Modèles d'intégration de la connaissance pour la fouille des données d'expression des gènes

Ricardo Martinez

THESE

présentée à

UNIVERSITE DE NICE - SOPHIA ANTIPOLIS

pour obtenir le titre de

DOCTEUR EN SCIENCES

Ecole Doctorale Sciences et Technologies de l'Information et de la Communication

Specialité:

Bioinformatique & Modèlisation du Vivant

presenté

par

Ricardo Martinez

Titre

Knowledge Integration Models for Mining Gene Expression Data

Directeur de thèse: Martine Collard

Sophia Antipolis, France

[Juillet] [2007]

Abstract

In the framework of this thesis we develop new data mining models for knowledge discovery with gene expression profiles. Data mining is the science of automatically extracting knowledge hidden in large data sets. Gene expression technologies are powerful methods for studying biological processes through a transcriptional point of view. These technologies have produced vast amounts of data by measuring simultaneously the expression levels of thousands of genes under different biological conditions.

One of the great potentials of this technology is that the data generated contain hidden information about the biological processes that govern cell behavior. A main challenge in gene expression analysis is the interpretation of results via combination of gene expression analysis with associated sources of biological information. This process may also be referred to as *integration of biological knowledge with gene expression data*. Sources of biological information are for instance molecular databases, ontologies, taxonomies, semantic networks or bibliographic databases. Our work focusses on the issue of integrating existing biological knowledge, particularly within non-supervised (or clustering) and supervised learning algorithms applied to the data.

In this thesis, we first present an original point of view for the state of the art on methods developed for interpreting gene expression results through corresponding gene annotations. Then, we tackle the non-supervised learning issue of class discovery among gene expression profiles, and we propose two specific approaches on this subject: CGGA (Co-expressed Gene Groups Analysis) and GENMINER (Gene-integrated analysis using association rules mining). CGGA is a knowledge-based approach which automatically integrates gene expression profiles and gene annotations obtained from genome-wide information databases such as Gene Ontology. GENMINER is a co-clustering and bi-clustering approach which automatically integrates at once gene annotations and gene expression profiles to discover intrinsic associations between these two heterogeneous sources of information.

Finally, we focus on the supervised learning issue of class prediction, and we propose GENETREE (GENE-integrated analysis for biological sample prediction using decision TREEs), an approach which takes advantage of the well known decision tree algorithm C5.0 and adapts its entropy splitting principle with several ontology-based criteria.

Contents

List of Figures

List of Tables

Introduction

In the framework of this thesis we develop novel data mining models for knowledge discovery with gene expression technologies. We understand gene expression technologies as the ones that intend to measure gene expression such as microarray and Serial Analysis of Gene Expression (SAGE). A gene is expressed when, through the transcription process, its DNA coding is transferred to an RNA molecule. Transcription is the process of synthesizing RNA using genes as template. Indeed, we have two up-to-date, dynamic and complex topics: data mining and gene expression technologies, and we need to find the jointures between them.

Data mining is the science of automatically extracting knowledge and information from large data sets or databases. This science can be divided in two main groups: descriptive tasks such as unsupervised learning, clustering, etc., and predictive ones such as supervised learning, classification, discriminant analysis etc.

Gene expression technologies are powerful methods for studying biological processes through a transcriptional point of view. Since many years these technologies have produced vast amounts of data by measuring simultaneously the expression levels of thousands of genes under tens of different biological conditions. One of the great promises of this technology is that the data generated contain hidden but potentially rich information about the biological processes that govern cell behavior.

One of the main goals of gene expression technology is to discover hidden information within the biological experiments to generate biological knowledge. But, what kind of information is commonly searched within a gene expression biological experience? We can mention three main issues: identifying the co-expressed genes (genes with common expression profile), finding the coherent gene expression patterns (collective trend in expression levels of co-expressed genes groups), and class prediction (predicting the correspondent class for different kinds of tumors, diseases, responses etc., via their gene expression patterns).

The first two issues are class discovery issues, commonly solved by descriptive or unsupervised learning models. In contrast, the third one, class prediction, is often recognized as a typical supervised learning problem. In descriptive methods the classes are unknown and need to be discovered from the data. In predictive methods, on the other hand, the classes are predefined and the task is to understand the basis of their classification for predicting the class of future observations.

Nowadays, one of the main challenges in gene expression technologies is the interpretation of results via integration of gene expression measures with associated sources of biological information. We can divide these sources of information in six main groups:

1. Minimal microarray information (genes and biological conditions characteristics).
2. Molecular databases (GenBank, Embl, Unigene, etc.).
3. Semantic sources as thesaurus, ontologies, taxonomies or semantic networks (UMLS, GO, taxonomy, etc.).

4. Gene expression databases (GEO, Arrayexpress, Microarray database, etc.).
5. Bibliographic databases (Medline, Biosis, etc.).
6. Gene/protein related specific sources (ONIM, KEGG, etc.).

The analysis of gene expression data consists of five steps: data generation, statistical data treatment, analysis of differentially expressed genes, classification of the genes, and knowledge discovery via data interpretation. This field, as well as the whole bioinformatics field, have several inherent questions and problems to solve:

- The field is an ever-increasingly volume of scattered and disordered genomic data, information and knowledge.
- Existing sources of biological information must be well-structured, up-to-date and without ambiguities.
- There is no consensus concerning essential information issues as gene name, data structure, results description, data availability, knowledge acquisition etc.
- Existing tools and computational technical progress have to be enhanced for manipulating high dimensionality of gene expression data (thousands of genes), low number of sample experiments (tens of biological conditions) and tons of scattered biological information and knowledge.
- In order to minimize loss of information when analyzing gene expression data, from initial noisy data and passing through the five-step analysis until the discovery of knowledge, gene expression technologies and analysis tools have to be improved.
- Analysis tools for each specific gene expression technology have to be built specifically. Since approaches to tackle each of the five steps are multiple and heterogeneous, a consensus is needed on specific techniques for each step.

In this thesis, we focus on the last analysis step devoted to data interpretation. The issue is to integrate two elements, the numeric element represented by the gene expression measures and the biological knowledge element represented by gene annotations (pieces of biological information related to the gene as relational, syntactical, functional etc.) issued from different sources of biological information. In other words, the issue is the interpretation of gene expression results via integration of gene expression profiles with corresponding biological gene annotations extracted from biological knowledge databases. We have divided our task in two big data mining goals:

1. Highlighting the main co-expressed and co-annotated gene groups (co-annotated gene groups are groups sharing the same annotation) using at least one source of biological knowledge. This descriptive issue is generally refered to as class discovery.
2. Building a predictive model for disease-type classification using at once gene expression measures and at least one source of biological knowledge. This issue is commonly known as class prediction.

Currently, these two main goals are achieved manually by domain experts who combine their own knowledge with biological sources of information. Due to the huge complexity of biological processes, experts need automatic or semi-automatic tools to help them.

History and Previous Work

Concerning the class discovery problem

At the beginning of gene expression technology, researches focused on manipulating only gene expression measures for identifying groups of co-expressed genes. There were reported a variety of unsupervised learning approaches which identify groups of co-expressed genes (class discovery problem) based only on gene expression measures as the only source of information to study, as [69, 90, 26, 107, 298, 303]. A common characteristic of these purely numerical approaches is that they determine gene groups (or clusters) of potential interest. However, they leave to the expert the task of discovering and interpreting biological similarities hidden within these groups. These methods are useful, because they guide the analysis of the co-expressed gene groups. Nevertheless, their results are often incomplete, because they do not include biological considerations based on prior biological knowledge.

Currently, one of the major challenges in bioinformatics researches is interpreting gene expression data via the automatic integration of biological knowledge from different sources of information with numerical gene expression profiles [13]. The biological knowledge comes from several databases, ontologies and scientific publications mainly. It provides textual indications on genes which are referred to as annotations. The interpretation step may be defined as the result of the integration of gene expression profiles and corresponding gene annotations. In this context, integration means matching co-expressed and co-annotated genes.

Nowadays, most interpretation approaches are based on gene expression measures which are often noisy data, thus the results can be severely biased. In contrast, the few existing knowledge-based interpretation approaches, based on biological information, deal with the problem of currently scattered, badly structured and incomplete biological sources of information. So, they often lead to insufficient interpretation results.

Co-clustering approaches represent the best compromise in terms of integration between expression profiles and biological knowledge. Nevertheless, they have to deal with the algorithmic issue of integrating these two elements at once. Thus, they often give more weight to one of these two elements, carrying as well their intrinsic defaults.

Concerning the class prediction problem

The prediction problem on gene expression technologies has been targeted mainly in medical applications for predicting the state of an organ (e.g. cancer vs normal), special types of a disease (e.g. young diabete, diabete, normal) and the effectiveness of a medicament.

Since the beginning of gene expression technologies, a variety of supervised learning algorithms have been used to solve the prediction problem for disease-type applications. These algorithms take into account only gene expression profiles without integrating any gene annotation in the algorithm itself. Among the most remarquable methods we can list:

- Linear discriminant analysis (LDA) like in Dudoit [101] for predicting cancer tumors and in Hakak [137] for predicting schizofrenia types.
- K-nearest neighbor (KNN) method like in Pomeory [242] to predict embryonal tumors.
- Support Vector Machines (SVM) like in Ramaswamy [250] for classifying tumors and Furey [119] in predicting organ classes
- Weighted voting techniques like in Golub [133] to predict leukemia classes.
- Decision trees like in Zhang [335] for tumor prediction and Ramanathan [249] for disease-type prediction.

These methods are useful for predicting the studied class or disease-type at a certain degree of effectiveness. However, the actual use of predictive algorithms in gene expression technology field present several weaknesses :

- Error Estimation procedures should be applied externally to the gene selection process, and not internally as it is commonly done [102]. Thus, estimators are biased.
- Dimensionality of gene expression data where the number of objects or sample experiments is very low (tens of biological conditions) and the number of attributes is extremly high (thousands of genes). Low number of samples overfits the solutions.
- Lack of optimisation techniques on data learning parameters. Thus, the classifier cannot be robust enough to treat similar gene expression data sets [33].
- Lack of biological knowledge as an inherent part of the classifier building procedure.

The use of supervised algorithms for solving prediction problems in gene expression technologies is a relatively new field compared to other domains. It is necessary to include the best avalaible tools of maching learning methods in gene expression technology to close the gap between machine learning field and bioinformatics. The wide research problem of biological knowledge integration (as the available gene annotations) in any supervised algorithm remains completely open.

Contributions

In this thesis we strongly encourage the use of the existing biological knowledge within non-supervised and supervised algorithms in order to enhance the interpretation of the results of gene expression technology.

We propose a new framework, interpreting gene expression throughput as the result of the integration of gene expression profiles and corresponding gene annotations. As a basis of our contribution we start by presenting an original point of view on existent interpretation approaches. The classification we give, consists in three axes: *knowledge-based* axis, *expression-based* axis, and *co-clustering* axis. Our classification emphasizes the weight of the integration process scheduling on the final interpretation results. A survey, a description of each remarkable approach, a comparative among them, and a full discussion are presented in this document (see chapter 5).

Concerning the class discovery problem we have developed two approaches: CGGA (Co-expressed Gene Groups Analysis) and GENMINER (Gene-integrated analysis using association rules discovery) described below.

CGGA is a knowledge-based approach which automatically integrates the results of gene expression technology, i.e. gene expression profiles, and the biological annotations of the genes obtained by the genome-wide information databases such as Gene Ontology. By applying CGGA to well-known microarray experiments, we identify the main functionally enriched and co-expressed gene groups, and we show that this approach enhances and optimizes the interpretation of microarray experiments. Conception, implementation[1], experimentation and validation are presented in this document (see chapter 6).

GENMINER is a co-clustering and biclustering (biclustering means finding subsets of genes for a subset of biological conditions or viceversa) approach which automatically integrates at once gene annotations and gene expression profiles to discover intrinsic associations between both data sources based on frequent patterns. Our algorithm is an adaptation of traditional association rules mining techniques, that takes advantage of CLOSE [229] algorithm to generate low support, high confidence and non redundant rules in an efficient way.Validation was done using famous gene expression data sets in which genes were annotated by several sources of information. Automatically extracted associations reveal significant groups, meaning important biological relationships between gene attributes and patterns. Many of these relationships are supported by recently reported work. Conception, implementation, experimentation and validation were done (see chapter 7).

Concerning the class prediction problem we have developed GENETREE (GENE-integrated analysis for biological sample prediction using decision TREe algorithms).

GENETREE is decision tree based algorithm which automatically integrates the information contained on gene expression profiles with biological knowledge obtained by genome-wide sources of information. This algorithm takes advantage of the well known decision tree algorithms ID3, C4.5 and C5.0 proposed by Quinlan [247] and it extends the

[1] CGGA program is available at http://www.i3s.unice.fr/~rmartine/CGGA

entropy splitting criterion to more complex one which takes into account several criteria obtained from different sources of gene annotations as ontologies, molecular databases and gene/protein related sources of information. Main characteristics of this algorithm are presented in this document (see chapter 8).

In order to answer to the discretization question, an essential requirement in many supervised algorithms, we have developed a novel algorithm named **NORDI** (normal discretization), specially fitted to gene expression technologies. NORDI is based on statistical detection of outliers and the continuous application of normality tests for transforming the initial distribution "almost normal" to a "more normal" one. The term "almost" means that the sample S_j can be normally distributed without the presence of outliers. Conception, implementation[2], and validation are presented in this document (see chapter 8).

In order to answer to relevant biological conditions selection, an essential requirement for data pretreatment concerning prediction problem, we have developed a novel algorithm for sample outliers detection, specially fitted to gene expression technologies. Our algorithm is based on data mining detection of sample outliers using alternatively two techniques Principal Componenet Analysis (PCA) and Unweighted Pair Group Method with Arithmetic mean (UPGMA) hierarchical algorithm. Conception, implementation and validation are presented in this document (see chapter 4).

Manuscript Layout

This manuscript is divided in three parts. *Part I* (chapters 1 to 3) is devoted to the basic genomic, transcriptomic and molecular biology concepts involved in gene expression technologies.Then, it describes the current technologies, Microarray and SAGE, for measuring the gene expression. Next, it explains the complete multi-step analysis procedure to handle with gene expression data. Later on, it gives a fully overview of the biological information sources truly available to deal with the interpretation step. *Part II* contains two chapters: a practical one (chapter 4) and a state of the art or theoretical one (chapter 5). The first presents a complete gene expression data analysis for two data sets issued from different gene expression technologies: SAGE and Microarray. In the second we develop our interpretation framework for the knowledge integration step, presenting a complete overview of remarkable integration approaches. Finally, *Part III* (chapter 6 to 9) fully explains the knowledge integration models we propose: CGGA, GENMINER and GENETREE respectively. Each one of the models is developed in detail: method principles, algorithm implementation, applications, discussion and an outlook. In the following, a detailed summary of each chapter is given.

