuratezza pari al:

- 97,8% per *kernel* gaussiano
- 94,8% per *kernel* polinomiale

e di RCI pari al:

- 96,8% per *kernel* gaussiano
- 95% per *kernel* polinomiale

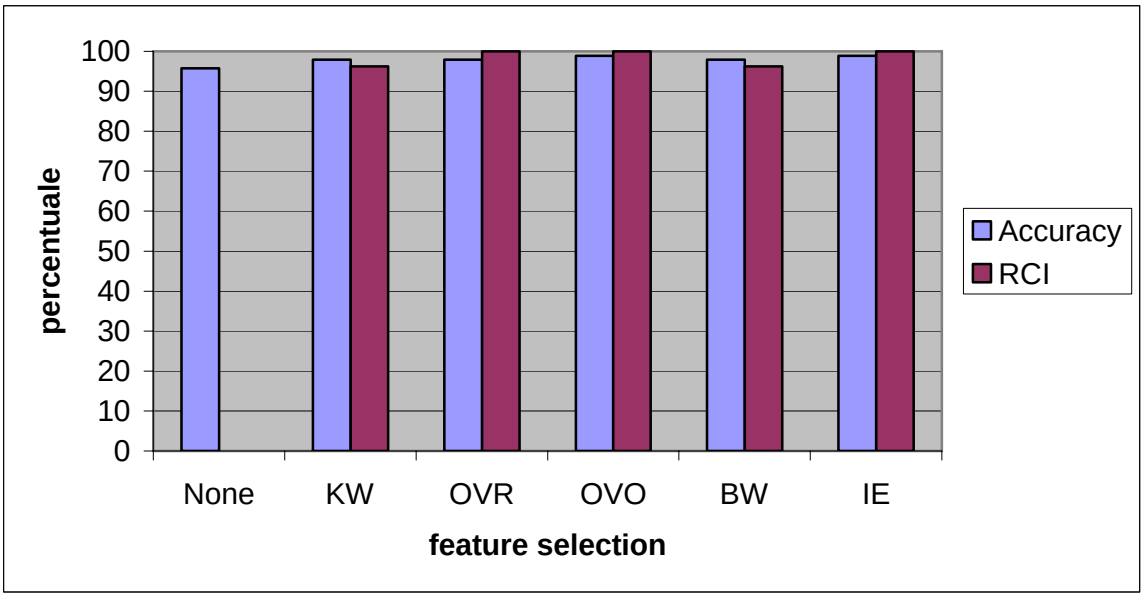

Figura 7.7 - Performance del classificatore OVR con *kernel* gaussiano

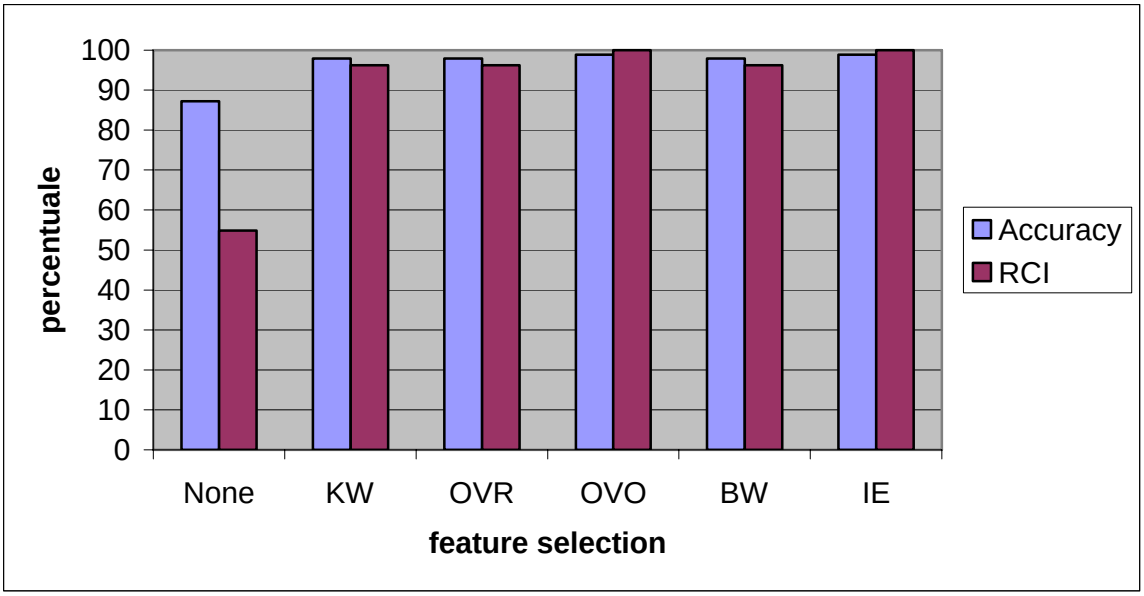

Figura 7.8 - Performance del classificatore OVR con *kernel* polinomiale

Per i classificatori OVR le prestazioni tra le due tipologie di *kernel* e i vari metodi di *feature selection* non sono molto diverse, sia calcolate con l'accuratezza che con l'RCI. I benefici portati dalla *feature selection* per l'accuratezza si notano maggiormente con il *kernel* polinomiale, seppur la differenza non sia così significativa come con altri algoritmi di classificazione.
Infatti si ha un miglioramento per il *kernel* polinomiale del 10-11%, mentre per quello gaussiano l'incremento delle prestazioni è del 2-3%.
Per quanto riguarda le performance misurate con l'RCI i benefici della selezione sono migliori per il *kernel* gaussiano in quanto si passa dallo 0% al 96-100%, mentre per il polinomiale si parte dal 55% per arrivare al 96-100%.
I metodi migliori sono OVO e IE che per entrambe le tipologie di *kernel* ottengono 98,9% di accuratezza e 100% di RCI.
I peggiori invece risultano KW e BW che hanno un'accuratezza del 97,9% e un RCI del 96,2% per entrambi i *kernel*.
Il comportamento medio dei metodi di *feature selection* per l'*accuracy* è:

- 98,3% per il *kernel* gaussiano
- 98,3% per il *kernel* polinomiale

e per l'RCI è:

- 98,5% per il *kernel* gaussiano
- 97,7% per il *kernel* polinomiale

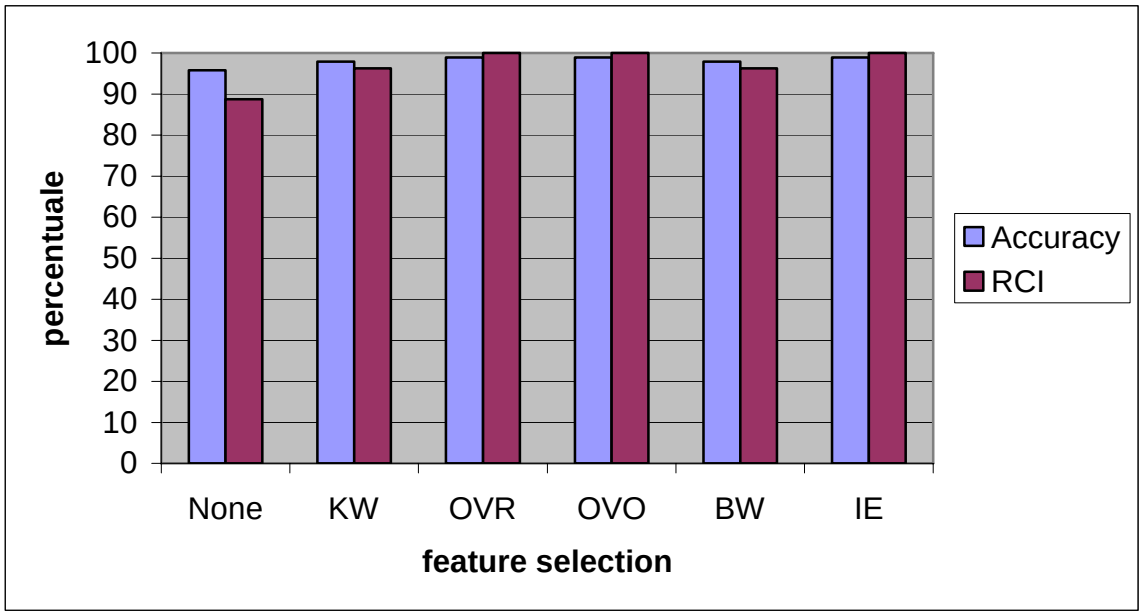

Figura 7.9 - Performance del classificatore WW con *kernel* gaussiano

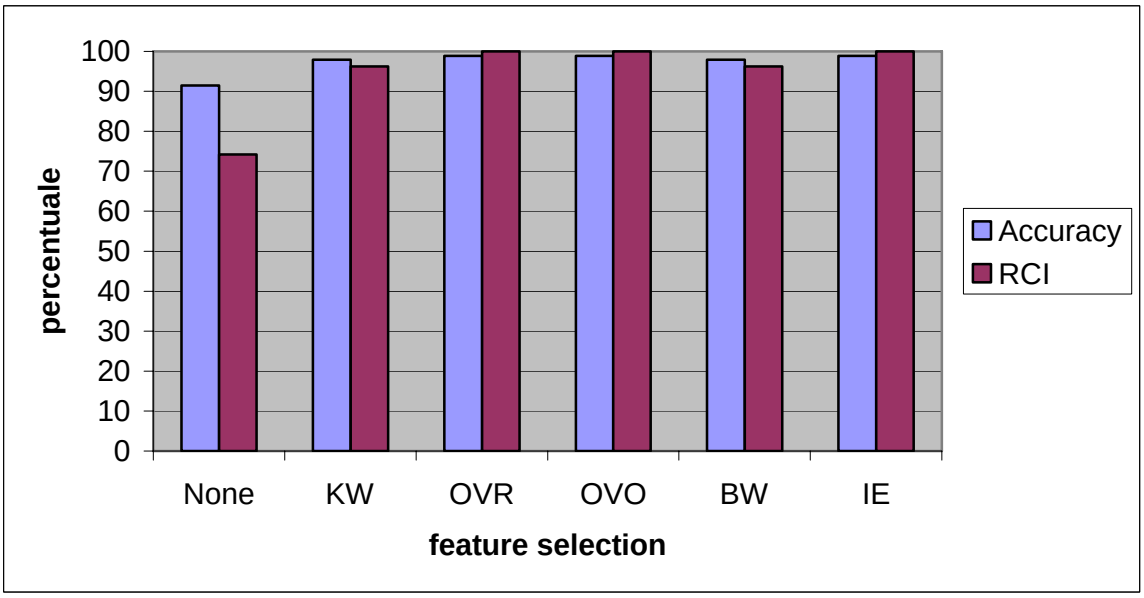

Figura 7.10 - Performance del classificatore WW con *kernel* polinomiale

Le performance del classificatore WW sono molto simili sia con *kernel* guassiano che polinomiale. Inoltre l'impiego della *feature selection* non comporta notevoli miglioramenti in quanto già il classificatore da solo ottiene degli ottimi risultati in fatto di accuratezza e di RCI, fatta eccezione per quello polinomiale calcolato tramite RCI.

Infatti per il *kernel* gaussiano si ha un incremento delle prestazioni del 2-3% per l'*accuracy* e del 7-11% per l'RCI. Per quello polinomiale il miglioramento risulta essere del 6-7% per l'accuratezza e del 22-26% per l'RCI.
Non essendoci grandi differenze di prestazioni i metodi migliori risultano OVR, OVO e IE con accuratezze del 98,9% e con RCI del 100%, mentre gli algoritmi peggiori sono KW e BW. Questi ultimi ottengono delle prestazioni in termini di *accuracy* pari al 97,9% e per l'RCI pari al 96,2%.
Mediamente il comportamento degli algoritmi di *feature selection* è del 98,5% sia per l'accuratezza che per l'RCI per entrambe le tipologie di *kernel*.
Possiamo notare da questi risultati che la *feature selection* ha un'influenza molto rilevante sulle performance dei modelli costruiti dagli algoritmi SVM.
Ogni metodo di *feature selection* identifica, un elenco ordinato di geni in base a criteri "statistici", senza valutarne le prestazioni di classificazione. Pertanto il passo di *feature selection* è indipendente dalla costruzione dei modelli o da qualsiasi altro passo dell'algoritmo di classificazione.
Sui risultati ottenuti bisogna fare ancora alcune osservazioni.
Come si nota dai grafici il metodo degli intervalli di espressione, che abbiamo ricavato empiricamente dagli esperimenti sui dati del colon, risulta quasi sempre il migliore per le prestazioni che produce, addirittura arrivando ad un risultato del 100% nel caso del CS con *kernel* polinomiale.
Questi risultati da un lato non ci stupiscono poiché questo metodo affronta il problema in modo binario, selezionando i geni che distinguono al meglio due classi, senza considerare i componenti di altre classi. Quindi il metodo degli intervalli di espressione scompone il problema di selezione multiclasse in problemi binari di complessità inferiore e di facile interpretazione.
Un'analisi approfondita mostra che i risultati ottenuti sono influenzati da una serie di errori che si verificano sempre sullo stesso gruppo di pazienti.
Per comprendere meglio questo aspetto sono stati analizzati tutti gli esperimenti svolti e calcolato il numero di errori commessi dai vari classificatori, differenziati solamente per tipologia di *kernel*, per ogni campione.
Come possiamo vedere dai grafici in Figura 7.11 e in Figura 7.12 ci sono tre pazienti, i cui numero d'ordine sono 5, 12 e 27, che vengono classificati in modo errato in molti esperimenti.
Questo ci fa capire che questi campioni sono dei punti particolari nello spazio N-dimensionale dei valori di espressione dei geni e molto probabilmente sono punti limitrofi per la determinazione del vettore di supporto.
Per questo motivo risultano di grande interesse, dal momento che la loro presenza o meno influisce in modo significativo nelle performance del classificatore. Infatti se questi campioni sono parte del training set le performance risultano migliori di quanto vengono considerati nel test set.

