Departamento de Ciencias de la Computación e Inteligencia Artificial
Konputazio Zientziak eta Adimen Artifiziala Saila
Department of Computer Science and Artificial Intelligence

Consensus policies to solve bioinformatic problems through Bayesian network classifiers and estimation of distribution algorithms

by

Rubén Armañanzas Arnedillo

Supervised by: Iñaki Inza and Pedro Larrañaga

Dissertation submitted to the Department of Computer Science and Artificial Intelligence of the University of the Basque Country as partial fulfilment of the requirements for the PhD degree in Computer Science

Donostia - San Sebastián, April 2009

Acknowledgements

It is with great pleasure that I can dedicate the following lines to acknowledge all the people that have helped and supported my work during the last four years. It has been a long journey to complete such ambitious work and it is of justice to cite here all of them.

First, I would like to thank my thesis advisers Iñaki Inza and Pedro Larrañaga. I am deeply indebted to them for their great help, friendly guidance and for giving me a good example to follow. They had always been here to help me whenever I needed their support.

Secondly, my gratitude to all the members of the Intelligent Systems Group with whom I have shared these years: Alex Mendiburu, Endika Bengoetxea, Teresa Miquélez, Rosa Blanco, Ramón Sagarna, Roberto Santana, Aritz Pérez, Juan Diego Rodríguez, José Luis Flores, Dinora Morales, Josu Galdiano and Jose Antonio Lozano. Special thanks are dedicated to Borja Calvo with whom I have shared much time working in a domain in which we were kind of pioneers. I will forever remember our work atmosphere as a personal and professional reference for my life; being part of this irreplaceable human group has been an honour.

This dissertation wouldn't have been possible without the financial support offered by the Basque Government under a personal grant AE-BFI-05/430 for training of doctoral researchers. I would also like to cite other projects that supported my research during these years such as the Etortek, Saiotek and Research Groups 2007-2012 (IT-242-07) programs.

My most sincere gratitude to Yvan Saeys, who hosted me in the Bioinformatics and Evolutionary Genomics (BEG) group at Ghent University, Belgium. The experience there is in part reflected in this work.

Lastly, I owe the person I am to my family. My father and mother taught me how to behave in life and that every goal is always achievable with hard work. My sister and brother-in-law gave me constant support throughout these years. I owe all of them the happiness that overwhelms me when writing these last sentences of my dissertation.

Contents

List of Figures

List of Tables

Part I

Introduction

1

Introduction

Biology and machine learning are two scientific fields destined to keep working together. The milestone of the human genome sequencing (International Human Genome Sequencing Consortium, 2001, 2004) was the official starting point of the genomics era within biological research. From that time on, this field has exponentially grown in a very short period of time. And, parallelly, machine learning and data mining disciplines have come across biology so as to deal with the new era data.

We are now aware of the complexity of our own biological mechanism. Human beings have 23 chromosome pairs; the haploid human genome occupies a total of just over 3 billion DNA base pairs; it contains an estimated 20,000 to 25,000 protein-coding genes; and also, RNA genes, regulatory sequences, introns and *junk* DNA. Unfortunately, we are still far from understanding all these intricate mechanisms and, more importantly, being able to repair it when needed.

Disciplines of bioinformatics and computational biology have emerged from the convergence of the new *omics* fields and the computational tools that are needed to manage, store and analyze the huge amount of data produced by them. But, just as the classical biology was not ready to have such a great revolution, neither was classical machine learning ready to deal with the biological data without adaptations.

One of the most classical problems in computational biology research is to extract knowledge from population studies where different phenotypes are considered. Classical examples are, among others, cancer studies that compare healthy individuals with patients, early biomarker discovery for different disease states, drug responses and clinical trials. This kind of research is mapped by the machine learning discipline into the supervised classification problems.

Roughly speaking, supervised classification can be seen as learning from experience. The supervised classification task uses data where the class or the group structure is known in order to learn a mathematical model which is able to classify unseen data samples where the class is unknown. Several models exist to accomplish this task, but, in order to extract useful biological knowl-

edge, the classifiers based on Bayesian networks (Castillo *et al.*, 1997; Jensen and Nielsen, 2007; Neapolitan, 2003; Jensen and Nielsen, 2007) are of the most useful. Bayesian network classifiers (Friedman, 1997; Larrañaga *et al.*, 2005) are a particular type of probabilistic graphical models (Pearl, 1988; Whittaker, 1991; Lauritzen, 1996) that have become very popular paradigms to represent uncertainty.

Optimization approaches have found a new field of application in the omics data. Classical search strategies are unfeasible to deal with high-dimensionality biological problems, where the current computer power is still insufficient to provide exhaustive searches. For instance, to adjust kinetic equations, look for particular patterns in the genomic sequences of different specimens or protein folding by energy minimization, are new optimization problems. In such cases, stochastic search techniques such as estimation of distribution algorithms (Mühlenbein and Paaß, 1996; Larrañaga and Lozano, 2002; Pelikan, 2005) are a perfect solution to tackle these problems.

Nevertheless, machine learning and optimization procedures need accommodation to the specificities of the novel biological data. This dissertation aims to contribute to the state of the art of machine learning techniques adapted for dealing with computational biology problems. In addition, we provide and discuss a set of tools to adequately analyze DNA microarray and mass spectrometry data, which are two of the most popular research domains in computational biology. In the following sections we clarify these contributions, which are introduced throughout the dissertation.

1.1 Contributions of the dissertation

The main contributions of this dissertation are presented as six elements. A brief explanation of each one is given next. Section 1.2 includes the full thesis overview, pointing to the particular chapters and sections where each item is presented and discussed.

A. Consensus approaches within bioinformatic problems

Dealing with a low number of cases is a great challenge to get valid results in a data mining analysis. Even more when the ratio between the number of cases and the number of features is completely unbalanced to the features. In this scenario, we tackle four different machine learning approaches: univariate relevance metrics, discretization policies, reliable dependence Bayesian classifiers and feature selection by estimation of distribution algorithms.

As the first consensus approach, we propose seven different univariate ranking metrics and a way to combine them into a single order ranking. Such a relevance ranking is derived from the positional combination of each individual metric one. Due to its low computational cost, this method is a good approach to have an initial idea of each feature relevance. When a practitioner wants

to retrieve a set of relevant features, machine learning discipline proposes the use of feature subset selection algorithms. We explore here the combination of different discretization methods with a correlation-based feature subset selection. By repeating the same analysis stages using different discretizations, we look for a set of minimum prototypical and relevant features, regardless of the discretization technique used.

The third contribution in the consensus course is a reliably-flexible dependence Bayesian network classifier. By combining a bootstrap approach and a feature subset selection technique, it is possible to induce a hierarchy of directed acyclic graph structures. The sparsity of the network structure can be tuned by increasing/decreasing the minimum confidence level demanded for the edge set. This flexible structure is proposed as a inducer for gene interaction networks and it can be also used on classification purposes by integrating the class variable on the structure.

The last methodological contribution belongs to the optimization field and comprises the consensus population on estimation of distribution algorithms. The classical algorithm outputs the best solution of the search process but does not pay attention to the good intermediate solutions visited throughout the full search. We propose a consensus way to combine all this information to retrieve more robust and stable solutions. Instead of a single one, our proposal outputs a set of solutions to work with. Two metrics to study the consistency and stability of feature subset selectors are also introduced.

B. Quality criteria for microarray data

DNA microarray data is perhaps the most spread gene expression platform worldwide. This high-throughput biological device is able to measure the gene activity of thousands of genes in a single appliance. Despite this capability, it presents a crucial side-effect: the noise that the raw results include. In this dissertation we include a set of metrics to evaluate the reliability of each of the measures a microarray returns.

These criteria are presented from two points of view: to discard/accept a full microarray or to discard/accept a particular gene measuring through a cohort of microarrays. The criteria set are based both on statistics and biology assumptions. Furthermore, different quality metrics are presented for the two main microarray manufacturers, Affymetrix and Agilent.

C. Research on the pathogenesis of two autoimmune diseases

As a first direct application of the consensus policies presented previously, we tackle a gene expression analysis on systemic lupus erytemathosus and primary antiphospholipid syndrome. These are two autoimmune diseases, classified as *rare* due to their low population prevalence, that have an unknown origin and pathogenesis. In order to make an in-depth data analysis, we apply the univariate metric consensus and the consensus gene selection by different discretizations. The found relevant set of genes are validated from a statistical

and biological point of view. Results successfully point to previous reported findings and, more importantly, uncover new possible biological hypothesis to work on.

D. Gene and micro RNA interaction networks

Another two genomics applications are collected in the third part of the dissertation. In the first case, we analyze gene expression data from a local project on colorectal cancer research. The main methodological tool in this case is the reliable dependence identification approach to infer a gene interaction network. Apart from the reliable dependences, results from its application may also identify new possible biomarkers for this kind of cancer. The whole study can be of great importance because a European patent has been submitted with a diagnosis kit based on these and other complementary results.

The induction of reliable interactions is not only limited to gene expression data, and, we also present an open research with a recently discovered genetic molecule, the micro RNA. These small molecules come from the chromosomical DNA but they do not constitute proper genes. MicroRNA are small molecules that are supposed to repress the expression of certain target genes. Within this study two interaction network structures are induced from micro RNA expression data coming from multiple sclerosis and healthy samples. Results enlighten the importance of some micro RNAs which are expected to have a relevant function on the disease.

E. Preprocessing tasks for mass spectrometry data

A mass spectrometer is another general-purpose high-throughput biological device dedicated to the elucidation of the elemental composition of a sample or molecule. It is a high complex device and its outputs are signal profiles of mass charge ratio abundances. Such is the case in the microarrays, the physics of spectrometer devices biases their outcome, adding chemical noise, signal shifts and artifacts. A standard pipeline of *cleaning* tasks is not provided by the scientific community. In the continuous search to reach a standard, we propose our own pipeline of tasks to remove all this unwanted noise from the raw signal. This pipeline, known as preprocessing pipeline, ends with a peak profiling algorithm that identifies possible relevant points in each spectrum. This preprocessing pipeline has been tested with four mass spectra datasets, showing promising results. A Matlab implementation of the pipeline is publicly available on the Internet.

F. Peak selection on mass spectra data by estimation of distribution algorithms

The last application of this thesis work is the selection of relevant peaks on mass spectrometry datasets using the population consensus in estimation of distribution algorithms. This application starts with the proposal of a whole

data analysis workflow to get rid of the overfitting problems that some feature selection schemes present. The analysis workflow includes two validation loops, an honest way to make the preprocessing and relevant peak selections and a multiobjective filter to the population consensus results.

Four public datasets are analysed by the algorithm and the results are compared in terms of predicted classification accuracy, stability degree and coincidence with the original works. In addition, we present a new chart, called as peak frequential plot, that allows an expert to have a straight vision of the results. Using these plots not only are previous findings corroborated, but new research lines are also pointed out.

1.2 Overview of the dissertation

This dissertation is divided into thirteen chapters, which are organized into four main parts. The first part consists of five chapters. The first chapter is an introduction to the dissertation where the reader can find a synthesis of the contributions and how the dissertation is structured. Chapter 2 introduces the molecular biology concepts that are used throughout the dissertation and presents an scheme of how the new omics disciplines interacts among them. Chapter 3 is devoted to explaining the classification tasks in machine learning, focussing on supervised classification and feature selection. The basic mathematical notation is presented as well. Chapter 4 introduces probabilistic graphical models with special attention on Bayesian network models. Finally, Chapter 5 presents estimation of distribution algorithms, provides a taxonomy of them and makes a revision of their current impact in the bioinformatics field.

Part II is dedicated to the consensus adaptations of some machine learning and data mining techniques. Chapter 6 gives an introduction to the drawbacks related with the new high-throughput biological devices, asserting the need for consensus policies when analyzing data produced by those devices. Chapter 7 includes the first consensus approach to univariately assess the relevance of a variable by supervised filter metrics. A multivariate relevance technique is described in Chapter 8 and a way to combine different discretization metrics to select relevant genes on gene expression problems. Chapter 9 proposes a method to identify robust arc dependences in Bayesian classifiers and how this proposal is fitted to the induction of gene interaction networks. Chapter 10 introduces a method to reach a consensus for different populations when using estimation of distribution algorithms as wrapper feature selectors. Two metrics to measure the stability among different subset selections are also presented.

Part III turns the interest to the molecular biology field and illustrates different applications of the new approaches to some computational biology problems or datasets. Chapter 11 reviews the DNA microarray field, introducing different quality criteria for that kind of data. Regarding gene expression

applications, two in depth studies are collected: one for two autoimmune diseases (systemic lupus erythematosus and primary antiphospholipid syndrome) and another for colorectal cancer. In addition, an open study on multiple sclerosis by means of recently discovered molecules, namely micro RNAs, closes the chapter. Chapter 12 exemplifies how the consensus population is applied to four proteomic mass spectra datasets. Results are discussed from different points of view and a new tool to graphically inspect the analyses is presented.

Part IV concludes the dissertation with Chapter 13. This chapter presents some general conclusions, the list of publications and proposals for future work.

2

Mollecular biology, computational biology and bioinformatics

This dissertation lays its foundation in the crossroads between computer science and biology. The computer science paradigms, concepts and techniques used throughout it are introduced in the following Chapters 3, 4 and 5. However, a wide range of biological concepts are also used in the applications, where sometimes a not so profound introduction is made.

In this chapter, we present a brief description of the majority of those biological (sometimes philosophical) concepts. The aim of the dissertation is to present new methodological approaches within the disciplines of bioinformatics and computational biology. Therefore, this chapter only presents a limited introduction to sometimes very vast concepts. In the case that some element remains still unclear, we refer the reader to the huge amount of biology and genetics books, e.g. (Griffiths *et al.*, 2002).

2.1 Cell biology

The cell is the structural and functional unit of all known living organisms. It is the smallest unit of an organism that is classified as living, and is often called the building brick of life. Some organisms, such as most bacteria, consist of a single cell (unicellular). Other organisms, such as humans, are multicellular (estimations for humans are in 10^{14} cells). A typical cell size is 10 μm with an average mass of 1 nanogram. The largest known cell is an unfertilized ostrich egg cell.

The word cell comes from the Latin *cellula*, meaning, a small room. Each cell is a small container of chemicals and water wrapped in a membrane. Each cell can take in nutrients, convert them into energy, carry out specialized functions, and reproduce as necessary. They are usually known as self-contained and self-maintaining entities. There are two types of cells: eukaryotic and prokaryotic.

- Prokaryotic cells are usually independent and they show the simplest structure. A prokaryotic cell lacks a nucleus and most of the internal organs or

organelles that an eukaryotic presents. There are two kinds of prokaryotes, bacteria and archaea, with a similar overall structure.

- Eukaryotic cells are often found in multicellular organisms. The major difference between prokaryotes and eukaryotes is that eukaryotic cells contain membrane-bound compartments in which specific metabolic activities take place. Among these differences, the most important is the presence of a cell nucleus: a membrane-delineated compartment that houses the eukaryotic cell's DNA (see Section 2.1.2).

Eukaryotic cells also have other specialized organelles. As the name implies, you can think of organelles as small organs. There are a dozen different types of organelles commonly found in eukaryotic cells. Figure 2.1 list the primary components of the eukaryotic cell.

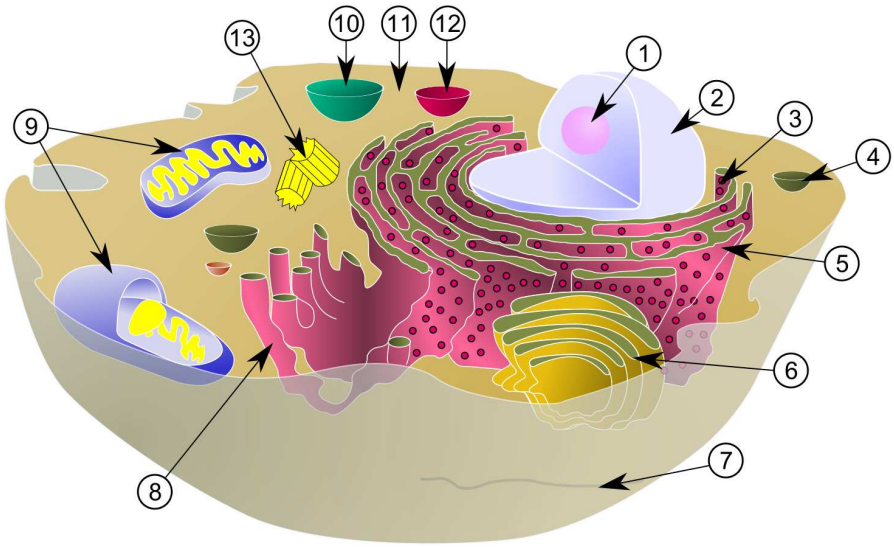


Fig. 2.1. Diagram of a typical eukaryotic cell. Organelles are labelled as follows: 1. Nucleolus; 2. Nucleus; 3. Ribosome; 4. Vesicle; 5. Rough endoplasmic reticulum; 6. Golgi apparatus; 7. Cytoskeleton; 8. Smooth endoplasmic reticulum; 9. Mitochondrion; 10. Vacuole; 11. Cytosol; 12. Lysosome; 13. Centriole.

Some organelles, such as the nucleus or Golgi apparatus, are solitary, while others, such as mitochondria and lysosomes, can be numerous (hundreds to thousands). The cytosol is the gelatinous fluid that fills the cell and surrounds the organelles. Three organelles carry out key tasks for the survival of the cell:

- Mitochondria - The mitochondria are the principal energy source of the cell (thanks to the cytochrome enzymes of terminal electron transport and the enzymes of the citric acid cycle, fatty acid oxidation, and oxidative phosphorylation). The mitochondria convert nutrients into energy as well as

doing many other specialized tasks. Each mitochondrion has a chromosome composed of DNA that is quite different from the chromosomes in the nucleus. We inherit our mitochondrial chromosome from our mother, thus, it is transmitted in a matrilinear manner. It is much smaller than the regular nucleus chromosomes and there are many copies of it in every cell, whereas there is normally only one set of chromosomes in the nucleus.

- Ribosomes - The ribosome is a large complex of RNA and protein molecules. This is where proteins are produced in the translation process (see Section 2.1.4).
- Cell nucleus - It is the most noticeable organelle found in a eukaryotic cell. The nucleus is spherical in shape and separated from the cytoplasm by a double membrane called the nuclear envelope. It houses the cell's chromosomes, and it is responsible for maintaining the integrity of these chromosomes and controling the activities of the cell by regulating gene expression. The nucleolus is a specialized region within the nucleus where ribosome subunits are assembled. The nucleus is where the transcription process takes place (see Section 2.1.4).

2.1.1 Genome

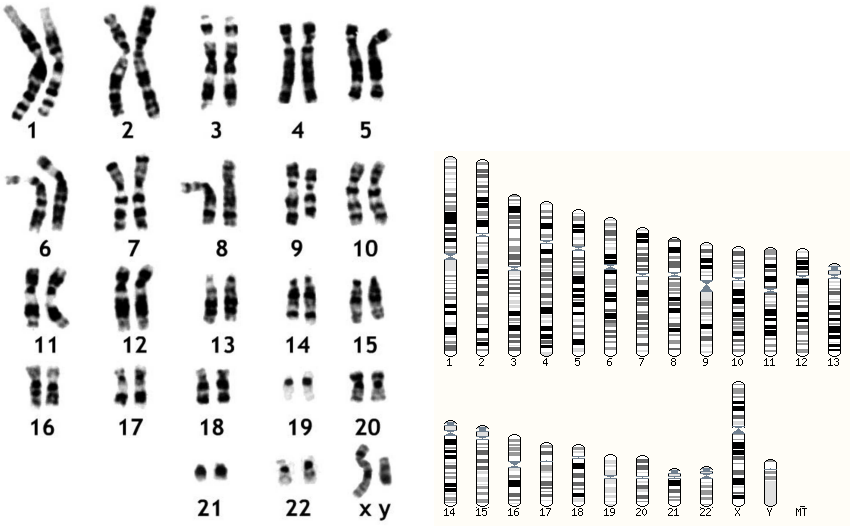
The nucleus of most human cells contains two sets of chromosomes, one set given by each parent. Each set has 23 single chromosomes, 22 autosomes and an X or Y sex chromosome. There are notable exceptions including the egg and sperm cells (each of which have only 23 chromosomes containing half the usual amount) and mature red blood cells (which no longer have a nucleus and, so, lack chromosomes).

Chromosomes can be seen under a light microscope. Differences in size and composition allow the 24 chromosomes to be distinguished from each other, an analysis called a karyotype (see Figure 2.2 (a)). A few types of major chromosomal abnormalities, including missing or extra copies or gross breaks and rejoinings (translocations), can be detected by microscopic examination. For example, Down's syndrome is due to the inclusion of a third copy of chromosome 21 in an individual's cells.

Near the center of each chromosome is its centromere, a narrow region that divides the chromosome into a long arm (q) and a short arm (p). We can further divide the chromosomes using special stains that produce stripes known as a banding pattern. Each chromosome has a distinct banding pattern, and each band is numbered to help identify a particular region of a chromosome. This method of mapping known as cytogenetic mapping gives a bird's eye view of each chromosome. Figure 2.2 (b) presents the Ensembl [1] representation of the homo sapiens cytogenetic mapping.

Chromosomes are made of deoxyribonucleic acid (DNA), and genes are special units of chromosomal DNA. Each chromosome is a very long DNA

[1] `http://www.ensembl.org`



(a) Example of a complete human karyotype.

(b) Classical cytogenetic map representation.

Fig. 2.2. Different representations of human chromosomes. Subfigure (a) displays a microscope karyotype; Subfigure (b) shows the schematic view of a cytogenetic mapping.

molecule, so it needs to be wrapped tightly around proteins for efficient packaging. Chromosomes contain the entire human genome, roughly speaking all the genetic information necessary to build a human being.

2.1.2 DNA molecule

DNA is a double-stranded molecule held together by weak hydrogen bonds between base pairs of nucleotides. The molecule forms a double helix in which two strands of DNA spiral about one another. These two strands run in opposite directions to each other and are therefore anti-parallel. The double helix looks something like an immensely long ladder twisted into a helix, or coil. The sides of the *ladder* are formed by a backbone of sugar and phosphate molecules, and the *rungs* consist of nucleotide bases weakly joined in the middle by the hydrogen bonds (see Figure 2.3).

The DNA chain is 22 to 26 Ångströms wide (2.2 to 2.6 nanometres), and one nucleotide unit is 3.3 Å long. Although each individual base is very small, DNA polymers can be very large molecules containing millions of nucleotides. For instance, the largest human chromosome, chromosome number 1, is approximately 220 million base pairs (bp) long.

There are four nucleotides in DNA. A DNA nucleotide is made of a five-carbon sugar, a molecule of phosphoric acid, and a molecule called a base. The

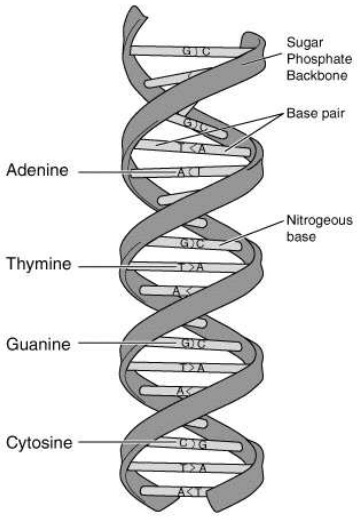


Fig. 2.3. The double helix of the DNA molecule. Each spiral strand, composed of a sugar phosphate backbone and attached bases, is connected to a complementary strand by hydrogen bonding (non- covalent) between paired bases, adenine (A) with thymine (T) and guanine (G) with cytosine (C). Adenine and thymine are connected by two non-covalent hydrogen bonds while guanine and cytosine are connected by three.

bases are the *letters* that spell out the genetic code. In DNA, the code letters are A, T, G, and C, which stand for the chemicals adenine, thymine, guanine, and cytosine, respectively. In DNA base pairing, adenine always pairs with thymine, and guanine always pairs with cytosine (see Figure 2.4). Due to this chemical property, the base sequence of each single strand of DNA can be simply deduced from that of its partner strand.

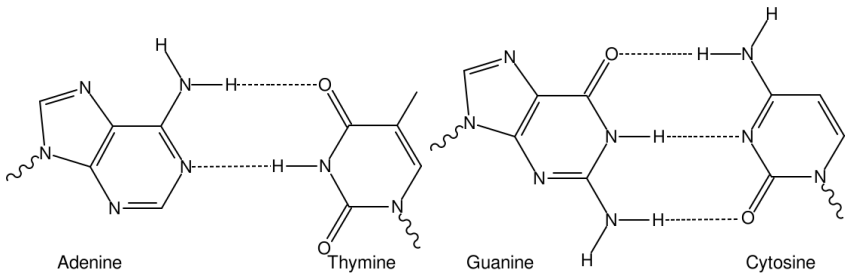


Fig. 2.4. Chemical scheme of one A–T and G–C base pair bonds. Non-covalent hydrogen bonds between the pairs are shown as dashed lines.

It is the sequence of these four bases along the DNA's backbone that encodes information. The DNA genome of an organism is in fact comprised of the sequence of all the nucleotide bases for all the existing chromosomes. However, this raw information is not directly read/translated by the cells; it is encoded in what is known as genetic code.

This genetic code is the set of rules by which information encoded in genetic material is translated into proteins (amino acid sequences) by living cells. The canonical genetic code defines a mapping between tri-nucleotide sequences, called codons, and amino acids. A triplet codon in a nucleic acid sequence usually specifies a single amino acid (see Section 2.1.3 and Table 2.1). Nevertheless, there are many variant codes to the canonical one: e.g. human protein synthesis in mitochondria relies on a genetic code that varies from the canonical code. And, most importantly, not all the genetic information is stored as genetic code. All organisms' DNA contain regulatory sequences, intergenic segments, chromosomal structural areas, which operate using a distinct sets of rules maybe not as straightforward as the codon-to-amino acid paradigm.

2.1.3 RNA molecule

RNA stands for ribonucleic acid, a nucleic acid molecule similar to DNA. RNA and DNA differs in a few important structural details: in the cell, RNA is usually single-stranded, while DNA is usually double-stranded; RNA nucleotides contain ribose while DNA contains deoxyribose; and, the nucleotide base thymine (T) that is present in DNA is replaced by uracil (U) in RNA. RNA play crucial roles in protein synthesis and other cell activities.

RNA is formed upon a DNA template. Synthesis of RNA is usually catalyzed by the RNA polymerase –an enzyme (protein) that assembles the RNA from ribonucleotides–. RNA polymerase produces the RNA strand by using DNA as a template in a process known as transcription. Initiation of transcription begins with the binding of the enzyme to a promoter sequence in the DNA. The DNA double helix is unwound by the helicase activity of the enzyme. Then, the enzyme progresses along the template strand synthesizing a complementary RNA molecule. The DNA sequence also dictates where termination of RNA synthesis will occur.

There are more than thirty classes of RNA molecules. Among the most important ones we can cite the following three:

- Messenger RNA (mRNA) - mRNA is the RNA that carries information from DNA to the ribosome, the factories of protein synthesis in the cell. The coding sequence of the mRNA determines the amino acid sequence in the protein that is produced. The sequence is again elucidate in basis of the genetic code read in codons. Table 2.1 presents the codification from mRNA nucleotide bases to protein amino acids.
- Transfer RNA (tRNA) - tRNA is a short-chain type of RNA present in cells. There are 20 varieties of tRNA. Each variety combines with a specific

amino acid and carries it along, leading to the formation of protein with a specific amino acid arrangement dictated by DNA.

- Ribosomal RNA (rRNA) - rRNA is a component of ribosomes. Ribosomal RNA functions as a nonspecific site for making polypeptides.

Amino acid	**Coding codon(s)**	**Amino acid**	**Coding codon(s)**
Alanine (A)	GCU, GCC, GCA, GCG	Leucine (L)	UUA, UUG, CUU, CUC, CUA, CUG
Arginine (R)	CGU, CGC, CGA, CGG, AGA, AGG	Lysine (K)	AAA, AAG
Asparagine (N)	AAU, AAC	Methionine (M)	AUG
Aspartic acid (D)	GAU, GAC	Phenylalanine (F)	UUU, UUC
Cysteine (C)	UGU, UGC	Proline (P)	CCU, CCC, CCA, CCG
Glutamine (Q)	CAA, CAG	Serine (S)	UCU, UCC, UCA, UCG, AGU, AGC
Glutamic acid (E)	GAA, GAG	Threonine (T)	ACU, ACC, ACA, ACG
Glycine (G)	GGU, GGC, GGA, GGG	Tryptophan (W)	UGG
Histidine (H)	CAU, CAC	Tyrosine (Y)	UAU, UAC
Isoleucine (I)	AUU, AUC, AUA	Valine (V)	GUU, GUC, GUA, GUG
START	AUG	*STOP*	UAG, UGA, UAA

Table 2.1. Genetic code for the translation between mRNA codons and amino acids.

Once the transcription is carried out by the RNA polymerase, the RNA strand is then processed to give messenger RNA (mRNA), which is free to migrate through the cell. There, mRNA molecules bind to ribosomes located in the cytosol, where they are translated into polypeptide sequences according to the rules specified by the genetic code (see Table 2.1). This process is known as translation, and it proceeds in four phases: activation, initiation, elongation and termination (all describing the growth of the amino acid chain, or polypeptide that is the product of translation).

Translation starts with a chain initiation codon (start codon). The codon alone is not sufficient to begin the process and nearby sequences and initiation factors are also required to start translation. The most common start codon is AUG, which also codes for methionine. The three stop codons are UAG, UGA and UAA. Stop codons are also called termination codons and they signal release of the nascent polypeptide from the ribosome. The new polypeptide then folds into a functional three-dimensional protein molecule.

2.1.4 Gene expression

Each of the 46 human chromosomes contains the DNA for thousands of individual genes, the units of heredity. There are estimated 20,00025,000 human protein-coding genes, although this number could drop. Human genes are distributed unevenly across the chromosomes: each chromosome contains various gene-rich and gene-poor regions. In cells, a gene is a portion of DNA that contains both coding sequences that determine what the gene does, and non-coding sequences that determine when the gene is expressed (active).

Coding sequences are known as exons and code for a specific portion of a complete protein. Depending on the context, exon can refer to the sequence in the DNA or its RNA transcript. Its counterpart, DNA non-coding sequences are called introns (intra-genic regions). These non-coding sections are present in the precursor mRNA (pre-mRNA) directly translated for the DNA sequence, and removed by a process called splicing during the processing to mature RNA. The mature RNA molecule can be a messenger RNA or a functional form of a non-coding RNA such as rRNA or tRNA. As a general view, we will only consider mRNA that, after intron splicing, consists only of exons, which are translated into a protein.

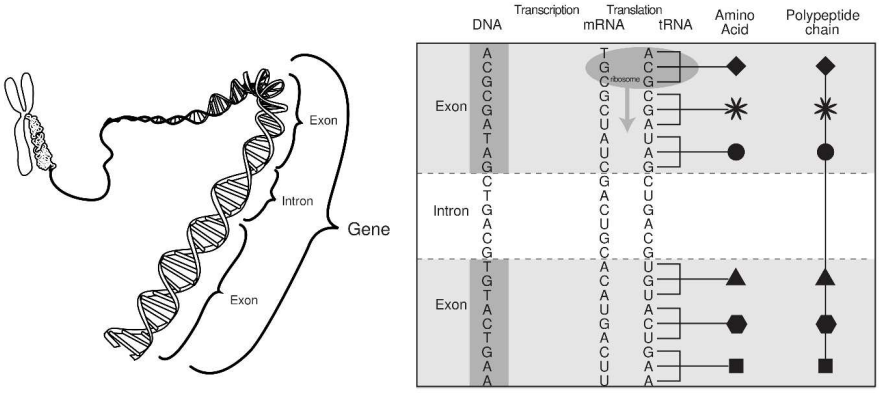


Fig. 2.5. Detailed description of a gene expression process. The left part presents the intron-exon configuration of a gene. The right part illustrates how, from the DNA sequence, the transcription and translation processes produce a polypeptide chain in basis of the genetic code.

In summary, when a gene is active, the coding and non-coding sequences are transcribed, producing an RNA copy of the gene's information. This piece of RNA can then direct the synthesis of proteins via the genetic code (see Figure 2.5). In other cases, the RNA is used directly, e.g. tRNA, ribosomal RNA, microRNA, and other non-coding RNA genes. The molecules resulting from gene expression, whether RNA or protein, are known as gene products

or transcripts, and are responsible for the development and functioning of all living things.

2.1.4.1 Gene expression monitoring

A gene activity in an organism is determined by the rates of its expression and degradation. There are nowadays different techniques to measure such gene activity, among the most known and used are the Northern blot, Real-time PCR, Multiplex PCR, SAGE or DNA microarray. In all of them and in order to robustly detect and accurately quantify the expression from small amounts of DNA/RNA, amplification of the gene transcript is necessary.

For this purpose the polymerase chain reaction was invented by K. Mullis in 1983 . The polymerase chain reaction (PCR) is a molecular biology technique whose name derives from one of its key components, the DNA polymerase molecule. PCR is used to amplify a piece of DNA by in vitro enzymatic replication. As the reaction progresses, the DNA generated is used again as a template for replication, in consequence the DNA template is exponentially amplified. With PCR it is possible to amplify a single or few copies of a piece of DNA across several orders of magnitude.

Three major steps are involved in a PCR. These three steps are repeated for 30 or 40 cycles. The cycles are done on an automated cycler, a device which rapidly heats and cools the test tubes containing the reaction mixture. Each of the three steps takes place at different temperatures:

- Denaturation (94°C) - The double-stranded DNA melts and opens into two pieces of single-stranded DNA.
- Annealing (54°C) - The reaction primers pair up with the single-stranded DNA sequence to be copied. On the small length of double-stranded DNA, the polymerase attaches and starts copying the template.
- Extension (72°C) - DNA building blocks complementary to the template are coupled to the primer, making a double stranded DNA molecule.

With one cycle, a single segment of double-stranded DNA template has thus been amplified into two separate pieces of double-stranded DNA. These two pieces are then available for amplification in the next cycle.

PCR has found widespread and innumerable uses: to diagnose genetic diseases, do DNA fingerprinting, find bacteria and viruses, study human evolution, clone the DNA of an Egyptian mummy, etc.. Accordingly, PCR has become an essential tool for biologists, DNA forensics labs, and many other laboratories that study genetic material. The reaction is easy to execute and requires no more than a test tube, a few simple reagents, and a source of heat.

Based on the polymerase chain reaction, the real-time polymerase chain reaction, also called quantitative real time polymerase chain reaction (Q-PCR/qPCR), was invented. qPCR makes use of the polymerase chain reaction to amplify and simultaneously quantify a targeted DNA molecule. Both the detection and quantification of a particular DNA sequence is possible by means

of qPCR. The procedure follows the general principle of polymerase chain reaction. The key feature is that the amplified DNA is in real time quantified as it accumulates in the reaction after each amplification cycle. The biological foundations of qPCR are complex and there have been different approaches to the quantification of the DNA/RNA sequence increase. For a more in depth explanation, we refer to the revision by (Nolan *et al.*, 2006).

2.2 Omics

The suffix *-om-* refers to totality of some sort. The English neologism *omics* informally addresses those biology fields in which the objects of study conform a totality. The first example was genomics, which stands for the study of the genome. Due to the success of high-throughput biological devices and new analytical tools, the suffix *-om-* has also been picked up by a wide array of other large-scale quantitative biology fields. Many of them are very recent terms that are not fully accepted by all the research community[2], among the most established we can find:

Genomics	study of the genome, the entire DNA sequence of organisms, the genetic mapping and its activity which also includes studies of intragenomic phenomena and other interactions between loci and alleles within the genome.
Transcriptomics	study of the transcriptome, the mRNA complement of an entire organism, tissue type, or cell.
Proteomics	study of the proteome, the protein complement of an entire organism, tissue type, or cell.
Metabolomics	study of the metabolome, the totality of metabolites in an organism.
Spliceomics	study of the spliceosome, the totality of the alternative splicing protein isoforms.
Glycomics	study of glycome, the totality of glycans, carbohydrate structures of an organism, a cell or tissue type.
Lipidomics	study of the lipidome, the totality of lipids.

In general, the term omics focuses on large scale and holistic data research to understand life in encapsulated omes. It is common to use the term omics referring to the comprehensive integration of analyses from different layers of the biological systems. New technology is developing constantly and quickly, so omics disciplines will not only have an impact on our understanding of biological processes, but the prospect of more accurately diagnosing and treating disease is becoming a reality.

[2] A full list of omics disciplines can be found at `http://omics.org`

2.3 Computational biology and bioinformatics

There is nowadays some ambiguity when dealing with these two terms. Some authors state that bioinformatics is the main discipline while computational biology is one of its integrated subdisciplines. Others stress the opposite, that bioinformatics is a subdiscipline of computational biology. Since there is no accepted consensus for this discussion, we here gather definitions that support the latter statement, which we believe is the most adequate.

Computational biology can be defined as a new interdisciplinary field that applies the techniques of computer science, applied mathematics and statistics to address biological problems. Being a very general definition, it encompasses many different fields:

- Computational biomodeling - A field within biocybernetics concerned with building computational models of biological systems. These biomodels try to mathematically emulate the biological mechanisms involved in a particular system. Examples in this discipline can be genetic networks, enzyme kinetics, cancer cell models, or bigger scale models such as the modelling of the heart.
- Computational genomics - A field within genomics which studies the genomes of cells and organisms. Computational genomics focuses on understanding the human genome, and more generally the principles of how DNA controls the biology of any species at the molecular level. In addition, high-throughput genome sequencing produces lots of data, which requires extensive post-processing genome assembly. It uses DNA microarray technologies to perform statistical analyses on the genes expressed in individual cell types. This can help find genes of interest for certain diseases or conditions. This field also studies the mathematical foundations of sequencing.
- Molecular modeling - A field complementary to the computational biomodeling, it focuses attention on modelling the behaviour of molecules of biological importance. The fields of application can range from small chemical systems to large biological molecules. The main difference with computational biomodeling is that molecular modeling mimics the external behaviour of the elements under study, whereas biomodeling tries to mimic the full entity. Typical examples are potential or energy functions that simulate biological behaviours by mathematics.
- Protein structure prediction - Also including structural genomics, this field concentrates on systematically producing accurate structural models for three-dimensional protein structures that have not been determined experimentally. It deals with the prediction of a protein's tertiary structure from its primary structure. This tertiary structure (spacial distribution) of the amino acids sequence of a protein fixes its functionality and behaviour. Experimental methods such as X-ray crystallography or NMR spectroscopy are very expensive and high time-consuming, so computational approaches

are used to predict the real structure in the meantime. Protein structure prediction is of high importance in drug and novel enzymes designs.

- Computational biochemistry - Very related to the last item, this field makes extensive use of structural modeling and simulation methods in an attempt to elucidate the kinetics and thermodynamics of protein functions. Kinetic equations are differential equations without a close solution, in this field computational optimization is used to adjust the *a priori* kinetic models to the real measures obtained by experimental methods.

The last item is the bioinformatics. Bioinformatics can be defined as the field which applies algorithms and statistical techniques to the interpretation, classification and understanding of biological datasets. From this definition it is easy to induce that bioinformatics applies to all of the computational biology subdisciplines (all of them produce biological datasets for analysis). There are a wide range of bioinformatics applications: DNA (RNA) or protein sequence analysis and alignment, comparisons of homologous sequences, gene finding and prediction, gene expression analysis, protein-protein interactions, genome-wide association studies, the modeling of evolution, and many others.

One of the key features of bioinformatics and computational biology in general is its intensive use and development of data mining and machine learning algorithms (Inza *et al.*, 2009; Larrañaga *et al.*, 2006). We can think of algorithms to assess relationships among members of large data sets, methods to locate a gene within a sequence, predict protein structure and/or function, and cluster protein sequences into families of related sequences. However, it is relatively easy to find references to bioinformatics when referring to the implementation of tools that enable efficient access and management of various types of biological information. Although this discipline is a necessary consequence of the former bioinformatics definition, the term biodata managing seems to be more appropriate in this case.

Recently, especially from year 2,000 onwards, a new discipline related to all of the previous has arisen: the systems biology. Most of the times, computational biology disciplines most of the times reduce the complexity of the domain under study by imposing design constraints or limitations. Sometimes this is the only way to tackle complex systems and to obtain a good but not perfect result or approach. Conversely, systems biology tries to integrate the study of complex interactions in biological systems in a *holism* way: the sum of the parts does not explain the whole. This new perspective of systems biology is aimed at the discovery of properties that may arise from the systemic view and, thus, to better understand the entirety of processes that happen in a biological system.

In many occasions, the term systems biology is used to refer to a research paradigm in biology, the antithesis to the reductionist paradigm. In general, a systems biology point of view is the one that tries to integrate results from different omics and retrieve new conclusions from all the interactions that combination produces. As a graphical summary, Figure 2.6 presents a par-

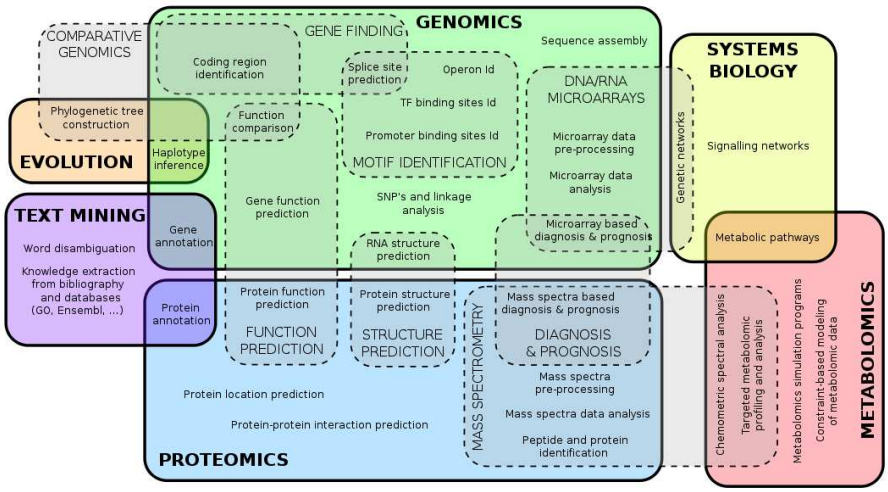


Fig. 2.6. Computational biology field with the data and problem interactions produced among its several subdisciplines.

tial vision of all the problems, applications and disciplines integrated within computational biology.

3

Classification tasks within machine learning

The verb to classify comes from the Latin *classificare* and literally means "to arrange or assign according to type". Machine learning borrows this definition and states that a classification task involves assigning a given dataset categories or labels from a set of possible classes. This assignment is not possible without the previous existence of a mathematical tool or classification model that allows such a task. To this end, classification includes two different subtasks: first, the way to create, induce or learn the classification model or classifier; and, secondly, the procedure to assign their corresponding labels or categories to similarly formed data (new observations).

The introduction to the former task will be discussed in Chapter 4, while the latter task is presented in the present one. A taxonomy of the classification problems is briefly discussed with more detail on the supervised classification. Since all the problems and advances presented in this discussion belong to the supervised classification domain, the principal concepts to evaluate the goodness of a classifier are also presented in detail. Lastly, the basics of the supervised feature subset selection are gathered.

But first of all, let us set up some notation and probability concepts that will be present throughout all this dissertation.

3.1 Notation from the probability theory

A random variable is a function that associates a numerical value with every outcome of a random experiment. A unidimensional random variable is denoted by X and x denotes a particular value for that random variable. When a random variable belongs to a n-dimensional space it is denoted as $\boldsymbol{X} = (X_1, \ldots, X_n)$, where each X_i with $i = 1, \ldots, n$ is a unidimensional random variable.

A set of values for a n-dimensional random variable $\boldsymbol{X}$ is represented by $\boldsymbol{x} = (x_1, \ldots, x_n)$, and is also known as an instance of $\boldsymbol{X}$. A set of N different instances of $\boldsymbol{X}$ is called a dataset and denoted as $D = \{\boldsymbol{x}^1, \ldots, \boldsymbol{x}^N\}$.

Depending on the particular context of the discussion, the nouns instance, observation, case or sample are used indistinctly to refer to the same concept. In the same way, the nouns feature or variable are used to refer to the concept of random variable.

If the opposite is not stated, upper-case letters denote random variables while lower-case letters denote the values of such variables. Boldface typeset represents a vector, either of random variables or instances of them.

The generalized joint probability distribution of $\boldsymbol{X}$ over a point $\boldsymbol{x}$ is represented by $\rho(\boldsymbol{X} = \boldsymbol{x})$ or just by $\rho(\boldsymbol{x})$. Similarly, the marginal generalized probability distribution of X is denoted as $\rho(x)$ and the conditional generalized distribution of X_i given X_j through $\rho(x_i|x_j)$.

A discrete random variable, or discrete variable, is a random variable that takes a numerable number of values. In opposition, a continuous random variable is that one whose domain is not numerable. If every unidimensional random variable X_i of a n-dimensional one is discrete, then $\rho(\boldsymbol{x}) = p(\boldsymbol{x})$ is known as the joint probability mass function of $\boldsymbol{X}$. Similarly $p(x)$ and $p(x_i|x_j)$ refers to the marginal and conditional probability mass functions, respectively. Analogously for the continuous case, $\rho(\boldsymbol{x}) = f(\boldsymbol{x})$ is the joint density function of $\boldsymbol{X}$ and $f(x)$ and $f(x_i|x_j)$ represents the marginal and conditional density functions.

3.2 Unsupervised, semisupervised or supervised classification

A classification dataset D is comprised of a set of N observations (cases or instances), each of which is described by $n+1$ random variables. The first n variables, $X_1, X_2, ..., X_n$, are known as *predictive variables*, and the variable in the $n+1$ position is the *class* variable C, or the *supervised variable*. Table 3.1 gathers the classical disposition of D into observations.

	X_1	X_2	...	X_i	...	X_n	C
1	x_1^1	x_2^1	...	x_i^1	...	x_n^1	c^1
⋮	⋮	⋮	⋱	⋮	⋱	⋮	⋮
j	x_1^j	x_2^j	...	x_i^j	...	x_n^j	c^j
⋮	⋮	⋮	⋱	⋮	⋱	⋮	⋮
N	x_1^N	x_2^N	...	x_i^N	...	x_n^N	c^N

Table 3.1. Data matrix of a (supervised) classification task.

However, the full knowledge of C and/or some or all of its values is not always at hand. The consequence is that the classification problem can be

divided into different subtypes, namely, *unsupervised*, *semisupervised* or *supervised classification*. The boundary among them relies on the degree of knowledge we have about the class variable. Given a classification dataset D, if all the data is present with all its different categories or classes then we talk about a supervised classification problem. When the degree of knowledge is not so great, and some of the data could be assured to belong to a class but nothing can be said for the rest of the data, we have a semi-supervised problem. Finally, when we have uncategorised data and our aim is to find the inherent categories that could explain the data characteristics and group the observations into those categories, then we shall talk about an unsupervised classification or clustering problem.

Among these three categories, more particular subtypes of classification problems have already been reported. The following enumeration presents some key characteristics of the main classification categories, plus some from the more specific classification subtypes:

- *Supervised classification.* In general, these are classification problems for which all the samples are labeled beforehand. The classical formulation (Duda *et al.*, 2001; Bishop, 2006) needs the explicit presence of samples from all the classes. A particular variant of this scheme, namely *one class classification* (Manevitz and Yousef, 2001; Tax and Duin, 2002), occurs when the class can only take two values but the data available only contain samples from one of the classes. Classification applications are found in a vast number of scientific fields: from the early introduction of computers, supervised classification was also introduced in the modern way of life. Among many others, we can cite applications in medicine, computer vision, statistics and biology.
- *Semi-supervised classification.* There are domains in which to state the membership or class of all samples is not possible. Under this contraint, a dataset may contain labeled samples of one or more classes, but, at the same time, samples whose membership is unknown. Due to that uncertainty, these problems are known as semi-supervised (Zhu, 2005; Chapelle *et al.*, 2006). Similarly to the supervised case, there are particularisations of the general scheme. For instance, the *positive unlabelled* (Denis *et al.*, 2002; Calvo, 2008) problem, where the class only takes two values and the dataset only contains positive and unlabelled samples. Other domains for semi-supervised classification are the web-mining and text-mining domains where there are only a few labelled examples and a huge amount of unlabelled instances.
- *Unsupervised classification.* Widely known as clustering problem (Forgy, 1965; Jardine and Sibson, 1971; Bezdek, 1981), in this problem there is no knowledge about the class variable. Not only are the samples unlabelled, but also the number of possible values for the class is unknown. Clustering constitutes a well-established and successful application of classification in many fields such as biology, medicine, population surveys, image seg-

mentation, chemistry or geology, among others, and it is possible to find a huge number of works on its state of the art.

All the problems tackled in this dissertation belong to the supervised classification domain. Therefore, all the methodologies presented are devoted to taking advantage of the supervised knowledge. Let us introduce how the supervised classification can be formally modelled by means of probability theory. A classifier can be seen as a function that assigns labels to observations,

$$\gamma : (x_1, \ldots, x_n) \rightarrow \{1, 2, \ldots, m\},$$

where $\boldsymbol{x} = (x_1, \ldots, x_n) \in \mathcal{R}^n$ conforms the observation and $\{1, 2, \ldots, m\}$ are the range of possible values for the class variable C. The real class is denoted by c and takes values among that range. The main assumption is the existence of an unknown underlying probability joint distribution where the observations come from

$$p(x_1, \ldots, x_n, c) = p(c|x_1, \ldots, x_n)p(x_1, \ldots, x_n) = p(x_1, \ldots, x_n|c)p(c). \quad (3.1)$$

In practice, this joint probability distribution $p(x_1, \ldots, x_n, c)$ can be estimated from a random sample,

$$\left\{(\boldsymbol{x}^1, c^1), \ldots, (\boldsymbol{x}^N, c^N)\right\},$$

extracted from the true joint probability distribution.

There exists a cost matrix, $cost(r, s)$ with $r, s \in \{1, \ldots, m\}$, for which the associated cost of misclassifications are collected. In particular, $cost(r, s)$ contains the associated cost of classifying an element from class r as belonging to class s. In the case that a 0/1 loss function is employed, we will have:

$$cost(r, s) = \begin{cases} 1 \text{ if } r \neq s \ , \\ 0 \text{ if } r = s \ . \end{cases}$$

Thus, the main objective is to induce a classifier that minimizes the cost of the total number of misclassifications. This is done by means of the *Bayes classifier* (Duda and Hart, 1973):

$$\gamma(\boldsymbol{x}) = \arg\min_k \sum_{c=1}^{m} cost(k, c)p(c|x_1, \ldots, x_n) \ .$$

The classification paradigms discussed in this dissertation belong to the generative or informative classification family. Generative classifiers aim to model the probability distribution of Equation 3.1 that generated the data. This purpose its attained by using the Bayes rule to obtain the class conditional probabilities,

$$p(c|x_1, \ldots, x_n) = \frac{p(c, x_1, \ldots, x_n)}{\sum_{c'} p(c', x_1, \ldots, x_n)} \ .$$

When the classification scheme assumes a 0/1 loss function (Friedman, 1997), the Bayes classifier assigns to a given observation $\boldsymbol{x} = (x_1, \ldots, x_n)$ the class with higher *a posteriori* probability (Duda and Hart, 1973):

$$\gamma(\boldsymbol{x}) = \arg\max_c p(c|x_1, \ldots, x_n) \ .$$

On a more general scheme, the observations could be classified according to different criteria, that is, more than one class. This scheme is known as *multidimensional classification* and presents a multidimensional class variable $\boldsymbol{C}$, that integrates each individual criterion. Multidimensional classification is a very recent topic under study and there exists very few works on the simultaneous prediction of more than one class (Van der Gaag and de Waal, 2006; de Waal and Van der Gaag, 2007; Rodríguez and Lozano, 2008).

3.3 Classifier evaluation

On a classification problem, a way to measure the goodness of a given classifier or classification paradigm is imperative. Not only to have an idea of the classification ratio, but also to be able to choose the most suited classification paradigm for a given problem. This evaluation is comprised of two parts, a performance measure and a way to estimate it. Both tasks have been long discussed and in this section we present the most representative techniques to tackle both issues.

3.3.1 Measures of performance

The *classification error* of a classifier γ, ϵ_γ, is defined as the probability that γ mistakes the real class of a new instance $\boldsymbol{x}$. Formally,

$$\epsilon_\gamma = \sum_{\boldsymbol{x}} p(\gamma(\boldsymbol{x}) \neq c) p(\boldsymbol{x}) \ ,$$

where c is the actual class of $\boldsymbol{x}$.

Most often, the classification error is expressed in terms of its complementary measure, that is, the *classification accuracy*. The accuracy of a given classifier, Acc_γ, is thus the probability of correctly classifying a new instance $\boldsymbol{x}$:

$$Acc_\gamma = \sum_{\boldsymbol{x}} p(\gamma(\boldsymbol{x}) = c) p(\boldsymbol{x}) \ .$$

Classification error and/or accuracy are the most often used performance measures to illustrate the goodness of a classification model. We should be aware that this is only fair when the error cost is equally distributed for all classes. However, when this cost is not independent of the class distribution, the total cost should be decomposed and other more specialized performance

measures are set out. Moreover, when a data set is unbalanced (the number of samples in different classes vary greatly) the error rate of a classifier is not representative of the true performance of that classifier.

Large research has been undertaken on binary classification problems in order to propose new and more specialized performance measures. In this kind of problem, we can inspect the performance of a classifier showing its evaluation *confusion matrix*. Table 3.2 presents the classical disposition of a confusion matrix on dichotomic classification. Each column gathers how many instances have been classified as been either Positive or Negative. The rows indicate how many of those classifications were according to the reality or actual class label and how many were not.

		PREDICTED $\gamma(\boldsymbol{x})$	
		Positive	Negative
REAL C	Positive	TP	FN
	Negative	FP	TN

Table 3.2. Confusion matrix in binary classification problems.

The main diagonal values in a confusion matrix correspond to the corrected classified instances, which are the number of *true positive* (TP) and the number of *true negatives* (TN). The missclassification values are divided into *false negatives* (FN) and *false positives* (FP), depending on the direction of the mistake.

Usually, the cost function is configured as a 0/1 loss function. In such cases, both the cost of a false positive error is the same as the cost of a false negative one. Then, the classifier's accuracy value corresponds to the ratio between the sum of TP and TN and the total number of classified instances. However, other specific measures are available, such as the *sensitivity* and the *specificity* values.

The sensitivity, S_n, is the ratio of positive instances that are correctly classified as positive. Also known as *recall*, r, or *true positive rate*, TPR, it is computed as

$$S_n = r = \mathrm{TPR} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}.$$

The counterpart of sensitivity for the negative instances is the specificity, S_p. The ratio of actual negative cases that are correctly classified is thus the specificity value,

$$S_p = \frac{\mathrm{TN}}{\mathrm{TN} + \mathrm{FP}}.$$

In some domains, e.g. health care, the cost of a false negative can be orders of magnitude higher than the cost of a false positive: Imagine a patient with cancer that is suggested as not having. Oppositely, there are other domains, e.g. finance systems or real time systems, where the cost of a false positive is orders of magnitude higher than the cost of a false negative. Imagine a high investment in a financial asset that later falls in the market, or the detection of a person when it was an animal passing by. In all those scenarios, two more measures are of great interest when developing the classification systems, the *false positive rate* and the *precision* of a classifier.

The false positive rate, FPR, is the ratio of negative cases that have been classified as positive and can be estimated as

$$\text{FPR} = \frac{\text{FP}}{\text{TN} + \text{FP}} = 1 - S_p.$$

Its counterpart is the precision, p_r, that can be defined as the probability that an instance classified as positive is actually positive. From the confusion matrix, we compute it as

$$p_r = \frac{\text{TP}}{\text{TP} + \text{FP}}.$$

There are even more performance measures in the state of the art of classification in general. Of remarkable relevance could be the *area under the ROC curve* (AUC) (Spackman, 1989) and the *F measure* (van Rijsbergen, 1979; Goutte and Gaussier, 2005).

3.3.2 Estimation of the performance measures

For the aims of this dissertation, the classification error and/or accuracy are the chosen performance measures. In most occasions, this value is used to state which classifier is best suited to a certain problem among a set of classification paradigms. However, those statements are made on the basis of estimations, because the real error value is still unknown. Furthermore, accuracy estimation is usually performed with a (very) limited number of instances and that fact prevents a proper estimation.

The error rate of a classifier can be decomposed into two additive terms (Kohavi and Wolpert, 1996): the error *bias* and its *variance*. The error bias refers to the error due to the difference between the true error (as mentioned, impossible to know in real domains) and the estimated one by some of the methods that we will now discuss. The variance component of the error comes from the fact that, even if we assume that the bias value is very low, there will always be an intrinsic variance on the datasets (Rodríguez *et al.*, 2009).

The decomposition of the total error into its two components is an open issue in the classification field. Many authors defend that the error can be decomposed as

$$\epsilon_\gamma = \text{bias}^2 + \text{variance}.$$

Other studies suggest that the term is more a ratio than an addition,

$$\epsilon_\gamma = \text{bias}/\text{variance}.$$

Under the latter formulation (Friedman, 1997), the bias effect is of less importance in comparison to the variance term if and only if the class predicted by the classifier is correct. Notice that the bias contribution to the total error directly depends on what decomposition is chosen.

Besides the own error value, two unwanted side effects may arise in the estimation process. The first one is the *overfitting problem* related to the bias term. When a classifier is highly specialized towards the training set, that is, overfitted to that training set, the bias component of the error is expected to be large when new instances are presented to that classifier. The second effect is due to the number of instances on the test set(s). If the cardinality of these sets is low, the variance of the estimation tends to be high (Braga-Neto, 2005). Similarly to all estimations from a population sample, the variance tends to decrease as the number of cases increases.

Historically, three different ways to estimate the classification error have been used (Toussaint, 1974), namely *resubstitution* (Smith, 1947), *hold-out* (Larson, 1931; Wherry, 1931) and *k-fold cross-validation* (Hills, 1966; Cochran, 1968; Lachenbruch and Mickey, 1968).

- The resubstitution error (Smith, 1947) estimation is the simplest one and it consists of inducing a classification model with the full available dataset and testing its performance again on the same whole dataset. Roughly speaking, the training and test sets are the same one, which corresponds to the original dataset. Although the variance of the accuracy estimation is zero, this is not a desirable method. Resubstitution estimators present a high bias due to the overfitting and, thus, they provide accuracy estimations which are too optimistic.
- In order to obtain an honest or fair estimation, the classification model must be evaluated in a set of samples independent from the ones used to induce it. This is the idea behind the hold-out (Larson, 1931) estimation or *H estimator*: split the dataset into two disjoint sets, one to induce the model and the other one to estimate its performance. Usually, the proportion of instances in each set is 2/3 for the train and 1/3 for the test, although this proportion can be changed by the user. H estimator is well suited to problems in which there are a large number of instances. For problems with a low number of instances, since the test set is again smaller, the variance component of the error could be high (Horst, 1941).
- The k-fold cross-validation (Hills, 1966) (k-fold CV) constitutes a generalization of the H estimation. The dataset is divided into k randomly chosen and exclusive subsets. Iteratively, $k-1$ of these subsets are configured as the train set and the remaining one as the test set. The process is repeated k times, evaluating all possible combinations. At the end, the accuracy estimator is formed by the average of all folds' accuracies and its standard

deviation. The selection of the k value is a design decision that may vary the estimations in a great manner (Stone, 1974).

A direct improvement to both the H and the k-fold cross-validation estimators is to look for robustness by repeating the process several times. For the H estimator, it is called *random subsampling*, while for the latter, it is called *repeated k-fold cross-validation* (Kohavi, 1995). Both ways produce estimations with lower bias and variance. Another enrichment to obtain more realistic estimations was proposed by Breiman *et al.* (1984) and consists of obtaining the partitions while trying to keep the original proportion of classes. This constraint is usually denoted as *stratification* and in the case of the k-fold CV, it is usually taken as the standard way.

A particular case for the k-fold CV is the case where $k = N$. Within this configuration, known as *leaving-one-out cross-validation* or LOOCV (Mosteller and Tukey, 1968), the test set is always comprised of a single instance while the train sets are formed by all instances except that instance. LOOCV obtains almost unbiased accuracy estimations (Lachenbruch and Mickey, 1968) although it can return optimistic errors in some domains.

The above presented methods are, in general, accepted and widely used by the machine learning community, though they have also been criticized (Ng, 1997; Provost *et al.*, 1998; Nadeau and Bengio, 2003). As alternatives, other methods try to reduce the bias and variance of the estimators: *jackknife* (Rao and Shao, 1992), *bootstrap* (Efron, 1983) or *bolstered* estimator (Braga-Neto and Dougherty, 2004a).

3.4 Supervised feature selection

As introduced in Section 3.2, the main goal of supervised classification is to induce a classifier model that allows us to classify unseen examples $E^* = \{\boldsymbol{x}^{N+1}, \ldots, \boldsymbol{x}^{N+Q}\}$ that are described by the values of their n features or predictive variables. During the induction of the model, a total of N samples $E = \{\boldsymbol{x}^1, \ldots, \boldsymbol{x}^N\}$, belonging to m different classes $\Omega_C = \{c_1, \ldots, c_m\}$, are used and the model is expressed on the basis of the values of each corresponding n features $\boldsymbol{X} = \{X_1, \ldots, X_n\}$.

The feature selection or FS approach deals with the fact that, for many classification sets, the reduction in the number of features carries out a gain in the final accuracy of the induced classification model. Not only a better predictive accuracy can be achieved, but also an improvement in the comprehension of the model and a reduction in both the induction time and the cost of the data acquisition (Saeys *et al.*, 2007).

Parsimonious theory states that, in general, mathematical models with the smallest number of parameters are preferred, as each parameter introduced into the model adds some uncertainty to it. Following this aim, it has been already proven that the classification accuracy of supervised classification algorithms is not monotonic with respect to the addition of features (Liu and

Motoda, 2008). The predictive accuracy of such models could be lessened by irrelevant or redundant features. Therefore, the quest for a subset of variables $\boldsymbol{X'} \subset \boldsymbol{X}$ is in many occasions a must rather than an alternative task. Feature selection as a general approach deals with the problem of choosing some features given a classification problem. However, we also focus our attention on a more specific subtask: to choose a subset of relevant features from the original ones. This is usually known as feature subset selection or FSS and in many times both terms, feature selection and feature subset selection, are indistinctly used. Examples of feature selectors as just univariate relevance metrics are presented in Chapter 7 of this dissertation. In contrast, Chapter 8 introduces a feature subset selector and all its elements.

Feature subset selection is, basically, a search problem: each state in the search space corresponds to a different configuration for the subset $\boldsymbol{X'}$, and associated to each configuration there is a value of an objective function that measures the goodness of such a configuration. Exhaustive evaluation of all possible subset configurations is most of the times infeasible due to the NP-hard nature of the problem (Kohavi and John, 1997). Thus, the use of heuristic-based search approaches is usually suggested to deal with such searches. Let us present the main four pillars that every FSS approach needs to settle.

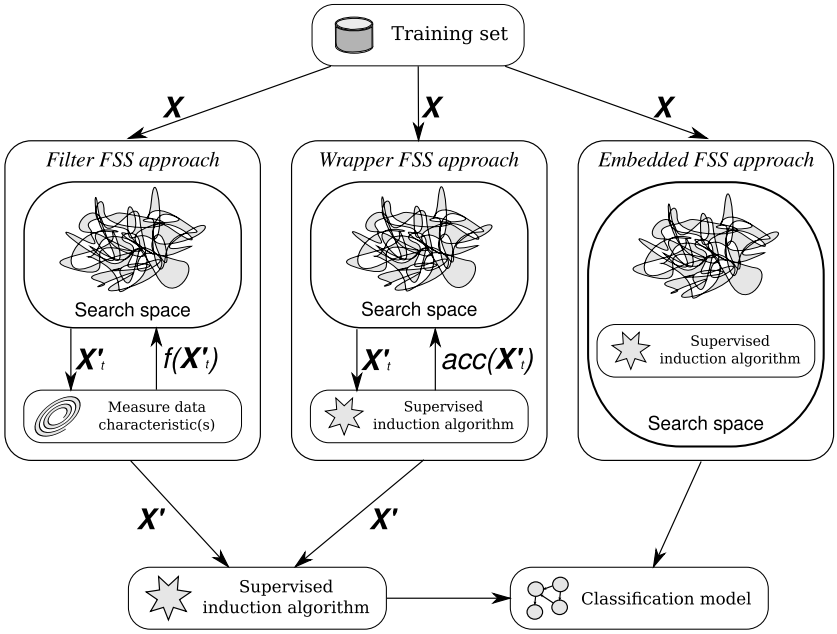


Fig. 3.1. Evaluation policies on a feature subset selection procedure. The subindex t in $\boldsymbol{X'}_t$ refers to the t-th iteration of the search.

A. Starting point

The search might begin with no features selected at all and successively add them. On the contrary, there could be a backward search policy, beginning with all the features and iteratively removing them while the objective function keeps improving. A middle alternative is to set up an initial set of selected features and then explore the nearby search space.