[2] NORDI program is available by request, and soon it will be available in bioconductor project: http://www.bioconductor.org/ .

Part I: Molecular Biology and Gene Expression Technologies Basics

Chapter 1 Principles of Life in Molecular Biology and Gene Expression Technologies

This chapter presents the basic concepts of molecular biology and genomics: genes, proteins, nucleic acids, etc. We focus our interest in the information flow from gene to protein, emphasizing the transcription of the information contained within the DNA, best known as gene expression. Then, we describe the most important gene expression measure technologies: Microarray (spotted cDNA chip, spotted oligos chip, in situ oligos-chip, etc.) and SAGE. Finally we explore several important applications of these technologies.

Chapter 2 Gene Expression Data Analysis Procedure

This chapter explains the five steps for analyzing microarray data technology: data generation, statistical data treatment, analysis of differentially expressed genes, classification of genes and knowledge discovery via data interpretation. Each section describes the analysis step and gives an insight into some methods to deal with it.

Chapter 3 Biological Sources of Information

This chapter gives an overview of the different biological information sources available for the microarray data analysis. The chapter starts with a brief description of the minimal biological information obtained about a microarray experiment. Then it explores the different sources of biological information: molecular sources (EMBL, GenBank,etc.), semantic sources (UMLS, GO, Taxonomy, etc.), bibliographic databases (Medline, Biosis, OMIN, etc.), gene expression databases (GEO, Arrayexpress, Microarray, etc.), and gene-related or protein-related sources (KEGG, GeneCards, etc.).

Part II: Introductory Works and Knowledge Discovery Interpretation Approaches

Chapter 4 First Application Works: SAGE and Microarray Data Analysis

In this chapter we have fully analyzed the gene expression throughput data issued from two different technologies: SAGE, and Microarray. For each technology, we follow up the whole data analysis procedure. Each step is detailed with a selection of the currently available tools and methods. Here, we explains our algorithm OutSample (sample outliers detection) specially conceived for gene expression technologies.

Chapter 5 Biological Knowledge Interpretation Approaches

This chapter represents the framework of the biological knowledge integration models we propose. In this chapter we develop three different knowledge integration axes: prior or

knowledge-based axis, standard or expression-based axis and co-clustering axis (presented in Chapter 2). The chapter starts with a brief discussion of the purpose and usefulness of this classification. Then, every section gives an insight into the basics and remarquable approaches. At the end of each section we summarize with a comparison of different approaches of each axis. Finally, we finish by a discussion in analyzing three main facets: gene annotations, gene expression profiles and gene selection. It ends by a general conclusion overall interpretation axes.

Part III: Data Mining Models for Knowledge Discovery via Biological Interpretation

Chapter 6 CGGA: Co-expressed Gene Groups Analysis

This chapter develops entirely the Co-expressed GeneGroup Analysis integration method we propose. It starts by providing a brief explanation of this knowledge-based approach. Next, it explains validation data sets and used methods. Then, it gives a full explanation of CGGA algorithm. This chapter continues with a complete analysis of real microarray data, showing the effectiveness of the method. It ends by a discussion of the advantages and drawbacks of CGGA and it gives an outlook for further research

Chapter 7 GENMINER: Gene-integrated analysis using association rules discovery

In this chapter we completely develop the GENMINER. The chapter starts with a global view of association rules basics and methods. It continues giving a general survey of association rules applications in bioinformatics.Next, it describes the GENMINER algorithm foundations and implementation aspects. Afterwards, the viability of this method is tested by analyzing two cDNA spot data sets. Discussion and an outlook for future research are given in the last two sections.

Chapter 8 GENETREE: GENE-integrated analysis for biological sample prediction using decision TREes algorithm

This chapter focuses on the class prediction problem issued in gene expression technologies. After presenting a global view of supervised methods, it gives an overview of the process for building prediction models in gene expression technologies. Since discretization is a key problem for predictive variables, it continues with an assessement of the discretization approaches used in this field, emphasizing in our novel discretization algorithm: NORDI. Then, it describes the decision trees basics and it explain the principles of GENETREE algorithm. It ends with a brief discussion and it gives the future expectations of this model.

Glossary of Terms

The reference for the glossary terms are marked with the symbol * in the text.

Alignment is the process of lining up two or more sequences to achieve maximal levels of identity (and conservation, in the case of amino acid sequences) for the purpose of assessing the degree of similarity and the possibility of homology.

Algorithm is a fixed procedure embodied in a computer program.

Apoptosis: A genetically directed process of cell self-destruction that is marked by the fragmentation of nuclear DNA, called also programmed cell death.

Bagging in machine learning is also called aggregating bootstrap and is an algorithm to improve classification models in terms of stability, classification accuracy, reducing variance and avoiding overfitting. Given a standard training set D of size N, we generate L new training sets D_i also of size N' (N'$< N$) by sampling examples uniformly from D, and with replacement. The L models are fitted using the above L bootstrap samples and combined by voting (in case of classification).

Base pairs are two nitrogenous bases (adenine and thymine or guanine and cytosine) or strands of DNA which are held together in the shape of a double helix by weak hydrogen bonds. The base pairs number is often used as a measure of length of an organism's genome.

Bioinformatics is the merger of biotechnology and information technology with the goal of revealing new insights and principles in biology.

Bit Score is the value S' is derived from the raw alignment score S in which the statistical properties of the scoring system used have been taken into account. Because bit scores have been normalized with respect to the scoring system, they can be used to compare alignment scores from different searches.

BLAST (Basic Local Alignment Search Tool): A sequence comparison algorithm optimized for speed used to search sequence databases for optimal local alignments to a query. For additional details, see one of the BLAST tutorials (Query or BLAST).

Boosting is a machine learning which occurs in stages, by incrementally adding to the current learned function. At every stage, a weak learner (i.e., one that has an accuracy only slightly greater than chance) is trained with the data. The output of the weak learner is then added to the learned function, with some strength (proportional to how accurate the weak learner is). Then, the data is reweighted: examples that the current learned function gets wrong are "boosted" in importance, so that future weak learners will attempt to fix the errors.

Bootstrapping: In statistics, is a modern, computer-intensive, general purpose approach to statistical inference, falling within a broader class of resampling methods. It is used for estimating the sampling distribution of an estimator by resampling with replacement from the original sample, most often with the purpose of deriving estimates of standard errors and confidence intervals of a population parameter like a mean, median, proportion, odds ratio, correlation coefficient or regression coefficient.

cDNA or Complementary DNA: DNA synthesized from a mature mRNA template.

cDNA are often used as probes of microarray and in cloning.

Co-annotated Gene Group is a group of genes with the same annotation.

Co-expressed Gene Group is a group of genes with a common expression profile.

Co-expressed Genes are the genes that exhibit a common expression profile.

Coherent Gene Expression Patterns are patterns that characterize the collective trend of the expression levels of a group of co-expressed genes.

Combination: In mathematics is a combination of r elements of a set is any subset of r elements from the set without regard to order. If the set has n elements, then the number of combinations of r elements is denoted by $C(n, r)$.

Concatemer: A DNA segment composed of series of sequences linked end to end.

Concept: An abstract and general idea of something inferred from specific instances or occurrences.

Contingency Tables are tables used in statistics to record and analyse the relationship between two or more variables, most usually categorical variables.

Data Modeling is the process of structuring and organizing data. The data structures are often implemented in a database management system.

DNA Replication is the process of copying a double-stranded DNA strand. Since DNA strands are antiparallel and complementary, each strand can serve as a template for the reproduction of the opposite strand. The template strand is preserved as a whole piece and the new strand is assembled from nucleotide triphosphates.

Entropy: In information theory, entropy is a measure of the uncertainty associated with a random variable introducted by Shannon. It can be interpreted as the average shortest message length, in bits, that can be sent to communicate the true value of the random variable to a recipient.

Expressed Gene is a gene which coding is transferred to the RNA molecule during the transcription process.

Expressed Sequenced Tags (EST) is a short sub-sequence of a transcribed spliced nucleotide sequence (either protein-coding or not). They are intended as a way to identify gene transcripts, and are instrumental in gene discovery and gene sequence determination.

Fisher's Exact Test: A statistical significance test used in the analysis of categorical data. It's equivalent to the hypergeometric test. More details in [113].

Fuzzy Set Theory: Fuzzy sets are an extension of classical set theory and are used in fuzzy logic. In classical set theory the membership of elements in relation to a set is assessed in binary terms according to a given condition, i.e. an element either belongs or not to the set. By contrast, fuzzy set theory permits the gradual assessment of the membership of elements in relation to a set, it is usually described with a membership function $\mu \rightarrow [0, 1]$.

Gene Annotation is a piece of biological information related to the gene that can be relational, syntactical, functional, etc.

Gene Expression Profile is the outline of the sorted (in time, by experimental condition etc.) expression measures of a gene.

Gene Ontology Level is the position of a gene annotation or GO term in the GO hierarchie. For example a level of 4 means that it has three high level ancestors.

Gene Product is the biochemical material, either RNA or protein, resulting from expres-

sion of a gene.

Genetic Map (also called linkage map): A chromosome map of a species or experimental population that shows the position of its known genes and/or markers relative to each other, rather than as specific physical points on each chromosome. A genetic map is a map based on the frequencies of recombination between markers during crossover of homologous chromosomes.

Gene Regulatory Network: (also called a GRN or genetic regulatory network) is a collection of DNA segments in a cell which interact with each other (indirectly through their RNA and protein expression products) and with other substances in the cell, thereby governing the rates at which genes in the network are transcribed into mRNA.

Ground Truth: The actual facts of a situation, without errors introduced by sensors or human perception and judgment.

Heteroduplexes are double-chained nucleic acid molecules (DNA-DNA or DNA-RNA) which contain regions of nucleotide mismatches (non-complementary). They can be produced either by hybridization or by mutations.

Hybridization is the process of combining complementary, single-chained nucleic acids into a single molecule.

Hybridization Probe is a short piece of DNA (on the order of 100-500 bases) that is denatured (by heating) into single chains and then radioactively labeled, usually with phosphorus (32P or 33P).

In Situ Synthesis takes place via a covalent reaction between the 5' hydroxyl group of the sugar of the last nucleotide to be attached to the chip and the phosphate group of the next nucleotide. Each nucelotide added to the oligonucleotide probe anchored to the glass chip has a protective group at its 5' position. This protective group is then converted to an hydroxyl group, using either acid or light [294].

Junk DNA is a region of DNA that usually consists of a repeating DNA sequence, does not code for protein, and has no known function.

Messenger Ribonucleic Acid or mRNA is a molecule of RNA encoding a chemical "blueprint" for a protein product. mRNA is transcribed from a DNA template, and carries coding information to the sites of protein synthesis: the ribosomes.

Metabolic Pathway: In biochemistry, it consists of a series of chemical reactions occurring within a cell, catalyzed by enzymes, resulting in either the formation of a metabolic product to be used or stored by the cell, or the initiation of another metabolic pathway (then called a flux generating step). Many pathways are elaborate, and involve a step by step modification of the initial substance to shape it into the product with the exact chemical structure desired.

Neural Networks: In machine learning, parallel distributed processing network, consists of interconnected processing elements called nodes or neurons that work together to produce an output function. The output of a neural network relies on the cooperation of the individual neurons within the network to operate. Processing of information by neural networks is characteristically done in parallel rather than in series (or sequentially) as in earlier binary computers.

NP-complete Problems: In Informatics, NP-complete problems are the most difficult problems in NP ("non-deterministic polynomial time ") in the sense that they are the smallest subclass of NP that could conceivably remain outside of P, the class of deter-

ministic polynomial-time problems. The reason is that a deterministic, polynomial-time solution to any NP-complete problem would also be a solution to every other problem in NP. The complexity class consisting of all NP-complete problems is sometimes referred to as NP-C.

Oligonucleotides are short sequences of nucleotides (RNA or DNA), typically with twenty or fewer bases. Oligonucleotides are often used as probes for detecting complementary DNA or RNA because they bind readily to their complements.

Organelle is a specialized substructure of the cell, such as a mitochondrion, Golgi complex, lysosome, endoplasmic reticulum, ribosome, centriole, chloroplast, cilium, or flagellum.

Overfitting: In machine learning is the problem that occus when a supervised algorithm adapts to the training samples too exactly, losing sufficient ability to generalize in the prediction of new samples.

Physical Maps represent the real arrangement of the sequences of ADN on the chromosomes.

Polymerase Chain Reaction (PCR) is a technique for enzymatically replicating DNA without using a living organism. The technique allows a small amount of DNA to be augmented exponentially. As PCR is an in vitro technique, it can be performed without restrictions on the form of DNA, and it can be extensively modified to perform a wide array of genetic manipulations.

Principal Component Analysis (PCA) is a technique for simplifying a data set, by reducing multidimensional data sets to lower dimensions for analysis. Technically speaking, PCA is an orthogonal linear transformation that transforms the data to a new coordinate system such that the greatest variance by any projection of the data comes to lie on the first coordinate (called the first principal component), the second greatest variance on the second coordinate, and so on. PCA can be used for dimensionality reduction in a data set while retaining those characteristics of the data set that contribute most to its variance, by keeping lower-order principal components and ignoring higher-order ones.

Proteomics is the large-scale study of proteins, particularly about their structures and functions.

Restriction Enzymes are enzymes that catalyze the splitting of DNA at specific points to produce discrete fragments.

RNA Polymerase II is an enzyme that polymerises ribonucleotides in accordance with the information present in DNA.

RT-PCR is the nomenclature used to indicate the chip that use PCR for augmenting the probes and reverse transcriptase for converting RNA target into DNA. So, they can be hybridized on the chip.

Saccharomyces Cerevisiae is also called budding yeast, is the common yeast used in baking ("baker's yeast") and brewing ("brewer's yeast").

Semantic Weight involves an heuristic measure of the interplay of concrete data with theoretical concepts.

Sequence Tag is a detached fragmentary piece of DNA segment. It's supposed that 14 bp tags occurs but once in the genome. Indeed, 14 bp was not long enough, and now scientists use 17bp or even 21bp.

Sequencing is the process of determining the nucleotide order of a given DNA or RNA

fragment.

Sequencing by Hybridization (SBH) is a class of methods for determining the order in which nucleotides occur on a strand of DNA.

Signal-to-Noise Ratio (SNR) is an electrical engineering concept defined as the ratio of a signal power to the noise power corrupting the signal. In microarray terms, signal-to-noise ratio compares the intensity level of one colored dye to the level of its background noise. The higher the ratio, the less obtrusive the background noise is.