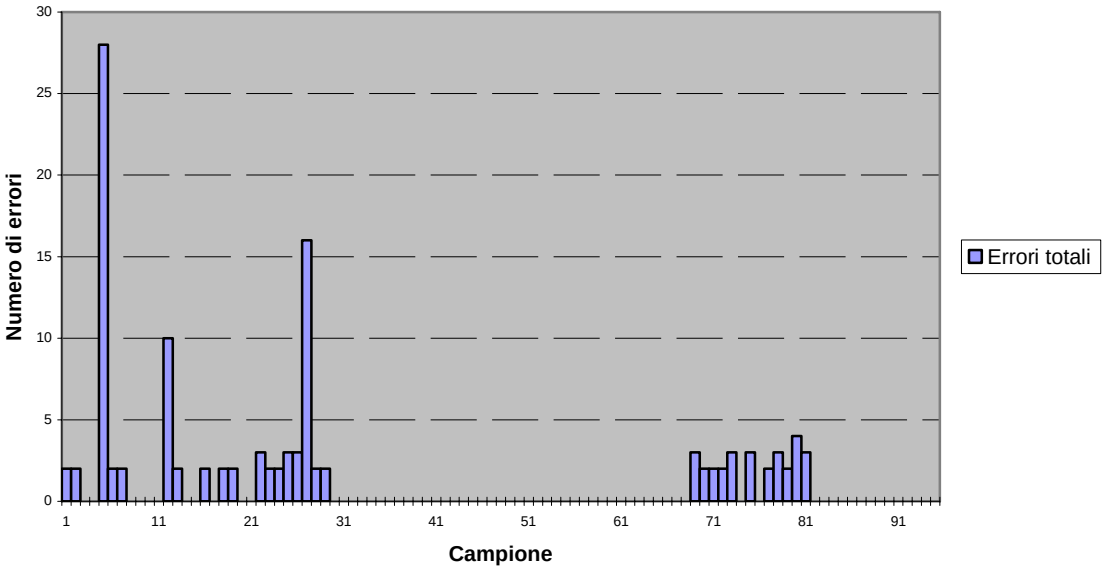

Figura 7.11 - Errori totali per campione con *kernel* gaussiano

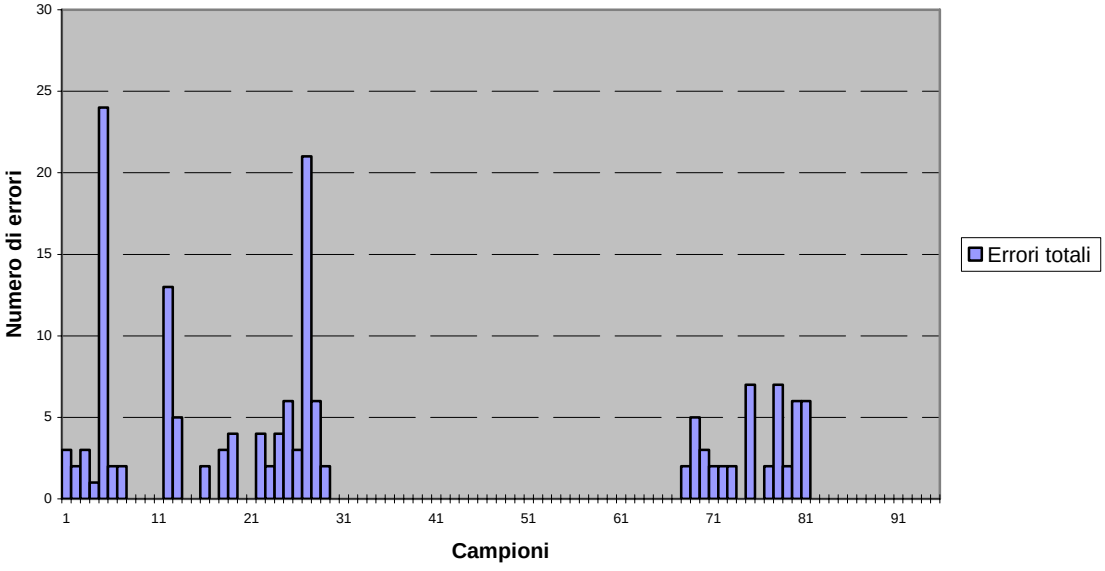

Figura 7.12 - Errori totali per campione con *kernel* polinomiale

7.2.1. Influenza della feature selection sulle prestazioni del classificatore

L'influenza della selezione di alcuni geni, considerati dall'algoritmo in esame come i migliori, non porta solo ad un miglioramento delle prestazioni sotto l'aspetto dell'accuratezza delle prestazioni del classificatore, ma anche nel tempo di calcolo per la costruzione del classificatore stesso.
Per esempio, utilizzando un calcolatore dotato di un processore Centrino da 1.8 GHz con una memoria RAM da 1 GB, la generazione di un modello di classificazione con l'algoritmo CS senza *feature selection* comporta un tempo di calcolo di 457 secondi, mentre con la *feature selection* di 500 geni con il metodo BW un tempo di 21,3 secondi. Anche la stima delle performance del classificatore viene influenzata dalla selezione delle *feature*. Infatti, con le stesse condizioni dell'esempio precedente, senza *feature selection* il tempo di esecuzione è pari a 4037 secondi, ossia 1 h 07 minuti e 17 secondi, mentre con il metodo BW solamente 108,5 secondi, ossia meno di due minuti.
Questa osservazione è di facile comprensione se pensiamo solamente alla complessità del problema di ottimizzazione del SVM. Nel caso di *feature selection* si viene a ridurre il numero di dimensioni su cui l'SVM deve lavorare, portando quindi a una diminuzione sostanziale del numero di operazioni svolte dall'algoritmo.
Un aspetto interessante è che i vari metodi non riescono ad ottenere prestazioni uguali a parità di geni selezionati. Ciò è dovuto alle diverse "informazioni statistiche" che i vari algoritmi considerano, che possono portare a un vantaggio nella costruzione del classificatore, ma anche ad un peggioramento della situazione.
Con il seguente esempio vogliamo illustrare i pro e i contro che la *feature selection* comporta.

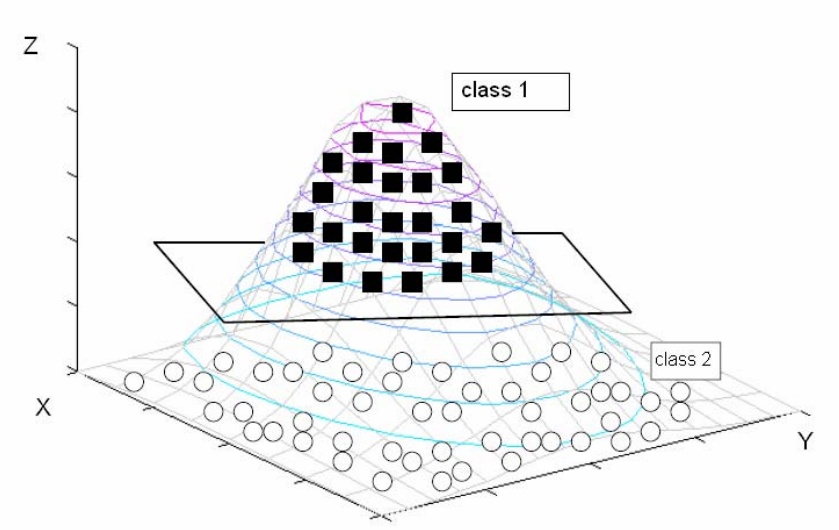

Figura 7.13 - Esempio di distribuzione dei campioni in uno spazio 3D

Se consideriamo il problema binario mostrato in Figura 7.13, possiamo notare che i punti appartenenti alle due classi possono essere divisi facilmente da un piano parallelo al piano XY. Naturalmente essendo un problema in tre dimensioni comporterà una mole di calcolo maggiore di uno a due dimensioni. Per questo motivo vogliamo applicare una selezione sulle dimensioni (i geni) per ridurre il tempo di calcolo e cercare di migliorare ancora le prestazioni se è possibile.
In Figura 7.14 e in Figura 7.15 sono rappresentate rispettivamente le proiezioni sui piani XY e YZ dei punti delle due classi. Come si può ben notare la proiezione sul piano YZ rende di facile costruzione un classificatore SVM di tipo binario, mentre l'altra proiezione rende il calcolo molto complesso e sicuramente con un'accuratezza inferiore.

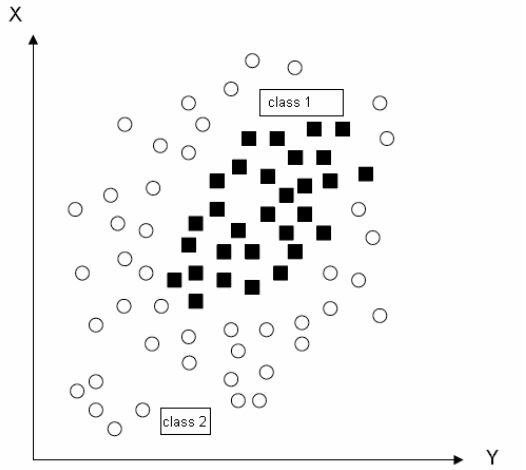

Figura 7.14 - Proiezione dei campioni sul piano XY

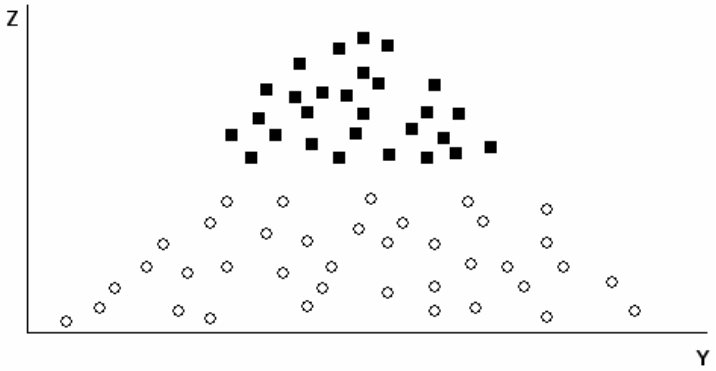

Figura 7.15 - Proiezione dei campioni sul piano YZ

Con questo esempio si è voluto far notare l'importanza che la *feature selection* ricopre nella costruzione del modello e la sua influenza sulle performance di quest'ultimo.

Per verificare il ragionamento descritto sui dati in nostro possesso, abbiamo eseguito alcuni degli esperimenti analizzati in precedenza selezionando solamente 50 geni. I risultati ottenuti sono quelli mostrati in Figura 7.16.
Si nota che su dieci casi analizzati, 4 hanno un miglioramento delle prestazioni, 4 invece un peggioramento delle prestazioni e 2 rimangono invariati.
Pertanto se si scelgono 50 geni con un criterio che possa minimizzare gli effetti negativi mostrati precedentemente, si ottengono vari benefici.
Per prima cosa l'algoritmo di classificazione viene reso più veloce in termini di tempo di esecuzione, in quanto il carico computazionali è minore.
Un altro beneficio è quello che possiamo notare dai risultati ottenuti. Mediamente con 50 geni l'accuratezza è del 97,8%, mentre con 500 geni si ottiene un risultato del 97,2%. Quindi le prestazioni mediamente sono migliori con 50 geni, anche se non in modo netto.