B. Organization of the search

Basically, we can divide this issue into *complete* or *heuristic*. Complete search will systematically examine every possible feature subset, while the heuristic procedures will take advantage of some auxiliar function in order to avoid exploring the whole search space. Heuristic algorithms are also divided into *deterministic* –sequential forward and backward selection, best-first search, ...– and *non-deterministic* –genetic algorithms, simulated annealing, ant colony optimization, etc.–.

C. Evaluation of feature subsets

There are three possible policies to evaluate a feature subset in the context of a classification problem, *filter*, *wrapper* and *embedded* methods.

Filter techniques (Ben-Bassat, 1982) assess the relevance of features by looking only at the intrinsic properties of the data. In most cases a feature relevance score is calculated, and low-scoring features are removed. Afterwards, this subset of features constitutes the input of the classification algorithm. Wrapper methods (Kohavi, 1995) embed the classification induction algorithm within the feature subset search. In this setup, the evaluation of a specific subset of features is obtained by training and testing a specific classification model, the final subset of features being only suited to that specific classification algorithm. The third kind of evaluation policy, known as embedded techniques, performs a search for an optimal subset of features at the same time the classifier model is built, and can be seen as a hybrid combination between the former two policies. Figure 3.1 graphically illustrates the behaviour of these three evaluation approaches.

D. Search stop criterion

This criterion is usually set by the user. Valid options could be the non-improvement of the evaluation function, setting up a limit on the number of possible solutions, or other options more particular for the search strategy in use.

A good review of general FSS methods can be found in (Liu and Motoda, 1998) and in (Liu and Motoda, 2008). In particular for the bioinformatic field, the review by (Saeys *et al.*, 2007) explores the applications of FSS into the computational biology field.

4

Bayesian networks for classification purposes

In this section we will introduce a type of probabilistic graphical model known as Bayesian networks (Pearl, 1988; Jensen and Nielsen, 2007; Neapolitan, 2003) that has been used during the last decade for reasoning in domains with an intrinsic uncertainty. Statistics and machine learning fields have developed different approaches to solve the supervised classification problem: classification trees (Breiman *et al.*, 1984), classifier systems (Holland, 1975), discriminant analysis (Fisher, 1936), k-NN classifiers (Cover and Hart, 1967), logistic regression (Hosmer and Lemeshow, 1989), neural networks (McCulloch and Pitts, 1943), rule induction (Clark and Niblett, 1989) and support vector machines (Cristianini and Shawe-Taylor, 2000) among others. Within these approaches, Bayesian networks represents one of the models that requires less effort by humans to produce an interpretation. They have a straight graphical representation allowing to observe and understand the underlying probabilistic classification process.

Starting by introducing the mathematical concepts that constitute the basics of probabilistic graphical models, the first part of this chapter is devoted to what Bayesian networks are and how to induce them from data. The second part of the chapter is devoted to introducing the Bayesian network classifiers that are used in the experimental parts of this dissertation.

4.1 Probabilistic graphical models

The probabilistic graphical models or PGMs theory is built from the roots of graph theory. Before the presentation of PGMs and Bayesian networks, a set of concepts from graph theory should be introduced. A reader interested in a more in-depth study of all the following concepts may check the work by Castillo *et al.* (1997).

4.1.1 Introductory concepts from graph theory

We define a graph $\boldsymbol{G}$ to be a pair $\boldsymbol{G} := (\boldsymbol{X}, \boldsymbol{L})$, where $\boldsymbol{X}$ is a finite set of vertices, also called nodes, of $\boldsymbol{G}$, and $\boldsymbol{L}$ is a subset of the set $\boldsymbol{X} \times \boldsymbol{X}$ of order pairs of vertices, called the edges or links of $\boldsymbol{G}$. As $\boldsymbol{L}$ is a set, the graph $\boldsymbol{G}$ has no multiple edges. We require that $\boldsymbol{L}$ consist of pairs of distinct vertices so that there are no loops.

If both ordered pairs (α, β) and (β, α) belong to $\boldsymbol{L}$, we say that we have an undirected edge between α and β, and write $\alpha \sim \beta$ (or $\alpha \sim_G \beta$ to indicate the relevant graph $\boldsymbol{G}$). We also say that α and β are neighbours, α is neigbour of β, or β is neigbour of α. The set of neighbours of a vertex β is denoted by $\boldsymbol{Ne}(\beta)$ or $\boldsymbol{Ne}_\beta$.

If $(\alpha, \beta) \in \boldsymbol{L}$ but $(\beta, \alpha) \notin \boldsymbol{L}$, we call the edge directed, and write $\alpha \to \beta$ ($\alpha \to_G \beta$). We also say that α is a parent of β, and that β is a child of α. The set of parents of a vertex β is denoted by $\boldsymbol{Pa}(\beta)$ or $\boldsymbol{Pa}_\beta$, and the set of children of a vertex α by $\boldsymbol{Ch}(\alpha)$ or $\boldsymbol{Ch}_\alpha$. The family of β, denoted as $\boldsymbol{Fa}(\beta)$ or $\boldsymbol{Fa}_\beta$, is $\boldsymbol{Fa}_\beta = \{\beta\} \cup \boldsymbol{Pa}_\beta$. If $(\alpha, \beta) \in \boldsymbol{L}$ or $(\beta, \alpha) \in \boldsymbol{L}$, we say that α and β are joined. Then $\alpha \nsim \beta$ indicates that α and β are not joined, i.e., both $(\alpha, \beta) \notin \boldsymbol{L}$ and $(\beta, \alpha) \notin \boldsymbol{L}$. We also write $\alpha \nrightarrow \beta$ if $(\alpha, \beta) \notin \boldsymbol{L}$. A graph is called complete if every pair of vertices is joined.

Given a subset $\boldsymbol{A}$ of $\boldsymbol{X}$, $\boldsymbol{A} \subset \boldsymbol{X}$, the expressions $\boldsymbol{Pa}_\mathbf{A}$, $\boldsymbol{Ch}_\mathbf{A}$ and $\boldsymbol{Ne}_\mathbf{A}$ will denote the collection of parents, children and neighbours, respectively, of the elements of $\boldsymbol{A}$, but exclude any element in $\boldsymbol{A}$.

When all the edges of a graph are directed, we say that it is a directed graph. Conversely, if all the edges of a graph are undirected, we say that it is an undirected graph. The undirected version $\boldsymbol{G}^\sim$ of a graph $\boldsymbol{G}$ is the undirected graph obtained by replacing the directed edges of $\boldsymbol{G}$ by undirected edges.

We call $\boldsymbol{G}_\mathbf{A} := (\boldsymbol{A}, \boldsymbol{L}_\mathbf{A})$ a subgraph of $\boldsymbol{G} := (\boldsymbol{X}, \boldsymbol{L})$ if $\boldsymbol{A} \subseteq \boldsymbol{X}$ and $\boldsymbol{L}_\mathbf{A} \subseteq \boldsymbol{L} \cap (\boldsymbol{A} \times \boldsymbol{A})$. Thus, it may contain the same vertex set but possibly fewer edges. If, in addition, $\boldsymbol{L}_\mathbf{A} = \boldsymbol{L} \cap (\boldsymbol{A} \times \boldsymbol{A})$, we say that $\boldsymbol{G}_\mathbf{A}$ is the subgraph of $\boldsymbol{G}$ induced by the vertex set $\boldsymbol{A}$.

A path of length r from α to β is a sequence $\alpha = \alpha_0, \ldots, \alpha_r = \beta$ of distinct vertices such that $(\alpha_{i-1}, \alpha_i) \in \boldsymbol{L}$ for all $i = 1, \ldots, r$. Thus, a path can never cross itself and movement along a path never goes against the directions of arrows. If the path of length r from α to β given by the sequence $\alpha = \alpha_0, \ldots, \alpha_r = \beta$ is such that for at least one $i \in \{1, \ldots, r\}$ there is a directed edge α_{i-1}, α_i, we say that the path is directed. We will write $\alpha \mapsto \beta$ if there is a path from α to β, and say that α leads to β.

An r-cycle is a path of length r with the modification that the end points are identical. Similarly a directed r-cycle is a directed path with the modification that the end points are identical. We say that a graph is acyclic if it does not possess any cycles. Consequently, a directed graph which is acyclic is called a directed acyclic graph or DAG.

It is always possible to well-order the nodes of a DAG by a linear ordering or numbering such that, if two nodes are connected, the edge points from the lower to the higher of the two nodes with respect to the ordering. Note that a DAG may not have a unique well-ordering. If a DAG is well-ordered, the predecessors of a node α, denoted by $\boldsymbol{Pr}(\alpha)$ or $\boldsymbol{Pr}_\alpha$, are those nodes that have a lower number than α.

Given a DAG, the set of vertices α such that $\alpha \to \beta$ but not $\beta \to \alpha$ is the set $\boldsymbol{An}(\beta)$ or $\boldsymbol{An}_\beta$ of the ancestors of β. The descendants of α, $\boldsymbol{De}(\alpha)$ or $\boldsymbol{De}_\alpha$, are the vertices β such that $\alpha \to \beta$ but not $\beta \to \alpha$. The ancestral set $\mathbf{A}$ of α is comprised of α and of all the ancestors of the vertices in $\mathbf{A}$. Roughly speaking, a subset of vertices $\mathbf{A}$ within a DAG is an ancestral set if, for every vertex in the set, all ancestors of that vertex are also in the set.

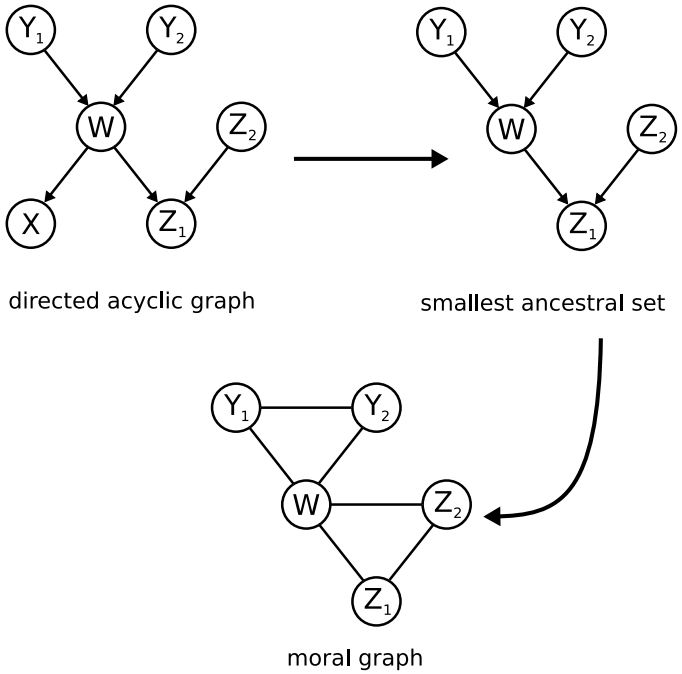


Fig. 4.1. Example of the U-separation definition. We want to check whether $\boldsymbol{W} = \{W\}$ D-separates $\boldsymbol{Y} = \{Y_1, Y_2\}$ and $\boldsymbol{Z} = \{Z_1, Z_2\}$ or not. First we have to obtain the smallest ancestral set of nodes containing $\boldsymbol{Y}$, $\boldsymbol{Z}$ and W, which is the original graph except the node X. Then we moralise the resulting graph and check whether W U-separates $\boldsymbol{Y}$ and $\boldsymbol{Z}$ in the undirected graph or not. As any path between Y_1 or Y_2 and Z_1 or Z_2 contains the node W we can say that W U-separates $\boldsymbol{Z}$ and $\boldsymbol{Y}$ in the moral graph of the smallest ancestral set and, thus, W D-separates $\boldsymbol{Z}$ and $\boldsymbol{Y}$ in the original DAG. Thus, we can state that, in the original DAG, $\boldsymbol{Z}$ and $\boldsymbol{Y}$ are conditionally independent given W.

Let $\boldsymbol{G}$ be a DAG, the moral graph associated to $\boldsymbol{G}$ is the graph obtained by adding undirected edges between all pairs of parents of each vertex which are not already joined, and then making all edges undirected. This process is called moralization of $\boldsymbol{G}$.

Finally, we present the two criteria that allow the theoretical analysis of a graph as a set of conditional (in)dependences, and viceversa. Those are the U-separation in undirected graphs and the U-separation in DAGs. Figure 4.1 shows the application of the U-separation definition through a graphical example.

Let $\boldsymbol{Y}$, $\boldsymbol{Z}$ and $\boldsymbol{W}$ be three disjoint subsets of vertices in an undirected graph $\boldsymbol{G}$. We say that $\boldsymbol{W}$ U-separates $\boldsymbol{Y}$ and $\boldsymbol{Z}$ in $\boldsymbol{G}$ iff every path between each node in $\boldsymbol{Y}$ and each node in $\boldsymbol{Z}$ contains, at least, one node in $\boldsymbol{W}$. Thus, subset $\boldsymbol{Y}$ will be graphically independent of $\boldsymbol{Z}$ given $\boldsymbol{W}$, if $\boldsymbol{W}$ U-separates $\boldsymbol{Y}$ and $\boldsymbol{Z}$.

Let $\boldsymbol{Y}$, $\boldsymbol{Z}$ and $\boldsymbol{W}$ be three disjoint subsets of nodes in a DAG $\boldsymbol{G}$. We say that $\boldsymbol{W}$ D-separates $\boldsymbol{Y}$ and $\boldsymbol{Z}$ in $\boldsymbol{G}$ iff $\boldsymbol{W}$ U-separates $\boldsymbol{Y}$ and $\boldsymbol{Z}$ in the moral graph of the smallest ancestral set containing $\boldsymbol{Y}$, $\boldsymbol{Z}$ and $\boldsymbol{W}$. Similarly to the previous criterion, subset $\boldsymbol{Y}$ will be graphically independent of $\boldsymbol{Z}$ given $\boldsymbol{W}$ on a DAG, if $\boldsymbol{W}$ D-separates $\boldsymbol{Y}$ and $\boldsymbol{Z}$.

4.1.2 Probabilistic graphical models based on directed acyclic graphs

Probabilistic graphical models or PGMs represent multivariate joint probability distributions, $\rho(\boldsymbol{x})$, via a product of terms, each of which involves only a few variables. The structure of this product is represented by a graph that relates variables that appear in a common term. This graph specifies the product form of the distribution and also provides tools for reasoning about the properties entailed by the product (Lauritzen and Spiegelhalter, 1988). For a sparse graph, the representation is compact and in many cases allows effective inference and learning.

PGMs based on DAGs make use of the concept of conditional independence to obtain the joint probability distribution.

Definition 1. *Let $\boldsymbol{Y}$, $\boldsymbol{Z}$ and $\boldsymbol{W}$ be three disjoint sets of random variables. $\boldsymbol{Y}$ is conditionally independent of $\boldsymbol{Z}$ given $\boldsymbol{W}$, $CI(\boldsymbol{Y}, \boldsymbol{Z}|\boldsymbol{W})$, iff*

$$\rho(\boldsymbol{y}|\boldsymbol{z}, \boldsymbol{w}) = \rho(\boldsymbol{y}|\boldsymbol{w}) \ ,$$

for any possible configuration $\boldsymbol{y}$, $\boldsymbol{z}$ and $\boldsymbol{w}$.

To parse the graphical independences to a probabilistic domain, we need to define what an I-map and a minimal I-map are:

Definition 2. *A graph $\boldsymbol{G}$ is known as an independence map or I-map from a model of dependences $\boldsymbol{M}$ if*

$$CI(\boldsymbol{Y}, \boldsymbol{Z}|\boldsymbol{W})_{\mathbf{G}} \Longrightarrow CI(\boldsymbol{Y}, \boldsymbol{Z}|\boldsymbol{W})_{\mathbf{M}} \ ,$$

that is, if all the conditional independences from $\boldsymbol{G}$ *are also verified by* $\boldsymbol{M}$*.*

Definition 3. *A graph* $\boldsymbol{G}$ *is a minimal I-map of a model of dependences* $\boldsymbol{M}$*, if* $\boldsymbol{G}$ *is an I-map of* $\boldsymbol{M}$*, but it loses this property when any of its edges is removed.*

By means of a DAG, $\boldsymbol{G} := (\boldsymbol{X}, \boldsymbol{L})$, we can represent the variables of a domain as vertices or nodes in $\boldsymbol{X}$. The graphical structure in $\boldsymbol{L}$ represents the graphical (in)dependences between triplets of variables. Therefore, given a DAG $\boldsymbol{G}$ and the D-separation criterion, we can gather all the graphical independences from the structure of $\boldsymbol{G}$. If $\boldsymbol{G}$ is a minimal I-map, all the graphical independeces correspond to probabilistic independences, and then we can parse the graphical independences to conditional independences and to the joint probability distribution.

The chain rule gives us the joint probability distribution of $\boldsymbol{X}$ as a product of factors of the form

$$\rho(\boldsymbol{x}) = \rho(x_1, \ldots, x_n) = \prod_{i=1}^{n} \rho(x_i|x_1, \ldots, x_{i-1}) \ .$$

The structure of the DAG on a PGM can be assumed to follow an ancestral ordering where each node X_i takes the i-th position in that ordering. Thus, for every ancestral node X_j of X_i, we can state that $j < i$.

Let $\boldsymbol{G} = (\boldsymbol{X}, \boldsymbol{L})$ be the DAG of a PGM that follows an ancestral ordering, the set of parents of a node X_i, $\boldsymbol{Pa_i}$, D-separates X_i from any previous node in the ancestral ordering. Consequently, X_i is conditionally independent of any X_j, with $j < i$, given its parents.

Finally, we can combine this property with the chain rule, and then induce the joint probability distribution encoded by $\boldsymbol{G}$ as

$$\rho(\boldsymbol{x}) = \prod_{i=1}^{n} \rho(x_i|\boldsymbol{pa_i}) \ .$$

The last element of PGMs is the set of parameters $\boldsymbol{\theta} \in \boldsymbol{\Theta}$ needed to fully describe the factorisation induced from the graphical structure. Thus, the factorisation should be rewritten taking into account this extra component:

$$\rho(\boldsymbol{x}) = \prod_{i=1}^{n} \rho(x_i|\boldsymbol{pa_i}, \boldsymbol{\theta}) \ .$$

The estimation of these numerical parameters depends on the probabilistic graphical model and the nature of its nodes. For the purposes of this dissertation, the estimation of $\boldsymbol{\theta}$ will be discussed in the following section.

4.1.3 Bayesian networks

Bayesian networks are probabilistic graphical models based on DAGs where the nodes are discrete random variables. A random variable $X_i \in \boldsymbol{X}$ represents a unidimensional discrete random variable with r_i possible states $\{x_i^1, \ldots, x_i^{r_i}\}$. On a Bayesian network, each variable X_i is associated with a conditional probability distribution $p(X_i = x_i \mid \boldsymbol{Pa}_i = \boldsymbol{pa}_i)$, where $\boldsymbol{Pa}_i \subset \mathbf{X}$ is the set of parents of X_i.

A Bayesian network is fully described by the pair of elements that constitutes any PGM: the graphical skeleton $\mathbf{G}$, given by a directed acyclic graph, and the set of parameters $\boldsymbol{\theta}$ that are associated to the local probability distribution of each variable X_i in $\mathbf{G}$. The joint probability distribution encoded by $\mathbf{G}$ follows the product expression,

$$p(\mathbf{x}) = p(x_1, \ldots, x_n) = \prod_{i=1}^{n} p(x_i \mid \boldsymbol{pa}_i) \; .$$

The set of parameters $\boldsymbol{\theta} = (\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_n)$ is given by $\boldsymbol{\theta}_i = (\theta_{ijk})$ where θ_{ijk} represents the conditional probability of X_i when taking its k-th state given that its parents set $\boldsymbol{Pa}_i$ takes its j-th configuration,

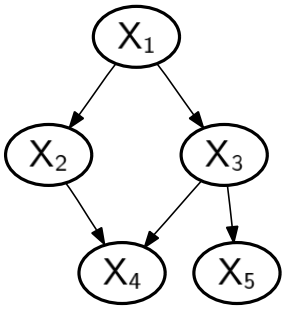
$$\theta_{ijk} = p(x_i^k | \boldsymbol{pa}_i^j) \; .$$

Notice that the set of parents may represent a multidimensional variable, and, thus, it may take $\prod_{X_g \in \boldsymbol{Pa}_i} r_g$ different configurations. Being conditional probabilities, they fulfill the Kolmogorov axioms, $\theta_{ijk} > 0$ and $\sum_{k=1}^{r_i} \theta_{ijk} = 1$.

As an example, Figure 4.2 contains a Bayesian network formed by five dichotomic variables $X_1, \ldots, X_5$. In Figure 4.2.a the corresponding DAG is displayed, while on the right side, Figure 4.2.b, we illustrate the list of parameters to be estimated. Notice that the rest of parameters are directly computed as the difference towards 1 for each variable. On the bottom, the joint probability factorization is addressed. In the case that all the $\boldsymbol{\theta}$ values were needed, the number of parameters would have reached 31, while with the factorisation given by the DAG we only have to assess 11 values.

Therefore, to assess a Bayesian network $\mathcal{B} = (G, \boldsymbol{\theta})$ it is necessary to specify:

- The structure by means of a directed acyclic graph which reflects the set of conditional (in)dependencies among the variables. Thus, the concept of conditional independence between triplets of variables is the semantic to understand and interpret the Bayesian network framework. Subsequently, the structure constitutes the qualitative part of the model.
- The unconditional probabilities for all *root nodes* –nodes with no predecessors– as well as the conditional probabilities for all other nodes, given all possible combinations of their direct predecessors. These unconditional and conditional probabilities constitute the quantitative part of the model.



a. Bayesian network structure

$p(\overline{x}_1) = 0.20$
$p(\overline{x}_2 \mid \overline{x}_1) = 0.80$
$p(\overline{x}_2 \mid x_1) = 0.80$
$p(\overline{x}_3 \mid \overline{x}_1) = 0.20$
$p(\overline{x}_3 \mid x_1) = 0.05$
$p(\overline{x}_4 \mid \overline{x}_2, \overline{x}_3) = 0.80$
$p(\overline{x}_4 \mid x_2, \overline{x}_3) = 0.80$
$p(\overline{x}_4 \mid \overline{x}_2, x_3) = 0.80$
$p(\overline{x}_4 \mid x_2, x_3) = 0.05$
$p(\overline{x}_5 \mid \overline{x}_3) = 0.80$
$p(\overline{x}_5 \mid x_3) = 0.40$

b. Parameters

Fig. 4.2. Achieved joint probability factorisation with the attached Bayesian network: $p(x_1, x_2, x_3, x_4, x_5) = p(x_1)p(x_2 \mid x_1)p(x_3 \mid x_1)p(x_4 \mid x_2, x_3)p(x_5 \mid x_3)$.

Once the Bayesian network is built, it constitutes an efficient device to perform probabilistic inference (Lauritzen and Spiegelhalter, 1988). It gives us the chance to assess a probability distribution over some variables of interest given evidence of the value of some other variables in the net. Nevertheless, the problem of building a Bayesian network remains. The structure and conditional probabilities necessary for characterising the Bayesian network can be provided either externally by experts –time consuming and subject to mistakes– or by automatic learning from a database of cases. However, the learning task can be separated into two subtasks: structure learning, that is, to identify the topology of the Bayesian network, and parametric learning. This second subtask is related with the estimation of the numerical parameters (conditional probabilities) for a given Bayesian network topology.

4.1.3.1 Learning Bayesian networks from data

It is common to classify the different approaches to Bayesian network model induction according to the nature of modelling in approaches based on i) the detection of conditional (in)dependencies between triplets of variables, and, ii) *score+search* methods.

The output for algorithms belonging to the first approach is a directed acyclic graph that represents a large percentage –and even all of them if possible– of these relations. Once the structure has been learnt, the conditional probability distributions required to completely specify the model are estimated from the database. See (Spirtes *et al.*, 1993) for more details about this approach to Bayesian networks modelling from data.

Although the approach to model elicitation based on detecting conditional (in)dependences is quite appealing, due to its closeness to the semantic of Bayesian networks, a big percentage of the developed structure learning algorithms belongs to the category of score+search methods. To use this learning

approach, we need to define a metric that measures the goodness of every candidate Bayesian network with respect to a data-file of cases. In addition, we also need a procedure to move in an intelligent way through the space of possible directed acyclic graphs. The most usual score metrics are penalised maximum likelihood, a Bayesian score known as marginal likelihood, and scores based on information theory. With respect to the search procedure there are a lot of different alternatives in the literature: greedy search, simulated annealing, genetic algorithms, tabu search, etc. For a review on score+search methods for learning Bayesian networks from data, the paper by Heckerman *et al.* (1995) can be consulted.

Throughout this dissertation, we will focus our attention on the first way to induce Bayesian networks, that is, to induce the pair $\mathcal{B} = (G, \boldsymbol{\theta})$ from data by means of the detection of conditional (in)dependences. This process constitutes a blind learning in the sense that no human previous knowledge biases the final outcome. The complexity degree of the graph structure will be discussed over the next Section 4.2, and we now illustrate how the parameters are learnt when a DAG structure is already given.

In order to estimate the associated parameters $\boldsymbol{\theta}$ given a network structure G, two assumptions should be made:

1. The dataset D contains no missing data.
2. The parameter vectors θ_{ij} are mutually independent, known as parameter independence (Spiegelhalter and Lauritzen, 1990).

There exist two broadly known approaches to estimate the parameter configuration under these assumptions: the maximum a posteriori or MAP estimation, and the maximum likelihood or ML.

The ML estimation maximizes the probability of the dataset given the model, that is,

$$\hat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} p(D|G, \boldsymbol{\theta}) \ ,$$

where $p(D|G, \boldsymbol{\theta})$ is the likelihood function

$$p(D|G, \boldsymbol{\theta}) = \prod_{d=1}^{N} p(\boldsymbol{x}^{(d)}|G, \boldsymbol{\theta}) \ .$$

The value of $\boldsymbol{x}^{(d)}$ is the value of $\boldsymbol{X}$ in the d-th sample of the dataset D. The maximization of this function outputs the ML parameters as

$$\hat{\theta}_{ijk} = \frac{N_{ijk}}{N_{ij}} \ .$$

This ML approach is the one used to assess the estimators in all the Bayesian network classifiers that are introduced in the next section. Bearing that in mind, the next section reviews how the Bayesian networks are adapted to a classification purpose and the resulting classification paradigms.

4.2 Bayesian network classifiers

The use of Bayesian network structures in classification tasks give rise to what is broadly known as Bayesian network classifiers. The majority of these classification paradigms assume that the classification variable is a parent of all the predictive variables, that is, the classification output conditionally depends on each predictive variable. From this prior condition, the relationships' structure among the predictive variables may lead from the simplest structure (no dependences) to an unrestricted Bayesian network structure.

Bayesian network classifiers are generative classifiers that graphically encode the joint probability distribution of the domain variables by means of a Bayesian network structure. Under this assumption, the estimation of the model parameters is straightforward computed from the maximum likelihood estimators.

There exists a wide spectrum of Bayesian network classifiers, however, this section is mostly devoted towards two of them that are of special interest to the developed work: the naïve Bayes classifier and the k-dependence Bayesian classifier. Nevertheless, a brief introduction to other members of this classification family is carried out.

4.2.1 Naïve Bayes classifier

The simplest Bayesian network classifier has its roots in the pattern recognition community (Duda and Hart, 1973). It first appearance in the machine learning literature took place in the 80's (Cestnik *et al.*, 1987) with the purpose of comparing its results against more sophisticated paradigms. Soon, its potential and robustness turned it into a gold standard within classification tasks. Many names refer to it, namely, idiot Bayes (Ohmann *et al.*, 1988), simple Bayes (Gammerman and Thatcher, 1991), independent Bayes (Todd and Stamper, 1994) or naïve Bayes (Minsky, 1961). Throughout this dissertation, the naïve Bayes denomination is used.

The pillars of the naïve Bayes classifier are two assumptions between the predictive variables (findings or symptoms) and the variable to predict (class or diagnosis):

1. the diagnoses are exclusive, that is, the class variable C can only take one of its m possible values: $\{c_1, \ldots, c_m\}$;
2. the findings are conditionally independent given the diagnosis. If the class value is known, the knowledge of whatever predictive variable is irrelevant to the remaining ones.

In Section 3.2, Equation 3.1 presents the joint distribution from which the observations of a classification problem are expected to come from. It is possible to expand the unknown term by means of the chain rule as

$$\begin{aligned} p(x_1, \ldots, x_n|c) = & \, p(x_1 \mid x_2, \ldots, x_n, c) \cdot \\ & \, p(x_2 \mid x_3, \ldots, x_n, c) \cdot \\ & \, \ldots \\ & \, p(x_n \mid c) \; . \end{aligned}$$

On the basis of the naïve Bayes assumption of conditional independence between the predictive variables and the class, we also have that

$$p(x_i \mid x_{i+1}, \ldots, x_n, c) = p(x_i \mid c) \; ,$$

for every $i \in \{1, \ldots, n\}$. Combining both terms, we obtain the general simplification of the naïve Bayes paradigm,

$$p(x_1, \ldots, x_n|c) = \prod_{i=1}^{n} p(x_i|c) \; . \tag{4.1}$$

Therefore, the search for the most probable diagnosis, c^*, once all the symptoms $\mathbf{x} = (x_1, \ldots, x_n)$ of a given instance are known, is reduced to find the value

$$c^* = arg \max_c p(c) \prod_{i=1}^{n} p(x_i|c) \; . \tag{4.2}$$

In terms of complexity, the number of parameters to be estimated for a naïve Bayes with discrete predictive variables is

$$(m-1) + \sum_{i=1}^{n} m(r_i - 1) \; ,$$

where r_i corresponds to the number of states the variable X_i can take. The first $m-1$ parameters are needed to specify the *a priori* probability of the class variable C and the rest correspond to the conditional probability distributions of the variables.

Since the number of states of the predictive variables could be variable, the cost of a discrete naïve Bayes is usually simplified to a dichotomic classification with all binary predictive variables, resulting in $2n + 1$ parameters.

In the case that the n predictive variables present continuous values, the naïve Bayes classifier looks for the value c^* that maximizes the *a posteriori* probability of the class variable C given an instance $\mathbf{x}$. The search c^* value is, thus, the one that verifies

$$c^* = arg \max_c p(c) \prod_{i=1}^{n} f_{X_i|c}(x_i|c) \; ,$$

where $f_{X_i|c}(x_i|c)$ represents, for each $i = 1, \ldots, n$, the density function of X_i conditioned to a c value for the class variable C.

In order to model the probability distribution of the predictive variables it is very common to use normal densities (John and Langley, 1995). For each c value of C, and for each $i \in \{1, \ldots, n\}$, we assume that

$$f_{X_i|c}(x_i|c) \rightsquigarrow \mathcal{N}(x_i; \mu_i^c, (\sigma_i^c)^2) \ .$$

The naïve Bayes classifier then searches the prediction value c^* as

$$c^* = arg \max_c p(c) \prod_{i=1}^{n} \left[\frac{1}{\sqrt{2\pi}\sigma_i^c} e^{\frac{-1}{2}\left(\frac{x_i - \mu_i^c}{\sigma_i^c}\right)^2} \right] \ .$$

The number of parameters to estimate in the continuous case is of $(m-1) + 2mn$, the class *a priori* $m-1$ estimations and $2mn$ due to the normal parameters conditioned to each class value.

In general, a mixture scenario can appear in which some predictive variables are continuous while others have a discrete number of states. In such cases, the paradigm is divided according to the nature of each variable. So, given a total of n predictive variables, n_1 of them will correspond to discrete variables, $X_1, \ldots, X_{n_1}$, while the rest $n_2 = n - n_1$, $Y_1, \ldots, Y_{n_2}$, will belong to the continuous domain. Bearing in mind the main naïve Bayes assumption, we can express the *a posteriori* probability of each class value given an instance as

$$p(c|x_1, \ldots, x_{n_1}, y_1, \ldots, y_{n_2}) \propto p(c) \prod_{i=1}^{n_1} p(x_i|c) \prod_{j=1}^{n_2} f_{Y_j|c}(y_j|c) \ .$$

Graphically, a naïve Bayes classifier can be displayed as the network structure of Figure 4.3.

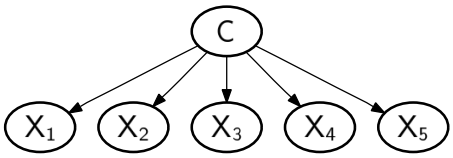


Fig. 4.3. Structure of a naïve Bayes model with five predictive variables.

Even when the root assumptions of the naïve Bayes are hardly ever fulfilled in real domains, in many cases this classifier obtains promising results and is competitive with the most sophisticated ones. Some successful applications include medical domains (Kononenko, 1990; Ohmann *et al.*, 1996; Mani *et al.*, 1997; Movellan *et al.*, 2002), web site classification according to user interests (Pazzani *et al.*, 1996), collaborative filter approaches (Miyahara and Pazzani, 2000), text classification (McCallum and Nigam, 1998) or failure detection (Hamerly and Elkan, 2001).

4.2.2 Advances on the naïve structure

Despite its good performance, the basic naïve Bayes presents hard limitations in terms of the representation of dependences. Although very robust against irrelevant features, it is very sensitive to highly correlated features. Thus, and in order to overcome these concerns and extend the dependences representation possibilities, a set of naïve augmented models has arisen in the last years.

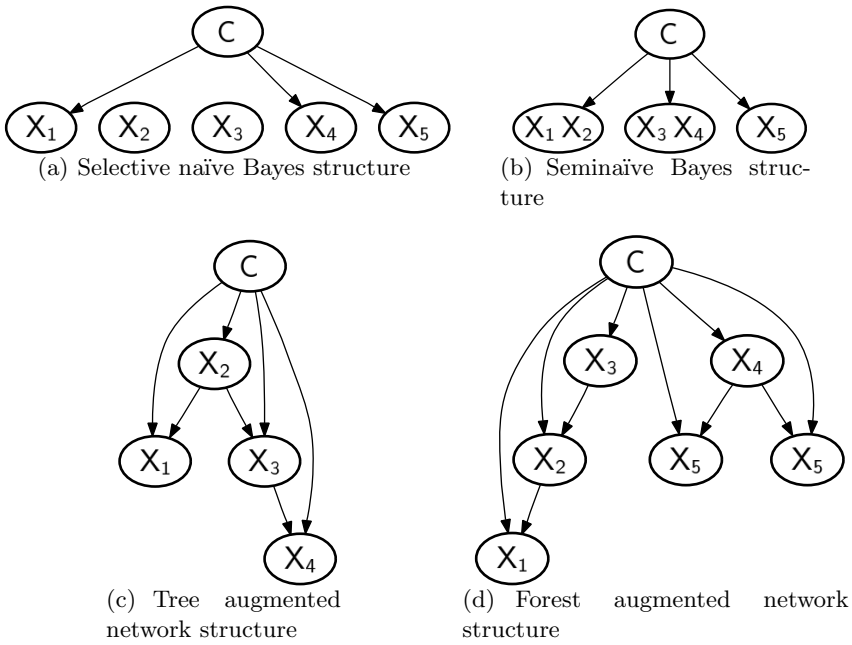
The most direct improvement to the basic scheme is to perform, beforehand, a feature subset selection on the original variables. After removing redundant and/or irrelevant features, the original naïve structure is used. This first advance was introduced by (Langley and Sage, 1994) and is known as selective naïve Bayes. Originally, Langley and Sage (1994) proposes a wrapper feature selection process with a greedy forward search procedure. However, any feature subset selection scheme may suit the selection of a variables' subset (see Section 3.4).

A more ambitious advance is considered in (Kononenko, 1991) and (Pazzani, 1997). Formally known as seminaïve Bayes, both authors propose the Cartesian product of variables in order to create new ones that smooth the independence assumption of the original model. In Kononenko (1991) the variables are joined removing the original ones, while Pazzani (1997) proposes two algorithms –namely forward sequential selection and joining FSSJ, and, backward sequential elimination and joining BSEJ– to remove irrelevant variables and combine relevant ones.

But, the most widely known improvement to the classical naïve Bayes is the tree augmented naïve Bayes classifier or TAN. Firstly introduced by Friedman *et al.* (1997), the main idea of this paradigm is to, first, build a dependence tree-like structure among the variables and, then, connect all the predictive variables with the class one. In this way, conditional dependences between the variables are explicitly captured. The algorithm presented in (Friedman *et al.*, 1997) is basically an adaptation of the (Chow and Liu, 1968) algorithm. While this algorithm is based on the joint mutual information between two variables, the TAN algorithm includes dependences with respect to the value of the conditional mutual information between two given predictive variables and the value of the class. Both TAN and Chow and Liu algorithms fulfill an important theoretical property: the asymptotically correction. That is, having enough instances coming from a real tree-dependence structure, they are able to perfectly recover such a structure from the sample.

Nevertheless, the restrictions can be still quite strong. One of the posterior attempts to create more flexible models is the forest augmented network algorithm FAN (Lucas, 2004), in which the dependences are represented by a forest of tree structures rather than a single tree structure. For both TAN and FAN their hard restriction lays on the fact that the predictive variables are only allowed to have up to one parent (excluding the class variable). This re-

striction will be overcome by the Bayesian paradigm presented in the following section.



(a) Selective naïve Bayes structure

(b) Seminaïve Bayes structure

(c) Tree augmented network structure

(d) Forest augmented network structure

Fig. 4.4. Examples of the structure of different Bayesian network classifiers.

Figure 4.4 visually illustrates all the above introduced improvements to the naïve Bayes basic classifier. For a more detailed review, both historical and technical, of all of these improvements to the naïve Bayes, the reader may check the work by Blanco (2005).

4.2.3 *k*-dependence Bayesian classifier

Sahami (Sahami, 1996) presents an algorithm called k-dependence Bayesian classifier (kDB) that allows to go through the wide spectrum from the naïve Bayes to a complete Bayesian network. The algorithm has its basis in a naïve Bayes structure that allows each predictive variable to have a maximum number of k parent variables (excluding the class one).

The simple naïve Bayes classifier corresponds to the 0-dependence Bayesian classifier, the TAN model would be the 1-dependence and the complete Bayesian classifier –structure where there is no independence– would correspond to a $(n-1)$-dependence Bayesian classifier. The kDB induction pseudocode is presented in Figure 4.5.

Step 1. For each predictive variable X_i, $i = 1, \ldots, n$, compute the mutual information with respect to the class variable C, $I(X_i, C)$
Step 2. For each pair of predictive variables, compute the mutual information conditioned to the class, $I(X_i, X_j|C)$, with $i < j$ and $i, j = 1, \ldots, n$
Step 3. Initialize to empty the list of used variables $\aleph$
Step 4. Initialize the Bayesian network classifier to build, BN, to a single node, the one corresponding to the C variable
Step 5. Repeat until $\aleph$ includes all the variables
  Step 5.1. Choose among the variables not included in $\aleph$, that variable X_{max} with highest mutual information with respect to C
  Step 5.2. Add X_{max} into BN
  Step 5.3. Add an arc from C to X_{max} in BN
  Step 5.4. Add $m = \min(|\aleph|, k)$ arcs from the m different variables X_j of $\aleph$ that have the highest values for $I(X_{max}, X_j|C)$
  Step 5.5. Add X_{max} into $\aleph$
Step 6. Compute the conditional probabilities needed to specify the Bayesian network classifier BN

Fig. 4.5. kDB algorithm pseudocode (Sahami, 1996).

The main idea of this algorithm is to extend the algorithm proposed by Friedman et al. (Friedman *et al.*, 1997) allowing a variable to have a number of parents, excluding the class variable C, bounded by k. This k parameter will allow the expert to vary the sparsity degree of the results, focusing on single interactions or on more complex ones. As in the TAN model, the mutual information conditioned to the class variable is used to decide which edges are included and in which order. Its value is computed through the expression,

$$I(X, Y|C) = \sum_{i=1}^{v} \sum_{j=1}^{w} \sum_{r=1}^{m} p(x_i, y_j, c_r) \log \frac{p(x_i, y_j|c_r)}{p(x_i|c_r)p(y_j|c_r)} ,$$

being X and Y two discrete predictive variables conditioned to the class variable C. Figure 4.6 shows an example of how the kDB induction algorithm builds the network structure with a set of five predictive variables.

Sahami also introduces a modification in Step 5.4 of the algorithm. The variant, named kDB-θ, does not consider all the possible parent's set bounded by the k value, it only includes those dependences which surpass a given threshold θ within the conditional mutual information $I(X_{max}, X_j|C)$. The main drawback for this kDB variation is the determination of the θ value.

As of our knowledge, the conditional mutual information can be formulated as

$$I(X_i, X_j|C) = \sum_k p(c_k) I(X_i, X_j|C = c_k) ,$$

where each of the $I(X_i, X_j|C = c_k)$ terms was proven to follow, under the null hypothesis of independence of X_i and X_j variables when $C = c_k$, a

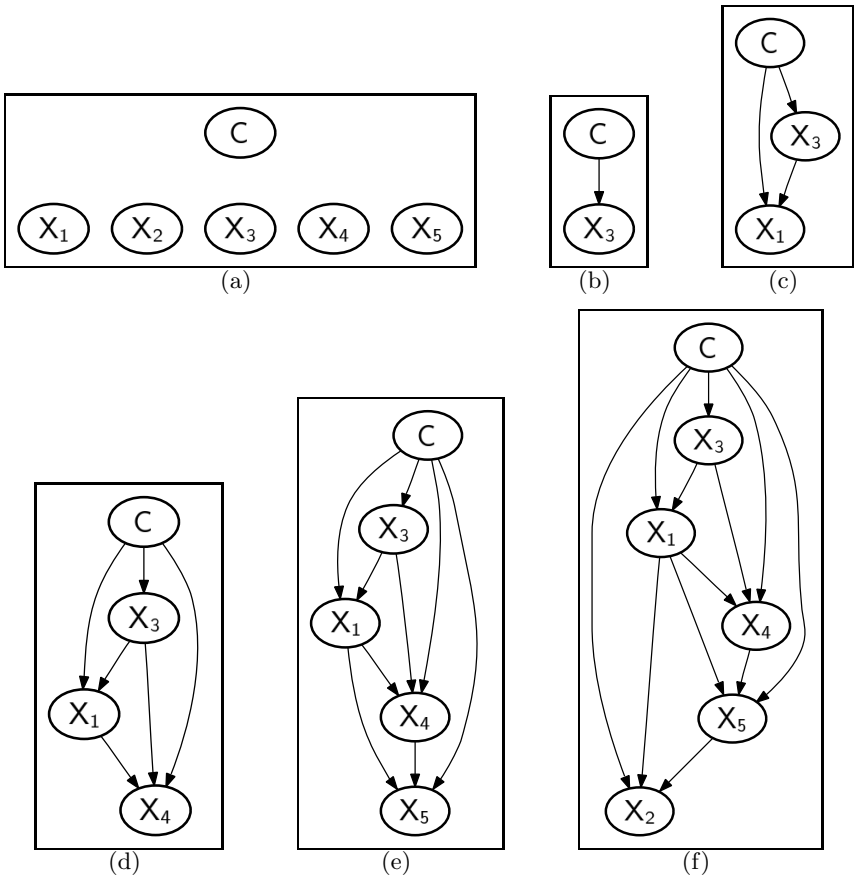


Fig. 4.6. Structural learning of a kDB model with a k value of 2. $I(X_3, C) > I(X_1, C) > I(X_4, C) > I(X_5, C) > I(X_2, C)$ $I(X_3, X_4|C) > I(X_2, X_5|C) > I(X_1, X_3|C) > I(X_1, X_2|C) > I(X_2, X_4|C) > I(X_2, X_3|C) > I(X_1, X_4|C) > I(X_4, X_5|C) > I(X_1, X_5|C) > I(X_3, X_5|C)$. The joint probability is then expressed as: $p(c|x_1, x_2, x_3, x_4, x_5) \propto p(c)p(x_1|x_3, c)p(x_2|x_1, x_5, c)p(x_3|c)p(x_4|x_1, x_3, c)p(x_5|x_1, x_4, c)$.

$\chi^2_{(r_i-1)(r_j-1)}$ statistical distribution (Blanco *et al.*, 2005). Unfortunately, the distribution for the joint conditional $I(X_i, X_j|C)$ is unknown under the same null hypothesis. This fact makes it impossible to fix beforehand a θ value for a given confidence level by means of statistical tests.

Computing a kDB network structure requires $O(n^2Nmv^2)$, where n is the number of variables, N is the number of cases, m is the number of classes and v is the maximum number of discrete values a predictor variable may take. For the conditional probability tables of the network structure, the algorithm takes $O(n(N + v^k))$ time. In most cases v and k are small values, thus, the

computing time for the network parameters scales linearly with N, the amount of data available.

There exist other Bayesian classifiers based on more generic paradigms, such as the Bayesian network augmented naïve Bayes (Cheng and Greiner, 2001) or the full Bayesian network (Jensen and Nielsen, 2007). However, for the aims of the present work and due to the intrinsic characteristics that the biological data usually present (course of dimensionality), we consider these other paradigms out of our scope. The reader can find more details and applications of the Bayesian classifiers in the work by Santafé (2008).

5

Estimation of distribution algorithms

Estimation of distribution algorithms (EDAs) are a novel class of evolutionary optimization algorithms that were developed as a natural alternative to genetic algorithms in the last decade. The principal advantages of EDAs over genetic algorithms are the absence of multiple parameters to be tuned (e.g. crossover and mutation probabilities) and the expressiveness and transparency of the probabilistic model that guides the search process. In addition, EDAs have been proven to be better suited to some applications than GAs, while achieving competitive and robust results in the majority of tackled problems.

5.1 EDA basics

Estimation of distribution algorithms (Bosman and Thierens, 1999; Larrañaga and Lozano, 2002; Lozano *et al.*, 2006; Mühlenbein and Paaß, 1996; Pelikan, 2005) are evolutionary algorithms that work with a multiset (or population sets) of candidate solutions (points). Figure 5.1 illustrates the flow chart for any EDA approach.

Set $t \leftarrow 0$. Generate M points randomly
do
Evaluate the points using the fitness function
Select a set S of $N \leq M$ points according to a selection method
Estimate a probabilistic model for S
Generate M new points sampling the distribution represented in the model
$t \leftarrow t + 1$
until Termination criteria are met

Table 5.1. Estimation of distribution algorithms: evolutionary computation based on learning and simulation of probabilistic graphical models.

Initially, a random sample of points is generated. These points are evaluated using an objective function. An objective function evaluates how accurate each solution is for the problem. Based on this evaluation, a subset of points is selected. Hence, points with better function values have a bigger chance of being selected.

Then, a probabilistic model of the selected solutions is built, and a new set of points is sampled from the model. The process is iterated until the optimum has been found or another termination criterion is fulfilled.

For more details, Table 5.1 sets out the pseudocode that implements a basic EDA. There is a complete running example of an EDA in (Larrañaga, 2002).

Essentially EDAs assume that it is possible to build a model of the promising areas of the search space, and use this model to guide the search for the optimum. In EDAs, modeling is achieved by building a probabilistic graphical model that represents a condensed representation of the features shared by the selected solutions. Such a model can capture different patterns of interactions between subsets of the problem variables, and can conveniently use this knowledge to sample new solutions.

Probabilistic modeling gives EDAs an advantage over other evolutionary algorithms that do not employ models, such as GAs. These algorithms are generally unable to deal with problems where there are important interactions among the problems' components. This, together with EDAs' capacity to solve different types of problems in a robust and scalable manner (Lozano *et al.*, 2006; Pelikan, 2005), has led to EDAs sometimes also being referred to as competent GAs (Goldberg, 2002; Pelikan *et al.*, 2002).

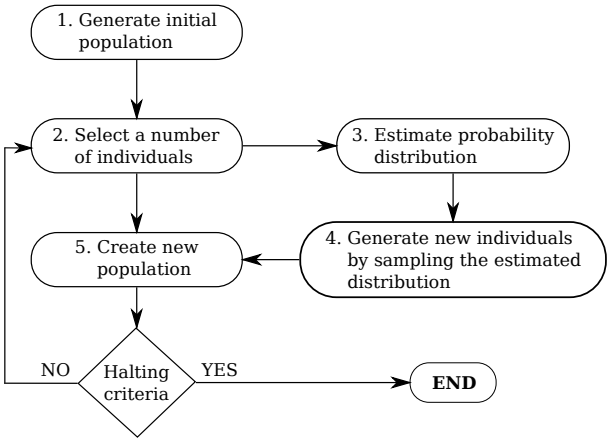


Fig. 5.1. Diagram of how an estimation of distribution algorithm works. This overview of the algorithm is further specified by the pseudocode shown in Table 5.1.

5.2 A taxonomy of EDAs

Since several EDAs have been proposed with a variety of models and learning algorithms, the selection of the best EDA to deal with a given optimization problem is not always straightforward. One criterion that could be followed in this choice is to trade off the complexity of the probabilistic model against the computational cost of storing and learning the selected model. Both issues are also related to the problem dimensionality (i.e. number of variables) and to the type of representation (e.g. discrete, continuous, mixed).

Researchers should be aware that simple models generally have minimal storage requirements, and are easy to learn. However, they have a limited capacity to represent higher-order interactions. On the other hand, more complex models, which are able to represent more involved relationships, may require sophisticated data structures and costly learning algorithms. The impact that the choice between simple and more complex models has in the search efficiency will depend on the addressed optimization problem. In some cases, a simple model can help to reach non-optimal but acceptable solutions in a short time. In other situations, e.g. deceptive problems, an EDA that uses a simple model could move the search away from the area of promising solutions.

Another criterion that should be taken into consideration to choose an EDA is whether there is any previous knowledge about the problem structure, and which kind of probabilistic model is best suited to represent this knowledge. The following classification of EDAs is intended to help the bioinformatic researcher to find a suitable algorithm for his or her application.

EDAs can be broadly divided according to the complexity of the probabilistic models used to capture the interdependencies between the variables: univariate, bivariate or multivariate approaches. Univariate EDAs, such as PBIL (Baluja, 1994), cGA (Harik *et al.*, 1999) and UMDA (Mühlenbein and Paaß, 1996), assume that all variables are independent and factorize the joint probability of the selected points as a product of univariate marginal probabilities. Consequently, these algorithms are the simplest EDAs and have also been applied to problems with continuous representation (Sebag and Ducoulombier, 1998).

The bivariate models can represent low order dependencies between the variables and be learnt using fast algorithms. MIMIC (De Bonet *et al.*, 1997), the bivariate marginal distribution algorithm BMDA (Pelikan and Mühlenbein, 1999), dependency tree-based EDAs (Baluja and Davies, 1997) and the tree-based estimation of distribution algorithm (Tree-EDA) (Santana *et al.*, 1999) are all members of this subclass. The latter two use tree and forest-based factorizations, respectively. They are recommended for problems with a high cardinality of the variables and where interactions are known to play an important role. Trees and forests can also be combined to represent higher-order interactions using models based on mixtures of distributions (Santana *et al.*, 1999).

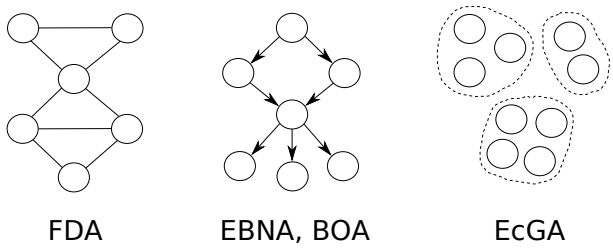


Fig. 5.2. Diagram of probability models for the proposed EDAs in combinatorial optimization with multiple dependencies (FDA, EBNA, BOA, and EcGA).

Multivariate EDAs factorize the joint probability distribution using statistics of order greater than two. Figure 5.2 shows some of the different probabilistic graphical models covered by this category. As the number of dependencies among the variables is higher than in the above categories, the complexity of the probabilistic structure, as well as the computational effort required to find the structure that best suits the selected points, is greater. Therefore, these approaches require a more complex learning process. Some of the EDA approaches based on multiply connected Bayesian networks are:

- The (Factorized Distribution Algorithm) FDA (Mühlenbein *et al.*, 1999) is applied to additively decomposed functions for which, using the running intersection property, a factorization of the mass-probability based on residuals and separators is obtained.
- In (Etxeberria and Larrañaga, 1999), a factorization of the joint probability distribution encoded by a Bayesian network is learnt from the selected set in every generation. The estimation of Bayesian network algorithm (EBNA) uses the Bayesian information criterion (BIC) score as the quality measure for the Bayesian network structure. The space of models is searched using a greedy algorithm.
- The Bayesian optimization algorithm (BOA) (Pelikan *et al.*, 1999) is also based on the use of Bayesian networks. The Bayesian Dirichlet equivalent metric is drawn on to measure the goodness of every structure. The algorithm enacts a greedy search procedure. BOA has been improved by adding dependency trees and restricted tournament replacement. The resulting, more advanced, hierarchical BOA (hBOA) (Pelikan, 2005) is one of the EDAs for which extensive experimentation has been undertaken. The results show good scalability behavior.
- The extended compact Genetic Algorithm (EcGA) proposed in (Harik *et al.*, 1999) is an algorithm in which the basic idea is to factorize the joint probability distribution as a product of marginal distributions of different size.

There are alternatives to the use of Bayesian networks for representing higher order interactions in EDAs. Markov network-based EDAs (Alden, 2007;

Shakya and McCall, 2007; Santana, 2005) could be an appropriate choice for applications where the structure of the optimization problem is known and can be easily represented using an undirected graphical model. EDAs that use dependency networks (Gámez *et al.*, 2007) can encode dependencies that Bayesian networks cannot represent. Both classes of algorithms need relatively complex sampling procedures based on the use of Gibbs sampling (Geman and Geman, 1984).

In addition to the order of complexity encoded by the probability model, there is another key feature when dealing with an EDA algorithm: the way that model is learned. There are two alternatives: induce the model structure and its associated parameters, or induce just the set of parameters for *an a priori* given model. The first class is denoted as *structure+parameter learning*, whereas the second is known as *parameter learning*. Both approaches need to induce the parameters of their models, but the first approach's need for structural learning makes it more time consuming. By contrast, parameter learning is dependent on the fixed model, whereas structure+parameter learning exhibits a greater power of generalization.

Population-based incremental learning (PBIL) (Baluja, 1994), the compact GA (cGA) (Harik *et al.*, 1999), the univariate marginal distribution algorithm (UMDA) (Mühlenbein and Paaß, 1996) and the factorized distribution algorithm (FDA) (Mühlenbein *et al.*, 1999) which use a fixed model of interactions in all generations, are all parameter approaches. On the other hand, the mutual information maximization for input clustering algorithm (MIMIC) (De Bonet *et al.*, 1997), the extended compact GA (EcGA) (Harik *et al.*, 1999) and EDAs that use Bayesian and Gaussian networks (Etxeberria and Larrañaga, 1999; Mühlenbein and Mahnig, 2001; Ochoa *et al.*, 2000b,a; Pelikan, 2005; Pelikan and Mühlenbein, 1999) belong to the structural+parameter class.

So as to have a graphical taxonomy of the subdivisions presented through this section, Table 5.2 illustrates all the above features and models providing a graphical taxonomy of the subdivisions presented throughout this section. It also includes some useful tips to choose among the available EDAs, such as their pros and cons.

5.3 Estimation of distribution algorithms as feature selectors

One big drawback of many classical search strategies for FSS is their inability to explore different regions of the search space rather than those in which the initialization processes have set up. Stochastic policies outperform this problem by their random components. A population-based search includes an initial random population that can be completely different from one run to another. Moreover, these techniques are able to perform jumps on the search space to unveil new solutions. On the other hand, the user should be aware

Statistical order	Advantages	Disadvantages	Examples
Univariate	Simplest and fastest	Ignore feature dependencies	PBIL (Baluja, 1994)
	Suited for high cardinality problems	Bad performance for deceptive problems	UMDA (Mühlenbein and Paaß, 1996)
	Scalable		cGA (Harik *et al.*, 1999)
Bivariate (statistics of order two)	Able to represent low order dependencies	Possibly ignore some feature dependencies	MIMIC (De Bonet *et al.*, 1996)
	Suited for many problems	Slower than univariate EDAs	Dependency trees EDA (Baluja and Davies, 1997)
	Graphically inquire the induced models		BMDA (Pelikan and Mühlenbein, 1999)
			Tree-EDA / Mixture of distributions EDA (Santana *et al.*, 1999)
Multivariate (statistics of order greater than two)	Parameter learning (*only interaction model parameters*)		
	Suited for problems with known underlying model	Possibly ignore complex feature dependencies	FDA (Mühlenbein *et al.*, 1999)
		Greater memory requirements than bivariate	Markov network-based EDA (Shakya and McCall, 2007)
	Structure+parameter learning (*interaction model & parameters*)		
	Maximum power of generalization	Greatest computation time	EcGA (Harik *et al.*, 1999)
	Flexibility to introduce user dependencies	Greatest memory requirements	EBNA (Etxeberria and Larrañaga, 1999)
	Online study of the induced dependencies		BOA / hBOA (Pelikan *et al.*, 1999, 2005)
			Dependency networks EDA (Gámez *et al.*, 2007)

Table 5.2. A taxonomy of some representative EDAs. We highlight a set of characteristics that can guide the choice of a particular EDA suited to the goals and properties of a given problem.

that the optimum set of features can not be always retrieved, although at least a good approximation can.

In this scenario, the adaptation of EDAs to work as feature subset selectors is straight forward. Each individual will be formed by a binary array of size $n =\mid X \mid$, which corresponds to the total number of features of the problem. Thus, each individual is in its own a subset of features, those that are set to a *true* value are selected while the *false* ones are not. With this genotype codification we let the EDA evolve until it outputs its best solution, that is, a set of true selected features, thus, the feature subset selected.

Once the codification is defined, we shall set up the five main components required on a FSS problem:

1. The starting point. Here the starting point is comprised of the initial or first population. This population is usually random generated by sampling a Bernoulli distribution with p parameter set to 0.5. The value of p can be used to initially shift a population from a very sparse to a more dense one.
2. Individuals' evaluation. Here the EDAs follow a wrapper approach. Given a dataset, it is divided into the corresponding training and test sets and the goodness of each individual is measured in terms of its estimate predictive accuracy. The classification algorithm used is a design decision. In principle, there is no limitation to any classifier, but light induction algorithms are more desirable because of the low learning complexity. Notice that a validation should be performed for each individual in the population.
3. Search policy. From their own nature, the search policy of an EDA on a FSS problem will be heuristic and non-deterministic. The main behaviour of the search is somehow determined by what kind of probabilistic distribution is estimated for each population (see Section 5.2).
4. Stop criteria. The stop criterion is always to achieve a perfect classification, 100% in the accuracy estimation for the best individual. But this is not always the case, so other stop criteria should be added to avoid stacking on a deadlock, for instance, to reach a fixed number of generations.

By setting all these elements, EDAs constitute a good option to tackle the FSS domain. Nevertheless, it is crucial to evaluate the complexity that the algorithm could have beforehand, both in terms of computing time and memory space. The evaluation step costs $M \times k \times f(n)$, where k is the number of folds to evaluate an individual and $f(n)$ is the cost of learning each fold classification model. To this product, we have to add the learning cost of the distribution model for each population $h(n, M)$. And the total cost is always in function of the number of generations the search evolves, g.

In one of the simplest cases, with an UMDA distribution estimator and a naïve Bayes model for a dichotomic classification, $f(n) = 2n + 1$ (see Section 4.2.1) and $h(n, M) = n \times M$. In the worst case, the whole running cost could reach a value of $Mng(2k + k/n + 1)$, whose asymptotically order is

$\Theta(Mng)$. In most occasions, $M \geqslant g \geqslant n$, and then the minimum cost is upper bounded by $\Theta(n^3)$. In terms of memory space, this scheme linearly scales with the larger number of individuals or features (space cost of $M + n$).

Consequently, the UMDA scheme is one of the most used EDAs paradigms, not only over FSS problems but also on many other optimization problems (Larrañaga and Lozano, 2002; Lozano *et al.*, 2006). The next section will introduce what is the state of the art of all the EDAs paradigms in the bioinformatics field. It provides a review of each application and the modifications needed to tackle the computational biology field.

5.4 EDAs in bioinformatics

Due to advances in modern high-throughput biotechnology devices, large and high-dimensional data sets are obtained from analyzed genomes and tissues. The heuristic scheme provided by EDAs has proved to be effective and efficient in a variety of NP-hard genomic problems. Because of the huge cardinality of the solution spaces of most of these problems, researchers are aware of the need for an efficient optimization algorithm. In this way, authors have preferred simple EDA schemes that assume that the variables are independent. These schemes have obtained accurate and robust solutions in reasonable CPU times. Together with a brief definition of each tackled genomic problem, we describe the main characteristics of each EDA scheme, with a special emphasis on the codification used to represent the search individuals.

5.4.1 Applications in genomics

5.4.1.1 Gene structure analysis

As genomes are being sequenced at an increasing pace, the need for automatic procedures for annotating new genomes is becoming more and more important. A first and important step in the annotation of a new genome is the location of the genes in the genome, as well as their correct structure. As a gene may contain many different parts, the problem of gene structure prediction can be seen as a segmentation or parsing problem. To solve this problem automatically, pattern recognition and machine learning techniques are often used to build a model of what a gene looks like. This model can then be used to automatically locate potential genes in a genome (Mathé *et al.*, 2002; Majoros, 2007).

A gene prediction framework consists of different components, where each component (often modeled as a classifier) aims at identifying a particular structural element of the gene. Important structural elements include the start of the gene (start codon), the end of a gene (stop codon) and the transitions between the coding and non-coding parts of the gene (splice sites).

The exact mechanisms that the cell uses to recognize genes and their structural elements are still under research. As this knowledge is missing, one major problem in this context is to define adequate features to train the classifiers for each structural element. Consequently, large sets of sequence features are extracted in the hope that these sets will contain the key features. However, it is known that not all of these features will be important for the classification task at hand, and many will be irrelevant or redundant.

To find the most relevant features for recognizing gene structural elements, feature subset selection (FSS) techniques can be used. These techniques try to select a subset of relevant features from the original set of features (Liu and Yu, 2005; Saeys *et al.*, 2007). As this is an NP-hard optimization problem with 2^n possible subsets for evaluation (given n features), population-based heuristic search methods are an interesting engine for driving the search through the space of possible feature subsets. Each solution in the population decodes a feature subset as a binary string: features having a value of 1 are included in the subset, whereas those having a value of 0 are discarded.

As a natural alternative to genetic algorithms, the use of EDAs for FSS was initiated in (Inza *et al.*, 1999) for classic benchmark problems, and their use in large scale feature subset selection domains was reported to yield good results (Inza *et al.*, 2001; Saeys *et al.*, 2003). Furthermore, the EDA-based approach to FSS was shown to generalize to feature weighting, ranking and selection (Saeys *et al.*, 2006). This has the advantage of getting more insight into the relevance of each feature separately, focusing on strongly relevant, weakly relevant, and irrelevant features.

The application of EDA-based FSS techniques in gene structure prediction was pioneered for the most important gene prediction components in (Saeys, 2004). Its most important application was the recognition of splice sites. Using naïve Bayes classifiers, support vector machines and C4.5 decision trees as base classifiers, an UMDA-based FSS scheme was used to obtain higher performance models.

In addition to better models, an UMDA-based approach was also used to get more insight into the selected features. This led to both the identification of new characteristics, as well as the confirmation of important previously known characteristics (Saeys *et al.*, 2004).

5.4.1.2 Gene expression data by means of DNA microarrays

The quantitative and qualitative DNA analysis is one of the most important areas of modern biomedical research. DNA microarrays can simultaneously measure the expression level or activity level of thousands of genes under a set of conditions. Microarray technology has become a popular option for partial DNA analysis since (Golub *et al.*, 1999)'s pioneering work.

The starting point of the following applications is the so called gene expression matrix, where rows represent genes, columns represent experimental

conditions (or samples), and the values at each position of the matrix characterize the expression level of the particular gene under the particular experimental condition. Additional biological information about the genes and the experimental conditions can be added to the matrix in the form of gene and/or sample annotation. Depending on how we treat the annotation, gene expression data analysis can be either supervised or unsupervised. When sample annotation is used to split the set of samples into two or more classes or phenotypes (e.g. healthy or diseased tissues), supervised analysis (or class prediction) tries to find patterns that are characteristic of each of the classes. On the other hand, unsupervised analysis (or class discovery) ignores any annotation. Examples of such analysis are gene clustering, sample clustering and gene expression data biclustering.

A. Classification of DNA microarray data

It is broadly assumed that a limited number of genes can cause the onset of a disease. Within this scenario biologists demand a reduction in the number of genes. In addition, the application of a FSS technique to microarray datasets is an essential step to achieve an accurate classification performance for any base classifier.

Although univariate gene ranking procedures are very popular for differential gene expression detection, the multivariate selection of a subset of relevant and non-redundant genes has borrowed from the field of heuristic search engines to guide the exploration of the huge solution space (there are 2^n possible gene subsets, where n is the number of initial genes). Two research groups have proven that the EDA paradigm is useful for this challenging problem. Both groups have implemented efficient algorithms that have achieved accuracy levels comparable to the most effective state-of-the-art optimization techniques:

- Using a naïve Bayes network as the base classifier and the UMDA as the search algorithm, (Blanco *et al.*, 2004) achieve competitive results in two gene expression benchmarking datasets. The authors show that the predictive power of the models can be improved when the probability of each gene being selected in the first population is initialized using the results provided by a set of simple sequential search procedures.
- (Paul and Iba, 2004, 2005) propose two variations of the PBIL search algorithm to identify subsets of relevant and non-redundant genes. Using a wide variety of classifiers, notable results are achieved in a set of gene expression benchmarking datasets with subsets of extremely low dimensionality.