Single-Gene Defects are genetic disorders determined by a single gene (mendelian disorders). It may be autosomal or X-linked, dominant or recessive.

Single Nucleotide Polymorphism or SNP is a DNA sequence variation occurring when a single nucleotide: A, T, C, or G - in the genome differs between members of a speciesor between paired chromosomes in an individual.

Target: In microarray speaking, target is a RNA or DNA sequence which represents (in a transcriptional way) the biological studied issue in a tissue, often modified by an external stimulus. The implication is that a molecule is "hit" by a signal and its behavior is thereby changed.

Test and Reference Samples are the two common divisions in epidemiological studies. In gene expression technologies speaking: test samples are the ones that own the studied aim (disease tissue, treated tissue, mutated tissue etc.) and the reference sample is the one that owns the starting point. (normal tissue, non-treated tissue, non-mutated tissue).

Transaction: In data-mining, it refers to the operations made in Market Basket Analysis MBA domain. MBA extracts associations among products that are frequently sold together in the same transation (market basket). Thus, data-mining transaction term it's different from the transaction database term.

Transfer RNA or tRNA is a small RNA chain (73-93 nucleotides) that transfers a specific amino acid to a growing polypeptide chain at the ribosomal site of protein synthesis during translation.

Z-score Test is an statistical test based in the normal distribution using the normal score as statistic. Normal score statistic is a dimensionless quantity derived by subtracting the population mean from an individual score and then dividing the difference by the population standard deviation.

Web Site Glossary

The reference for the terms containing in this glossary are marked with SMALL CAPS.

ARRAYEXPRESS: **http://www.ebi.ac.uk/arrayexpress/**

BIOBASE (bibliographic database of the worldwide biological research):
http://www.elsevier.com/wps/find/bibliographic_browse.cws_home

BIOCONDUCTOR is an open source and open development software project for the analysis and comprehension of genomic data:
http://www.bioconductor.org/

BIOGRID: **http://www.thebiogrid.org**

BIOSIS (Biology browser): **http://www.biologybrowser.org/**

BMRB contains data derived from NMR spectroscopic investigations of biological macromolecules: **http://www.bmrb.wisc.edu/**

CCD (THE CONSERVED DOMAIN DATABASE) contains protein domain models from several databases, such as Smart and Pfam:
http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?db=cdd

CGAP (CANCER GENOME ANATOMY PROJECT):
http://cgap.nci.ih.gov/Chromosomes/Mitelman

CGC (CANCER GENE CENSUS):
http://www.sanger.ac.uk/genetics/CGP/Census/

CLEMENTINE (SPSS): **http://www.spss.com/clementine**

COGs CLUSTERS OF ORTHOLOGOUS GROUPS OF PROTEINS:
http://www.ncbi.nlm.nih.gov/COG/

DBEST (EXPRESSED SEQUENCE TAGS DATABASE):
http://www.ncbi.nlm.nih.gov/dbEST/index.html

DDBJ (DNA DATA BANK OF JAPAN): **http://www.ddbj.nig.ac.jp/**

EMBL (NUCLEOTIDE SEQUENCE DATABASE): **http://www.ebi.ac.uk/embl/**

ENSEMBL PROJECT: **http://www.ensembl.org/index.html**

EUROPEAN BIOINFORMATICS INSTITUTE (EBI): **http://www.ebi.ac.uk/**

FLYBASE: **http://flybase.bio.indiana.edu/**

GENBANK: **http://www.ncbi.nlm.nih.gov/Genbank/**

GENE EXPRESSION OMNIBUS: **http://www.ncbi.nlm.nih.gov/geo/**

GENE ONTOLOGY (GO) PROJECT: **http://www.geneontology.org/**

GENECARDS: **http://www.genecards.org**

GENMAPP: **http://www.genmapp.org**

GENOMEPROJECT:
http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?db=genomeprj

Genomes:
http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?db=Genom

GGEG (Global Gene Expression Group):
http://sciencepark.mdanderson.org/ggeg

GO compendium: **http://www.geneontology.org/GO.tools.shtml**

GO Database: **http://www.godatabase.org/dev/database/**

HomoloGene:
http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?db=homologene

HPI (The human proteome Initiative): **http://www.expasy.ch/sprot/hpi/**

HUGO Human Nomenclature Committee:
http://www.gene.ucl.ac.uk/nomenclature/HUGO

Human Developmental Anatomy: **http://www.ana.ed.ac.uk/anatomy/humat/**

Human Genome Database: **http://www.gdb.org/**

Human Genome Project:
http://www.ornl.gov/sci/techresources/Human_Genome/home.shtml

Institute de Pharmacologie Moleculaire et Cellulaire (IMPC):
http://www.ipmc.cnrs.fr

Interpro: **http://www.ebi.ac.uk/interpro/**

IPI (International Protein Index): **http://www.ebi.ac.uk/IPI/IPIhelp.html**

KEGG: **http://www.genome.jp/kegg**

LinKBase: **http://www.landcglobal.com/pages/linkbase.php**

MeSH: **http://www.nlm.nih.gov/mesh**

MIAME: **http://www.nature.com/supplementary_info/**

MIPS (Munich Information Center for Protein Sequences):
http://mips.gsf.de/

Mouse Atlas Project: **http://genex.hgu.mrc.ac.uk/**

Mouse Genome consortium: **http://www.sanger.ac.uk/Projects/M_musculus/**

MSD-EBI is the european project for the collection, management and distribution of data about macromolecular structures: **http://www.ebi.ac.uk/msd/**

National Center for Biotechnology information (NCBI):
http://www.ncbi.nlm.nih.gov/

National Institute of Health (NIH) is the US departement of health and human services: **http://www.nih.gov/**

NCBI taxonomy:
http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?db=Taxonomy

NCBI-GEO: **http://www.ncbi.nlm.nih.gov/projects/SAGE/**

NetAFFX: **http://www.affymetrix.com/analysis**

Nucleotide: **http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?db=Nucleotide**

Online Mendelian Inheritance in Man is a catalog of human genes and genetic disorders: **http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?db=OMIM**

PDB: **http://www.pdb.org/pdb/home/home.do**

PDBj (Protein Data Bank Japan): **http://www.pdbj.org/**

Pfam (Protein families of alignments and HMMs):
http://www.sanger.ac.uk/Software/Pfam/

Pubmed browser:
http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?DB=pubmed

Pubmed/Medline:
http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?DB=pubmed

PubMed: **http://www.pubmed.gov**

RefSeq (Reference Sequence): **http://www.ncbi.nlm.nih.gov/RefSeq/**

SAGE libraries: Two sites centralized all libraries:
http://www.ncbi.nlm.nih.gov/SAGE/index.cgi and
http://cgap.nci.nih.gov/SAGE (cancer studies).

SAGE map resources repository:
http://www.ncbi.nlm.nih.gov/projects/SAGE

SAGElyzer: **http://www.bioconductor.org/repository/devel/package/html/SAGElyzer.html**

SAGEmap: **http://www.ncbi.nlm.nih.gov/projects/SAGE/**

SAM (Significance Analysis of Microarrays):
http://otl.stanford.edu/industry/resources/sam.html

SAM open source program: **http://www-stat.stanford.edu/~tibs/SAM**

SGD (Saccharomyces Genome Database): **http://www.yeastgenome.org/**

SGD's file of phenotype: **ftp://genome-ftp.stanford.edu/pub/yeast/data_download/literature_curation/phenotypes.tab**

SGD's manually curated papers: **ftp://genome-ftp.stanford.edu/pub/yeast/data_download/literature_curation/gene_literature.tab**

STACK project hosted and managed by South African National Bioinformatics Institute:
http://ww2.sanbi.ac.za/Dbases.html

Stanford Microarray Database: **http://smd.stanford.edu/**

Swissprot(Swiss Protein Knowledge): **http://expasy.org/sprot/**

TIGR (from Institute for Genomic Research): **http://www.tigr.org/**

Tools to analyze SAGE data: For human and mouse data, tools:
http://cgap.nci.nih.gov/SAGE. For yeast data at:
http://www.yeastgenome.org/help/querySAGE.html, yeast.

UMLS: **http://umlsinfo.nlm.nih.gov/**

UniGene (An organized view of the transcriptome):
http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?db=unigene

Uniprot(Universal Protein Resource):
http://www.ebi.uniprot.org/index.shtml

Woldwide protein data bank: **http://www.wwpdb.org/**

Part I: Molecular Biology and Gene Expression Technologies Basics

Chapter 1

Principles of Life in Molecular Biology and Gene Expression Technologies

The study of life has been one of the major goals for humanity. Since the earlier times, humans wanted to solve the mystery of life. Recurrent questions have been asked by human: How do the living organisms work? What are their components? Before answering this questions they began to describe the organisms and classify them into complex taxonomies. In the beginning of the nineteenth century, humans began to study cells and tissues of different organisms making possible the functional accounts of physiology. The revolution in biology over the last three decades resulted from understanding cells in terms of their chemistry [155].

These insights began with descriptions of molecules involved in living processes and providing an understanding of molecular structures and functions that are the key objects and actions of all organisms. More and more of the functions of life (e.g. cell division, immune reaction, neural transmission) are coming to be understood as the interactions of complicated, self-regulating networks of chemical reactions. The genetic material of the cell, specifies how to create proteins, as well as when and how much to create. Despite the complexity of these functions and components, insights in them are emerging rapidly. One of the reasons for that progress is the conception of life as a kind of information processing [155]. Thus, living systems are continuously acting guided by a set of instructions. These instructions are coded in 4 molecular letters (bases): adenine (A), guanine (G), cytosine (C) and thymine (T).

The new field created by this new conception of life is named bioinformatics. As define by the NATIONAL INSTITUTE OF HEALTH (NIH), bioinformatics consists of *"research, development, or application of computational tools and approaches for expanding the use of biological, medical, behavioral or health data, including those to acquire, store, organize, or visualize such data*". Thus, we can summarize bioinformatics as the science that studies computer databases and data mining algorithms to analyze proteins, genes, and complete collections of deoxyribonucleic acid (DNA) on a genome-wide level [238].

The last decade has seen a fast development of biological technologies. One of their huge results is the complete sequencing* of many important organisms. For example: the first sequenced genome of a organism with 1,830,137 base pairs*: *Haemophilus influenza* [117], the first eukaryotic genome with 12,068 kilo bases: *Saccharomyces cerevisiae* [131], the fruit fly, *Drosophila melanogaster*, with 120 mega bases [2] and many more specially microbial genomes. In 2003, the HUMAN GENOME PROJECT was completed obtaining 3 giga base pairs. Today the goal consists on understanding the functions hidden in the genome sequence, this period of time is also known as post-genomic era.

An essential challenge in the post-genomic era is the management and analysis of huge quantities of sequence data. The purpose of any organizational schema would be to provide biologists with a full catalog of genes and their functions used to assemble a living creature [171]. Recently, the advances in gene expression technologies have made it possible to monitor the expression levels* of thousands of genes in parallel. These technologies offer the first promising tool for addressing the challenges of the post-genomic era, by providing a systematic way to evaluate variations in DNA and RNA.

This chapter is divided in three sections: the first one provides a brief introduction to the salient characteristics of organisms, cells, nucleic acids, genes and their study under the optics of gene expression technologies in bioinformatics field. The second section describes the gene expression technology principles and it explains the different types of gene expression technologies: microarray and SAGE. It concludes with the presentation of several important applications of these technologies.

1.1 The Molecular Building Blocks of Life

The reader's understanding of the material discussed in the following chapters needs some familiarity with the molecular biology fundamentals of life. This section is intended to provide a basic background of this science; it is not a full explanation of the subject. Here, we explain some basic concepts knowing that there are few strict rules in this field, furthermore, these rules have exceptions. Therefore, this general presentation will not explore deviations and variations of the general principles studied here. For a more in-depth examination of molecular biology, readers can see: Science of Biology [245], Genes [178], Molecular Biology [314] and Computational Molecular Biology [274].

This section is organized as follows: the first sections define the molecular biology fundamentals of life as cells, proteins, nucleic acids as DNA and RNA, chromosomes, then, we present the central dogma of molecular biology, finally, we explain the protein synthesis process focusing in the information flow from gene to protein, emphasizing the transcription process, thus gene expression.

1.1.1 Organisms and cells

All organisms are constituted of cells, which can be decomposed into organelles, these organelles* into molecules, and so downward into smaller structures. The chemical composition of a cell is constant over his entire life. About 70% of any cell is water. About 4% are sugars and inorganic ions. Proteins make up from 15% to 20% of the cell. DNA and RNA range from 2% to 7% of the cell weight. The cell membranes, lipids and other similar molecules make up the remaining 4% to 7%. [6].

The Cell Theory [292] states three main principles:

1. All living things are composed of one or more cells
2. Cells are basic units of structure and function in an organism.
3. Cells come only from the reproduction of existing cells.

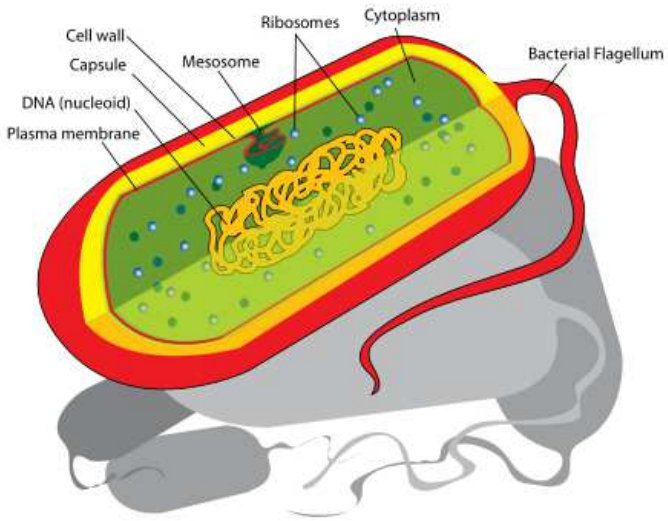

FIG. 1.1: Structure of a prokaryote cell

There are two types of cells: eukaryotes and prokaryotes, which are distinguished by their size and the type of internal structures or organelles that they contain, such as a mitochondrion, golgi complex, lysosome, endoplasmic reticulum, ribosome, chloroplast, etc. The structurally simpler prokaryotic cells are represented by bacteria and blue algae. However, most organisms which we can see as yeast, mushrooms, trees, butterflies, bats, dogs and humans consist of structurally more complex eukaryotic cells [162]. The distinction between these two kinds of cells is rather important, because many of the cellular building blocks and processes are quite different between them [162].