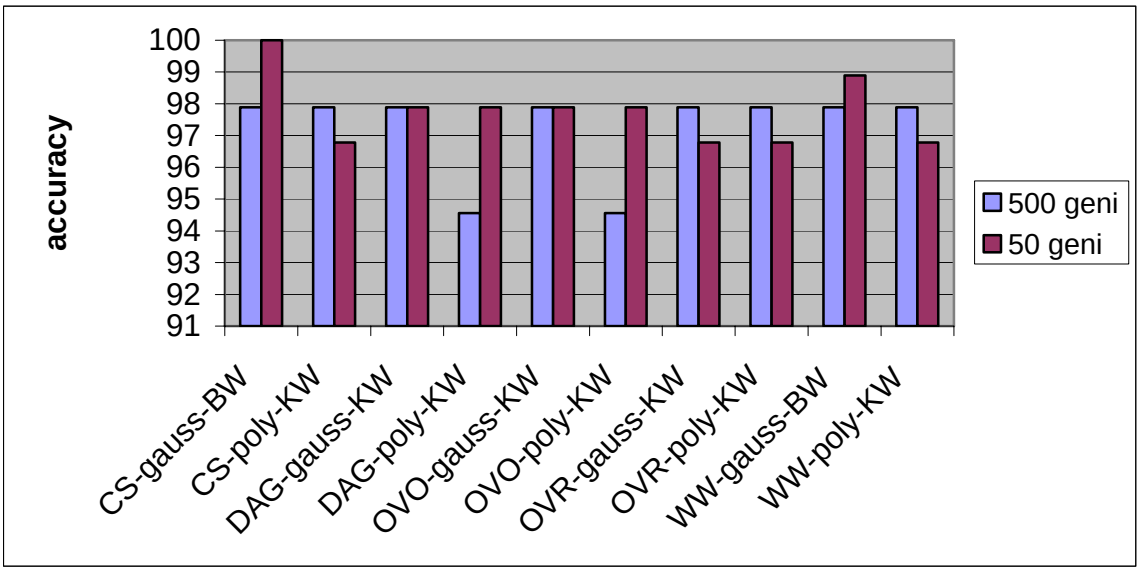

Figura 7.16 - Comparazione feature selection di 500 geni vs 50 geni

7.2.2. Risultati di mcSVM

Dopo aver analizzato i risultati che ha prodotto GEMS, ed aver verificato che la *feature selection* migliora notevolmente le prestazioni in termini di accuratezza per tutti i metodi SVM messi a disposizione da questo tool, si è passati a valutare l'importanza della selezione con mcSVM.
Dal momento che mcSVM non fornisce di suo una selezione degli attributi, si sono presi come riferimento i geni selezionati sia da GEMS per ogni algoritmo disponibile sia dal metodo degli intervalli di espressione. Questi geni sono stati selezionati considerando l'intero dataset, perciò sono indipendenti dal *training set* utilizzato nelle varie prove.

Per avere un'effettiva visione dell'efficacia della selezione e dell'importanza del training set, si sono divisi in modo casuale i campioni a nostra disposizione, come è stato illustrato nel Capitolo 6.
Come possiamo notare dai grafici sotto (Figura 7.17, Figura 7.18, Figura 7.19, Figura 7.20, Figura 7.21, Figura 7.22, Figura 7.23 e Figura 7.24), la *feature selection* porta quasi sempre notevoli miglioramenti. In particolar modo gli SVM con *kernel* gaussiano guadagnano notevolmente in termini di accuratezza rispetto ai classificatori senza *feauture selection* ma anche rispetto a quelli con *kernel* polinomiale.
Per quanto riguarda il metodo degli intervalli di espressione, i risultati sono abbastanza soddisfacenti, anche se non al livello di quelli precedenti. Infatti in alcuni casi questo metodo non risulta essere il migliore, anzi addirittura il peggiore come nel caso del training set con 70 campioni e test set con 25 campioni con *kernel* di tipo gaussiano. Nonostante le prestazioni non sempre ottimali, il metodo degli intervalli di espressione apporta notevoli benefici al classificatore rispetto all'inutilizzo della *feature selection*, migliorando sempre le prestazioni sia in termini di accuratezza che di velocità di creazione del modello.
Un discorso a parte bisogna fare per gli esperimenti con un training set di 21 campioni e un test set di 74. Questo esperimento è stato fatto per annullare la probabilità a priori delle varie classi, derivante dal numero così diversificato dei componenti di ognuna, e valutare l'influenza della distribuzione dei dati.
Il training set, infatti, è composto da 7 pazienti per ogni classe, quindi si ha una probabilità a priori per ciascuna classe pari al 33,3%. In questo caso si è ottenuto che il classificatore con *kernel* gaussiano si comporta come negli esperimenti precedenti, ossia presenta un miglioramento delle prestazioni con l'utilizzo della *feature selection*, mentre quello con *kernel* polinomiale ha dei gravi problemi di classificazione.
Infatti le prestazioni con il *kernel* polinomiale e con *feature selection* peggiorano in alcuni casi rispetto al classificatore senza selezione. Inoltre la classe che prima era dominante come distribuzione diventa la più difficile da classificare.
La spiegazione a questo comportamento è da ricercare nella tipologia del *kernel*.
Un *kernel* gaussiano ipotizza che la distribuzione dei dati sia simile a quella del training set, proprio attraverso una distrubuzione gaussiana. Il *kernel* polinomiale, invece, utilizza un polinomio interpolante per determinare la distribuzione e quindi risolvere il problema di ottimizzazione del SVM. In questo modo i classificatori costruiti con quest'ultima tipologia di *kernel* sono facilmente influenzabili dai dati rumorosi, dagli *outlier* e da distribuzioni dei dati particolari.
Di seguito presentiamo i risultati per ogni coppia di training set e test set utilizzati, divisi per tipologia di *kernel*.

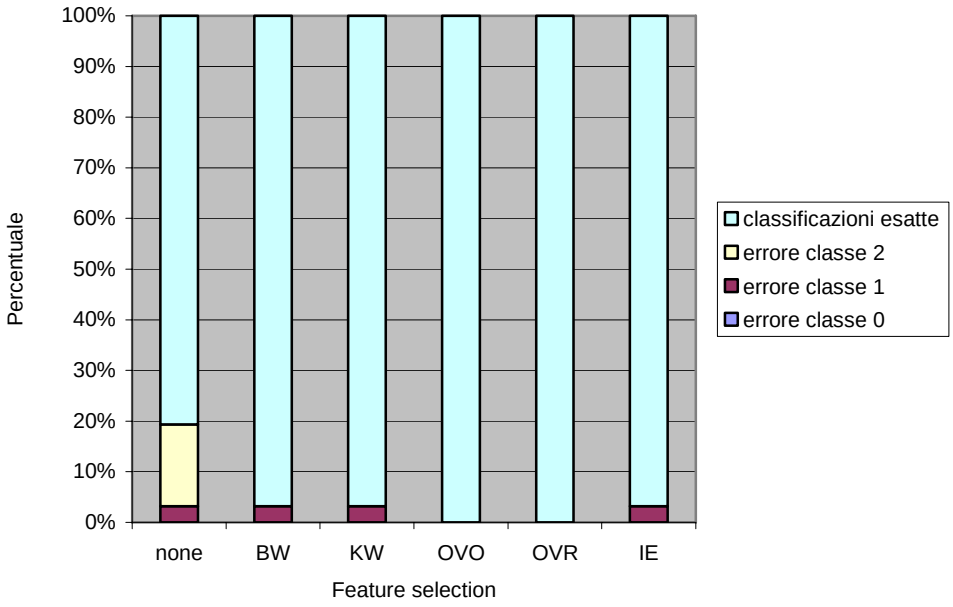

Figura 7.17 - Valutazione mcSVM con *kernel* gaussiano in base alle varie *feature selection* per training set di 60 campioni e test set di 31 campioni

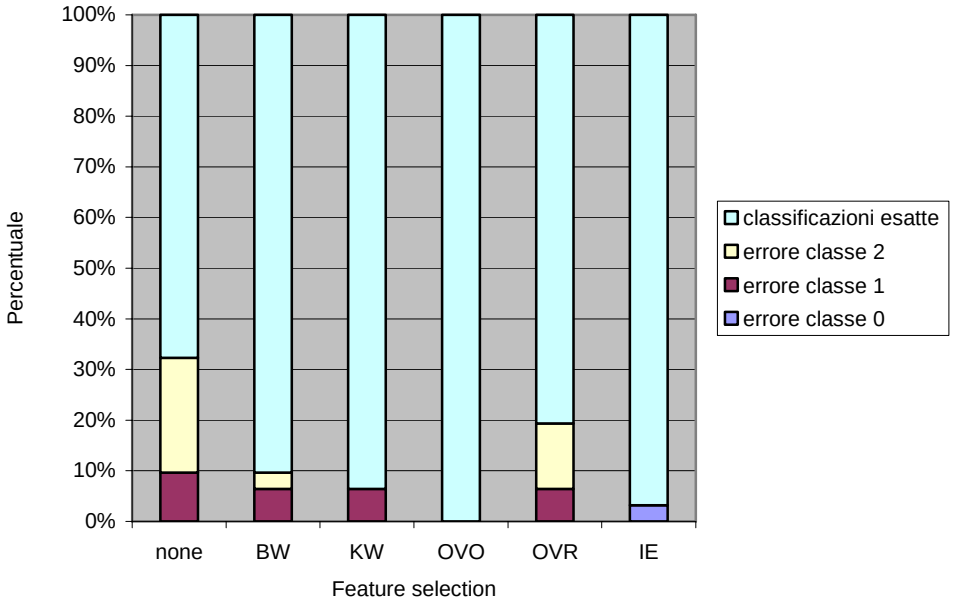

Figura 7.18 - Valutazione mcSVM con *kernel* polinomiale in base alle varie *feature selection* per training set di 60 campioni e test set di 31 campioni

I classificatori costruiti con *kernel* gaussiano ottengono prestazioni migliori rispetto a quelli polinomiali, sia senza che con l'utilizzo della *feature selection*. In entrambe le tipologie di *kernel* il metodo di selezione OVO risulta essere il migliore con un'accuratezza del 100%.
Gli algoritmi di selezione diminuiscono l'errore effettuato senza *feature selection* per i campioni della classe 2, azzerandolo il più delle volte a parte i casi con *kernel* polinomiale e metodi di selezione BW e OVR. La classe 1 risulta invece quasi sempre critica dal momento che non sempre l'errore viene eliminato grazie alla selezione. Il metodo degli intervalli di espressione è l'unico che introduce nel caso del *kernel* polinomiale un errore per la classe 0.

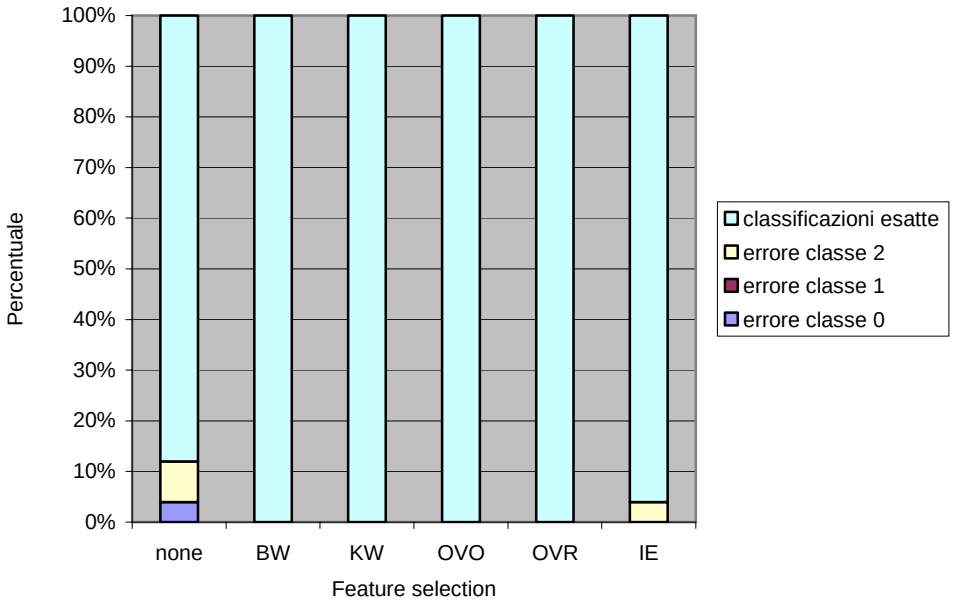

Figura 7.19 - Valutazione mcSVM con *kernel* gaussiano in base alle varie *feature selection* per training set di 70 campioni e test set di 25 campioni

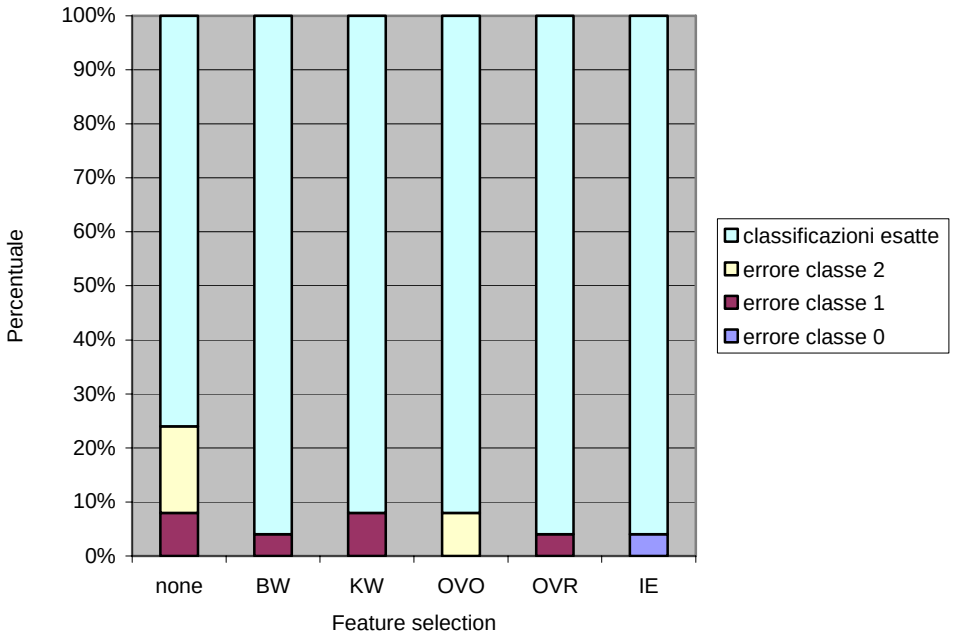

Figura 7.20 - Valutazione mcSVM con *kernel* polinomiale in base alle varie *feature selection* per training set di 70 campioni e test set di 25 campioni

Con un training set composto da 70 campioni il *kernel* gaussiano ottiene prestazioni migliori rispetto al *kernel* polinomiale. Inoltre utilizzando i metodi di selezione BW, KW, OVO, OVR, si ottengono accuratezze del 100%.
L'SVM costruito con *kernel* polinomiale ottiene però notevoli benefici dall'impiego di metodi di selezione, passando da un'accuratezza del 76% a prestazioni superiori al 90%.
Con il *kernel* gaussiano solamente il metodo degli intervalli di espressione effettua un errore nella classe 2.
Con il *kernel* polinomiale invece i vari metodi si comportano differentemente commettendo degli errori sia nella classe 1 e 2. Solamente il metodo degli intervalli di espressione commette un errore nella classe 0, classificando però in modo esatto tutti i campioni delle altre classi.