Using a continuous-value version of the UMDA procedure, EDAs have been used as a new way of regularizing the logistic regression model for microarray classification problems (Bielza *et al.*, 2008). Regularization consists of shrinking the parameter estimates to avoid their instability when there are

a huge number of variables compared to a small number of observations (as in the microarray setting).

Therefore, the parameter estimators are restricted maximum likelihood estimates, i.e. the maximum value of a new function including the likelihood function, plus a penalty term where the size of the estimators is constrained. There are different norms for measuring estimators size. This leads to different regularized logistic regression names (Hastie *et al.*, 2001): ridge, Lasso, bridge, elastic net, etc.

EDAs could be used to optimize these new functions and be a good optimization method especially in some cases where numerical methods are unable to solve the corresponding non-differentiable and non-convex optimization problems. However, another possibility, taken up in (Bielza *et al.*, 2008), is to use EDAs to maximize the likelihood function without having to be penalized (which is a simpler optimization problem) and to include the shrinkage of the estimates during the simulation of the new population. New estimates are simulated during EDA evolutionary process in such a way that guarantees their shrinkage while maintaining their probabilistic dependence relationships learnt in the previous step. This procedure yields regularized estimates at the end of the process.

B. Clustering of DNA microarray data

Whereas the above papers propose a supervised classification framework, clustering is one of the main tools used to analyze gene expression data obtained from microarray experiments (Ben-Dor *et al.*, 1999). Grouping together genes with the same behaviour across samples, that is, gene clusters, can suggest new functions for all or some of the grouped genes. We highlight two papers that use EDAs in the context of gene expression profile clustering:

- (Peña *et al.*, 2004) present an application of EDAs for identifying clusters of genes with similar expression profiles across samples using unsupervised Bayesian networks. The technique is based on an UMDA procedure that works in conjunction with the EM clustering algorithm. To evaluate the proposed method, synthetic and real data are analyzed. The experimentation with both types of data provides clusters of genes that may be biologically meaningful and, thus, interesting for biologists to research further.
- (Cano *et al.*, 2006) use UMDA and genetic algorithms to look for clusters of genes with high variance across samples. A real microarray dataset is analyzed, and the Gene Ontology Term Finder is used to evaluate the biological meaning of the resulting clusters.

Like clustering, biclustering is another NP-hard problem that was originally considered by (Morgan and Sonquistz, 1963). Biclustering is founded in the fact that not all the genes of a given cluster should be grouped into the same conditions due to their varying biological activity. Thus, biclustering

assumes that several genes will only change their expression levels within a specified subset of conditions (Cheng and Church, 2000). This assumption has motivated the development of specific algorithms for biclustering analysis.

An example is the work by (Palacios *et al.*, 2006), which applies an UMDA scheme to search the possible bicluster space. They get accurate results compared to genetic algorithms when seeking single biclusters with coherent evolutions of gene expression values. Like the classic codification discussed for the FSS problem, the authors use two concatenated binary arrays to represent a bicluster, $(x_1, ..., x_n \mid y_1, ..., y_m)$. The first array represents each gene of the microarray, where the size is the number of genes. The second array represents each condition, with a size equal to the number of conditions. A value of 1 in the i^{th} position of the first array shows that the i^{th} gene has been selected for inclusion in the bicluster. Likewise, a value of 1 in the j^{th} position of the second array indicates that the j^{th} condition has been selected for inclusion in the bicluster. This codification results in a space of 2^{n+m} possible biclusters.

5.4.1.3 Inference of genetic networks

The inference of gene-gene interactions from gene expression data is a powerful tool for understanding the system behaviour of living organisms (Armañanzas *et al.*, 2008a).

This promising research area is now of much interest for biomedical practitioners, and a few papers have applied EDAs to this domain. One of these early works uses Bayesian networks as the paradigm for modeling the interactions among genes, while an UMDA approach explores the search space to find the candidate interactions (Dai and Liu, 2005). The subsequent literature evaluation of the most reliable interactions unveils that many of them have been previously reported in the literature.

5.4.2 Protein structure prediction and protein design

The objective of protein structure prediction is to predict the native structure of a protein from its sequence. In protein design, the goal is to create new proteins that satisfy some given structural or functional constraints. Frequently, both problems are addressed using function optimization. As the possible solution space is usually huge, complex and contains many local optima, heuristic optimization methods are needed. The efficiency of the optimization algorithm plays a crucial role in the process. In this section, we review applications of EDAs to different variants of protein structure prediction and protein design problems.

Protein structure prediction and protein design are usually addressed by minimizing an energy function in the candidate solution space. Two essential issues in the application of EDAs and other optimization algorithms to these problems are the type of protein representation employed and the energy function of choice.

There are many factors that influence the stability of proteins and have to be taken into account to evaluate candidate structures. The native state is thought to be at the global free energy minimum of the protein. Electrostatic interactions, including hydrogen bonds, van der Waals interactions, intrinsic propensities of the amino acids to take up certain structures, hydrophobic interactions and conformational entropy contribute to free energy. Determining to what extent the function can represent all of these factors, as well as how to weight each one are difficult questions that have to be solved before applying the optimization method.

Simplified protein models omit some of these factors and are a first problem-solving approximation. For example, the approximate fold of a protein is influenced by the sequence of hydrophobic and hydrophilic residues, irrespective of what the actual amino acids in that sequence are (Steipe, 1998). Therefore, a first approximation could simply be constructed by a binary patterning of hydrophobic and hydrophilic residues to match the periodicity of secondary structural elements. Simplification can be further developed to consider proteins represented using this binary patterning and to approximate the protein structure prediction problem as two- and three-dimensional lattices. In this case, the energy function measures only hydrophobic and hydrophilic interactions. An example of this type of representation is shown in Figure 3, where a sequence of 64 aminoacids is represented on a two-dimensional lattice.

5.4.2.1 EDA approaches

Depending on how sophisticated and detailed the protein model used is, EDAs can be divided into two groups: EDAs applying a simplified model (Bacardit *et al.*, 2007; Santana *et al.*, 2004; Santana, 2006; Santana *et al.*, 2008b) and EDAs using more detailed (atomic-based) models (Belda *et al.*, 2005; Santana *et al.*, 2007b, 2008a). A more thorough classification is related to the type of problems addressed:

- Protein structure prediction in simplified models (Santana *et al.*, 2004, 2008b).
- Protein side chain placement (Santana *et al.*, 2007b, 2008a).
- Design of protein peptide ligands (Belda *et al.*, 2005).
- Protein design by minimization of contact potentials (Santana, 2006; Santana *et al.*, 2007a).
- Aminoacid alphabet reduction for protein structure prediction (Bacardit *et al.*, 2007).
- Using EDAs as a simulation tool to investigate the influence of different protein features in the protein folding process (Santana *et al.*, 2008a).

In (Santana *et al.*, 2004; Santana, 2006; Santana *et al.*, 2008b), EDAs are used to solve bi-dimensional and three-dimensional simplified protein folding problems. The hydrophobic-polar (HP) (Dill, 1985), and functional protein

models (Hirst, 1999) are optimized using EDAs based on probabilistic models of different complexity (i.e. Tree-EDA (Santana *et al.*, 2001), mixtures of trees EDA (MT-EDA) (Santana *et al.*, 2001) and EDAs that use k-order Markov models (MK-EDA$_k$) (Santana *et al.*, 2004)).

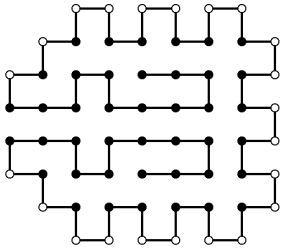


Fig. 5.3. Optimal solution of an HP model found by an EDA that uses a Markovian model.

The results achieved outperform other evolutionary algorithms. For example, the configuration shown in Figure 5.3 is the optimal solution found by MK-EDA$_2$. Due to the particular topology of this instance, other evolutionary algorithms consistently fail to find the optimal solution (Santana *et al.*, 2004).

Side chain placement problems are dealt with using UMDA with discrete representation in (Santana *et al.*, 2007b, 2008a). The approach is based on the use of rotamer libraries that can represent the side chain configurations using their rotamer angles. For these problems, EDAs have achieved very good results in situations where other methods fail (Santana *et al.*, 2008a). Results are better when EDAs are combined with local optimization methods as in (Santana *et al.*, 2008a), where variable neighborhood search (Mladenović, 1995) is applied to the best solutions found by UMDA.

(Belda *et al.*, 2005) use different EDAs to generate potential peptide ligands of a given protein by minimizing the docking energy between the candidate peptide ligand and a user-defined area of the target protein surface. The results of the population based incremental learning algorithm (PBIL) and the Bayesian optimization algorithm (BOA) are compared with two different types of genetic algorithms. Results showed that some of the ligands designed using the computational methods had better docking energies than peptides designed using a purely chemical knowledge-based approach (Belda *et al.*, 2005).

In (Santana *et al.*, 2007a), three different EDAs are applied to solve a protein design problem by minimizing contact potentials: UMDA, Tree-EDA and Tree-EDA^r (the structure of the tree is deduced from the known protein structure, tree parameters are learned from data). Combining probabilistic models able to represent probabilistic dependencies with information about residue interactions in the protein contact graph is shown to improve the

search efficiency for the evaluated problems. In (Santana, 2006), EDAs that use loopy probabilistic models are combined with inference-based optimization algorithms to deal with the same problems. For several protein instances, this approach manages to improve the results obtained with tree-based EDAs.

The alphabet reduction problem is addressed in (Bacardit *et al.*, 2007) using the extended compact genetic algorithm (EcGA). The problem is to reduce the 20-letter amino acid (AA) alphabet into a lower cardinality alphabet. A genetics-based machine learning technique uses the reduced alphabet to induce rules for protein structure prediction features. The results showed that it is possible to reduce the size of the alphabet used for prediction from twenty to just three letters resulting in more compact rules.

Results of using EDAs and the HP model to simulate the protein folding process are presented in (Santana *et al.*, 2007a). Some of the features exhibited by the EDA model that mimics the behaviour of the protein folding process are investigated. The features considered include the correlation between the EDA success rate and the contact order of the protein models, and the relationship between the generation convergence of EDAs for the HP model and the contact order of the optimal solution. Other issues analyzed are the differences in the rate of formation of native contacts during EDA evolution, and how these differences are associated with the contact separation of the protein instance.

5.5 Summary

Throughout this chapter, the estimation of distribution algorithms have been put on stage. Sections 5.1 presents the basics of these kind of evolutionary algorithms and Section 5.2 divides them in terms of the complexity degree of the dependences they are able to deal with. In Section 5.3 the feature subset selection or FSS is explained, as well as, how EDAs could be configured to be a feature subset selector. Finally, Section 5.4 makes an in depth review of the works currently available in the bioinformatics field which use EDAs to solve or, at least, tackle them.

Part II

Consensus course in computational biology knowledge discovery

6

Introduction

6.1 The curse of dimensionality

The *curse of dimensionality* is a term coined by (Bellman, 1961) to describe the problem caused by the exponential increase in volume associated with adding extra dimensions to a mathematical space. The curse of dimensionality can directly affect many fields such as, for example, problems in statistics, economics, optimization and machine learning.

In our case, machine learning problems that involve learning an unknown distribution from a finite (low) number of data samples in a high-dimensional feature search space are very often not affordable. This is due to the sparsity effect produced by the dimensionality problem: the amount of data to sustain a certain spatial density increases exponentially with the dimensionality of the input space, or alternatively, the sparsity increases exponentially given a constant amount of data, with points tending to become equidistant from one another.

Formally, a problem presents the curse of dimensionality when its associated data matrix Δ, with a dimension of $n \times m$, complies with the inequality $n \gg m$, where n is number of features and m the number of cases included in the problem.

Within the computational biology field the curse of dimensionality is often present. In fact, all the applications presented throughout this thesis belong to this class of problems. As a general rule, all the recent high-throughput biological devices retrieve datasets with this singularity. These devices are able to measure a huge amount of features from a given sample, but the numbers of samples at hand is always very low. In particular to gene expression data or mass spectrometry problems, the dimensionality problem is exemplified by a number of features, n, of order $10^3 - 10^5$ while the number of samples, m, is of order $10^1 - 10^2$.

The direct solution to this problem is to have a proportional number of samples, but, in practice, this solution is unfeasible. The amount of observations (samples) should be enormous to obtain good estimations in the machine

learning approach. The reader may think of the number of cancer or rare diseases patients in a hospital compared to few thousand genes to measure.

Nevertheless, there is light in such a dim scenario (Donoho, 2000). One positive characteristic is known as the *concentration of measure*. Roughly speaking, this formulation states that many of the data come from constant distributions on most of the space, so there are possibilities of correct inference in this kind of data.

The second positive characteristic is related to the former and it is known as *dimension asymptotics*. When the data dimension goes to infinity, the distributions converge to some limiting distribution. In many cases, it becomes possible to obtain predictions that work for moderate dimensions but which are derived by using the limiting distributions.

The third point is when the data is a sampled version of a continuous phenomenon, namely *approach to continuum*. Since what is measured is continuous, the space of observed data will show signs of compactness that can be exploited.

6.2 The theory of consensus

The first and usually very effective way of dealing with high-dimensional data is to reduce the number of dimensions. This is principally done by removing some dimensions that seem irrelevant to the problem. However, this removal process must be carefully done because the limited number of cases could lead to deceitful relevance statements.

As an initial approach, we propose this relevance determination by means of a set of decisions rather than relying only on a particular one. This making decision approach belongs to the *consensus theory* which is defined both as a *general agreement* and as *the process of getting* to such agreement.

Consensus is a general policy applied to many questions: politics, philosophy, polling, intelligence, engineering and, of course, computing. The original consensus term comes from an ancient criterion of truth, *consensus gentium* in Latin, which states *that which is universal among men carries the weight of truth* (Ferm, 1962).

The Boolean algebra also comes across consensus with two algebraic theorems:

$$(a \vee b) \wedge (b \vee c) \wedge (\neg a \vee c) \equiv (a \vee b) \wedge (\neg a \vee c) \ ,$$

$$(a \wedge b) \vee (b \wedge c) \vee (\neg a \wedge c) \equiv (a \wedge b) \vee (\neg a \wedge c) \ .$$

The term which is left out is called the consensus term. The logical simplification of this term states that given a pair of terms for which a variable appears in one term, and its complement in the other, then the consensus term is formed by combining the original terms together, leaving out the selected variable and its complement. These theorems algebraically expose how

it is possible to reduce the data when there is redundance in it and obtain the same outcome.

A mathematical description of consensus is to say that there is an iterative process through $(d + e)$-dimensional parameter space, starting from initial guesses at a solution in d-dimensional parameter space, which tries to converge to find a common solution in $(d + e)$-dimensional parameter space.

The machine learning approach must face three important issues when dealing with the curse of dimensionality: affordability, results' variance and overfitted models. There exist basic techniques to tackle such problems:

- Feature selection - As the first approach to the curse of dimensionality, the practitioner can make use of a dimensionality reduction by using some feature selection approach. This is not only restricted to feature subset selections but also to ranking metrics, as will be introduced in the next chapter.
- Bootstrap approaches - When the dataset is comprised of a very low number of instances, a way to alleviate the possible overfitted and variable results is to perform repeated resampling of the dataset and repetitions of the analysis techniques. A bootstrap approach will reduce the posibility of obtaining statistical artifacts when relying on repeteated and stable findings (Friedman *et al.*, 1999).
- Classifier combination - Following the same policy as with the bootstrap, a combination of different or similar classification paradigms can bring stability to the results. There is a full research field within the machine learning in this issue (Kuncheva, 2004). In addition, the combination of classifiers may significantly improve the prediction accuracy of a full classification system (e.g. automatic medical systems in the help of diagnosis/prognosis).

Through all the methodological proposals included in this part, we confront the former problematic issues by means of consensus adaptations of these and other classical machine learning techniques. The aim of such consensus approaches is always add more reliability, robustness and generalization, in order not to get trapped by the dimensionality problematic. Notice that all the consensus proposals included are designed to deal with specific computational biology problems. Despite this fact, the proposals are introduced in a general way as far as it is possible and some benchmark results couple their formulations. For a full application of all these techniques in challenging real computational biology problems, the reader can consult Part III of the dissertation.

7

Consensus over univariate ranking metrics

The impressive growth that the biological/bioinformatics datasets have undergone through the last decade is an important challenge to the data mining discipline. When a practitioner in this field tackles a new problem, one of the first questions is: –*What is the importance of this gene, protein, sequence or entity in my problem?*

This question can be quickly addressed through the statistics using different relevance metrics. These metrics belong to the *filter* approach and are very fast in their computation, so, they are perfectly suited to having initial idea about the features under evaluation.

Through this chapter, we introduce a set of relevance metrics to measure such relevance. All of them are designed to deal with supervised classification problems. Looking for more robustness in the final output, we propose a way to combine a set of univariate metrics into a single consensus decision. This approach could be of special interest when dealing with problems of very low number of cases. It is possible to find more relevance metrics in the state-of-the-art literature (Saeys *et al.*, 2007) appart from those included in this chapter.

7.1 Univariate relevance metrics

Within the filter approaches to feature selection (see Section 3.4 for details), the simplest and probably most extended approach is to measure the goodness or relevance of a feature in the dataset that is under study. Since this measurement is performed individually, it only takes into account the values that the feature takes. These univariate metrics output a coefficient that quantifies the degree of relevance that a feature has in the problem. Almost all of these metrics are formulated for the supervised classification field and its measurement is directly related to the separability that each feature has in the classification problem.

If we consider all the feature set of a particular problem, each feature coefficient can be seen as a punctuation or merit and therefore it is possible to sort all of them in a relevance *ranking*. In this way, a ranking of feature importance is computed according to a particular relevance metric.

All the metrics presented in this chapter have their formulations based on heuristics derived from divergence measures between data distribution functions. The coefficients retrieved for each feature constitute the ranking value for which the features are sorted. Univariate metrics assign higher coefficients to the most relevant features. However, there are metrics based on minimization, that is, the lower the coefficient is, the more relevant the feature is. One of the most important characteristics of the univariate metrics is their speed in terms of computational time. Since the evaluation is individual and there is no classification paradigm to learn, the complexity order is nearly always linear or close to linear.

Another important issue is that their formulation is not based on any *a priori* data distribution assumption, such as Gaussian or Poisson distribution. All these relevance metrics are thus categorized into the non-parametrical statistics, a fact that makes them ideal when the number of instances available is low or very low. In these cases, such as in many bioinformatic problems, other parametrical relevance techniques are forced to test the parametrical assumptions (Gaussan distribution and heterocedasticity), assumptions that are not easily fulfilled (Jafari and Azuaje, 2006).

Based on the work by (Ben-Bassat, 1982), we adapt here seven different univariate filter metrics. Most of them are originally proposed for dichotomic problems. In these cases, we extend its use to multiclass problems by weighting the dichotomic metric in function of the marginal probability of each class. For computing the global coefficient for the multiclass problem, the marginal dichotomic coefficients are computed and added as

$$Filter_{multiclass} = \sum_{i=1}^{r_c} \sum_{j=1}^{j<i} p(c_i)p(c_j)Filter(c_i, c_j) \ ,$$

where $Filter(x, y)$ is the original dichotomic metric and r_c is the number of classes or states for the class variable.

Notice that all the following metrics expect discrete values as input. If the problem under consideration includes continuous values (e.g. gene expressions or PCR values), all those values need to be discretized into a number of finite states. In the following Chapter 8, Section 8.2 is exclusively devoted to introducing possible discretization approaches. We refer the reader to that section for a more detailed revision.

A. Mutual information

One of the most known and widely used relevance metrics is the mutual information (Shannon, 1948). Based on the information theory, mutual information

computes the relation that exists between a pair of variables. Its formulation measures how much uncertainty is unveiled by the knowledge of a variable about the possible state that the second variable could have. This metric ranges from 0 to 1 where values close to 1 imply a high correlation between the variables, whereas values close to 0 point to independence between them.

As a ranking metric, the variables to compare are, first, the variable under evaluation and, second, the supervised variable. Its formulation is

$$I(X,C) = \sum_{i=1}^{r_x} \sum_{j=1}^{r_c} p(x_i, c_j) \log \frac{p(x_i, c_j)}{p(x_i)p(c_j)} \ ,$$

where X is the variable to evaluate, C is the class variable, and r_x and r_c are, respectively, the number of states that both variables can take.

Mutual information presents a disadvantage when comparing its values for different pairs of variables. Due to its formulation, the mutual information metric score benefits those variables with a large number of states. Therefore, when the number of states of two variables is not equal, the direct comparison of their relevance in terms of mutual information is unfair.

B. Matusita metric

The original Matusita distance (Matusita, 1955) is intended for measuring the distance between two probability distributions. On the following adaptation, we measure the average distance between the marginal distributions of each variable values and the values from the class. Using the same notation as in the previous case, its mathematical expression is

$$MA(X,C) = \sum_{i=1}^{r_c} \sum_{j=1}^{j<i} p(c_i)p(c_j) \left[\sum_{k=1}^{r_x} \sqrt{p(x_k|c_i)p(x_k|c_j)} \right] \ .$$

C. Kullback-Leibler divergence

The Kullback-Leibler divergence (Kullback, 1987) is the most known technique to measure the difference between two probability distributions $P(X)$ and $Q(X)$, taking one of them as reference. The general formulation is

$$KL\left(P(X), Q(X)\right) = \sum_{x_i} p(x_i) \log \frac{p(x_i)}{q(x_i)} \ .$$

Two different approaches are proposed to instantiate the probability distributions. First, compare the a priori marginal probabilities (mode 1). And, secondly, compare the a priori conditional probabilities. Remember that since this is a dichotomic metric we have to initially weight each class marginal probability as

$$KL(X,C) = \sum_{i=1}^{r_c} \sum_{j=1}^{j<i} p(c_i)p(c_j)\ KL_{ij}(X,C)_a\ ,$$

where $a = \{1,2\}$ is the two possible modes. The mode 1 divergence is formulated as

$$KL_{ij}(X,C)_1 = KL\left(P(X|c_i), P(X)\right) + KL\left(P(X|c_j), P(X)\right)\ ,$$

and, in mode 2, as

$$KL_{ij}(X,C)_2 = KL\left(P(X|c_i), P(X|c_j)\right) + KL\left(P(X|c_j), P(X|c_i)\right)\ .$$

D. Shannon entropy

Shannon's entropy (Shannon, 1948) is one of the most extended metrics to measure the goodness of a given variable. Its dichotomic formulation is adapted to the multiclass case as

$$SH(X,C) = \sum_{i=1}^{r_c} \sum_{j=1}^{j<i} p(c_i)p(c_j)\ H_{ij}(X)\ ,$$

where

$$H_{ij}(X) = -\sum_{k=1}^{r_x} p(x_k|c_i)\log_2 p(x_k|c_j) + p(x_k|c_j)\log_2 p(x_k|c_i)\ .$$

E. Bhattacharyya metric

This metric (Bhattacharyya, 1943) measures the degree of dependence between two probability distributions. We are going to compare the a priori probability of a variable versus the probability conditioned to the class value. The higher this degree is, the more important the variable should be for the classification problem. Its formulation is then

$$Bh(X,C) = \sum_{i=1}^{r_c} -\log\left[p(c_i)\sum_{j=1}^{r_x}\sqrt{p(x_j|c_i)p(x_j)}\right]\ .$$

F. Euclidean distance

The last metric does not derive from any information analysis or probabilistic function. It comes from the definition of Euclidean distance in a n-dimensional space. Let $P = (p_1, \ldots, p_n)$ and $Q = (q_1, \ldots, q_n)$ two points of an Euclidean n-space, the Euclidean distance between them is defined as

$$ED(P,Q) = \sqrt{\sum_{i=1}^{n}(p_i - q_i)^2} \ .$$

We propose a multiclass metric based on this distance as a contrast to the six previous metrics based on probabilistic theories. The metric is formulated as

$$ED(X,C) = \sqrt{\sum_{i=1}^{r_x}\sum_{j=1}^{r_c}\sum_{k=1}^{k<j} p(c_k)p(c_j) \ \mid p(x_i|c_k)^2 - p(x_i|c_j)^2 \mid} \ .$$

7.2 Positional consensus

As previously exposed for the curse of dimensionality, it is common to find datasets in computational biology that contain a limited number of instances, whereas a large number of features or variables are present. In these scenarios, the main drawback for the univariate metrics is the high variance associated to the data values. This variance reflects the fact that the data distribution is scattered and, as a consequence, the relevance rankings could vary significantly if we compute them through different metrics.

The first approach to the consensus is thus to make a *positional consensus* (Armañanzas *et al.*, 2009a). This consensus is a way to not only rely on one particular metric but to allow the evaluation of all of them. Once all the rankings are computed, we calculate the *average* ranking for all the metrics. This average ranking will be the single output ranking, instead of returning many similar but different ones.

These kinds of problems were firstly described by (Condorcet, 1785) in the context of voting and distribution of seats. In fact, the problem can be reformulated as an aggregation of individual preferences (Kemeny, 1959) and solved by more sophisticated approaches of the linear ordering problem (LOP) (Garey and Johnson, 1979).

Formally, given a supervised classification problem with a feature set $\mathcal{X} = \{X_1, \ldots, X_n\}$ and a class variable C, it is possible to apply u different univariate relevance metrics in the problem obtaining u different relevance rankings. Each feature X_i presents, in each ranking, a set of different positions $p^1_{X_i}, p^2_{X_i}, \ldots, p^u_{X_i}$. Then, for a given feature X_i, we can take the u associated positions and compute its associated consensus position as

$$p^{cons}_{X_i} \equiv \text{OR}\left\lfloor \frac{\sum_{j=1}^{u} p^j_{X_i}}{u} \right\rfloor \ ,$$

where OR stands for the order (ties are randomly broken) of the value $\sum_{j=1}^{u} p^j_{X_i}/u$ within the set

$$\left\{ \frac{\sum_{j=1}^{u} p_{X_1}^j}{u}, \ldots, \frac{\sum_{j=1}^{u} p_{X_n}^j}{u} \right\}.$$

The features are sorted using this consensus position as the ordering criterion, obtaining the consensus relevance ranking. This is the easiest approach to the consensus. The final ranking illustrates, as a general criterion, the more relevant (first positions) and irrelevant (last positions) features in the supervised problem. Since the metrics are univariate, no dependence between the features is explored. It is very likely to find close features in the ranking that may present a high level of redundancy between them.

8

Consensus over gene selection

From the statistics and data mining fields, the expression level of a gene is represented as a random variable of a probabilistic process. Such a random variable could be measured in different cohorts of samples belonging to different phenotypes. As exposed in the previous chapter, the first approach is to measure the individual relevance of each of those variables in the supervised problem.

However, in a second study phase, the practitioner would like to get a more precise result: a set of relevant genes in the undergone experimentation. Here, the biology borrows the feature subset selection approaches from the machine learning discipline. But the cardinality of the bioinformatic problems is usually a problem and the classical approaches need to be adapted.

Through this chapter we first explore a relatively recent filter feature subset selection. Namely correlation-based feature subset selection, its formulation makes it ideal when dealing with gene expression data coming from microarray experiments.

Gene expression data is always expressed in a continuous scale. This continuous expression needs to be translated into categorical values so as to apply all the proposals included in this part of the dissertation. There are different ways to perform such categorization and this discretization process biases the original data. In order to shorten these biases, we here propose a combination between feature subset selection and discretization approach to look for a small and very relevant set of genes. The consensus approach to do so is called consensus gene selection or CGS and detailed in Section 8.3. Some experiments that demonstrate its good performance in three benchmark microarray datasets are also presented.

8.1 Correlation-based feature subset selection for gene selection

Selecting an optimal subset of relevant genes from a large collection is an NP-hard combinatorial problem (Garey and Johnson, 1979). Due to its own nature, the filter approach is quicker than the wrapper approach: when the number of features increases to more than hundreds, the computing time a wrapper algorithm needs is not affordable with the current resources. On the other hand, the filter approach is independent from the learning algorithm, that is, the selected features are presumed to be good for whatever learning algorithm used afterwards. Lastly, and due to the usual low number of samples in DNA microarray problems, there is a high risk for wrapper procedures to overfit the data. Due to these reasons, filter techniques are an adequate approach to gene selection in such contexts (Inza *et al.*, 2004; Xing *et al.*, 2001; Yu and Liu, 2003, 2004).

When searching for an optimal feature subset, two issues are fundamental: redundancy and irrelevancy. The desired feature subset should have the lowest redundancy among the selected features, and, at the same time, the most relevant features of the problem. These features are closely related to the problem class label.

In 1997, Hall and Smith presented a feature selection method called correlation-based filter selection (CFS) (Hall and Smith, 1997), which deals with these two issues. Based on a hill-climbing search strategy guided by a heuristic evaluation function, CFS accomplishes the redundancy and irrelevancy issues in a linear time, obtaining competitive results in comparison with the wrapper approaches over large, different domains (Hall and Smith, 1999). For a feature subset $\mathcal{S} \subseteq \mathcal{X}$, the CFS filter-inspired heuristic, G_s, is computed as follows:

$$G_s = \frac{k\overline{r}_{ci}}{\sqrt{k + k(k-1)\overline{r}_{ii'}}} \,. \tag{8.1}$$

Equation 8.1 has three components: k represents the subset size, $\overline{r}_{ci}$ denotes the mean correlation between the selected features and the class feature, and $\overline{r}_{ii'}$ denotes the average intercorrelation between the selected features. The numerator can be seen, in a pairwise way, as an indicator of how well a group of features can predict a class. The denominator measures, also from a bivariate point of view, the redundancy that exists among the features of the subset.

A derived consequence of this heuristic is crucial in the microarray context: irrelevant features will not be included as they will be poor predictors of the class, and redundant ones will be ignored because of its close correlation with the features previously included. Bear in mind that there can be features with a high correlation coefficient in relation to the class that are not included in the subset that is finally selected. This is due to the fact that another feature accomplishes the heuristic metric better, although its class correlation is lower than that of the first.

The metric which measures the correlation level between a pair of features and between each feature and the class is based on the *conditional entropy*. If X and Y are random discrete variables with possible states given by r_x and r_y respectively, the *a priori* entropy of Y is defined as:

$$H(Y) = -\sum_{y=1}^{r_y} p(y)\log(p(y)) \ . \tag{8.2}$$

In a similar way, we define the conditional entropy of variable Y, when the value of variable X is observed as:

$$H(Y|X) = -\sum_{x=1}^{r_x} p(x)\sum_{y=1}^{r_y} p(y|x)\log(p(y|x)) \ . \tag{8.3}$$

On the basis of the conditional entropy, the correlation measure between Y and X, also called the *uncertainty coefficient* of Y given X, is defined as:

$$Corr(Y|X) = \frac{H(Y) - H(Y|X)}{H(Y)} \ . \tag{8.4}$$

This coefficient can take values between 0 and 1. A value of 0 indicates that X and Y have no relation; a value of 1 indicates that knowing the X value completely predicts the Y value. These values are used to compute the mean class correlation and the average feature intercorrelation coefficients $\overline{r}_{ci}$ and $\overline{r}_{ii'}$, respectively.

The last issue to consider is the search procedure. We have selected a forward greedy hill-climbing procedure due to two main reasons: affordability for the NP-hard search and the fact that forward selection supplies optimal subsets of small sizes, contrary to backward selection, which supplies bigger ones. Especially, for the DNA microarrays domain, backward selection could select several thousand genes. Many biological studies consider that the number of genes involved in a biological process is not higher than twenty or thirty relevant genes (Golub *et al.*, 1999; Li and Yang, 2002). Hence, forward selection search is chosen.

With all these fixed parameters, a detailed analysis of the original algorithm reveals that there is a large number of repetitive computations within a complete run. To start with, we can change the expression of the correlation measure between Y and X by using the mutual information metric:

$$Corr(Y|X) = \frac{I(X,Y)}{H(Y)} = \frac{I(Y,X)}{H(Y)} \ .$$

If we extend the greedy forward search runs step by step, it is possible to formulate a recurrent expression that minimizes the number of computations. The correlation between the selected feature set and the class variable, $\overline{r}_{ci}$, can be obtained as

$$\overline{r}_{ci} = \frac{k-1}{k}\overline{r}'_{ci} + \frac{I(C, A_{new})}{k\, H(C)} \ .$$

The average intercorrelation between the selected features, $\overline{r}_{ii'}$, is also simplified to the following recurrence:

$$\overline{r}_{ii'} = \frac{k-2}{k}\overline{r}'_{ii'} + \sum_{i=1}^{k-1} R(A_i, A_{new}) \ ,$$

where

$$R(A_i, A_j) = \frac{I(A_i, A_j)\ [H(A_i) + H(A_j)]}{H(A_i)\, H(A_j)} \ .$$

The term A_{new} refers to the new added feature, $\overline{r}'_{ci}$ and $\overline{r}'_{ii'}$ are the intercorrelation values computed in the previous search iteration. In the very first search iteration, the expression of the heuristic is just reduced to compute the mutual information between each one of the features and the class variable, $I(C, A_i)$. Consequently, the first selected feature is the one with the highest mutual information metric coefficient.

In summary, we consider that CFS is an ideal procedure for feature subset selection when dealing with DNA microarray data. Three main reasons support this statement:

- First, the amount of sequences a microarray is now able to analyze. In a problem with more than several thousand genes, a filter approach is the most affordable in terms of the present computational resources.
- Second, a DNA microarray takes a snapshot of many diverse genes, even genes with no relationship with the studied experiment. These genes are supposed to not show any special activity, that is, to be irrelevant. They have to be ignored.
- Third, CFS tries to select uncorrelated features. This statistical orthogonality can be taken to the biological domain, checking whether any of these selected features directly supports a biomarker.

The computational cost of a CFS run directly depends on the selected search policy and on the database's characteristics. By using forward greedy search, this cost can be approached in function of an α parameter that relates the number of original features n and the number of selected features s ($s \approxeq \alpha \cdot n$). In the worst case, the total number of operations for a CFS run will be delimited by the polynomial expression $\alpha n^3 + (N - \alpha)n^2 - Nn$, where N is the number of database cases.

8.2 Discretization matters

A great number of machine learning methods are designed to deal only with discrete data, such as CFS is. In order to use this battery of procedures,

this restriction makes it necessary to translate the data from continuous to discrete value-domains. This translation is made by means of discretization transformation, from the continuous values to an ordinal set of values for each variable or feature. The process can make the original data lose precision, even degrading its original quality.

On the basis of its biological activity, the general assumption towards the possible expression states of a gene is that each gene can only be in a few functional states. As a usual criterion in this field (Causton *et al.*, 2003; Friedman *et al.*, 2000), our assumption is that these possible states are three, using the idea of over-expression, under-expression or base-line activity for the gene.

Most of the original studies dealing with discretized microarray data uses a fixed discretization policy (Friedman *et al.*, 2000). The problem within this strategy is that not all the genes show the same numerical behaviour. For instance, for two different genes, the expression value of an over-expressed profile could be the same as a base-line profile for another one. Another issue about discretizing policies is the fact that once the discretization is performed, the new data fitting is not tested. If the discretization process has biased the original data, this bias affects all the posterior knowledge discovery processes, especially when little data is provided. This could be critical when simple discretization policies are used.

Due to the usual small number of samples, and searching for a more robust data analysis, many DNA microarray-related papers propose to construct different models for the data (Blanco *et al.*, 2004; Dudoit *et al.*, 2002; Lee *et al.*, 2005). Once the models are built, the most adequate one is chosen in function of the problem's context or objectives. Thus, in the context of biological data and due to sample number dependency, the effects of a unique discretization policy for all the data can even be critical.

For the present dissertation, we evaluate the use of three well-known and widely-used discretization policies. Two of them are based on classical statistics: equal frequency (Catlett, 1991) and equal width (Kerber, 1992). The third one is a well-known supervised discretization technique that comes from the machine learning discipline: the entropy discretization of Fayyad and Irani (Fayyad and Irani, 1993).

Equal width and equal frequency techniques do not take into account any information about the class variable distribution in the problem: both are unsupervised univariate policies. Given a number of bins, b, equal width simply sorts the values a feature can take and divides the observed range into b equally sized intervals (Friedman *et al.*, 2000; Tuzhilin and Adomavicius, 2002). Equal frequency divides the range into b bins which gather the same number of occurrences (Sheng *et al.*, 2003).

Entropy discretization is a non-parametrical technique that is considered one of the best and most commonly used discretization policies in machine learning. This technique is known as *entropy* because it uses the entropy measure to identify the optimal bins of a continuous feature. Entropy discretiza-

tion makes use of the data class distribution over the problem, in conjunction with a minimal description length-based algorithm (MDL) (Rissanen, 1978). For each attribute independently, this technique finds the appropriate cut-off points in such a way that the class entropy within each resulting interval is minimum, while balancing this by introducing as few cut-off points as possible. Such progressive search is very robust when the data distribution is skewed. Therefore, no assumption about the data distribution is needed *a priori*.

8.3 Consensus gene selection

In order to tackle the gene selection problem, we seek to find a robust knowledge discovery process and to overcome the problems previously exposed. Therefore, not only is a single discretization evaluated, but the use of many different discretization techniques will be researched. Beginning with a microarray dataset discretized in different ways, we propose to look for a consensus result with larger reliability and robustness than usual single-discretization modelling. These types of consensus strategies have demonstrated good results and they are a typical topic in DNA microarray-related studies (Monti *et al.*, 2003; Swift *et al.*, 2004). Thus, beginning with a microarray dataset discretized in different ways, we search for a consensus set of relevant genes by applying a feature subset selection to all of them. This consensus is expected to return a limited number of relevant genes that may be augmented in function of the experimentation needs (Armañanzas *et al.*, 2005b).

Formally, let the discrete datasets $S_1, \ldots, S_N$ be the result of different discretization policies of the original O microarray dataset. N feature subset selections are performed on the basis of these S_i discrete datasets, producing the following subsets of genes: $G_1, \ldots, G_N$. The consensus gene subset Γ will be the intersection between all of them, that is

$$\Gamma = \bigcap_{i=1}^{N} G_i \ ,$$

with $|\Gamma| = m \leqslant \min_{i=1,\ldots,N} |G_i|$.

In order to amplify the final output gene set, for each of the m selected genes, its q most univariately correlated genes are also selected. Fig. 8.1 graphically exemplifies the whole information flow.

The objective of the overall process was to enhance the robustness of the final solution. The use of different discretization procedures adds independence from a specific discretization task, and obtains a compact gene set $\Gamma = \bigcap_{i=1}^{N} G_i$. Although the original set O is the same, the Γ set contains genes that are considered relevant in different discrete datasets. This fact demonstrates the importance of these m selected genes, showing its relevance

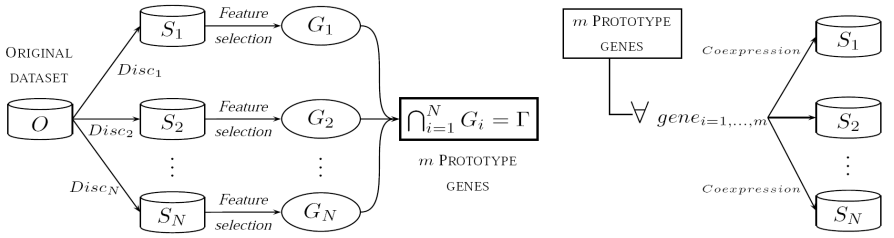


Fig. 8.1. Machine learning data flow to identify the relevant gene set: Identifying the prototype genes and the genes mostly correlated with them.

over the experiment studied. Hence, these genes are considered as *statistical prototypes* showing different behaviour profiles among them.

Due to the conservative formulation of the intersection consensus, there may be genes related to the class that are left as unselected. That is why an amplification of the Γ set is performed and the q most univariately correlated genes with the m prototypes are also selected. Notice that this correlation is computed for each discrete dataset S_i, adding up to $m \times q$ genes to the final selected set. This posterior enlargement of the outcome can add valuable information contained in each of the starting gene sets. It is very likely that when augmenting the consensus selection there could appear repeated genes in the selection. These repetitions are explained by means of their inherent relevance to the problem.

8.4 CGS specification

Although the consensus gene selection in presented as a general approach, the user needs to set up the different methods that it contains in order to run it. The comparison study between different methods applied in the CGS is complex and out of scope of this dissertation. Nevertheless, we propose and argue the use of three of them specially well-suited for the gene selection over DNA microarray data.

Three are the parameters to instantiate from the general approach:

- Discretization procedures – Following the discussion introduced in Section 8.2, we choose the three techniques described: equal width, equal frequency and entropy. Assessing gene activity, the number of bins in equal frequency and equal width discretizations is fixed at three (parameter not needed by the entropy discretization due to its free-parameter nature).
- Feature subset selection technique – Also presented and discussed in Section 8.1, correlation-based feature selection is able to identify the genes most correlated with the phenotype distribution, keeping the redundancy

among them minimum. CFS has been widely used in this bioinformatics context, reporting good results both in time and in relevant genes (Hall and Smith, 1999; Wang *et al.*, 2005b; Sáenz *et al.*, 2008).

- Coexpression measure – Mutual information (see Section A.) has no sign consideration in its formulation and the relationships found could be direct and inverse in the gene profiling. Thus, it can cover a key biological process: positive or negative transcription regulation. Once the prototype genes are found, the number of genes most correlated with each prototype gene is set at nine for each S_i discrete dataset.

The computation time of the full technique is dominated by the feature subset selection step. In the case of the suggested specification, the CFS is clearly the bottle-neck task. Although a lot of mutual informations are computed on the last stage, the computational order for the mutual information is linear (for c classes and v states for the predictive variable is of $O(ncv)$). Moreover, CFS needs to compute in total far more times mutual informations through the search process. Therefore, the general cost of the full pipeline can be delimited by the cost of the number of CFS runs.

8.4.1 Benchmark examples of application

As exposed, the objective of the overall process proposed is to enhance solution robustness. The use of different discretization procedures adds independence of a specific discretization task, obtaining a compact and consensed gene set $\Gamma = \bigcap_{i=1}^{N} G_i$. So as to study how this first stage works, we tested its behaviour using three well-known microarray benchmark datasets:

- *Colon* (Alon *et al.*, 1999) - This array set comes from a colon gene expression study of 62 samples –40 tumoral and 22 non-tumoral– with 1,989 features from the original 2,000 (removing 11 Affymetrix microarray control sequences). Feature intensity values of each microarray are scaled into an average intensity value of 50.
- *Leukemia* (Golub *et al.*, 1999) - Leukemia dataset is composed of 72 samples in two classes of leukemias: Acute Lymphoblastic Leukemia (ALL) and Acute Myeloid Leukemia (AML). From the 7,070 original features, only those with 75% presence value in the raw data are included in this study, that is, 1,161 features. The two phenotypes are distributed as 47 (ALL patients) and 25 (AML patients) samples per class. Features have been scaled using the factors provided by its authors.
- *Lymphoma* (Alizadeh *et al.*, 2000) - One of the very first works in high throughput microarray technology was the analysis of different cells coming from a variety of lymphoma tumors. The set is originally composed of 96 samples and 4,026 probes were measured. There are nine diagnosis classes corresponding with different lymphocite cell types with cardinalities 46, 2, 2, 10, 6, 6, 9, 4 and 11, respectively.

The discretization and feature selection techniques are those recommended above: equal width, frequency, entropy and CFS, respectively. Since we want to analyze the consensus selection, we are not expanding the selected genes but only the intermediate and final consensus genes are analysed. Therefore, the selected intermediate genes and the final prototypes are used for a *leaving-one-out cross validation* (LOOCV) over the continuous datasets, using four different broadly used classification paradigms: logistic regression, k-NN, naïve Bayes with Gaussian assumption and random forest. Estimated accuracies by the LOOCV process are gathered in Table 8.1.

	Genes	**Log. reg.**	**k-NN***	**N. Bayes**	**R. forest**
Colon	1,989				
$\Gamma = \bigcap_3 G_i$	**03**	**83.87**	80.64	87.10	85.48
$G_{Eq.Freq.}$	22	72.58	**83.87**	**93.55**	85.48
$G_{Eq.Width}$	24	74.19	80.65	91.94	85.48
$G_{Entropy}$	40	74.19	82.26	**93.55**	**91.94**
Leukemia	1,161				
$\Gamma = \bigcap_3 G_i$	**04**	**86.11**	83.33	87.50	**87.50**
$G_{Eq.Freq.}$	28	77.78	90.28	90.28	84.72
$G_{Eq.Width}$	19	76.39	88.89	**93.05**	79.17
$G_{Entropy}$	48	80.55	**95.83**	91.67	84.72
Lymphoma	4,026				
$\Gamma = \bigcap_3 G_i$	**16**	87.50	89.60	**87.50**	86.46
$G_{Eq.Freq.}$	198	**97.92**	**94.80**	85.42	**89.58**
$G_{Eq.Width}$	125	94.79	**94.80**	85.42	87.50
$G_{Entropy}$	165	77.08	**94.80**	81.25	88.54

Table 8.1. Estimated accuracy percentages for the benchmark datasets. *k-NN is computed with Euclidean distance and k=2.

Table 8.1 includes in bold, for each classifier and each selected gene set, the highest estimated accuracy values. The consensus selection reduces the number of features by a degree of magnitude in comparison with the single discretization and selections. Even with such a drastic reduction, the consensus features are able to achieve the highest accuracy at least once for each dataset: in Colon with logistic regression, in Leukemia with random forest and logistic regression, and, in Lymphoma with naïve Bayes. Although the differences are not statistically tested, we can check that the core of genes selected by the consensus is able to work out a high degree of class separability. In general, with a reduced number of features, the accuracy estimations always achieve competitive values.

In Section 11.3 an application of this approach is presented to deal with data coming from two autoimmune diseases. The results from the method are validated both from the statistical and biological point of view. This applica-

tion finds previously reported insights into the diseases and, at the same time, points out new biological hypothesis to work on (Armañanzas *et al.*, 2009a).

9

Reliable gene interaction networks

The next step in a biological study is to take a look at the possible relations that the relevant genes present among themselves. DNA microarrays allow the practitioner to measure thousands of gene expressions ar the same time. These data constitute the numeric seed for the induction of such a gene networks known as *gene interaction networks*. The main purpose of a gene interaction network is therefore to map the relationships of the genes that are out of sight when a genomic study is carried out.

In this chapter, we propose a new approach to build gene networks by means of Bayesian classifiers, variable selection and bootstrap resampling. The interactions induced by the Bayesian classifiers are based both on the expression levels and on the phenotype information of the supervised variable.

Feature selection and bootstrap resampling add reliability and robustness to the overall process, removing possible false positive findings. The consensus among all the induced models produces a hierarchy of dependences and, thus, of variables. The practitioner can define the depth level of the model hierarchy so the set of interactions and genes involved can vary from a sparse to a dense set. In addition to their utility as an hypothesis research tool, once a confidence level is set, the network structure can be used as a supervised classifier.

Running examples with DNA benchmark microarray datasets illustrate our proposal. Experimental results show how these networks perform well on classification tasks and how the sparsity degree of the networks can be tuned.

9.1 Introduction

Gene networks or gene interaction networks (Friedman, 2004) are currently a topic under heavy research in the computational biology field. High throughput biological devices have reduced the gap between the traditional medicine and what is known nowadays as biomedicine. But in this context, not only the proof of a certain gene activity is necessary, but also the investigation of

how a set of genes interact among themselves is crucial for the understanding of different complex diseases.

However, there is still a tendency to analyze gene expression data only from a pure numeric point of view, that is, to look for the smallest and most accurate set of genes that are able to distinguish between two or more phenotypes (Bontempi, 2007; Lin *et al.*, 2006; Wang *et al.*, 2007; Yang *et al.*, 2006). This analysis strategy still falls into the problems related with the curse of the dimensionality of these domains. As the high throughput devices, like the DNA microarray devices, begin to be less expensive, the amount of available data will allow to overcome these problems such as, for instance, the overfit effect (Braga-Neto and Dougherty, 2004b; Michiels *et al.*, 2005).

Apart from these studies, computational techniques have proven their capacity to help physicians to analyse the gene activities of complex diseases. In order to understand such complex relations, many approaches have gone on stage. From pure Bayesian networks (Friedman *et al.*, 2000; Peña *et al.*, 2005b; Pe'er *et al.*, 2001) to statistical validations by multiple random simulation (Baker and Kramer, 2006), new graphical models to match gene interactions (Shmulevich *et al.*, 2003; Wang *et al.*, 2005a) or biological validation of previously reported interactions (Hartemink *et al.*, 2001; Rapaport *et al.*, 2007). The main corpus of all these works is to assume that a gene behaves as a random variable of an unknown probabilistic distribution. Over that distribution, the regulatory interactions between the genes are expected to produce corresponding probabilistic dependences within their expression levels (Pe'er *et al.*, 2006).

In this framework, the majority of the works just look for differentially expressed genes to build their models. However, few of them are explicitly focused on the statistical information that the comparison of different sample types contributes. The conditional probabilities learnt through the phenotype statistical distribution in the database will be used to report interactions among genes, not only based on their individual expression levels, but also on their behaviour through the different conditions. This fact involves the addition of the probabilistic relationship that associates the sample class or phenotype with each relevant gene or feature under the study, that is, a supervised-class experimental design (Larrañaga *et al.*, 2006). Our proposal belongs to these supervised studies, stressing the search of robust results by means of a hierarchy of supervised Bayesian classifiers.

Based on the frequency of appearance of each arc within an induced pool of Bayesian classifiers, our approach assigns confidence levels to those arcs. Depending on the confidence level fixed by the expert, the final model can vary from a very simple structure including a small set of dependences to a deep forest-like one with hundreds of them. This property allows to retrieve a hierarchy of autoinclusive models: from the simplest and most reliable one with only one interaction to the most complex one that includes all the detected interactions. These hierarchical networks are computed by means of a set of

tools well-suited for the biological characteristics, taken from the machine learning and statistics fields:

- The estimations produced by stratified sampling with replacement, known as *non-parametric bootstrap*, are cautious. The ratio of false positives in the features induced with this procedure is very low (Friedman *et al.*, 1999). This fact is significantly important when dealing with biological data in which the number of samples is still very low.
- A small set of genes gathers most of the information in an entire microarray. A feature selection procedure must be applied to reduce the dimensionality from thousands to only hundreds of candidate genes (Saeys *et al.*, 2007).
- No *a priori* biological information is used by the Bayesian classifiers, only the phenotype distribution is considered. Therefore, no previous biological premise will bias the final models.
- Consensus conclusions in the analysis of microarray data have already demonstrated good results (Li and Yang, 2002; Monti *et al.*, 2003; Swift *et al.*, 2004). When seeking for robust gene interactions, finding a parsimonious set of both genes and dependences, which have a high degree of confidence on the basis of the data, guarantees a low number of false positives in the final network.

9.2 Induction of reliable Bayesian networks

Specifically, our approach combines a resampling method with an inner feature selection technique and a Bayesian k-dependence classifier (see Section 4.2.3) to obtain a gene interaction network formed by arcs which surpass a certain confidence level. The expert can fix the complexity threshold of the relationships among the genes in the output network so it can be used as a tool to unveil or corroborate biological hypothesis.

The use of Bayesian classifiers to tackle this task implies that, first, the statistical dependences among the genes can reveal real interactions among them. Secondly, the gene interactions not only describe relationships solely among genes, but also describe different biological behaviours based on the phenotype distribution of each gene's expression. Similar studies with the same aim (Friedman *et al.*, 2000; Pe'er *et al.*, 2001; Zhou *et al.*, 2004) make use of the classical *score+search* Bayesian learning scheme and focus their attention on partially directed models. Our method returns directed acyclic models with directed edges and it can be configured with both different variable set selections and Bayesian classifier inductors.

Because of this flexibility, the approach can also be seen as a consensus feature selection if the expert is only interested in the genes or variables connected by the arcs of the output model. Therefore, two different biological validations can be performed: the discussion of the selected genes' relevance and the discussion of the relations reported among them. According to this idea, the

reliability of the results collected in this work is also discussed in both ways: from a pure classification and from a biological point of view (Armañanzas *et al.*, 2008a).

9.2.1 Robust arc identification

The disposal of a low number of instances forces every kind of machine learning technique to look for robustness in its results. In the gene expression context and with this purpose, we propose the combination of two widely known techniques: a *stratified bootstrap* resampling (Efron, 1979) and a feature subset selection.

The bootstrap approach was first introduced by Efron (Efron, 1979). It is based on sampling intermediate databases from the original one. These databases are conformed by instances randomly selected from the original dataset with replacement. The proportion between classes in the original dataset is maintained in each resampled dataset, which is known as stratified bootstrap. This bootstrap scheme is known as *non-parametrical bootstrap* (Friedman *et al.*, 1999) due to the fact that it needs no external parameter to adjust or compute. On domains where the number of cases is low, the bootstrap scheme is widely used to analyse these data (Simon, 1997).

Step 1. Repeat B times
  Step 1.1. Stratified randomly sample N instances with replacement from the original dataset
  Step 1.2. Select an optimal feature subset and reduce the sampled dataset to only those selected features
  Step 1.3. Run the induction algorithm on the new reduced dataset, learning a kDB classification model
Step 2. Compute the confidence level of each arc as the relative frequency of its presence among all the B induced models

---

Fig. 9.1. Robust arc identification algorithm.

After the stratified sampling of the dataset, an intermediate feature subset selection step is undertaken. Throughout this step, we look for the most relevant features in each different resampled dataset; datasets that can show differences among them due to the stochastic nature of the bootstrap resampling. The relevant feature selection constitutes a running parameter to be chosen by the researcher. Feature selection methods that return sets of variables rather than individual relevances are recommended in this step.

Subsequently, a k-dependence Bayesian classifier (Sahami, 1996) is induced for each resampled dataset reduced to the found relevant features. On the basis of all the induced kDB graphical structures, the confidence of each configured

arc between a pair of variables is computed as the relative frequency of its presence in the B induced classification models. Figure 9.1 shows the proposed algorithm.

In a k-dependence Bayesian classifier model all the nodes of its structure graph conditionally depend on the class node. These common dependences will not be taken into account: our aim is to find repeated dependency structures among the predictive variables, as well as to identify which variables are reported by those dependences.

9.2.2 Bayesian networks with high confidence dependences

Let l_{ij} be the arc from variable X_i to variable X_j. On the basis of the robust arc identification algorithm presented in the previous section, we can define a_{ijr} as

$$a_{ijr} = \begin{cases} 1, & \text{if } l_{ij} \text{ is present in the } r\text{-th induced graph,} \\ 0, & \text{otherwise.} \end{cases}$$

The number of occurrences of a certain arc l_{ij} over the B induced classifiers can be expressed as

$$o_{ij} = \sum_{r=1}^{B} a_{ijr} \,. \tag{9.1}$$

From now on, each arc l_{ij} will be associated with its corresponding number of occurrences, o_{ij}. The set of arcs L that have been configured at least once over all the models can be expressed as

$$L = \{l_{ij} \mid o_{ij} \geqslant 1\}. \tag{9.2}$$

Let t be the confidence threshold or reliability level, that is, the number of times that sets the confidence border of the features for an in-depth study. In our case, the set of arcs from L that overcome the threshold t, hereafter known as the set of t-reliability dependences, L_t, is then defined as

$$L_t = \{l_{ij} \in L \mid o_{ij} \geqslant t\}. \tag{9.3}$$

Analogously, the set of variables included in a set of t-reliability dependences L_t, $S(L_t)$, is defined as

$$S(L_t) = \{\boldsymbol{X}_t \subseteq \{X_1, \ldots, X_n\} \mid \forall X_i \in \boldsymbol{X}_t \; \exists X_j \in \boldsymbol{X}_t \;\; l_{ij} \in L_t\}. \tag{9.4}$$

According to L_t and $S(L_t)$, it is possible to build a probabilistic graphical model G_t of t-reliability dependences. In this model, we can find cycles between two variables due to the inclusion of the same arc, but in opposite directions. In such cases, we only take into account the dependence that shows the larger number of occurrences.

Changing the reliability level t, we can build a hierarchy of models, from an empty model to a model that includes almost all the found dependences.

The simplest model corresponds to a reliability level of $t = max\{o_{ij}\}$ $i, j \in \{1, \ldots, n\}$, when this maximum is unique, L_t only comprises a single dependence and G_t includes two variables and one link between them. At the limit, when $t = 1$, almost every dependence is included; only those that are removed to avoid cycles are not included. In this way, when the value of t varies, the autoinclusion property between all the models is verified, reporting a hierarchy of graphical model structures that can be profoundly analysed:

$$G_{max\{o_{ij}\}} \subseteq \ldots \subseteq G_t \subseteq \ldots \subseteq G_0. \tag{9.5}$$

Finally, once a t level is set, the structure of the model G_t can be retrieved and then the parameters obtained from the dataset (Heckerman *et al.*, 1995). The autoinclusive property adds a new characteristic to this gene interaction network: the capability to study how the sets of dependences and variables evolve step-by-step throughout all the models. The $G_{max\{o_{ij}\}}$ model will presumably include just two variables and an arc between them. Since we decrease the threshold, more variables and arcs will appear in the general model. Thus, it is possible for an expert to control the depth of the study and to isolate findings that could comprise a future work hypothesis. In the biodata mining field, the control over the false positives is of crucial interest. So, work hypothesis based on high confidence thresholds is presumed to be far from a statistic artifact.

9.3 Performance analysis

9.3.1 Suggested running parameters

The methodological proposal previously introduced includes a set of running parameters to be fixed, principally the feature subset selection, a boundary for the maximum number of parents k for the k-dependence Bayesian classifier and the number of times that the boostrap loop is performed. Moreover, and especially in the gene expression context, all these parameters are expected to set a scenario in which the running time could be affordable.

For the subset selection step we suggest the use of the already presented correlation-based feature subset selection (Hall and Smith, 1997) (see Section 8.1 for a detailed explanation). Similarly to the consensus gene selection of Section 8.4, the search strategy for the CFS is configured in a classical forward greedy hill-climbing search that starts from an empty set of features. This search strategy guarantees that the cardinality of the output subsets is not of a high dimension.

Once the dataset is reduced by the CFS, the kDB Bayesian classifier to be learnt is configured with a k value of 4. This value allows the graphical models to be both flexible and not sparse when inducing the structures of dependences. Moreover, it implies a sufficient value so none of the possible relevant dependences can be outside the models.

Finally, the proposed algorithm in Section 9.2.1 is repeated a thousand times, that is, the bootstrap parameter B is set to a value of 1,000. This way, we search for arcs that occur a number of times that can be widely considered as reliable.

9.3.2 Computational cost

The complexity order of the full algorithm configured with these parameters can be estimated as the product of the bootstrap parameter B times the computational cost of the feature subset selection and the kDB structure induction. Computing the kDB network structure requires $O(n^2Nmv^2)$, where n is the number of variables, N is the number of cases, m is the number of phenotypes and v is the maximum number of discrete values a predictor variable may take (three in our case).

The computational cost of the CFS step was already discussed in Section 8.1. In the worst case and by using forward greedy search, the computational cost can be expressed in function of an α parameter that relates the number of original features n and the number of selected features s ($s \cong \alpha n$). For each bootstrap iteration the value of α changes, but we will only consider its maximum value for all the B iterations. In such cases, the total number of operations for a CFS run is delimited by the polynomial expression $\alpha n^3 + (N - \alpha)n^2 - Nn$.

In short, the result of the joint algorithm is asymptotically of $\Theta(B\alpha n^3)$ order and the time for computing the conditional probability tables, when the structure is used as a classifier, linearly depends on the number of variables and dependences included when setting the reliability threshold.

9.3.3 Benchmark datasets

The proposed method is tested using the same three benchmark array sets previously presented in Section 8.4.1: Colon, Leukemia and Lymphoma. All of them are well-known microarray benchmark sets and have been widely used for this purpose.

Since Bayesian classifiers can only deal with discrete variables, a discretization process of the original continuous data is approached. On the basis of its biological activity, we assume that a gene can only be in a few different numbers of activity states. As discussed in Section 8.2 a general criterion in microarray analysis (Friedman *et al.*, 2000; Causton *et al.*, 2003) is that this number of states is three: an up-regulated, a down-regulated and a baseline or null activity. Following this idea, we consider the equal width (Kerber, 1992) discretization in three different bins as the most appropriate method to parse the continuous values into discrete states. Because of its formulation, the possible bias included by the discretization is not expected to affect the real gene profiling behaviour.

9.3.4 Graphical outputs

Table 9.1 presents a summary of the numeric results provided for each microarray set. Column $|S(L_1)|$ shows the number of probes that are selected at least once from the original set. The next column, $\overline{|L_1|}$, reflects the average number of arcs configured through all the induced classification models –removing those that create cycles among them–. Lastly, column *Arc* collects the most times configured probabilistic relationship for each array set; within each set, the reported arc is included in a total of $max\ t$ models out of a thousand models.

	Features	$\mathbf{\lvert S(L_1)\rvert}$	$\mathbf{\overline{\lvert L_1\rvert}}$	**max t**	**Arc**
Colon	1,989	617	10.67	317	M76378 → J02854
Leukemia	1,162	587	15.96	205	D49400_at → U46751_at
Lymphoma	4,027	3,710	180.64	321	g4012X → g1171X

Table 9.1. Run statistics for the benchmark microarray sets.

For the Colon array set, the total number of variables in $S(L_1)$ selected represents 31% of the original set. The variables not included can be safely discarded for the subsequent knowledge discovery process. Moreover, and taking into account the arcs configured at least a hundred times (threshold $t = 100$), we can radically reduce this number to only 13 variables. Figure 9.2 shows the graphical structure compounded by all the arcs included in at least a hundred of the models (shaded nodes match variables without parents apart from the class variable). On each arc, the number of times that arc has been included is displayed. Moreover, the graphical thickness of each arc is proportional to each arc's weight. This way it is possible to study the relevance of each dependence and the variables involved within at a glance.

As for the Leukemia dataset, Figure 9.3 reflects the dependences found at least in a hundred runs out of the total thousand. The most configured arc is included in a total of 205 models (D49400_at → U46751_at) which implies that both variables have been jointly selected by the feature selection algorithm and linked by the kDB induction algorithm in such a number of resamplings. From the 1,161 original variables, in Figure 9.3 only 14 are collected, showing the potential of this technique as a feature subset selection approach.

Regarding the Lymphoma array set, the fact that there is no significant reduction in the number of selected variables is interesting. Almost 93% of all the original 4,026 variables, a total of 3,710 ones, are selected at least one time throughout the experiments. This high number of variables explains the high average number of arcs configured, more than 180 arcs per model. Both effects come from the fact that this array set is very complex in its phenotype separability: nine classes distributed throughout only 96 samples. With such a low number of instances per class, the conditional statistics evaluated for the classification models make them very dispersed.

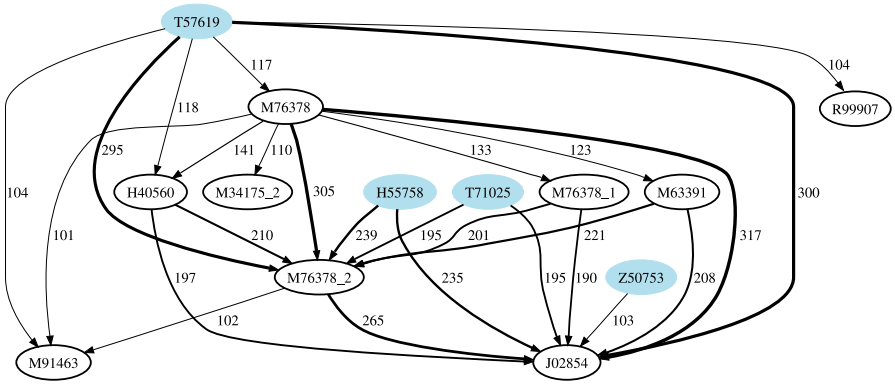


Fig. 9.2. Graphical structure of the high reliable dependences network for the Colon dataset and a t value of 100.

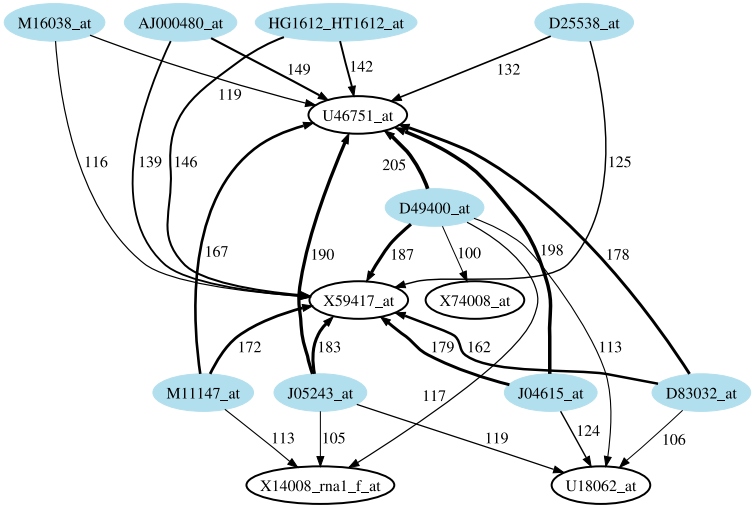


Fig. 9.3. Graphical structure of the high reliable dependences network for the Leukemia dataset and a t value of 100.