The internal structure and functions of eukaryotic cells are much more complex than those of prokaryotic ones. FIGURE 1.1 and 1.2 show the internal structures of a prokaryotic and an eukaryotic cell. Both cells contain a nuclear region which houses the cell's genetic material. The genetic material, DNA, of a prokaryotic cell is present in the nucleoid, which is a poorly-demarcated region of the cell, see FIG. 1.1. In contrast, eukaryotic cells possess a nucleus, a region bounded by a complex membranous structure called nuclear envelope, see FIG. 1.2. This difference in nuclear structure is the basis for the cells prokaryotic ($pro = before$, $karyon = nucleus$) and eukaryotic ($eu = true$, $karyon = nucleus$ [162]. In FIG. 1.3, we shows the structure of an eukaryotic nucleus and some of their main components as nucleolus and chromosomes. In the body of the nucleus is the chromosome territory and next to the nuclear lamina is the transcription site, as seen in FIG. 1.3.

Both prokaryotic and eukaryotic cells share a similar molecular chemistry. The most important molecules in the chemistry of life are proteins and nucleic acids. Speaking roughly, proteins determine what a living being is and does in a physical sense, while nucleic acids are responsible for encoding the genetic information and passing it along to subsequent generations [274].

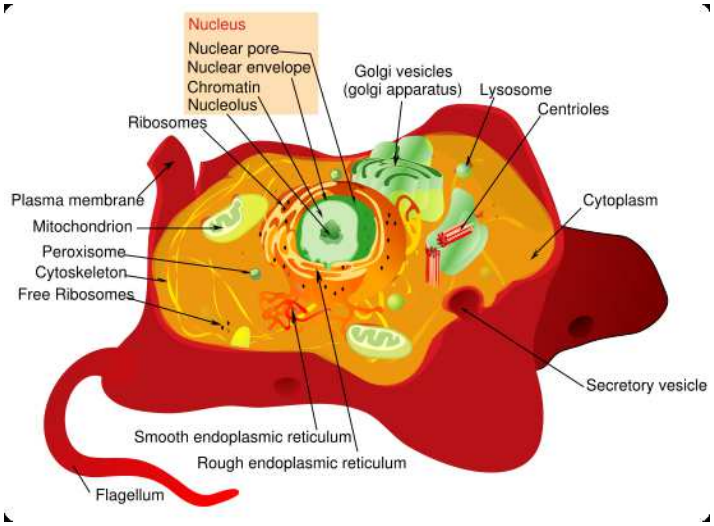

FIG. 1.2: Structure of an eukaryotic cell

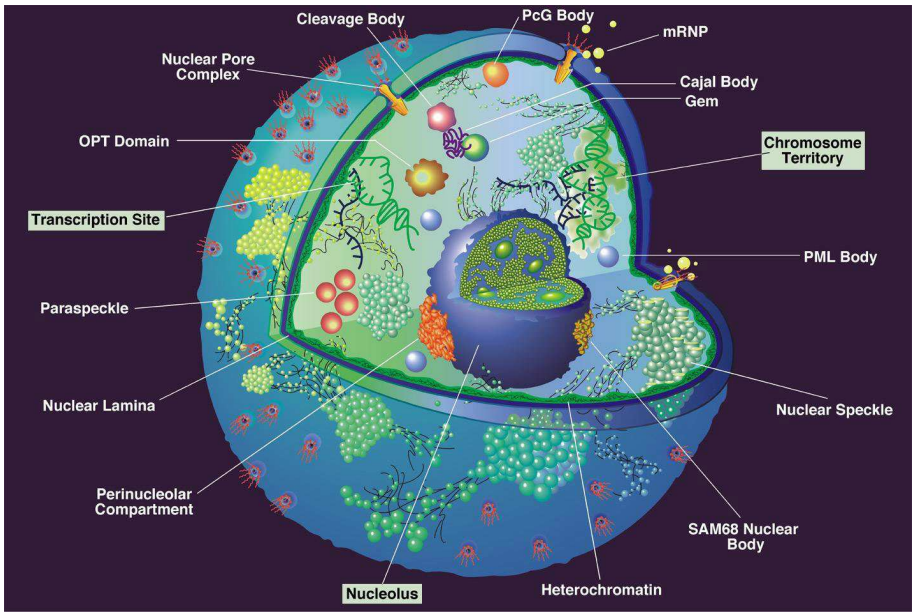

FIG. 1.3: Structure of a nucleus (Image from http://spectorlab.cshl.edu/index.html with permission of the author)

1.1.2 Molecules of life

There are four basic types of molecules involved in life: small molecules and macromolecules: proteins, DNA and RNA. In the following sections, we provide a brief description of these four kind of molecules.

1.1.2.1 Small molecules

Small molecules as water, sugars, fatty acids, amino acids and nucleotides can play different and independent roles. They are responsible for signal transmission, source of energy or material for a cell and even can be the building blocks of the three macromolecules mentioned above. For instance, there are 20 different amino acids molecules (see TABLE 1.1) that occur in nature, these molecules are the building blocks of proteins. Every amino acid is organized around a central carbon atom or alpha carbon. Other components of an amino acid include a hydrogen atom, an amino group NH_2, a carboxyl group $COOH$ and a side chain $R - group$ [245]. In FIG. 1.4 we show the structure of an amino acid. All living things (and even viruses, which do not fully meet "life" criteria) are made of various combinations of the same twenty amino acids.

#	Name	One-letter abrev.	Three letter abrev
1	Alanine	A	Ala
2	Cysteine	C	Cys
3	Aspartic Acid	D	Asp
4	Glutamic Acid	E	Glu
5	Phenylalanine	F	Phe
6	Glycine	G	Gly
7	Histide	H	His
8	Isoleucine	I	Ile
9	Lysine	K	Lys
10	Leucine	L	Leu
11	Methionine	M	Met
12	Asparagine	N	Asn
13	Proline	P	Pro
14	Glutamine	Q	Gln
15	Arginine	R	Arg
16	Serine	S	Ser
17	Threonine	T	Thr
18	Valine	V	Val
19	Tryptophan	W	Trp
20	Tyrosine	Y	Tyr

TABLE 1.1: The twenty amino acids commonly found in proteins.

1.1.2.2 Proteins

According the famous sentence of scientist Russell Doolittle *"We are proteins"*, proteins are the main building blocks and functional molecules of the cell, taking almost 20% of an eukaryotic cell's weight. Structurally, proteins are polypeptidic sequences, that is chain of amino

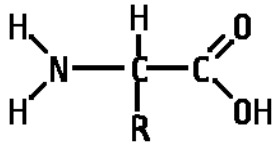

FIG. 1.4: General structure of an amino acid

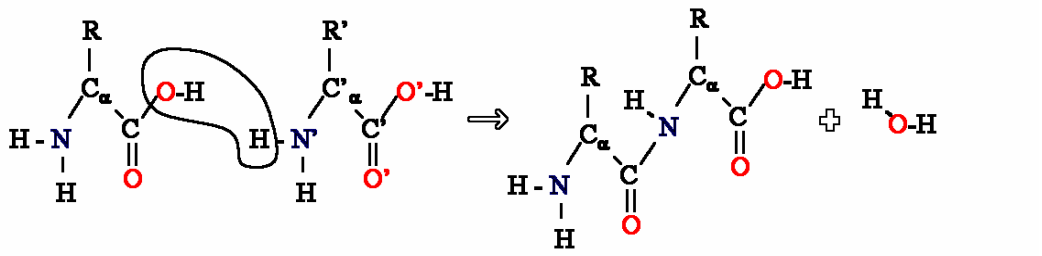

FIG. 1.5: Formation of a peptide bond between two amino acids by the dehydration of the amino end of one amino acid and the acid end of the other amino acid

acids which are linked together by peptide bonds (chemical bonds formed between the carboxyl groups and amino groups of neighboring amino acids). In FIG. 1.5 we show a peptide bond between two amino acids.

The sequence of molecules in a polypeptide is called the primary structure of a protein (see FIG. 1.6). This primary structure folds in three dimensions resulting in secondary, tertiary and quaternary structures. The three-dimensional shape of a protein determines its function. In FIG. 1.7 we can see three different tridimensional representations of the structure a protein. Structural biologists think that currently there are about 1,500 different representative protein structures known. Predicting protein structure from the amino acid sequence is the most important proteomics* problem in bioinformatics. For example, structural proteins, such as collagens, perform a variety of functions in living things as building tendons, connecting tissues, moving corneas etc.

There are several types of proteins: *structural* proteins that form part of a cellular structure, *enzymes* which catalyze almost all biochemical reactions occurring within a cell, *regulatory* proteins that control the expression of genes or the activity of other proteins, and *transport* proteins that carry other molecules across the cell membrane or around the body [10]. FIGURE 1.7 shows three possible representations of the three-dimensional structure of the protein triose phosphate isomerase. Left: all-atom representation colored by atom type. Middle: simplified representation illustrating the backbone conformation, colored by secondary structure. Right: Solvent-accessible surface representation colored by residue type (acidic residues red, basic residues blue, polar residues green, nonpolar residues white).

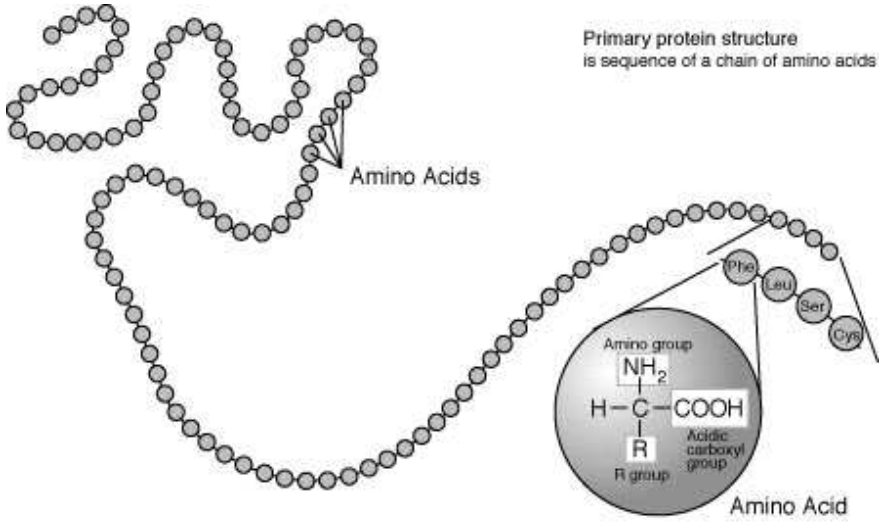

FIG. 1.6: Primary structure of a protein formed of amino acids chains and bound by peptide bonds

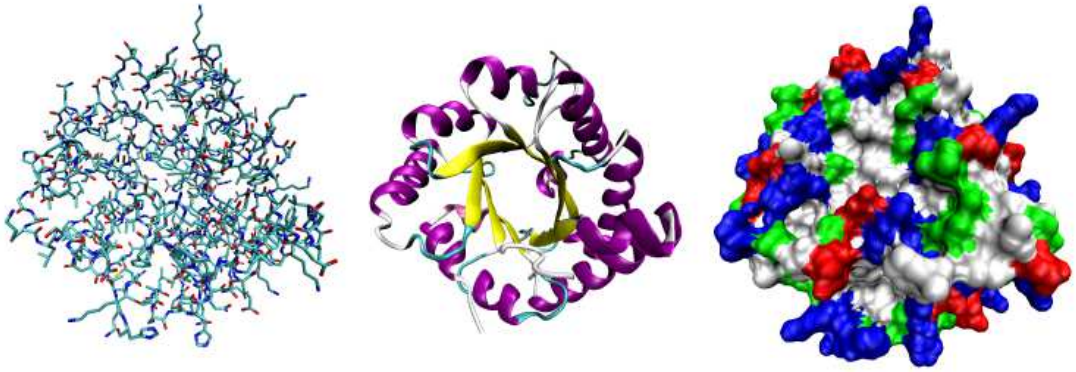

FIG. 1.7: Different views of the tridimensional representation of a protein.

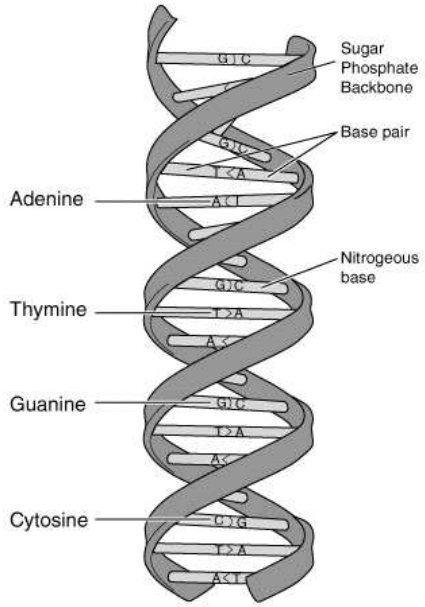

FIG. 1.8: Double helix structure of DNA composed of 4 bases: A, G, C and T.

1.1.2.3 Nucleic acids

Nucleic acids encode the information necessary to produce proteins and are responsible for passing along this "code" to subsequent generations [274]. There are two basic types of nucleic acid: *desoxyribonucleic acid* (*DNA*) and *ribonucleic acid* (*RNA*). The polypeptidic sequence (chains of amino acids joined together by peptide bonds) which forms the primary structure of a protein is directly related to the sequence of information in the RNA molecule, which, in turn, is a copy of the information in the DNA molecule, as stated in the *central dogma of molecular biology* (explained below).

DNA

DNA is a nucleic acid that carries the main information in a cell. It consists of two long strands of *small molecules* called nucleotides twisted into a helical structure and joined by hydrogen bonds (see FIG. 1.8).

Nucleotides are composed of three functional groups: a base, a sugar and a phosphate, however there are often referred next to the name of their base component. There are four different bases grouped into two types, purines: adenosine (A) and guanine (G) and pyrimidines: cytosine (C) and thymine (T) (see FIG. 1.8). The lector should notice that bases and nucleotides are not synonymous.