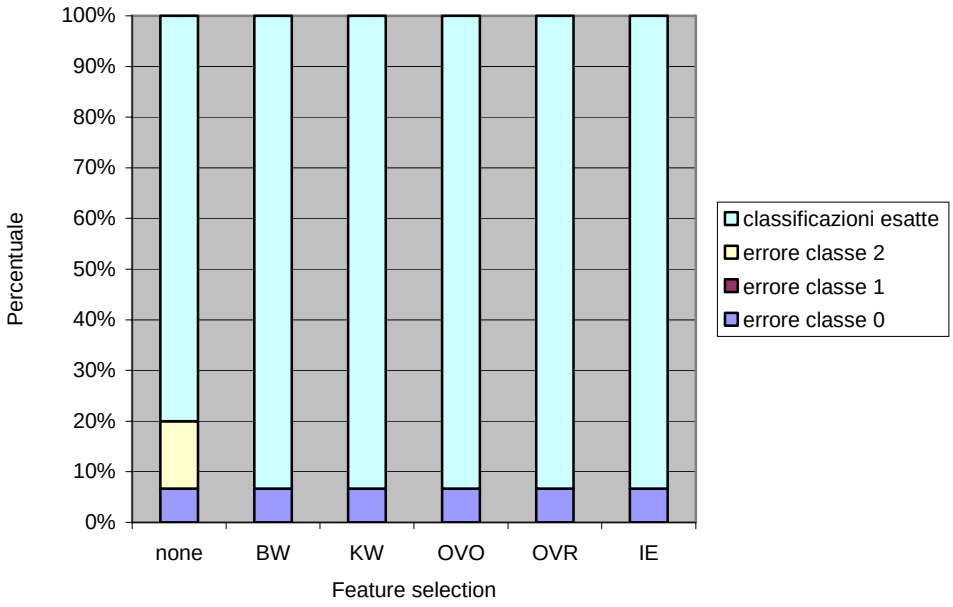

Figura 7.21 - Valutazione mcSVM con *kernel* gaussiano in base alle varie *feature selection* per training set di 80 campioni e test set di 15 campioni

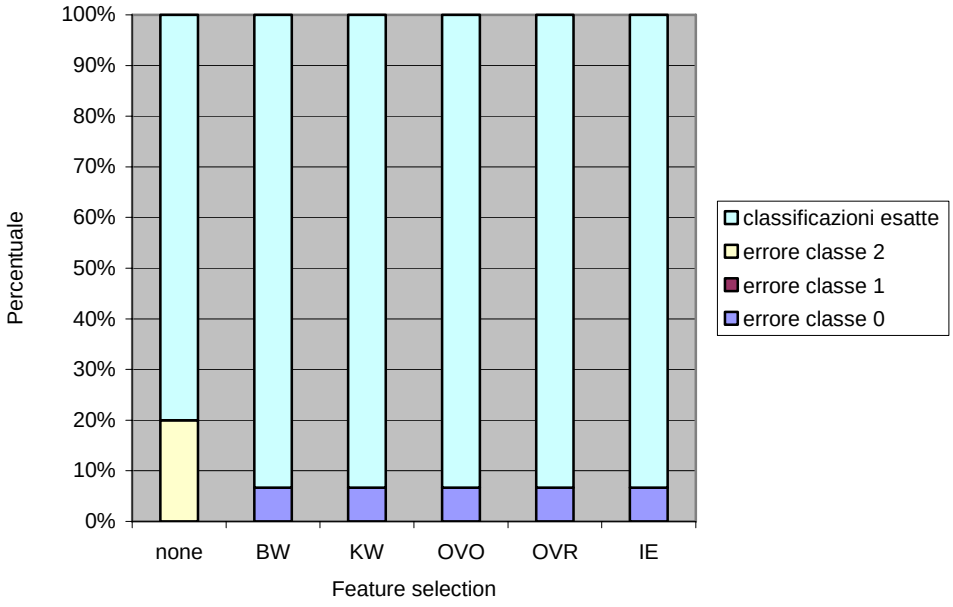

Figura 7.22 - Valutazione mcSVM con *kernel* polinomiale in base alle varie *feature selection* per training set di 80 campioni e test set di 15 campioni

Le prestazioni con un *training set* di 80 campioni sono uguali con qualsiasi tipologia di *kernel* utilizzata. L'unica differenza è nei modelli costruiti senza *feature selection*, dove il *kernel* polinomiale classifica senza errore i campioni della classe 0, sbagliando però a classificare dei campioni della classe 2, mentre quello gaussiano sbaglia anche nella classe 0, riducendo però l'errore nella classe2.
Sia con *kernel* gaussiano sia con kernel polinomiale i metodi di *feature selection* commettono un errore nella classe 0.
Nonostante questo errore le prestazioni passano da un 80% di accuratezza senza l'utilizzo di algoritmi di selezione al 93,3%.

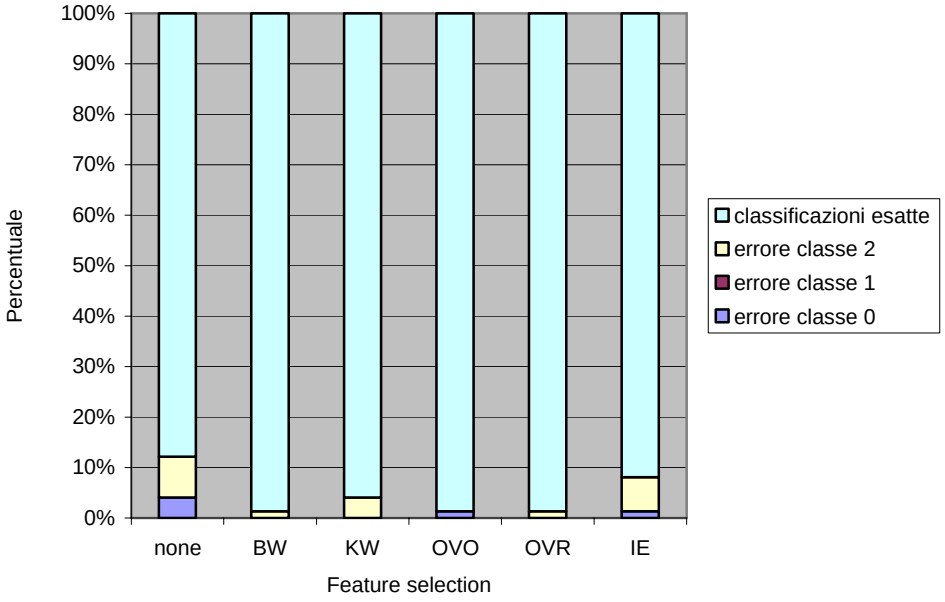

Figura 7.23 - Valutazione mcSVM con *kernel* gaussiano in base alle varie *feature selection* per training set di 21 campioni e test set di 74 campioni

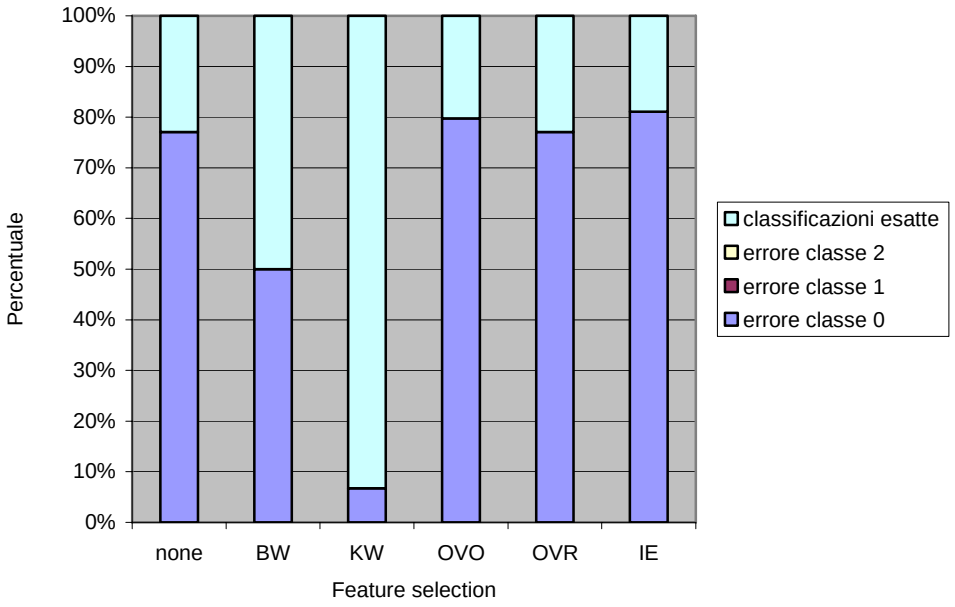

Figura 7.24 - Valutazione mcSVM con *kernel* polinomiale in base alle varie *feature selection* per training set di 21 campioni e test set di 74 campioni

I modelli costruiti con *kernel* polinomiale su un training set di 21 campioni, 7 per classe, hanno notevoli problemi di classificazione, a causa proprio della tipologia di *kernel*. Solamente il metodo KW ottiene performance accettabili, sbagliando solo 5 campioni della classe 0, avendo così un'accuratezza del 93,2%. Per i metodi degli intervalli di espressione e OVO le prestazioni addirittura peggiorano.
Il *kernel* gaussiano, invece, ottiene dei risultati accettabili e confrontabili come prestazioni con i risultati ottenuti con i training set precedenti. Infatti si passa da un'*accuracy* del 87,8% al 91,9-98,6%.

7.2.3. Geni selezionati

Negli esperimenti descritti finora si è valutata l'influenza che la *feature selection* ha sulle prestazione dei classificatori, senza far riferimento in modo preciso a quali dei 17694 geni siano stati selezionati.
Per rendere più agevole la consultazione dei geni selezionati e le osservazioni in merito ai risultati che si ottengono tramite i vari metodi di *feature selection*, l'elenco dei geni selezionati singolarmente da ciascun metodo sono riportati nell'Appendice A.
In questa sezione vengono analizzati i vari algoritmi di selezione al fine di determinare tra tutti i geni selezionati quali siano i migliori in assoluto. Per fare ciò si è analizzato il numero di geni comune ai vari algoritmi.

La matrice presentata in Tabella 7.6 mostra il numero di geni in comune agli algoritmi presi due a due.

	BW	KW	S2N_OVO	S2N_OVR	IE
BW	-	273	250	280	80
KW	273	-	147	335	37
S2N_OVO	250	147	-	239	172
S2N_OVR	280	335	239	-	68
IE	80	37	172	68	-

Tabella 7.6 - Numero di geni in comune per coppie di metodi di *feature selection*

Da notare come la riga e la colonna, che corrispondono alle combinazioni con il metodo degli intervalli di espressione siano nettamente inferiori alle altre. Questo è dovuto al fatto che il metodo degli intervalli di espressione seleziona solamente 231 geni, mentre gli altri algoritmi ne hanno selezionati 500.