9.3.5 Classification accuracy

Although the priority of our proposal is to present and apply a new knowledge discovery method, a reliable set of dependences can also be used in a pure classification application. For this purpose, firstly, the expert has to fix a certain value for the dependency threshold t to return the set of variables and arcs which surpass that level, obtaining a single model. This way, the complexity of the models can be tuned, assessing the scope of the study, variables or aims. After that, the class node is included in the model, adding arcs from it to the rest of the variables. This way the graphical structure is

completed and the corresponding conditional probabilities are computed by their maximum likelihood estimators (see Section 4.1.3.1 for details on the maximum likelihood estimators). Figure 9.4 represents the model structures for the Lymphoma array set for threshold $t = 300$, that is, each model contains the probabilistic relationships that have been jointly selected and configured 300 times at least.

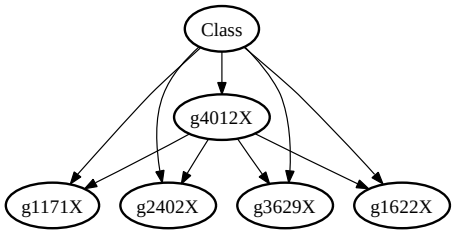


Fig. 9.4. Example of the graphical structure of the network classifier configured from the high confidence dependences set in the case of the Lymphoma array set (the threshold is set at 300).

As the confidence threshold falls, the sparsity degree of the models decreases and, thus, the number of variables to be evaluated increases. Therefore, it is of interest to study how the classification models evolve from the very simplest to the most dense models. In order to analyse this effect, an evaluation of the classification accuracy of each model is performed. Due to the number of models to be evaluated, the total runs and the required computing time for the whole process, a five fold cross validation method is used to estimate the final classification accuracy. This estimation scheme was proven to be well suited for the microarray context (Bouckaert and Frank, 2004; Statnikov *et al.*, 2005), guaranteeing a fair and not overfitted accuracy percentage. For each fold, the run parameters are equal to the ones used in Section 9.3.1: a thousand bootstrap loops, CFS as multivariate filter method and a value of 4 for the kDB classifiers.

Table 9.2 gathers, for each array set and for each fold, the number of selected variables, the total number of arcs induced in all the models, the number of times the most often retrieved dependence is recovered, and the maximum average accuracy achieved. Notice that the accuracies shown are jointly evaluated for a fixed confidence threshold.

The low number of instances in the test set of each fold forces the mean accuracy to have a high level of standard deviation. Thus, accuracy percentages for each array set do not improve the state-of-the-art error rates, but clearly show that recovered high confidence structures are also able to clear up a significant piece of the phenotype information. All these genes and dependences can be of great interest to reveal new underlying biological knowledge.

	$\mathbf{Train_1}$	$\mathbf{Train_2}$	$\mathbf{Train_3}$	$\mathbf{Train_4}$	$\mathbf{Train_5}$	**Mean**	**Std**
Colon (1,989 vars)							
$\mid S(L_1) \mid$	461	652	636	668	513	586	92.92
$\overline{\mid L_1 \mid}$	6.43	11.56	10.24	12.85	7.38	9.69	2.73
max t	352	267	411	265	336	326.2	61.65
max acc. ($t = 264$)	76.92	92.31	83.33	100	66.67	83.85	13.00
Leukemia (1,162 vars)							
$\mid S(L_1) \mid$	545	489	492	413	534	494.6	51.93
$\overline{\mid L_1 \mid}$	15.00	11.02	12.52	8.71	12.05	11.86	2.29
max t	209	241	271	217	284	251.25	33.43
max acc. ($t = 88$)	86.67	60.0	85.71	85.71	64.29	76.48	13.18
Lymphoma (4,027 vars)							
$\mid S(L_1) \mid$	2,511	2,495	2,434	2,501	2,505	2,489.2	31.40
$\overline{\mid L_1 \mid}$	70.28	75.30	72.04	76.90	85.69	76.04	6.00
max t	462	395	343	454	259	382.6	84.26
max acc. ($t = 99$)	70	84.21	94.74	89.47	89.47	85.58	9.47

Table 9.2. Details about the number of variables and arcs for each cross validation fold. The cardinality of the highest configured arc is included.

As a visual tool to study the tendency in classification, we have collected for each threshold the number of variables, arcs, mean accuracies and standard deviation in a single plot (see the example for Leukemia in Figure 9.5). These kinds of charts can be useful to decide to which degree of complexity a biologist is willing to analyse, taking into account the number of variables, arcs and the accuracy level that the model is able to reach.

Inspecting these results shows that there is no direct relationship between the number of arcs/variables and the accuracy of the model. Figure 9.5 illustrates how, despite the addition of new arcs and thus more variables, there is no guarantee that the accuracies of a more complex model would be higher than those from a simpler model. There is a nuclear set of variables/arcs that are able to work out a high degree of the classification separability: more complex models do not necessarily correspond with higher accurate models. For instance, in the Leukemia set results, at a confidence level of $t = 200$, four variables with four arcs correctly predict 70% of the samples. This fact corroborates other studies regarding gene expression classification based on a reduced number of genes (Wang *et al.*, 2007; Baker and Kramer, 2006; Li *et al.*, 2004).

Moreover, recent results in biological gene networks have proven that gene networks are sparsely connected, and that the average number of upstream-regulators per gene is less than two (Leclerc, 2008). Theoretical methods for selection robust gene networks will favor this kind of minimally complex networks. Biological experiments suggest that a sparse, minimally connected, genetic architecture may be a fundamental design constraint shaping the evolution of gene network complexity.

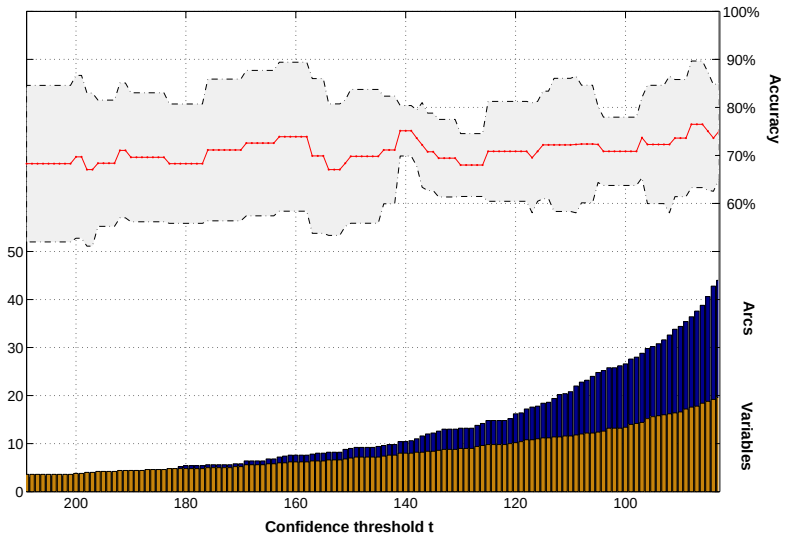


Fig. 9.5. Estimated accuracy tendency over the Leukemia array set. Mean accuracies are presented with their associated standard deviation for each confidence threshold, as well as the number of variables and edges included for that threshold.

9.4 Conclusions

Throughout this chapter a new approach to identify gene interactions has been proposed based on the consensus of Bayesian networks learnt from a pool of bootstrap samples. A major feature of our proposal is the possibility to set confidence levels in order to rely only on interactions highly supported by the expression data. It offers to the expert a broad range of probabilistic dependences to be studied, depending on the available time and laboratory resources.

Bayesian classifiers induce their structure by means of class-conditional probabilities, therefore, studies that compare control against illness samples are feasible targets for this technique. The conjunction of a triplet of well-known machine learning procedures (a stratified boostrap, a feature selection and a Bayesian k-dependence classifier) assures a robust set of results, and, even more importantly, a low number of false positives. A hierarchy of structures is computed, allowing the user to set a threshold in the frequency of appearance of each arc in the pool of bootstrap models. The hierarchy reports for this given threshold both a set of dependences and a set of variables; therefore, it also constitutes a variable subset selector.

Reported results have also shown the potentiality of the induced models in a pure classification task. Reduced sets of dependences/variables are

able to achieve a competitive degree of accuracy when performing a class-discrimination procedure, corroborating previous statements in the microarray analysis field.

In addition, and despite the numeric results, the proposed method is able to point out new research targets. As exposed in Section 11.4 and 11.5 of the applications part of the dissertation, this knowledge discovery method brings into focus a new set of tools to help understand complex diseases that show relationships of different degrees among the involved genes.

10

Population consensus on estimation of distribution algorithms

Estimation of distribution algorithms (EDAs) emerged as a natural alternative to classical genetic algorithms (GAs). EDAs turn the population statistics to their advantage and eliminate the need for the crossover and mutation operators used by traditional GAs. EDAs have produced competitive results in a great many domains (Larrañaga and Lozano, 2002; Lozano *et al.*, 2006), and they have already demonstrated this potential for tackling high dimensional data problems in the field of computational biology (Armañanzas *et al.*, 2008c).

Throughout this chapter, we propose population consensus on top of the general EDA scheme. This consensus approach enhances the robustness of the results and, again, it is designed to deal with the problems that appear due to the curse of dimensionality of some biological data.

Consensus approaches have reported good results on high dimensionality and noisy data in the past (Swift *et al.*, 2004; Valkenborg *et al.*, 2008), especially in terms of reliability and low false-positive findings (Armañanzas *et al.*, 2009a). Specifically, the consensus we propose allows an expert to select a confidence threshold and rely on findings above the set level only.

To test the robustness of the approach, two stability measures are also presented. These stability metrics allow the researcher to quantify the stability behaviour of a subset selection technique and, of more interest, to compare different feature selectors in terms of their stability behaviour.

10.1 Introduction

Estimation of distribution algorithms have been introduced in detail in Chapter 5. As a brief summary to better understand the following concepts, we include their main scheme again in Figure 10.1.

The main characteristic that sets apart current EDA procedures is how the probability distribution $p_g(\mathbf{x})$ is learned. It is not affordable to compute all the parameters needed to specify the full probability model. Thus, the different

$g \leftarrow 0$.
$D_g \leftarrow$ Generate and evaluate M random individuals (the initial population).
do
  $D_g^S \leftarrow$ Select $N \leqslant M$ individuals from D_g according to a selection method.
  $p_g(\mathbf{x}) = p(\mathbf{x} \mid D_g^S) \leftarrow$ Estimate the joint probability distribution of the selected individuals.
  $D_{g+1} \leftarrow$ Sample and evaluate M individuals from $p_g(\mathbf{x})$ (a new population).
  $g \leftarrow g + 1$.
until A stopping criterion is met.

Fig. 10.1. Main scheme of the estimation of distribution algorithm approach.

EDA families must assume different factorizations according to a probability model and to the problem dimensionality. Based on these assumptions, EDAs can be divided into univariate, bivariate or multivariate families (see Section 5.2 for the complete taxonomy).

10.2 Feature selection using an UMDA population consensus

Of the currently developed factorizations of EDAs, the simplest approach is the univariate marginal distribution algorithm (UMDA) (Mühlenbein and Paaß, 1996). UMDA factorization is usually suited to high-dimensional problems in which the possible relationships among the problem variables are unclear. In fact, this technique assumes that the probability distribution of each feature is marginal, that is, no dependence between the problem variables is taken into account when learning the factorization. Thus, the n-dimensional joint probability distribution factorizes as a product of n univariate and independent probability distributions:

$$p_g(\mathbf{x}) = \prod_{i=1}^{n} p_g(x_i) \ .$$

This formulation implies that the learning process is fast compared with other more complex models. Moreover, UMDA scalability is one of its best characteristics because it has a running complexity of nM for the learning process and of $M + \sum_{i=1}^{n}(k_i - 1)$ in memory requirements (k_i is the number of states for feature X_i).

Good results have been reported for UMDAs used to address feature subset selection, especially within the computational biology field (Armañanzas *et al.*, 2008c; Saeys *et al.*, 2004). The UMDA algorithm can be easily adapted to search relevant features in a supervised classification domain by setting up the following elements:

- *Genotype encoding.* Each individual (or candidate feature subset) is represented as a binary array of size n. Each position of the array maps each problem's variable. A value of 1 implies that the respective variable is selected, whereas a value of 0 denotes that the variable is left out.
- *Evaluation function.* The evaluation function for ranking the merit of each individual is the classification accuracy estimated by a *k-fold cross-validation* process.
- *Stopping criteria.* The stopping criterion is either to achieve a perfect classification (100% accuracy estimation) or to have reached a fixed number of generations g.

This scheme is a classical *wrapper feature selection* because it includes the classification process (see Section 3.4). The final output of the algorithm is the best individual in the search, i.e. the feature subset that achieved highest accuracy.

It is very worthwhile to analyze what the selection tendency is over the evolved populations and to investigate if the selected set of features is robust (Saeys *et al.*, 2008). Especially in problems with many features, it is advisable to enhance the robustness and reliability of the selection of relevant peakbins. The classical UMDA has to be adapted to achieve higher rates of robustness. Therefore, we propose building a hierarchy of the best solutions found throughout the search instead of keeping just one best solution (Armañanzas *et al.*, 2009b). These consensus approaches have already been reported to perform well on similar problems (Saeys *et al.*, 2007).

This improvement to the basic algorithm keeps all the best individuals found in the search and evaluates which are the features that have been flagged as selected throughout those solutions. Formally, given a set of solutions S consisting of r individuals, $S = \{\mathbf{x}^1, \ldots, \mathbf{x}^r\}$, of the form $\mathbf{x}^j = (x_1^j, x_2^j, \ldots, x_n^j)$ with $x_i^j \in \{1, 0\}$, the consensus solution over S with a confidence level T ($T \leqslant |S|$) is defined in Equation 10.1:

$$\mathbf{x}_T^C(S) = (x_{1,T}^C, \ldots, x_{n,T}^C) \text{ with } x_{i,T}^C = 1 \Longleftrightarrow \sum_{j=1}^{|S|} \delta(x_{i,T}^j, true) \geqslant T \qquad (10.1)$$

where

$$\delta(x_{i,T}^j, true) = \begin{cases} 1, & \text{if } x_i^j = 1, \\ 0, & \text{if } x_i^j = 0. \end{cases}$$

The consensus solution $\mathbf{x}_T^C(S)$ contains the features in S that are selected at least T times. Obviously, the maximum value for T is $|S|$. This corresponds to the features that have always been flagged as selected in S. By decreasing the value of T, a hierarchy of consensus solutions can be built. This hierarchy fulfills the inclusion property, stating in this case that given S and any $T_0 \leqslant T$, the following implication $x_{i,T}^C = 1 \Longrightarrow x_{i,T_0}^C = 1$ holds for all $i = 1, \ldots, n$.

As the value of T decreases, the number of features selected in the consensus solution should increase. In fact, the addition of new features guarantees that the search procedure does not get trapped in a local optimum and other parts of the search space are also considered. Thanks to this flexibility a whole range of subsets can be evaluated instead of just a single one. Therefore, the user can set up a maximum and a minimum value for T, and the procedure will output all the consensus solutions within that range. Thus, the features returned by this consensus approach are expected to be the most reliable and, at the same time, best suited to the classifier used by the wrapper evaluation.

10.3 Consistency measures and stability index

Stability analysis is a recent topic in the feature selection domain (Kalousis *et al.*, 2005; Kuncheva, 2007). The main aim of stability analysis is to provide a means to state whether the features selected by a given selection approach are robust to changes in the data. In domains where knowledge discovery is a key objective, the stability of the selected features is a highly desirable property. The currently available stability studies rely on the concept of consistency between solutions. A consistency measure between two different subsets of selected features quantifies the degree of (dis)similarity between the two subsets. There exist different ways to measure this consistency, and different interpretations of the measures in terms of the source of the compared solutions: solutions could come from different runs of the same algorithm or from runs of different algorithms.

In stochastic searches, two subsets (A and B) seldom contain an equal number of features. A consistency metric should deal with this effect and be able to analyze subsets of different sizes. In principle, this difference in size should be a penalization term. In this scenario, we present two different metrics to measure consistency and show how to combine them into a stability index.

Let X be the set of available problem features and A and B two subsets of it, $A, B \subset X$. Furthermore, let $n = |X|$ denote the number of features or cardinality of the set X. Let $|A| = k_A$, $|B| = k_B$ and $r = |A \cap B|$ be the cardinalities of the subsets A, B and $A \cap B$. It is possible to define a consistency index between two subsets A and B of different sizes k_A and k_B by adapting Kuncheva's original metric (Kuncheva, 2007). The difference in size is taken into account in the index by selecting the highest cardinality between the two subsets, $k_M = \max\{k_A, k_B\}$. Kuncheva's consistency index can then be reformulated as

$$I_K(A, B) = \frac{rn - k_M^2}{k_M(n - k_M)} \, .$$

Despite the different sizes of A and B, there exists another consistency index able to compare feature subsets of different sizes. Usually known as the

Jaccard similarity coefficient, it has already been used to measure consistency in feature selection problems (Kalousis *et al.*, 2005). The Jaccard index is based on comparing the number of common features in A and B and the total number of selected features:

$$I_J(A,B) = \frac{r}{k_A + k_B - r} \ .$$

The boundary limits of both indices are different: I_K varies between -1 and 1, while I_J ranges from 0 to 1. Therefore, it is not possible to directly compare their values.

The stability index is defined as a metric for comparing the consistency between a set of solutions rather than just two solutions A and B. The mathematical formulation of this metric is straightforward: compute the average of all pairwise consistency measures. Therefore, given a set of solutions $\mathbf{S} = \{S_1, S_2, \ldots, S_m\}$, the stability among them can be computed as

$$\Sigma(\mathbf{S}) = \frac{2}{m(m-1)} \sum_{i=1}^{m-1} \sum_{j=i+1}^{m} I(S_i, S_j) \ ,$$

where $I(S_i, S_j)$ is one of the two possible consistency indices presented above: I_K or I_J.

10.4 Application on mass spectrometry data

The population consensus in UMDA and its behaviour is put on stage to discover biomarkers in mass spectrometry data. This kind of data comes from the proteomics field and has two key characteristics: it is very noisy and the ratio between features and instances is similar to the DNA microarray field. Chapter 12 is fully devoted to all this experimentation. The consensus approach is applied to four different mass spectrometry data. The results reported are discussed from the machine learning point in terms of accuracy estimations, stability and multiobjective optimization. From the biological point of view, the biological meaning of the results is also discussed and compared with the original findings for each dataset (Armañanzas *et al.*, 2009b).

We refer the reader to Section 12.6 to have a joint set of conclusions between the population consensus and its ability to find relevant information in the mass spectrometry datasets.

Part III

Applications in computational biology

11

Genomics

The term genomics refers to the study of the organisms' genomes. Its main effort is focused on determining the entire DNA sequence of such organisms and the hidden genetic map. Once this genetic sequence is uncovered, expression studies research on how that DNA sequence translates into genes and proteins and interacts with the organisms. In this chapter we present two genomics high throughput devices for making expression studies: the DNA microarray and the micro RNA arrays. More interestingly, two microarray and one micro RNA studies are presented and discussed in-depth. Studies for which all the methodological contents of Part II of this dissertation are applied.

The DNA microarray (or just *array*) technology is relatively recent (Holloway *et al.*, 2002) and from the very beginning two different ways of microarray manufacturing took predominant positions. The first one gathers the devices that simultaneously compare two different samples or specimens, while the latter only measures the genetic activity of one sample or specimen. As classical works on the second approach we can find (Alon *et al.*, 1999; Golub *et al.*, 1999), whereas (Alizadeh *et al.*, 2000; van't Veer *et al.*, 2002) were pioneering works comparing two samples in the same array. In detail, (Golub *et al.*, 1999) uses microarrays for finding dysregulated genes within samples of two different leukemias (AML and ALL). In the work by (Alon *et al.*, 1999), the expression of 40 tumoral colon samples are compared against 22 non-tumoral colon samples. (Alizadeh *et al.*, 2000) have a total of 96 samples to analyse, 46 control and 50 belonging to nine different lymphoma types. Lastly, (van't Veer *et al.*, 2002) deal with breast cancer with 98 samples of sporadic and non-sporadic tumors.

Despite the demonstrated power of these high throughput gene expression profiling approaches, some important limitations have been noted. DNA microarray and micro RNA analyses are typically hypothesis-driven, in the sense that the experiments are designed to address a scientific question (Fathman *et al.*, 2005), an approach that could lead to a biased interpretation of the results. Additionally, because microarrays are inherently noisy, they impact

on data quality (Drobyshev *et al.*, 2003; Yang *et al.*, 2002). Moreover, present expression studies usually include a very low number of samples under study. In this context, the reliability of a single data mining technique is no guarantee at all. Clear evidence of these effects is the differences found within the results of data analysis techniques for the same biological data (Li *et al.*, 2001).

The discipline of machine learning, in combination with data mining techniques has been very useful in diverse fields of research, including the bioinformatics discipline, to overcome this technology-intrinsic data noise and to obtain relevant knowledge out of a high volume of data (Larrañaga *et al.*, 2006).

Recently, a new expression and protein synthesis modulators were identified: the small non-coding RNA molecules (micro RNA, microRNA or miRNA). These miRNA molecules are single-stranded RNA molecules of about 20-25 nucleotides (ntd) encoded by nuclear genes (70-150 ntd) and highly conserved among species. These small molecules are not translated into proteins like the normal messenger RNA or mRNA. Instead, they are are processed from primary transcripts (called pri-miRNA) to short stem-loop structures called pre-miRNA and finally to functional miRNA (mature miRNA).

They were first described in 1993 (Lee *et al.*, 1993) although the term microRNA was coined in 2001 (Ruvkun, 2001). The expression pattern of miRNA varies over time and between tissues. Evenmore, in plants, miRNA behaves in an opposite way than in animal organisms. Animal miRNAs are usually complementary to a site in the 3' UTR, whereas plant miRNAs are usually complementary to coding regions of mRNAs.

The mature miRNA molecules are partially complementary to one or more mRNA sequences (target mRNA or target genes) and their function is to down-regulate gene expression. This repression is done via mRNA degradation or inhibition of translation (Bartel, 2004).

The number of miRNA molecules is currently unknown. Initial estimates suggest that there are more than 500 validated human miRNA (Griffiths-Jones *et al.*, 2006; Griffiths-Jones, 2004), although in the public database around 700 were proposed in October 2008[1]. Association studies between miRNAs and complex diseases are still at an early stage. Noteworthy examples are the several links between some miRNAs and some types of cancer (He *et al.*, 2005; O'Donnell *et al.*, 2005) and the essential role of miRNAs for heart conditions in murine (Chen *et al.*, 2008a; Zhao *et al.*, 2007).

Throughout this chapter we present successful applications of the methodologies presented in Chapters 7 to 9 to different microarray and microRNA-based studies. A detailed introduction of how microarray and microRNA devices work is also provided. In addition, different quality criteria are presented to remove possible data artifacts from the raw data. Biological validations of

[1] Data obtained from the miRBase (Griffiths-Jones *et al.*, 2006) at `http://microrna.sanger.ac.uk`.

the results of the data mining techniques are included in the cases where a wet lab collaboration was possible.

11.1 Microarray data basics

DNA microarray technology (Lockhart *et al.*, 1996) offers the possibility to simultaneously analyze the expression of hundreds to thousands of genes (Schena *et al.*, 1995, 1996). In particular, DNA microarrays are assays for quantifying the types and amounts of mRNA transcripts present in a collection of cells. The number of mRNA molecules derived from transcription of a given gene is an approximate estimate of the level of expression of that gene.

RNA is extracted from the specimen and the mRNA is isolated. The mRNA transcripts are then converted to a form of labeled polynucleotides (usually known as *targets*) and placed on the microarray. The microarray is made of a solid surface on which strands of polynucleotides have been attached in predefined positions. We refer to the polynucleotides immobilised on the solid surface as *probes*. The probes consist either of cDNA printed on the surface or shorter *iconoclasts* (chain of nucleotides) synthesized or deposited on the surface. The biological mechanism is simple: the labeled targets bind by hybridization to the probes on the array with which they share sufficient sequence complementary.

After enough time for the full hybridization reaction to take place, the excess sample is washed off the solid surface. At that point, each probe on the microarray should be bound to a quantity of labeled target that is proportional to the level of expression of the gene represented by that probe. Finally, by measuring the fluorescent intensity produced by a laser blast we obtain numbers that ought to estimate the expression level of all the corresponding probes.

In the microarray field, the *experimental design* is defined as the most adequate way to set out both samples and arrays. This planning should answer a clear objective, experimental hypothesis or investigation. Two factors are crucial: the microarray platform available, and the number of samples on the cohort. The experimental design is a key factor and directly affects the subsequent data analysis. Not all the experimental designs produce data that can be directly translated to a machine learning point of view.

As previously mentioned, two main tendencies exist in this field: *single channel* platforms (only includes one sample per array) or *dual channel* platforms (includes two samples per array). One major consequence of this difference is the number of samples that each platform is able to analyse. On the single channel case, the practitioner will have the same number of arrays as original samples whereas, on the dual channel platforms, the way to combine the samples usually decreases the number of arrays available after the experimentation. Even more, and as is later detailed, the scanning technologies,

quantification algorithms and statistics analyses are different between both platforms.

As a general suggestion (Moreno and Solé, 2004), with a low number of samples the recommended experimental design is a sample pair-wise comparison. When there is a sufficient number of samples, an interesting approach is to make reference biological pools (mixture of samples) and compare the expression levels with respect to the individual samples (Sundaresh *et al.*, 2005; Zhang and Gant, 2005). Economically, the pooling is cheaper than the pair-wise design, and, when the number of samples is enough, the statistical results from both of them are similar (Zhang and Gant, 2005).

When working with microarray data, the common tendency is not to deal with the raw expression levels. Instead, the logarithmic transformation is used, the known *logExpression* or *logRatio* value. The logRatio is a logarithmic transformation, in base two, of the intensity differences observed between two targets. The logExpression is just the logarithmic transformation of an individual expression intensity. Roughly speaking, when a target shows a logRatio higher than or equal to an absolute value of one, the corresponding gene is considered as being 'expressed'. Its expression level can be positive (overexpression) or negative (underexpression).

11.1.1 Affymetrix technology

Affymetrix is a manufacturer of DNA microarrays that was founded in 1992. Affymetrix's commercial name for its microarrays is *GeneChip*. The company went public in 1996 with an HIV genotyping GeneChip. Affymetrix manufactures its GeneChips using photolithography over quartz slides. The company has GeneChip models for more than 30 organisms, the most famous being the human genome (HG) arrays, such as the HG-U133A 2.0 or the HG-U133 Plus 2.0.

GeneChip technology is a single channel (or one-color) technology (Lockhart *et al.*, 1996), which displays the intensity expression levels of a single sample of cRNA. That is, only one sample is hybridized against the oligonucleotides synthetised on the array slide. Other biotechnology companies such as Applied Biosystems (*CodeLink* microarrays) or Eppendorf (*DualChip & Silverquant* microarrays) also manufacture this kind of single channel arrays.

A GeneChip microarray from Affymetrix consists of a number of cells (square-shaped areas) in which many copies of a unique oligonucleotide sequence have been in-situ synthesized. These probes are 25 nucleotides long in the classical Affymetrix microarrays, such as the HG-U133A and the HG-U133 Plus models.

Probes are tiled in probe pairs consisting of a perfect match (PM) and a mismatch (MM) of the reference sequence. Basically, the sequences gather in the PM and MM cells are the same, expect for a nucleotide substitution in the middle of the MM probe sequence. The Affymetrix microarrays are then organized in probe sets or series of probe pairs (usually 11 to 16 pairs)

that represents a transcript. These transcripts are defined by the own manufacturer and receive unique accession numbers, such as 58094_at, 12345_a_at, 224847_s_at or 210006_x_at. The last three characters (_at) identify the probe set strand. Probe sets that are designed to detect the anti-sense strand of the gene of interest are annotated with '_at'. The intermediate letter that shows some probe sets' ids (_a, _s or _x) indicates that the genomic sequence is not fully matched within the probes included, the sequence is usually split in parts and more than one probe set contains those parts. The unique probe sets, such as 58094_at, perfectly match a full genomic sequence. Affymetrix provides on its web more detailed information about the naming and probe set matching system[2]. A graphical illustration of all these elements is presented in Figure 11.1.

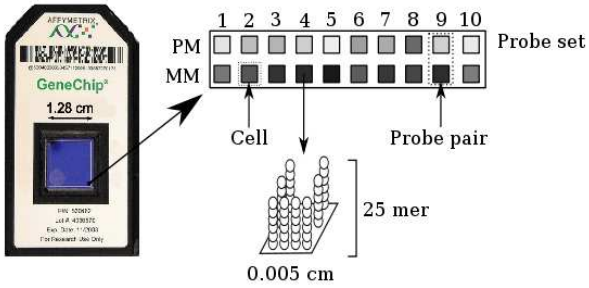


Fig. 11.1. Elements of a classical Affymetrix GeneChip platform.

Once the mRNA sample is transcribed from the original cRNA sample, it is labeled with a biotinylated ribonucleotide analog and fragmented into smaller strands, then the mixture is hybridized with the microarrays oligonucleotides. After hybridization, the chip is stained with a fluorescent molecule (streptavidin-phycoerythrin) that binds to biotin and provides an amplified fluor that emits light when the chip is scanned with a confocal laser. This light is captured by the scanner device and translated as the data image of the microarray (see Figure 11.1).

The signal for a given probe set is calculated using the one-step Tukeys biweight (or bisquare) estimate (Hampel *et al.*, 2005), which yields a robust weighted mean that is relatively insensitive to outliers. The Tukeys biweight method gives an estimate of the amount of variation in the data, exactly as standard deviation measures the amount of variation for an average.

The intensity (signal) of a probe set is computed subtracting a deviation estimate based on the intensity of the MM signal from the PM signal. However, to avoid possible negative values in cases where the MM signal outweighs the PM signal, an adjusted value is used.

[2] `http://www.affymetrix.com/support/index.affx`

One of the most important issues on the Affymetrix technology is known as *detection call* of a probe set. The detection call tries to answer the question if the transcript of a particular probe set is reliably detected by the microarray. The Affymetrix *detection algorithm* solves this question assigning three different states to the call: *absent* (A), *present* (P) or *marginal* (M).

The algorithm is divided into two sequential tasks. The first task is to set a discrimination value that is used as a filter to remove from further considerations all probe sets with insignificant differences between the PM and MM pairs signals. One discriminative value is computed per each pair of signals as $(PM - MM)/(PM + MM)$. Then for the whole probe set, the median of the discrimination ratios of all probe pairs is compared to a user-modifiable parameter τ. The default τ parameter suggested by the company should be set to a value of 0.015 and all the probe sets that do not surpass that level are considered as absent calls.

The second part of the algorithm computes a one-sided Wilcoxon's signed rank test (Wilcoxon, 1945) comparing the signal of all PM cells and all MM cells of the probe set. Then, to state the detection call, the p-value of the test is examined on an axis with two user-definable thresholds α_1 and α_2. If the p-value is lower than α_1, the detection is set to a present value, if its value is between α_1 and α_2, a marginal value is used, and, if the p-value is equal or higher than α_2, the detection would be an absent vale. Affymetrix suggests setting these thresholds at $\alpha_1 = 0.04$ and $\alpha_1 = 0.06$. Although user configurable, these standard values are seldomly changed.

11.1.2 Agilent technology

Agilent Technologies or Agilent, is a company which designs and manufactures instruments and equipment for measurement and evaluation in the field of biology. Originally, it was a division of Hewlett-Packard but in 1999 the products related with the live sciences were grouped together and the new firm split as an independent company.

Its roots in the printing industry have a direct influence in the way that some of its products are conceived. In particular to its microarray products, the technology to in-situ synthesize the nucleotides forming the oligonucleotide chains is *similar* to a big biological printer.

Agilent have developed methods of in-situ synthesis of oligonucleotides on glass arrays using ink-jet technology that does not require photolithography. This ink-jet technology can also be used to attach pre-synthesized DNA probes to glass slides. Thus, Agilent microarrays deals with cDNA probes robotically printed on a microscope slide coated with poly-lysine or poly-amine to enhance absorption of the DNA probes (Schena, 2000).

Agilent oligonucleotide microarrays consist of 60-mers contrasting with the short 25-mers probes employed by Affymetrix. The 60-mer format provides enhancements in sensitivity over the 25-mer format partly due to the larger area available for hybridization (Hughes *et al.*, 2001). A major advantage of

Agilent oligonucleotide microarrays is that they require only one probe per gene or transcript, whereas Affymetrix usually split the transcript mapping different probesets.

Because of cDNA probes are 60 bases long, stringent hybridization conditions are employed and cross-reactivity is almost null. However, the robotic printing often results in substantial variability in the size and shape of corresponding spots on different arrays. For the cDNA arrays, the labeled sample is not usually uniformly distributed across the face of the array and thus the distribution of the sample may differ among identical arrays. For this reason, a direct comparison of intensities of corresponding probes on different arrays is problematic. This interarray variability can be eliminated by a normalization or smoothing task.

Another direct way to avoid this variability is the use of co-hybridization, that is, the use of two samples on the same array (dual channel arrays). In the case of Agilent, the two cDNA samples are labeled with different fluorescent dyes, typically Cyanine-3 (Cy3) and Cyanine-5 (Cy5). As we will later introduce in Section 11.2.2, these compounds reacts at different laser wavelength impulses and that is the way to measure the different intensities for each target.

Depending on the experimental design, the second sample may represent either a specimen whose expression profile relative to the first one is of biological interest, or a reference sample used on all arrays in order to control experimental variability (e.g. pooling samples).

11.2 Quality criteria for processing microarray data

The different microarrays technologies and manufacturers nowadays available still present a common problem: the high presence of noise in the raw results. Due to the measurement technology, the experimental procedures and the intrinsic biological stochastic process, the very first data retrieved by the scanner contains a high noisy component (Yang *et al.*, 2002). Since all the research community is aware of this problem, most of the papers in the field include their own way to remove (or at least reduce) such noise. The bad news is that there is not a universal method (or pipeline of methods) to preprocess and clean the microarray data.

However, some of the most used techniques are becoming a *de facto* standard such as the data normalization by *lowess regression* (Dudoit *et al.*, 2002) or robust multichip analysis (RMA) (Irizarry *et al.*, 2003). Lately, the American National Institute of Health (NIH) has promoted an international project [3] to reach a consensus over this and other open issues within the microarray technology.

[3] Microarray quality control project (MAQC) - `http://www.fda.gov/nctr/science/centers/toxicoinformatics/maqc`

The following sections present the different criteria that have been used in the quality assessment both of a microarray itself and to the probes contained in it. There is a large number of examples of preprocessing policies for single and dual channel platforms (Alon *et al.*, 1999; Golub *et al.*, 1999; Notterman *et al.*, 2001; van't Veer *et al.*, 2002; Walker *et al.*, 2004). Here we present a set of criteria to state that a microarray is not trustable (Section 11.2.1) and another set of criteria to assess if the intensity level reported by a given probe is reliable (Section 11.2.2).

11.2.1 Individual chip quality criteria

We refer to this set of criteria as *individual* because its aim is to state whether a full microarray should be included in a study or, oppositely, rejected from the study dataset. To this end, this individual criteria explores the whole tendency over the array and presents a global value of acceptance or not, rather than individual quality values for each probe in the array.

In the case of the Affymetrix GeneChip technology, the company itself published three criteria to study the reliability of the values read from its microarrays. In addition, one more criterion is included throughout this thesis' study on systemic lupus erythematosus by means of Affymetrix arrays. The four criteria are as follows:

- **Spike control BioB** - *Spike controls* are control probes for sequences that are included in the hybridization mixture. The presence of these controls indicates that the hybridizing, washing, developing and scanning processes are correct. The least represented spike control in the mixture is *BioB*, that is, the control used to evaluate the experiment's sensitivity.
- **Housekeeping control GAPDH** - *Housekeeping controls* are gene probes that are thought to be expressed in all types of tissues. In the microarray, there are probes corresponding to the 3', central, and 5' regions of these genes. The relation between the hybridization signals for the 3' probes with respect to the 5' probes shows the integrity of the synthesized cRNAs. This relation measures the original RNA quality. An array can be considered valid if the 3'/5' relation is smaller than three. The most frequently used housekeeping control, among all the genes represented in the array, is *GAPDH* (glyceraldehyde-3-phosphate dehydrogenase).
- **P call %** - It denotes the percentage of probes identified as present (P) for the detection value in each array. This percentage confirms the quality of an array: an acceptable range of tolerance for the *HGU133A* GeneChips is 40-60%.
- **Array outlier %** - It measures the percentage of probe sets that behave unexpectedly in relation to the pattern shown by the same probe sets in the rest of the experiment arrays. The software used for this analysis – *dChip* (Li and Wong, 2003) from Harvard University– sets the tolerance limit at 5%.

In the case of the dual channel Agilent platform, the individual rejection or acceptance comes early in the experimentation pipeline: the samples must comply with different biological qualities in order to continue in the study. After the arrays are hybridized and scanned, they are rejected only if some present large defects. To investigate these defects, the following section proposes the use of quality metrics that inspect how the genomic reactions behave for each probe in the arrays.

11.2.2 Probeset/probe/spot filtering

One of the most common mistakes within the fluorescent image field is to think that the brighter an image is, the better it should be. Different types and brand of detectors, different image processings, analog-to-digital converters resolutions and other design differences may produce different intensity values for the same microarray spot. In addition, the selected color map, monitor parameters such as the brightness or contrast have an influence on the apparent image brightness.

In order to assess the real quality of an image, we can make use of the *detection limit*. This limit indicates the minimum amount of signal a device is able to correctly quantify. Signal devices will be able to quantify signals beneath the limit, but, inaccurately. The most reliable measure to set a threshold to this detection limit is the *signal-to-noise ratio* (SNR). Roughly speaking, the SNR shows how well a system isolates the real signal level from the background noise or brightness. Dealing with images, the SNR of a signal is defined as

$$SNR = \frac{\text{signal intensity} - \text{background intensity}}{\text{standard deviation of the background intensity}} .$$

In the image field, when an image's SNR is equal or lower than a value of 3, the quantification will not be accurate (Pickett, 2003). Below that level, even though the signal could be visible to the naked eye, its quantification is not reliable.

In the Affymetrix case, the SNR is conceptually replaced by the detection call algorithm presented in Section 11.1.1. On the basis of the presence or absence reported by the algorithm, a probeset filter process is tackled. The process will remove from any further analysis all probesets that, throughout all available arrays, presents as much as a 5–10% of absence values (A). In other words, only those probesets that are detected as present (P) at least in 90-95% of the arrays are considered to be valid. The exact filtering threshold must be set by the practitioner and it greatly depends on the number of microarrays in each experiment. Marginal values (M) for the detection call are normally treated as absence values (A) in such a way so as to be as conservative as possible.

The dual channel microarrays (Eppendorf, Arrait, Agilent) have the disadvantage that these technologies need to measure and quantify signals in two different wavelengths: 635 nm for the Cy5 dye and 532 nm for the Cy3 dye.

Moreover, and to properly quantify both signals, two intensities are needed per dye, the transcript signal and the correspondant associated background.

These four intensity readings are carried out by quantifying the regions or pixels that map each probe within the microarray slide. The final value is computed as an average (or median) of all unitary values (one per pixel), and the associated standard deviation (or the median absolute deviation) is also returned. Throughout this section, we will refer to the following parameters associated to a given probe p as their abbreviated form:

$\mathrm{S}_{\mathrm{r}}^{p}$	Signal intensity measurement in the red channel (635 nm) for the p probe.
$\mathrm{S}_{\mathrm{g}}^{p}$	Signal intensity measurement in the green channel (532 nm) for the p probe.
$\mathrm{B}_{\mathrm{r}}^{p}$	Background intensity measurement in the red channel for the p probe.
$\mathrm{B}_{\mathrm{g}}^{p}$	Background intensity measurement in the green channel for the p probe.
$\mathrm{SNR}_{\mathrm{r}}^{p}$	SNR detected on the red channel for the p probe.
$\mathrm{SNR}_{\mathrm{g}}^{p}$	SNR detected on the green channel for the p probe.
$\mu_{\mathrm{B}_{\mathrm{r}}^{\cdot}}$	Global average for all the values $\mathrm{B}_{\mathrm{r}}^{\cdot}$ of a microarray.
$\sigma_{\mathrm{B}_{\mathrm{r}}^{\cdot}}$	Standard deviation of $\mu_{\mathrm{B}_{\mathrm{r}}^{\cdot}}$.
$\mu_{\mathrm{B}_{\mathrm{g}}^{\cdot}}$	Global average for all the values $\mathrm{B}_{\mathrm{g}}^{\cdot}$ of a microarray.
$\sigma_{\mathrm{B}_{\mathrm{g}}^{\cdot}}$	Standard deviation of $\mu_{\mathrm{B}_{\mathrm{g}}^{\cdot}}$.

In the work by (Chen *et al.*, 2002), a theoretical modelization of the gene expression profiling is presented. Using those statistical models, we adapt three different ratios to assess the reliability of a probe's detected signal on a dual channel microarray experiment. All these *quality metrics* take values in the range 1 to 0, being 1 the maximum reliability value and 0 the minimum.

11.2.2.1 Fluorescent intensity measurement quality

Given a probe p, a SNR that surpasses a value of 6 implies that the signal is very strong relative to the background variation and, thus, its quality is perfect. On the contrary, we have previously stated that a SNR value equal or less than 3 implies a non trustable signal.

Looking for a conservative quality metric, we will use the minimum values of both channels' SNRs to define the quality coefficient of the fluorescent, w_{I}^{p}, for the p probe:

$$w_{\mathrm{I}}^{p} = \begin{cases} 0, & \text{if } min\{\mathrm{SNR}_{\mathrm{r}}^{p}, \mathrm{SNR}_{\mathrm{g}}^{p}\} \leqslant 3, \\ \dfrac{min\{\mathrm{SNR}_{\mathrm{r}}^{p}, \mathrm{SNR}_{\mathrm{g}}^{p}\}}{6}, & \text{if } 3 < min\{\mathrm{SNR}_{\mathrm{r}}^{p}, \mathrm{SNR}_{\mathrm{g}}^{p}\} \leqslant 6, \\ 1, & \text{otherwise.} \end{cases}$$

11.2.2.2 Background flatness quality

Manufacturing faults in the microarray slide can cause problems when measuring the background intensities. One way to detect such bad measures is to compare the background intensity of a particular probe p in one channel against the global average of the same background intensity over all the probes in the same channel.

Then, for the red channel, if the background intensity of p, $\mathrm{B}_{\mathrm{r}}^{p}$, is less than $\mu_{\mathrm{B}_{\dot{\mathrm{r}}}} + 4 \cdot \sigma_{\mathrm{B}_{\dot{\mathrm{r}}}}$ then the background signal is within the background flatness requirements. If not, the quality coefficient for the background flatness, $w_{\mathrm{B_r}}^{p}$, is linearly computed from 1 to 0 until the value of $\mathrm{B}_{\mathrm{r}}^{p}$ reaches $\mu_{\mathrm{B}_{\dot{\mathrm{r}}}} + 6 \cdot \sigma_{\mathrm{B}_{\dot{\mathrm{r}}}}$:

$$w_{\mathrm{B_r}}^{p} = \begin{cases} 1, & \text{if } \mathrm{B}_{\mathrm{r}}^{p} < \mu_{\mathrm{B}_{\dot{\mathrm{r}}}} + 4 \cdot \sigma_{\mathrm{B}_{\dot{\mathrm{r}}}}, \\ \dfrac{(\mu_{\mathrm{B}_{\dot{\mathrm{r}}}} + 6 \cdot \sigma_{\mathrm{B}_{\dot{\mathrm{r}}}}) - \mathrm{B}_{\mathrm{r}}^{p}}{3 \cdot \sigma_{\mathrm{B}_{\dot{\mathrm{r}}}}}, & \text{if } \mu_{\mathrm{B}_{\dot{\mathrm{r}}}} + 4 \cdot \sigma_{\mathrm{B}_{\dot{\mathrm{r}}}} \leqslant \mathrm{B}_{\mathrm{r}}^{p} < \mu_{\mathrm{B}_{\dot{\mathrm{r}}}} + 6 \cdot \sigma_{\mathrm{B}_{\dot{\mathrm{r}}}}, \\ 0, & \text{if } \mathrm{B}_{\mathrm{r}}^{p} \geqslant \mu_{\mathrm{B}_{\dot{\mathrm{r}}}} + 6 \cdot \sigma_{\mathrm{B}_{\dot{\mathrm{r}}}}. \end{cases}$$

Similarly for the green channel, the coefficient for the background flatness, $w_{\mathrm{B_g}}^{p}$, of a probe p must be computed using the respective values for that green channel: $\mathrm{B}_{\mathrm{g}}^{p}$, $\mu_{\mathrm{B}_{\dot{\mathrm{g}}}}$ and $\sigma_{\mathrm{B}_{\dot{\mathrm{g}}}}$. The joint flatness coefficient for both channels is then defined as the minimum value of both of them: $w_{\mathrm{B}}^{p} = min\{w_{\mathrm{B_r}}^{p}, w_{\mathrm{B_g}}^{p}\}$.

11.2.2.3 Signal intensity consistency quality

Due to other physical problems in the whole chain of microarray process, it is possible to get an intensity level that does not properly reflect the real intensity due to an unexpected high color deviation in the pixels. To analyse such problems, we borrow the *coefficient of variation*, $cv = \frac{\sigma}{\mu}$, from the statistics field.

Let cv_{min}^{p} be the minimum coefficient of variation from red and green channels for a given probe p, the signal-intensity-consistency metric, w_{S}^{p}, can be then defined as:

$$w_{\mathrm{S}}^{p} = \begin{cases} 0, & \text{if } 1.1 < cv_{min}^{p}, \\ \dfrac{cv_{min}^{p} - 0.9}{0.2}, & \text{if } 0.9 < cv_{min}^{p} \leqslant 1.1, \\ 1, & \text{if } cv_{min}^{p} \leqslant 0.9. \end{cases}$$

The thresholds for this metric come from the analysis of the coefficient of variation in the different probability distributions. This way, distributions that show a $cv < 1$ (such as an Erlang distribution) are considered low-variance, while those with a $cv > 1$ (such as a hyper-exponential distribution) are considered high-variance. In the middle, the exponential distribution has

a coefficient of variation of 1 because its standard deviation is equal to its mean. In order to allow some degree of freedom, the constraints are set to 0.9 and 1.1, while in the range between them, the quality metric is linearly estimated from the variation coefficient.

11.2.2.4 Global quality metric

Once the three quality metrics are computed for a given probe, we must define how to combine their values to have a single value. There exist no criterion that states which metric is more important, on the contrary, the three seem to be equally important. Following this idea, the global quality metric of a probe should be conservative. In the biological domain is a good policy to avoid false positives as much as possible. Therefore, the global quality metric, w^p, is defined as the minimum value of the three individual criteria:

$$w^p = min\{w_{\mathrm{I}}^p, w_{\mathrm{B}}^p, w_{\mathrm{S}}^p\} \ .$$

However, within a cohort of microarrays, we need to have a criterion to assess the quality of a probe throughout all the arrays. This criterion will state whether a probe should be left out of the following data mining tasks or should be included. The usual way to do so is to compute, for each probe p, the average over all global quality metrics $\overline{w}^p$. Then, we need to fix a minimum threshold of acceptance (usually in the neighborhood of 0.99) to reject all probes that do not achieve this level.

11.3 Microarray analysis of autoimmune diseases by machine learning procedures

DNA microarray technology has been applied successfully to better classify many cancers and to understand the molecular pathways involved in several pathologies (Baechler *et al.*, 2006). Genome-wide gene expression profiles of autoimmune diseases, such as systemic lupus erythematosus, rheumatoid arthritis or Sjogren's syndrome have also been obtained (Alarcón-Segovia, 2004). These studies have identified genes with a dysregulated expression in autoimmune diseases. Further application of microarray analyses should facilitate the identification of pathways that are common in autoimmunity, but more importantly, genes and pathways that uniquely define patients with a particular disease phenotype, which could be useful for the development of specific treatments (Gregersen and Behrens, 2006).

We have applied machine learning procedures to DNA microarray data derived from samples of patients suffering from systemic lupus erythematosus (SLE) and primary antiphospholipid syndrome (PAPS) in order to obtain an unbiased identification of genes that could be relevant to the pathogenesis of

these diseases (Armañanzas *et al.*, 2009a). An important feature of such procedures relies on the fact that no prior knowledge of the system under study is necessary to run the analysis, thus constituting a blind process for which the final results are only based on the characteristics of the raw data. Due to this blindness, a strict validation of the results needs to be tackled: statistical relevance, laboratory qPCR validation, bibliographic revision, regulatory activity evaluation and dysregulation of transcription factors among others.

11.3.1 Introduction

The systemic lupus erythematosus (SLE) is an inflammatory disease with autoimmune features and unknown origin. It can affect multiple organs and body systems, including skin, joints, kidney and the central nervous system (Wallace and Hahn, 2002). The first discoveries related with a *lupus*[4] disease date back to the beginning of the 19th century (Willan, 1808). Nevertheless, it is usually accepted that Biett (Biett, 1857) discovered the erythematosus lupus type in 1833.

The presence of antinuclear antibodies is usually used as the confirmation evidence of the illness, but its early detection is much more complex. In 1982, the American College of Rheumathology (Tan *et al.*, 1982) published a checklist with eleven diagnostic classification criteria for SLE (see Table 11.1). If patients present four out of these eleven criteria, whether or not at the same time, they are diagnosed as SLE with 96% confidence.

According to the National Institute of Arthritis and Musculoskeletal and Skin Disease[5], there are three types of lupus disease:

- Discoid (cutaneous) - It is limited to skin affection and it is easily detected by face, neck and scalp rashes. Diagnosis of discoid lupus is probed by means of a skin biopsy of these rashes. Ten percent of discoid lupus patients may develop the systemic type without knowing why. Frequently, this is due to the existence of undetected systemic disease from the very beginning.
- Systemic - It can affect any body organ, usually, articulations, lungs and kidneys. There is no couple of SLE patients with the same symptoms. When the term lupus is used, it usually refers to the systemic form of the disease.
- Secondary to drugs - After a certain time taking drugs prescribed for different diseases (obviously not SLE), the illness appears. Symptoms of this lupus type are similar to those of the systemic one. The number of patients that suffer from it is very small, and giving up the prescribed drugs lowers the symptoms until they finally disappear.

[4] This name refers to the wolf –*lupus* in Latin– because the skin rashes are similar to the bites of that animal.

[5] `http://www.niams.nih.gov/`

Criterion	Definition
1. Malar rash in butterfly	Fixed erythema, flat or raised, over the malar eminences, tending to spare the nasolabial folds.
2. Discoid rash	Erythematous raised patches with adherent keratotic scaling and follicular plugging; atrophic scarring may occur in older lesions.
3. Photosensitivity	Skin rash as a result of unusual reaction to sunlight, by patient history or physician observation.
4. Oral ulcers	Oral or nasopharyngeal ulceration, usually painless, observed by physician.
5. Arthritis	Nonerosive arthritis involving 2 or more peripheral joints, characterized by tenderness, swelling, or effusion.
6. Serositis	a. Pleuritis: convincing history of pleuritic pain or rubbing heard by a physician or evidence of pleural effusion. b. Pericarditis: documented by ECG or rub or evidence of pericardial effusion.
7. Renal disorder	a. Persistent proteinuria greater than 0.5 grams per day or greater than 3+ if quantitation not performed. b. Cellular casts: may be red cell, hemoglobin, granular, tubular, or mixed.
8. Neurologic disorder	Seizures or psychosis in the absence of offending drugs or known metabolic derangements, e.g., uremia, ketoacidosis, or electrolyte imbalance.
9. Hematologic disorder	a. Hemolytic anemia with reticulocytosis. b. Leukopenia: less than 4,000/ml total on 2 or more occasions. c. Lymphopenia: less than 1,500/ml on 2 or more occasions. d. Thrombocytopenia–less than 100,000/ml in the absence of offending drugs.
10. Immunologic disorder	a. Positive LE cell preparation. b. Anti-DNA: antibody to native DNA in abnormal titer. c. Anti-Sm: presence of antibody to Sm nuclear antigen. d. False positive serologic test for syphilis known to be positive for at least 6 months and confirmed by treponema pallidum immobilization or fluorescent treponemal antibody absorption test.
11. Antinuclear antibody	An abnormal titer of antinuclear antibody by immunofluorescence or an equivalent assay at any point in time and in the absence of drugs known to be associated with "drug-induced lupus" syndrome.

Table 11.1. SLE classification criteria of the American College of Rheumatology.

Regarding the genetic basis of the disease, there are no complete families known to be affected by SLE. However, if a first-line relative suffers from SLE, the risk of developing it is 3% higher than the average risk among the population.

It is accepted that this genetic susceptibility is due to multiple genes (Sullivan, 2000), and a certain threshold of susceptibility must be reached before an external process may trigger SLE. The incidence of SLE varies in different populations because the set of gene variants involved in the genetic susceptibility also varies from population to population. The highest ratio, 1 out of 250, belongs to African-American women between 15 and 44 (NIAMS, 1994). Within men of the same ethnic origin, the risk is approximately 10-fold lower. Perhaps the lowest prevalence is for Caucasians, who have a 20 in 100,000 ratio.

It is also believed that, due to the different natural history of the populations, different combinations of genes can play different roles in genetic susceptibility. There is clear evidence that African-American and Asian SLE pa-

tients suffer from more aggressive variations of the disease, while, Caucasians are more likely to only develop skin-related affection and platelet destruction.

To sum up, more than a hundred genes are now thought to be involved in SLE genetic susceptibility (Sullivan, 1999). It is quite clear that the specific genes involved in such susceptibility are indeed different in different ethnic groups (Sullivan, 2000).

As for the antiphospholipid syndrome, APS, or *Hughes syndrome*, it was first described in 1983 by Graham Hughes and his team in London (Hughes, 1983). APS, also known as "sticky blood" syndrome, is another immunological disease characterized by the repeated appearance of thrombosis (in veins, arteries and capillaries), a high number of miscarriages in the second and third gestation quarters, and thrombopenia or hemolitic anemia. All of these symptoms are associated with the presence of antiphospholipid antibodies (aPL), among which the most known ones are the cardiolipine antibodies (AAC) and the lupic anticoagulant (AL).

APS is called "sticky blood" due to the patient's high blood coagulation tendency. The syndrome can manifest by itself, in the absence of lupus symptoms (primary APS, PAPS), or can develop secondarily in a subset of lupus patients, implying that some pathogenic pathways are common to both autoimmune diseases. (Bertolaccini *et al.*, 2005). In this case, the syndrome is the secondary disease; SLE patients tend to develop APS in 20-30% of the cases.

Neither SLE nor PAPS can be diagnosed clearly. Different criteria have to be evaluated in order to assess its presence. Although a unified diagnosis criterion does not exist (Alarcón-Segovia *et al.*, 1992; Harris, 1990), there are symptoms that point out its possible presence: arterial and/or vein thrombosis, recurrent miscarriages, thrombopenia, and high levels of AAC (IgG and IgM types).

As previously mentioned, the relationship between PAPS and SLE is known to be very close. When a patient has been diagnosed as SLE, this patient can develop a secondary APS (Pons *et al.*, 2001) for unknown reasons. If the patient shows nine out of the eleven fixed criteria for SLE diagnosis, the patient is considered to have possibly acquired secondary APS. Two criteria excluded from the list are photosensitivity and neurologic disorders.

A severe variant of APS has also been detected: catastrophic APS. There have been cases of patients with both diseases that present a secondary multiorgan failure followed by a multisystemic thrombosis of large and small blood vessels. The evolution of this organ failure is almost fatal.

Contrary to SLE, in APS there are families with many members showing positive AAF antibodies. This could indicate that a genetic hereditary component is involved in the development of the illness in these families.

11.3.2 Study participants

After informed consent, patients and controls provided a peripheral blood sample, and PBMC were isolated from whole blood by ficoll gradient purification. All patients were Caucasian women, and had physician-verified SLE or PAPS. Data on age, clinical characteristics, disease activity and current medication are summarized in Table 11.2 and Table 11.3. Disease activity in SLE patients was determined using SLEDAI score (Bombardier *et al.*, 1992).

Id	Age	Smoker	Diagnosis	SLEDAI	Treatment
C6	32	No	Healthy	–	–
C7	53	No	Healthy	–	–
C11	26	Yes	Healthy	–	–
C12	37	No	Healthy	–	–
C14	61	Yes	Healthy	–	–
LV3	24	No	SLE	6	Hydroxychloroquine and NSAID*
LV4	35	No	PAPS	–	ASA**
LB8	40	Yes	SLE	8	NSAID
LB10	31	No	SLE + sec.APS	–	Hydroxychloroquine and anticoagulant
LV11	42	Ex	SLE	8	Hydroxychloroquine and NSAID
LV15	39	No	PAPS	–	No treatment

Table 11.2. Extended personal information about the samples of the microarray experiment; both healthy and ill women. * Non-steroid anti-inflammatory drugs. ** Acetylsalycilic acid.

11.3.3 Sample processing and chip hybridization

For microarray experiments, four patients with SLE, two patients with primary APS and five healthy individuals were used (see Table 11.2). RNA was extracted from PBMC using triZOL followed by RNeasy cleanup. The isolated RNA was amplified and labeled as described in the *GeneChip Expression Analysis Technical Manual*, and subsequently hybridized to HG-U133A Genechip microarrays and scanned according to the manufacturer's recommendations. The labeling, hybridization and scanning procedures were carried out in Progenika.

11.3.4 Microarray quality metrics

After the hybridization and the developing and scanning processes, the final process provided twelve different images, each one corresponding to each *HGU133A* microarray.

Id	Age	Smoker	Diagnosis	SLEDAI	Treatment
C2	26	No	Healthy	–	–
C8	70	No	Healthy	–	–
C9	45	No	Healthy	–	–
C18	43	Yes	Healthy	–	–
C21	49	No	Healthy	–	–
C25	–	–	Healthy	–	–
LB1	27	Yes	SLE	2	No treatment
LV7	46	No	SLE	2	Hydroxychloroquine, prednisone, NSAID
LV9	53	No	SLE	0	Hydroxychloroquine, aceclofenac
LV13	49	No	PAPS	–	Danazol, hydrochlorothiazide, acenocoumarol, ASA
LV14	68	No	PAPS	–	ASA
LV20	33	No	PAPS	–	ASA
LV21	44	No	PAPS	–	Enalapril, simvastatin, acenocoumarol
LV25	–	–	PAPS	–	ASA

Table 11.3. Extended personal information about the samples of the qPCR experiment; both healthy and ill women.

The next step is to test the quality of each microarray. To evaluate the reliability of the microarrays in the experiment, diverse values are measured. The arrays that do not comply with the reliability criteria have to be considered outliers and, therefore, rejected. The four criteria considered for each microarray are the ones introduced in Section11.2.1: presence of control BioB, ratio of GAPDH, percentage of P calls and percentage of outliers.

Id	BioB	GAPHD	P Call %	Array outlier %
μa_1	Present	0.87	37.7	0.24
μa_2	Present	1.08	40.0	0.04
μa_3	Present	1.12	44.3	0.07
μa_4	Present	1.29	45.6	0.16
μa_5	Present	1.47	42.1	0.13
μa_6	***Absent***	1.79	***29.9***	***39.09***
μa_7	Present	1.25	47.1	0.21
μa_8	Present	1.35	44.0	0.02
μa_9	Present	1.08	47.1	0.09
μa_{10}	Present	1.07	44.6	0.17
μa_{11}	Present	1.08	40.3	0.04
μa_{12}	Present	1.38	42.4	0.11

Table 11.4. Realibility criteria values for each performed microarray in the experiment.

Table 11.4 lists the values of the reliability controls for each array. The μa_6 microarray does not reach a sufficient reliability level: in three out of the four measures, it does not verify the minimum acceptance thresholds. Furthermore, the percentage of outlier probe sets is extremely high, pointing to a bad quality in the original tissue or some failure in the intermediate processes. Due to this, the μa_6 array is rejected and it is not included in any further analysis.

However, our purpose is to study the intensity change showed by each probe set between *sample* and *reference* tissues. Due to the fact that the Affymetrix technology only includes one tissue on each microarray, after all the physic processes have finished, the user must compute these comparisons between the reference and the sample microarrays in a synthetic way. This *sample vs reference* experimental design is the direct translation of the classical dual channel comparison made on cDNA arrays (Yang and Speed, 2002).

In order to produce this kind of comparison, MAS (Microarray Affymetrix Suite) software can generate synthetic hybridization-crosses on the basis of the original single channel microarrays. Using one microarray as the comparison *baseline* or reference, and another microarray as the compared one or sample, it generates a synthetic cross between them. For each of these synthetic crosses, the most valuable fields are the *comparison* and the relative expression change or *logRatio* values. The comparison value shows the behaviour observed for each probe set between the baseline and the compared intensity levels, that is, between both channels. Similar to the detection value, comparison is computed on the basis of a Wilcoxon's rank test, and its possible values are increase (I), marginal increase (MI), no change (NC), marginal decrease (MD) and decrease (D).

It is worth mentioning that it is possible to find genes that show an increase (I) or decrease (D) value for comparison when their detection value is absent (A). This collateral effect could be caused by a very little transcript quantity. Hence, and not relying on the qualitative interpretation given by the comparison value, the first filter stage comprises discarding all probesets showing an absent detection value in both channels (AA). Microarray internal control sequences are also removed. In our experimental process, after this filtering process, the amount of probes decrease from 22,067 to 8,808; these sequences form our starting gene set, our critical set of predictive variables.

Finally, a last issue arises at this preprocess stage: the μa_{10} microarray corresponds to a patient who has developed secondary APS (see Table 11.2). This fact can be problematic if the genetic pattern shown by the patient is more similar to primary APS than to SLE. The μa_{10} generates five different instances out of the twenty SLE class labeled instances. In order to study the similarity degree of these five instances with respect to the rest of SLE instances, a hierarchical clustering is carried out. The clustering parameters were set to: Pearson correlation and average linkage. The analysis clearly showed that instances from μa_{10} are clustered with the rest of SLE patients and not with PAPS patients (see Fig. 11.2). Furthermore, clustering of all

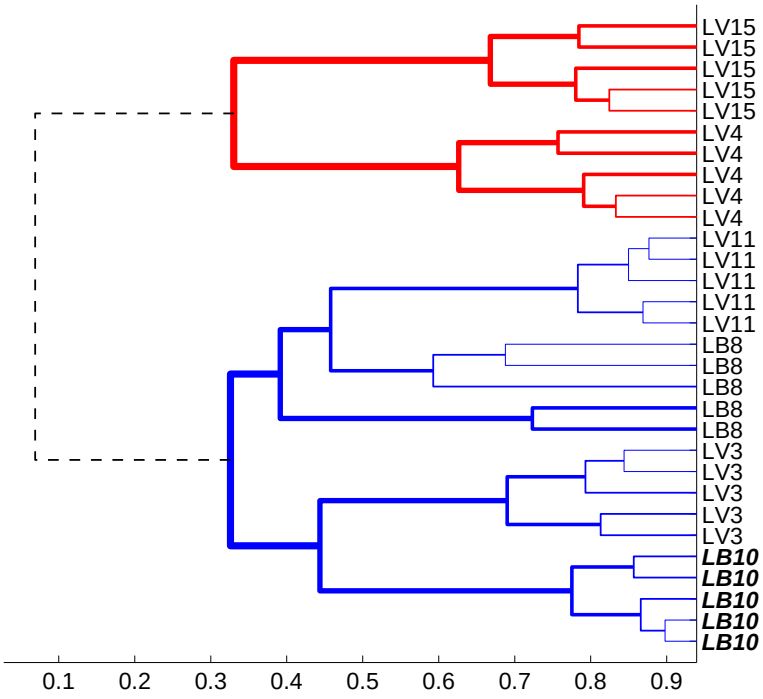


Fig. 11.2. Clustering of the illness instances (SLE & PAPS). Branches coloured in blue belong to SLE class labeled instances, while branches in red belong to PAPS instances. The expression profiles of each test sample were compared separately with the expression profiles of control samples, resulting in five comparisons for each SLE or PAPS sample. The five instances generated by sample LB10 flawlessly behave as the rest of SLE instances, being all clustered in the same group.

SLE instances is very similar and homogeneous, implying that a secondary acquisition of an antiphospholipid syndrome does not significantly modify the transcriptional profile that characterizes SLE.

11.3.5 Experimental design

Based on the nature of both diseases, the study of the following three different diagnostic problems is defined as medically and biologically relevant. We will regard each diagnostic category as two different problem classes:

Healthy *vs* SLE *vs* APS - We are interested in genes that are differentially expressed in each disease: genes that present different expression profile behaviours in each diagnostic category are highly interesting.

Healthy *vs* Unhealthy - Diagnosis of both diseases is not easy, so the identification of genetic patterns is helpful. Differential patterns between medically fit individuals and patients suffering from SLE and APS can help detect and diagnose, when there is doubt.