A single DNA strand has an orientation determined by the number of the carbon atoms, which, by convention starts at the $5'$ end and finishes at the $3'$ end, with the coding strand at top [274]. Two such strands are called complementary, if one can be obtained from the other

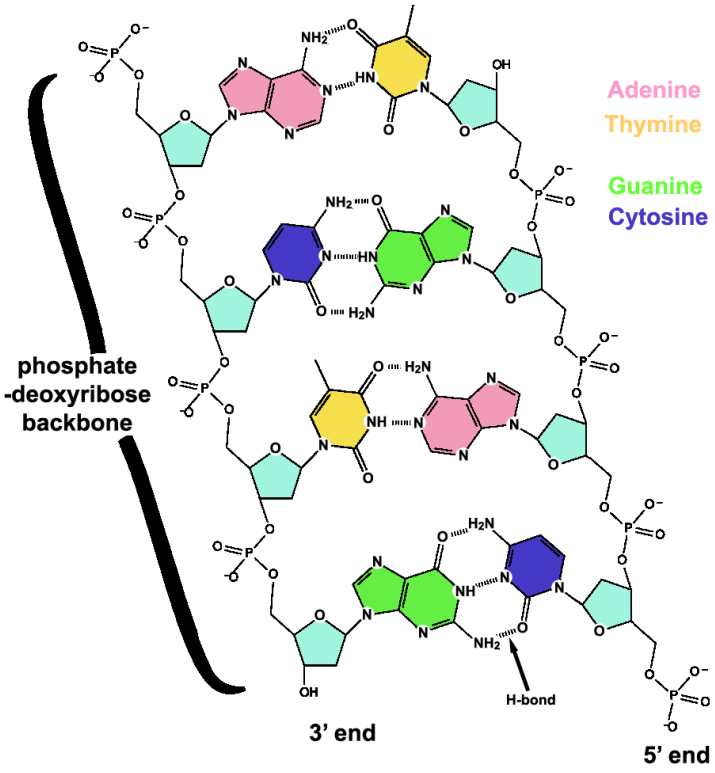

FIG. 1.9: Pairs of complementary bases form hydrogen bonds that hold the two strands of the DNA double helix together

by mutually exchanging A with T and C with G, and changing the direction of the molecule to the opposite (see FIG. 1.9)

DNA molecules are tied together with bases to the center (like steps on a ladder) and sugar-phosphate units along the sides of the helix (see FIG. 1.9). This double helix structure was discovered by Watson and Crick in 1953 (later on they got the Nobel prize for this discovery). The bases on the two strands are paired according to the *complementary base pairing rules* (also called Watson-Crick) base pairing rules): adenine (base A) only pairs with thymine (base T), and guanine (base G) only pairs with cytosine (base C) (see FIG. 1.9). The pairs are called *base pairs*, they provide the unit of length most frequently used when referring to DNA molecules and it is abbreviated in bp. Thus, we can state that a certain piece of DNA is 95,000 bp long, or 95kbp [274]. Although each individual bond is weak, the cumulative effect of many such bonds is sufficiently strong to bind the strands tightly together. So, DNA is chemically inert and is a generally stable carrier of genetic information [10].

As mentioned above, each DNA strand runs in a $5'$ to $3'$ orientation, they are antiparallel and complementary (see FIG. 1.9). As a consequence, it is possible to infer the sequence of one strand if we know the sequence of the other through an operation called *reverse complementation*. For example, given the strand $c = AGCTAAC$ in the $5'$ to $3'$ orientation, we first reverse c by $c^i = CAATCGA$, and then we apply the complementary base pairing rules of Watson-Crick obtaining $c^! = GTTAGCT$, which is the reverse compliment of the strand c [274].

This reverse complementation is precisely the mechanism that allows DNA in a cell to replicate. When the structure of DNA was deduced, it was understood that the complementary structure of the DNA molecule would allow exact self-replication, that means that information could be passed on from generation to generation [10].

RNA

RNA molecules are similar to DNA molecules, with the following basic compositional and structural differences [274]:

- The sugar component of RNA is ribose and not deoxyribose.
- In RNA, thymine (T) is replaced by uracil (U), which also binds with adenine (see FIG. 1.11).
- RNA does not form a double helix. RNA-DNA hybrid helices sometimes occur, or parts of an RNA molecule may bind to other parts of the same molecule by complementarity. The three-dimensional structure of RNA is far more varied than that of DNA (see FIG. 1.10).

DNA and RNA also differ in that while DNA performs essentially one function (that of encoding informations), cells contain a variety of RNA types, as *mRNA* and *tRNA,* each performing different functions [274].

There is a hypothesis that life on earth may have been RNA based. RNA can encode genetic information, is replicable, forms complex 3D structures (see FIG. 1.10) and can

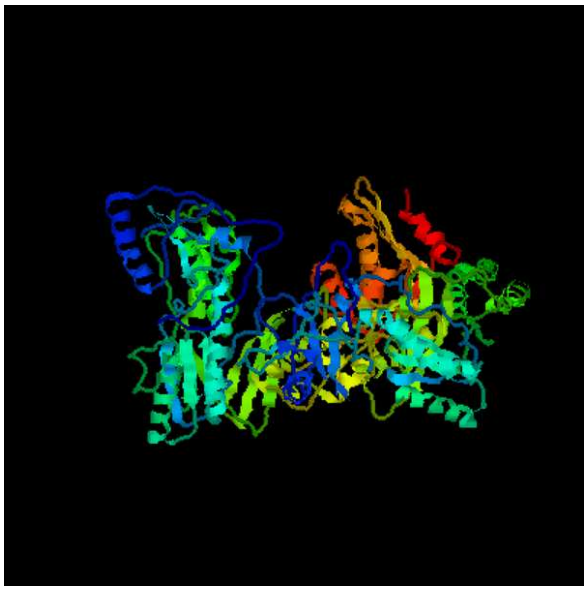

FIG. 1.10: Three dimensional representation of RNA

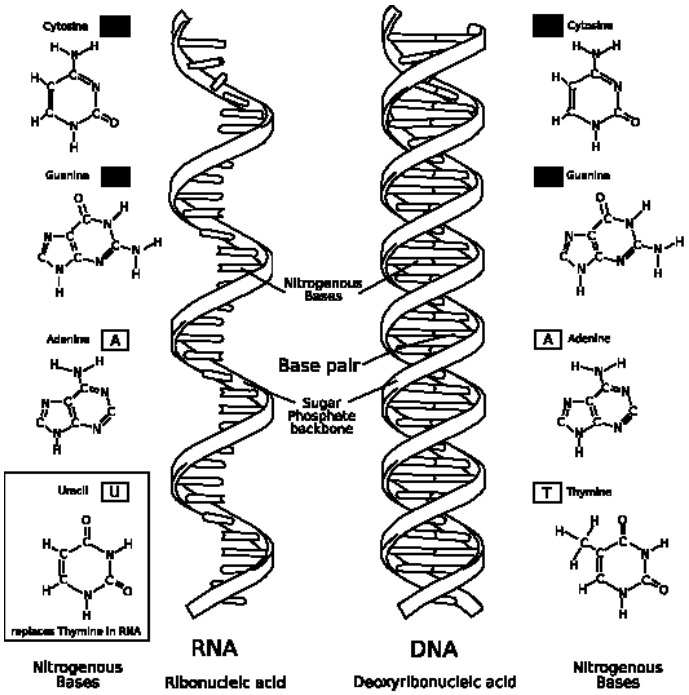

FIG. 1.11: Differences between RNA and DNA.

also act as a catalyst for certain chemical reactions related to splicing (protein synthesis is explained below) [49]. The next section will focus on DNA encoding mechanism and genes.

1.1.2.4 Genome, genes and genetic code

Each cell of an organism has one or more DNA molecules. Each DNA molecule forms a *chromosome* (see FIG. 1.12). The complete set of chromosomes in a cell is called a *genome*. All organisms have genomes and they are believed to encode almost all the hereditary information of the organism. All cells in a organism contain identical genomes (with few rather special exceptions) as a result of DNA replication* at each cell division. In eukaryotes, chromosomes are in the nucleus, which is not the case of prokaryotes which contain chromosomes in the cytoplasm (see FIG. 1.3 and FIG. 1.1 respectively). For example, every cell in humans has 46 chromosomes, in fruit flies 8 chromosomes and 32 chromosomes in yeast.

A DNA molecule contains certain contiguous stretches which encode information for building proteins. However, some portions of the DNA molecule do not contain encoded information but rather are termed junk* DNA. In this thesis[3], a **gene** *is a continuous stretch of a genomic DNA molecule, which contains the information necessary (encoded as a strand of A, T, G and C bases) to build a particular type of protein or many different proteins or even a RNA molecule* (see FIG. 1.12) This definition is not precise, and to better understand it we need to describe the molecular machinery making proteins based on the information encoded in genes. In FIG. 1.12 it is shown the internal DNA structure of a chromosome contained in the nucleus of a cell, as well as a fragment of DNA which encodes for at least a protein, this fragment is well known as gene.

The mechanism by which genes specify the sequence of amino acids in a protein is called *genetic code*. Specifically, a triplet of nucleotides or bases is used to characterize each amino acid. Such a triplet is called a *codon*. Given the four base types, the total number of possible base combination within triplets is $4 * 4 * 4 = 64$. However, these 64 combinations can only refer to the 20 existent amino acids. There is therefore redundancy in coding, and several different triplets will correspond to the same amino acid. For example, AAG and AAA code for lysine. Moreover, the three codon: TGA, TAG and TAA) do not code for any amino acid. Such redundancy is actually a valuable feature of the genetic code, rendering it more robust in the event of small errors in the protein synthesis process.

The next section explains the fondamental dogma of molecular biology that describes the information flow from DNA via RNA and thus to the proteins, which are the molecules of life.

1.1.3 The central dogma of molecular biology

The Central Dogma of Molecular Biology was first described by Crick in 1958, while studying the DNA molecules functions. As stated by Crick: "*DNA is responsible for two basic functions of every living organism: replication and protein synthesis*". The first is the basis of the

[3] There are many discussions between biologists to find a comprehensive unique definition of a gene. In fact, there exist several definitions of what a gene is.

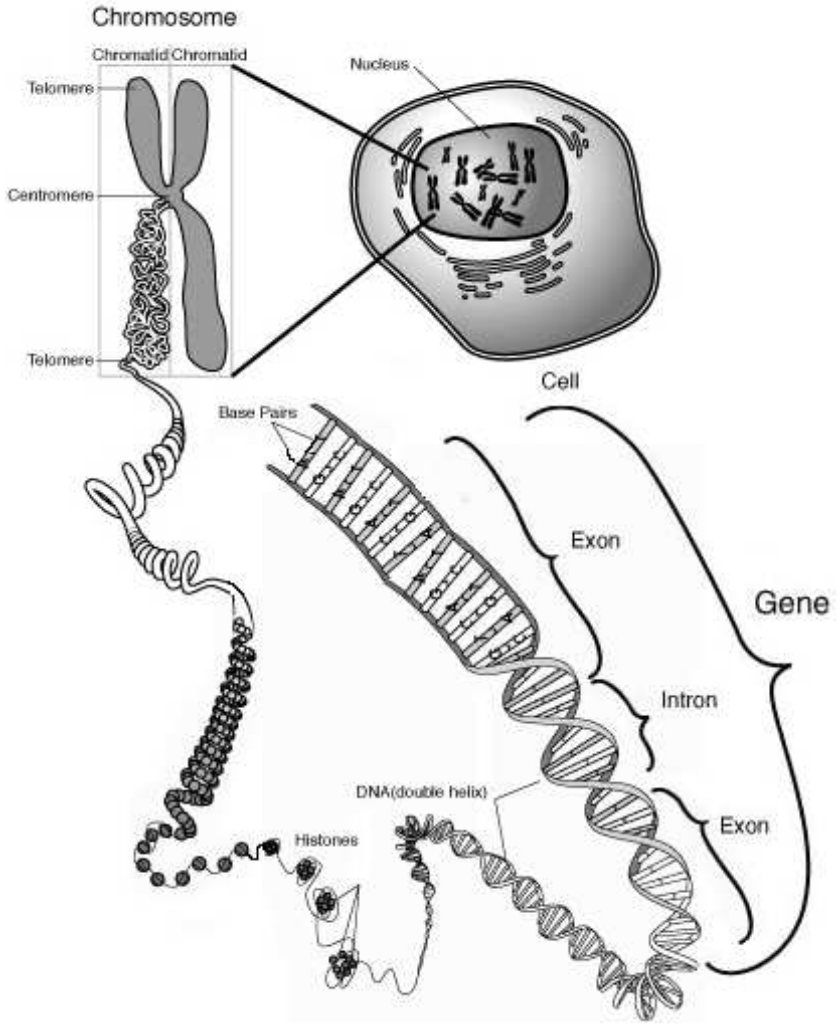

FIG. 1.12: Chromosomes, DNA and genes.

transmission of information from cell to cell via the cellular reproduction process. The second is the essence of living, i.e. encoding the information necessary to build each protein found in an organism.

In 1970 Crick [82] have formalized the so-called *central dogma of molecular biology*, which describes the information flow from DNA via RNA and thus to the protein, this process includes the following four major points :

1. The information contained in DNA is duplicated via the *replication process*[4].
2. DNA directs the production of encoded messenger RNA* (mRNA) through a process called *transcription*.
3. In eukaryotic cells, the mRNA is then processed by splicing and it migrates from the nucleus to the cytoplasm of the cell.
4. In the final stage of the information-transfer process, messenger RNA carries the encoded information to protein-synthesizing structures called *ribosomes*. Through a process called *translation*, the ribosomes use this coded information to direct protein synthesis.

The first point concerns the replication process that makes possible the transmission of hereditary information from cell to cell in the reproduction process as seen in Fig. 1.13. The three last points (2-4) represented by transcription, splicing, translation and protein synthesis constitute the four major steps for protein synthesis as seen in Fig. 1.13 and Fig. 1.14. The third point is achieved only by eukaryotic cells, not by prokaryotic cells that pass from transcription to translation directly.

In Fig. 1.13 we show the general information flow (with blue arrows) of a living organism. All the DNA in each cell is reproduced via the replication process and all the living processes are built via the DNA-RNA-Protein process as stated in the central dogma of molecular biology. In red arrows we show some special information flow processes accomplished by viruses only, lower life forms as prokaryotes and laboratory experiences.

The protein synthesis process has four essential stages, as stated in points 2-4 of the central dogma of molecular biology, which are: transcription, splicing, translation and protein synthesis (see Fig. 1.14). These steps are discussed in the next sections.

Transcription and Gene Expression

Transcription is the process of copying one DNA strand into a complementary mRNA (m stands for messenger) by the protein complex RNA polymerase II*. During transcription process, **genes play the role of copy templates and they are expressed, when its coding is transferred to an RNA molecule** (see Fig. 1.14).

To initiate a transcription process, the DNA double helix is opened at the starting point $5'$. Only one DNA strand serves as a template strand. Then RNA polymerase II enzyme

[4] DNA replication or DNA synthesis is the process of copying a double-stranded DNA strand. Since DNA strands are antiparallel and complementary, each strand can serve as a template for the reproduction of the opposite strand. The template strand is preserved as a whole piece and the new strand is assembled from nucleotide triphosphates.