Combinazione metodi	Numero di geni in comune
BW + KW + S2N_OVO	118
BW + KW + S2N_OVR	213
BW + KW + IE	30
BW + S2N_OVO + S2N_OVR	172
BW + S2N_OVO + IE	80
BW + S2N_OVR + IE	54
KW + S2N_OVO + S2N_OVR	132
KW + S2N_OVO + IE	36
KW + S2N_OVR + IE	26
S2N_OVO + S2N_OVR + IE	68
BW + KW + S2N_OVO + S2N_OVR	109
BW + KW + S2N_OVO + IE	30
BW + KW + S2N_OVR + IE	24
BW + S2N_OVO + S2N_OVR + IE	54
KW + S2N_OVO + S2N_OVR + IE	26
BW + KW + S2N_OVO + S2N_OVR + IE	24

Tabella 7.7 - Numero di geni in comune per tutte le combinazioni dei metodi di *feature selection*

In Tabella 7.7 sono consideriate le combinazioni di più di due metodi ed è riportato il numero di geni in comune dalle varie combinazioni.
Da questi confronti possiamo notare che ci sono delle combinazioni che risultano avere lo stesso numero di geni in comune, nonostante il numero di metodi utilizzati sia diverso. Questo succede quando si aggiunge il metodo S2N_OVO ad una combinazione nella quale è già stato considerato il metodo degli intervalli di espressione, poiché i due metodi

selezionano un numero molto grande di geni in comune e le combinazioni degli gli altri metodi fanno in modo che le differenze si annullino del tutto.
Il metodo S2N_OVO però è l'unico che ha un'affinità con il metodo degli intervalli di espressione, dal momento che la combinazione di quest'ultimo con gli altri algoritmi abbassa sempre di molto il numero di *feature* in comune.
Possiamo quindi affermare che i geni selezionati più spesso da tutti gli algoritmi sono i 24 elencati in Tabella 7.8.

I_958643	I_1221791	NM_003663.2
I_1109891	I_962355	I_962269
I_931231	I_1100621	I_928537
I_960055	I_961000	I_966484
I_937382	I_1100742	NM_138444.1
I_1000458	I_945374	I_929205
I_1152582	I_960366	I_966324
I_966239	I_958100	NM_003376.3

Tabella 7.8 - Geni selezionati da tutti i metodi di *feature selection*

7.2.4. Albero decisionale utilizzando il metodo degli intervalli di espressione

Come si è visto nel paragrafo precedente il metodo degli intervalli di espressione influisce in modo significativo sulla selezione dei geni maggiormente scelti. Possiamo quindi focalizzare la nostra attenzione sui geni scelti da questo metodo al fine di costruire un albero di decisione che permetta una classificazione dei pazienti in base ad un numero limitato di geni.
Il metodo degli intervalli di espressione è stato derivato empiricamente e funziona solo in un contesto binario. Per questo motivo, se viene applicato ad un problema multiclasse, come quello dell'analisi dei dati della prostata, bisogna scomporre il problema in più problemi binari. Facendo ciò determiniamo i geni che differenziano meglio due classi in base agli intervalli dei livelli di espressione di ciascun gene.

Geni selezionati con metodo IE per le classi 0 e 1					
I_928537					
Geni selezionati con metodo IE per le classi 0 e 2					
I_966239					
Geni selezionati con metodo IE per le classi 1 e 2					
I_958643	I_1221791	NM_003663.2	I_1109891	I_962355	I_962269
I_931231	I_1100621	I_928537	I_960055	I_961000	I_966484
I_937382	I_1100742	NM_138444.1	I_1000458	I_945374	I_929205
I_1152582	I_960366	I_966324	I_966239	I_958100	NM_003376.3

Tabella 7.9 - Geni selezionati da IE e in comune con gli altri metodi di *feature selection*

Avendo una lista di geni per ogni coppia di classi, possiamo determinare se un nuovo campione appartiene ad una classe rispetto ad un'altra solamente se i valori di espressione di alcuni geni ricadono in determinati intervalli.
In Tabella 7.9 sono indicati i geni selezionati per ogni combinazione di accoppiamento delle classi, considerando solamente i geni in comune agli altri metodi.
Utilizzando questi geni possiamo creare dei modelli di classificatori utilizzando Weka, in particolare costruendo alberi decisionali.
Come si è visto nel capitolo relativo al data mining (Capitolo 4) gli alberi di decisione costruiscono un classificatore che ha il vantaggio di essere facilmente interpretabile. Questa loro caratteristica è molto utile per far comprendere la dinamica e le regole di etichettatura dei campioni futuri. Sicuramente risultano meno "potenti" di altri algoritmi di classificazione per quanto riguarda le performance, ma essendo consultabili in modo veloce ed efficace sono molto utili al fine di riassumere le nozioni apprese attraverso analisi più complesse.
Per costruire un albero decisionale semplice, abbiamo scelto, tra quelli disponibili in Weka, l'algoritmo REPTree [27].
Questo metodo costruisce un albero decisionale o di regressione usando il guadagno d'informazione o la varianza e potando l'albero utilizzando il *reduced-error pruning* (con *backfitting*) [28].
Questo algoritmo denominato REP inizialmente determina un set di dati detto *pruning examples*, indicato d'ora in avanti con *S*. A questo punto il metodo lavora in due fasi:

- per prima cosa il set *S* viene classificato utilizzando l'albero costruito inizialmente che deve essere "potato". Dei contatori che tengono traccia del numero di campioni di S di ogni classe che passano attraverso ogni nodo vengono aggiornati simultaneamente.
- nella seconda fase, detta di *pruning bottom-up*, elimina i rami dell'albero che possono essre rimossi senza aumentare l'errore delle rimanenti ipotesi.

L'utilizzo del *backfit* aumenta la probabilità di stima delle foglie dell'albero.
Ciò viene fatto considerando inizialmente un sub-training set, che è una porzione dei dati totali, e facendo dopo la costruzione di un sub-classificatore un secondo *"backfit"*, inserendo il training set originale nell'albero decisionale senza cambiare la relativa struttura.
Eseguendo questo algoritmo il risultato è mostrato in Figura 7.25 mentre le relative statistiche in Tabella 7.10, Tabella 7.11 e Tabella 7.12:

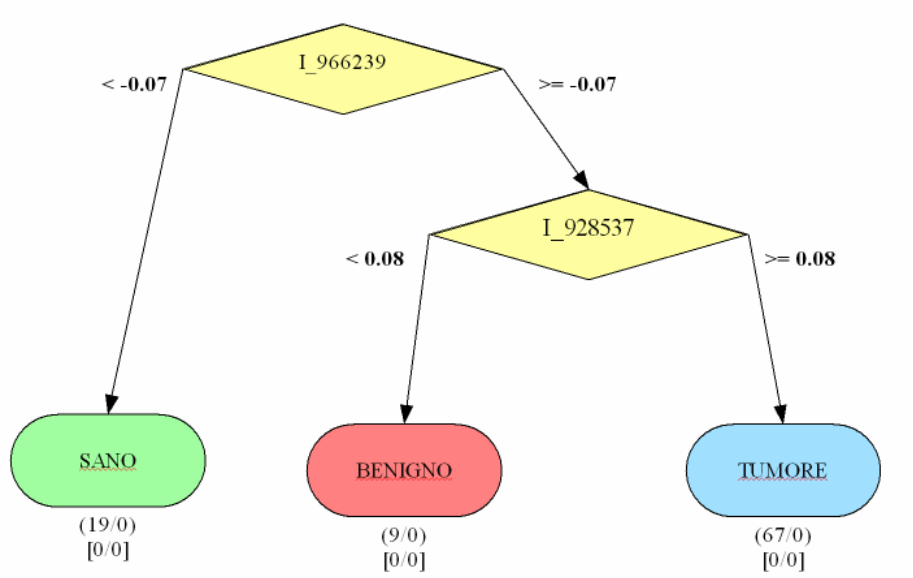

Figura 7.25 - Decision tree. Dimensione dell'albero: 5. Tempo di creazione: 0.03 secondi

Correctly Classified Instances	*91*	*95.7895 %*
Incorrectly Classified Instances	*4*	*4.2105 %*
Kappa statistic	*0.9072*	
Mean absolute error	*0.0281*	
Root mean squared error	*0.1675*	
Relative absolute error	*9.1301 %*	
Root relative squared error	*43.0405 %*	
Total Number of Instances	*95*	

Tabella 7.10 - Sommario della cross-validation

TP Rate	*FP Rate*	*Precision*	*Recall*	*F-Measure*	*Class*
0.97	*0.071*	*0.97*	*0.97*	*0.97*	*Tumour*
0.889	*0.012*	*0.889*	*0.889*	*0.889*	*benigno*
0.947	*0.013*	*0.947*	*0.947*	*0.947*	*Sano*

Tabella 7.11 - Accuratezza dettagliata per classe

a	*b*	*C*	*← classified as*
65	*1*	*1*	*a = tumore*
1	*8*	*0*	*b = benigno*
1	*0*	*18*	*c = sano*

Tabella 7.12 - Matrice di confusione

In Figura 7.26 possiamo visualizzare i due geni scelti dall'albero decisionale su un piano cartesiano. In questo modo si riescono a distinguere i tre cluster che si vanno a formare. Come si può notare ci sono dei punti che sono sul limite tra gli intervalli delle varie classi. Questi punti sono importanti per la classificazione perché determinano dei confini tra le classi, anche se nell'esecuzione della cross-validation portano ad errori di classificazione quando vengono inseriti nel test set e non nel *training set*, abbassando le performance dell'albero decisionale.

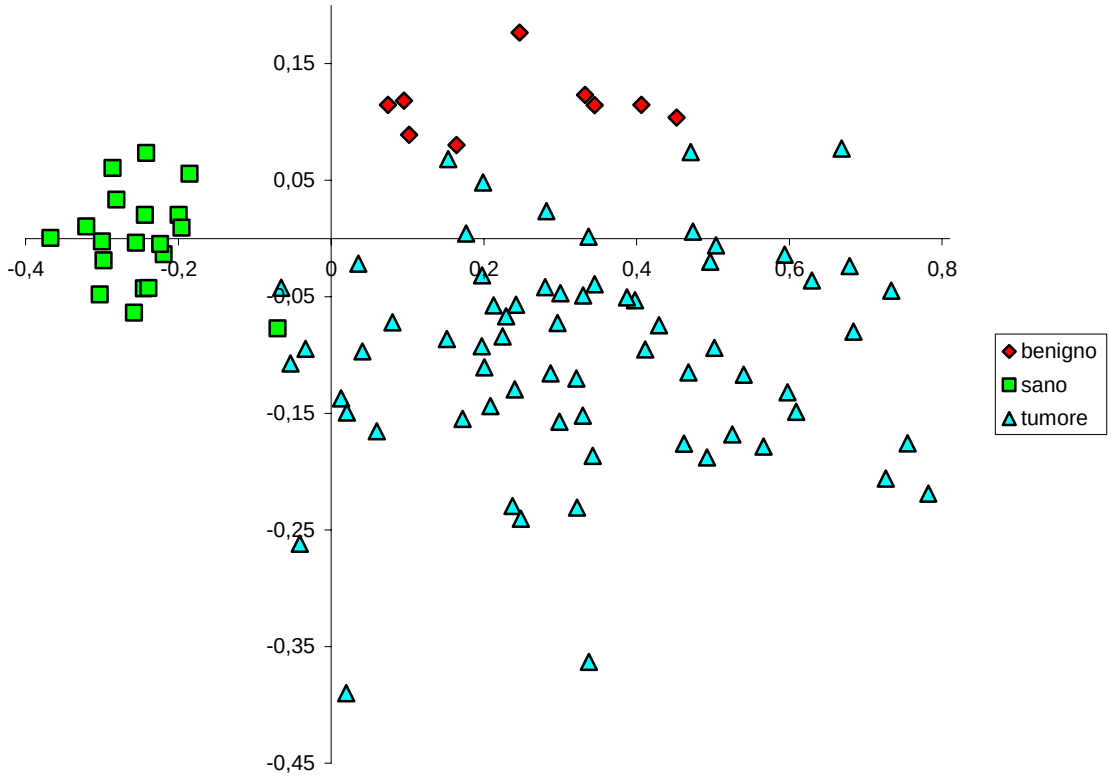

Figura 7.26 – Plot dei campioni secondo i valori dei geni *I_966239* (ascisse) e *I_928537* (ordinate).

Capitolo 8
Conclusioni

In questo capitolo si presentano le conclusioni che si sono ottenute dall'analisi di tutti gli esperimenti fatti.
Dal momento che i dataset utilizzati hanno delle caratteristiche intrinseche differenti, e poiché gli esperimenti svolti sono in parte dissimili proprio a causa delle loro differenze, l'analisi conclusiva viene divisa in base alla provenienza dei dati.
In questo modo si avrà una visione particolareggiata per ogni tipologia di dato. Bisogna ricordare che un'analisi complessiva del problema non si può avere poiché i dati e i geni che si analizzano variano a seconda del tessuto analizzato e di tutte le procedure che vengono effettuate nelle analisi preliminari.