SLE *vs* APS - Both diseases have many similarities: APS can appear simultaneously as a secondary disease, but the standard case is to detect them individually. Although a large number of genes show the same expression profiles due to their similarities, a differential analysis between them is desirable in order to find genes with different behaviours in both diseases.

Based on the patients' collected data, this work tackles the first diagnostic problem exposed. From the machine learning point of view, these diagnostic problems can be seen as supervised classification tasks: the second and the third modelizations can be considered subproblems included in the first one.

This three-class supervised problem resulting from the experimental methodology comes from the three different ways in which the synthetic crosses are performed:

- Ten of them correspond to the crosses between the control arrays among themselves (never an array against itself), conforming the HEALTHY instances.
- Another ten correspond to the crosses between the five control and the two APS patient arrays, conforming the ten APS instances.
- The last twenty correspond to the crosses between control and SLE arrays (the SLE instances).

All of these forty comparisons form the set of forty instances of our machine learning task. Figure 11.3 shows the logRatio expression values for the subset of 8,808 preselected genes and the forty instances, showing no *a priori* unwanted or unlikely behaviour in any of them.

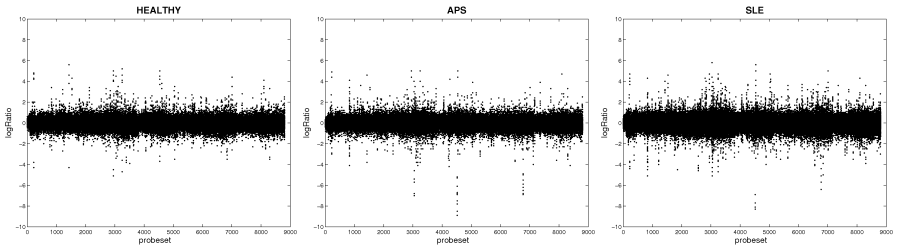


Fig. 11.3. Expression logRatios for all problem categories; HEALTHY (left), APS (center) and SLE (right) patients.

11.3.6 Results & discussion

To identify genes that discriminate SLE and PAPS patients from healthy controls we applied the data mining workflow introduced in Chapter 8, namely the *consensus gene selection*. The supervised classification scheme in use is a three-class classification problem also previously introduced in Section 11.3.5.

For setting up the gene selection method in use, we used a set of three different discretization policies, namely: equal width (Kerber, 1992), equal frequency (Catlett, 1991) and entropy (Fayyad and Irani, 1993). As a usual criterion in this field (Friedman *et al.*, 2000; Causton *et al.*, 2003), our assumption is that a gene could be in three possible states, using the idea of over-, under- or base-line activity, so the number of bins was set to three. As for the feature selection approach, we used the CFS method presented also in Section 8.1. To enhance the jointly selected genes, we used the classical mutual information as a co-expression measure (Cover and Thomas, 1991).

A total of 150 probes or genes out of an original set containing more than 22,000 genes were detected as the relevant genes whose differential expression confidently discriminates among SLE patients, PAPS patients and healthy controls. The complete list of relevant genes, including the Affymetrix probe ID, the gene symbol, their locus, their relative gene expression in SLE or PAPS patients and a short description is available online through the Supplementary content page [6]. The significantly different expression profile exhibited by these genes in patients samples relative to samples of healthy controls could contribute to the pathogenesis of the autoimmune conditions analyzed in this work.

Many of the genes identified in our study have not been previously implicated in SLE or PAPS, and represent new biomarkers of these diseases. Interestingly, a link with autoimmunity, and in particular to lupus, has already been established for a significant number of the genes included in this group of dysregulated genes. Such is the case of the genes SSB, SP100, H1F0, all of which are lupus autoantigens (Scofield, 2003; Srivastava *et al.*, 2003; Maddison *et al.*, 1988; Wichmann *et al.*, 2003; Hardin and Thomas, 1983) or TAP-1, a transporter gene with polymorphisms showing genetic association with SLE (Correa *et al.*, 2003; Hualupusng *et al.*, 2004) among others.

11.3.6.1 Statistical analysis

From the 150 total genes returned by the consensus gene selection, there were a total of eight statistical prototypes (CPSF1, SLC25A12, UQCRB, NADK, MICAL2, KIAA0776, PARL and CECR1). It is mandatory to note that prototype genes are the result of a statistical process and their aim is to comprise the statistical axes of the problem. Although a direct biological translation could be made, it would have no biological link to the statistical and the

[6] Supplementary page at `http://www.sc.ehu.es/ccwbayes/members/ruben/sle`

biological interpretations: the prototype genes could not have any special biological contribution in the domain of the diseases under study.

The statistical analysis of the selected genes was performed in two ways: by measuring the relevance of the selected genes, and the estimated prediction accuracy in a supervised class prediction procedure. *A priori*, the selected genes should be relevant to the problem, showing a high correlation degree with the phenotype distribution.

Using the Elvira platform (Elvira Consortium, 2002), and on the basis of the three discrete datasets, seven different univariate filter rankings (Inza *et al.*, 2004) were calculated. As explained in Section 7, in order to obtain an average ranking for each dataset, each gene was weighted with a coefficient proportional to the relative positions shown in each ranking. The consensus rankings are accessible through the online Supplementary content page, presenting a list of the 8,808 (924 in the case of entropy discretization) genes ordered by their correlation level with respect to the problem class label.

Gene	**EF**	**EW**	**Entropy**
CPSF1	22	93	46
SLC25A12	13	26	32
UQCRB	33	33	51
NADK	106	106	142
MICAL2	11	14	20
KIAA0776	31	35	30
PARL	1	3	12
CECR1	19	38	68

Table 11.5. Positions of the statistical prototypes over the consensus rankings for the three discrete sets (EF, EW and Entropy)

Table 11.5 shows the positions of each statistical prototype in the consensus rankings of each discrete dataset. It is easy to check that the selected genes appear in the top positions of the rankings, with average positions of 29.5, 43.5 and 50.1 for equal frequency (EF), equal width (EW) and entropy discretization, respectively. Such average positions significantly differ from the ones obtained if a random selection is made: 4,004.5, 4,004.5 and 462.5, respectively.

The second aspect of the statistical analysis comprised the estimation of the prototypes' strength when classifying a new instance not included in the original set. This strength was evaluated on the basis of different classifier performance tests. Due to the great difference between the number of genes (predictive variables) and the instances of each experiment, many distinct and equally effective classifiers may exist for the same training set (Dudoit *et al.*, 2002; Lee *et al.*, 2005; Inza *et al.*, 2004; Michiels *et al.*, 2005). This fact led us to consider four different classification models instead of only one. Furthermore, and trying to cover a wide range of classical paradigms, the four

models chosen come from different classification families and are commonly used in DNA microarray class prediction studies (Lee *et al.*, 2005):

- **Logistic regression** - Logistic regression (Kleinbaum, 1994) has become a very widely used classification paradigm in life sciences because its parameters can be interpreted as risk factors. Logistic regression is based on the *logistic function* and it allows an interpretation in probability terms. A set of parameters has to be estimated from the problem data, usually known as *regression coefficients*. Usually, regression coefficients are estimated using the maximum likelihood estimation method, but there are adaptations that penalize this maximum likelihood with other factors. The logistic regression model used in this work penalizes the likelihood estimation with an estimator known as the *ridge* estimator (Cessie and Houwelingen, 1992).
- ***k*-Nearest Neighbor** - The k-NN algorithm (Aha *et al.*, 1991) proceeds with the classification task in terms of similarity: unlabeled examples are classified based on their distance to the examples in the training set. k-NN is a classification paradigm with no explicit classification model. In other words, there is no learning stage in which a mathematical model is induced, and from which the categorization stage is tackled. It finds the k closest features in the data and assigns them to the class that most frequently appears within the k-subset. In this work, k-NN is computed with Euclidean distance and a k value of three.
- **Naïve Bayes** - Continuous naïve Bayes (John and Langley, 1995) belongs to the Bayesian classifier family. The model parameters are estimated with a factorization based on the normal distribution assumption for each variable. A detailed explanation is included in Section 4.2.1 of this dissertation.
- **Random forest** - This classification paradigm belongs to the tree-like classification family. Random forest (Breiman, 2001) builds a forest composed of t random trees. When building these trees, a random variable selection is performed. The random tree set is learnt using a bootstrap instance selection, and the built trees are not pruned. For our work, no variable selection is configured at the induction step, because a feature selection has already been performed. Thus, using all the predictive variables provided, ten random trees are built for each forest.

In order to obtain a fair estimation of each classifier performance a crucial task arose: the choice of the most suitable accuracy estimation method in the context of the microarrays. Classical methods such as hold-out, simple or leaving-one-out cross-validations have been demonstrated to not fit on the intrinsic microarray dimensionality problem (Braga-Neto and Dougherty, 2004b; Statnikov *et al.*, 2005). Nowadays, there are two main approaches commonly accepted as the best estimation techniques for this domain: the *corrected bootstrap estimator* (Efron, 1983) and *nested stratified cross-validation* (Weiss and Kulikowski, 1991).

We chose the use of a nested stratified cross-validation scheme as the accuracy estimation method. This method comprises the performance of two different stages: one internal (or *inner*) loop in which the parameters of the classification methods are estimated; and an external (or *outer*) loop in which the classifiers are induced and validated against previously unseen instances. In our specific case, the feature selection methods were run throughout the inner loop and the different classifiers were induced on the basis of these selected features. This classifier is tested over instances not previously seen in the induction stage.

The next parameter to adjust was the number of times that all this process is done for each classifier and for each feature selection method. Taking advantage of previous works, it is proven that the ten times repetition of ten of these stratified cross-validations obtains suited accuracy estimations (Bouckaert and Frank, 2004; Statnikov *et al.*, 2005). This validation scheme is usually known as 10-times 10-fold cross-validation.

As for the study of the prototype's classification strength, we performed the validation over four different gene sets: the intermediate genes selected by the correlation feature selection over the three different discretized data sets, and the consensus prototype genes. The number of selected genes and estimated accuracies are presented in Table 11.6. All the induction and validation processes were computed using the Waikato environment for knowledge analysis (WEKA) framework (Witten and Frank, 2005).

	$\mathbf{\Gamma} = \bigcap_{\mathbf{3}} \mathbf{G_i}$	$\mathbf{G_{Eq.Width}}$	$\mathbf{G_{Eq.Freq.}}$	$\mathbf{G_{Entropy}}$
Set size	6.17±0.72	25.57±3.19	38.83±2.01	41.42±3.02
Log. reg.	86.75±3.72	95.25±3.05	95.25±2.61	95.00±3.35
Naïve Bayes	86.75±5.01	97.00±1.00	96.25±2.02	96.75±2.25▲
k-NN	88.50±4.50	100.0±0.00▲	100.0±0.00▲	98.75±2.02▲
Rand. forest	80.25±8.98	95.75±2.75	93.00±4.44	90.25±4.39

Table 11.6. Estimated accuracies obtained for the 10-times 10-fold cross-validation on each classification paradigm.

To assess the significance and reliability of the consensus genes in comparison with each one of the intermediate gene sets, a *corrected repeated k-fold cv test* (Bouckaert and Frank, 2004) was performed. This statistical test has been proven as one with the most suited relation between the Type I and II errors (Nadeau and Bengio, 2003; Bouckaert and Frank, 2004) and a high replicability degree (Bouckaert and Frank, 2004). The test compares the differences between two different classification algorithms by a special corrected t-test. The null hypothesis is that both algorithms have the same classification behaviour; the alternative hypothesis states that one algorithm outperforms the classification degree of the other.

For each base classifier, the accuracy of each single discretization policy was compared with respect to the consensus approach. For all the twelve com-

parisons, only four of them (values with a ▲ symbol in Table 11.6) rejected the null hypothesis for an $\alpha = 0.05$ significance level. For a 0.01 level, none of them showed statistical differences –the null hypothesis was not rejected in any of them–. These results let us state that, while suffering a decrease in the classification accuracies, the differences between the use of the consensus prototypes and the intermediate selected gene sets are not statistically significant in many comparisons.

11.3.6.2 Biological analysis

A. Verification of microarray hybridization by quantitative Q-PCR analysis

In order to perform a rigorous validation, we used a second cohort of SLE and PAPS samples. DNA purified from these six healthy donors, three SLE patients and five PAPS patients (see Table 11.3) was reversed transcribed into cDNA.

Quantitative TaqMan PCR analyses were performed for the following five genes (all previously related to the diseases by other studies): H1F0, PPIA, GNLY, SSB and SP100. In addition, qPCR of TBP was performed on all samples, which served as an internal control. The primers and TaqMan probes for all the genes were obtained from Applied Biosystems. The reactions were run by triplicate on an ABI 7900HT Fast Real-Time PCR System from Applied Biosystems at the Genomics Facility of the University of the Basque Country, using standard cycling conditions. Results were analysed with the Sequence Detection System (SDS) Software v2.0 to obtain the Ct values for each sample.

A DCt value was calculated reflecting the difference between the average Ct of the replicate samples obtained for the control gene (TBP) and the average Ct of the replicate samples obtained for the test gene to be validated. Using these DCt values as the raw expression value in the qPCR experiment, we first determined the median DCt for all the healthy control samples. Next, we calculated the difference between the DCt of each test sample and the DCt values of the healthy controls, thus obtaining a set of DDiff values for each phenotype and gene.

Two out of the five monitored genes (H1F0 and SP100) are IFN-regulated genes previously associated with SLE (Hardin and Thomas, 1983; Wichmann *et al.*, 2003). Our microarray data showed that both genes were upregulated in SLE patients, but only H1F0 expression was increased in PAPS, whereas SP100 expression was downregulated.

SSB, also called La autoantigen, is a ribonucleoprotein involved in chromatin metabolism, and is known to elicit autoantibody responses in SLE (Maddison *et al.*, 1988). Its expression in SLE samples was similar to controls, but it was downregulated in PAPS. The remaining two genes (GNLY and PPIA) that were selected for PCR analysis are known to have a role in the execution of immune functions (Deng *et al.*, 2005; Jin *et al.*, 2004; Zander *et al.*, 2003) and were found to be downregulated in PAPS samples.

Gene	Phenotype	Median	1st-quartil	3rd-quartil	Expected	p-value
H1F0	Control	0.0	-0.99	0.99	–	–
	SLE	1.66	1.0	2.16	UP	< 0.0001
	PAPS	2.08	1.01	3.14	UP	0.00264
PPIA	Control	-0.32	-1.37	1.04	–	–
	SLE	0.8	0.4	1.56	BASELINE	0.12506
	PAPS	-1.39	-1.99	-0.7	DOWN	0.06598
GNLY	Control	0.0	-1.34	1.26	–	–
	SLE	-0.1	-0.79	0.58	BASELINE	0.95635
	PAPS	-2.65	-3.26	-1.93	DOWN	< 0.0001
SSB	Control	0.0	-1.51	1.51	–	–
	SLE	0.8	-0.28	1.66	BASELINE	0.20110
	PAPS	-3.17	-3.56	-1.41	DOWN	< 0.0001
SP100	Control	-0.16	-1.09	0.97	–	–
	SLE	3.19	2.26	3.33	UP	0.00076
	PAPS	-1.22	-2.03	-0.89	DOWN	0.07220

Table 11.7. qPCR output values and expected activity for five genes from the relevant genelist.

For each gene and phenotype, the median values of their DDiff, together with the expected gene expression activities are shown in Table 11.7. As a dispersion measure of the results, the values for the first and third quartile of each group of values are also shown. Fig. 11.4 graphically summarizes the results obtained for each gene within the qPCR validation.

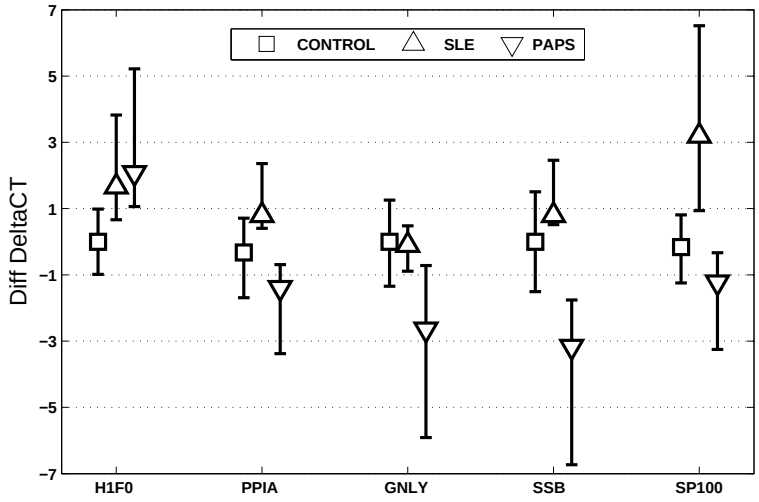


Fig. 11.4. qPCR validation summary for five genes from the relevant genelist.

As a criterion, a median DDiff value between -1 and 1 was considered a baseline activity, that is, unchanged with respect to healthy controls. Median values higher than 1 reflect an upregulated activity, while values lower than -1 reflect a downregulated activity. Clearly, the expected gene expression profiling as measured by microarray quantitation is fully validated by the qPCR experiment.

As a final validation criterion for the expected gene expression activities, a statistical test was performed comparing the DDiff expression values between the control samples and either SLE or PAPS samples. Column p-value in Table 11.7 gathers the output of a non-parametric Mann-Whitney hypothesis test (Mann and Whitney, 1947), showing that all the values of over or under expression are statistically significant for an $\alpha = 0.10$ significance level. In addition, the p-values for the baseline activities show no statistically significant differences between these cases and the control expression. Thus, all these results are consistent with the expected expression activity for each gene.

B. Functional characterization of the relevant genes in SLE and PAPS

To check whether the results made sense from a biological point of view, we have analysed the list of dysregulated genes with the FatiGO+ tool (Al-Shahrour *et al.*, 2006). FatiGO+ can be used to search for the GO annotations[7] that are overrepresented in a list of genes. The significance of the overrepresentation is assessed by means of a Fisher exact test. From the 150 dysregulated genes identified, only 112 have GO annotations. Fig. 11.5 shows the results obtained with this tool (for terms in the level 3 of GO biological process annotations). As we can see in the figure, immune-system-related annotations, such as *defense response* or *immune system process*, are overrepresented in the list of dysregulated genes.

A comparison with previous work on microarray analyses revealed a notable similarity in the functional categories of the genes found to be dysregulated in our analysis (Mandel *et al.*, 2004), confirming their importance in the pathogenesis of the disease (see Figure 11.5). Such is the case of the categories defense response, immune system process, death, cell communication or response to chemical stimulus. In addition, our analysis revealed other relevant functions, including metabolism, establishment of localization or regulation of biological processes that have not been previously associated with SLE or PAPS, and that may provide important clues about the pathogenesis of these diseases.

C. Regulatory pathways dysregulated in SLE and PAPS

It is believed that mutations in susceptibility genes that contribute to the pathogenesis of a given disease result in an altered expression and/or activity of genes regulated by them, thus revealing a particular molecular signature

[7] Gene Ontology Consortium `http://www.geneontology.org`

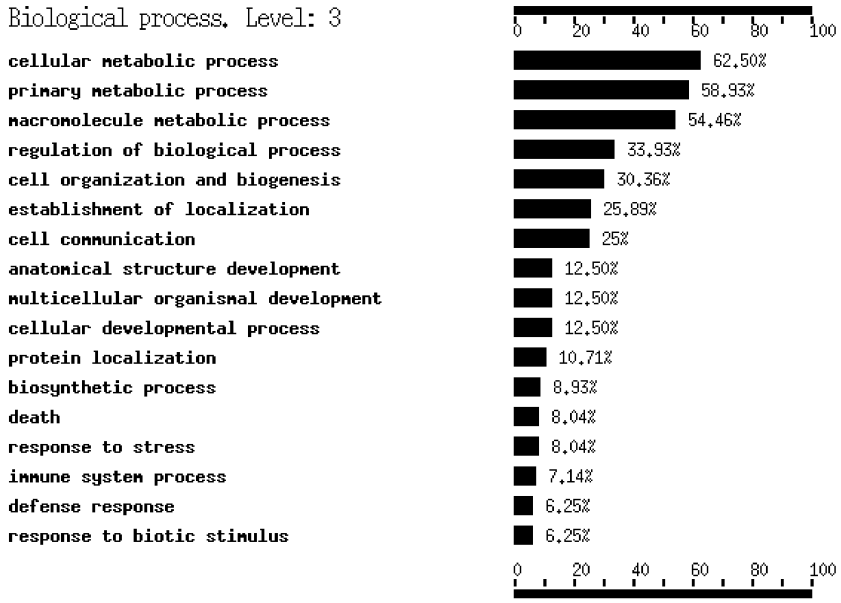


Fig. 11.5. Gene ontology biological process (level 3) annotations that are significantly overrepresented in the list of dysregulated genes. Annotations with an incidence level lower than 5% are not shown.

of the disease (Burmester and Haupl, 2004). Taking this idea into account, we searched for factors that could be regulating the genes whose expression is altered in SLE and/or PAPS. For each of the 150 genes identified in the original set, and using the Ingenuity Pathways Analysis tool[8], the factors that could be involved in the regulation of their expression or activity were identified. Only those genes with known regulators were considered for subsequent analyses. Furthermore, genes regulated by other genes included in the original set were discarded because it is not possible to know whether these genes are dysregulated due to a mutation in the genes that regulate their expression or because their regulators are dysregulated themselves.

The resulting filtered set includes a total of 129 genes (out of 150), and the gene set known to regulate them contains a total of 299 genes. Only the genes regulating three or more target genes were considered, which resulted in a final number of seventeen regulatory genes controlling the expression of a total of 45 dysregulated genes (see Table 11.9). Finally, their location within the genome was sought. Remarkably, nearly half of them (8 out of seventeen) were found to be located in chromosome regions previously reported as susceptibility regions for SLE (Johansson, 2004; Koskenmies, 2004; Tsao, 2003; Shai *et al.*, 1999;

[8] For a detailed description of Ingenuity Pathways Analysis, visit `http://www.ingenuity.com`

Functional group	Genes
Cellular metabolic process	GAS7, DDX5, RRAGD, MBD4, CECR1, CUTL1, PTMA
Primary metabolic process	SNRPD2, CPSF1, SSB, SFRS2IP, PPIG, NADK, DNPEP, HSD17B4, NAGPA, ZNF202, ABCA1, COMT
Macromolecule metabolic process	EIF4A1, EIF3S8, RPL18A, MRPL9, HSF2, HSPA6, PFDN4, HSPA8, HSP90B1
Regulation of biological process	KRAS, TMEM97, GPS1, UQCRB, CYB561, TNKS2, ZNF83, ZNF587, POLR2K, GMEB2, SP100
Cell organization and biogenesis	HIST1H1C, TSPYL4, H1F0
Establishment of localization	RIN3, SLC25A12, UQCRB, CYB561, SYPL1, GOSR1, KDELR2, C14orf108, GOLGA4, KPNB1, SLC25A5, PEX1
Cell communication	GNB2L1, CDC42EP3, IQGAP1, PKN1, RAB2, TAX1BP3, ARF3, ANK3
Defense response	WAS, TAP1, GNLY, NMI, CD160
Immune system processes	ISG15, IGHM, HLA-DQB1, GPSM3, MX1
Response to chemical stimulus	HSPA6, HSPA8
Death	BAG5, TNFRSF10B, CASP1, SIRT1

Table 11.8. Some of the GO functional groups identified from the relevant genelist and the genes that belong to each group.

Aringer and Smolen, 2004; Horiuchi *et al.*, 2006; Lee *et al.*, 2006; Croker and Kimberly, 2005).

Recent microarray reports have suggested that the interferon regulatory pathway could be an important contributor of SLE, based on the dysregulated expression of numerous interferon-inducible genes in lupus samples (Rozzo *et al.*, 2001; Baechler *et al.*, 2003; Bennett *et al.*, 2003). Remarkably, our analyses revealed that three of the regulatory genes are interferon proteins (IFNG, IFNA1 and IFNA2), regulating the expression and/or activity of nine genes identified in the microarray experiments. Seven of these nine genes were overexpressed in SLE patients, consistent with previous findings. Moreover, three more regulator proteins (IL15, MYC, TNFSF10) are also known to be regulated by IFNs. These results indicate that the IFN pathway regulates nearly half of the regulatory genes identified in our analysis, either directly or indirectly, corroborating the importance of the interferon signature in SLE, and suggesting an important role for this pathway also in PAPS pathogenesis.

Other regulatory genes with a known or suspected role in autoimmunity were also present in the search. Such is the case of PTEN, a phosphatase involved in the regulation of the PI3K pathway (Patel and Mohan, 2005), TNF, tumor necrosis factor (Hsu *et al.*, 2006) or the antiapoptotic protein Bcl-2 (Deming and Rathmell, 2006). It will be worth examining these regulatory

Regulator	Regulated genes	Locus	Reference
IL15	BTG1, GNLY, PFDN4, POLR2K, SELPLG, SLC25A5, SP100	4q31	
IFNG	ABCA1, CYB561, HSPA8, MX1, PSMB4, TNFRSF10B	12q14	
MYC	ARF3, DDX5, PPIA, SLC25A5, TNFRSF10B	8q24.12-13	(Johansson, 2004; Koskenmies, 2004)
TP53	COMT, DLG1, ISG15, THRAP2, TNFRSF10B	17p13.1	(Johansson, 2004; Koskenmies, 2004)
EGF	GNB2L1, HK1, MVP, WAS	4q25	
HGF	ANK3, ISG15, HK1, TMEM97	7q21.1	(Tsao, 2003)
TNF	ANXA2, BTG1, ISG15, MVP	6p31.3	(Aringer and Smolen, 2004; Horiuchi *et al.*, 2006; Lee *et al.*, 2006)
PTEN	BTG1, CYB561, ISG15, RPL36A	10q23.31	
TGFB1	ANXA2, KDELR2, KPNB1, TAX1BP3	19q13.1	(Johansson, 2004; Koskenmies, 2004)
IFNA1	CYB561, ISG15, MX1, NMI	9p22	
IFNA2	MX1, SP100, TNFRSF10B	9p22	
CDC42	IQGAP1, PKN1, WAS	1p36.1	(Koskenmies, 2004; Shai *et al.*, 1999)
BCL2	CASP1, IGHG1, KRAS	18q21.3	(Johansson, 2004; Shai *et al.*, 1999)
FOS	CPSF1, SNRPD2, HSP90B1	14q24.3	
TNFSF10	ISG15, SP100 , TNFRSF10B	3q26	(Johansson, 2004)
MYCN	EIF4A1, RPL37A, RPS25	2q24.1	
SRC	ACP1 , ANXA2 , GNB2L1	20q12-13	(Koskenmies, 2004; Tsao, 2003)

Table 11.9. Location of the detected regulator genes.

networks in more detail, to determine their contribution to the pathogenesis SLE or PAPS, as well as their usefulness as markers of these diseases.

D. Analysis of transcription factor binding sites in the promoters of genes relevant for SLE and PAPS identification

Genes that participate in a particular pathway often share a common transcription factor binding site. We next explored the possibility that the dysregulated genes in SLE and PAPS could be regulated by common transcription factors involved in the development of autoimmunity. We reasoned that if a significant number of dysregulated genes in SLE and PAPS were regulated by

a common transcriptional factor, then this factor could somehow be associated with the disease.

	IRF2	**IRF1**	**PAX2**	**SP1**	**MEF2**	**P300**	**E2F**	**CDXA**	**CREBP1-CJUN**
Frequency									
z-value	2.81	2.05	2.71	2.23	2.15	2.14	2.10	-2.33	-2.08
sample mean	0.061	0.078	0.026	2.278	0.035	2.148	0.157	2.470	0.009
population mean	0.021	0.039	0.006	1.824	0.013	1.857	0.095	3.234	0.055
ratio	2.9	2.0	4.2	1.2	2.8	1.2	1.7	0.8	0.2
p-value	0.0049	0.0408	0.0067	0.0256	0.0318	0.0323	0.0353	0.0200	0.0377
Incidence									
number	5	7	3	78	4	105	17	81	1
sample mean	4.35%	6.09%	2.61%	67.83%	3.48%	91.30%	14.78%	70.43%	0.87%
population mean	2.02%	3.79%	0.62%	68.09%	1.25%	82.54%	8.86%	79.47%	5.28%
ratio	2.2	1.6	4.2	1.0	2.8	1.1	1.7	0.9	0.2
p-value	0.0840	0.1474	0.0355	0.4043	0.0572	0.0060	0.0251	0.0059	0.0132

Table 11.10. Deregulated transcription factors found based on the identified relevant genelist.

We made use of the Transcription Element Listening System (Cole *et al.*, 2005), also known as TELiS, to identify transcription factor-binding motifs (TFBM) that are overrepresented in the promoters of a given gene set. This analysis considers two variables for any binding motif: (i) the frequency of this motif per gene: a comparison is made between the average frequency within the whole microarray and the frequency in the genes of the relevant list; (ii) the number of genes exhibiting this motif in the relevant list compared to the whole microarray. Using these values, it is possible to compute a z-value statistic (Kanji, 2006) and to perform a two-tailed hypothesis test based on a Bernouilli-set trials that examines which TFBMs are overrepresented in the test list with respect to the expected occurrence computed from the original microarray list.

In our case, the parameters for the genome scan were as follows: the promoter search interval was fixed between -1000 and +200 bp, and the stringency of the test is fixed to a 0.9 value. TELiS analysis identified a total of 115 genes from the total of 141 mapped genes in the relevant gene set. Within these parameters, TELiS reported a total number of seven overrepresented transcription factor binding motifs (see Table 11.10). Importantly, two interferon response elements (IRF1 and IRF2) appeared as overrepresented, again revealing the importance of the IFN-regulated pathway in these autoimmune diseases. The binding sites for SP1 and P300 were also significantly overrepresented, particularly with regard to the frequency of binding sites per gene. However, the increase in frequency as well as in incidence were minimal with respect to the reference control, and it is unlikely to be biologically meaningful.

Pax2 and Mef2 are transcription factors that are known to be involved in cellular differentiation and organ development (Berkes and Tapscott, 2005). The finding that their binding sites are overrepresented in our analysis, sug-

gests that factors regulating differentiation also play a role in autoimmunity. Remarkably, E2F binding sites were found overrepresented in this analysis. E2F constitutes a family of transcription factors involved in the transcriptional regulation of genes necessary for cell-cycle control (Nevins, 2001). Recently, functional inactivation of E2F2, a member of this family, has been found to promote a lupus-like autoimmune disease in a mouse model, linking cell cycle regulation to autoimmunity (Murga *et al.*, 2001). Additionally, reduced expression of E2F2 has been reported in SLE patients (Baechler *et al.*, 2003). These findings project a role for E2F in the regulation of autoimmunity, and suggest that modulation of E2F levels could be beneficial in these diseases.

11.3.7 Conclusions

Consensus approaches are alternative techniques that try to overcome the technology-intrinsic data noise in microarray experiments. Throughout this chapter, we have applied the supervised consensus gene selection method, aiming to add robustness to the biomarker identification procedures by means of Affymetrix DNA microarrays.

Microarray studies must deal with "the course of dimensionality" and "the course of sparsity" (Somorjai *et al.*, 2003): in a problem with a huge number of variables (features or genes), there are only a small number of instances (cases or samples) whereas there are several thousand variables. Therefore, their results must be strictly proven to assess reliability over the given statements. This assessment has been carried out by a successful in-depth statistical and biological validation.

It is important to stress the importance of being conservative when dealing with findings coming from a low number of samples. With some rare diseases, such as SLE and PAPS, it is very difficult for physicians and clinics to find samples cohorts. Studies in these fields must be able to deal with these adversities while they bring some light into the present genomic research. Of special importance is the posterior validation of the findings by means of qPCR analysis with outer samples not used in the previous statistical stages.

Among these findings, the statistical techniques applied have corroborated the importance of the IFN pathway in SLE and PAPS, and have also revealed the existence of other gene signatures that could be playing an important role in the pathogenesis of these diseases. Future clinical and/or biological tests over the presented results could throw light on the molecular basis of SLE and PAPS diseases.

11.4 Colorectal cancer biomarker discovery through gene interaction network induction

A *tumoral biomarker* is defined as a molecule that unveils the presence of a cancer, or, a molecule that provides relevant information about the possible

development of a tumour in the future. The first tumoral biomarker was detected in 1848 when Bence-Jones discovered the presence of a certain protein in a urine sample coming from a patient of osteomalacia (softening of the bones). A hundred years later, such protein was identified as the light chain of immunoglobulins (Edelman *et al.*, 1961). From that moment on, it was evident that there exist a series of proteins and glycoproteins directly produced by tumours that can be detected by means of immunological trials.

The presence of tumoral biomarkers can be due to the increase of a particular gene expression, to internal causes related to the cancer or to both of them. Biomarkers may be useful for one or more of the following purposes (Duffy, 2001):

- Early screening of a subjacent disease.
- Help on a diagnosis process.
- Clues on the prognosis (how the patient will progress) of a given disease.
- Prediction of a therapy efficacy.
- Monitoring of a patient who underwent surgery.

Despite all the research in this biomarker field, there are currently very few tumoral biomarkers clinically accepted as so. Over the last years, many new molecules have been proposed as new biomarkers (Crawford *et al.*, 2003; Allen and Johnston, 2005) but almost all should be still analysed and evaluated in depth.

The present chapter is intended to apply the reliable dependence detection presented in Chapter 9 to seek for these biomarkers. Not only is the identification of possible biomarkers of interest, but also the study of possible relations between them, in our case, those biological relations coming from the conditional dependences of the consensus networks.

11.4.1 Introduction

Colorectal cancer (CRC), also called colon cancer, is the third most common form of cancer and the second leading cause of death among cancers in the Western World. In Spain, 25,000 new cases of CRC per year are diagnosed, representing an incidence of 50 new cases per 100,000 inhabitants[9]. It constitutes the second cause of death in Spain, following lung cancer.

CRC is classified in hereditary, familial or sporadic colon cancer (Calvert and Frucht, 2002), which is the most common form of colon cancer, representing 70% of the diagnosed cases. It is the result of the accumulation of multiple mutations that affect tumor suppressor genes, as well as oncogenes or mismatch repair genes (Weinberg, 1994; Calvert and Frucht, 2002). Current prognostic models are based on histoclinical parameters, but sometimes are not accurate enough for prediction in individual patients. Despite the huge

[9] Information obtained from the Spanish Association Against Cancer, `http://www.aecc.org`.

amount of studies carried out on CRC, little is known about the molecular alterations, and no molecular marker has been validated for use as a new diagnostic or prognostic tool.

One of the most common classification systems to define in which stage the CRC is in a patient is the *Dukes staging* (Dukes, 1932). Originally proposed in 1932, this system placed patients into one of three categories (A, B and C). Later in 1954, Astler *et al.* (Astler and Coller, 1954) included another stage (D) and subdivided stages B and C into two more substages (B1, B2, C1 and C2). More recently, the system was augmented with two new substages (B3 and C3). Table 11.11 contains the medical description of each of these stages and Figure 11.6 graphically illustrates how a tumour usually progress within the colon wall into the different stages.

Stage	**Description**
A	Tumour limited to the *mucosa*.
B1	Tumour reaches the *muscularis propia*, but it is not invasive.
B2	Same as B2 but with an invasive behaviour.
B3	Same as B3 but the tumour has already invaded adjacent structures.
C1 C2 C3	Similar incidence as B1, B2 and B3 but with metastasis in the nodules.
D	Tumoral cells have invaded the blood or lymphatic strem and have produced distance metastasis.

Table 11.11. Augmented Dukes classification system for colorectal tumours.

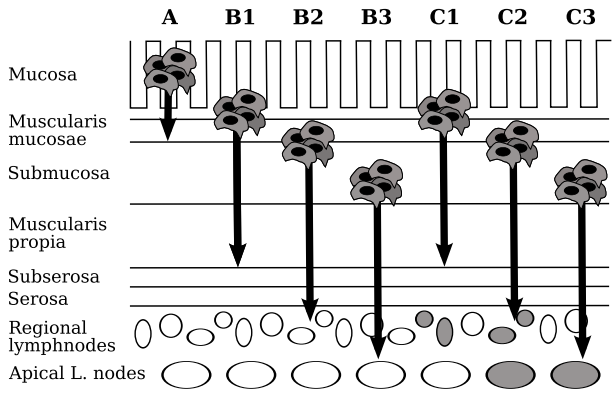


Fig. 11.6. Colorectal cancer classification following the augmented Dukes stage system.

When the colorectal tumour is detected in the early stage A the survival rate relative to the following 5 years is higher than 90%. Unfortunately, only 39% of the cases are detected in this stage. As the tumour evolves, the survival rate in the following 5 years decreases. In particular, for the B stages, this rate falls into 83-85% of the cases. If there exists metastasis (stages C), the drop down from 72% (C1 stage) to 44% (C3 stage). When the tumour has spread to other organs, stage D, the survival rate in the next 5 years is only 8% (O'Connell *et al.*, 2004).

Currently, the presented colorectal staging is based on clinicopathologic features such as bowell wall penetration and lymph node metastasis. Unfortunately these clinical staging systems often fail to discriminate the biologic behavior of a large number of tumors, resulting in the systematic overtreatment or undertreatment of patients with adjuvant therapies and it can only be applied after complete surgical resection. Recently developed microarray technology has permitted the development of cancer classifiers, identification of tumor subclasses, discovery of progression markers and prediction of disease outcome in many types of cancer. Unlike clinicopathologic staging, molecular staging has shown promise in predicting the long-term outcome of any one individual based on the gene expression profile of the tumor at diagnosis. Every tumor contains informative gene expression signatures that at the time of diagnosis can direct the biologic behavior of the tumor over time.

Gene expression data have proven highly informative of disease state, particularly in the area of oncology, where accurate and early diagnosis, followed by appropriate treatment, can prove critical. Studies on clinical samples have shown that gene expression data can be used not only to distinguish between tumour types, define new subtypes and identify misclassified cell lines, but also to predict prognostic outcomes (Golub *et al.*, 1999; Alizadeh *et al.*, 2000). Microarrays can be used in combination with other diagnostic methods to add more information about the tumour specimen by looking at thousands of genes concurrently. It not only classifies tumour samples into known and new taxonomic categories and discovers new diagnostic and therapeutic markers, but also identifies new subtypes that correlate with treatment outcome.

11.4.2 Sample processing

A total of 120 paired (from tumoral and distal non tumoral tissues) samples were processed. They were obtained with informed consent from 60 patients with sporadic colorectal tumours in the following different stages of development according to Dukes classification: B, C or D (substages B1, B2, B3 or C1, C2, C3 were respectively grouped together in B or C). Immediately after surgery, an anatomopathologic analysis was carried out on the samples to confirm diagnosis as well as tumour staging. Samples were collected in a tube containing RNA later solution to preserve the RNA from degradation and kept at $-80°$C until their use.

Total RNA was extracted from the 120 samples using the RNeasy mini spin kit (Qiagen). RNA quantity and integrity was determined by 2100 Bioanalyzer (Agilent Technologies). The RIN algorithm allows calculation of RNA integrity using a trained artificial neural network based on the determination of the most informative features that can be extracted from the electrophoretic traces out of 100 features identified through signal analysis. The RNA integrity number quantitative assesses the integrity of a given RNA sample (Schroeder *et al.*, 2006).

For solid tissues, RIN number is usually between 6 and 8 (Fleige and Pfaffl, 2006) and the suggestion is not to use a sample with a RIN number below a value of 5 (Wolber *et al.*, 2006). Following this recommendation, the lowest accepted value in the study was of 5.6, rejecting the samples that showed a lower RIN. In total, 32 tumoral and 42 non tumoral samples were available. Clinical features of the correspondent patients are listed in Table 11.12.

Patient	**Dukes'**	**Age**	**Sex**	**Patient**	**Dukes'**	**Age**	**Sex**
833030	D	46	F	1180611	D	83	M
1178604	B2	69	M	1028566	D	74	M
1179837	B3	68	M	69098	C2	71	F
986007	D	63	M	850184	C2	65	M
1017459	C1	87	F	118160	B2	57	M
1174501	C2	68	F	103933	B1	71	M
1050010	D	81	M	120680	B1	68	M
938226	D	77	F	118406	B1	59	M
622444	B1	73	M	78895	B2	71	M
1020521	C3	47	M	121154	C2	70	M
1179816	B2	71	F	121148	B2	76	F
1057760	B2	72	M	120524	C3	57	F
67363	C2	73	F	85035	B2	63	M
119781	B3	73	M	70576	B2	63	M
41072	B2	55	M	93318	–	–	–
36692	B2	66	F	134597	–	–	–
31980	B1	77	M	105224	B2	75	F
1179575	B1	46	M	20505	B2	60	M
736934	C2	47	M	12032	D	69	F
1186314	D	50	M	63823	C1	62	F
1180734	C3	67	F	121345	C2	67	M

Table 11.12. Clinical parameters of colorectal cancer patients included in the study. Sex: M, male, F, female. DUKES stages: localized tumour without nodes (B1, B2, B3), localized tumour with nodes (C1, C2, C3) and initial metastasis (D).

11.4.3 Experimental design and chip hybridization/scanning

The hybridization platform is the Agilent microarray Human 1A(V2) Oligo Microarrays that measures 22,574 probes simultaneously. We used one reference pool of non tumoral samples in one channel, while the second channel was formed by the individual tumoral and non tumoral samples. Previously referred to in Section 11.1, the experimental design based on pooling reduces the effects due to the proper biological samples variation (Churchill and Oliver, 2001). At the same time, a pooling design mantains the common behaviours on the transcripts' expressions (Kendziorski *et al.*, 2005). There are several works which use the same experimental design to research colon cancer samples (Birkenkamp-Demtroder *et al.*, 2002; Kim *et al.*, 2005; Zou *et al.*, 2002).

The 33 non tumoral samples selected to form the pool were chosen taking into account, along with the RIN quality, that they should have at least 2 g of RNA, then they were aliquoted and stored at $-80°$C. The pool NT was constituted by 2μg of total RNA from 33 of the non tumoral samples selected for the study. Aliquots for each round of hybridization were prepared and stored at $-80°$C. 32 tumoral samples (T) and 33 non tumoral samples (NT) were hybridized against the constructed pool (Pool NT). Thus, a total of 65 arrays were available, see Figure 11.7 for a graphical illustration of the comparisons.

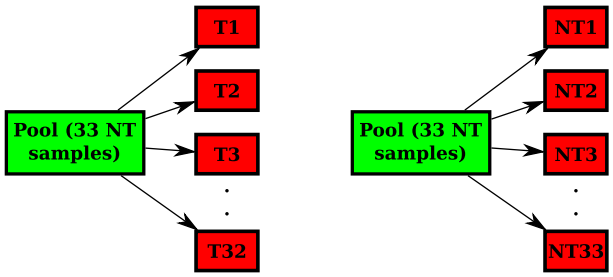


Fig. 11.7. Experimental design of the study. The microarrays were hybridized according to the above scheme.

Reverse transcription was performed on 500 ng of total RNA extracted to synthesize the first and second strands of cDNA using the Agilent Low RNA Input Fluorescent Linear Amplification kit. Next, the cRNA was created using T7 RNA polymerase which simultaneously incorporates the fluophores. The pool was labelled with Cyanine-3-CTP, whereas the tumoral and non tumoral samples were labelled with Cyanine-5-CTP. Labelled probes were measured in the spectrophotometer at two different wavelengths, at 550 nm the samples labelled with Cy3 and at 650 nm the ones labelled with Cy5. In order to determine the efficiency of the labelling, the ratio of the picomoles of cyanine

dye and μg of cRNA was calculated. Only those probes with a ratio between 10 and 20 were suitable to hybridize.

0.75 μg of cRNA labelled samples were hybridized onto the arrays using the Agilent in situ hybridization kit-plus. The arrays were placed inside the hybridization oven and we performed the hybridization reaction for 17 hours at 60°C. The slides were washed at room temperature in 20X SSPE and 20% N-laurylsarcosine and dried with the stabilization and drying solution provided by Agilent.

The hybridized arrays were scanned using the Axon GenePix 4000B dual laser slide scanning system at the wavelengths corresponding to each of the flurophore. The image was processed with the GenePix Pro 6.0 (Axon), with 10 μm resolution. Microarray internal controls named 'NA', 'NegativeControl', 'N/A', 'BrightCorner', 'Pro25G' and 'eQC' were removed from the raw data.

11.4.4 Probes quality preprocessing

Following the criteria described in Section 11.2.2, the first step on every preprocessing task on a microarray experimentation is to remove those probes whose values might not be reliable. To this end, the three criteria were computed for all the probes of the 65 available microarrays:

- The fluorescent intensity measurement quality,
- the background flatness quality,
- the signal intensity consistency quality.

Once the global quality metric was computed, the threshold for a probe to be accepted was set up on a 0.99 value. Taking this value into account, from the original 17,986 probes, only 11,104 probes were retained as being valid and with reliable values.

The next tackled step for preparing the data for a data mining process is to normalize the data values. Within the microarray analysis field, this normalization is referred to as the reallocation or smoothing of the data values on the basis of a biological fact: in a biological experiment, the number of genes to be dysregulated has to be low.

However, due to different physical and biological factors such as the systematic variations in the dyes properties, the efficiency of dye incorporation or the experimental variability in hybridization, this statement is systematically unfulfilled. Thus, so that biological differences can be more easily distinguished and comparison of expression levels between slides allowed, the intensity values must be corrected.

The most graphical tool to understand this need is the *MA-plot* (Dudoit *et al.*, 2002). An MA-plot graphically compares the intensity values of a dual channel array. The plot's axes are defined by $M = log_2(\mathrm{S_r}/\mathrm{S_g})$ as the ordinate and $A = log_2\sqrt{\mathrm{S_r} \times \mathrm{S_g}}$ as the abscissa. An MA-plot is basically a char of the dispersion level that the values present. Figure 11.8 displays an array MA-plot

before a normalization process and how the number of underexpressed genes is extremely high. The same Figure 11.8 shows how this defect is corrected after the normalization process.

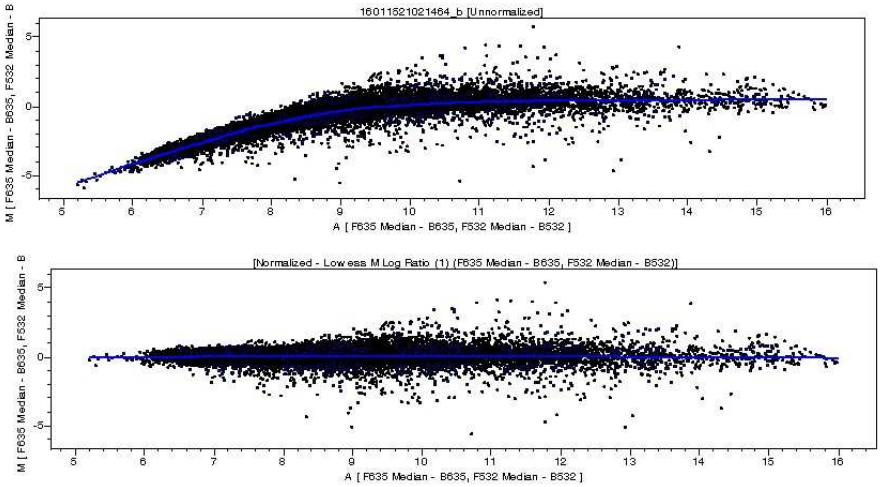


Fig. 11.8. MA-plots of one of the CRC microarrays in the study. The top chart corresponds to the unnormalized data, whereas the bottom chart shows the values after a normalization process.

There are different techniques to perform this normalization task: global normalization, scale to a given value, normalization with respect to control or housekeeping genes, statistical standardization, etc. Nevertheless, two techniques have gained interest and the microarray community consider both to be the gold standard: *Robust multi-array* (RMA) normalization (Irizarry *et al.*, 2003) and *locally weighted linear regression* (or *lowess*) normalization (Dudoit *et al.*, 2002; Yang *et al.*, 2002). Lowess normalization has been reported to retrieve low variance data and thus more reliable probe expressions (Zahurak *et al.*, 2007). Therefore, and since there is no consensus in the community of which of them is better, we chose to use lowess to normalize all the microarray data values.

Data coming from microarrays may also include lost or unknown values due to physical problems on the hybridization process. Figure 11.9 presents three examples of these bad hybridizations (all detected in our CRC data).

Usually, the scanning software is in charge of the automatic detection of these hybridization defects. The software identifies the problematic probes and its associated values are flagged with a lost-value identifier. Seldomly, these probes may be identified by the scanner operator and flagged manually. The presence of lost values is an obstacle to tackle any posterior data mining

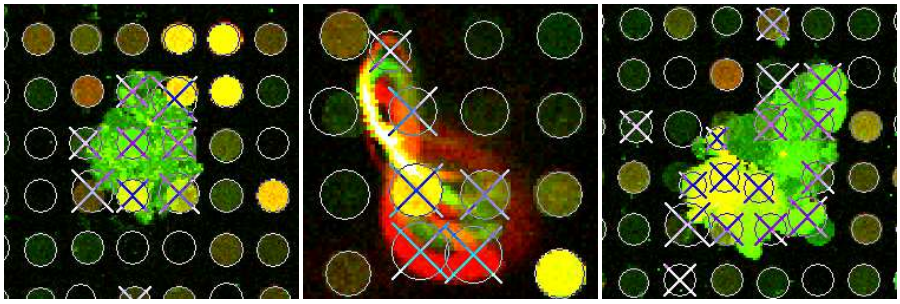


Fig. 11.9. Examples of physical problems on the hybridization process. Incorrect binding of the cRNA (left), external fiber inclusion in the array (center) and again an incorrect binding of the cRNA to the slide (right).

process. The great many machine learning and data mining techniques need to have a *complete* dataset, that is, without any lost value.

However, the practitioner should be aware that the completion of the lost values may bias the data if the number of lost values is high in comparison with the actual known values. Within our dataset, we found a total number of 17,147 lost probes throughout the 65 arrays, which only constitutes 1.47% of the values. This number suggested that the lost values *filling* process, commonly known as lost values imputation, was not substantially biasing the data.

Contrary to previous preprocessing steps, in the microarray lost values imputation problematic, one approach has gained the approval of almost all the research community: the *k-NN* imputation of (Troyanskaya *et al.*, 2001). k-NN impute is based on the classical k-Nearest Neighbors supervised classifier proposed by (Aha *et al.*, 1991). The imputation algorithm uses a similar scheme to the classification one. It looks for the closest k probes among all the array set (in terms of a given distance between all the values, usually euclidean). Then, a correlation coefficient is computed between the probe to impute and all the selected neighbours. Finally, the lost value is computed taking into account the neighbours values as a weighted mean value by the correlation coefficient.

Following the recommendations of its authors, the value of k was set in 15 neighbours and a total of 7,534 values were imputed. This number of probes corresponds to the lost values found in the probes that have already passed the quality criteria process.

The colon samples come from biopsias taken in open colon surgeries. This origin implies that there are several kind of tissues in each sample (e.g. mucosa, muscle or lymphatic cells among others). A side effect appears then: between samples grouped on the same phenotype there could be extreme differences in the expression profiles of some particular genes. This "chaotic" behaviour

is found to be of no interest because it is far from being reliable in a general study (Kitahara *et al.*, 2001).

Thus, following the physicians recommendations, the probes that showed an intra-class variability equal or higher than 2-fold were removed from the analysis. Starting with a valid set of 11,120 probes, a total of 3,016 were removed, keeping 8,104 probes which comprised the set of probes/genes/-variables/features of the supervised classification problem.

Section 8.2 discusses the discretization problematic when dealing with continuous values in biological researches. Presented in Secion 8 and put on stage in Section 11.3, the consensus gene selection is a way to alleviate possible data biases of the discretization policies. However, in the case of CRC dataset we decide to discretize the data only with one policy, equal width (Kerber, 1992), because of its unsupervised nature and fitness for gene expression data. As a general criterion on the gene expression studies, the number of functional stadios a gene can be is low (Causton *et al.*, 2003; Friedman *et al.*, 2000). In the case of arrays experiments this number is commonly set to three states: over expression, under expression or baseline (null) activity. Thus, borrowing this criterion, we considered that the most fitted discretization policy was a three-bin equal width discretization. The procedure is easy, each probe's continuous values are sorted and the range is divided into the number of bins (Tuzhilin and Adomavicius, 2002). In addition, it does not use the supervised information of the class separation, a fact that adds independence from the supervised classification problem.

11.4.5 Supervised classification approach

From the machine learning point of view, the translation of this experimental design into a supervised classification problem is straightforward. The supervised dataset will be comprised of 65 instances (one per hybridized array) with two values for the supervised variable: tumoral (arrays when a tumor sample (T) is compared against the NT pool) and non tumoral (each non tumoral sample is individually compared against the NT pool). The number of instances for each class respectively is thus of 32 tumoral and of 33 non tumoral ones (Armañanzas *et al.*, 2008a). Similar schemes in cancer studies have been previously used (Birkenkamp-Demtroder *et al.*, 2002; Kitahara *et al.*, 2001; Zou *et al.*, 2002).

This supervised modelling could be augmented by subdiving the tumoral instances into their corresponding Dukes' stadio. This way, the class variable should take four different states: non tumoral (NT), Dukes B stadio(B_state), Dukes C stadio (C_state) or Dukes D stadio(D_state). Within this augmented scheme, one instance must be removed because it comes from a Dukes A stadio and there were no others from the same stadio. Finally, the distribution could be as follows: 33 NT, 13 B_state, 10 C_state and 8 D_state instances respectively. Although of interest, this supervised scheme presents very few

instances in each state, a fact that may significantly penalize the generalization of the results. The previous dichotomic scheme is therefore used in the subsequent analysis.

11.4.6 Data analysis results

11.4.6.1 Running parameters

The methodological proposal introduced in Chapter 9 includes a set of running parameters to be fixed, principally the feature subset selection, a boundary for the maximum number of parents k for the k-dependence Bayesian classifier and the number of times that the boostrap loop is performed. Moreover, and especially in the microarray context, all these parameters are expected to set a scenario in which the running time could be affordable.

The feature subset selection technique was, as in the case of SLE and PAPS, the correlation-based feature subset selection (CFS). Once the dataset is reduced by the CFS, the Bayesian classifier to be learnt is a kDB with a k value of 4. This value allows the graphical models to be both flexible and not sparse when inducing the structures of dependences. Moreover, it implies a sufficient value so none of the possible relevant dependences could be outside the models. It has been tested experimentally that the kDB-θ does not add any improvement to this design within the microarray domain.

Finally, the proposed algorithm in Section 9.2.1 is repeated a thousand times, that is, the bootstrap parameter B is set to a value of 1,000. This way, we search for arcs that occur a number of times that can be widely considered as reliable.

11.4.6.2 Reliable gene-dependence network

After all the running pipeline finished, we found the following statistical results: the total number of probes that were selected as relevant at least once from the original $8,104$ set was of $|S(L_1)| = 1,723$. On the basis of these variables, the average number of arcs configured through all the induced kDB classification models –removing those that create cycles among them– was of $\overline{|L_1|} = 35.19$. Among them, the most times configured probabilistic relationship was the arc TCF3 $\rightarrow$ ENC1 that was included a total of 799 times from the 1,000 total runs.

The reduction range in the number of variables is up to 78.74% of the original set, and the number of arcs is consistent with this reduction. The high number of times that the arc TCF3 $\rightarrow$ ENC1 is included is noticeable, see Figure 11.10 for a complete view. At least 80% of all the feature selection runs selected both variables, and, within all those runs, 799 out of $1,000$ models induced this arc. This fact entails a very high degree of confidence to this dependency. In Section 11.4.6.4 the biological background of this finding and others are discussed in detail, showing a clear correspondence between the statistical models and the biological findings.

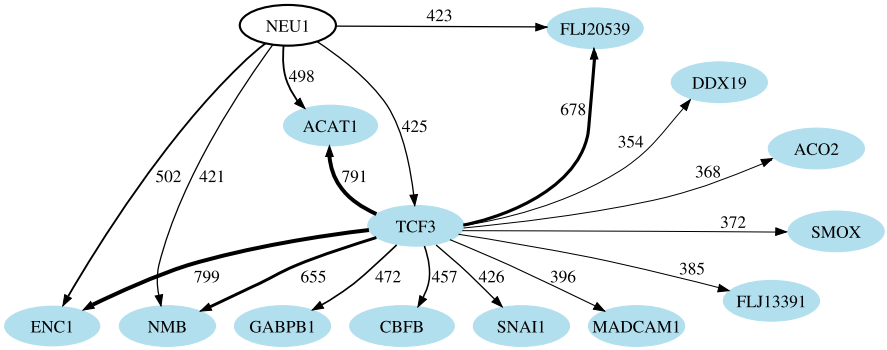


Fig. 11.10. Graphical structure of the high reliable dependences network for the CRC dataset and a t value of 350. Notice that each arc width is proportional to the associated number of times it was configured.

11.4.6.3 Classification accuracy

Although the priority of this analysis was to apply the reliable dependence detection algorithm, a reliable set of dependences can also be used in a pure classification application. For this purpose, firstly, the expert has to fix a certain value for the dependency threshold t to return the set of variables and arcs which surpass that level, obtaining a single model. This way, the complexity of the models can be tuned, assessing the scope, variables or aims of the study. After that, the class node is included in the model, adding arcs from it to the rest of the variables. This way the graphical structure is completed and the corresponding conditional probabilities are computed by their maximum likelihood estimators. Figure 11.11 represents the model structures for the CRC array sets for threshold $t = 400$, that is, each model contains the probabilistic relationships that have been jointly selected and configured 400 times at least.

As the confidence threshold falls, the sparsity degree of the models decreases and, thus, the number of variables to be evaluated increases. Therefore, it is of interest to study how the classification models evolve from the very simplest to the most dense ones. In order to analyse this effect, an evaluation of the classification accuracy of each model is performed. Due to the number of models to be evaluated, the total runs and the required computing time for the whole process, a five fold cross validation method is used to estimate the final classification accuracy. This estimation scheme was proven to be well suited for the microarray context (Bouckaert and Frank, 2004; Statnikov *et al.*, 2005), guaranteeing a fair and not overfitted accuracy percentage. For each fold, the run parameters are equal to the ones used in Section 11.4.6.1: a thousand bootstrap loops, CFS as multivariate filter method and a value of 4 for the kDB classifiers.

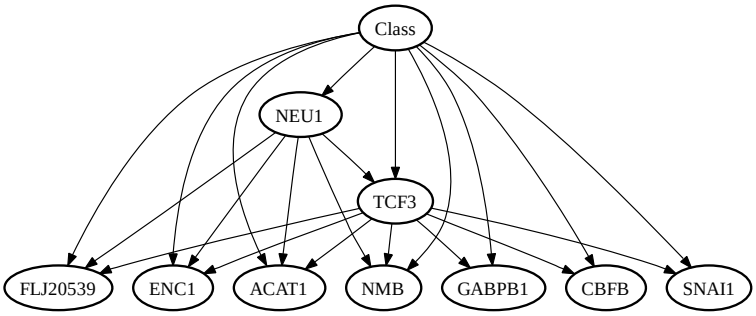


Fig. 11.11. Example of the graphical structures of the network classifiers configured from the high confidence dependences set: the corresponding threshold t is set at 400.

Table 11.13 gathers the number of selected variables, the total number of arcs induced in all the models, the number of times the most often retrieved dependence is recovered, and the maximum average accuracy achieved. Notice that the accuracies shown are jointly evaluated for a fixed confidence threshold. As a visual tool to study the tendency in classification, we have collected for each threshold the number of variables, arcs, mean accuracies and standard deviation in a single plot (see Figure 11.12). This Figure could be useful to decide to which degree of complexity a biologist is willing to analyse, taking into account the number of variables, arcs and the accuracy level that the model is able to reach.

	$\mathbf{Train_1}$	$\mathbf{Train_2}$	$\mathbf{Train_3}$	$\mathbf{Train_4}$	$\mathbf{Train_5}$	**Mean**	**Std**
CRC (8104 vars)							
$\lvert S(L_1)\rvert$	1223	1205	1188	1073	952	1128.2	114.62
$\overline{\lvert L_1\rvert}$	23.55	25.63	23.65	19.08	17.54	21.89	3.41
max t	689	380	433	439	323	452.8	140.11
max acc. ($t = 89$)	100	100	100	100	83.33	96.67	7.45

Table 11.13. Details about the number of variables and arcs for each cross validation fold. The cardinality of the highest configured arc is included.

Inspecting these results shows that there is no direct relationship between the number of arcs/variables and the model's accuracy. Figure 11.12 illustrates how, despite the addition of new arcs and thus more variables, there is no guarantee that the accuracies of a more complex model would be higher than the ones from a simpler model. There is a nuclear set of variables/arcs that is able to work out a high degree of the classification separability: more complex models do not necessarily correspond with higher accurate models. At a level of $t = 310$, the estimation of the accuracy with only four variables and three arcs achieves a mean value of 96%. This fact corroborates other

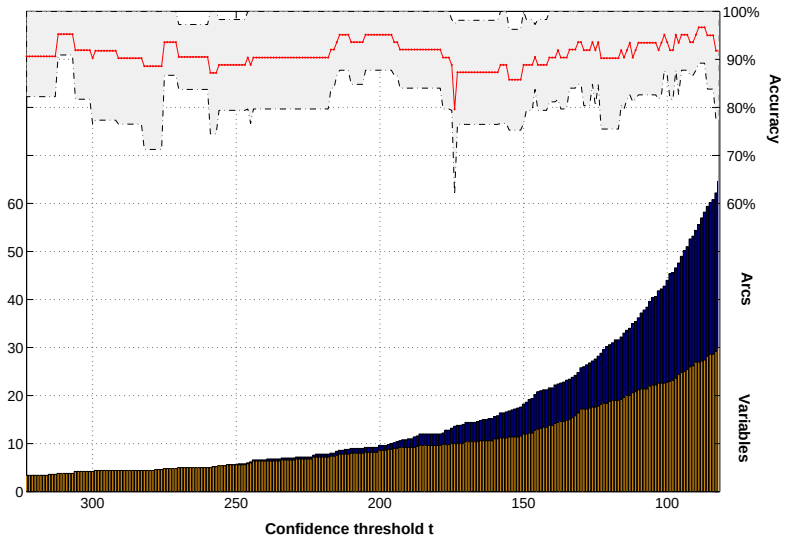


Fig. 11.12. Estimated accuracy tendency over the CRC array set. Mean accuracies are presented with their associated standard deviation for each confidence threshold, as well as the number of variables and edges included for that threshold.

studies regarding gene expression classification based on a reduced number of genes (Wang *et al.*, 2007; Baker and Kramer, 2006; Li *et al.*, 2004).

The low number of instances in the test set of each fold forces the mean accuracy to have a high level of standard deviation. Thus, accuracy percentages for each array set do not improve the state-of-the-art error rates, but clearly show that recovered high confidence structures are also able to clear up a significant piece of the phenotype information. All these genes and dependences can be of great interest to reveal new underlying biological knowledge.

11.4.6.4 Biological discussion

The graphical dependency structure reported in Figure 11.11 gathers a total of nine genes given a t threshold of 400. From all of them, TCF3 results in a kernel gene that shows dependences with all of the rest of the genes. This finding perfectly matches the biological function of TCF3, which is the transcription factor 3 or E2A immunoglobulin enhancer binding factors E12/E47. TCF3 coordinately regulates the expression of genes involved in cell survival, cell cycle progression, lipid metabolism, stress response, and lymphoid maturation (Schwartz *et al.*, 2006).

In the downstream dependences we find the gene FLJ20539, also known as GBP. (Lee *et al.*, 2001) describes a physiological regulation of [beta]-catenin stability by TCF3 and CK1epsilon. Moreover, another of the reported

genes, CBFB encodes a protein that belongs to the beta subunit of a heterodimeric core-binding transcription factor belonging to the PEBP2/CBF transcription factor family which master-regulates a host of genes specific to hematopoiesis (Bayly *et al.*, 2004) (e.g. RUNX1) and osteogenesis (e.g. RUNX2). The expression of CBFB is down regulated in a significant portion of gastric cancer cases, which may be involved in gastric carcinogenesis (Sakakura *et al.*, 2005). In addition, several studies suggest that lack of RUNX3 function is causally related to the genesis and progression of human gastric cancer, but potential roles of other members of the RUNX family genes have not yet been reported. Furthermore, CBFB, the gene encoding the cofactor of RUNX1, -2, -3, was also downregulated in significant fraction (32%, $p < 0.05$). The percentage of downregulation of RUNX1, RUNX3 and CBFB increases as the cancer stage progresses. All these findings and relationships constitute a serious biological hypothesis between the activity of CBFB and the gastric or colorectal carcinogenesis.

For its part, SNAI1 gene is present in activated mesenchymal cells indicating its relevance in the communication between tumor and stroma and this fact suggests that it can promote the conversion of carcinoma cells to stromal cells (Francí *et al.*, 2006). Its expression in colorectal tumors is also associated with downregulation of E-cadherin (CDH1) and vitamin D receptor gene products (Peña *et al.*, 2005a). The work by Takahashi et al. (Takahashi *et al.*, 2004) demonstrated that inhibition in SNAI1 is directly induced by TCF3. In mice, a human ortholog of human TCF3 is reported as a direct sequence-specific activator of negative vitamin D response element (Murayama *et al.*, 2004), which clearly supports the SNAI1 findings in colorectal tumors.