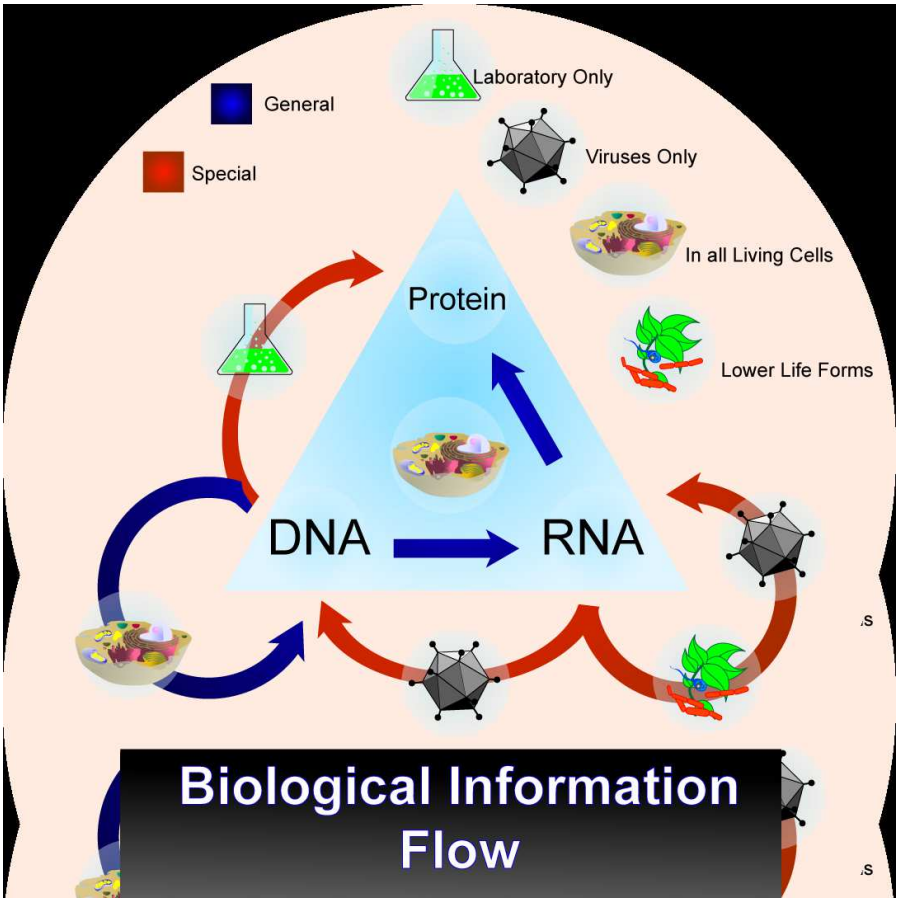

FIG. 1.13: Central dogma of molecular biology

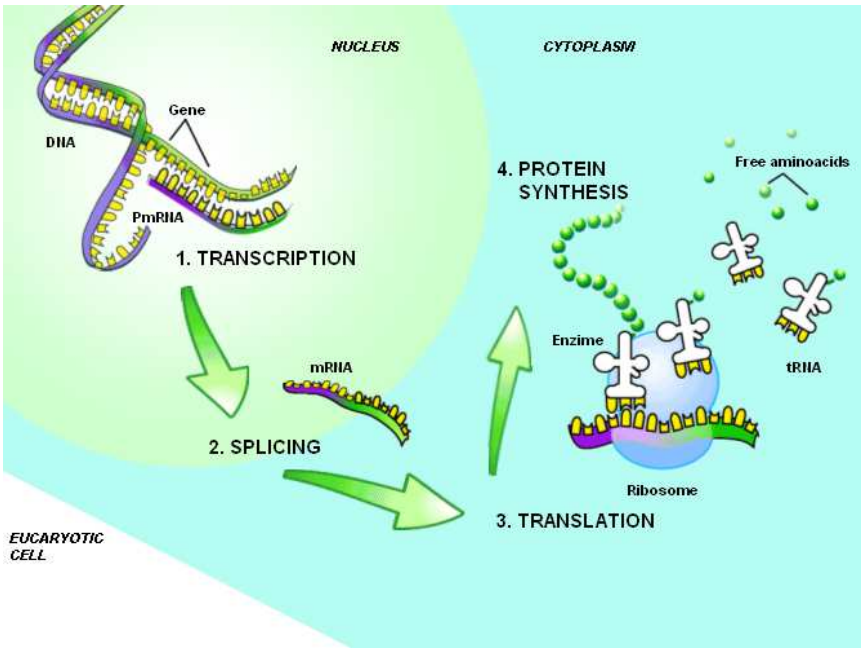

FIG. 1.14: Protein synthesis process as stated by the central dogma of molecular biology: transcription, splicing, translation and protein synthesis

copies from the start point $5'$ to the final point $3'$ the information contained in the single DNA strand into an preliminary $PmRNA$ molecule (see FIG. 1.14). This polymerization process consists in linking together complementary ribonucleotides to the template strand until the stop signal is reached. The resulting preliminary mRNA molecule contains the same ribonucleotide sequence as the DNA strand, with the base U substituted by T (see FIG. 1.11).

Transcription as described above is valid for prokaryotes and eukaryotes. Nevertheless, eukaryotes need an middle step before translation step, named splicing step.

Splicing

In eukaryotes, genomic DNA that corresponds to the coding part of genes is not continuous, but consists of exons and introns (see FIG. 1.12). Exons are the part of the gene that code for proteins and they are interspersed with non coding introns. After transcription process, the introns are removed out from the preliminary mRNA through the splicing process. The result of splicing is the mRNA molecule (see FIG. 1.14). *Alternative splicing* occurs when the same genomic DNA can give rise to two or more different mRNA molecules on the basis of alternative selection of introns and exons, generally resulting in the production of different proteins (please refer to [178] for details of introns, exons and splicing).

In genetics, *genomic DNA* is the entire gene or sequence as found in the chromosomes. The spliced sequence made up exons only is named *complementary DNA* or *cDNA*. The *c*DNA can be obtained through a reverse transcription process, which transforms mRNA back into cDNA.

Translation and Protein Synthesis

Once the transcription process has generated properly-encoded mRNA, the translation process which synthesizes proteins is initiated. Translation is the process of producing proteins by joining together amino acids in the order given by the mRNA (see FIG. 1.14). This process takes place in the cytoplasm where the mRNA interacts with ribosomes, which are large complexes of proteins and RNA molecules (precise interactions and functions of all proteins in ribosomes are not yet fully understood).

This process begins by the tRNA* or transfer RNA making possible the connection between a codon of the mRNA and the corresponding amino acid (contained in the ribosome). Each tRNA molecule has, on one side, an anticodon that has high affinity for a specific codon and, on the other side, an amino acid attachment site that binds easily to the corresponding amino acid. The attached amino acid falls in place just next to the previous amino acid in the protein chain being formed. After the translation of information from genetic codons to amino acids is finished the protein synthesis process begins (see FIG. 1.14).

In protein synthesis, a suitable enzyme catalyzes the addition of the current translated amino acid to the protein chain, releasing it from the tRNA. In this way, a protein is constructed amino acid by amino acid. When a stop codon appears, no tRNA associates with it and the synthesis ends. The messenger RNA is released and degrades by cell mechanisms into ribonucleotides, which will then be recycled to make other RNA [274] (see FIG. 1.14).

Until recently, biologists used to believe in the paradigm "One gene implies one protein". Now this assertion is known to be false. Due to alternative splicing and post-translational modifications one gene can produce a variety of proteins. There are also genes that do not encode proteins but RNA (for instance tRNA and ribosomal RNA) [49].

1.2 Gene Expression Technologies: Microarray and SAGE

It is widely believed that genes and their products* in a given living organism work in a complex and orchestrated way that creates life. One of the main challenges for gene expression technologies is to discover the hidden information and knowledge contained in the expressed genes while they are coding for any biological process A gene is expressed when its coding is transferred to an RNA molecule, through the transcription process (explained in section 1.1.2).

In past years (early nineties) traditional methods in molecular biology generally worked "on a one gene in one experiment basis", which means that the throughput is very limited and the "whole picture" of the gene functions and interactions was difficult to obtain [208]. The recent development of high-throughput micro-technologies has changed the experimental limits for gene expression quantification. In this manner, the expression levels of thousands of genes under tens of biological conditions can be monitored using high throughput gene expression technologies. Among these technologies we have: Microarray or DNA chips (*spotted cDNA chips* or *in situ DNA chips*) and SAGE (Serial Analysis of Gene Expression).These two technologies quantify the gene expression while the transcription molecular process. Never-

theless, microarray is based on hybridization* of DNA complementary strands, whereas *SAGE* is based on sequencing* sampling technique.

In this thesis we have mainly focused in microarray technology because of their inherent advantages over SAGE technology for many biological application (explained in the next sections).

In this section we briefly explain the gene expression technologies characteristics, focusing in the microarray technology[5]. First, we make a brief introduction to microarray and we describe the two main groups: spotted cDNA chips and in situ DNA chips. Next, we explain the four basic steps for microarray experiments: manufacture, sample preparation and labelling, hybridization, image scanning and processing. Finally, we explain the SAGE gene expression technology. Since now we will use the generic term of microarray to refer the sequencing by hybridization technology.

1.2.1 Microarray technology

In the most general form, a microarray or DNA chip is a solid support or chip made of nylon membrane, glass or plastic (or some other material). Usually, the chip is structured in a regular grid-like pattern. Segments of DNA strands are either deposited or synthesized within individual grids. These individual grids are normally fixed locations or spots. Each one of the spots contains a single defined species of a nucleic acid strand. Microarray technology is based on hybridization of nucleic acids [314]. In this technology, sequence complementarity leads to the hybridization between two single-stranded nucleic acid molecules, one of which is immobilized on a solid support ([289]). We can see the whole microarray process in Fig. 1.17.

Microarray is a generic term used to any gene expression technology that uses the sequencing by hybridization technique which contains about $10^2 - 10^8$ DNA molecules (or fragments of DNA) by spot and tens to thousands of spots by chip. Microarray term is used in opposition to the old macroarray technology.[208]. For gene expression studies, each of these molecules ideally should identify one gene or one exon in the genome. The first DNA chip containing all the genome of an organism was the *Saccharomyces cerevisiae** or *budding yeast* with about 6000 genes, have been available since 1997. DNA chips are high throughput technologies that allow scientists to analyze the expression of thousands of genes in a single experiment. They represent a major advance and a powerful tool to understand organism processes and many others applications (more details in section 1.3).

Modern microarray technology originated from Southern blot in 1988 (named after E. Southern british biologist). This technique employs radioactively labeled DNA (or RNA) hybridization probes* to identify very similar DNA sequences placed on a nitrocellulose filter called a blot [289]. Similar techniques are the Northern blot and the Western blot, which, respectively, employ RNA strands and proteins in place of DNA sequences [334]. McLachlan [208] briefly reviewed the history of the microarray technology. In the 1980's, a group led by R. P. Ekins was the first to use simple microspotting techniques to manufacture chips

[5] In this thesis we have mainly used microarray datasets for testing our interpretation models developped in the last chapters. The principal reason is the advantages presented by these tehnologies.

for high sensitivity immunoassay studies [108]. Numerous groups of researchers have further contributed to this technology. In 1987 the first patent on sequencing by hybridization* (SBH) has been done by R. Drmanac. In the late eighties, a patent battle over microarray technology began. Thus, companies as *Southern*, *Affymetrix*, *HySeq*, *Hoffman*, *LA Roche*, *Abbot* fought to obtain the microarray patent rights. Since the late nineties, numerous commercial entities and academic groups have contributed to advancements in microarray [120]. The fruits of these contributions are several technologies based in the SBH principle, but with technical differences in the manufacturing process.

One of the most common sources of confusion in microarray technology is the unclear use of the name because different types of microarray technologies are referred to with the same name. Furthermore, particular and general names may be used indistinctly. There are generic different names for microarray technology as DNA chips, DNA/RNA arrays, Biochips, GeneChips, Genome Chips, Gene arrays etc.

More particular names are given according to different parameters within microarray technology such as the manufacture process or the type of hybridization probe. If we refer to the manufacture process, microarray technologies are divided in two main approaches: *spotting* and *in situ synthesis* [270, 187, 271]. The spotting can be done with prefabricated oligonucleotides* or with PCR* products as probes. The spotting with PCR reaction is known as: RT-PCR, cDNA chips, spotted chips, spotted arrays. The spotting of oligonucleotides is known as: oligonucleotides chips, spotted chips, spotted arrays. In situ synthesis technology is always made with oligonucleotides as probes and it is synthesized by photolithography, ink jet printing and electrochemical synthesis. In situ synthesis of oligonucleotides is known as oligonucleotides chips, in situ chips and even by name of the enterprise that manufactures them: Affymetrix chips (that is the most used nowadays), Agilent chips, and so on. [98].

In order to avoid name confusion in this thesis we use the following agreements:

- The term for DNA chip is microarray.
- The term for spotted chip and in situ chip are the two general kinds of microarray.
- The term for spotted chip using PCR products as probes is RT-PCR* chip.
- The term for spotted chip using oligonucleotides as probes is spotted oligos-chip
- The term for in situ chip is in situ oligos-chip. In the case of a particular in situ chip we refer to as Affymetrix chip, Agilent chip etc.

In the next section, we describe the two main approaches to manufacture the microarray chips: deposition of DNA fragments by robotic spotting and in situ sythesis [172]. We explain briefly the characteristics of each of these manufacture technologies. We focus on the two currently most widely-used chips: the spotted chip and in situ oligos-chip, as examples of the spotting and in situ manufactures.

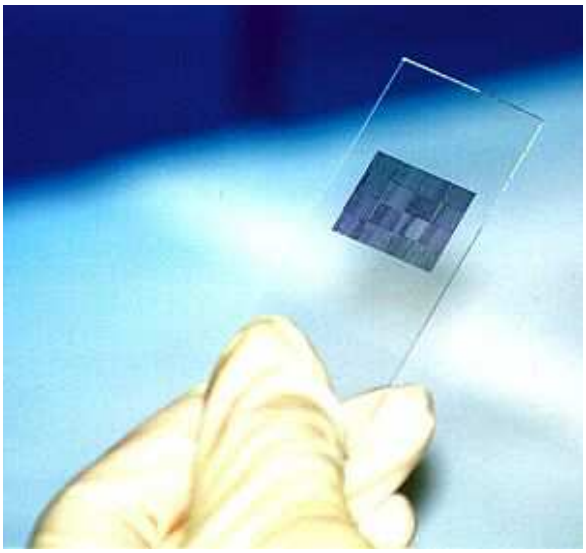

FIG. 1.15: Manufactured DNA chip within thousands of spotted probes in it.