8.1. Dati del colon

L'analisi dimostra che è possibile individuare un numero molto elevato di geni (3269), ciascuno dei quali è in grado di distinguere le due categorie di pazienti (sani e malati). L'applicazione di metodi di clustering o classificazione ai dati forniti permette di identificare perfettamente le due classi di pazienti, ma sposta il problema alla selezione dei geni più promettenti per la classificazione di futuri pazienti.
A questo proposito sono stati confrontati i geni individuati da PAMR con quelli ottenuti applicando vari criteri di ordinamento ai 3269 geni candidati. Mentre alcuni criteri di ordinamento confermano molti geni individuati anche da PAMR, altri presentano risultati discordanti.
Considerazioni analitiche relative all'importanza dei due parametri di ordinamento previsti (ampiezza e distanza degli intervalli di espressione) hanno portato alla scelta dei due criteri più adeguati tra quelli confrontati. Tali criteri hanno permesso di indicare i 128 geni più promettenti per la classificazione delle due categorie di pazienti e ulteriori 58 che meritano particolare attenzione.

Successive analisi richiedono di valutare il significato biologico dei risultati presentati e di approfondire i seguenti aspetti:

- lo studio di filtri differenti sui dati originali (22175 geni) in modo da individuare un sottoinsieme da analizzare diverso rispetto a quello attualmente considerato (8466 geni);
- l'applicazione di test statistici, che potrebbe essere inefficace a causa della scarsa numerosità dei pazienti a disposizione;
- l'uso di informazioni di contesto, quali la funzionalità dei geni (se nota) per migliorare la selezione di quelli più adatti a distinguere le due classi di pazienti.

8.2. Dati della prostata

I dati della prostata a differenza di quelli precedenti sono maggiormente difficili da classificare. La principale causa di questa difficoltà è la nuova dimensione del problema. Infatti non si tratta più di distinguere in due classi ma in tre, facendo sì che il problema diventi multiclasse.
Per questo motivo il primo obiettivo è ottenere delle prestazioni ottimali con alcuni algoritmi di classificazione. Analizzando vari algoritmi si sono scelti i Support Vector Machine che, seppur nati su problemi binari, sono stati modificati per analisi multiclasse ottenendo prestazioni ben superiori ad altri algoritmi. Nonostante siano adatti all'analisi di dati derivanti da microarray le prestazioni non risultano sempre ottimali.
Grande importanza ricopre la scelta del kernel utilizzato dal classicatore per determinare la distribuzione dei dati. Un kernel gaussiano rispetto ad uno polinomiale ottiene in linea generale prestazioni migliori, ipotizzando distribuzioni gaussiane dei dati e diminuendo in questo modo la probabilità di misclassifcazioni.
Per migliorare le prestazioni dei vari classificatori non è sufficiente scegliere la tipologia di kernel migliore, ma è necessario ridurre la complessità del problema utilizzando la feature selection.
Infatti con la feature selection si riduce il numero delle dimensioni, ossia i geni, che l'algoritmo di classificazione deve analizzare per determinare la classe di appartenenza del campione.
Non sempre la feature selection porta benefici. Infatti se viene scelto di eliminare un gene che facilita la distinzione tra le classi si possono ottenere prestazioni non ottimali. Nonostante la criticità della selezione, le prestazioni ottengono sempre un miglioramento rispetto alla costruzione di un classificatore considerando tutti i geni a disposizione.
Oltre agli algoritmi maggiormente utilizzati per la selezione si è analizzato l'impiego del metodo sperimentato con successo sui dati del colon. Questo metodo basato sugli intervalli di espressione è stato modificato per renderlo applicabile ad un problema multiclasse. Le

prestazioni sono buone anche con questa tipologia di dati essendoci ben 231 geni che distinguono le tre classi.
Dal momento che il numero di geni influisce sia a livello di prestazioni di classificazioni che di complessità computazionali, si sono trovati i geni che vengono selezionati da tutti gli algoritmi. Questi 24 geni in comune a tutti gli algoritmi di selezione vengono considerati i migliori poiché hanno alto contenuto d'informazione ed inoltre, derivando dalla selezione del metodo degli intervalli di espressione (IE), distinguono bene le classi a gruppi di due.
Utilizzando questi geni si è potuto anche costruire dei classificatori facilmente interpretabili come gli alberi decisionali, ottenendo prestazioni molto soddisfacenti.
Analisi future richiederanno di valutare e approfondire i seguenti aspetti:

- l'utilizzo di informazioni di contesto, quali la funzionalità dei geni per migliorare la selezione;
- l'analisi delle informazioni sui pazienti, quali l'età e il sesso;
- l'analisi dell'andamento temporale dei valori di espressione dei geni.

Bibliografia

[1] G. Valle, M. Helemr Citterich, M. Attimonelli, G. Pesole, *"Introduzione alla bioinformatica"*, Zanichelli, 2003

[2] C.Priami, *"Informatica e biologia dei sistemi"*, Mondo digitale, 2004

[3] T. K. Attwood, D. J. Parry-Smith's, *"Introduction to Bioinformatics"*, Prentice-Hall, 1999

[4] M. S. Boguski, *"Trends Guide to Bioinformatics"*, Trends Supplement, 1998

[5] G. Balsamo, M. Erbace, I. Forte, G. Scocozza, G. Vitale, *"Elementi di Biologia Molecolare e Bioinformatica"* , 2004

[6] F. S. Collins, E. D. Green, A. E. Guttmacher, M. S. Guyer, *"A vision for the future of genomics research"*, Nature, 2003

[7] J. Chen, C. Chen, *"Microarray Gene Expression"*, 2003

[8] Regev, E. Shapiro, *"Cells computation"*, Nature, 2002

[9] K. Gordon, T. Speed, *"Normalization of cDNA Microarray Data"*, Aprile 2003

[10]D Ghosh, A Chinnaiyan, *"Mixture modelling of gene expression data from microarray experiments"*, 2001

[11]Y. Moreau, F. De Smet, G. Thijs, K. Marchal, B. De Moor, *"Functional Bioinformatic of Microarray Data: from expression to regulation"*, 2002

[12]Hovatta, K. Kimppa, A. Lehmussola, T. Pasanen, et al., *"DNA Microarray Data Analysis"*, CSC, 2005

[13]Han, M. Kamber, *"Data Mining: Concepts and Technique"*, 2004

[14]T. Speed, *"Statistical analysis of gene expression microarray data"*, 2003

[15]R. Tibshirani, T. Hastie, B. Narasimhan, G. Chu, *"Diagnosis of multiple cancer types by shrunken centroids of gene expression"*, 2002

[16]E. Acuña, L. Pericchi, "*Non-Parametric and Bayesian statistical methods for knowledge discovery in microarray data"*, 2003

[17]Statnikov, C. F. Aliferis, I. Tsamardinos, D. Hardin, S. Levy, "*A comprehensive evaluation of multicategory classification methods for microarray gene expression cancer diagnosis",* 2005

[18]Statnikov, C. F. Aliferis, I. Tsamardinos, D. Hardin, S. Levy, "*Appendix",* 2005

[19]D. Anguita, S. Ridella, D. Sterpi, *"A New Method for Multiclass Support Vector Machine"*, Proc. Of the IEEE Int. Joint Conf. on Neural Networks, Luglio 2004

[20]D. Anguita, A. Boni, S. Ridella, F. Rivieccio, D. Sterpi, *"Theorical and Practical Model Selection Methods for Support Vector Classifier"*, 2004

[21]D. Anguita, S. Ridella, F. Rivieccio, D. Sterpi, *"mcSVM v.1.0.1"*, Febbraio 2005

[22]T.R. Golub, *"Molecular Classification of Cancer: Class Discovery and Class Prediction by Gene Expression Monitoring"*, 1999

[23]S. Dudoit , *"Comparison of Discrimination Methods for the Classification of Tumors Using Gene Expression Data"*, 2000

[24]M. Sayan, *"Classifying Microarray Data Using Support Vector Machines"*, 2002

[25]E. Frank, M. Hall, G Holmes, H. Witten, *"Data mining in bioinformatics using Weka"*, Aprile 2004

[26]V. Sidhwani, P.Bhattacharya, S. Rakshit, *"Information Theoric Feature Crediting in Multiclass Support Vector Machines"*, 2002

[27]M. Kaariainen, T. Malinen, "*Selective Rademacher Penalization and Reduced Error Pruning of Decision Trees*", 2004

[28]E. Bauer, R. Kohavi, "*An empirical comparison of voting classification algorithms: Baggins, Boosting and Variants*", 1999

[29]http://www.is.titech.ac.jp/~shimo/prog/pvclust/

[30]http://www.wikipedia.it

Appendice A – Elenchi dei geni selezionati

A.1. Metodo BW

I_958100	I_965578	I_932430	I_930056	I_966239	I_966324
I_931429	I_960055	I_960101	I_928503	I_944210	I_929528
I_1002307	I_929030	I_931619	I_959406	I_1002262	I_1110373
I_960642	I_960094	I_932443	NM_144705.1	I_945374	I_960898
I_942164	I_928298	NM_145200.1	I_962071	I_930025	NM_144990.1
I_966387	NM_024725.2	I_963659	I_958095	NM_003376.3	I_1000458
I_959074	I_960522	I_1152303	I_957136	I_964215	I_935779
I_956858	I_930112	I_1152582	I_964459	NM_145055.1	NM_003663.2
I_2030617.FL1	I_931207	I_932622	I_937382	I_957930	I_959488
NM_144994.4	I_930157	I_960936	I_962292	I_929883	I_962243
I_960366	I_966228	I_962093	NM_002823.2	I_928893	I_932771
NM_000522.2	I_964029	I_930000	I_1152193	I_962637	I_1151846
NM_144775.1	I_956836	I_963513	I_957547	I_1000526	I_928537
I_1221791	I_964221	I_959185	I_1152081	I_930340	I_960821
I_964353	I_1100080	I_929720	I_959122	I_930161	I_962032
I_964069	I_960650	I_944037	I_960066	I_958992	I_931231
I_928805	I_965539	I_1152230	I_961155	I_962335	I_958643
I_1100816	I_960842	I_959866	I_953745	I_965802	I_1109891
I_966319	I_961585	NM_002284.2	I_932479	NM_153754.1	I_960757
I_1221898	I_932569	NM_002883.1	I_930614	I_1000320	I_966225
I_931152	I_962211	I_1100729	I_958779	I_962355	I_959683
I_960171	NM_004474.1	I_936330	I_966386	I_961850	NM_032527.1
I_1109535	I_957187	I_966532	NM_145208.1	NM_005257.1	I_943964
NM_138444.1	I_1110108	NM_000827.2	I_960743	I_957728	I_960249
I_962339	I_962657	NM_033397.1	I_965876	I_935811	I_963286.FL1
I_934479	I_1002505	NM_153031.1	I_936441	I_1152553	I_938856
I_930066	I_961838	I_928768	I_962269	I_931649	I_928291
I_931348	I_928270	I_935775	I_965464	NM_152772.1	I_931585
I_1109180	I_966602	NM_144606.2	I_947021	I_964454	I_959832
I_943047	NM_152342.1	NM_152353.1	I_932620	I_961778	I_965546
I_958526	I_1100061	I_956932	I_1109818	I_930406	I_966597

NM_020948.1	I_959119	I_963028	I_2029694.FL1	I_957177	NM_174932.1
I_958319	I_957927	I_1002299	I_966315	I_1100124	I_961545
I_962210	I_1109097	I_957153	I_931322	I_932553	I_944055
I_932091	I_931326	I_1100833	I_934078	I_1002215	NM_007157.1
I_947012	NM_006928.2	I_1000711	I_959360	I_966484	NM_012206.1
I_1109310	I_1110087	I_1110213	I_935246	I_959055	I_928649
I_929272	I_949415	I_1152238	I_931862	I_963173	I_963227
I_929176	I_928450	I_1002415	I_932566	I_942166	NM_153039.1
I_931087	I_928409	I_1100670	I_1110381	I_957004	I_966353
NM_178514.1	I_964267	I_930284	NM_020960.1	NM_006266.1	I_961000
NM_173473.1	I_930585	NM_145263.1	I_929483	I_958911	I_2019746
I_930951	I_944014	I_1952335	I_960830	I_1151977	I_962213
I_959245	I_960784	I_958415	I_957473	I_936331	I_1000653
I_1109748	NM_021038.1	NM_007147.1	I_1000502	I_1152579	I_931244
I_966107	I_928282	I_932193	I_959855	I_958514	NM_003569.1
I_1201912	I_963959	I_962212	I_960019	I_1984412	I_929687
I_928943	I_963277	I_957332	I_959675	I_966247	NM_006569.1
I_930686	I_957848	I_1100580	I_1000263	I_1100433	I_1100744
I_1000211	I_931754	I_962642	I_964009	I_964501	NM_133328.1
I_957033	I_929205	I_966112	I_957501	I_1002167	NM_153256.1
I_932189	I_1100127	I_959206	I_1000163	I_963988	I_1110300
I_962840	I_944052	I_938789	I_1877956.FL1	I_1221950	I_963775
I_930907	I_1100542	I_1100736	I_958458	I_958048	I_963097
I_1109854	I_928308	I_958626	I_930447	I_963273	I_934486
I_928316	I_943813	I_930768	I_966288	I_2031437.FL1	I_943800
I_931236	I_933139	I_930982	I_1100742	I_960199	I_966203
I_1109284	I_928967	I_961100	I_1109570	I_932392	I_961488
I_1002378	I_962265	I_961092	I_1110192	I_931929	I_930813
I_961599	I_1109966	I_1000080	NM_031214.1	I_1100255	I_936437
I_1152241	I_932098	I_930794	I_930930	I_928603	NM_005180.4
I_1109743	I_959510	I_964774	I_928797	I_957516	I_930219
I_936455	I_959816	I_959553	I_931191	I_1000339	NM_014494.1
I_945879	I_961809	I_957145	I_956978	NM_153217.1	NM_000848.2
I_960582	NM_004050.2	I_1002181	I_932755	I_929218	I_1152309
I_956845	I_963373	I_962114	I_930258	I_1152031	I_963141
NM_017420.1	I_934738	I_932250	I_965210	I_962546	I_928362
I_959708	I_1152571	I_960851	I_1002064	NM_004449.2	I_934124
I_1002450	NM_152633.1	I_930689	I_935638	NM_178858.1	NM_022457.4
I_932022	I_928697	I_959904	I_1109336	I_1109642	NM_016073.2
I_1100621	I_1100001	I_964974	I_959697	I_1000315	I_956979
I_960396	I_932330	NM_020922.1	I_1109614	I_1109272	I_929899
I_1000639	I_930081	I_1100172	I_964376	I_957508	I_931196