Another dependence found by our method shows a relation between NEU1 and the transcription factor TCF3. The protein encoded by NEU1 encodes the lysosomal enzyme, which cleaves terminal sialic acid residues from substrates such as glycoproteins and glycolipids. Upregulation of the NEU1 expression is important for the primary function of macrophages and there is a link between NEU1 and the cellular immune response (Liang *et al.*, 2006): data show that the differentiation of monocytes into macrophages is associated with the specific up-regulation of the enzyme activity of NEU1 (Stamatos *et al.*, 2005). Greenbaum et al. (Greenbaum *et al.*, 2004) reported that TCF3 is a negative regulator of a set of genes involved in the development of B lymphocites, thus, showing the link between TCF3 and NEU1.

Table 11.14 gathers the summary of the dependences that have been previously reported by biological works. Notice that the confidence levels for these arcs are very high ($t = 400$), which corroborates the reliability of these results. From the rest of these genes, two of them are directly related with colorectal cancers: ENC1 and NMB. ENC1 (ectodermal-neural cortex) belongs to the p53-induced gene set and it is also known as PIG10 gene (Polyak *et al.*, 1997). The influence of this PIG with colorectal cancer was firstly published by (Fujita *et al.*, 2001).This work states that ENC1 is regulated by β-catenin/TCF pathway and its altered expression contributes to colorectal carcinogenesis by

suppressing differentiation of colonic cells. NMB (neuromedin B) is associated with eating behaviors and obesity (Bouchard *et al.*, 2004); NMB and its receptor are coexpressed by proliferating cells in which they act in an autocrine fashion with similar and modest potency in both normal and malignant colonic epithelial cells (Matusiak *et al.*, 2005).

Dependence	**Confidence level**	**Reference**
TCF3 → FLJ20539	$o_{ij} = 678$	(Lee *et al.*, 2001)
TCF3 → CBFB	$o_{ij} = 457$	(Bayly *et al.*, 2004)
TCF3 → SNAI1	$o_{ij} = 426$	(Takahashi *et al.*, 2004)
		(Murayama *et al.*, 2004)
NEU1 → TCF3	$o_{ij} = 425$	(Greenbaum *et al.*, 2004)

Table 11.14. High confidence ($o_{ij} \geqslant 400$) interactions reported by both our method and also by the biological literature for the CRC array set.

The last two genes are not yet related to the cancer field. ACAT1 (acetyl-Coenzyme A acetyltransferase 1) is associated with the alpha-methylacetoaceticaciduria disorder, an inborn error of isoleucine catabolism characterized by urinary excretion of 2-methyl-3-hydroxybutyric acid, 2-methylacetoacetic acid, tiglylglycine, and butanone. The length of ACAT1 is approximately of 27 kb and contains 12 exons. Due to this high dimension, many mutations have been found for this gene (Fukao *et al.*, 2003) and many of them are under study (Zhang *et al.*, 2006). Regarding its colon activity (Ancona *et al.*, 2006), ACAT1 has been shown to play a pivotal functional role in the intestinal absorption of cholesterol, the hepatic secretion of VLDL, the biosynthesis of steroid hormones, the production of cholesterol esters in macrophages in atheroma and the secretion of biliary cholesterol (Smith *et al.*, 2004). Lastly, GABPB1 (GA-binding protein transcription factor, beta subunit) stimulates transcription of target genes. The encoded protein may be involved in activation of cytochrome oxidase expression and nuclear control of mitochondrial function. Biologists have identified multiple transcript variants encoding distinct isoforms of the protein. All of this suggests a general purpose compound that may be found in many biological processes.

11.4.7 Conclusions

A gene interaction network represents information in a richer way than univariate lists of genes. It describes groups of closely connected genes, unveiling biological knowledge or work hypothesis for both biologists and physicians. Hypothesis driven studies can converge with this data driven technique: it opens the possibility to study how a given gene or dependence interacts with the rest of the genes included in a study. This way, a beforehand hypothesis could be corroborated by a 'blind' data mining approach.

The biological analysis of the results for the CRC array set has proven a flawless correspondence between our method's findings and the evidence found in the biological state-of-the-art. Besides, reported results have also shown the potentiality of the induced models in a pure classification task. Reduced sets of dependences/variables are able to achieve a competitive degree of accuracy when performing a class-discrimination procedure, corroborating previous statements in the microarray analysis field.

As important as the accomplishment of previous hypothesis is the pointing out of new research targets. This knowledge discovery application brings into focus a new set of tools to help understand complex diseases that show relationships of different degrees among the involved genes. Results of this analysis and the analyses of the augmented experimental design (see Section 11.4.5), jointly with an in-depth biological discussion and validation, have been submitted to the European Patent Office (García *et al.*, 2008).

11.5 New insights in multiple sclerosis on the basis of a micro RNA bioinformatic analysis

The following study presents a computational biology research in a pioneering domain: the micro RNA regulation of the gene activity in a complex disease such as multiple sclerosis or MS. In this study, we report the expression study of 364 miRNA from three different phenotypes: MS patients during relapse, MS patients during remission and healthy controls. The main aim of this research is to enlighten whether miRNA activity interacts with the regulatory mechanisms of the MS stages.

The work database comprises quantitative expressions of 364 miRNA in the different available samples. These expression values were obtained by means of qPCR using a TLDA system from Applied Biosystem (Section 11.5.3 includes the technical details). Roughly speaking, the TLDA (Taqman Low Density Array) is a high-throughput platform that is able to simultaneously perform 364 different qPCR reactions in a microfluidic device. This kind of device allows the practitioner to simultaneously analyse a number of RNA sequences that were unfeasible to tackle years ago.

As in the case of DNA microarrays, many of the sequences included in the TLDA device won't be related with the disease under study. To seek the relevant sequences, a machine learning approach is again mandatory. The results obtained suggest the importance of some miRNAs in the molecular mechanisms implicated in multiple sclerosis. Even more, the machine learning approach opens new work hypothesis to keep researching on the relation between miRNA and MS (Otaegui *et al.*, 2009).

11.5.1 Introduction

Multiple sclerosis is an autoimmune demyelinating disease of the central nervous system (CNS). It begins most commonly during late adolescence, young adulthood, or mid-life, and it is one of the most incapacitating diseases in this age range.

MS causes attacks of neurological dysfunction (loss of vision, difficulty in walking or moving a limb, vertigo, loss of sensation) or progressive dysfunction in these same areas. These attacks, also known as *relapses*, typically last for a few days, and resolve spontaneously. However, patients may not always completely recover from an attack and are sometimes left with a disability. Although most patients experience attacks with little or no progressive disability, approximately 10-15% have progressive symptoms from onset, called primary progressive forms. Furthermore, more than 80% of patients will ultimately develop progressive symptoms after a prolonged period of exacerbations, usually after 10-20 years.

Etiologically, MS is a complex disease in which both genetic and environmental factors play a role. The genetics of MS is also complex without a clear inheritance pattern. The most relevant candidate genomic region is the HLA system (Haines *et al.*, 1996; Sawcer *et al.*, 1996; Oksenberg and Barcellos, 2005), although several other genes are currently being described as important risk factors involved in MS, as for example IL2RA (Alcina *et al.*, 2009) or IL7R genes (Gregory *et al.*, 2007).

Gene expression profiling has been a useful tool to provide information about the molecular pathways involved in MS pathogenesis (Achiron *et al.*, 2004; Baranzini *et al.*, 2005b; Ramanathan *et al.*, 2001). Several new studies have identified different expression patterns between relapses and remission (Otaegui *et al.*, 2007; Satoh *et al.*, 2008) showing that this clinical differentiation of two states of the disease also has a molecular correlation. Besides gene expression activity, it has been recently predicted that also micro RNA molecules (miRNA) may regulate around 30% of all cellular mRNA, so they should play a critical role in virtually all cellular functions (Lewis *et al.*, 2005).

Although misregulation of miRNA expression has been characterized mostly in cancer, it has lately been studied in many other diseases. In these studies, miRNA has been proposed as a regulator of immune cell development (Baltimore *et al.*, 2008), playing a role in the inflammatory response (O'Connell *et al.*, 2007), and as a key player in the pathogenesis of neurodegenerative diseases (Nelson *et al.*, 2008).

11.5.2 Study participants

All patients were recruited by the Neurology Department of Hospital Donostia, located in the region of Gipuzkoa (the Basque Country, Spain). The study was approved by the local institutional review board and all the samples were obtained with the written informed consent of the subjects. The patients were

diagnosed as having MS according to the Mc Donald Criteria (McDonald *et al.*, 2001; Poser, 2006).

As a first sample group (group A), 21 blood samples were obtained: 9 from patients in remission, 4 from patients during a relapse before the administration of steroids and 8 from healthy volunteers. Total RNA, including miRNA, was extracted from these samples to carry out the micro RNA expression study. Demographic information of the samples is presented in Table 11.15. Blood extraction was always performed in the early morning and RNA extraction was carried out no more than 2 hours after the blood was collected.

Status	**Stage**	**Age**	**Sex**	**TEV**	**Onset**	**EDSS**
Control	–	31	Male	–	–	–
Control	–	31	Female	–	–	–
Control	–	25	Female	–	–	–
Control	–	33	Female	–	–	–
Control	–	29	Female	–	–	–
Control	–	22	Female	–	–	–
Control	–	25	Female	–	–	–
Control	–	32	Male	–	–	–
MS	Remitting	45	Female	22	23	2.5
MS	Remitting	37	Female	7	30	1.5
MS	Remitting	54	Female	11	43	2
MS	Remitting	45	Female	12	33	6
MS	Remitting	69	Female	22	47	2
MS	Remitting	41	Female	20	21	3.5
MS	Remitting	38	Female	2	36	2
MS	Remitting	33	Male	11	21	6
MS	Remitting	45	Male	5	40	2
MS	Relapse	35	Male	1	34	3
MS	Relapse	53	Male	10	43	4.5
MS	Relapse	38	Male	11	27	3.5
MS	Relapse	46	Female	19	27	5

Table 11.15. Clinical description of the group A cohort of individuals. Columns *Status* and *Stage* show if the individual is a control or if he/she is a MS patient, in that case, the disease stage is included. Within the MS patients, column *TEV* displays the number of years or time of evolution. Column *Onset* includes at what age the patient had the first attack and column *EDDS* presents the expanded disability status scale or EDSS (Kurtzke, 1983).

Two other cohorts of samples were collected from other non-related groups. All these samples were used to independently validate some of the results obtained from the expression analysis of group A samples (see Section 11.5.6). Group B[10] includes mRNA samples of 42 individuals, 14 in remitting stage, 13 in relapse stage and 15 healthy controls. The last group (referred to as

[10] Clinical data from groups B and C is not available.

Group C[10]) is composed of miRNA samples extracted from 14 individuals, 3 in remitting, 4 in relapse and 7 controls.

11.5.3 RNA extraction, reverse transcription (RT) and quantitative PCR (qPCR)

Total RNA was extracted from blood using the AM1923 Ambion Leucolock kit working with the alternative protocol so as to keep the small RNA fraction. The RNA obtained was quantified in triplicate using a NanoDrop spectrophotometer.

A common bias in the interpretation of the miRNA profiles from whole blood may be introduced by the high concentration of miRNA from erythrocytes (Chen *et al.*, 2008b). In order to avoid such a bias, we firstly filtered out the blood samples, keeping only the peripheral blood mononuclear cells or PBMCs –a PBMC is a blood cell that has a round nucleus, like the lymphocites or the monocytes–. After this separation, the RNA purification was done.

In the case of group B, the samples came from RNA samples that are being collected systematically at the Hospital Donostia (San Sebastián, Spain). They were extracted using a Versagene TM Kit. This kind of extraction method entails the loss of the small molecules of RNA, e.g. miRNA. Therefore, in order to detect those possibly lost miRNA, cDNA was synthesized from total RNA using a Multiplex RT for Taqman array kit. This kit consists of 8 predefined RT primer pools containing up to 48 RT primers each. Each of these 8 pools contains the same endogenous control (RNU48). Unfortunately, this technology has been developed to detect only full length mature miRNA but not their precursors or their partially-degraded products.

We performed qPCR using the Taqman Low Density Array (TLDA) Human MicroRNA Panel v1.0 from Applied Biosystems. This TLDA includes 365 miRNA assays plus an endogenous control. The qPCR was performed using an Applied Biosystems 7900 Sequence Detection System. CT values were determined using the automatic threshold in RQ manager v1.1 analysis software.

Two normalization steps were used: the first normalization consisted of loading, for all the pools, the same quantity of template RNA and, the second, of normalizing the data against an endogenous gene. This endogenous control (RNU48) was chosen for this study as the least variable of all endogenous genes included in the TLDA assays. Consequently, data collected from each sample was independently normalized using the associated RNU48 expression, avoiding in this way bias in the results.

Relative quantification of miRNA expression was calculated with the $2^{-\Delta\Delta CT}$ method (Applied Biosystems, 2001). Quality of the data and quantification was computed using Real-Time Statminer software[11].

[11] More info about the Statminer software is available at `http://www.integromics.com`.

11.5.4 Supervised experimental design

Understanding the mechanisms by which MS triggers new episodes or attacks is a crucial task in the medical field. There is no clear evidence of how or when a remitting MS patient may develop the next crisis. In this scenario, an experimental design which could mix the three phenotypes at the same time could lead to unclear conclusions rather than enlighten some biological mechanism. This is the reason why the experimental design was divided into two dichotomic supervised problems: the comparison of the miRNA profiles of relapse and remitting patients (relapse *vs* remitting) and the comparison of the miRNA profiles of remitting patients and healthy controls (remitting *vs* control).

The aim of the first approach is to look for dysregulations in the miRNA activity between the MS patients. Some of the dysregulated miRNA could be involved or be part of other hidden biological mechanisms that may give clues about the relapse of the disease. The second approach is focussed on identifying differences between a healthy miRNA expression and the miRNA expression of a MS patient when the disease is *slept* or *latent*. Again, a biological clue at the level of miRNA could point to other major body dysfunctions.

In order to get a system biology and multivariate view of these two comparisons, we undertook the construction of highly reliable dependence networks using the algorithms presented in Chapter 9. The group A of samples was used for inducing the gene networks. Samples of groups B and C were used as independent samples to biologically validate the remarkable results from the networks.

11.5.5 Data analysis results

11.5.5.1 Running parameters

The running parameters for building the gene networks were configured with values similar to the ones used in the application to the CRC dataset (see Section 11.4.6.1). That is, the feature subset selection is set to the correlation-based feature selection (CFS) (Hall and Smith, 1997) and the value of k for the kDB classifiers was set to four.

The only difference was the number of iterations that the bootstrap loop performs (B). Since the number of samples was very low (4 relapse, 9 remitting and 8 control), we set B at 10,000 bootstrap iterations, trying to avoid, as far as possible, false positive dependences.

Similarly to the CRC analysis of Section 11.4, the quantitative expression values were discretized by the equal width procedure. The number of bins was as well configured at three bins (see Chapter 8 and Section 11.4.4 for more details).

11.5.5.2 Reliable gene-dependence networks

On average, 45% of the analyzed miRNAs were expressed in all the experimental samples. The results from the reliable dependence algorithm show diverse network topologies. However, the highest cardinalities for the confidence thresholds are not so high, taking into account that the process was repeated ten-thousand times.

Figure 11.13 shows the network structure of the comparison between remitting and control samples when the confidence threshold t is brought down to 500. The topology of the network suggested at first sight that the miRNA with id miR_96 had a key role in a possible jointly regulation.

By exploring the same confidence level in the comparison between relapse and remitting samples, we found a more complex dependence structure than in the previous case. Figure 11.14 presents that network, where the miRNA labeled as miR_18b seems to be playing a prominent role.

The field of miRNA is still under heavy development and the research is still in its adolescence. In the case of coding genes, there is a wide range of possibilities to research these dependences. However, in the miRNA case, the discussion on the biological translation of the statistical dependences is currently unaffordable. However, the induction of reliable interactions not only points to possible co-regulations, but also identifies which are the most relevant variables of the domain.

A total of six miRNAs were selected for an in depth biological study and validation of their biological relevance. From the network in Figure 11.13, we selected four miRNA, namely miR_148a, miR_184, miR_193a and miR_96. miR_184 and miR_193a presented the highest robustness values on their associated dependences with miR_96 (1,557 and 1,358 respectively). From the network topology, we also included in the analyses miR_148a because of its possible downward activity.

In the case of Figure 11.14 the selection was not so straightforward. As previously mentioned, miR_18b shows to have an important role in the network structure and, thus, it was also included in the list of candidate miRNAs to explore. Moreover, the most interesting fact was that, inspecting its expression values, we found that its activity showed an increasing expression in the relapsing group compared with its expression in the control samples. Similarly, the second miRNA selected from this network was the miR_599. This miRNA showed expression both in the relapsing and remitting groups but not in the control samples.

To ensure the selection of these last two miRNAs, we performed an expression validation on an independent cohort of 14 unseen samples (group C). The qPCR was performed in a 7900 sequence detection system using pre-designed Applied Biosystems Taqman probes. The selection was reinforced by the fact that miR_18b and miR_599 were four and five times more over-expressed in the relapse group than in the controls.

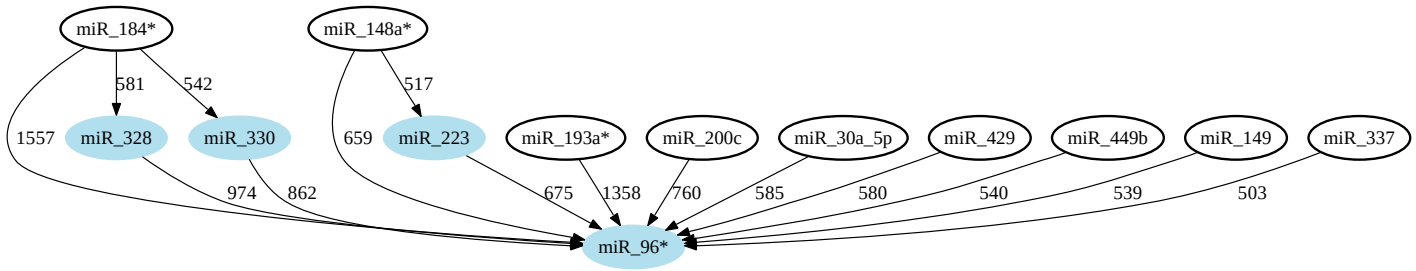


Fig. 11.13. Interaction network obtained for the comparison between qPCR expression of remitting and control samples. The confidence threshold t is fixed to a value of 500. The miRNA marked with an asterisk are biologically discussed.

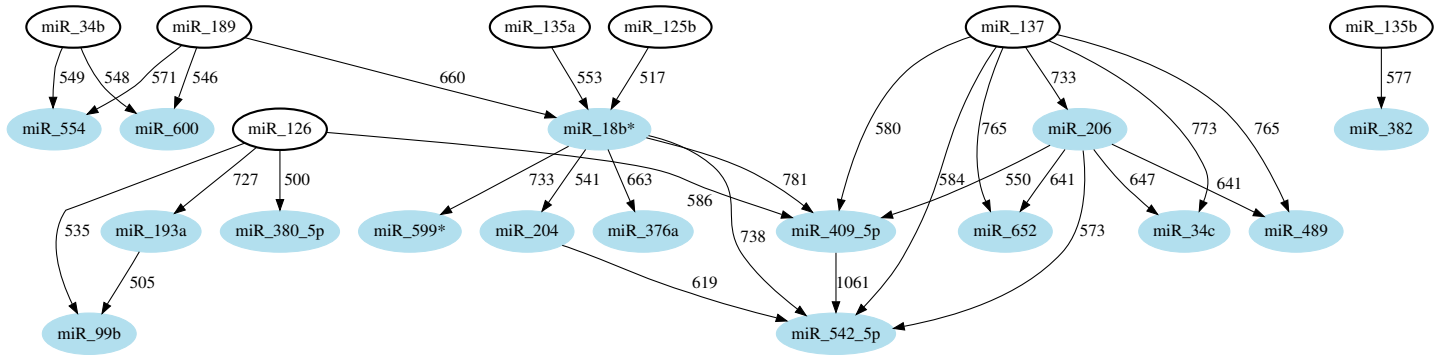


Fig. 11.14. Interaction network obtained for the comparison between qPCR expression of relapse and remitting samples. The confidence threshold t is fixed to a value of 500. The miRNA marked with an asterisk are biologically discussed.

11.5.6 Biological validation of the results

The function of mature miRNA molecules is to down-regulate gene expression. This is accomplished because their short strands (21-23bp) are partially or totally complementary to one or more mRNA molecules. It is possible therefore to establish regulation relationships between each miRNA sequence and a gene sequence (or target gene). By the comparative analysis of the genomic sequence, these associations are retrieved in a computational way. However, there are no clear criteria to state which miRNA could bind with which target gene. As a result, there are different genomic browsers that have been proposed to inspect the possible target genes.

In order to provide a biological interpretation of our findings, we searched the predicted targets of each of our six relevant miRNA in three different databases: *miRBase* targets v5 (Griffiths-Jones, 2004; Griffiths-Jones *et al.*, 2006), *TargetScan* v4.2 (Lewis *et al.*, 2003, 2005; Grimson *et al.*, 2007) and *Pictar* (Krek *et al.*, 2005). As expected, the results reported large differences in the possible target associations. Table 11.16 lists the number of predicted targets for the six miRNA according to each database (in the case of miRBase, two different searches with different confidence thresholds were performed).

	miRBase		**Pictar**	**TargetScan**	**Common**
	$p < 0.05$	$p < 0.005$			
miR_96	909	361	698	592	57
miR_184	819	289	22	17	3
miR_148a	918	353	429	434	46
miR_193a	819	362	134	208	14
miR_18b	775	327	151	149	14
miR_599	783	185	–	173	11

Table 11.16. Predicted target genes for the selected miRNA reported by three different databases. Column *Common* represents the number of targets jointly predicted by the three databases.

Theoretically, these miRNA should inhibit the expression of a certain number of target genes. The databases offer predicted information about the targets, but there are few experimental results to support it. So, to be as conservative as possible, in our analysis we selected as target genes only the common results from the three different prediction algorithms (in the miRBase case we selected the $p < 0.005$ column).

To validate the obtained results, we checked the expression in blood of four of these target genes in an independent cohort of 42 unseen samples (group B). The expression was analyzed by qPCR using SYBRgreen as fluorescent and pre-designed primers from geneglobe (`http://www.geneglobe.com`). The assay codes can be found in Table 11.17. We studied the expression of genes ARHGEF12, CELSR2, TAOK3 and GAB1. Table 11.17 presents the

miRNA(s) associated to each gene and the group in which it is expected to be down-regulated.

Gene	GeneID	micro RNA	Down in	Assay code
ARHGEF12	23365	96 / 148 / 193	Remitting	QT00006762
CELSR2	1952	96	Remitting	QT00010948
TAOK3	51347	599	Relapse	QT00059843
GAB1	2549	18b	Relapse	QT00014154

Table 11.17. Target genes studied with their gene ID, the miRNA(s) that binds to the gene, the group in which these genes are expected to be down-regulated and the Geneglobe Assay code.

Unfortunately, the results reported no statistical differences in the expression pattern of the four genes. However, most of the miRNA targets are predicted from bioinformatic analysis and have not yet been validated in biological studies. The fact that the hypothesis of under expression influence of the miRNAs is not directly validated by these four genes could be due to many factors. Notice that these four genes were randomly selected from the set of 145 possible common target genes. In addition, the measurement in blood may be inappropriate to verify their possible repression because not all dysregulated transcripts can be measured in blood. Another possible cause could be a regulation of the miRNA at the translational level rather than at the expression level.

In order to go beyond these limitations, we carried out another experiment, this time taking all the target genes into account. Since miRNA are highly conserved across species (Weber, 2005; Ibañez-Ventoso *et al.*, 2008), we used the murine EAE model to validate our findings. To this end, we mined a large multi-tissue, longitudinal gene expression profiling dataset in mouse EAE lymph node (Otaegui *et al.*, 2007) and spinal cord (Baranzini *et al.*, 2005a), focusing on the same target genes as those reported as common in Table 11.16. In order to check whether our selected target genes were really related with the disease, we randomly picked as control a group of 11 miRNA from those that were not differentially expressed in our first analysis. As in the case of the six relevant miRNA, we obtained the target genes of the new control 11 randomly chosen miRNA. The aim is to study how the expression of these target genes is significantly different between those associated with the relevant and those associated with the control miRNAs.

We checked the expression of all these target genes at the peak of the disease and after it. The expressions were then classified in three groups: up-regulated, down-regulated and equally-expressed. Table 11.18 shows the percentage of genes that were grouped in each of the three categories for both groups of target genes (*experiment* or *chance*).

Results of Table 11.18 show that the target genes associated to our relevant miRNAs are significantly more dysregulated than the target genes of 11

	EAE lymph node			EAE spinal cord		
	Up-reg	Down-reg	Equal	Up-reg	Down-reg	Equal
Experiment	15%	28%	57%	24%	29%	47%
Chance	2 %	8%	90%	11%	12%	77%

Table 11.18. Percentage of target genes that showed up-regulated, down-regulated or equal expression profiles within the two animal models under consideration. Row *Experiment* refers to the target genes associated to the relevant miRNAs under evaluation, whereas row *Chance* include the values of the target genes associated to the random control set of miRNAs.

randomly chosen miRNAs. In the case of the lymph node model, 43% were (de)activated (up- or down-regulation), while, in the case of the spinal cord model, 53% of them were also activated. These activations contrast with just 10% and 23% respectively in the case of the random selected genes. Of special interest is the fact that all these activities were monitored in the relapse stage of the disease, which reinforced the hypothesis of relevance for our set of miRNA.

We had hypothesized that if a given miRNA was over-expressed in a particular group of samples, the targets of this miRNA should be down-regulated. Results from this last experiment with the models showed that a large number of the target genes associated to the six differentially expressed miRNA appeared significantly down-regulated more times than the ones from a random target list. Curiously, the same effect could be seen in the up-regulated genes. These could be a retroactive regulation of the mRNA that are regulated in a translational repression form.

Similarly to the enrichment analysis carried out in Section 11.3.6.2, we also research into the pathway enrichment that the target genes could show for each relevant miRNA. To do so, we picked the target genes associated to each of our six relevant miRNA and we conducted a pathway analysis through the Panther (Mi *et al.*, 2007) database. Again, as a control parameter, the target genes associated to the random selected miRNA were also included in the analysis.

Reported results from *Panther* found only the gene set associated to miR_96 enriched from the total of 17 gene sets (11 from random and 6 from relevant). Table 11.19 includes the eight pathways found to be enriched. Column *miR_96* in the table includes the number of genes involved in each pathway in contrast with column *NCBI* which includes the total number of known genes included in the pathway.

Within the pathway list of Table 11.19, we found a classic immunologic associated pathway, the *interleukin signaling pathway*. Two other pathways, the *metabotropic glutamate receptor group I* and the *muscarinic acetylcholine receptor 1 and 3 signaling*, both related with glutamate, are also present. Glutamate has been widely related with pathological mechanisms of the multiple sclerosis, e.g. exocitotoxicity (Vallejo-Illarramendi *et al.*, 2006; Matute,

Pathway	NCBI	miR_96	Expected	Ratio	p-value
Muscarinic acetylcholine receptor 1 and 3 signaling pathway	62	5	0.14	35.7	$5.39e^{-5}$
Alpha adrenergic receptor signaling pathway	29	3	0.06	50.0	$6.88e^{-3}$
Endothelin signaling pathway	98	4	0.22	18.2	$1.23e^{-2}$
Interleukin signaling pathway	194	5	0.43	11.6	$1.29e^{-2}$
Wnt signaling pathway	348	6	0.78	7.7	$2.18e^{-2}$
Histamine H1 receptor mediated signaling pathway	43	3	0.1	30.0	$2.19e^{-2}$
Metabotropic glutamate receptor group I pathway	44	3	0.1	30.0	$2.35e^{-2}$
Angiotensin II-stimulated signaling through G proteins and beta-arrestin	53	3	0.12	25.0	$4.04e^{-2}$

Table 11.19. Pathway enrichment analysis of the target genes associated to miR_96. Column *Expected* includes the number of genes that are expected to belong to each pathway in proportion to the 57 original target genes of miRNA miR_96. Column *miR_96* how many of the target genes of miR_96 belong to each pathway. In the *Ratio* column, the proportion between the expected and the found genes is included. Lastly, the p-value of each enrichment is included.

2007). These mechanisms are related with the central nervous system but the associated genes could be expressed in blood by the activated T-cells.

The *wnt signaling pathway* is also present in the found pathways. The gene WNT has been proposed as an important player in the development of effector T-cells and in the activation of regulatory T-cell (Staal *et al.*, 2008). All these pathways may be potential subjects for more in-depth studies but, at this point, their immunological role makes our data more reliable.

In conclusion, all these results strongly suggest that miR_96 could be playing a key role in multiple sclerosis and constitutes an important candidate for further studies. The topology structure of the network dependences of Figure 11.13 corroborates the key role of this miRNA, especially in the remitting stage. miR_96 seems to be characteristic of the remitting phase of the disease: it is more expressed in remitting samples than in controls, and less in relapse samples than in remitting.

11.5.7 Conclusions

A relationship between miRNA expression and MS is expected since some of the functions attributed to the miRNA include stress response, immunomodulation (de Yébenes *et al.*, 2008; Baltimore *et al.*, 2008) and neuroprotection (Nelson *et al.*, 2008). Computational predictions propose that 30% of the human genes are regulated by micro RNAs (Ross *et al.*, 2007). We therefore hypothesize that a sizeable proportion of the mRNA differentially expressed

between samples from patients during a relapse and during remission ought to be regulated by micro RNA.

Throughout this section, a full study on the association between miRNA and MS has been presented. The methodological tool used to link both fields was the reliable dependences interaction networks introduced in Section 9.2.1. From the produced networks, six relevant micro RNAs were selected for further bioinformatic research and biological validations. The validation results were not as good as expected but this could happen in such a new domain where there is no proven evidence.

The work is still in progress and more validation and experiments are needed. However, one of the detected miRNA by the network structures (miR_96) is outstanding as a potential biomarker in multiple sclerosis. And, at least two more (miR_18b and miR_599) show potential to be good targets for future biomarker studies to characterize the relapse status.

12

Mass spectrometry

Progress is being continually made in the quest for stable biomarkers in complex diseases. Mass spectrometers are one of the devices for tackling this problem. The data profiles they produce are noisy and unstable and, within these profiles, biomarkers are detected as signal regions or peaks, where control and disease samples behave differently (Armañanzas *et al.*, 2009b).

Mass spectrometry (MS) data generally contains a limited number of samples described by a high number of features. The same problem as in the Genomics chapter, the curse of dimensionality is again presented in this biological data. Throughout the present chapter, we first present a full preprocessing pipeline to parse the raw data into a classical machine learning dataset.

As the methodological approach to deal with this problem, we collect here the results obtained by the population consensus in EDAs presented in Chapter 10 of this dissertation. Moreover, an entire data workflow is designed to yield unbiased results, and interesting findings are discussed about the need to estimate classification accuracies fairly. As spin-off results of this schema, we also discuss the consistency and stability of our results and how the classification estimation accuracies in a feature subset selection problem may overfit the training and test sets in use.

Four publicly available MS datasets (two MALDI-TOF and another two SELDI-TOF) are analyzed. The results are compared with the original works, and a new plot (PF plot) for graphically inspecting the relevant peaks is introduced.

12.1 Mass spectrometry data basics

A mass spectrometer is a general-purpose device dedicated mostly to the elucidation of the elemental composition of a sample or molecule. It is composed of three main parts: an *ion source*, a *mass analyser* and a *detector*. Its working principle is simple: to ionize chemical compounds, generating

charged molecules or fragments, and then, measure the ratios between each molecule/fragment mass and its electrical charge.

Once a sample is introduced into the device, the ion source digests the molecule(s) and splits them into ions. Then the mass analyzer sorts the ions by their masses through electromagnetic fields. Lastly, the detector will measure the abundance of each ion present in the sample.

The mass spectrometry has several uses and it is nowadays of common use in analytical laboratories that study physical, chemical, or biological properties of a great variety of compounds. Some aplications of this high-throughput device are to identify unknown compounds, determining isotopic compositions of a molecule, identification of the structure of a compound by studying its fragmentation composition, quantifying the amount of a compound in a sample, or even study the chemical properties of ions and neutrals in vacuum. Figure 12.1 includes an example of a typical MS spectra and a zoom to a portion of it. A didactical review on the MS principles and aplications can be consulted in (Gross, 2006).

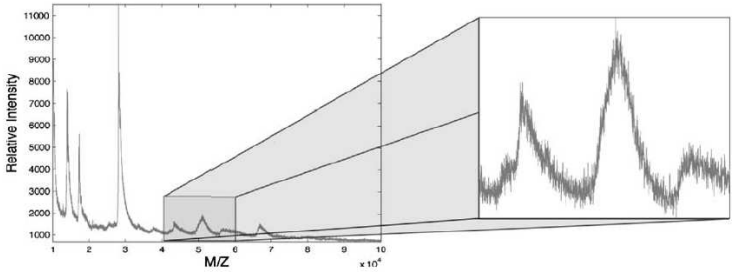


Fig. 12.1. Example and zoom of a mass spectrum. The zoom illustrates the high amount of noise that the signal includes.

In the bioinformatics discipline, mass spectrometry is an important method for the characterization of proteins. In this case, the sample is formed by a single protein compound. The protein is digested into its peptides and the peptides are run through the spectrometer. Each peptide produces what is called a peptide mass fingerprint (PMF) that individually identifies the peptide. By inspecting the PMF databases, the sequencing of a protein is retrieved from the peptides found in its digestion by following a bottom-down scheme.

An emerging application of MS in bioinformatics and biomedicine is the analysis of metabolites and proteins coming from complex samples. In essence, the purpose is to find biomarkers whose detection could be done with a mass spectrometer device. The samples are usually blood or plasma so this method promises a new and non-invasive facility to help physicians in their daily work (Inza *et al.*, 2009). Still in consolidation, the main application in this way is to study cohorts of samples from different diseases. The comparisons between the findings from control or healthy and disease samples can shed light

on new key metabolites. Throughout this chapter, we explore the population consensus on top of the general EDA scheme to deal with the discovery of biomarkers in MS data. Due to the small sample size of these MS datasets, the consensus approach is expected to enhance the robustness of the results.

Originally developed by (Karas *et al.*, 1987), matrix-assisted laser desorption/ionization (MALDI) technology can simultaneously measure peptide abundances in a given sample (serum samples in this case) by enzymatically digesting the sample and running it through a mass spectrometer device. To the same end, (Hutchens and Yip, 1993) introduced a variation in the way the sample is attached to the chemical matrix and named it surface-enhanced laser desorption/ionization (SELDI).

Both techniques are usually coupled by a time-of-flight (TOF) detector. This detector measures not only the peptide abundance, but also the time each peptide takes to reach the spectrometer's detector. Samples analyzed by this platform produce what is generally known as SELDI-TOF or MALDI-TOF data spectra. These spectra sort the abundances based on the ratio of each peptide's mass to its charge, known as the mass-to-charge (m/z) ratio.

(Petricoin *et al.*, 2002) were the first to use this technology to identify proteomic biomarkers in complex diseases. Since then, many authors have followed their example (Hilario *et al.*, 2006; Shin and Markey, 2006), reporting promising results for inducing classification systems and even more interesting findings for further research on the biology of such diseases (Ressom *et al.*, 2008). However, the analysis of this kind of data is still far from being standardized, and the scientific community is developing robust and novel methodologies.

The physics of spectrometer devices biases their outcome, adding chemical noise, signal shifts and artifacts that the subsequent analysis must deal with. As an initial contribution, we present a full preprocessing pipeline to remedy all these unwanted side-effects (Coombes *et al.*, 2007). The preprocessing ends with a peak profiling algorithm that identifies possible relevant points in each spectrum. These points, commonly known as *peaks* or *peakbins*, are the features whose values are used as the input of the feature subset selection procedure.

12.2 From raw data to machine learning features

The preprocessing stage is an elementary and critical part of the design analysis protocol (DAP (Barla *et al.*, 2008)). The DAP stage converts the data from its raw, initial form into a compact and homogeneous matrix forming the input for subsequent methods, such as machine learning or pattern recognition techniques. Thus, the main objective of the preprocessing task is to clean the data and detect the true signals in the noisy spectra.

MS data pose similar problems to most classical signal processing problems. Additionally, since the sample composition is often unknown or overly

complex, the original signal decomposition is unknown. There have been attempts to mathematically model the *true* signal in a MS experiment but with limited or no success. Although far from being perfect, the most accepted formulation is shown in Equation 12.1:

$$f(t) = B(t) + N \cdot S(t) + \varepsilon(t) \; . \tag{12.1}$$

The first term $f(t)$ is the observed signal. $B(t)$ is a visually identifiable additive baseline component, and $S(t)$ is the expected true signal, which is modified by a normalization factor N. The last element, $\varepsilon(t)$, is an unknown noise component that groups the remaining variations.

There is no standard preprocessing pipeline for MS data. Although a core set of preprocessing tasks have already been identified and accepted as a quasi-standard, pipelines do not all perform the same steps or tackle them necessarily in the same order. The most accepted dataflow core stages are: baseline removal or correction, inter-spectra normalization, signal noise reduction or smoothing, peak detection and, finally peak alignment. Other additional tasks could be outlier detection (Sauve and Speed, 2004; Ressom *et al.*, 2007) and raw signal binning (Ressom *et al.*, 2007, 2005).

In the following sections, we present and discuss our proposal of a standard preprocessing pipeline. Notice that, after the preprocessing, our main aim is to obtain a set of relevant peaks by means of a feature subset selection approach. This must be taken into account when designing each preprocessing task in an attempt to obtain as unbiased a dataset as possible. However, the reader can find other preprocessing compendia in the *state-of-the-art* literature (Sturm *et al.*, 2008).

We provide the community with a set of Matlab scripts containing an implementation of the proposed techniques[1].

12.2.1 Baseline removal

At the low range of the spectrum, the intensity values are always found to be amplified. This side effect is the consequence of chemical noise from the matrix compounds required to fix the biological sample. The amplification effect tends to lessen until the m/z values increase (Shin and Markey, 2006).

To minimize this effect, the true signal must be estimated, and the difference between the observed and estimated signal should be removed. This can be viewed as a filtering step, in the sense that each spectrum is evaluated individually and transformed into a (partially) corrected spectrum. To our knowledge, there is no agreement within the scientific community on which is the best way to tackle this problem. The most popular techniques include: smoothing by local linear regression (*loess*) (Barla *et al.*, 2008; Tibshirani *et al.*, 2004), multiple shifted windows with spline approximations (Ressom

[1] See supplementary content page at `http://www.sc.ehu.es/ccwbayes/members/ruben/ms`.

et al., 2007, 2005) and a non-linear filter approach from the field of morphological mathematics: the top-hat operator (Sauve and Speed, 2004; Breen *et al.*, 2000), and its variations (Prados *et al.*, 2006; Noy and Fasulo, 2007).

They all have the same aim, i.e. to flatten the signal by removing the estimated chemical noise. No significant difference has been reported in the literature, and there has been no systematic comparison of the different techniques. Therefore, we propose the use of the top-hat morphological operator (Soille, 1999) since it is the least time consuming and has proven its merits in the image analysis domain, where it is a filter in widespread use (Zeng *et al.*, 2006; Liu and Motoda, 2008).

The top-hat filter is a nonlinear positive low pass operator. Also known as the *white tophat*, it removes the result of performing a morphological opening operation using a predefined structuring element from the input signal. For application to MS data, each spectrum is configured as a binary array of values, and the neighbourhood element (or mask) should also be a 1-dimensional array.

12.2.2 Spectra normalization

MS spectra of similar samples are not always quantified within the same amplitude range. A normalization step is needed to compare the real intensities fairly. Using normalization, we convert all the spectra to the same intensity ranges. Many different approaches have been proposed and used to tackle this issue. Of these different approaches, one is emerging as a *gold standard*: the total ion current (TIC). TIC really encodes the average area under the curve (Alfassi, 2004).

However, a recent study of eight normalization procedures (Meuleman *et al.*, 2008) states that it is better to use a local normalization method than a global one (such as the TIC). In other words, the rescaling parameter of the spectra should be estimated by windowing the m/z axis rather than a global computation over all intensities. In addition, median values have proven to be more robust than averages as scale factors against possible outlying peaks (de Noo *et al.*, 2005), and their use is also very well-established in other normalization processes, such as, for example, DNA microarrays (Wit and McClure, 2004).

Accepting both observations, we propose combining both approaches in our normalization technique: use local estimators over m/z windows with rescaling to the median value of the TIC. The window width is a free parameter tuned by an expert. As a default value, for MALDI/SELDI spectra we suggest using 200 m/z units.

12.2.3 Signal smoothing

Once all the spectra have had their baseline corrected and been translated into the same range of intensities, the next processing step is to smooth the

signal wave from the input signal. As mentioned before, the original signal is always perturbed by white noise $\varepsilon(t)$ supposedly coming from the detection instruments. Nevertheless, this preprocessing step has not always been tackled independently, as some authors prefer to combine the smoothing step with some of the other preprocessing stages, e.g. with normalization (Prados *et al.*, 2006).

The main idea of signal smoothing is to avoid the low signal fluctuations. A great many false-positive peaks are likely to be found if the signal has not been previously smoothed and all the low resolution peaks removed. Therefore, this step should be taken before any peak detection algorithm is used. Even after applying noise reduction, we cannot rule out some of the detected peaks being due to noise perturbations, although this is minimized.

The most common signal smoothing technique is wavelet denoising proposed by (Coombes *et al.*, 2005, 2007). It makes use of the undecimated wavelet transformation to estimate the wavelet coefficients. These are then used to denoise the signal and obtain a smoothed signal. There exists an online library, namely the *Cromwell package*[2], that includes all the denoising and smoothing functions. Figure 12.2 presents a small portion of the m/z axis of one illustrative spectrum.

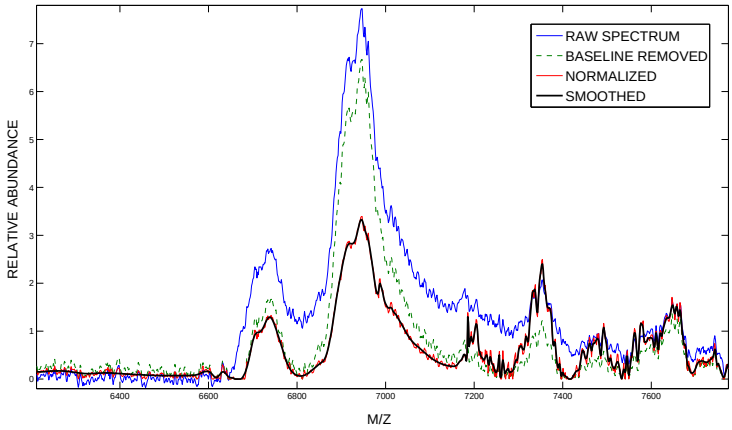


Fig. 12.2. Graphical example of the first preprocessing tasks in our proposed pipeline: baseline removal, normalization and wave smoothing. The thin solid line plots the observed relative intensity in a segment of the m/z axis. The other lines show how the original data is corrected in each of the three preprocessing stages.

[2] `http://bioinformatics.mdanderson.org/cromwell.html`

12.2.4 Peak detection

The problem of peak detection we refer to here consists of distinguishing an m/z position corresponding to a true peak in the spectrum. Many biological works define a *true peak* as the peak really associated with a peptide in the biological sample (Cruz-Marcelo *et al.*, 2008). This could be useful when the composition of the sample peptides is known beforehand and the spectra size is small. However, in the case of more complicated mixtures (e.g. blood serum), the real peaks are generally unknown.

Therefore, our aim at this stage is to prescreen a great many peaks that could later be grouped into peakbins. Bearing in mind that the machine learning analysis to be applied afterwards will separate relevant from non-relevant peaks, the impact of including some artifacts at this early stage is not so crucial. The peak detection algorithm is thus individually applied to each separate spectrum, and then, a list of candidate peaks is retrieved for each spectrum.

Even though it is by far the hottest issue in the MS preprocessing field, there is agreement only on three conditions that a candidate peak must meet (Barla *et al.*, 2008):

1. the peak must have higher intensity than its neighbours;
2. the peak must be above a chosen threshold;
3. the peak must have an associated signal-to-noise ratio (SNR) higher than a set threshold.

We will take the peak detection algorithm proposed in (Prados *et al.*, 2006) as the starting point. Our algorithm will follow the same top-down scheme, starting with the highest point of the overall signal and iteratively evaluating the lower points. To see whether a point p is considered a peak, we set a stricter criterion: there must exist a point l (respectively, r) on its left (respectively, right) before the previous (next) peak. This point must satisfy two conditions. First, the value of the candidate point p must be higher than a sensitivity threshold T and, second, the candidate point p must have an SNR higher than or equal to 3 within the intensity window framed by l and r.

To estimate the SNR of a signal window, our algorithm computes the SNR value as the ratio between the point's height and the median absolute deviation (MAD) in the window $[l, r]$ under consideration (Shin *et al.*, 2008). The criterion that the SNR must be higher than or equal to a value of three is borrowed from the image analysis field and has been previously mentioned in the microarray quality metrics (see Section 11.2.2).

The main advantage of this peak detection algorithm is that it takes into account all the individual characteristics rather than the evaluation of an average spectrum that could hide independent features (Coombes *et al.*, 2007). In addition, the spectra maintain their original m/z shape obviating the need for a shifting or alignment process. On the downside, the computation time increases linearly with the number of spectra since all spectra are investigated.

12.2.5 Peakbin assembly and quantification

There is no definite order in which this and the former (peak detection) tasks should be performed: peak alignment followed by peak detection (Coombes *et al.*, 2007) or vice versa (Slotta *et al.*, 2003). The peak or spectra alignment tries to match similar peaks detected across all the spectra. Again due to measurement-induced noise, the exact m/z value of a peak can differ from one spectrum to another and there may be slight deviations or shifts over different runs, even if analyzing the same sample. This shift is widely known as the *mass error* effect (Shin and Markey, 2006). In order to align the peaks, each spectrum is modified by shifting the signal until the peaks match.

All these variations may include signal shifts and potentially hide isotopic formations or very close compounds. Moreover, this effect is more likely when dealing with very complex mixtures.

Step 1. For each peak/peakbin p_i and for each spectrum s_j, compute the intensity value $v_{ij} = f(p_i, s_j)$.

Step 2. Compute the linear correlation matrix $\mathbf{R}$ between each pair of subset values $v_{i\cdot} = f(p_i, \cdot)$ and $v_{i+1\cdot} = f(p_{i+1}, \cdot)$.

Step 3. If all values $\mathbf{R}(i, i+1) < \rho$, then return $\mathbf{P} = [p_i]$ and $\mathbf{V} = [v_{ij}]$.
Else, for each pair p_i and p_{i+1} for which $\mathbf{R}(i, i+1) \geqslant \rho$, combine p_i and p_{i+1} into a single peakbin. Go to *Step 1.*

Fig. 12.3. Peakbin assembling algorithm pseudocode. Threshold ρ is the minimum permitted correlation threshold among two consecutive peaks or peakbins. Matrices $\mathbf{P}$ and $\mathbf{V}$ are the computed list of peakbins and the spectra values for those bins respectively.

To overcome this artificial shifting, we propose to assemble peakbins of different widths. In this way, a set of close peaks on the m/z axis across different spectra would be clustered into the same *peakbin* if their intensity levels are similar. Classical clustering approaches have already been used to tackle this problem(Barla *et al.*, 2008; Tibshirani *et al.*, 2004; Meuleman *et al.*, 2008; Slotta *et al.*, 2003). Instead, our preprocessing pipeline uses the Pearson linear correlation coefficient to group the peaks, as the computation time and memory demands are much lower. Peakbins are scanned recursively, and their signal values are quantified as the maximum value found in the bin (Prados *et al.*, 2006). The stopping criterion is met when there is no single peak or peakbin that shows a correlation value greater than a given threshold ρ. Figure 12.3 details the assembling algorithm. The output of this final preprocessing stage is thus composed of a list of peakbins, each one with a starting and ending point on the m/z axis, coupled with the maximum signal value within each spectrum.

12.3 Data analysis workflow for predictive proteomic profiling

A data analysis workflow (DAW) refers to the whole pipeline of tasks that a database under research follows (Barla *et al.*, 2008). This workflow is sometimes designed carelessly and without much concern about the side effects it could have on the final results (Baggerly *et al.*, 2004). Critical DAW aspects that could potentially bias the results have been identified in the machine learning field. The most important include:

- Performance of any preprocessing task on the whole dataset instead of first splitting the training from the test sets. In fact, if a workflow were to imitate a real scenario, the new cases would arrive at the end of the analysis (Saeys *et al.*, 2007).
- Setting the learning parameter values. This is especially tricky in wrapper schemes where the estimations are carried out on the same data that are afterwards used to train the model (Statnikov *et al.*, 2005). In a wrapper approach to feature selection, the feature selector accuracy must be estimated with a set of previously unseen instances (Reunanen, 2003).
- Previously setting a number of features to be kept could lead to overfitting. If we set the number of features to be retained, the feature selection algorithm is forced to look not only for the relevant features, but also for the features that achieve the highest accuracies when classifying phenotypes. The consequence is that the classification model is accurate in datasets with not many instances, but generalizability will be lacking when a new set of instances is provided (Liu and Motoda, 2008).
- Procedures that include stochastic elements in their formulations should be run on different multistarts. Since stochasticity drifts apart from deterministic behaviours, a single run of such techniques does not guarantee the reliability of the outputs. This effect is usually coupled with the internal variance shown by different shufflings of the instances in a k-fold cross validation estimation (Efron, 1983).

Bearing in mind all the mentioned drawbacks, Figure 12.4 introduces the data analysis workflow for the whole MS profiling experiment. It is designed to overcome the above issues and can be divided into three main parts. Before doing anything, the MS database should be baseline corrected. Since this task is independent for each spectrum (see Section 12.2.1), it can be considered as a separate task.

The first main part, namely the *outer iteration*, in Figure 12.4 corresponds to the first k-fold split. To proceed with a fair estimation in the subsequent validations, the training and test sets should be completely separated from the very beginning (Saeys *et al.*, 2007). Therefore, the *outer iteration* splits $k-1$ folds as the training set and keeps the remaining fold as the *outer test set*. This is the set for which the accuracy estimations are computed later on. After this division, the remaining preprocessing tasks are applied separately

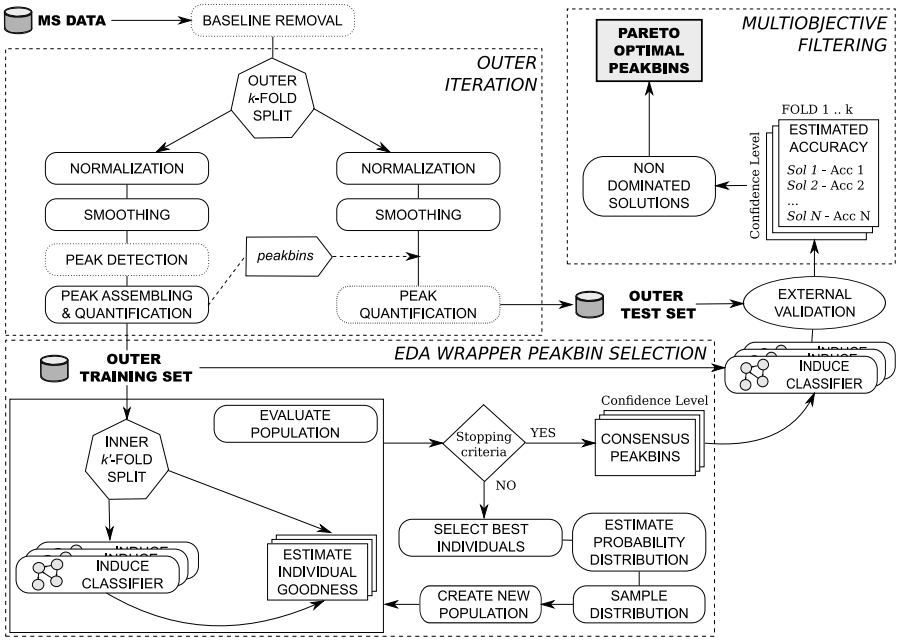


Fig. 12.4. Data analisys workflow. Tasks performed individually are included in boxes with dotted lines.

to the outer training and test set. Since the main aim of this first part is to reproduce a real validation with new unseen instances, peakbins are detected and assembled only in the training set (left workflow branch), and the resulting peakbins are then quantified in the test set (right side). This external loop is repeated k times, and different outer training and test sets are computed for each one.

The second major part comprises wrapper peakbin selection using the proposed UMDA consensus approach presented in Chapter 10. We refer to this part as the internal loop or *inner k'-fold split* or *validation*. The wrapper peakbin selector uses the classification accuracy estimation as the evaluation function to measure the merit of each individual (subset of peakbins) over the search. Note that in this internal evaluation only the outer training set is available and the outer test set remains unseen. In addition to the subset of selected peakbins, the algorithm also outputs the estimated accuracy but only on the training set. This accuracy is referred to as the internal or inner accuracy (estimated only on the inner validation). After the inner search, a classifier is induced taking into account only the values of the outcome peakbins and the outer training test. This classifier is then fairly evaluated with the outer test to output what we call the outer or external accuracy.

To illustrate this point, Section 12.5.2 shows a comparison between the inner and outer accuracies. Large differences are observed between the fair and the inner validations, sometimes as much as 5% in accuracy estimation.

The values of k (outer) and k' (inner) could be different, but we recommend a low value for k' because one inner cross validation procedure is performed for each individual and each evaluated population. In the internal search, once the stopping criterion is met, we can keep just the solution produced by the classical UMDA approach, or we can apply the consensus peakbin approach. A confidence range $T_1 < T_2 < \cdots < T_t$ is then set, and a group of different solutions $\Phi_{T_1,T_t}(S)$ is collected from S: $\Phi_{T_1,T_t}(S) = \{\mathbf{x}^C_{T_1}(S), \mathbf{x}^C_{T_2}(S), \ldots, \mathbf{x}^C_{T_t}(S)\}$. This set of solutions is thus formed by different consensus solutions at different confidence thresholds (see Section 10.2). As mentioned before, each consensus solution is evaluated afterwards using the outer test set.

The input for the last part of the data workflow after all the external folds haven been completed is all the accuracy estimations and the set of consensus solutions for each fold k, $\Phi_{T_1,T_t}(S_k)$. All this collected information can then be sorted by the confidence threshold values. To this end, for each confidence threshold $T_i \in \{T_1, \ldots, T_t\}$, we have k accuracy estimations achieved by k consensus solutions $\mathbf{x}^C_{T_i}(S_l)$ with $l = 1, \ldots, k$ and $i = 1, \ldots, t$. Note that the number of peakbins included in each solution is variable due to the intrinsic stochasticity of the UMDA approach. Thus, for a given confidence degree, there can exist two solutions with the same mean accuracy over the k folds but with a different number of peakbins.

The results of this consensus approach suggest using a multiobjective filter rather than forcing the selection of a single threshold or solution. It is very worthwhile studying how each confidence level solution behaves. To this end, there are four different objectives: the mean accuracy, its associated standard deviation, the average number of peakbins and also its standard deviation. Since there could be many solutions, we should keep only the really profitable ones. The next section presents the multiobjective dominance criterion used as the filter.

12.3.1 Multiobjective sifter

As mentioned earlier, the proposed DAW gives the expert the chance to study a full range of solutions instead of just one. Moreover, these solutions are the result of two conflicting criteria: the accuracy estimation and the size of the peakbin set (feature set). Previous studies on feature selection explored how the accuracy of the classification models evolves when the number of features increases or decreases. In general, these tendencies are dependent on the problem, however, it is generally accepted that the accuracy increases with the addition of features from an empty set until a size is reached where the accuracy no longer improves or even decreases.
Therefore, instead of using a single criterion to assess the goodness of a solution, we propose four different ones:

1. how large is the mean estimated accuracy;
2. how small is the average peakbin set size;
3. how low is the standard deviation associated with the estimated accuracy;
4. how low is the standard deviation associated with the peakbin set size.

Of two solutions with the same average size and mean accuracy, the one with the lowest variance for one or both of the objectives should be kept. All the above perfectly fit the concept of dominance (Handi *et al.*, 2007). Formally, a solution $\mathbf{u}$ can be expressed in terms of all the o objectives to be evaluated, $\mathbf{u} = (u_1, \ldots, u_o)$, where each u_i is the evaluation of the elements that form $\mathbf{u}$ in the i-th objective. Within minimization, the dominance criterion states that a solution $\mathbf{u} = (u_1, \ldots, u_o)$ dominates another solution $\mathbf{v} = (v_1, \ldots, v_o)$, $\mathbf{u} \prec \mathbf{v}$, if

$$\mathbf{u} \prec \mathbf{v} \iff \forall i \in \{1, \ldots, o\}\,,\ u_i \leq v_i \ \text{ and } \ \exists j \in \{1, \ldots, o\} \mid u_j < v_j\ .$$

The set of non-dominated solutions is known in operational research as the Pareto frontier or Pareto set (Pareto, 1896; Handi *et al.*, 2007). The Pareto frontier will only include the set of solutions $\mathbf{v}$ that are not dominated by any other solution $\mathbf{u}$. This Pareto set thus comprises all the solutions that cannot be improved for any objective without simultaneously degrading some other objective value.

Table 12.1 contains an example with six different solutions from a solution set S. Each solution is retrieved at a different confidence level T_i, and the four objectives are included. The first four solutions are non-dominated, and they will be output as valid consensus approach solutions, while the last two will be removed because $\mathbf{x}^C_{T_5}(S)$ is dominated by $\mathbf{x}^C_{T_3}(S)$ and $\mathbf{x}^C_{T_6}(S)$ by the above four.

	Accuracy	**Std**	**Peakbins**	**Std**	**Pareto front**
$\mathbf{x}^C_{T_1}(S)$	94.5	2.3	15	4	✓
$\mathbf{x}^C_{T_2}(S)$	80.6	10.1	5	3	✓
$\mathbf{x}^C_{T_3}(S)$	96.0	1.8	30	10	✓
$\mathbf{x}^C_{T_4}(S)$	80.1	9.0	6	2	✓
$\mathbf{x}^C_{T_5}(S)$	96.0	1.8	31	10	×
$\mathbf{x}^C_{T_6}(S)$	80.0	11.0	40	15	×

Table 12.1. Example of the multiobjective sifter for a set of six solutions at different confidence thresholds $T_1, \ldots, T_6$. The last column indicates which solutions are not dominated by any other and belong to the Pareto front.

12.4 Mass spectrometry datasets

Four different datasets have been used to illustrate the presented peakbin selection processing. Two are from a SELDI, whereas the other two are from

a MALDI spectrometer. The number of samples, phenotypes and available m/z readings varies noticeably. All these datasets are available at the sites of their respective authors (Petricoin *et al.*, 2002, 2004; Ressom *et al.*, 2006, 2008). Unfortunately, there is currently no uniform storage protocol for this kind of data. Thus, we had to use a parsing algorithm to adapt the original raw data files. None of the provided plain text files for all the datasets share the same m/z axis, even within the same dataset. So, we had to set a resolution of 0.025 and average all points over their maximum and minimum values using bins of this width. When there were no values available for an interval, a null value was assigned. A detailed description of each dataset's features follows.

- Ovarian cancer profiling (OVA) (Petricoin *et al.*, 2002). Being one of the pioneering works on MS data profiling from serum samples, the work by Petricoin *et al.* (2002) is now one of the most analysed benchmark MS datasets. The aim is to separate serum samples of a female population with ovarian cancer from control samples of unaffected women using a small set of proteomic markers. The available data contain 200 SELDI spectra of 121 cancer samples and 79 controls. The m/z values range from 700.116 to 12,000 with a total of 45,200 values per spectrum.
- Detection of drug-induced toxicity (TOX) (Petricoin *et al.*, 2004). In this work, rat models are analysed using a serum proteomic pattern diagnostic device based on a SELDI-TOF spectrometer. The study intends to find biomarkers able to distinguish between anthracycline- and anthracenedione-induced cardiotoxicity and control samples. The separation of the training and test sets in the original work is confusing. Consequently, we just picked the samples diagnosed as *definite positive* or *definite negative*. Our TOX dataset then is composed of 62 samples of two phenotypes with 28 and 34 samples each. As in the previous dataset, a total of 45,200 m/z values are configured, ranging from 799.115 to 12,000.
- Hepatocellular carcinoma (HCC) (Ressom *et al.*, 2006). This study sets out to help discover early markers for hepatocellular carcinomas triggered by viral infections. The samples were obtained from the Kasr El-Aini Hospital (Cairo, Egypt), where this carcinoma is a primary health problem. After removing proteins greater than 50 kDA (including albumin), the spectra are generated by a MALDI-TOF instrument. The dataset includes 36,802 m/z final readings for 150 samples, 78 affected and 72 non-affected controls.
- Detection of glycan biomarkers (DGB) (Ressom *et al.*, 2008). Ressom *et al.* (2008) propose a method for systematically selecting glycan structures able to distinguish subjects from pre-labeled groups. Over a set of three different phenotypes, the glycans are released from their associated proteins through an enzymatic treatment and later methylated to avoid solubility. The available data comprises a total of 128 MALDI-TOF spectra: 78 from healthy controls, 25 from hepatocellular carcinoma and another 25 from

chronic liver disease samples. The m/z values are in the 1,499.8 to 5,518.3 interval, with a total of 16,075 points for each sample.

12.5 Results and discussion

Stochastic approaches take advantage of their random search policies to inspect the search space. However, this is a drawback rather than an advantage when a precise result is required: the need to repeat the search approach several times (Liu and Motoda, 2008). Usually known as multistart or re-run, the aim of a systematically run repetition is to find a stable outcome.

In our case, this random component is present in several stages of the DAW, namely, in each outer and inner fold, in each initial population and in the stochastic behaviour of the UMDA itself. Thus, the multistart run is a must rather than a choice. To avoid this intrinsic variance, all the results presented throughout this section are extracted from a set of 500 multistart runs for each of the analyzed MS datasets.

12.5.1 Running parameters

For reproducibility purposes, we include here all the parameter settings used in the experiments. The full DAW can be divided into two main steps: the preprocessing stage and the UMDA peakbin consensus selection.

Table 12.2 sets out all the running parameters needed to fully configure the preprocessing stage. Due to the different data distributions, the baseline removal, normalization, peak detection SNR and correlation threshold ρ were the same for the four datasets, while the other parameters needed to be adapted individually. Notice that all the preprocessing steps are performed for each external fold of the DAW (see Section 12.3).

Top-hat structural mask	[0; *ones(298,1)*; 0]*			
Window size for normalization scanning	200			
Minimum SNR on peak detection	3			
Threshold ρ to agglomerate peakbins	0.80			
	OVA	**TOX**	**HCC**	**DGB**
Smoothing wavelet threshold	5	6	6	6
Power of 2 for wavelet denoising	3	7	10	10
Minimum intensity T on peak detection	5	20	4000	1500

Table 12.2. Running parameters configured for the preprocessing task. Note that the first four parameters are shared across all the datasets, while the last three need to be adapted individually. (*) The formulation represents a 300-position binary array containing ones except for each end position whose value is zero.

Regarding the rest of the running parameters, note that the number of times the outer loop k is performed produces a large increase in the total

computational time and, even more so when the number of inner folds k' is high. Our running scheme uses $k = 5$ folds on the outer loop and $k' = 5$ inner folds on each individual accuracy estimation. For this wrapper accuracy estimation, the classification model is a continuous naïve Bayes (John and Langley, 1995) with conditional normal density distribution for the features (this Bayesian network classifier is explained in 4.2.1).

The initial population of each UMDA selection is randomly drawn from a Bernouilli distribution with a success probability $p = 0.1$ of each peakbin being initially selected. This was found to be the best value for reaching a compromise between the number of selected peakbins and their performance in a wrapper selection. There are two stopping criteria: either to achieve a 100% accuracy estimation, or to reach a hundred generations. Each population is formed by 100 individuals and the truncation threshold is set at 50% as usual.

To output the set of consensus solutions, the confidence range is set up with confidence levels of $T_1 = 10\%$ and $T_t = 100\%$ with a 1% step. This means that, for the set of best individuals, features selected less than 10% of the times are rejected. In the outermost case, 100 different best individuals are collected (one per population), and the total number of consensus solutions on each outer fold also reaches 100. Since all these solutions are sifted by the multiobjective filter, only those belonging to the Pareto front will be retained as valid results.

12.5.2 Differences in the accuracy estimations between the outer and inner loops

As previously discussed in Section 12.3, some works have already pointed out the misleading results produced when the same sample set is used to search relevant features and the same set is used again to estimate a classification accuracy (Statnikov *et al.*, 2005; Reunanen, 2003). Table 12.3 illustrates this point numerically. For each dataset, the best solution found by the UMDA is evaluated both in the internal loop (*inner accuracy*), which guides the search, and with the external test set (*outer accuracy*), the set of new instances not seen in the search-train process. Values in Table 12.3 reflect the average estimations in both cases, plus their associated standard deviation.