1.2.1.1 Spotted chips manufacture

Manufacturing by robotic spotting may proceed through the deposition of PCR cDNA* clones or the printing of already synthesized oligonucleotides. FIGURE 1.15 shows a manufactured spotted chip with thousands of spotted probes.

The manufacturing of spotted chips involves three steps: selection of DNA probes[6], preparation of the probes, and the printing process (as seen in the first column of FIG. 1.17). Here, we explain this three-step process in the case of RT-PCR chips.

Selection of DNA probes

The choice of probes to be included in the chip depends on their biological application and the availability of the chosen sequences in the databases. In many cases, they are taken directly from sequence databases as the GenBank [29], dbEST [40], and UniGene [273], the resource of the microarray technologies [45, 104] (see FIG. 1.17).

Preparation of the probes

cDNA probes are prepared apart from the chip. Probes are PCR products. PCR technique creates billions of copies of specific fragments of DNA from a single DNA molecule. Then, the PCR products are partially purified by precipitation to remove salts, detergents and proteins present in the PCR cocktail [104] (see FIG. 1.17).

[6] In this chapter, we use the nomenclature proposed by Duggan [104] and refer to the DNA on the chip as probes and to the labeled DNA in solution as target.

Printing Process

Robots collect the cDNA probes and they begin the spotting printing process of the probes in the chip. Usually, the DNA is spotted onto a number of different chips, depending on the number of chips to be made [294].

An analogous three-step process is made to build spotted oligos-chip. The main difference is in the preparation of probes that are not PCR products, but already synthesized oligonucleotides. The first column of FIG. 1.17 shows us the three steps of spotted chips manufacture: probes selection, preparation and PCR cloning, finishing with the printing process on the chip.

1.2.1.2 In situ oligos-chip manufacture

In situ synthesized* chips are fundamentally different from spotted chips. FIGURE 1.16 shows the structure of a manufactured affymetrix GeneChip. Here, we present the three-step process of in situ oligos-chip manufacture, and the difference with the spotted chips manufacture [98].

Selection of DNA probes

Probe selection is performed based on sequence information alone. Hence, every probe synthesized on the array is known. In contrast on spotted chips which deal with expressed sequenced tags* (EST), the function of the sequence corresponding to a spot is often unknown.

Additionally, in situ oligos-chip selection approaches avoid duplicating identical sequence among gene family members. Thus, it can distinguish and quantitatively monitor closely-related genes [120]

Preparation of the probes

The probes are photochemically synthesized base-by-base on the surface of the array. There is no cloning and no PCR process involved.

Printing Process

Since the probes are synthesized on the surface of the chip, no printing process is needed.

The elimination of cloning, amplification, and printing of DNA reduces many sources of potential noise in the spotted chips system and thus constitutes a great advantage of in situ oligos-chip technology.

1.2.1.3 Microarray experiments procedure

Regardless of the microarray technology employed, a DNA chips experiment consists of five basic steps: manufacture, sample preparation and labelling, hybridization, image scanning, and image processing. In this section, the RT-PCR chip technology serves as a basis for a general discussion of these steps. The procedure of a RT-PCR chip experiment is illustrated in FIG. 1.17.

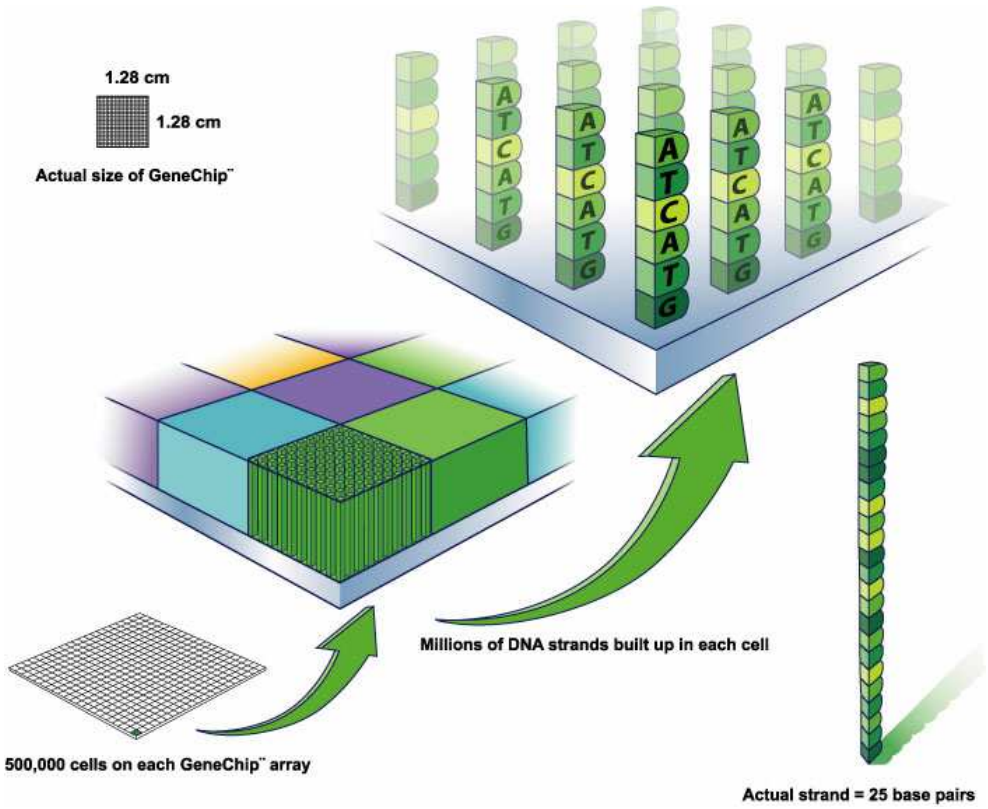

FIG. 1.16: Structure of an affymetrix genechip which contains 500,000 cells (in a chip of size 1.28cm X 1.28cm). Each cells contains millions of DNA chains. Each chain is built of 25 bases approximately

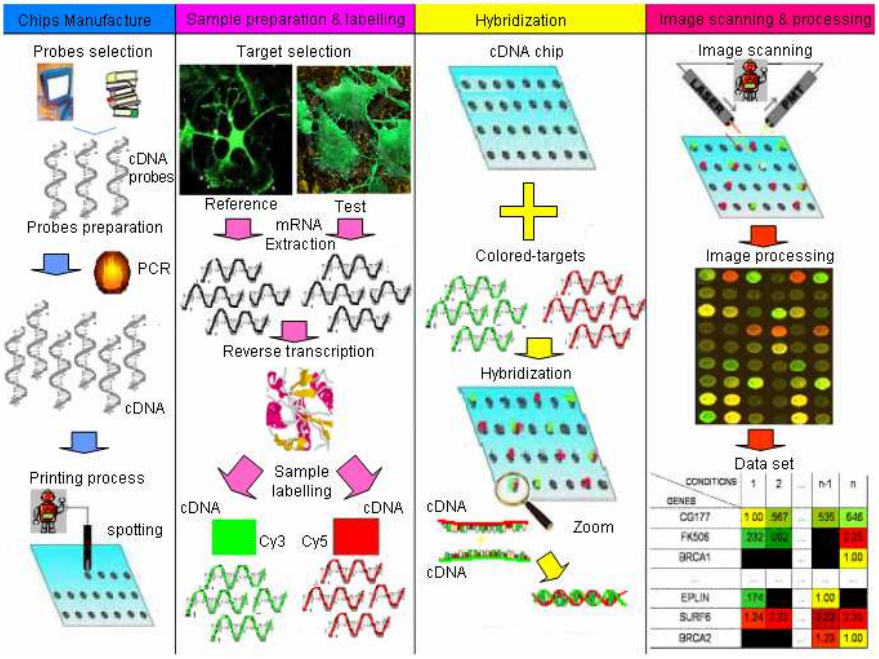

FIG. 1.17: Procedure of RT-PCR chip experiment

Manufacture

The manufacture process was explained above as a three-step procedure: selection of DNA probes, preparation of the probes and printing the probes in the surface of the chip. The goal of this process is to build the chip with the hybridization probes in it by spotted or synthesized manufacturing. In a general manner, hybridization probes are the DNA sequences that form the genes or a fragment of them, which we want to hybridize in the chip.

The manufacture process of an RT-PCR chip was explained above as the three-step procedure of spotted manufacture section. In the first column of FIG. 1.17 we illustrate the three steps of manufacture process: probe's selection, probe's preparation and printing of the probes in the chip.

Sample preparation and labeling

Biological sample preparation involves extracting and purifying the mRNAs from the tissue of interest. These extracted mRNAs are named the target* of the microarray experiment. Due to a number of challenges, the preparation can be quite variable [10, 294]. Among the problems of sample preparation we have the following:

- The target mRNA typically accounts for only a small fraction (less than 3%) of all mRNA in a cell.
- The target mRNA is very difficult to isolate completely. For example, mRNA comes from a heterogeneous range of cells, thus diseased tissue contains a mixture of normal tissue, inflammatory cells, necrotic tissue etc).

- The target mRNA degrades very quickly.

In order to avoid the rapid degradation of the mRNA target, in RT-PCR chip preparation mRNA is usually reverse-transcribed into more stable cDNA immediately after extraction [10] (see second column of Fig. 1.17). Afterwards, to allow detection of these cDNA sequences in the hybridization process on the chip, the cDNA goes through a platform-dependent labelling process.

Detection of cDNA on RT-PCR chips was previously performed using radioactively-labeled DNA, but actually it is more common to use dyes which fluoresce when exposed to a specific wavelength of light. In most experiments, two samples are hybridized to arrays, each labeled with Cy3 and Cy5 dye, which are excited by green and red lasers respectively [10]. This results in a two-channel RT-PCR chip experiment and allows the simultaneous measurement of both samples: Test* and Reference* (see second column of Fig. 1.17).

Hybridization

Hybridization is the step in which the DNA probes on the microarray and the labeled DNA (or RNA) target form heteroduplexes* according to the Watson-Crick base-pairing rule (see section 1.2) [294]. The biological principle here states that a single-stranded DNA molecule will bind to another single-stranded DNA molecule with a precisely matching sequence with much higher affinity than that to an imperfectly matching sequence [10]. We can see hybridization step illustrated in the third column of Fig. 1.17.

In fact, hybridization is a complex process, and a DNA segment may also bind well to a sequence similar but not identical to its complementary target, a phenomenon called cross-hybridization. This is influenced by many conditions, including temperature, humidity, salt concentration, target solution volume, and hybridization operator [294].

Hybridization may be performed either manually or by a robot. After hybridization, the microarray is washed to eliminate any excess labeled sample so that only the DNA complementary to the probes remains hybridized on the chip. Finally, the microarray is dried using a centrifuge or by blowing clean compressed air [10].

Image scanning

After the hybridization process, the surface of the hybridized chip is scanned to produce a microarray image. As previously mentioned, samples are labeled with fluorescent dyes that emit detectable light when stimulated by a laser. The emitted light is captured by the photomultiplier tube in a scanner, and the intensity is recorded [294]. We can see the image scanning step illustrated at the top right of Fig. 1.17.

Although the scanner is only intended to detect light emitted by the target DNA strands, it also will capture incidental light from various other sources. These other sources may include labeled DNA samples which have hybridized non-specifically to the glass slide, residual labeled samples, various chemicals used in processing the slide, and even the slide itself. This incidentally-captured light is called background [10].

The final result is a monochrome image of an in situ oligos-chip or two-color images in cDNA chips, which are usually stored in a typical image file format (often TIFF: tagged image format). We can see in the right middle part of the Fig. 1.17 the image scanning results of a RT-PCR chip technology experiment.

Image processing

The microarray image generated by the scanner forms the raw data of the experiment. Prior to data analysis, the image must be converted from image format into the numerical information that quantifies gene expression. The manner in which this is accomplished will have a major impact on the quality of the resulting data and the success of further analysis.

In the case of in situ chips, the commercial brands as Affymetrix and Agilent have integrated image-processing algorithms into their software packages, allowing end-users to directly generate quantified microarray data.

In contrast the images from cDNA chips consist of spots arranged in regular grid-like patterns. The processing of these images is done in four basic steps:

1. *Spot identification* involves locating the position of individual signal spots in an image and estimating their size.
2. *Image segmentation* involves decomposing an image into a set of non-overlapping regions. It consists in the differentiation of those pixels which form the spot and should be included in the calculation of the signal from those pixels which are background or noise and should be eliminated.
3. *Spot quantification* involves calculating the intensity for each spot. Here, pixel intensity values are combined into a unique value representing the expression level of a gene deposited in a given spot.
4. *Spot quality assessment* involves calculating some quality-control measures which evaluate the quality of both the entire chip and the individual spots on the chip. These measures can help human inspectors in determination of the data reliability and the identification of those spots with questionable quality values.

In Fig. 1.17 we can see the raw data of an RT-PCR chip experiment after image processing. These data are presented like a big matrix of thousands of genes as rows and tens of biological conditions as columns. Each position (i, j) in the matrix represents the gene expression measure of gene i under the biological condition j. For example in the bottom right part of Fig. 1.17 we can see the third row matrix that shows the gene BRCA1. This gene is not expressed at all (black color) under biological conditions 1 and 2, but is equally expressed in reference and test sample under the condition n (the quotient reference vs test is equal to 1).

1.2.2 SAGE

The Serial Analysis of Gene Expression (SAGE) method for detection of mRNA transcripts in eukaryotes is based on the sequencing of concatemers* of short sequence tags* that originate

from a known position (after the 3'-nearest cutting site of a restriction enzyme) to estimate transcripts abundance [311] (see FIG. 1.18).

The original technique was developed by Dr. Victor Velculescu [311]. Several variants have been developed since, most notably a more robust version, LongSAGE (developed by Dr. Saurabh Saha), which enables annotation of existing genes and discovery of new genes within genomes [265].

1.2.2.1 SAGE basics

SAGE takes advantage of the transcription output, i.e. messenger RNA, which contains the gene expression information. SAGE technology measures the number of transcripts[7] (or sequences of mRNA) produced in a biological experiment. In order to measure this number, it uses sequencing techniques[8]. Sequencing goal is reading all the nucleotide spelling (A, C, G, U) of a sequence [266]. Sequencing is a complex procedure, and reading the entire sequence of every RNA in a cell would take decades. So, SAGE scientists have proposed the use of only fourteen letters or nucleotides of a transcript in order to match an RNA to the precise gene that produced it [311].