I_958607	I_1100023	I_958709	I_960569	NM_032318.1	I_929375
I_931854	I_932113	I_934425	I_935625	I_1221771	I_943467
I_1221813	I_960854	I_965857	I_950146	I_932720	I_1221974
I_962397	I_956909	I_958091	I_962019	I_963452	I_936889
I_931125	I_1201895	I_1152502	I_961626	I_965259	I_930877
I_929323	NM_005388.2	I_1151896	I_1872538	I_1110268	I_957753
I_1000517	I_962696	I_1109183	I_958312	I_960333	NM_020439.1
I_959729	I_959163	NM_178177.1	I_929342	I_959566	I_1002385
NM_004588.2	I_932542	I_1152585	I_961398	I_929435	I_929211
I_934250	I_963960	I_930261	NM_145050.1	I_959913	I_1002405
I_944100	I_962473				

A.2. Metodo KW

I_932430	I_931429	I_928503	I_959406	I_932443	I_1152303
I_960101	I_966324	I_930056	I_960055	I_965578	I_958100
I_966239	I_928298	NM_024725.2	I_2029694.FL1	NM_000522.2	I_964221
I_966319	I_930025	I_2030617.FL1	I_966386	I_960898	I_960094
I_932622	I_1152193	I_932771	I_964459	I_957136	I_929883
I_928943	I_966387	I_928805	I_960366	I_928537	I_945374
I_1002415	I_966315	I_1109818	I_944210	I_1100061	I_929483
NM_001454.1	I_1002307	NM_144606.2	I_962243	I_934078	NM_014494.1
NM_006266.1	I_1000458	I_1100816	I_961027	NM_144994.4	I_931649
I_1952335	I_960743	I_962657	I_930161	I_928409	I_933139
I_964009	I_931658	I_929176	I_929648	I_931207	I_963028
NM_173473.1	I_962637	NM_033397.1	I_1110373	I_963882	I_958643
I_958319	I_966228	I_1109891	I_935779	I_930585	I_943467
I_1110300	I_935246	I_1110213	I_960642	I_957779	I_957516
I_1000263	I_958626	I_931974	I_959185	I_931585	I_957930
I_1152230	I_960757	I_965876	I_1110187	I_1109791	NM_138444.1
I_957848	I_966484	I_1100833	I_1152553	I_932620	I_961092
I_928291	I_964029	I_966225	I_1000163	I_961599	NM_153039.1
I_959122	I_1002215	I_1110381	I_957728	I_928697	NM_145055.1
I_1151977	I_959697	NM_145263.1	I_1000320	I_959219	I_930406
I_960784	I_963227	I_930794	I_962785	I_965539	I_931089
I_961809	NM_003663.2	I_961626	I_1110140	I_962019	I_1221898
I_960886	I_963273	I_957068	I_937382	I_960830	I_956836
NM_016617.1	I_929323	I_1152582	I_1000732	I_958214	I_936437
I_1109748	I_959206	I_957753	I_928308	I_957641	NM_178858.1
I_965210	I_943726	NM_003376.3	I_961398	I_964069	I_930073

I_957004	I_932553	I_963543	I_958345	I_962252	I_930003
I_959553	I_960821	I_928768	I_1109652	I_1109112	I_1000339
I_932030	I_944011	I_961879	I_957547	I_960851	NM_152903.1
I_962479	I_1221913	I_1100742	I_959598	I_964501	I_1000653
I_940093	NM_002856.1	I_947012	I_966345	I_1002167	I_928282
I_928797	I_962135	I_931619	I_934123	I_932593	I_959510
I_956978	I_959866	I_1151896	NM_133328.1	I_943964	I_958573
I_936493	I_957508	I_962506	NM_145200.1	I_966633	I_959053
I_944052	I_2019746	I_947021	NM_007147.1	I_1152571	I_929272
I_958992	I_932316	I_1100736	I_962276	I_957213	I_966288
I_1109553	NM_006569.1	I_1109814	I_1000080	I_1109272	I_962397
I_1110236	NM_021925.1	I_964376	NM_000848.2	I_943018	I_949415
I_962446	I_1100451	I_961867	I_1100107	I_963960	I_1000202
I_943704	I_958317	I_938795	I_957534	I_1002064	I_957473
I_950146	NM_152342.1	I_960584	NM_019074.1	NM_002883.1	I_959576
I_958787	I_1002439	I_1221950	I_1221791	I_944055	I_1152271
I_931244	I_966738	I_962803	I_958617	I_930143	I_931314
I_964639	I_961545	I_966117	I_962269	I_932720	I_1884945
I_1109284	NM_005715.1	I_957116	I_964267	I_1221981	NM_006769.2
NM_006391.1	I_930727	I_1002181	I_944025	I_962141	I_960753
I_944014	I_929698	I_957568	I_1109642	I_1100080	I_931238
NM_002605.1	I_960171	I_1100452	I_959904	I_931425	I_962634
I_931241	I_1100001	I_1110268	NM_020960.1	I_959885	I_1100542
I_934124	I_957823	I_961322	NM_173354.1	I_932483	I_931191
I_929686	I_963066	I_932490	I_1002302	I_957663	I_930112
I_1100255	I_930066	I_964282	I_932022	I_934738	I_958754
I_932448	I_963373	I_962292	I_928967	I_961601	I_931854
I_931486	I_935638	I_1000526	I_928960	I_966472	I_962597
I_1110192	NM_178840.1	I_929831	I_963820	I_931169	I_1000419
NM_178514.1	I_965031	I_930686	NM_032527.1	I_957395	I_1000685
I_957326	NM_012482.1	I_1221845	I_961000	I_929565	I_1100729
NM_133466.1	I_963716	NM_002734.1	I_1110306	I_931881	I_937752
I_960650	I_929375	I_937311	NM_173826.1	I_962355	NM_032239.2
I_932755	I_1100580	NM_002979.2	I_930474	I_963164	NM_032318.1
I_957104	I_935625	I_931792	I_962152	I_1201912	I_958458
I_1100773	I_963379	I_931007	I_930219	I_1110242	I_963361
I_930447	I_958526	I_962840	I_962064	I_966918	I_929205
NM_015087.2	I_959708	I_929229	I_932479	I_945181	NM_024959.1
I_959307	I_1000502	I_959816	I_958403	NM_145171.1	I_961838
I_935251	I_961155	NM_006910.1	I_1002385	I_957775	I_960897
I_966088	I_966085	I_931862	I_932356	I_956979	I_929817
I_959276	NM_138462.1	I_1100103	I_939371	I_930081	I_961659
I_960249	I_930636	NM_018459.1	I_930118	I_928936	I_928365
NM_004050.2	I_938452	I_962462	I_931231	I_960638	I_959725
I_961585	I_961462	NM_145039.1	I_1152023	I_964546	NM_005257.1

I_932569	NM_144775.1	I_931925	I_962885	I_929687	I_966353
I_958639	NM_144654.1	I_1000207	I_931859	I_1109275	I_1100853
I_949040	I_931395	I_1152238	I_960522	I_963959	I_964774
NM_021705.1	I_930005	I_966203	I_930998	NM_032606.2	I_1109517
I_965259	I_957941	NM_138959.1	I_1109570	I_933015	NM_178350.1
I_1100777	I_928517	I_957451	I_963141	I_957092	I_960987
I_1100456	I_928743	I_1109183	I_960630	I_961550	I_965802
I_928649	I_959217	I_966602	I_928362	I_1109854	NM_022361.2
I_957415	I_962642	I_930754	I_934742	I_944113	I_1000657
I_935775	I_929129	I_959691	I_962254	I_961287	I_1110310
I_929083	I_929665	I_961345	I_931774	I_1109180	I_1152579
I_966582	I_1100621	I_962093	I_932612	NM_005180.4	I_960446
I_3292030.FL1	I_1152497				

A.3. Metodo S2N_OVO

I_929030	I_958100	I_966324	I_957930	I_930754	I_930614
I_935811	I_960936	NM_144705.1	I_966239	I_953745	I_1152582
I_958643	I_1151846	I_1000458	I_956858	I_959488	I_1002262
I_965578	I_931152	I_930686	I_928537	I_931231	I_1152230
I_958091	I_963294	NM_002823.2	I_959683	I_930056	I_932548
I_943813	I_1221791	I_957374	I_964459	I_1152309	I_932430
NM_022457.4	I_958095	I_960055	I_1110213	I_930454	I_962269
NM_003663.2	I_943047	I_929687	I_1100742	I_932443	I_1100729
I_960366	I_936986	NM_173510.1	I_962339	I_932330	I_957473
NM_004474.1	I_960171	I_963106	I_929375	I_932420	I_1100522
I_928450	I_1100797	I_961000	I_966907	I_958733	I_962803
I_960249	I_1109891	I_936330	I_929205	I_932622	I_957581
I_930000	I_935225	I_1109632	I_963939	I_1152007	NM_144990.1
NM_021038.1	I_931512	I_961488	I_959276	I_957870	I_963513
NM_153754.1	I_931429	I_961509	NM_138444.1	I_962355	I_959691
I_962855	I_960842	I_962114	I_958470	I_928362	I_928797
I_961850	I_935775	I_1000653	NM_173473.1	I_1221913	NM_145200.1
NM_152353.1	I_942693	I_963659	NM_152407.1	NM_144994.4	I_957775
I_958744	I_932037	I_945374	I_963173	I_1100863	I_961367
I_960066	NM_020948.1	I_1000517	I_929726	I_960748	I_960541
I_960569	I_930849	NM_005388.2	NM_006928.2	I_931929	I_943800
I_931244	I_930081	I_1002299	I_931125	NM_000522.2	I_944210
I_959689	I_1000320	I_956822	I_940093	I_958992	I_1002450
I_1100439	NM_000827.2	I_1109642	I_928503	I_957187	I_932189

I_934078	I_931091	I_1000482	I_1110071	I_942166	I_962749
NM_006082.1	I_1002167	I_964406	I_959245	I_933554	I_959074
I_937382	I_960962	I_939856	I_1152031	I_963943	I_956836
I_1100713	I_932541	NM_003584.1	I_962523	I_932542	I_959885
NM_000704.1	NM_005257.1	I_963988	I_939261	I_960717	I_931387
I_966484	I_928671	NM_153261.1	I_963097	I_958312	I_930357
I_962839	I_960317	I_929080	I_929435	I_957688	NM_020126.3
I_1109743	I_1109535	I_1109305	I_1151977	I_958345	I_1000502
NM_024676.2	I_1109310	I_964105	I_930118	I_962354	I_1000263
I_1109414	I_958048	NM_145263.1	NM_032046.1	I_1100080	I_1984412
NM_006166.2	I_966117	I_1100621	I_930161	NM_005781.2	I_929412
I_1100542	I_1100124	I_929883	I_963911	I_960582	I_1109776
NM_174932.1	I_1109103	I_958911	I_930447	NM_013305.2	NM_005180.4
I_938856	NM_018704.1	I_1002064	I_1214078	I_959455	NM_007131.1
I_960396	I_958319	I_1877956.FL1	I_929746	I_957501	I_962696
I_958272	NM_004158.2	NM_178429.1	I_944073	I_931645	I_935638
I_932331	I_932098	I_963615	I_1152603	I_930025	I_1152081
I_962642	I_959735	NM_003610.2	I_961413	I_966532	I_958514
I_934732	I_1002405	NM_178350.1	I_961155	I_932344	I_962444
NM_012206.1	NM_016073.2	I_1152579	I_960014	I_928282	I_928805
I_964501	NM_003569.1	I_1109570	I_932496	I_965259	I_959360
NM_020960.1	I_956845	I_935074	I_930763	I_957498	I_931419
I_931619	I_959979	I_958136	I_960743	NM_030790.2	I_957177
I_930930	I_935670	I_961268	I_966504	I_930907	I_1201837
I_1152454	I_966203	NM_174905.1	I_932517	I_1000413	I_959214
I_932193	I_961783	I_959055	NM_000895.1	I_1152443	I_928976
I_957843	I_1110279	I_957848	I_1100433	I_963292	I_957044
I_961585	I_965763	I_962211	I_930340	NM_024108.1	I_959708
I_966142	I_935851	I_928270	I_931754	I_1002378	I_957085
I_931322	I_932638	I_928603	I_932545	I_960919	I_961658
NM_003376.3	I_932569	I_957516	I_1152585	I_932566	I_958187
I_957153	I_944100	I_1100712	I_944055	I_961784	I_1152060
I_1151996	I_947994	I_957451	I_929193	I_959866	I_931854
I_964989	I_929720	I_1110187	I_962032	I_929067	I_929605
I_943953	I_1000345	I_957818	I_929218	I_953749	I_959119
I_932113	I_1100085	I_928298	NM_144775.1	NM_152539.1	I_957941
NM_145253.1	I_944099	I_929370	NM_022361.2	I_1151828	I_964353
I_962390	I_1000063	I_928629	I_962271	I_950764	I_1152303
I_959163	I_961838	I_931087	I_930157	I_960418	I_963668
I_961293	I_932483	I_960101	I_944025	I_932620	I_960019
I_959904	I_957508	I_929608	I_1109180	I_929011	I_931532
I_930813	I_960796	I_928748	I_928884	I_929009	I_929462
I_939867	I_958959	I_959562	NM_020967.1	I_957340	I_929686
I_965696	I_964029	I_934486	I_934250	I_959720	I_958458