	Inner accuracy	**Outer accuracy**
OVA	99.65±0.54	98.37±1.94
TOX	92.32±6.54	88.35±10.56
HCC	98.31±1.00	93.45±4.17
DGB	95.64±1.24	90.49±5.72

Table 12.3. Average accuracy estimations for the internal and the external evaluations. Estimations are computed for each fold, in both the inner and outer loops and include their associated standard deviation.

To see if all these differences are statistically significant, we use a hypothesis t-test of equal means to make comparisons between all fold accuracies (pairwise combinations of inner fold against outer estimations). The test rejects the null hypothesis of equal means in all cases where p-values are always less than 0.01. These results stress the fact that estimations made based on inner sets only are always too optimistic and, thus, not fair to the real data. A fair accuracy estimation by means of an inner-outer scheme is necessary.

These results also show that the inner estimations have a low variance, whereas the variance in the outer estimation is up to an order of magnitude greater than the inner ones. This variance is explained by the fact that the inner models overfit to that fold's training set and because their generalization power degrades when unseen instances are tested.

12.5.3 Multistart non-dominated solutions

Like most search procedures, the UMDA-wrapper peakbin selection only outputs a single solution. On many occasions, the search procedure could have explored parts of the search space with good solutions that, however, do slightly worse on the evaluation and are, thus, discarded. The retrieval of this useful information is the aim of the population consensus proposed in Section 10.2. Its first advantage is that whereas the classical scheme only retrieves one solution, the consensus approach may produce as many solutions as populations have been generated in a single run. Many of these solutions may be similar or even equal. Hence a filtering process is required to output the really interesting solutions. In our case, this sift is the non-dominance criterion with respect to the four objectives defined earlier.

The first row of Table 12.4 presents the total number of non-dominated solutions that have been reported throughout the whole set of multistart runs. This is tens of thousands for all datasets, whereas the classical UMDA approach reports only a total of 500 solutions (one per run). The second and third row show the mean number of solutions per run, as well as the mean number of peakbins in each solution, of the Pareto front.

The estimated accuracy of every one of the non-dominated solutions is also computed by the validation on the outer test sets. The difference from the estimation output by the classical UMDA approach (see *Outer acc.* column in Table 12.3) is clear. The consensus approach is able to find solutions that outperform the UMDA approach accuracy estimations for all datasets (see the outer accuracies in Table 12.3 for comparison). For the OVA dataset in particular, the estimated accuracy reaches a value of 100% for the 5-fold estimator in use. For the other three datasets, the mean estimator also reaches competitive percentages: 93.84%, 97.33% and 95.29%. Nevertheless, if we take the standard deviation into consideration, there are some folds for which accuracy is also 100%. Notice that this variance is numerically similar to the values reported in Table 12.3.

	OVA	TOX	HCC	DGB
Total number of solutions throughout 500 runs	72,744	66,277	62,597	38,735
Mean number of solutions on each Pareto front	35.15	29.25	27.16	16.26
Mean number of peakbins per Pareto solution	97.64	155.34	25.28	11.74
Maximum accuracy	100±0	93.84±6.43	97.33±2.79	95.29±5.06
Peakbins	39.80±17.06	114.60±60.08	20.60±19.03	8±4

Table 12.4. Descriptive overview of the multistart results produced by the population consensus proposal. The first row indicates the number of non-dominated solutions collected for all the runs. The mean values represent the mean number of non-dominated solutions per run and the mean number of peakbins in each solution. The last two rows show the accuracy and mean number of peakbins associated with the best solution found by the consensus.

The *peakbins* row in Table 12.4 shows the average number of peakbins included in the best consensus solution (*maximum accuracy*). An in-depth analysis of this characteristic illustrates another interesting effect: the parsimonious behaviour of our consensus approach. Figure 12.5 presents, for each multistart run and for each non-dominated solution of each of these runs, the average number of peakbins. The side color map adds a fourth component to the plot: the mean accuracy achieved by each solution. The first conclusion from the charts is that the solutions with the fewest peakbins do not achieve a good classification accuracy. However, when more peaks are added, the solutions achieve significant accuracy levels. It is when this number of newly added points increases that the parsimonious behaviour (Duda *et al.*, 2001) is observed: accuracy does not improve as a consequence. Since all the new peakbins are relevant for the problem, classification accuracy is not harmed. In terms of phenotype separability, however, the new points add no new information. The different number of non-dominated solutions in each run is clearly explained by the stochasticity discussed at the beginning of this section.

12.5.4 Peakbin stability comparison between the consensus and the classical UMDA approach

Apart from classification power, it is worthwhile analyzing how stable the non-dominated solutions are compared with the regular solutions output by the classical UMDA. A general stability index Σ was introduced in Section 10.3, and two different consistency measures (I_K and I_J) were also presented.

A consistency measure is used to quantify the (dis)similarity degree between two subsets of features in a feature subset selection problem. High levels of consistency between both subsets suggest that the feature selection approach is highly stable, a desirable behaviour in knowledge discovery tasks.

When there are more than two subsets (solutions), we can compute the stability degree Σ among all of the subsets as the average of all pairwise consistency comparisons. Notice that when there is a relatively high number of

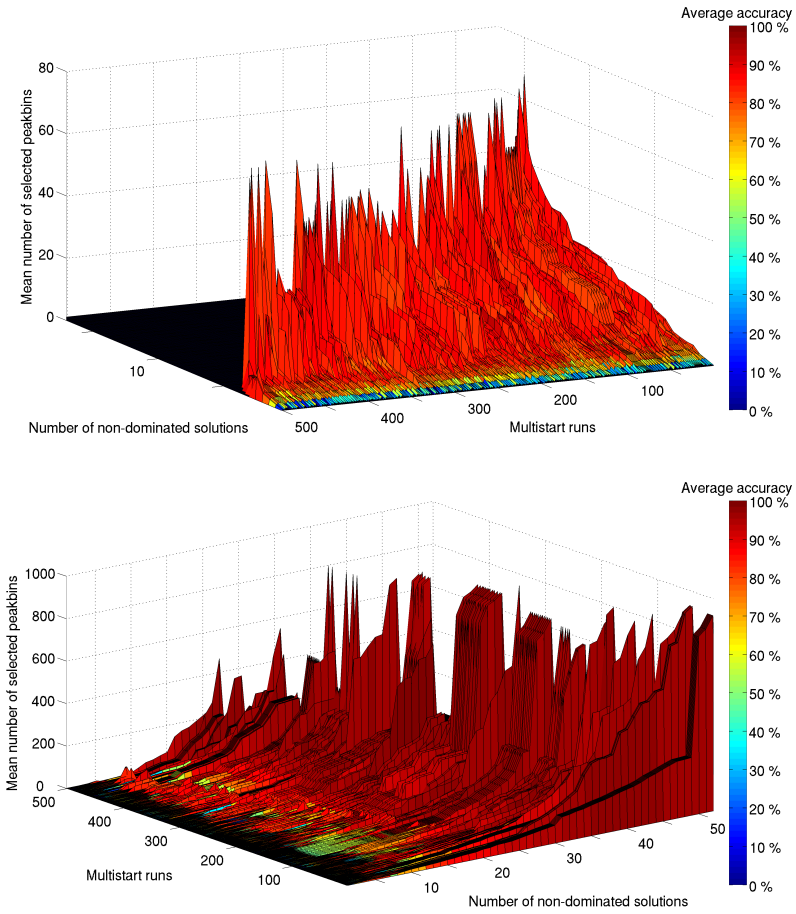


Fig. 12.5. Graphical representation of the non-dominated solutions collected in the multistart run process for two of the datasets. The top chart represents the DGB dataset results, whereas the bottom chart shows the results for the OVA dataset. In both charts, the Z axis presents the mean number of peakbins that each solution contains. The X and Y axis have been switched between the two charts for clarity. The color map of the surface represents the average accuracy estimated for each one of the points (solutions).

solutions, the combinatorial number of comparisons could lead to an unfeasible computational time.

The number of solutions that the consensus approach outputs prevents us from computing the global stability value by inspecting all possible combinations (see the total number of solutions in Table 12.4). Therefore, we propose analyzing stability by averaging the stability values of each multistart run rather than mixing the solutions from different runs.

As the multistart runs are based on five external folds, the classical UMDA solution selects five different peakbin sets for each i-th run, $\mathbf{S^i} = \{S_1^i, \ldots, S_5^i\}$. After choosing one of the two consistency measures, we can then compute the stability of this solution set in the i-th run, using the stability index, $\Sigma(\mathbf{S^i})$. Assuming B different multistart runs, the mean stability value $\Im$ is calculated straightforwardly as

$$\Im = \sum_{i=1}^{B} \frac{1}{B} \Sigma(\mathbf{S^i}) \ .$$

	OVA	**TOX**	**HCC**	**DGB**
$\Im_{I_K}$				
$\mathbf{S}$ = UMDA$_{\text{consensus}}$ (best accuracy)	0.0721	0.0087	0.2114	0.3678
$\mathbf{S}$ = UMDA$_{\text{consensus}}$ (max size)	0.2181	0.1466	0.3696	0.3040
$\mathbf{S}$ = UMDA$_{\text{classical}}$	0.1217	0.0647	0.3045	0.2721
$\Im_{I_J}$				
$\mathbf{S}$ = UMDA$_{\text{consensus}}$ (best accuracy)	0.0662	0.0251	0.1722	0.3362
$\mathbf{S}$ = UMDA$_{\text{consensus}}$ (max size)	0.1720	0.1406	0.2680	0.2243
$\mathbf{S}$ = UMDA$_{\text{classical}}$	0.0888	0.0680	0.2127	0.1945

Table 12.5. Mean stability values $\Im$ computed in terms of the I_K and I_J consistency measures. Classical rows present the values for the classical UMDA scheme. Values in consensus rows show the respective values for the most accurate set of solutions and for the solutions with the largest number of peakbins in each run, respectively.

In the case of the consensus approach, there is a variable number of non-dominated solutions per fold and run. To compare all these solutions fairly, we first need to select a representative solution from each Pareto set. We have chosen two criteria: i) the solution that achieves the highest accuracy in each fold (if there is a draw, the one with fewer peakbins is selected), and, ii) the solution that includes the maximum number of peakbins. Once the solutions are retrieved, the mean stability value is computed.

Results of all the mean stability values are set out in Table 12.5. Rows under $\Im_{I_K}$ and $\Im_{I_J}$ refer, respectively, to the average stability values using I_K and I_J consistency measures. Since all these values are based on averages, it is possible to statistically compare their differences using a t-test of equal means. All the comparisons between the classical UMDA and the consensus values

(the highest accuracy or the maximum number of peakbins) are significant at a significance level of $\alpha = 0.01$.

From the results, the first observation is the difference in the stability index between the consensus solutions in the highest accuracy and the largest number of peakbins. The low stability for most accurate consensus solutions is a consequence of the high variance in the number of peakbins in each solution. As discussed previously, the most accurate solutions only include the features necessary to tackle the classification problem (parsimonious tendency or Occam's razor). As a consequence, solutions from different folds can differ significantly.

Looking at the largest solutions (max size), however, the stability index improves significantly. Notice that the largest solutions are non-dominated, so their accuracy performance is mostly expected to be as good as the most accurate solutions (see Figure 12.5 and discussion on Section 12.5.3). The stability improvement is due to the fact that the addition of new peakbins in our consensus approach focuses on increasing the robustness of the selected set of variables.

Comparing the classical UMDA and the consensus stability results, we find that, in three out of four datasets, the classical UMDA solutions show higher consistency values compared with the more accurate but smaller and more diverse consensus solutions. However, when the classical UMDA solutions are compared with the largest non-dominated consensus solutions, they are defeated in terms of stability for all the four datasets. In the cases of OVA and TOX, the stability gauged by both Kuncheva's and Jaccard's consistencies is doubled by population consensus, and, as previously pointed out, they always show statistically significant differences.

12.5.5 Knowledge discovery using the consensus results

The data mining and machine learning disciplines provide computational biology with powerful tools to help in the analysis, diagnosis, prognosis and new knowledge discovery within data produced by high-throughput biological devices (Larrañaga *et al.*, 2006). Analysis, diagnosis and prognosis form what is known as personalized medicine. Although they are all in constant evolution, knowledge discovery is the topic that is most likely to enrich basic research and propose new hypotheses about complex biological problems or diseases.

Therefore, we consider that an optimization process, such as the search for relevant or discriminative peaks in mass spectra data, must also comply with the proposal of new biological hypotheses for validation. Throughout this section, all the consensus multistart results will be graphically presented and discussed with respect to the original author findings.

For quick reference, we have designed a combined plot. This new plot, referred to as the *peak frequential plot* or *PF plot*, is formed by two overlapped subplots. The first subplot illustrates the absolute intensity differences between the mean spectrum of the different phenotypes. The second subplot

includes the percentage of occurrence of each m/z position being selected as relevant. Applied to our approach, this percentage shows how many times each m/z position, in all the non-dominated solutions, is added as a new relevant peakbin. Figures 12.6 and 12.7 are examples of this peak frequential plot. When there are more than two phenotypes (as is the case of DGB), the top subplot is computed as the sum of all the pairwise differences between the mean spectrum of each phenotype.

The top subplot presents what could be considered as the simplest peak selector, whereas the second subplot sets out the results of the peak selection method. As we discuss later, the positions showing the largest differences are expected to be relevant for both methods. Even so, an expert may find more interesting a peakbin that is selected many times but whose average behaviours differ little.

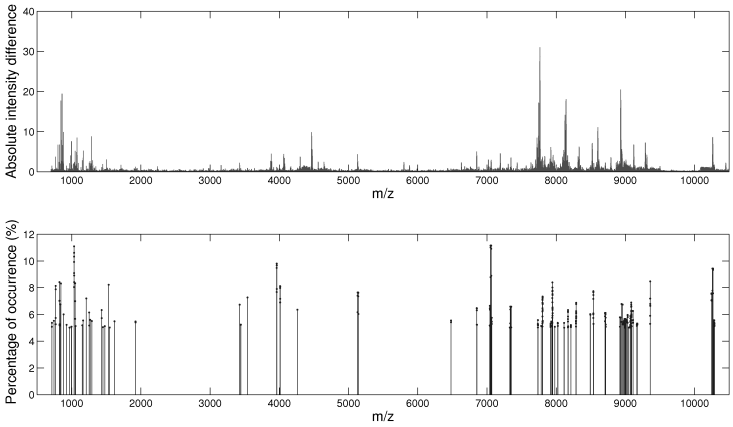


Fig. 12.6. Peak frequential plot for the OVA dataset. The top subplot shows the absolute differences among the average spectra of each phenotype. The bottom subplot sets out the results of the multistart consensus approach. It shows the percentage of occurrence of each m/z position being selected throughout the whole process (occurrences below 5% are not shown).

The original paper on the ovarian cancer profiling dataset (Petricoin *et al.*, 2002) reports a discriminative rule of five peaks that provided an almost perfect classification. Figure 12.6 presents the PF plot of our consensus approach for the OVA dataset. Already shown in Section 12.5.3, we are able to achieve the highest accuracy value in terms of spectra separability. However, our peakbin set did not compare with the set originally reported. The results reported by Petricoin *et al.* (2002) have been previously said to contain artifacts supposedly from an unfit denoising (Baggerly *et al.*, 2004, 2005).

Looking at Figure 12.6, the [7,052-7,061] peakbin has the highest occurrence level, and its width could suggest a possible isotopic configuration. Other interesting values with large occurrences are [1,034-1,036], [3,961-3,963] and [1,025.9116-1,026.7366]. Lastly, notice that for the peakbin configuration at [5,131-5,142.6], the associated difference is small, whereas the bin is often selected.

The authors also aimed for a panel of only five predictive peaks for the TOX dataset (Petricoin *et al.*, 2004). The original results are calculated based on a different sample distribution, so the outcome of comparing their panel and our results might be slightly different. Nevertheless, our results are able to identify four out of five members of the suggested peakbin panel.

Since the preprocessing proposals are not equal, the observed intervals for the m/z axis do not exactly match in both studies. For instance, the peak at 810.33765 maps in our data to the bin [810.115-810.365] with an occurrence of 4.20%. Similarly, the peak at 981.8242 matches the [981.615-981.865] bin and has an occurrence of 3.50%. The original peaks at 1,987.9727 and 2,013.5771 are also detected as relevant by the multistart process but with an insignificant occurrence level.

The PF plot for this TOX dataset is included in the online *Supplementary content*. The phenotype spectra have a high variance in this dataset. As a consequence, the estimators in the classification have a high associated standard deviation: the classifier is able to achieve up to 100% accuracy in some folds, whereas, on the same run, accuracy for other folds is only 88%. Either way, the peak frequential plot shows other m/z values that seem to be of biological interest.

Results for the HCC hepatocellular carcinoma show a significant match. (Ressom *et al.*, 2006) presented a MS biomarker panel of six peakbins in the study of hepatocellular carcinomas triggered by viral infections. Our results reported a full coincidence with this six peakbin panel. Table 12.6 presents the original m/z bins, our corresponding m/z bin and the percentage of occurrence of each bin. Not only are all six values found to be relevant by our consensus approach, but the percentage of occurrence for these six is also remarkably high. Three of them present the highest occurrence values in the multistart process with a value of around 15%.

Apart from the above six relevant bins, the PF plot (see Figure 12.7) suggests that there are other relevant peakbins that may merit an in-depth analysis. A closer look at these peakbins suggests that, comparing the absolute difference and the percentage of occurrence, three, namely [1,445.725-1,454.475] with 10.50%, [3,307.975-3,309.725] with a 12.95% and [4,206.975-4,216.225] with a 11.80% of occurrence, respectively, are noticeable. The density of bins surrounding the latter peakbin also may suggest a possible isotopic effect on that m/z position.

The last dataset was produced with the aim of selecting glycan structures able to distinguish subjects from pre-labeled groups. The authors (Ressom *et al.*, 2008) identify a panel of 10 markers with different frequencies in their

Original m/z bin	Current m/z bin	Occurrence (%)
933.6 - 938.2	933.475 - 938.225	14.77%
1,378.9 - 1,381.2	1,378.975 - 1,381.225	10.32%
1,737.1 - 1,744.6	1,737.225 - 1,744.475	15.13%
1,863.4 - 1,871.3	1,863.975 - 1,870.225	15.27%
2,528.7 - 2,535.5	2,528.725 - 2,535.475	12.07%
4,085.6 - 4,097.9	4,085.725 - 4,097.975	6.78%

Table 12.6. Original relevant peakbins reported by (Ressom *et al.*, 2006) for the HCC dataset. For each bin, the second and third column map, respectively, our correspondent m/z relevant peakbins and the occurrence percentage of each bin in the multistart.

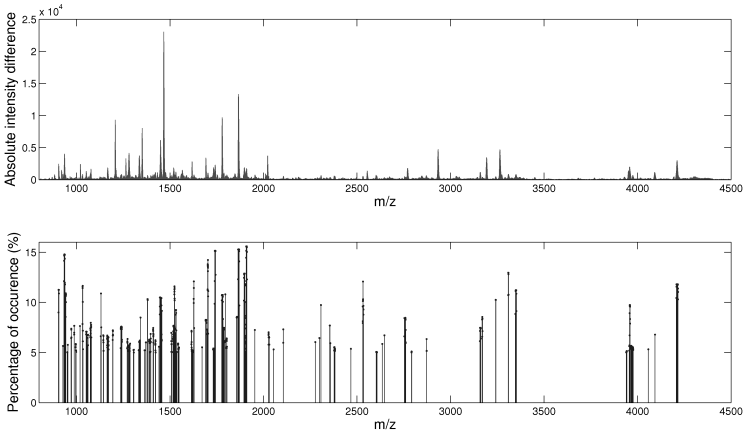


Fig. 12.7. Peak frequential plot for the HCC dataset. Top subplot shows the absolute differences among the average spectra of each phenotype. The bottom subplot sets out the results of the multistart consensus approach. It shows the percentage of occurrence of each m/z position being selected throughout the process (occurrences below 5% are not shown).

results. When compared with our results, we find that 7 out of the 10 markers are also identified by the consensus proposal. Moreover, the percentage of occurrence in our multistart is also high for the most important ones. Figure 12.8 shows the PF plot of our results. Notice that, of the seven bins in common, five are highlighted by boxes in the figure.

A careful analysis of Figure 12.8 draws attention to three more peakbins that, either because of their high occurrence, or because of a large difference in intensities, may merit further research. The bin at [2,039.7615-2,041.7615] has the highest percentage of occurrence with 25%, whereas the bin located at [2,792.0115-2,795.0115] is associated with the largest absolute difference. We would like to also point out bin [4,400.2615-4,4035.1149], which has both

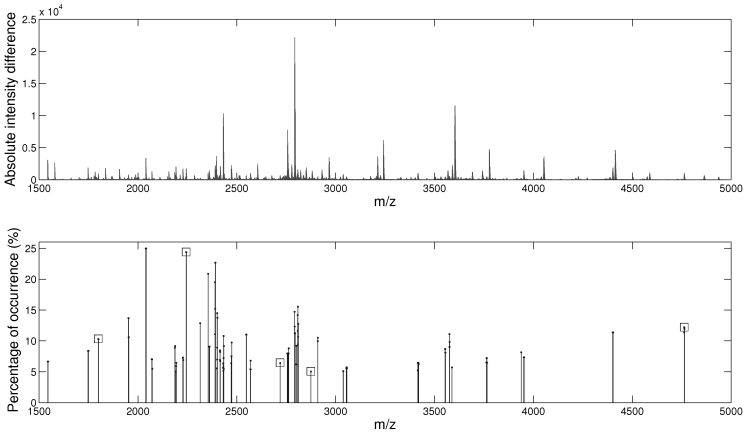


Fig. 12.8. Peak frequential plot for the DGB dataset. Top subplot shows the absolute differences among the average spectra of each phenotype. The bottom subplot sets out the results of the multistart consensus approach. It shows the percentage of occurrence of each m/z position being selected throughout the process (occurrences below 5% are not shown).

a sizeable percentage of occurrence and a visible intensity difference among phenotypes.

All the above peakbins could be of interest for further biological examination. The peak frequential or PF plots are thus a general and powerful proposal for graphically identifying relevant peaks. An expert can easily check or point out some point(s) of interest when inspecting these figures. A PF plot gives a broader view of the results, and opens up prospects for subsequent wet lab research.

12.6 Conclusions

On the basis of biomarker discovery in MS data, we find three important advantages. One is the fact that the sample in use is serum. Consequently the test for collecting the sample is almost non-invasive for the patient. Another issue is that the economic cost of a MS run is much cheaper than, for example, a classical cDNA microarray. Yet another good point is the possibility of looking for early-stage metabolic markers, an unfeasible search in the microarray field. Nevertheless, there is a big pitfall still to be overcome. This is the fact that the MS profiling results are intrinsically noisy, non-constant and difficult to analyze.

As a first step in the search for relevant peaks in MS data, the user encounters the problem of preprocessing the raw data to minimize all the noisy and

variance-related behaviours. To this end, we have presented a full pipeline of tasks, including baseline correction, spectra normalization, smoothing, peak detection and quantification. The preprocessing part of the analysis should be viewed as separate from the subsequent search for relevant peaks. Therefore, other preprocessing pipelines could be used.

Once the data is ready for a relevant peak selection task, the classical feature selectors come up against the curse of dimensionality. In this context, we propose the use of stochastic policies that are suited to dealing with the high number of features for evaluation. The low number of samples implies that the search is not always as robust as it should be. To improve the reliability of the output relevant peaks, we apply a consensus scheme over the search population. One straightforward advance in robustness is that an expert can set a confidence threshold and rely just on findings above this limit. A multi-objective filter of the solutions outputs only those sets of peaks that are better in terms of phenotype separability power, small set sizes or low variance in these two terms.

In addition, all the analysis is embedded into a workflow that imitates how all the tasks would behave when dealing with new and unseen samples. If there is no such workflow, results could be overfitted to the available data and may lose generalizability. Moreover, the results of this workflow also behave parsimoniously like supervised classification within feature subset selection procedures: small sets of features achieve good accuracy values, and these values are not improved when adding more predictive features.

The consensus approach allows us to study how stable the selection is. In actual fact, stability results quantitatively illustrate how the consensus approach is able to retrieve significantly more stable solutions than the classical UMDA approach. As expected, if the practitioner decides to rely on only the most accurate subsets of peaks (usually of small size) then the variability component is large and, thus, stability is penalized.

Nevertheless, finding locations of interesting masses should be coupled to a subsequent knowledge discovery stage in which those peaks are studied and evaluated. To this end, we have introduced a novel plot, the PF plot, to display the results of a peak selection method in supervised MS data problems. The new plot enables an expert to graphically explore the results and identify peaks of special interest. Although the presented PF plots include the results from the multistart runs of our consensus UMDA approach, they can be used by any kind of selection method.

The relevant peaks found by the consensus approach closely match the peaks reported by the original works. By inspecting the PF plots of our results, we extended the original findings with a series of peaks that could be of interest for a more in-depth biological analysis.

Part IV

Conclusions

13

Conclusions and future work

This chapter presents the general conclusions of this dissertation. More specific conclusions have been exposed in each corresponding chapter. Additionally, we include a list of publications and some considerations on future work.

Throughout the dissertation, we have presented different data mining and machine learning techniques to deal with problems within the computational biology field, especially focused on bioinformatics. In the course of a bioinformatic research, the practitioner usually gets biased results due to the unbalanced dimensionality of the datasets. In this scenario, we propose the use of different consensus approaches in order to minimize those biases and to gain reliability and robustness.

The first set of analytical tools presents several univariate metrics that are well fitted to have an initial vision of the features of a particular dataset. No assumption about the data distribution is made in the metrics computation, a fact that is ideal when dealing with a low number of cases. For taking the research to a more complex level, different discretizations are combined with a correlation feature selection with the aim of finding a nuclear set of high relevant and non-redundant features. Results from both approaches show their potentiality when applying them in gene expression problems.

Making use of the Bayesian network classifiers, a new hierarchical and edge-variable Bayesian classifier have also been presented. This paradigm allows the seeking of high reliable conditional dependences between pairs of features (and the supervised class variable). The main aim of this classifier is to assure that if a dependence is repeated in many occasions, that dependence will seldomly be a false positive dependence. Of course, this statement applies always to the available dataset but it needs to be experimentally corroborated when lent to the whole study domain. Its application is not limited to gene interaction networks and it may be used in any kind of domain where supervised classification may be applied too.

The last methodological contribution of this thesis is centered on the analysis of mass spectrometry datasets. To be precise, the analysis of spectra produced by serum samples of different phenotypes. In this field, there is still

a lot of work to do for many reasons. First, we can cite that the domain is extremely noisy; there is also a full pipeline of tasks proposed to remove all that noise. Second, the search space is extremely huge and classical exhaustive methods are not a feasible solution. The proposed population consensus in estimation of distribution algorithms aims to obtain a high relevant and consistent sets of features. One last reason is that there is no standard protocol to tackle mass spectrometry experiments. This sometimes makes the results over optimistic; an honest way to estimate the accuracy of a feature set is also gathered in our work.

All the methodological approaches presented through the dissertation are put on stage in four different bioinformatics applications. Successful results are collected in all cases: starting on new pathogenesis findings for two autoimmune diseases, the identification of previous findings on colorectal cancer, the proposal of new research targets in cancer and multiple sclerosis or the discovery of previously reported and new biomarkers. The collaboration between bioinformatics and biomedicine is proven to outperform the results that each part could obtain on its own. Nonetheless, an effort should be made by both parties to work together as their individual concepts are far apart.

13.1 List of Publications

The work presented in this dissertation has produced the following publications and submissions:

A. Technical reports

- R. Armañanzas. *Solving bioinformatics problems by means of Bayesian classifiers and feature selection.* Technical Report EHU-KZAA-IK-2/06, University of the Basque Country, 2006.
- R. Santana, C. Echegoyen, A. Mendiburu, C. Bielza, J. A. Lozano, P. Larrañaga, R. Armañanzas, S. K. Shakya. MATEDA: A suite of EDA programs in Matlab. Technical Report EHU-KZAA-IK-2/09, University of the Basque Country, 2009.

B. Book chapters

- R. Armañanzas, B. Calvo, I. Inza, P. Larrañaga, I. Bernales, A. Fullaondo, A. M. Zubiaga. Clasificadores Bayesianos con selección consensuada de genes en la predicción del lupus eritematoso sistémico. *Minería de Datos: Técnicas y Aplicaciones*, 107–135, 2005.
- I. Inza, R. Armañanzas, G. Santafé. Una aproximación al software WEKA. In *Aprendizaje Automático: Conceptos Básicos y Avanzados*, 23, 477–483, 2006.

- I. Inza, B. Calvo, R. Armañanzas, E. Bengoetxea, P. Larrañaga, J. A. Lozano. Machine learning: An indispensable tool in bioinformatics. In R. Matthiesen, editor, *Bioinformatics Methods in Clinical Research*. Humana Press, 2009.

C. Conference communications

- R. Armañanzas, B. Calvo, I. Inza, P. Larrañaga, I. Bernales, A. Fullaondo, A. M. Zubiaga. Selección de genes asociados a dos enfermedades autoinmunes a partir de microarrays de ADN. *VI Jornadas de Transferencia Tecnológica de Inteligencia Artificial, TTIA (AEPIA)*, 63–70, 2005.
- R. Armañanzas, I. Inza, P. Larrañaga. Consensus gene selection on DNA microarrays. *European Conference on Computational Biology*, 2005.
- A. García, A. Freije, R. Armañanzas, I. Inza, Z. Ispizua, P. Heredia, P. Larrañaga, G. López Vivanco, T. Suárez, M. Betanzos. Simultaneous search of genomic and proteomic biomarkers in human colorectal cancer. *Genomes to Systems Conference*, 2006.
- A. García, A. Freije, R. Armañanzas, I. Inza, Z. Ispizua, P. Heredia, P. Larrañaga, G. López Vivanco, T. Suárez, M. Betanzos. Gene expression model for the classification of human colorectal cancer and potential CRC biomarkers search. *Drug Discovery Technology*, 2007.
- R. Armañanzas, B. Calvo, I. Inza, P. Larrañaga, I. Bernales, A. Fullaondo, A. M. Zubiaga. Bayesian classifiers with consensus gene selection: A case study in the systemic lupus erythematosus. *Progress in Industrial Mathematics at ECMI 2006*, 12, 560–565, 2007.
- R. Armañanzas, Y. Saeys, I. Inza, M. García-Torres, Y. Van de Peer, C. Bielza, P. Larrañaga. Mass spectrometry data analysis: it's all in the preprocessing. In *Proceedings of the Benelux Bioinformatics Conference*, page 92, 2008.
- A. García, R. Armañanzas, A. Freije, Z. Ispizua, F. Goñi, I. Inza, G. López Vivanco, B. Calvo, M. Betanzos, B. Suárez-Merino. Identification of tumoral molecular markers for the diagnosis and prognosis of colorectal carcinoma. *The Fourth Annual Biomarkers Congress*, 2009.

D. Refereed journals

- P. Larrañaga, B. Calvo, R. Santana, C. Bielza, J. Galdiano, I. Inza, J. A. Lozano, R. Armañanzas, G. Santafé, A. Pérez, V. Robles. Machine learning in bioinformatics. *Briefings in Bioinformatics*, 7(1), 86–112, 2006.
- I. Zipitria, P. Larrañaga, R. Armañanzas, A. Arruarte, J. A. Elorriaga, A. Díaz de Ilarraza. What Is Behind a Summary Evaluation Decision ?. *Behavior Research Methods*, 40(2), 597–612, 2008.
- R. Armañanzas, I. Inza, P. Larrañaga. Detecting reliable gene interactions by a hierarchy of Bayesian network classifiers. *Computer Methods and Programs in Biomedicine*, 91(2), 110–121, 2008.

- R. Armañanzas, I. Inza, R. Santana, Y. Saeys, J. L. Flores, J. A. Lozano, Y. van de Peer, R. Blanco, V. Robles, C. Bielza, P. Larrañaga. A review of estimation of distribution algorithms in bioinformatics. *BioData Mining*, 1(6), 2008.
- A. Sáenz, M. Azpitarte, R. Armañanzas, F. Leturcq, A. Alzualde, I. Inza, F. García-Bragado, G. De la Herran, J. Corcuera, A. Cabello, C. Navarro, C. De la Torre, E. Gallardo, I. Illa, A. López de Munain. Gene expression profiling in limb-girdle muscular dystrophy 2A. *PLoS ONE*, 3(11), e3750, 2008.
- R. Armañanzas, B. Calvo, I. Inza, M. López-Hoyos, V. Martínez-Taboada, E. Ucar, I. Bernales, A. Fullaondo, P. Larrañaga, A. M. Zubiaga. Microarray analysis of autoimmune diseases by machine learning procedures. *IEEE Transactions on Information Technology in Biomedicine*, 13(3), 341–350, 2009.
- D. Otaegui, S. E. Baranzini, R. Armañanzas, B. Calvo, M. Muñoz-Culla, P. Khankhanian, I. Inza, J. A. Lozano, T. Castillo-Triviño, J. Olaskoaga, A. López de Munain. Differential micro RNA expression in PBMC from Multiple Sclerosis patients. *PLoS ONE*. Submitted, 2009.
- R. Armañanzas, Y. Saeys, I. Inza, M. García-Torres, C. Bielza, Y. Van de Peer, P. Larrañaga. Peak selection in mass spectrometry data using a consensus approach with estimation of distribution algorithms. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*. Submitted, 2009.

E. Patents

- A. García, B. Suárez, M. Betanzos, G. L. Vivanco, R. Armañanzas, I. Inza, P. Larrañaga. *Methods and kits for the diagnosis and the staging of colorectal cancer*. European Patent No. 08380279.3-2402. Submitted 26th Sept. 2008.

13.2 Future Work

Computational biology is a discipline in its early beginnings. A large research effort is continuously made in this emerging field to shed light on the huge problems that the biomedical and biological disciplines undergo. As it is and has been proven through the last decade, machine learning and optimization techniques constitute the key to the analysis and knowledge discovery in genomics and proteomics fields. And this tendency will grow even more in the close future. Interdisciplinary researches are now a must on all these fields and the new created disciplines are here to stay (Stein, 2008).

There is nowadays a battery of new developed high-throughput biological devices. The gene expression platforms have grown very fast with new full genome arrays, exon arrays, SNP arrays or microRNA arrays. But the DNA

sequencing is again opening new frontiers in the genomic biology discipline with big DNA sequencers such as the ones from Solexa or 454. A lot of companies are in a constant run to cover more market and the investment in R+D is always increasing: Affymetrix, Agilent, Applied Biosystems, Beckman Coulter, Bio-Rad, Illumina, Invitrogen, PerkinElmer, Qiagen, Roche, Siemnes and so on.

The first consequence is that the biological datasets will increase their size proportionally to this growth. As all these devices gain popularity, their prices will decrease and personalized medicine will be at hand. From a practical point of view, this is the first big challenge for the computational biology: to provide accurate tools and resources to analyze personal data and give suggestions on diagnosis, treatments, possible adverse reactions or prognosis.

From a pure research point of view all these new amounts of data enable the research community to look for robust results and conclusions. And this is where consensus policies could have a great impact. As the computation power increases, the more robust and consistent a result becomes and the more reliable it is for the community. If we especially think of life sciences, this is of crucial importance since a false positive in a treatment or diagnosis implies a high personal price.

The feature selection has started this path by proposing new algorithms to increase the stability of the selections. The forseen tendency is that stability will constitute a criterion at least as important as the predictive accuracy. This is a new field of application and there is still little work done on it. Stability, consistency and robustness are three of the most expected subjects in the feature selection community in the years to come.

The afore mentioned new datasets will also contribute to study the regulatory relationships at different levels: genomic, proteomic or metabolomic. In this field, the regulatory networks will again gain importance in proposing and revealing biological dependences undiscovered by a pure biological analysis. Moreover, Bayesian strategies become more robust and reliable as the number of instances to estimate the probability distributions grows. Bayesian classifiers will follow this path and become an indispensable tool in the systems biology discipline. If we consider them as tools to support medical decisions, once the induction stage is fulfilled and the structure and parameters are learnt, the classification of new cases is straightforward by means of the chain rule.

Within the biomedical field, non-invasive diagnostic devices are also constantly evolving. The roots for all these devices is the identification of (early stages) biomarkers in non-invasive samples, such as plasma, serum or urine. Mass spectrometry is and will be essential in this task. However, since the data is and will be noisy, robust search methods will be demanded. Estimation of distribution algorithms have demonstrated their effectiveness in several biodata mining tasks. We envision that, in life sciences, the number of applications where EDAs get fruitful results will increase. And especially in noisy or with high uncertainty level domains, the consensus and robust searches

will prevail. In the short term, the exploitation of population consensus with probabilistic models more complex than UMDA is a task to tackle.

Lastly, as a general proposal, we envision the integration of the results from different omics into a single interdisciplinary research. Current studies from genomics rarely integrate proteomic or metabolomic data and vice versa. The integration of all these categories in a more complex view will be demanded. For instance, an interdisciplinary project will consist of performing different high throughput experiments to recover full datasets from those stored samples: starting with DNA microarrays, microRNA TLDA arrays, SNP chips and mass spectrometry runs. After that, all the info would be at hand from the very beginning and, thus, the combination of all these results could open a major degree of knowledge discovery in any *bioresearch*.

References

Achiron, A., Gurevich, M., Friedman, N., Kaminski, N., and Mandel, M. (2004). Blood transcriptional signatures of multiple sclerosis: Unique gene expression of disease activity. *Annals on Neurology*, **55**(3), 410–417.

Aha, D. W., Kibler, D., and Albert, M. K. (1991). Instance-based learning algorithms. *Machine Learning*, **6**(1), 37–66.

Al-Shahrour, F., Mínguez, P., Tárraga, J., Montaner, D., Alloza, E., Vaquerizas, J. M., Conde, L., Blaschke, C., Vera, J., and Dopazo, J. (2006). BABELOMICS: A systems biology perspective in the functional annotation of genome-scale experiments. *Nucleic Acids Research*, **34**, w472–w476.

Alarcón-Segovia, D. (2004). Shared autoimmunity: The time has come. *Current Rheumatology Reports*, **6**, 171–174.

Alarcón-Segovia, D., Pérez-Vázquez, M. A., and Villa, R. A. (1992). Preliminary classification criteria for the antiphospholipid syndrome within systemic lupus erythematosus. *Seminars in Arthritis and Rheumatism*, **21**, 275–286.

Alcina, A., Fedetz, M., Ndagire, D., Fernández, O., Leyva, L., Guerrero, M., Abad-Grau, M. M., Arnal, C., Delgado, C., Lucas, M., Izquierdo, G., and Matesanz, F. (2009). IL2RA/CD25 gene polymorphisms: Uneven association with multiple sclerosis (MS) and type 1 diabetes (T1D). *PLoS ONE*, **4**, e4137.

Alden, M. A. (2007). *MARLEDA: Effective Distribution Estimation Through Markov Random Fields*. Ph.D. thesis, Faculty of the Graduate Schoool, University of Texas at Austin, USA.

Alfassi, Z. B. (2004). On the normalization of a mass spectrum for comparison of two spectra. *Journal of The American Society for Mass Spectrometry*, **15**(3), 385–387.

Alizadeh, A. A., Eisen, M. B., Davis, R. E., Ma, C., Lossos, I. S., Rosenwald, A., Boldrick, J. C., Sabet, H., Tran, T., Yu, X., Powell, J. I., Yang, L., Marti, G. E., Moore, T., Hudson, J. J., Lu, L., Lewis, D. B., Tibshirani, R., Sherlock, G., Chan, W. C., Greiner, T. C., Weisenburger, D. D., Armitage, J. O., Warnke, R., and Staudt, L. M. (2000). Distinct types of diffuse

large B-cell lymphoma identified by gene expression profiling. *Nature*, **403**, 503–511.

Allen, W. L. and Johnston, P. G. (2005). Role of genomic markers in colorectal cancer treatment. *Journal of Clinical Oncology*, **23**, 4545–4552.

Alon, U., Barkai, N., Notterman, D. A., Gish, K., Ybarra, S., Mack, D., and Levine, A. J. (1999). Broad patterns of gene expression revelead by clustering analysis of tumor and normal colon tissues probed by olinucleotide arrays. *Proceedings of the National Academy of Sciences of the USA*, **96**(12), 6745–6750.

Ancona, N., Maglietta, R., Piepoli, A., D'Addabbo, A., Cotugno, R., Savino, M., Liuni, S., Carella, M., Pesole, G., and Perri, F. (2006). On the statistical assessment of classifiers using DNA microarray data. *BMC Bioinformatics*, **7**, 387.

Applied Biosystems (2001). Relative quantitation of gene expression. User bulletin P/N 4303859.

Aringer, M. and Smolen, J. S. (2004). Tumour necrosis factor and other proinflammatory cytokines in systemic lupus erythematosus: A rationale for therapeutic intervention. *Lupus*, **13**(5), 344–347.

Armañanzas, R. (2006). Solving bioinformatics problems by means of Bayesian classifiers and feature selection. Technical Report EHU-KZAA-IK-2/06, University of the Basque Country.

Armañanzas, R., Calvo, B., Inza, I., Larrañaga, P., Bernales, I., Fullaondo, A., and Zubiaga, A. M. (2005a). Clasificadores Bayesianos con selección consensuada de genes en la predicción del lupus eritematoso sistémico. In J. A. Gámez, I. G. Varea, and J. H. Orallo, editors, *Minería de Datos: Técnicas y Aplicaciones*, pages 107–135. Ediciones Departamento de Informática de la Universidad de Castilla La Mancha.

Armañanzas, R., Inza, I., and Larrañaga, P. (2005b). Consensus gene selection on DNA microarrays. In *2005 European Conference on Computational Biology*, Madrid, Spain.

Armañanzas, R., Calvo, B., Inza, I., Larrañaga, P., Bernales, I., Fullaondo, A., and Zubiaga, A. M. (2005c). Selección de genes asociados a dos enfermedades autoinmunes a partir de microarrays de ADN. In S. Moral and A. G. Serrano, editors, *VI Jornadas de Transferencia Tecnológica de Inteligencia Artificial, TTIA (AEPIA)*, pages 63–70.

Armañanzas, R., Calvo, B., Inza, I., Larrañaga, P., Bernales, I., Fullaondo, A., and Zubiaga, A. M. (2007). Bayesian classifiers with consensus gene selection: A case study in the systemic lupus erythematosus. In L. L. Bonilla, M. Moscoso, G. Platero, and J. M. Vega, editors, *Progress in Industrial Mathematics at ECMI 2006*, volume 12 of *Mathematics in Industry*, pages 560–565. Springer.

Armañanzas, R., Inza, I., and Larrañaga, P. (2008a). Detecting reliable gene interactions by a hierarchy of Bayesian network classifiers. *Computer Methods and Programs in Biomedicine*, **91**(2), 110–121.

Armañanzas, R., Saeys, Y., Inza, I., García-Torres, M., de Peer, Y. V., Bielza, C., and Larrañaga, P. (2008b). Mass spectrometry data analysis: It's all in the preprocessing. In *Proceedings of the Benelux Bioinformatics Conference*, page 92.

Armañanzas, R., Inza, I., Santana, R., Saeys, Y., Flores, J. L., Lozano, J. A., Van de Peer, Y., Blanco, R., Robles, V., Bielza, C., and Larrañaga, P. (2008c). A review of estimation of distribution algorithms in bioinformatics. *BioData Mining*, **1**(6).

Armañanzas, R., Calvo, B., Inza, I., López-Hoyos, M., Martínez-Taboada, V., Ucar, E., Bernales, I., Fullaondo, A., Larrañaga, P., and Zubiaga, A. M. (2009a). Microarray analysis of autoimmune diseases by machine learning procedures. *IEEE Transactions on Information Technology in Biomedicine*, **13**(3), 341–350.

Armañanzas, R., Saeys, Y., Inza, I., García-Torres, M., Bielza, C., Van de Peer, Y., and Larrañaga, P. (2009b). Peak selection in mass spectrometry data using a consensus approach with estimation of distribution algorithms. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*. Submitted.

Astler, V. B. and Coller, F. A. (1954). The prognostic significance of direct extension of carcinoma of the colon and rectum. *Annals of Surgery*, **139**(6), 846–852.

Bacardit, J., Stout, M., Hirst, J. D., Sastry, K., Llorà, X., and Krasnogor, N. (2007). Automated alphabet reduction method with evolutionary algorithms for protein structure prediction. In *Proceedings of the Genetic and Evolutionary Computation Conference GECCO-2007*, volume I, pages 346–353.

Baechler, E. C., Batliwalla, F. M., Karypis, G., Gaffney, P. M., Ortmann, W. A., Espe, K. J., Shark, K. B., Grande, W. J., Hughes, K. M., Kapur, V., Gregersen, P. K., and Behrens, T. W. (2003). Interferon-inducible gene expression signature in peripheral blood cells of patients with severe lupus. *Proceedings of the National Academy of Sciences of USA*, **100**(5), 2610–2615.

Baechler, E. C., Batliwalla, F. M., Reed, A. M., Peterson, E. J., Gaffney, P. M., Moser, K. L., Gregersen, P. K., and Behrens, T. W. (2006). Gene expression profiling in human autoimmunity. *Immunological Reviews*, **210**, 120–137.

Baggerly, K. A., Morris, J. S., and Coombes, K. R. (2004). Reproducibility of SELDI-TOF protein patterns in serum: Comparing data sets from different experiments. *Bioinformatics*, **20**, 777–785.

Baggerly, K. A., Morris, J. S., Edmonson, S. R., and Coombes, K. R. (2005). Signal in noise: Evaluating reported reproducibility of serum proteomic tests for ovarian cancer. *Journal of the National Cancer Institute*, **97**(4), 307–309.

Baker, S. G. and Kramer, B. S. (2006). Identifying genes that contribute most to good classification in microarrays. *BMC Bioinformatics*, **7**, 407.

Baltimore, D., Boldin, M. P., O'Connell, R. M., Rao, D. S., and Taganov, K. D. (2008). MicroRNAs: New regulators of immune cell development and function. *Nature Immunology*, **9**(8), 839–845.

Baluja, S. (1994). Population-based incremental learning: A method for integrating genetic search based function optimization and competitive learning. Technical Report CMU-CS-94-163, Carnegie Mellon University.

Baluja, S. and Davies, S. (1997). Using optimal dependency-trees for combinatorial optimization: Learning the structure of the search space. In *Proceedings of the 14th International Conference on Machine Learning*, pages 30–38.

Baranzini, S. E., Bernard, C. C., and Oksenberg, J. R. (2005a). Modular transcriptional activity characterizes the initiation and progression of autoimmune encephalomyelitis. *Journal of Immunology*, **174**, 7412–7422.

Baranzini, S. E., Mousavi, P., Rio, J., Caillier, S. J., Stillman, A., Villoslada, P., Wyatt, M. M., Comabella, M., Greller, L. D., Somogyi, R., Montalban, X., and Oksenberg, J. R. (2005b). Transcription-based prediction of response to IFN beta using supervised computational methods. *PLoS Biology*, **3**(1), e2.

Barla, A., Jurman, G., Riccadonna, S., Merler, S., Chierici, M., and Furlanello, C. (2008). Machine learning methods for predictive proteomics. *Briefings in Bioinformatics*, **9**(2), 119–128.

Bartel, D. P. (2004). MicroRNAs: Genomics, biogenesis, mechanism, and function. *Cell*, **116**, 281–297.

Bayly, R., Chuen, L., Currie, R. A., Hyndman, B. D., Casselman, R., Blobel, G. A., and LeBrun, D. P. (2004). E2A-PBX1 interacts directly with the KIX domain of CBP/p300 in the induction of proliferation in primary hematopoietic cells. *Journal of Biological Chemistry*, **279**(53), 55362–55371.

Belda, I., Madurga, S., Llorá, X., Martinell, M., Tarragó, T., Piqueras, M., Nicolás, E., and Giralt, E. (2005). ENPDA: An evolutionary structure-based de novo peptide design algorithm. *Journal of Computer-Aided Molecular Design*, **19**(8), 585–601.

Bellman, R. E. (1961). *Adaptive Control Processes*. Princeton University Press.

Ben-Bassat, M. (1982). Use of distance measures, information measures and error bounds in feature evaluation. In P. R. Krishnaiah and L. N. Kanal, editors, *Handbook of Statistics*, volume 2, pages 773–791. North-Holland Publishing Company.

Ben-Dor, A., Shamir, R., and Yakhini, Z. (1999). Clustering gene expression patterns. *Journal of Computational Biology*, **6**(3/4), 281–297.

Bennett, L., Palucka, A., Arce, E., Cantrell, V., Borvak, J., Banchereau, J., and Pascual, V. (2003). Interferon and granulopoiesis signatures in systemic lupus erythematosus blood. *Journal of Experimental Medicine*, **197**(6), 711–723.

Berkes, C. A. and Tapscott, S. J. (2005). MyoD and the transcriptional control of myogenesis. *Seminars in Cell and Developmental Biology*, **16**(4–5), 585–595.

Bertolaccini, M. L., Khamashta, M. A., and Hughes, G. R. (2005). Diagnosis of antiphospholipid syndrome. *Nature Clinical Practice Rheumatology*, **1**(1), 40–46.

Bezdek, J. C. (1981). *Pattern Recognition with Fuzzy Objective Function Algorithms*. Plenum Press.

Bhattacharyya, A. (1943). On a measure of divergence between two statistical populations defined by their probability distribution. *Bulletin of the Calcutta Mathematical Society*, **35**, 99–110.

Bielza, C., Robles, V., and Larrañaga, P. (2008). Estimation of distribution algorithms as logistic regression regularizers of microarray classifiers. *Methods of Information in Medicine*, **16**, 345–366.

Biett, M. (1857). *Cutaneous Diseases*. Lea & Carey, Philadelphia.

Birkenkamp-Demtroder, K., Christensen, L. L., Olesen, S. H., Frederiksen, C. M., Laiho, P., Aaltonen, L. A., Laurberg, S., Sørensen, F. B., Hagemann, R., and Ørntoft, T. F. (2002). Gene expression in colorectal cancer. *Cancer Research*, **62**, 4352–4363.

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer.

Blanco, R. (2005). *Learning Bayesian Networks from Data with Factorisation and Classification Purposes. Applications in Biomedicine*. Ph.D. thesis, University of the Basque Country.

Blanco, R., Larrañaga, P., Inza, I., and Sierra, B. (2004). Gene selection for cancer classification using wrapper approaches. *International Journal of Pattern Recognition and Artificial Intelligence*, **18**(8), 1373–1390.

Blanco, R., Inza, I., Merino, M., Quiroga, J., and Larrañaga, P. (2005). Feature selection in Bayesian classifiers for the prognosis of survival of cirrhotic patients treated with TIPS. *Journal of Biomedical Informatics*, **38**(5), 376–388.

Bombardier, C., Gladman, D. D., Urowitz, M. B., Caron, D., and Chang, C. H. (1992). Derivation of the SLEDAI. A disease activity index for lupus patients. The Committee on Prognosis Studies in SLE. *Arthritis & Rheumatism*, **35**(6), 630–640.

Bontempi, G. (2007). A blocking strategy to improve gene selection for classification of gene expression data. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, **4**(2), 293–300.

Bosman, P. A. and Thierens, D. (1999). Linkage information processing in distribution estimation algorithms. In *Proceedings of the Genetic and Evolutionary Computation Conference GECCO-1999*, volume I, pages 60–67.

Bouchard, L., Drapeau, V., Provencher, V., Lemieux, S., Chagnon, Y., Rice, T., Rao, D. C., Vohl, M. C., Tremblay, A., Bouchard, C., and Pèrusse, L. (2004). Neuromedin beta: A strong candidate gene linking eating behaviors and susceptibility to obesity. *The American Journal of Clinical Nutrition*, **80**(6), 1478–1486.

Bouckaert, R. R. and Frank, E. (2004). Evaluating the replicability of significance tests for comparing learning algorithms. In *Proceedings of the Pacific-Asia Conference on Knowledge Discovery and Data Mining - PAKDD*, pages 3–12.

Braga-Neto, U. M. (2005). Small-sample error estimation: Mythology versus mathematics. In *Proceedings of SPIE*, volume 5916, pages 304–314.

Braga-Neto, U. M. and Dougherty, E. R. (2004a). Bolstered error estimation. *Pattern Recognition*, **37**, 1267–1281.

Braga-Neto, U. M. and Dougherty, E. R. (2004b). Is cross-validation valid for small-sample microarray classification? *Bioinformatics*, **20**(3), 374–380.

Breen, J. B., Hopwood, F. G., Williams, K. L., and Wilkins, M. R. (2000). Automatic Poisson peak harvesting for high throughput protein identification. *Electrophoresis*, **21**, 2243–2251.

Breiman, L. (2001). Random forests. *Machine Learning*, **45**(1), 5–32.

Breiman, L., Friedman, J., Olshen, R., and Stone, C. (1984). *Classification and Regression Trees*. Wadsworth.

Burmester, G. R. and Haupl, T. (2004). Strategies using functional genomics in rheumatic diseases. *Autoimmunity Reviews*, **3**(7–8), 541–549.

Calvert, P. M. and Frucht, H. (2002). The genetics of colorectal cancer. *Annals of Internal Medicine*, **137**, 603–612.

Calvo, B. (2008). *Positive Unlabelled Learning with Applications in Computational Biology*. Ph.D. thesis, University of the Basque Country.

Cano, C., Blanco, A., García, F., and López, F. J. (2006). Evolutionary algorithms for finding interpretable patterns in gene expression data. *International Journal on Computer Science and Information System*, **1**(2), 88–99.

Castillo, E., Gutiérrez, J. M., and Hadi, A. S. (1997). *Expert Systems and Probabilistic Network Models*. Springer.

Catlett, J. (1991). On changing continuous attributes into ordered discrete attributes. In *Proceedings of the European Working Session on Learning*, pages 164–178.

Causton, H. C., Quackenbush, J., and Brazma, A. (2003). *Microarray Gene Expression Data Analysis: A Beginner's Guide*. Blackwell Publishers.

Cessie, S. L. and Houwelingen, J. C. v. (1992). Ridge estimators in logistic regression. *Applied Statistics*, **41**(1), 191–201.

Cestnik, B., Kononenko, I., and Bratko, I. (1987). ASSISTANT-86: A knowledge elicitation tool for sophisticated users. In I. Bratko and N. Lavrac, editors, *Progress in Machine Learning*, pages 31–45. Sigma Press.

Chapelle, O., Zien, A., and Schökopf, B. (2006). *Semi-supervised Learning*. MIT Press.

Chen, J. F., Murchison, E. P., Tang, R., Callis, T. E., Tatsuguchi, M., Deng, Z., Rojas, M., Hammond, S. M., Schneider, M. D., Selzman, C. H., Meissner, G., Patterson, C., Hannon, G. J., and Wang, D. Z. (2008a). Targeted deletion of Dicer in the heart leads to dilated cardiomyopathy and heart

failure. *Proceedings of the National Academy of Sciences of USA*, **105**(6), 2111–2116.

Chen, S. Y., Wang, Y., Telen, M. J., and Chi, J. T. (2008b). The genomic analysis of erythrocyte microRNA expression in sickle cell diseases. *PLoS ONE*, **3**, e2360.

Chen, Y., Kamat, V., Dougherty, E. R., Bittner, M. L., Meltzer, P. S., and Trent, J. M. (2002). Ratio statistics of gene expression levels and applications to microarray data analysis. *Bioinformatics*, **18**(9), 1207–1215.

Cheng, J. and Greiner, R. (2001). Learning Bayesian belief network classifiers: Algorithms and system. In *Proceedings of the Fourteenth Biennial Conference of the Canadian Society on Computational Studies of Intelligence: Advances in Artificial Intelligence*, pages 141–152.

Cheng, Y. and Church, G. M. (2000). Biclustering of expression data. In *Proceedings of the Eighth International Conference on Intelligent Systems for Molecular Biology*, pages 93–103.

Chow, C. K. and Liu, C. N. (1968). Approximating discrete probability distributions with dependence trees. *IEEE Transactions on Information Theory*, **IT-14**(3), 462–467.

Churchill, G. A. and Oliver, B. (2001). Sex, flies and microarrays. *Nature Genetics*, **29**, 355–356.

Clark, P. and Niblett, T. (1989). The CN2 induction algorithm. *Machine Learning*, **3**, 261–283.

Cochran, W. G. (1968). Commentary on estimation of error rates in discriminant analysis. *Technometrics*, **10**, 204–205.

Cole, S. W., Yan, W., Galic, Z., Arevalo, J., and Zack, J. A. (2005). Expression-based monitoring of transcription factor activity: The TELiS database. *Bioinformatics*, **21**(6), 803–810.

Condorcet, M. J. A. N. (1785). *Essai sur l'application de l'analyse à la probabilité des décisions rendues à la pluralité des voix*. Paris.

Coombes, K. R., Tsavachidis, S., Morris, J. S., Baggerly, K. A., Hung, M.-C., and Kuerer, H. M. (2005). Improved peak detection and quantification of mass spectrometry data acquired from surface-enhanced laser desorption and ionization by denoising spectra with the undecimated discrete wavelet transform. *Proteomics*, **5**(16), 4107–4117.

Coombes, K. R., Baggerly, K. A., and Morris, J. S. (2007). Pre-processing mass spectrometry data. In W. Dubitzky, M. Granzow, and D. Berrar, editors, *Fundamentals of Data Mining in Genomics and Proteomics*, pages 79–102. Springer.

Correa, P. A., Molina, J. F., Pinto, L. F., Arcos-Burgos, M., Herrera, M., and Anaya, J. M. (2003). TAP1 and TAP2 polymorphisms analysis in Northwestern Colombian patients with systemic lupus erythematosus. *Annals of the Rheumatic Diseases*, **62**(4), 363–365.

Cover, T. M. and Hart, P. E. (1967). Nearest neighbour pattern classification. *IEEE Transactions on Information Theory*, **13**, 21–27.

Cover, T. M. and Thomas, J. A. (1991). *Elements of Information Theory*. Wiley-Interscience.

Crawford, N. P., Colliver, D. W., and Galandiuk, S. (2003). Tumor markers and colorectal cancer: Utility in management. *Journal of Surgical Oncology*, **84**, 239–248.

Cristianini, N. and Shawe-Taylor, J. (2000). *An Introduction to Support Vector Machines*. Cambridge University Press.

Croker, J. A. and Kimberly, R. P. (2005). Genetics of susceptibility and severity in systemic lupus erythematosus. *Current Opinion in Rheumatology*, **17**(5), 529–537.

Cruz-Marcelo, A., Guerra, R., Vannucci, M., Li, Y., Lau, C. C., and Man, T. (2008). Comparison of algorithms for pre-processing of SELDI-TOF mass spectrometry data. *Bioinformatics*, **24**(19), 2129–2136.

Dai, C. and Liu, J. (2005). Inducing pairwise gene interactions from time series data by EDA based Bayesian network. In *Proceedings of the 27th Annual Conference of IEEE Engineering in Medicine and Biology*, pages 7746–7749.

De Bonet, J. S., Isbell, C. L., and Viola, P. (1997). MIMIC: Finding optima by estimating probability densities. In M. C. Mozer, M. I. Jordan, and T. Petsche, editors, *Advances in Neural Information Processing Systems*, volume 9, pages 424–430.

de Noo, M. E., Tollenaar, R. A., Özalp, A., Kuppen, P. J., Bladergroen, M. R., Eilers, P. H., and Deelder, A. M. (2005). Reliability of human serum protein profiles generated with C8 magnetic beads assisted MALDI-TOF mass spectrometry. *Analytical Chemistry*, **77**(22), 7232–7241.

de Waal, P. R. and Van der Gaag, L. C. (2007). Inference and learning in multi-dimensional Bayesian network classifiers. In *ECSQARU*, volume 4724 of *Lecture Notes in Computer Science*, pages 501–511.

de Yébenes, V. G., Belver, L., Pisano, D. G., González, S., Villasante, A., Croce, C., He, L., and Ramiro, A. R. (2008). miR-181b negatively regulates activation-induced cytidine deaminase in B cells. *The Journal of Experimental Medicine*, **205**(10), 2199–2206.

Deming, P. B. and Rathmell, J. C. (2006). Mitochondria, cell death, and b cell tolerance. *Current Directions in Autoimmunity*, **9**, 95–119.

Deng, A., Chen, S., Li, Q., Lyu, S. C., Clayberger, C., and Krensky, A. M. (2005). Granulysin, a cytolytic molecule, is also a chemoattractant and proinflammatory activator. *Journal of Immunology*, **174**, 5243–5248.

Denis, F., Gilleron, R., and Tommasi, M. (2002). Text classification from positive and unlabeled examples. In *The Nineth International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems, IPMU 2002*, pages 1927–1934.

Dill, K. A. (1985). Theory for the folding and stability of globular proteins. *Biochemistry*, **24**(6), 1501–1509.

Donoho, D. L. (2000). High-dimensional data analysis: The curses and blessings of dimensionality. In *Proceedings of the AMS Math Challenges of the 21st Century Conference*.

Drobyshev, A. L., Machka, C., Horsch, M., Seltmann, M., Liebscher, V., Hrabé de Angelis, M., and Beckers, J. (2003). Specificity assessment from fractionation experiments (SAFE): A novel method to evaluate microarray probe specificity based on hybridisation stringencies. *Nucleic Acids Research*, **31**(2), E1–11.

Duda, R. O. and Hart, P. E. (1973). *Pattern Classification and Scene Analysis*. John Wiley and Sons.

Duda, R. O., Hart, P. E., and Stork, D. G. (2001). *Pattern Classification*. Wiley Interscience.

Dudoit, S., Yang, Y. H., Callow, M. J., and Speed, T. P. (2002). Statistical methods for identifying differentially expressed genes in replicated cDNA microarray experiments. *Statistica Sinica*, **12**, 111–139.

Duffy, M. J. (2001). Clinical uses of tumor markers: A critical review. *Critical Reviews in Clinical Laboratory Sciences*, **38**, 225–262.

Dukes, C. E. (1932). The classification of cancer of the rectum. *Journal of Pathology & Bacteriology*, **35**, 323–332.

Edelman, G. M., Benacerraf, B., Ovary, Z., and Poulik, M. D. (1961). Structural differences among antibodies of different specificities. *Proceedings of the National Academy of Sciences of USA*, **47**, 1751–1758.

Efron, B. (1979). Bootstrap methods: Another look at the jackknife. *Annals of Statistics*, **7**, 1–26.

Efron, B. (1983). Estimating the error rate of a prediction rule: Improvement on cross-validation. *Journal of The American Statistical Association*, **78**, 316–331.

Elvira Consortium (2002). Elvira: An environment for probabilistic graphical models. In J. A. Gámez and A. Salmerón, editors, *Electronic Proceedings of the First European Workshop on Probabilistic Graphical Models*.

Etxeberria, R. and Larrañaga, P. (1999). Global optimization using Bayesian networks. In *Proceedings of the Second Symposium on Artificial Intelligence (CIMAF-99)*, pages 151–173.

Fathman, C. G., Soares, L., Chan, S. M., and Utz, P. J. (2005). An array of possibilities for the study of autoimmunity. *Nature*, **435**(7042), 605–611.

Fayyad, U. M. and Irani, K. B. (1993). Multi-interval discretization of continuous-valued attributes for classification learning. In *Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence*, pages 1022–1027.

Ferm, V. (1962). *Consensus Gentium*. Runes.

Fisher, R. A. (1936). The use of multiple measurements. *Annals of Eugenics*, **7**, 179–188.

Fleige, S. and Pfaffl, M. W. (2006). RNA integrity and the effect on the real-time qRT-PCR perfomance. *Molecular Aspects of Medicine*, **27**, 126–139.

Forgy, E. (1965). Cluster analysis of multivariate data: Efficiency vs. interpretability of classifications. *Biometrics*, **21**, 768–769.

Francí, C., Takkunen, M., Dave, N., Alameda, F., Gómez, S., Rodríguez, R., Escrivà, M., Montserrat-Sentís, B., Baró, T., Garrido, M., Bonilla, F., Virtanen, I., and de Herreros, A. G. (2006). Expression of SNAIL protein in tumor-stroma interface. *Oncogene*, **25**(37), 5134–5144.

Friedman, N. (1997). On bias, variance, 0/1 - loss, and the curve-of-dimensionality. *Data Mining and Knowledge Discovery*, **1**, 55–77.

Friedman, N. (2004). Inferring cellular networks using probabilistic graphical models. *Science*, **303**(5659), 799–805.

Friedman, N., Geiger, D., and Goldszmidt, M. (1997). Bayesian network classifiers. *Machine Learning*, **29**, 131–163.

Friedman, N., Goldsmidt, M., and Wyner, A. (1999). Data analysis with Bayesian networks: A bootstrap approach. In *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence*, pages 196–205.

Friedman, N., Linial, M., Nachman, I., and Pe'er, D. (2000). Using Bayesian networks to analyze expression data. *Journal of Computational Biology*, **7**, 601–620.

Fujita, M., Furukawa, Y., Tsunoda, T., Tanaka, T., Ogawa, M., and Nakamura, Y. (2001). Up-regulation of the ectodermal-neural cortex 1 (ENC1) gene, a downstream target of the beta-catenin/T-cell factor complex, in colorectal carcinomas. *Cancer Research*, **61**(21), 7722–7726.