I_957823	I_932467	I_1100816	I_959382	I_964267	I_930768
I_1109359	I_960333	I_965945	I_961309	I_962915	I_930003
I_1109112	I_928649	I_928321	I_1152238	I_931326	I_1152024
I_932208	I_1100670	I_960522	I_959913	NM_002979.2	I_959099
I_966597	I_928565	NM_178168.1	I_1109748	I_1212867	I_931006
I_958779	I_1221789	I_957624	I_958403	I_959406	I_966235
NM_152633.1	I_929768	NM_006206.2	I_936493	I_932225	I_1221974
I_930258	I_929389	I_1201928	I_1100834	I_964774	I_963758
I_1002454	I_958415	I_931236	I_962421	I_931342	I_930689
NM_147128.1	I_961645	I_957033	I_928646	I_956909	I_932446
I_963348	I_965645	I_964376	I_1997873	I_931634	I_957326
I_1002341	I_1002302	I_932509	I_957598	I_1002181	I_958489
I_1100853	I_1152480	I_930982	I_956978	I_933111	I_966810
I_960094	I_958641	I_1109730	I_965600	I_957732	I_1000526
I_938795	I_942164	I_930469	I_1872538	I_958321	NM_145050.1
I_1000552	I_1110376				

A.4. Metodo S2N_OVR

I_932430	I_928503	I_958100	I_965578	I_931429	I_930056
I_1000458	I_966324	I_966239	I_932443	I_932622	I_928537
I_959406	NM_024725.2	I_1152303	I_957930	I_2030617.FL1	I_964459
I_960101	I_928805	I_958643	I_1152230	I_929030	I_930025
I_1100061	NM_144705.1	I_934078	I_1152582	NM_000522.2	I_1110213
NM_144994.4	I_957136	I_960898	I_958319	I_964009	I_966386
I_957516	I_930161	I_929176	NM_003663.2	I_964221	I_1000263
I_1100816	I_961027	I_930686	I_957568	I_931152	I_966315
I_960743	NM_173473.1	I_1000320	I_1002262	I_960055	I_962243
I_1109748	I_931658	I_928291	NM_145200.1	I_963028	I_1100729
I_1002415	I_1100080	I_959219	I_931649	I_928409	I_1110187
I_960171	I_959185	I_956836	I_966319	I_931231	I_960094
I_1221791	I_1151977	I_959885	NM_145263.1	I_929687	I_930081
I_1221913	I_931974	I_953745	I_966117	I_1002181	I_959683
I_931244	I_1100742	I_1109818	I_929205	NM_020960.1	I_1100833
I_1151846	I_957473	I_932771	I_966228	I_929375	I_935779
I_962637	NM_033397.1	I_958992	I_940093	I_1109112	I_928298
I_959206	I_1100736	I_1152553	I_936437	I_960249	I_961599
I_963882	I_928697	I_930073	I_1109284	I_1152193	I_960753
I_956858	I_944210	I_944025	I_935775	I_943467	I_961879
I_961092	I_961000	I_930614	I_966387	I_1100542	NM_178858.1

I_944052	I_958626	I_935811	I_957374	I_932483	I_958214
I_959866	I_962642	I_957823	I_945374	I_960936	I_964267
I_930585	NM_006569.1	I_962785	I_957508	NM_153039.1	I_962355
I_957068	I_962657	I_959245	I_958470	I_960851	I_959488
I_944055	I_1002215	I_966633	I_928362	NM_021038.1	NM_032239.2
I_928282	I_931089	I_932330	I_961550	I_961398	I_959697
I_929323	I_958095	I_964501	I_928768	I_962019	I_960784
NM_006391.1	NM_006266.1	I_935638	I_961488	I_2019746	I_966738
I_959708	I_929883	NM_020948.1	I_931619	I_1110236	I_938795
I_1110381	I_962339	I_943704	I_964376	I_966085	I_932553
I_1110300	I_932569	I_932448	I_936493	I_931169	I_961155
I_958787	I_963273	NM_144606.2	I_1152579	I_966345	NM_016617.1
I_963294	I_963959	I_957941	I_1152454	I_1000526	I_958403
I_1000163	I_959691	I_929483	I_943813	NM_007147.1	NM_178350.1
I_929686	I_962269	I_959598	NM_002823.2	NM_022361.2	I_2029694.FL1
NM_005180.4	I_959904	I_1000080	NM_004474.1	I_949415	I_966225
I_1110268	I_960366	I_945181	I_960886	I_1002307	I_966203
I_965600	I_1152271	I_959276	I_961818	NM_003569.1	I_1100085
I_950146	I_958458	I_1000339	I_959510	I_1100853	I_936330
I_1152007	I_1000419	NM_005715.1	I_961809	I_1002299	I_960757
NM_003376.3	I_957753	I_964774	I_1109791	I_961585	NM_173510.1
I_962803	I_963227	I_1000502	NM_005781.2	I_930447	I_1002302
I_1152571	I_1110192	NM_004050.2	I_1100107	I_931425	I_944014
I_932566	I_965259	I_957870	I_930727	I_1110242	NM_024676.2
NM_152903.1	NM_020127.1	I_933015	I_958317	NM_153261.1	I_1152564
NM_005388.2	NM_006928.2	NM_002856.1	I_956978	I_957092	I_1002154
NM_153754.1	I_960569	NM_022457.4	I_1151896	I_957501	I_962141
NM_138462.1	I_930000	I_962064	NM_002734.1	NM_006769.2	NM_002979.2
I_957534	I_931854	I_931314	I_930794	NM_014494.1	I_965210
I_963361	I_963939	I_960650	I_937382	I_1152309	I_943047
NM_144775.1	I_1100773	I_958091	I_964105	I_928450	NM_178514.1
I_1212867	NM_173826.1	I_1100621	I_959725	NM_144990.1	I_960522
I_1109891	I_929507	NM_000827.2	I_959119	I_958562	I_932548
I_932225	I_1110140	I_928270	NM_144654.1	I_963513	I_932022
I_966786	I_959074	NM_178429.1	I_957728	I_1109652	I_932755
I_962354	I_930191	I_928365	I_963960	I_962506	I_960642
NM_006166.2	I_929915	I_957340	I_961626	I_1002450	I_931326
I_930930	I_960013	I_939371	I_1152238	I_959713	I_957213
I_937311	I_1221950	I_1109553	I_929648	NM_144588.2	I_961838
I_962462	I_960638	I_1000202	I_943800	I_962473	I_956822
NM_000848.2	I_960066	NM_006136.1	I_928510	I_960317	I_932490
I_928649	I_956979	NM_138444.1	I_1109272	I_943018	I_964974
I_963066	I_929609	NM_024108.1	I_935832	I_1000691	I_1000732
I_959217	I_930219	I_959374	I_1151996	NM_007157.1	I_962046

I_931125	I_932113	NM_005205.2	NM_133466.1	NM_014033.1	I_966897
NM_002605.1	I_958187	I_928671	I_958911	I_931421	NM_080706.1
I_931532	I_928917	I_933139	I_957779	NM_174932.1	I_960584
I_1109180	I_948757	I_929462	I_963348	I_958639	I_931007
I_966918	I_959689	I_949040	I_958351	I_962252	I_928628
I_966532	I_1110310	I_962135	I_962032	I_932331	I_963173
I_936884	I_957326	NM_145039.1	I_932123	I_928336	I_1110188
I_930027	I_932030	I_1109854	I_1221966	I_931645	I_966088
I_958733	I_1100712	I_963668	I_958526	I_931859	I_928265
I_934738	I_931754	NM_018459.1	I_960330	I_1109672	I_931419
I_937059	I_1952335	I_964546	I_932037	I_963758	I_928976
I_959053	I_961532	I_963659	I_931207	I_931236	I_962093
I_960842	NM_152407.1	I_1100762	I_962656	I_964042	I_1109570
I_961069	I_1002329	I_1109632	I_961462	I_931925	I_957415
I_962885	I_957455	I_966484	I_1100452	I_959553	I_960328
I_934479	NM_152353.1	I_959214	I_961104	I_1109462	I_929698
I_965031	I_1109785	I_929303	I_931792	NM_030790.2	I_930785
I_957505	I_957673	I_932189	I_934486	I_929083	I_932098
I_957395	I_928509				

A.5. Metodo IE

Geni selezionati con metodo IE per le classi 0 e 1					
I_1109632	NM_144705.1	I_965600	I_1002262	I_928537	I_1151846
Geni selezionati con metodo IE per le classi 0 e 2					
I_966239					
Geni selezionati con metodo IE per le classi 1 e 2					
I_931524	I_958643	I_958954	I_929067	I_960353	I_943815
I_939261	I_1109776	NM_002086.1	I_1109642	I_1221791	I_931634
I_942166	I_930357	NM_152407.1	I_1100433	I_957775	NM_003663.2
I_961509	I_1002167	I_1109103	I_964029	I_1152081	NM_152353.1
I_958903	I_966504	NM_144990.1	I_961858	I_962114	I_1002378
I_929030	I_957732	I_963106	I_1100670	I_959979	I_956845
I_1109402	I_1201928	I_958959	I_1109891	I_929435	I_1100797
I_963097	I_962355	I_935811	I_959735	I_1109535	I_962269
I_928748	I_1152174	I_931231	I_953745	I_930973	I_960232
I_1109632	I_958690	I_958733	I_943813	I_929412	I_1152443
I_964406	I_957581	NM_178553.1	I_943800	I_959501	I_1100681
I_1100621	I_1151828	I_929193	NM_174905.1	NM_003378.2	NM_145253.1
I_958747	I_945833	I_958470	I_928450	I_932420	I_932208
I_932074	I_932548	NM_144705.1	I_930003	I_935225	I_930754

I_1002262	NM_145333.1	I_944099	I_928537	I_930454	I_930763
NM_005388.2	I_1221918	I_965705	I_929035	NM_178562.1	I_1984412
I_961850	I_962749	I_965645	I_932467	I_928884	I_928665
I_931774	I_960055	I_963294	NM_006928.2	I_1100863	NM_002823.2
I_1109956	NM_018064.1	I_961000	I_1002056	I_962839	I_931645
I_960936	I_958321	I_957754	I_931152	I_956858	I_1109310
I_939856	NM_007131.1	I_1109730	I_930129	NM_018704.1	I_1214078
I_957085	I_962155	I_958272	NM_021038.1	I_1002405	I_928783
I_931322	NM_001419.1	I_966212	I_966484	NM_178552.1	I_1109531
I_958136	I_960418	NM_004474.1	I_959360	I_957498	I_958048
NM_153754.1	I_965483	I_931512	I_937382	I_963367	I_959488
I_957374	I_950764	I_1109312	I_930849	I_961783	I_928797
I_963943	I_962211	I_961792	I_929758	I_935670	I_957044
I_961923	I_1110376	I_964269	I_932344	I_959691	I_960962
I_1100742	I_962803	I_964297	I_1000653	I_1151846	I_944010
I_958744	NM_003584.1	NM_138444.1	I_944073	I_1152003	I_957688
NM_002407.1	I_1110071	I_1100522	NM_173510.1	I_959455	I_966907
I_928441	I_1000517	I_1109363	I_1002341	I_1221789	I_958095
I_961367	I_1000458	I_959367	I_945374	I_961861	I_963939
I_928370	NM_004128.1	NM_022457.4	I_959683	I_957187	I_958345
I_928565	I_929829	I_966666	I_958091	I_936454	I_929205
I_930614	I_1152582	I_943047	I_956822	I_929726	I_929370
I_932620	I_944067	I_957624	I_932098	I_960366	I_966324
I_936330	I_1109721	I_960748	I_931387	I_932189	I_966239
I_958100	NM_003376.3				