Fukao, T., Matsuo, N., Zhang, G. X., Urasawa, R., Kubo, T., Kohno, Y., and Kondo, N. (2003). Single base substitutions at the initiator codon in the mitochondrial acetoacetyl-CoA thiolase (ACAT1/T2) gene result in production of varying amounts of wild-type T2 polypeptide. *Human Mutation*, **21**(6), 587–592.

Gámez, J. A., Mateo, J. L., and Puerta, J. M. (2007). EDNA: Estimation of dependency networks algorithm. In J. Mira and J. R. Álvarez, editors, *Bio-inspired Modeling of Cognitive Tasks, Second International Work-Conference on the Interplay Between Natural and Artificial Computation, IWINAC*, volume 4527 of *Lecture Notes in Computer Science*, pages 427–436.

Gammerman, A. and Thatcher, A. R. (1991). Bayesian diagnostic probabilities without assuming independence of symptoms. *Methods of Information in Medicine*, **30**, 15–22.

García, A., Freije, A., Armañanzas, R., Inza, I., Ispizua, Z., Heredia, P., Larrañaga, P., López Vivanco, G., Suárez, T., and Betanzos, M. (2006). Simultaneous search of genomic and proteomic biomarkers in human colorectal cancer. In *Genomes to Systems Conference*. Poster session.

García, A., Freije, A., Armañanzas, R., Inza, I., Ispizua, Z., Heredia, P., Larrañaga, P., López Vivanco, G., Suárez, T., and Betanzos, M. (2007). Gene expression model for the classification of human colorectal cancer and potential crc biomarkers search. In *Drug Discovery Technology*. Poster session.

García, A., Suárez, B., Betanzos, M., Vivanco, G. L., Armañanzas, R., Inza, I., and Larrañaga, P. (2008). Methods and kits for the diagnosis and the staging of colorectal cancer. European Patent No. 08380279.3-2402. Submitted 26th Sept. 2008.

García, A., Armañanzas, R., Freije, A., Ispizua, Z., Goñi, F., Inza, I., López Vivanco, G., Calvo, B., Betanzos, M., and Suárez-Merino, B. (2009). Identification of tumoral molecular markers for the diagnosis and prognosis of colorectal carcinoma. In *The Fourth Annual Biomarkers Congress*. Poster session.

Garey, M. R. and Johnson, D. S. (1979). *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman.

Geman, S. and Geman, D. (1984). Stochastic relaxation, Gibbs distributions, and Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **6**(6), 721–741.

Goldberg, D. E. (2002). *The Design of Innovation: Lessons from and for Competent Genetic Algorithms*. Kluwer Academic.

Golub, T. R., Slonim, D. K., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J. P., Coller, H., Loh, M. L., Downing, J. R., Caligiuri, M. A., Bloomfield, C. D., and Lander, E. S. (1999). Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring. *Science*, **286**(5439), 531–537.

Goutte, C. and Gaussier, E. (2005). A probabilistic interpretation of precision, recall and F-score, with implication for evaluation. In D. E. Losada and J. M. Fernández-Luna, editors, *Proceedings of the European Colloquium on IR Resarch (ECIR'05)*, volume LLNCS 3408, pages 345–359.

Greenbaum, S., Lazorchak, A. S., and Zhuang, Y. (2004). Differential functions for the transcription factor E2A in positive and negative gene regulation in pre-B lymphocytes. *Journal of Biological Chemistry*, **279**(43), 45028–45035.

Gregersen, P. K. and Behrens, T. W. (2006). Genetics of autoimmune diseases-disorders of immune homeostasis. *Nature Reviews Genetics*, **7**(12), 917–928.

Gregory, S. G., Schmidt, S., Seth, P., Oksenberg, J. R., Hart, J., Prokop, A., Caillier, S. J., Ban, M., Goris, A., Barcellos, L. F., Lincoln, R., McCauley, J. L., Sawcer, S. J., Compston, D. A., Dubois, B., Hauser, S. L., García-Blanco, M. A., Pericak-Vance, M. A., and Haines, J. L. (2007). Interleukin 7 receptor alpha chain (IL7R) shows allelic and functional association with multiple sclerosis. *Nature Genetics*, **39**, 1083–1091.

Griffiths, A. J. F., Gelbart, W. M., Lewontin, R. C., and Miller, J. H. (2002). *Modern Genetic Analysis: Integrating Genes and Genomes*. W. H. Freeman.

Griffiths-Jones, S. (2004). The microRNA registry. *Nucelic Acids Research*, **32**, D109–D111.

Griffiths-Jones, S., Grocock, R. J., van Dongen, S., Bateman, A., and Enright, A. J. (2006). miRBase: microRNA sequences, targets and gene nomenclature. *Nucelic Acids Research*, **34**, D140–D144.

Grimson, A., Farh, K. K.-H., Johnston, W. K., Garrett-Engele, P., Lim, L. P., and Bartel, D. P. (2007). MicroRNA targeting specificity in mammals: Determinants beyond seed pairing. *Molecular Cell*, **27**(1), 91–105.

Gross, J. H. (2006). *Mass Spectrometry: A Textbook*. Springer.

Haines, J. L., Ter-Minassian, M., Bazyk, A., Gusella, J. F., Kim, D. J., Terwedow, H., PericakVance, M. A., Rimmler, J. B., Haynes, C. S., Roses, A. D., Lee, A., Shaner, B., Menold, M., Seboun, E., Fitoussi, R., Gartioux, C., Reyes, C., Ribierre, F., Gyapay, G., Weissenbach, J., Hauser, S. L., Goodkin, D. E., Lincoln, R., Usuku, K., García-Merino, A., Gatto, N., Young, S., and Oksenberg, J. R. (1996). A complete genomic screen for multiple sclerosis underscores a role for the major histocompatibility complex. *Nature Genetics*, **13**, 469–471.

Hall, M. A. and Smith, L. A. (1997). Feature subset selection: A correlation based filter approach. In *Proceedings of the Fourth International Conference on Neural Information Processing and Intelligent Information Systems*, pages 855–858.

Hall, M. A. and Smith, L. A. (1999). Feature selection for machine learning: Comparing a correlation-based filter approach to the wrapper. In *Proceedings of the Florida Artificial Intelligence Research Symposium*, pages 235–239.

Hamerly, G. and Elkan, C. (2001). Bayesian approaches of failure prediction for disk drives. In *Proceedings of the Eighteenth International Conference on Machine Learning*, pages 202–209.

Hampel, F. R., Ronchetti, E. M., Rousseeuw, P. J., and Stahel, W. A. (2005). *Robust Statistics - The Approach Based on Influence Functions*. Wiley Series in Probability and Statistics. Wiley-Interscience.

Handi, J., Kell, D. B., and Knowles, J. (2007). Multiobjective optimization in bioinformatics and computational biology. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, **4**(2), 279–292.

Hardin, J. A. and Thomas, J. O. (1983). Antibodies to histones in systemic lupus erythematosus: Localization of prominent autoantigens on histones H1 and H2B. *Proceedings of the National Academy of Sciences of USA*, **80**(24), 7410–7414.

Harik, G. R., Lobo, F. G., and Goldberg, D. E. (1999). The compact genetic algorithm. *IEEE Transactions on Evolutionary Computation*, **3**(4), 287–297.

Harris, E. N. (1990). Annotation: Antiphospholipid antibodies. *British Journal of Haematology*, **74**, 1–9.

Hartemink, A. J., Gifford, D. K., Jaakkola, T. S., and Young, R. A. (2001). Using graphical models and genomic expression data to statistically validate models of genetic regulatory networks. In *Proceedings of the Pacific Symposium on Biocomputing*, volume 6, pages 422–433.

Hastie, T., Tibshirani, R., and Friedman, J. (2001). *The Elements of Statistical Learning: Data Mining, Inference and Prediction*. Springer-Verlag.

He, L., Thomson, J. M., Hemann, M. T., Hernando-Monge, E., Mu, D., Goodson, S., Powers, S., Cordon-Cardo, C., Lowe, S. W., Hannon, G. J., and Hammond, S. M. (2005). A microRNA polycistron as a potential human oncogene. *Nature*, **435**(7043), 828–833.

Heckerman, D., Geiger, D., and Chickering, D. M. (1995). Learning Bayesian networks: The combination of knowledge and statistical data. *Machine Learning*, **20**, 197–243.

Hilario, M., Kalousis, A., Pellegrini, C., and Müller, M. (2006). Processing and classification of protein mass spectra. *Mass Spectrometry Reviews*, **25**, 409–449.

Hills, M. (1966). Allocation rules and their error rates. *Journal of the Royal Statistical Society Series B*, **28**, 1–31.

Hirst, J. D. (1999). The evolutionary landscape of functional model proteins. *Protein Engineering*, **12**, 721–726.

Holland, J. H. (1975). *Adaptation in Natural and Artificial Systems*. The University of Michigan Press.

Holloway, A. J., van Laar, R. K., Tothill, R. W., and Bowtell, D. D. L. (2002). Options available –from start to finish– for obtaining data from DNA microarrays II. *Nature Genetics*, **32**(481–489).

Horiuchi, T., Morita, C., Tsukamoto, H., Mitoma, H., Sawabe, T., Harashima, S., Kashiwagi, Y., and Okamura, S. (2006). Increased expression of membrane TNF-alpha on activated peripheral CD8+ T cells in systemic lupus erythematosus. *International Journal of Molecular Medicine*, **17**(5), 875–879.

Horst, P. (1941). *Prediction of Personal Adjustment*, volume 48. Social Science Research Council.

Hosmer, D. and Lemeshow, S. (1989). *Applied Logistic Regression*. John Wiley and Sons.

Hsu, H. C., Wu, Y., and Mountz, J. D. (2006). Tumor necrosis factor ligand-receptor superfamily and arthritis. *Current Directions in Autoimmunity*, **9**, 37–54.

Hualupusng, C. M., Hang, L. W., Chen, C. L., Wu, J. Y., and Tsai, F. J. (2004). Polymorphisms of TAP1 transporter genes in Chinese patients with systemic lupus erythematosus in Taiwan. *Rheumatology International*, **24**(3), 130–132.

Hughes, G. R. V. (1983). Thrombosis, abortion, cerebral disease and the lupus anticoagulant. *British Medical Journal*, **287**, 1088–1089.

Hughes, T. R., Mao, M., Jones, A. R., Burchard, J., Marton, M. J., Shannon, K. W., Lefkowitz, S. M., Ziman, M., Schelter, J. M., Meyer, M. R., Kobayashi, S., Davis, C., Dai, H., He, Y. D., Stephaniants, S. B., Cavet, G., Walker, W. L., West, A., Coffey, E., Shoemaker, D. D., Stoughton, R., Blanchard, A. P., Friend, S. H., and Linsley, P. S. (2001). Expression profiling using microarrays fabricated by an ink-jet oligonucleotide synthesizer. *Nature Biotechnology*, **19**, 342–347.

Hutchens, T. W. and Yip, T. (1993). New desorption strategies for the mass spectrometric analysis of macromolecules. *Rapid Communications in Mass Spectrometry*, **7**(7), 576–580.

Ibañez-Ventoso, C., Vora, M., and Driscoll, M. (2008). Sequence relationships among C. elegans, D. melanogaster and human microRNAs highlight the extensive conservation of microRNAs in biology. *PLoS ONE*, **3**, e2818.

International Human Genome Sequencing Consortium (2001). Initial sequencing and analysis of the human genome. *Nature*, **409**(6822), 860–921.

International Human Genome Sequencing Consortium (2004). Finishing the euchromatic sequence of the human genome. *Nature*, **431**(7011), 931–945.

Inza, I., Larrañaga, P., Etxebarria, R., and Sierra, B. (1999). Feature subset selection by Bayesian networks based optimization. *Artificial Intelligence*, **27**, 143–164.

Inza, I., Merino, M., Larrañaga, P., Quiroga, J., Sierra, B., and Girala, M. (2001). Feature subset selection by genetic algorithms and estimation of distribution algorithms - a case study in the survival of cirrhotic patients treated with TIPS. *Artificial Intelligence in Medicine*, **23**(2), 187–205.

Inza, I., Larrañaga, P., Blanco, R., and Cerrolaza, A. J. (2004). Filter versus wrapper gene selection approaches in DNA microarray domains. *Artificial Intelligence in Medicine*, **31**(2), 91–103.

Inza, I., Armañanzas, R., and Santafé, G. (2006). Una aproximación al software WEKA. In B. Sierra, editor, *Aprendizaje Automático: Conceptos Básicos y Avanzados*, chapter 23, pages 477–483. Pearson Educación.

Inza, I., Calvo, B., Armañanzas, R., Bengoetxea, E., Larrañaga, P., and Lozano, J. A. (2009). Machine learning: An indispensable tool in bioinformatics. In R. Matthiesen, editor, *Bioinformatics Methods in Clinical Research*. Humana Press. In press.

Irizarry, R. A., Hobbs, B., Collin, F., Beazer-Barclay, Y. D., Antonellis, K. J., Scherf, U., and Speed, T. P. (2003). Exploration, normalization, and summaries of high density oligonucleotide array probe level data. *Biostatistics*, **4**(2), 249–264.

Jafari, P. and Azuaje, F. (2006). An assessment of recently published gene expression data analyses: Reporting experimental design and statistical factors. *BMC Medical Informatics and Decision Making*, **6**, 27.

Jardine, N. and Sibson, R. (1971). *Mathematical Taxonomy*. Wiley.

Jensen, F. V. and Nielsen, T. D. (2007). *Bayesian Networks and Decision Graphs*. Springer Verlag, second edition.

Jin, Z. G., Lungu, A. O., Xie, L., Wang, M., and Berk, C. W. B. C. (2004). Cyclophilin A is a proinflammatory cytokine that activates endothelial cells. *Arteriosclerosis, Thrombosis and Vascular Biology*, **24**(7), 1186–1191.

Johansson, C. (2004). *Exploring the Genetics of SLE with Linkage and Association Analysis*. Ph.D. thesis, Uppsala University.

John, G. H. and Langley, P. (1995). Estimating continuous distributions in Bayesian classifiers. In *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*, pages 338–345.

Kalousis, A., Prados, J., and Hilario, M. (2005). Stability of feature selection algorithms. In *Proceedings of the Fifth IEEE International Conference on Data Mining*, pages 218–225.

Kanji, G. K. (2006). *100 Statistical Tests*. SAGE publications.

Karas, M., Bachmann, D., Bahr, U., and Hillenkamp, F. (1987). Matrix-assisted ultraviolet laser desorption of non-volatile compounds. *International Journal of Mass Spectrometry and Ion Processes*, **78**, 53–68.

Kemeny, J. G. (1959). Mathematics without numbers. *Daedalus*, **88**, 577–591.

Kendziorski, C., Irizarry, R. A., Chen, K. S., Haag, J. D., and Gould, M. N. (2005). On the utility of pooling biological samples in microarray experiments. *Proceedings of the National Academy of Sciences of the USA*, **102**(12), 4252–4257.

Kerber, R. (1992). Chimerge: Discretization for numeric attributes. In *National Conference on Artificial Intelligence*, pages 123–128.

Kim, B. S., Kim, I., Lee, S., Kim, S., Rha, S. Y., and Chung, H. C. (2005). Statistical methods of translating microarray data into clinically relevant diagnostic information in colorectal cancer. *Bioinformatics*, **21**(4), 517–528.

Kitahara, O., Furukawa, Y., Tanaka, T., Kihara, C., Ono, K., Yanagawa, R., Nita, M. E., Takagi, T., Nakamura, Y., and Tsunoda, T. (2001). Alterations of gene expression during colorectal carcinogenesis revealed by cDNA microarrays after laser-capture microdissection of tumor tissues and normal epithelia. *Cancer Research*, **61**, 3544–3549.

Kleinbaum, D. G. (1994). *Logistic Regression. Statistics in Health Sciences*. Springer-Verlag.

Kohavi, R. (1995). *Wrapper for Performance Enhancement and Oblivious Decision Graphs*. Ph.D. thesis, Stanford University.

Kohavi, R. and John, G. (1997). Wrappers for feature subset selection. *Artificial Intelligence*, **97**(1–2), 273–324.

Kohavi, R. and Wolpert, D. H. (1996). Bias plus variance decomposition for zero-one loss functions. In *Proceedings of the Thirdteenth International Machine Learning Conference*, pages 275–283.

Kononenko, I. (1990). Comparison of inductive and naïve Bayesian learning approaches to automatic knowledge acquisition. In B. Wielinga, J. Boose, B. Gaines, G. Shereiber, and M. van Someren, editors, *Current Trends in Knowledge Acquisition*, pages 190–197.

Kononenko, I. (1991). Semi-naïve Bayes classifiers. In *Proceedings of the Sixth European Working Session on Learning*, pages 206–219.

Koskenmies, S. (2004). *Mapping of Susceptibility Genes for Systemic Lupus Erythematosus (SLE)*. Ph.D. thesis, University of Helsinki.

Krek, A., Grün, D., Poy, M. N., Wolf, R., Rosenberg, L., Epstein, E. J., MacMenamin, P., da Piedade, I., Gunsalus, K. C., Stoffel, M., and Rajewsky, N. (2005). Combinatorial microRNA target predictions. *Nature Genetics*, **37**(5), 495–500.

Kullback, S. (1987). The Kullback-Leibler distance. *The American Statistician*, **41**, 340–341.

Kuncheva, L. I. (2004). *Combining Pattern Classifiers: Methods and Algorithms*. Wiley-IEEE.

Kuncheva, L. I. (2007). A stability index for feature selection. In *Proceedings of the 25th IASTED International Multi-Conference, Artificial Intelligence and Applications*, pages 390–395.

Kurtzke, J. F. (1983). Rating neurologic impairment in multiple sclerosis: An expanded disability status scale (EDSS). *Neurology*, **33**(11), 1444–1452.

Lachenbruch, P. and Mickey, M. (1968). Estimation of error rates in discriminant analysis. *Technometrics*, **10**, 1–11.

Langley, P. and Sage, S. (1994). Induction of selective Bayesian classifiers. In *Proceedings of the Tenth Conference on Uncertainty in Artificial Intelligence*, pages 399–406.

Larrañaga, P. (2002). *Estimation of Distribution Algorithms. A New Tool for Evolutionary Computation*, chapter A review on estimation of distribution algorithms, pages 55–98. Kluwer Academic Publishers.

Larrañaga, P. and Lozano, J. A. (2002). *Estimation of Distribution Algorithms. A New Tool for Evolutionary Computation*. Kluwer Academic Publishers.

Larrañaga, P., Lozano, J. A., Peña, J. M., and Inza, I. (guest editors) (2005). Special issue on probabilistic graphical models for classification. *Machine Learning*, **59**(3).

Larrañaga, P., Calvo, B., Santana, R., Bielza, C., Galdiano, J., Inza, I., Lozano, J. A., Armañanzas, R., Santafé, G., Pérez, A., and Robles, V. (2006). Machine learning in bioinformatics. *Briefings in Bioinformatics*, **7**(1), 86–112.

Larson, S. C. (1931). The shrinkage of the coefficient of multiple correlation. *Journal of Educational Psychology*, **22**, 45–55.

Lauritzen, S. L. (1996). *Graphical Models*. Oxford University Press.

Lauritzen, S. L. and Spiegelhalter, D. J. (1988). Local computations with probabilities on graphical structures and their application to expert systems. *Journal of the Royal Statistical Society Series B*, **50**(2), 157–224.

Leclerc, R. D. (2008). Survival of the sparsest: Robust gene networks are parsimonious. *Molecular Systems Biology*, **4**, 213.

Lee, E., Salic, A., and Kirschner, M. W. (2001). Physiological regulation of beta-catenin stability by Tcf3 and CK1epsilon. *Journal of Cell Biology*, **154**(5), 983–993.

Lee, J. W., Lee, J. B., Park, M., and Song, S. H. (2005). An extensive comparison of recent classification tools applied to microarrays data. *Computational Statistics & Data Analysis*, **48**, 869–885.

Lee, R. C., Feinbaum, R. L., and Ambros, V. (1993). The C. elegans heterochronic gene lin-4 encodes small RNAs with antisense complementarity to lin-14. *Cell*, **75**(5), 834–854.

Lee, Y. H., Harley, J. B., and Nath, S. K. (2006). Meta-analysis of TNF-alpha promoter -308 A/G polymorphism and SLE susceptibility. *European Journal of Human Genetics*, **14**(3), 364–371.

Lewis, B. P., Shih, I. H., Jones-Rhoades, M. W., Bartel, D. P., and Burge, C. B. (2003). Prediction of mammalian microRNA targets. *Cell*, **115**, 787–798.

Lewis, B. P., Burge, C. B., and Bartel, D. P. (2005). Conserved seed pairing, often flanked by adenosines, indicates that thousands of human genes are microRNA targets. *Cell*, **120**, 15–20.

Li, C. and Wong, W. H. (2003). DNA-chip analyzer (dChip). In G. Parmigiani, E. S. Garrett, R. Irizarry, and S. L. Zeger, editors, *The Analysis of Gene Expression Data: Methods and Software*. Springer.

Li, L., Pedersen, L. G., Darden, T. A., and Weinberg, C. R. (2001). Computational analysis of leukemia microarray expression data using the GA/KNN method. In S. M. Lin and K. F. Johnson, editors, *Methods of Microarray Data Analysis: Papers from CAMDA'00*, pages 81–95. Kluwer Academic.

Li, T., Zhang, C., and Ogihara, M. (2004). A comparative study of feature selection and multiclass classification methods for tissue classification based on gene expression. *Bioinformatics*, **20**(15), 2429–2437.

Li, W. and Yang, Y. (2002). How many genes are needed for a discriminant microarray data analysis? In S. M. Lin and K. F. Johnson, editors, *Methods of Microarray Data Analysis: Papers from CAMDA'00*, pages 137–150.

Liang, F., Seyrantepe, V., Landry, K., Ahmad, R., Ahmad, A., Stamatos, N. M., and Pshezhetsky, A. V. (2006). Monocyte differentiation up-regulates the expression of the lysosomal sialidase, Neu1, and triggers its targeting to the plasma membrane via major histocompatibility complex class II-positive compartments. *Journal of Biological Chemistry*, **281**(37), 27526–27538.

Lin, T., Liu, R., Chen, C., Chao, Y., and Chen, S. (2006). Pattern classification in DNA microarray data of multiple tumor types. *Pattern Recognition*, **39**, 2426–2438.

Liu, H. and Motoda, H. (1998). *Feature Selection for Knowledge Discovery and Data Mining*. Kluwer Academic Publishers.

Liu, H. and Motoda, H., editors (2008). *Computational Methods of Feature Selection*. Chapman and Hall/CRC Press.

Liu, H. and Yu, L. (2005). Toward integrating feature selection algorithms for classification and clustering. *IEEE Transactions on Knowledge and Data Engineering*, **17**(4), 491–502.

Lockhart, D. J., Dong, H., Byrne, M. C., Follettie, M. T., Gallo, M. V., Chee, M. S., Mittmann, M., Wang, C., Kobayashi, M., Norton, H., and Brown, E. L. (1996). Expression monitoring by hybridization to high-density oligonucleotide arrays. *Nature Biotechnology*, **14**, 1675–1680.

Lozano, J. A., Larrañaga, P., Inza, I., and Bengoetxea, E., editors (2006). *Towards a New Evolutionary Computation: Advances on Estimation of Distribution Algorithms*. Springer-Verlag.

Lucas, P. (2004). *Advances in Bayesian Networks*, chapter Restricted Bayesian network structure learning, pages 217–234. Springer-Verlag.

Maddison, P. J., Isenberg, D. A., Goulding, N. J., Leddy, J., and Skinner, R. P. (1988). Anti La(SSB) identifies a distinctive subgroup of systemic lupus erythematosus. *British Journal of Rheumatology*, **27**(1), 27–31.

Majoros, W. (2007). *Methods for Computational Gene Prediction*. Cambridge University Press.

Mandel, M., Gurevich, M., Pauzner, R., Kaminski, N., and Achiron, A. (2004). Autoimmunity gene expression portrait: Specific signature that intersects or differentiates between multiple sclerosis and systemic lupus erithematosus. *Clinical & Experimental Immunology*, **138**, 164–170.

Manevitz, L. M. and Yousef, M. (2001). One-class SVMs for document classification. *Journal of Machine Learning Research*, **2**, 139–154.

Mani, S., Pazzani, M., and West, J. (1997). Knowledge discovery from a breast cancer database. In *Proceedings of the Sixth Conference on Artificial Intelligence in Medicine in Europe*, volume 1211 of *Lecture Notes in Computer Science*, pages 130–133. Springer-Verlag.

Mann, H. B. and Whitney, D. R. (1947). On a test of whether one of 2 random variables is stochastically larger than the other. *Annals of Mathematical Statistics*, **18**, 50–60.

Mathé, C., Sagot, M., Schiex, T., and Rouzé, P. (2002). Current methods of gene prediction, their strengths and weaknesses. *Nucleic Acids Research*, **30**(19), 4103–4117.

Matusiak, D., Glover, S., Nathaniel, R., Matkowskyj, K., Yang, J., and Benya, R. V. (2005). Neuromedin B and its receptor are mitogens in both normal and malignant epithelial cells lining the colon. *American Journal of Physiology. Gastrointestinal and Liver Physiology*, **288**(4), G718–G728.

Matusita, K. (1955). Decision rules based on distance for problems of fit, two samples and estimation. *Annals of Mathematical Statistics*, **26**, 631–641.

Matute, C. (2007). Interaction between glutamate signalling and immune attack in damaging oligodendrocytes. *Neuron Glia Biology*, **3**(4), 281–285.

McCallum, A. and Nigam, K. (1998). A comparison on event models for naïve Bayes text classification. In *Proceedings of the Fifteenth American Association for Artificial Intelligence. Workshop on Learning for Text Categorization*, pages 41–48.

McCulloch, W. S. and Pitts, W. H. (1943). A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, **5**, 115–133.

McDonald, W. I., Compston, A., Edan, G., Goodkin, D., Hartung, H. P., Lublin, F. D., McFarland, H. F., Paty, D. W., Polman, C. H., Reingold, S. C., Sandberg-Wollheim, M., Sibley, W., Thompson, A., van den Noort, S., Weinshenker, B. Y., and Wolinsky, J. S. (2001). Recommended diagnostic criteria for multiple sclerosis: Guidelines from the International Panel of the diagnosis of multiple sclerosis. *Annals of Neurology*, **50**(1), 121–127.

Meuleman, W., Engwegen, J. Y., Gast, M. W., Beijnen, J. H., Reinders, M. J., and Wessels, L. F. (2008). Comparison of normalisation methods

for surface-enhanced laser desorption and ionisation (SELDI) time-of-flight (TOF) mass spectrometry data. *BMC Bioinformatics*, **9**(88).

Mi, H., Guo, N., Kejariwal, A., and Thomas, P. (2007). PANTHER version 6: Protein sequence and function evolution data with expanded representation of biological pathways. *Nucleic Acids Research*, **37**, D247–D252.

Michiels, S., Koscielny, S., and Hill, C. (2005). Prediction of cancer outcome with microarrays: A multiple random validation strategy. *The Lancet*, **365**, 488–492.

Minsky, M. (1961). Steps toward artificial intelligence. *Transactions on Institute of Radio Engineers*, **49**, 8–30.

Miyahara, K. and Pazzani, M. (2000). Collaborative filtering with the simple Bayesian classifier. In *Proceedings of the Pacific Rim International Conference on Artificial Intelligence*, pages 679–689.

Mladenović, N. (1995). A variable neighborhood algorithm – a new metaheuristics for combinatorial optimization. In *Abstracts of Papers Presented at Optimization Days*, page 112.

Monti, S., Tamayo, P., Mesirov, J., and Golub, T. (2003). Consensus clustering: A resampling-based method for class discovery and visualization of gene expression microarray data. *Machine Learning*, **52**(1–2), 91–118.

Moreno, V. and Solé, X. (2004). Uso de chips de ADN (microarrays) en medicina: Fundamentos técnicos y procedimientos básicos para el análisis estadístico de resultados. *Medicina Clínica*, **122(Supl 1)**, 73–79.

Morgan, J. and Sonquistz, J. (1963). Problems in the analysis of survey data, and a proposal. *Journal of the American Statistical Association*, **58**, 415–434.

Mosteller, F. and Tukey, J. W. (1968). Data analysis, including statistics. In G. Lindzey and E. Aronson, editors, *Handbook of Social Psycholgy*, volume 2. Addison-Wesley.

Movellan, J., Wachtler, T., Albright, T., and Sejnowski, T. (2002). Naive Bayes coding of color in primary visual cortex. In *Proceedings of the Neural Information Processing Systems*, volume 15, pages 221–228.

Mühlenbein, H. and Mahnig, T. (2001). Evolutionary synthesis of Bayesian networks for optimization. In M. Patel, V. Honavar, and K. Balakrishnan, editors, *Advances in Evolutionary Synthesis of Intelligent Agents*, pages 429–455. MIT Press.

Mühlenbein, H. and Paaß, G. (1996). From recombination of genes to the estimation of distributions. I. Binary parameters. In *Lecture Notes in Computer Science 1411: Parallel Problem Solving from Nature, PPSN IV*, pages 178–187.

Mühlenbein, H., Mahnig, T., and Ochoa, A. (1999). Schemata, distributions and graphical models in evolutionary optimization. *Journal of Heuristics*, **5**(2), 213–247.

Mullis, K. (1990). The unusual origin of the polymerase chain reaction. *Scientific American*, **262**(4), 56–61.

Murayama, A., Kim, M. S., Yanagisawa, J., Takeyama, K., and Kato, S. (2004). Transrepression by a liganded nuclear receptor via a bHLH activator through co-regulator switching. *The EMBO Journal*, **23**(7), 1598–1608.

Murga, M., Fernández-Capetillo, O., Field, S. J., Moreno, B., Borlado, L. R., Fujiwara, Y., Balomenos, D., Vicario, A., Carrera, A. C., Orkin, S. H., Greenberg, M. E., and Zubiaga, A. M. (2001). Mutation of E2F2 in mice causes enhanced t lymphocyte proliferation, leading to the development of autoimmunity. *Immunity*, **15**(6), 959–970.

Nadeau, C. and Bengio, Y. (2003). Inference for the generalization error. *Machine Learning*, **52**, 239–281.

Neapolitan, E. (2003). *Learning Bayesian Networks*. Prentice Hall.

Nelson, P. T., Wang, W. X., and Rajeev, B. W. (2008). MicroRNAs (miRNAs) in neurodegenerative diseases. *Brain Pathology*, **18**(1), 130–138.

Nevins, J. R. (2001). The Rb/E2F pathway and cancer. *Human Molecular Genetics*, **10**(7), 699–703.

Ng, A. (1997). Preventing overfitting of cross-validation data. In *Proceedings of Fourteenth International Conference on Machine Learning*, pages 245–253.

NIAMS (1994). What black women should know about lupus. NIH Publication 93-3219, National Institute of Arthritis and Musculoskeletal and Skin Diseases.

Nolan, T., Hands, R. E., and Bustin, S. A. (2006). Quantification of mRNA using real-time RT-PCR. *Nature Protocols*, **1**(3), 1559–1582.

Notterman, D. A., Alon, U., Sierk, A. J., and Levine, A. J. (2001). Transcriptional gene expression profiles of colorectal adenoma, adenocarcinoma, and normal tissue examined by oligonucleotide arrays. *Cancer Research*, **61**, 3124–3130.

Noy, K. and Fasulo, D. (2007). Improved model-based, platform-independent feature extraction for mass spectrometry. *Bioinformatics*, **23**(19), 2528–2535.

Ochoa, A., Mühlenbein, H., and Soto, M. R. (2000a). A factorized distribution algorithm using single connected Bayesian networks. In M. Schoenauer, K. Deb, G. Rudolph, X. Yao, E. Lutton, J. J. Merelo, and H. Schwefel, editors, *Parallel Problem Solving from Nature - PPSN VI 6th International Conference*, pages 787–796.

Ochoa, A., Mühlenbein, H., and Soto, M. R. (2000b). Factorized distribution algorithms using Bayesian networks bounded complexity. In *Proceedings of the Genetic and Evolutionary Computation Conference GECCO-2000*, pages 212–215.

O'Connell, J. B., Maggard, M. A., and Ko, C. Y. (2004). Colon cancer survival rates with the new American Joint Committee on Cancer sixth edition staging. *Journal of the National Cancer Institute*, **96**(19), 1420–1425.

O'Connell, R. M., Taganov, K. D., Boldin, M. P., Cheng, G., and Baltimore, D. (2007). MicroRNA-155 is induced during the macrophage inflammatory

response. *Proceedings of the National Academy of Sciences of USA*, **104**, 1604–1609.

O'Donnell, K. A., Wentzel, E. A., Zeller, K. I., Dang, C. V., and Mendell, J. T. (2005). c-Myc-regulated microRNAs modulate E2F1 expression. *Nature*, **435**(7043), 839–843.

Ohmann, C., Yang, Q., Kunneke, M., Stolzing, H., Thon, K., and Lorenz, W. (1988). Bayes theorem and conditional dependence of symptoms: Different models applied to data of upper gastrointestinal bleeding. *Methods of Information in Medicine*, **27**, 73–83.

Ohmann, C., Moustakis, V., Yang, Q., and Lang, K. (1996). Evaluation of automatic knowledge acquisition techniques in the diagnosis of acute abdominal pain. *Artificial Intelligence in Medicine*, **8**, 23–36.

Oksenberg, J. R. and Barcellos, L. F. (2005). Multiple sclerosis genetics: Leaving no stone unturned. *Genes and Immunity*, **6**, 375–387.

Otaegui, D., Mostafavi, S., Bernard, C. C., López de Munain, A., Mousavi, P., Oksenberg, J. R., and Baranzini, S. E. (2007). Increased transcriptional activity of milk-related genes following the active phase of experimental autoimmune encephalomyelitis and multiple sclerosis. *The Journal of Immunology*, **179**, 4074–4082.

Otaegui, D., Baranzini, S. E., Armañanzas, R., Calvo, B., Muñoz-Culla, M., Khankhanian, P., Inza, I., Lozano, J. A., Castillo-Triviño, T., Olaskoaga, J., and López de Munain, A. (2009). Differential micro RNA expression in PBMC from multiple sclerosis patients. *PLoS ONE*. Submitted.

Palacios, P., Pelta, D. A., and Blanco, A. (2006). Obtaining biclusters in microarrays with population-based heuristics. In *EvoWorkshops*, volume 3907, pages 115–126.

Pareto, V. (1896). *Cours D'Economie Politique*, volume I and II. Lausanne.

Patel, R. K. and Mohan, C. (2005). PI3K/AKT signaling and systemic autoimmunity. *Immunology Research*, **31**(1), 47–55.

Paul, T. K. and Iba, H. (2004). Identification of informative genes for molecular classification using probabilistic model building genetic algorithms. In *Proceedings of the Genetic and Evolutionary Computation Conference GECCO-2004. Lecture Notes in Computer Science 3102*, pages 414–425.

Paul, T. K. and Iba, H. (2005). Gene selection for classification of cancers using probabilistic model building genetic algorithm. *BioSystems*, **82**(3), 208–225.

Pazzani, M. (1997). *Searching for dependencies in Bayesian classifiers*, pages 239–248. Artificial Intelligence and Statistics IV, Lecture Notes in Statistics. Springer-Verlag.

Pazzani, M., Murumatsu, J., and Billsus, D. (1996). Syskill and Webert: Identifying interesting web sites. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence*, pages 54–61.

Pearl, J. (1988). *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann Publishers.

Pe'er, D., Regev, A., Elidan, G., and Friedman, N. (2001). Inferring subnetworks from perturbed expression profiles. *Bioinformatics*, **17**, S215–S224.

Pe'er, D., Tanay, A., and Regev, A. (2006). Minreg: A scalable algorithm for learning parsimonious regulatory networks in yeast and mammals. *Journal of Machine Learning Research*, **7**, 167–189.

Pelikan, M. (2005). *Hierarchical Bayesian Optimization Algorithm. Toward a New Generation of Evolutionary Algorithms*, volume 170 of *Studies in Fuzziness and Soft Computing*. Springer.

Pelikan, M. and Mühlenbein, H. (1999). The bivariate marginal distribution algorithm. In R. Roy, T. Furuhashi, and P. K. Chawdhry, editors, *Advances in Soft Computing - Engineering Design and Manufacturing*, pages 521–535. Springer-Verlag.

Pelikan, M., Goldberg, D., and Cantú-Paz, E. (1999). BOA: The Bayesian optimization algorithm. In *Proceedings of the Genetic and Evolutionary Computation Conference GECCO-1999*, pages 525–532.

Pelikan, M., Goldberg, D. E., and Lobo, F. (2002). A survey of optimization by building and using probabilistic models. *Computational Optimization and Applications*, **21**(1), 5–20.

Peña, C., García, J. M., Silva, J., García, V., Rodríguez, R., Alonso, I., Millán, I., Salas, C., de Herreros, A. G., Muñoz, A., and Bonilla, F. (2005a). E-cadherin and vitamin D receptor regulation by SNAIL and ZEB1 in colon cancer: Clinicopathological correlations. *Human Molecular Genetics*, **14**(22), 3361–3370.

Peña, J. M., Lozano, J. A., and Larrañaga, P. (2004). Unsupervised learning of Bayesian networks via estimation of distribution algorithms: An application to gene expression data clustering. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, **12**, 63–82.

Peña, J. M., Björkegren, J., and Tegnér, J. (2005b). Growing Bayesian network models of gene networks from seed genes. *Bioinformatics*, **21**(Suppl. 2), ii224–ii229.

Petricoin, E. F., Ardekani, A. M., Hitt, B. A., Levine, P. J., Fusaro, V. A., Steinberg, S. M., Mills, G. B., Simone, C., Fishman, D. A., Kohn, E. C., and Liotta, L. A. (2002). Use of proteomic patterns in serum to identify ovarian cancer. *Lancet*, **359**(9306), 572–577.

Petricoin, E. F., Rajapaske, V., Herman, E. H., Ardekani, A. M., Ross, S., Johann, D., Knapton, A., Zhang, J., Hitt, B. A., Conrads, T. P., Veenstra, T. D., Liotta, L. A., and Sistare, F. D. (2004). Toxicoproteomics: Serum proteomic pattern diagnostics for early detection of drug induced cardiac toxicities and cardioprotection. *Toxicologic Pathology*, **32**(Suppl. 1), 122–130.

Pickett, S. C. (2003). Understanding and evaluating fluorescent microarray imaging instruments. *IVD Technology*, **9**(4), 45–50.

Polyak, K., Xia, Y., Zweier, J. L., Kinzler, K. W., and Vogelstein, B. (1997). A model for p53-induced apoptosis. *Nature*, **389**(6648), 300–305.

Pons, M., Mendibil, M., Arsich, A., Rojas, I., Kuhlmann, T., and Carranza, D. (2001). Lupus eritematoso sistémico y síndrome antifosfolipídico. *Archivos Argentinos de Pediatría*, **99**(4), 354–359.

Poser, C. M. (2006). Revisions to the 2001 McDonald diagnostic criteria. *Annals of Neurology*, **59**(4), 727–728.

Prados, J., Kalousis, A., and Hilario, M. (2006). On preprocessing of SELDI-MS data and its evaluation. In *Proceedings of the 19th IEEE Symposium on Computer-Based Medical Systems*, pages 953–958.

Provost, F., Fawcett, T., and Kohavi, R. (1998). The cases against accuracy estimation for comparing induction algorithms. In *Proceedings of the Fifteenth International Conference on Machine Learning*, pages 445–453.

Ramanathan, M., Weinstock-Guttman, B., Nguyen, L. T., Badgett, D., Miller, C., Patrick, K., Brownscheidle, C., and Jacobs, L. (2001). In vivo gene expression revealed by cDNA arrays: The pattern in relapsing-remitting multiple sclerosis patients compared with normal subjects. *Journal of Neuroimmunology*, **116**(2), 213–219.

Rao, J. and Shao, A. (1992). Jackknife variance estimation with survey data under hot deck imputation. *Biometrika*, **79**, 811–822.

Rapaport, F., Zinovyev, A., Dutreix, M., Barillot, E., and Vert, J. (2007). Classification of microarray data using gene networks. *BMC Bioinformatics*, **8**(1), 35.

Ressom, H. W., Varghese, R. S., Abdel-Hamid, M., Eissa, S. A., Saha, D., Goldman, L., Petricoin, E. F., Conrads, T. P., Veenstra, T. D., Loffredo, C. A., and Goldman, R. (2005). Analysis of mass spectral serum profiles for biomarker selection. *Bioinformatics*, **21**(21), 4039–4045.

Ressom, H. W., Varghese, R. S., Orvisky, E., Drake, S. K., Hortin, G. L., Abdel-Hamid, M., Loffredo, C. A., and Goldman, R. (2006). Ant colony optimization for biomarker identification from MALDI-TOF mass spectra. In *Proceedings of the 28th International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 4560–4563.

Ressom, H. W., Varghese, R. S., Drake, S. K., Hortin, G. L., Abdel-Hamid, M., Loffredo, C. A., and Goldman, R. (2007). Peak selection from MALDI-TOF mass spectra using ant colony optimization. *Bioinformatics*, **23**(5), 619–626.

Ressom, H. W., Varghese, R. S., Goldman, L., Loffredo, C. A., Abdel-Hamid, M., Kyselova, Z., Mechref, Y., Novotny, M., and Goldman, R. (2008). Analysis of MALDI-TOF mass spectrometry data for detection of glycan biomarkers. In *Proceedings of the Pacific Symposium on Biocomputing*, pages 216–227.

Reunanen, J. (2003). Overfitting in making comparisons between variable selection methods. *Journal of Machine Learning Research*, **3**, 1371–1382.

Rissanen, J. (1978). Modeling by shortest data description. *Automatica*, **14**, 465–471.

Rodríguez, J. D. and Lozano, J. A. (2008). Multi-objective learning of multi-dimensional Bayesian classifiers. In *Proceedings of the Eighth International Conference on Hybrid Intelligent Systems*, pages 501–506.

Rodríguez, J. D., Pérez, A., and Lozano, J. A. (2009). Sensitivity analysis of k-fold cross-validation in prediction error estimation. *Pattern Analysis and Machine Intelligence*. Accepted for publication.

Ross, J. S., Carlson, J. A., and Brock, G. (2007). miRNA: The new gene silencer. *American Journal of Clinical Pathology*, **128**(5), 830–836.

Rozzo, S. J., Allard, J. D., Choubey, D., Vyse, T. J., Izui, S., Peltz, G., and Kotzin, B. L. (2001). Evidence for an interferon-inducible gene, Ifi202, in the susceptibility to systemic lupus. *Immunity*, **15**, 435–443.

Ruvkun, G. (2001). Molecular biology. Glimpses of a tiny RNA world. *Science*, **294**(5543), 797–799.

Sáenz, A., Azpitarte, M., Armañanzas, R., Leturcq, F., Alzualde, A., Inza, I., García-Bragado, F., De la Herran, G., Corcuera, J., Cabello, A., Navarro, C., De la Torre, C., Gallardo, E., Illa, I., and López de Munain, A. (2008). Gene expression profiling in limb-girdle muscular dystrophy 2A. *PLoS ONE*, **3**(11), e3750.

Saeys, Y. (2004). *Feature Selection for Classification of Nucleic Acid Sequences*. Ph.D. thesis, Ghent University, Belgium.

Saeys, Y., Degroeve, S., Aeyels, D., Van de Peer, Y., and Rouzé, P. (2003). Fast feature selection using a simple estimation of distribution algorithm: A case study on splice site prediction. *Bioinformatics*, **19**(2), 179–188.

Saeys, Y., Degroeve, S., Aeyels, D., Rouzé, P., and Van de Peer, Y. (2004). Feature selection for splice site prediction: A new method using EDA-based feature ranking. *BMC Bioinformatics*, **5**(64).

Saeys, Y., Degroeve, S., and Van de Peer, Y. (2006). *Towards a New Evolutionary Computation: Advances in Estimation of Distribution Algorithms*, chapter Feature ranking using an EDA-based wrapper approach, pages 243–257. Springer.

Saeys, Y., Inza, I., and Larrañaga, P. (2007). A review of feature selection techniques in bioinformatics. *Bioinformatics*, **23**(19), 2507–2517.

Saeys, Y., Abeel, T., and Van de Peer, Y. (2008). Robust feature selection using ensemble feature selection techniques. In *Machine Learning and Knowledge Discovery in Databases*, Lecture Notes in Computer Science, pages 313–325.

Sahami, M. (1996). Learning limited dependence Bayesian classifiers. In *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining*, pages 335–338.

Sakakura, C., Hagiwara, A., Miyagawa, K., Nakashima, S., Yoshikawa, T., Kin, S., Nakase, Y., Ito, K., Yamagishi, H., Yazumi, S., Chiba, T., and Ito, Y. (2005). Frequent downregulation of the runt domain transcription factors RUNX1, RUNX3 and their cofactor CBFB in gastric cancer. *International Journal of Cancer*, **113**(2), 221–228.

Santafé, G. (2008). *Advances on Supervised and Unsupervised Learning of Bayesian Network Models. Application to Population Genetics*. Ph.D. thesis, University of the Basque Country.

Santana, R. (2005). Estimation of distribution algorithms with Kikuchi approximations. *Evolutionary Computation*, **13**(1), 67–97.

Santana, R. (2006). *Advances in Probabilistic Graphical Models for Optimization and Learning Applications in Protein Modelling*. Ph.D. thesis, University of the Basque Country.

Santana, R., Ponce de León, E., and Ochoa, A. (1999). The edge incident model. In *Proceedings of the Second Symposium on Artificial Intelligence (CIMAF-99)*, pages 352–359.

Santana, R., Ochoa, A., and Soto, M. R. (2001). The mixture of trees factorized distribution algorithm. In L. Spector, E. Goodman, A. Wu, W. Langdon, H. Voigt, M. Gen, S. Sen, M. Dorigo, S. Pezeshk, M. Garzon, and E. Burke, editors, *Proceedings of the Genetic and Evolutionary Computation Conference GECCO-2001*, pages 543–550.

Santana, R., Larrañaga, P., and Lozano, J. A. (2004). Protein folding in 2-dimensional lattices with estimation of distribution algorithms. In *Proceedings of the First International Symposium on Biological and Medical Data Analysis*, volume 3337 of *Lecture Notes in Computer Science*, pages 388–398.

Santana, R., Larrañaga, P., and Lozano, J. A. (2007a). The role of a priori information in the minimization of contact potentials by means of estimation of distribution algorithms. In E. Marchiori, J. H. Moore, and J. C. Rajapakse, editors, *Proceedings of the Fifth European Conference on Evolutionary Computation, Machine Learning and Data Mining in Bioinformatics*, volume 4447 of *Lecture Notes in Computer Science*, pages 247–257.

Santana, R., Larrañaga, P., and Lozano, J. A. (2007b). Side chain placement using estimation of distribution algorithms. *Artificial Intelligence in Medicine*, **39**(1), 49–63.

Santana, R., Larrañaga, P., and Lozano, J. A. (2008a). Combining variable neighborhood search and estimation of distribution algorithms in the protein side chain placement problem. *Journal of Heuristics*, **14**(5), 519–547.

Santana, R., Larrañaga, P., and Lozano, J. A. (2008b). Protein folding in simplified models with estimation of distribution algorithms. *IEEE Transactions on Evolutionary Computation*, **12**(4), 418–438.

Santana, R., Echegoyen, C., Mendiburu, A., Bielza, C., Lozano, J. A., Larrañaga, P., Armañanzas, R., and Shakya, S. K. (2009). MATEDA: A suite of EDA programs in Matlab. Technical Report EHU-KZAA-IK-2/09, University of the Basque Country.

Satoh, J., Misawa, T., Tabunoki, H., and Yamamura, T. (2008). Molecular network analysis of T-cell transcriptome suggests aberrant regulation of gene expression by NF-kappaB as a biomarker for relapse of multiple sclerosis. *Disease Markers*, **25**(1), 27–35.

Sauve, A. C. and Speed, T. P. (2004). Normalization, baseline correction and alignment of high-throughput mass spectrometry data. In *Proceedings of the Workshop on Genomic Siganl Processing and Statistics*.

Sawcer, S., Jones, H. B., Feakes, R., Gray, J., Smaldon, N., Chataway, J., Robertson, N., Clayton, D., Goodfellow, P. N., and Compston, A. (1996). A genome screen in multiple sclerosis reveals susceptibility loci on chromosome 6p21 and 17q22. *Nature Genetics*, **13**, 464–468.

Schena, M., editor (2000). *Microarray Biochip Technology*. Eaton Publishing Company.

Schena, M., Shalon, D., Davis, R. W., and Brown, P. O. (1995). Quantitative monitoring of gene expression patterns with a complementary DNA microarray. *Science*, **270**, 467–470.

Schena, M., Shalon, D., Heller, R., Chai, A., Brown, P. O., and Davis, R. W. (1996). Parallel human genome analysis: Microarray-based expression monitoring of 1000 genes. *Proceedings of the National Academy of Sciences of USA*, **93**(20), 10614–10619.

Schroeder, A., Mueller, O., Stocker, S., Salowsky, R., Leiber, M., Gassmann, M., Lightfoot, S., Menzel, W., Granzow, M., and Ragg, T. (2006). The RIN: An RNA integrity number for assigning integrity values to RNA measurements. *BMC Molecular Biology*, **7**(3).

Schwartz, R., Engel, I., Fallahi-Sichani, M., Petrie, H. T., and Murre, C. (2006). Gene expression patterns define novel roles for E47 in cell cycle progression, cytokine-mediated signaling, and T lineage development. *Proceedings of the National Academy of Sciences of the USA*, **103**(26), 9976–9981.

Scofield, R. H. (2003). Genetic knock out of 60 kD Ro (or SSA), a common lupus autoantigen, induces lupus. *Trends in Immunology*, **25**(1), 1–3.

Sebag, M. and Ducoulombier, A. (1998). Extending population-based incremental learning to continuous search spaces. In *Parallel Problem Solving from Nature - PPSN V*, pages 418–427.

Shai, R., Quismorio, F. P., Li, L., Kwon, O., Morrison, J., Wallace, D. J., Neuwelt, C. M., Brautbar, C., Gauderman, W. J., and Jacob, C. O. (1999). Genome-wide screen for systemic lupus erythematosus susceptibility genes in multiplex families. *Human Molecular Genetics*, **8**(4), 639–644.

Shakya, S. and McCall, J. (2007). Optimization by estimation of distribution with DEUM framework based on Markov random fields. *International Journal of Automation and Computing*, **4**(3), 262–272.

Shannon, C. E. (1948). A mathematical theory of communication. *Bell System Technical Journal*, **27**, 379–423, 623–656.

Sheng, Q., Moreau, Y., and De Moor, B. (2003). Biclustering microarray data by Gibbs sampling. *Bioinformatics*, **19**(Suppl. 2), 196–205.

Shin, H. and Markey, M. K. (2006). A machine learning perspective on the development of clinical decision support systems utilizing mass spectra of blood samples. *Journal of Biomedical Informatics*, **39**(2), 227–248.

Shin, H., Sheub, B., Joseph, M., and Markey, M. K. (2008). Guilt-by-association feature selection: Identifying biomarkers from proteomic profiles. *Journal of Biomedical Informatics*, **41**(1), 124–136.

Shmulevich, I., Gluhovsky, I., Hashimoto, R., Dougherty, E. R., and Zhang, W. (2003). Steady-state analysis of genetic regulatory networks modeled by probabilistic Boolean networks. *Comparative Functional Genomics*, **4**, 601–608.

Simon, J. L. (1997). *Resampling: The New Statistics*. Resampling Stats.

Slotta, D. J., Heath, L. S., Ramakrishnan, N., Helm, R., and Potts, M. (2003). Clustering mass spectrometry data using order statistics. *Proteomics*, **95**, 1687–1691.

Smith, C. A. B. (1947). Some examples of discrimination. *Annals of Eugenics*, **18**, 272.

Smith, J. L., Rangaraj, K., Simpson, R., Maclean, D. J., Nathanson, L. K., Stuart, K. A., Scott, S. P., Ramm, G. A., and de Jersey, J. (2004). Quantitative analysis of the expression of ACAT genes in human tissues by real-time PCR. *Journal of Lipid Research*, **45**, 686–696.

Soille, P. (1999). *Morphological Image Analysis*. Springer.

Somorjai, R. L., Dolenko, B., and Baumgartner, R. (2003). Class prediction and discovery using gene microarray and proteomics mass spectroscopy data: Curses, caveats, cautions. *Bioinformatics*, **19**(12), 1484–1491.

Spackman, K. A. (1989). Signal detection theory: Valuable tools for evaluating inductive learning. In *Proceedings of the Sixth International Workshop on Machine Learning*, pages 160–163.

Spiegelhalter, D. and Lauritzen, S. (1990). Sequential updating of conditional probabilities on directed graphical structures. *Networks*, **20**, 579–605.

Spirtes, P., Glymour, C., and Scheines, R. (1993). *Causation, Prediction, and Search*. Lecture Notes in Statistics 81, Springer-Verlag.

Srivastava, M., Rencic, A., Diglio, G., Santana, H., Bonitz, P., Watson, R., Ha, E., Anhalt, G. J., Provost, T. T., and Nousari, C. H. (2003). Drug-induced, Ro/SSA-positive cutaneous lupus erythematosus. *Archives of Dermatology*, **139**(1), 45–49.

Staal, F. J., Luis, T. C., and Tiemessen, M. M. (2008). WNT signalling in the immune system: WNT is spreading its wings. *Nature Reviews. Immunology*, **8**(8), 581–593.

Stamatos, N. M., Liang, F., Nan, X., Landry, K., Cross, A. S., Wang, L. X., and Pshezhetsky, A. V. (2005). Differential expression of endogenous sialidases of human monocytes during cellular differentiation into macrophages. *Journal of the Federation of European Biochemical Societies*, **272**(10), 2445–2456.

Statnikov, A., Aliferis, C. F., Tsamardinos, I., Hardin, D., and Levy, S. (2005). A comprehensive evaluation of multicategory classification methods for microarray gene expression cancer diagnosis. *Bioinformatics*, **21**(5), 631–643.

Stein, L. D. (2008). Bioinformatics: Alive and kicking. *Genome Biology*, **9**, 114.

Steipe, B. (1998). Protein design concepts. In P. V. R. Schleyer, N. L. Allinger, T. Clark, J. Gasteiger, P. A. Kollman, H. F. Schaefer III, and P. R. Schreiner, editors, *The Encyclopedia of Computational Chemistry*, pages 2168–2185. John Wiley & Sons.

Stone, M. (1974). Cross-validatory choice and assessment of statistical predictions (with discussion). *Journal of the Royal Statistical Society B*, **36**, 111–147.

Sturm, M., Bertsch, A., Gröpl, C., Hildebrandt, A., Hussong, R., Lange, E., Pfeifer, N., Schulz-Trieglaff, O., Zerck, A., Reinert, K., and Kohlbacher, O. (2008). OpenMS – An open-source software framework for mass spectrometry. *BMC Bioinformatics*, **9**(163).

Sullivan, K. E. (1999). The complex genetic basis of systemic lupus erythematosus (I). *Lupus Foundation of America Lupus News*, **19**(4).

Sullivan, K. E. (2000). The complex genetic basis of systemic lupus erythematosus (II). *Lupus Foundation of America Lupus News*, **20**(1).

Sundaresh, S., Hung, S., Hatfield, G. W., and Baldi, P. (2005). How noisy and replicable are DNA microarray data? *International Journal of Bioinformatics Research and Applications*, **1**(1), 31–50.

Swift, S., Tucker, A., Vinciotti, V., Martin, N., Orengo, C., Liu, X., and Kellam, P. (2004). Consensus clustering and functional interpretation of gene-expression data. *Genome Biology*, **5**(11), R94.1–R94.16.

Takahashi, E., Funato, N., Higashihori, N., Hata, Y., Gridley, T., and Nakamura, M. (2004). Snail regulates p21(WAF/CIP1) expression in cooperation with E2A and Twist. *Biochemical and Biophysical Research Communications*, **325**(4), 1136–1144.

Tan, E. M., Cohen, A. S., Fries, J. F., Masi, A. T., McShane, D. J., Rothfield, N. F., Schaller, J. G., Talal, N., and Winchester, R. J. (1982). The 1982 revised criteria for the classification of systemic lupus erythematosus. *Arthritis & Rheumatism*, **25**, 1271–1277.

Tax, D. M. and Duin, R. P. (2002). Uniform object generation for optimizing one-class classifiers. *Journal of Machine Learning Research*, **2**(2), 155–173.

Tibshirani, R., Hastie, T., Narasimhan, B., Soltys, S., Shi, G., Koong, A., and Le, Q.-T. (2004). Sample classification from protein mass spectrometry, by peak probability contrasts. *Bioinformatics*, **20**(17), 3034–3044.

Todd, B. S. and Stamper, R. (1994). The relative accuracy of a variety of medical diagnostic programs. *Methods of Information in Medicine*, **33**, 402–416.

Toussaint, G. T. (1974). Bibliography on estimation of misclassification. *IEEE Transactions on Information Theory*, **20**, 472–479.

Troyanskaya, O., Cantor, M., Sherlock, G., Brown, P., Hastie, T., Tibshirani, R., Botstein, D., and Altman, R. B. (2001). Missing value estimation methods for DNA microarrays. *Bioinformatics*, **17**(6), 520–525.

Tsao, B. P. (2003). The genetics of human systemic lupus erythematosus. *Trends in Immunology*, **24**(11), 595–602.

Tuzhilin, A. and Adomavicius, G. (2002). Handling very large numbers of association rules in the analysis of microarray data. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 396–404.

Valkenborg, D., Van Sanden, S., Lin, D., Kasim, A., Zhu, Q., Haldermans, P., Jansen, I., Shkedy, Z., and Burzykowski, T. (2008). A cross-validation study to select a classification procedure for clinical diagnosis based on proteomic mass spectrometry. *Statistical Applications in Genetics and Molecular Biology*, **7**(2), 12.

Vallejo-Illarramendi, A., Domercq, M., Pérez-Cerda, F., Ravid, R., and Matute, C. (2006). Increased expression and function of glutamate transporters in multiple sclerosis. *Neurobiology of Disease*, **21**(1), 154–164.

Van der Gaag, L. C. and de Waal, P. R. (2006). Muti-dimensional Bayesian network classifiers. In *Proceedings of the Third European Workshop in Probabilistic Graphical Models*, pages 107–114.

van Rijsbergen, C. J. (1979). *Information Retrieval*. Butterworth.

van't Veer, L. J., Dai, H., Van de Vijver, M. J., He, Y. D., Hart, A. A., Mao, M., Peterse, H. L., Van der Kooy, K., Marton, M. J., Witteveen, A. T., Schreiber, G. J., Kerkhoven, R. M., Roberts, C., Linsley, P. S., Bernards, R., and Friend, S. H. (2002). Gene expression profiling predicts clinical outcome of breast cancer. *Nature*, **415**(530–536).

Walker, P. R., Smith, B., Liu, Q. Y., Famili, A. F., Valdés, J. J., Liu, Z., and Lach, B. (2004). Data mining of gene expression changes in Alzheimer brain. *Artificial Intelligence in Medicine*, **31**(2), 137–154.

Wallace, D. J. and Hahn, B. H. (2002). *Dubois' Lupus Erythematosus*. Lippincott Williams & Wilkins.

Wang, J., Cheung, L. W., and Delabie, J. (2005a). New probabilistic graphical models for genetic regulatory networks studies. *Journal of Biomedical Informatics*, **38**, 443–455.

Wang, L., Chu, F., and Xie, W. (2007). Accurate cancer classification using expressions of very few genes. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, **4**(1), 40–53.

Wang, Y., Tetko, I. V., Hall, M. A., Frank, E., Facius, A., Mayer, K. F., and Mewes, H. W. (2005b). Gene selection from microarray data for cancer classification – a machine learning approach. *Computational Biology and Chemistry*, **29**, 37–46.

Weber, M. J. (2005). New human and mouse microRNA genes found by homology search. *The FEBS Journal*, **272**(1), 59–73.

Weinberg, R. A. (1994). Oncogenes and tumor suppressor genes. *CA: A Cancer Journal for Clinicians*, **44**, 160–170.

Weiss, S. M. and Kulikowski, C. A. (1991). *Computer Systems that Learn*. Morgan Kaufmann.

Wherry, R. J. (1931). A new formula for predicting the shrinkage of the multiple correlation coefficient. *Annals of Mathematical Statistics*, **2**, 440–457.

Whittaker, J. (1991). *Graphical Models in Applied Multivariate Statistics*. Series in Probability and Mathematical Statistics. Wiley Series in Probability and Mathematical Statistics.

Wichmann, I., Montes-Cano, M. A., Respaldiza, N., Alvarez, A., Walter, K., Franco, E., Sanchez-Román, J., and Nuñez-Roldán, A. (2003). Clinical significance of anti-multiple nuclear dots/Sp100 autoantibodies. *Scandinavian Journal of Gastroenterology*, **38**(9), 996–999.

Wilcoxon, F. (1945). Individual comparisons by ranking methods. *Biometrics*, **1**, 80–83.

Willan, R. (1808). *On Cutaneous Diseases*. J. Johnson, London.

Wit, E. and McClure, J. (2004). *Statistics for Microarrays. Design, Analysis and Inference*. John Wiley & Sons.

Witten, I. H. and Frank, E. (2005). *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, 2nd edition.

Wolber, P. K., Collins, P. J., Lucas, A. B., De Witte, A., and Shannon, K. W. (2006). The agilent in situ-synthesized microarray platform. *Methods in Enzymology*, **410**, 28–57.

Xing, E. P., Jordan, M. I., and Karp, R. M. (2001). Feature selection for high-dimensional genomic microarray data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, pages 601–608.

Yang, K., Cai, Z., Li, J., and Lin, G. (2006). A stable gene selection in microarray data analysis. *BMC Bioinformatics*, **7**(1), 228.

Yang, Y. H. and Speed, T. (2002). Design issues for cDNA microarray experiments. *Nature Reviews Genetics*, **3**, 579–588.

Yang, Y. H., Dudoit, S., Luu, P., Lin, D. M., Peng, V., Ngai, J., and Speed, T. P. (2002). Normalization for cDNA microarray data: A robust composite method addressing single and multiple slide systematic variation. *Nucleic Acids Research*, **30**(4), e15.

Yu, L. and Liu, H. (2003). Feature selection for high-dimensional data: A fast correlation-based filter solution. In *Proceedings of the Twentieth International Conference on Machine Leaning*, pages 856–863.

Yu, L. and Liu, H. (2004). Redundancy based feature selection for microarray data. In *Proceedings of the 2004 ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 737–742.

Zahurak, M., Parmigiani, G., Yu, W., Scharpf, R. B., Berman, D., Schaeffer, E., Shabbeer, S., and Cope1, L. (2007). Pre-processing agilent microarray data. *BMC Bioinformatics*, **8**, 142.

Zander, K., Sherman, M., Tessmer, U., Bruns, K., Wray, V., Prechtel, A. T., Schubert, E., Henklein, P., Luban, J., Neidleman, J., Greene, W. C., and Schubert, U. (2003). Cyclophilin A interacts with HIV-1 Vpr and is required for its functional expression. *Journal of Biological Chemistry*, **278**(44), 43202–43213.

Zeng, M., Li, J., and Peng, Z. (2006). The design of top-hat morphological filter and application to infrared target detection. *Infrared Physics & Technology*, **48**(1), 67–76.

Zhang, G., Fukao, T., Sakurai, S., Yamada, K., Michael Gibson, K., and Kondo, N. (2006). Identification of Alu-mediated, large deletion-spanning exons 2-4 in a patient with mitochondrial acetoacetyl-CoA thiolase deficiency. *Molecular Genetics and Metabolism*, **89**(3), 222–226.

Zhang, S. and Gant, T. W. (2005). Effect of pooling samples on the efficiency of comparative studies using microarrays. *Bioinformatics*, **21**(24), 4378–4383.

Zhao, Y., Ransom, J. F., Li, A., Vedantham, V., von Drehle, M., Muth, A. N., Tsuchihashi, T., McManus, M. T., Schwartz, R. J., and Srivastava, D. (2007). Dysregulation of cardiogenesis, cardiac conduction, and cell cycle in mice lacking miRNA-1-2. *Cell*, **129**(2), 303–317.

Zhou, X., Wang, X., Pal, R., Ivanov, I., Bittner, M., and Dougherty, E. R. (2004). A Bayesian connectivity-based approach to constructing probabilistic gene regulatory networks. *Bioinformatics*, **20**(17), 2918–2927.

Zhu, X. (2005). Semi-supervised learning literature survey. Technical Report 1530, Computer Sciences, University of Wisconsin-Madison.

Zipitria, I., Larrañaga, P., Armañanzas, R., Arruarte, A., Elorriaga, J. A., and Díaz de Ilarraza, A. (2008). What is behind a summary evaluation decision? *Behavior Research Methods*, **40**(2), 597–612.

Zou, T., Selaru, F. M., Xu, Y., Shustova, V., Yin, J., Mori, Y., Shibata, D., Sato, F., Wang, S., Olaru, A., Deacu, E., Liu, T. C., Abraham, J. M., and Meltzer, S. J. (2002). Application of cDNA microarrays to generate a molecular taxonomy capable of distinguishing between colon cancer and normal colon. *Oncogene*, **21**, 4855–4862.