Therefore, SAGE captures the mRNAs transcripts from the cells, then it reverse-transcribes it into more stable cDNA. Afterwards, it cuts all cDNA sequences into smaller fourteen-letter tags. Since it would take a long time to load tens of thousands of single tags into a sequencing machine, the method glues a lot of tags together into long molecules called concatemers. Later on, a machine called *sequencer* reads these molecules, counts the tags and analyses them by computer programs which relates every tag to an existing gene [266].

1.2.2.2 SAGE experiments procedure

We can divide the SAGE process in a general four-step procedure (as seen in FIG. 1.18) :

1. Sample preparation consists in extracting the mRNAs from the tissue of interest, and then converting these mRNAs into cDNAs by reverse transcription.
2. Building tags and concatemers involves converting the cDNAs sequences into fourteen-letters tags and gluing a group of tags together into concatemers.
3. Concatemers sequencing is a sequencing sampling technique which counts the number of times that every tag appears in the concatemers.
4. Tags-Genes correspondence involves matching the sequence of each tag with the gene that has produced the mRNA

[7] Transcript is a fragment of mRNA which contains the encoded information of a gene.

[8] Currently, most DNA sequencing is performed using the chain termination method developed by Frederick Sanger. However, there exist other methodologies as *454 sequencing* and *pyrosequencing*.

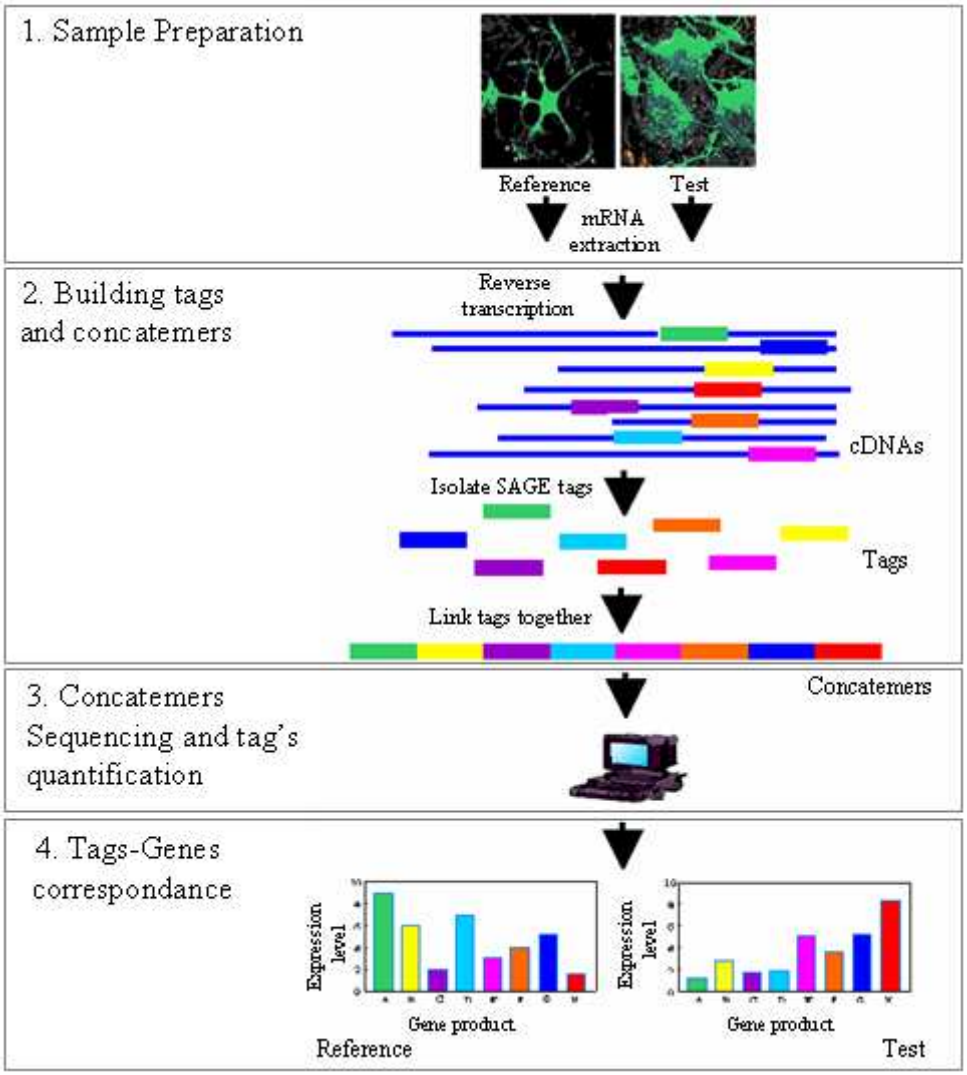

FIG. 1.18: Four step procedure of SAGE experiments: sample preparation, building tags and concatemers, contactemers sequencing and tags quantification and tags-genes correspondance.

1.2.2.3 Advantages and disadvantages of SAGE

Here we describe some of the advantages and disadvantages of the SAGE technology as stated in Martinez et al.[199].

Among the advantages of the SAGE technology we have the following:

- SAGE method estimates the expression level of transcripts without prior knowledge of their sequences and is more sensitive than the sequencing by hybridization (SBH) technique, but requires knowledge of the complete genome [297].
- SAGE method performs a random sampling of transcripts in a particular tissue, with little sequencing effort.

In contrast, SAGE presents well-known drawbacks:

- PCR and sequencing errors may be high.
- A single error may lead to non-recognition of a transcript or wrong attribution.
- Some tags may be present in more than one gene (in the case of 14 bp. tags).
- Restriction enzymes* may not cut with 100% efficiency. Thus, some tags may be wrong.

1.3 Gene Expression Technologies Applications

Gene expression technologies have already been extensively used in biological research to address a wide variety of questions. As stated by Collins [78], when applied to expression analysis, this approach facilitates the measurement of RNA levels for the complete set of transcripts of an organism. When applied to genotyping, these technologies allow the contemplation of whole genome-association studies to determine the genetic contribution of complex polygenic disorders. Moreover, the application of these technologies to mutation detection of disease genes opens the possibility of genetic testing for disease susceptibility of individuals, or even entire populations, into the sphere of practical reality.

In these section, we present a few examples of general applications of gene expression technologies. This overview represents only a fraction of the universe of potential applications.

1.3.1 Functional genomics in cells and tissues

Functional genomics is a new field of molecular biology that attempts to make use of the vast wealth of data produced by genomic projects to describe gene (and sometimes protein) functions and interactions. This science focus on the dynamic aspects of the protein synthesis process as gene transcription, translation, and protein-protein interactions.

Gene expression patterns provide indirect information about gene function and interactions. It is known that cells from different tissues perform different functions. Although they can be easily distinguished by their phenotypes, a detailed understanding of the mechanisms of these different behaviors remains undiscovered [10]. Cell function is determined by indi-

vidual proteins and protein synthesis is dependent on which genes are expressed or not, the expression pattern of a gene provides indirect information about cell function.

For example, a gene expressed only in the lung is unlikely to be directly involved in the pathology of schizophrenia [87]. Gene expression technologies can be used to identify those genes which are preferentially expressed in various tissues. This would enable scientists to gain valuable insights into the mechanisms that govern the functioning of genes and cells [10].

1.3.2 Gene expression patterns in model systems

Detailed profiling of gene expression in model systems (such as *Saccharomyces cerevisae*, *Drosophila melanogaster*, *Arabipopsis thaliana* etc.) yield valuable insights into the functions of genes and the mechanisms of important cellular processes, as well as to animal and human physiology. Such functional knowledge of the biological mechanisms could be critical to the discovery and validation of therapeutic targets [87].

For example, Spellman and his colleagues [290] used RT-PCR chips to create a comprehensive catalog of yeast genes of which the transcript levels vary periodically within the cell cycle of yeast. They reported 800 genes with the same expression profile that participate in cell cycle regulation.

In another study, Gasch et al. [123] used microarrays to observe genomic expression in yeast responding to two different DNA-damaging agents. They found that these are gene expression responses that are dependent on the Mec1 signaling pathway, which is a signal transducer required for cell cycle arrest and transcriptional responses prompted by damaged or unreplicated DNA. In particular a set of genes appeared to represent an Mec1-dependent expression signature of DNA damage, cell cycle, mutations, and stimulus.

1.3.3 Molecular pathology

Molecular Pathology is a new discipline focused on the use of gene expression technologies for specialised studies of disease in tissues and cells. One of the most attractive applications of gene expression technologies is the study of the differential gene expression in diseases. There are many genetic diseases that are the result of mutations in a gene or a set of genes. The mutations may cause genes to express inappropriately or to fail to express. For example, cancer could occur when certain regulatory genes, such as the $p53$ tumor suppressor gene, is always transcribed regardless of any regulatory factor [10].

Gene expression technology can be used to identify which genes are differentially expressed in diseased cells (test sample) versus normal cells (reference sample). The opportunity to compare the expression of thousands of genes between "diseased" and "normal" cells allows the identification of multiple potential targets [87]. This enables the development of drugs. Such drugs can be designed to specifically target a particular gene, protein or signaling cascade, and they are therefore less likely to cause undesirable side effects [10]. For example, rheumatoid tissue was analyzed using a microarray with thousands of genes, the results have shown that about 100 known genes have a role in inflammation of this kind of tissue, more details in [143].

1.3.4 Pharmacogenomics

Pharmacogenomics is the branch of pharmaceutics which deals with the influence of genetic variation on drug response in patients by correlating gene expression with a drug's efficacy or toxicity. Gene expression technologies are powerful tools for investigating the mechanism of a drug action. For example, $interferon-\beta$ is the most widely prescribed immunomodulatory therapy for multiple sclerosis (an autoimmune disease of the brain and spinal cord). To define the mechanism of $interferon-\beta$ and investigate the partial responsiveness of various patients, the expression levels of large numbers of genes were monitored for thirteen multiple sclerosis patients during a ten-point time series [315].

There are other applications for gene expression technologies in examining the effects of drugs on gene expression of organisms (yeast) as model system. These can lead to the identification and validation of novel therapeutics [87].

1.3.5 Pathogen genomics

Pathogen genomics concerns the study of gene expression patterns in *pathogenes* such as bacteria, viruses, parasites etc. For example, the activity in the sequencing of bacterial genomes is intense, with a new bacterial genome seemingly sequenced entirely every month [87]. The small size of these genomes allows the easy construction of individual DNA chips in which every gene from a given microbe is represented. For microbiologists, restricted for years to studying bacteria one gene at a time in a test tube under artificial growth conditions, the horizons appear unlimited. Gene expression technologies will identify genes that are turned on in vitro but not at the site of infection in vivo and vice versa. Such genes encode virulence determinants that are regulated by environmental signals such as transition from ambient temperature to body temperature [210]. Since traditional genetic techniques used to identify virulence genes are time consuming, they will be quickly replaced by gene expression technologies.

A similar approach is used to study viral gene expression during the time course of acute infection or during latency. These technologies can also be used to study the response of the host to challenges from the pathogen.

1.3.6 Developmental genetics

This branch of genetics primarily concerned with the manner in which genes control or regulate the organisms development. The genes in an organism's genome express differentially at different stages of the developmental process [10]. Interestingly, it has been found that there is a subset of genes involved in early development that is used and reused at different stages in the development of the organism, generally in different order in different tissues, with each tissue having its own combination. Crucial to these processes are growth factors, which can also, later in an organism's development, be involved in causing or promoting cancer (these genes are known as proto-oncogenes) [10].

Gene expression technologies can be used to track the changes in the organism's gene expression profile, tissue by tissue, over the series of stages of the developmental process, beginning with the embryo and up to the adult. Other applications in the same line of research include deducing evolutionary relationships among species and assessing the impact of environmental changes on the developmental process of an organism.

1.3.7 Gene mutation detection of complex diseases

Complex diseases are not caused by a few errors in genetic information but by a combination of small genetic variations (polymorphisms) which predisposes an individual to a serious problem [10]. The risk of such an individual contracting a complex disease tends to be amplified by non-genetic factors, such as environmental influences, diet and lifestyle. Multiple sclerosis, diabetes, schizophrenia are complex diseases in which the genetic makeup of the individual plays a major role in predisposing the individual to the disease. The genetic component of these disease is responsible for the increased prevalence within certain groups such as families, ethnic groups, groups of geographic regions, and gender. Gene expression technology experiments can be used to identify the genetic markers, usually a combination of SNP's* that may predispose an individual to a complex disease.

1.3.8 Genotypic analysis

Variation in DNA sequences cause most of the differences we observe within and between species. Locating, identifying and cataloging these genotypic differences represent the first steps in relating genetic variation to phenotypic variation in both normal and diseased states [182]. Lipshutz et al . [182] described a specific type of chip that is designated for this purpose.

Single-nucleotide polymorphisms (SNPs) are the most frequent type of variation in the human genome and they are recommended for genotypic analysis [182]. For example, the study of Wang and colleagues [313] identified 3241 candidate SNP's. Using microarray technology they have screened for variations among 8 individuals to identify candidate SNP's and create a third-generation genetic map for the human genome.

In other application, DNA chips are also be used to scan the genome for new SNP's, more details in [182].

Chapter 2

Gene Expression Data Analysis Procedure

Gene expression technologies facilitate the monitoring of changes in the expression patterns of large collections of genes. The analysis of gene expression data has become a computationally and methodologically intensive task that requires the development of bioinformatics technology for a number of key stages in the gene expression data analysis procedure. The goal of this chapter is to explain these key stages.

We suppose that the analysis of any kind of gene expression data regardless to the technology employed (microarray or SAGE) begins with the physical manufacture of the chip or experiment and ends with the quantification of the biological experiment. In this chapter, we summarize the gene expression experiments steps (explained in chapter 1) in the data generation step (as seen in the first row of FIG. 2.1). The output of the data generation step is the raw gene expression data.

The raw microarray data are real values that represent the light intensity or gene expression measure of thousands of genes in tens of conditions measured in a biological experiment. Raw microarray data can be noisy and sensible data because of the data generation procedure. Each step of the data generation procedure may contain several sources of noise. The following five sources of variation, may generate noisy microarray data [31]:

- Variations in the manufacture of the chip: preparing the glass, DNA amount, PCR yield, the spotting technique etc.
- Variations in the sample preparation of a microarray: culture extraction, RNA extraction, the reverse transcription, the labelling step etc.
- Variations in the hybridization procedure: differential sensibility of the genes, the unspecific cross-hybridization effect etc.
- Variations in the image scanning procedure: scanning technique, distortion, scanning manipulation errors, etc.
- Variations in the image processing procedure: methodology for estimating the spot intensity, adjustments of image parameters, etc.

Microarray data contain important information about several biological processes that could be crucial for knowledge discovery in any of the applications explained in section 1.3.

Extracting information and knowledge from gene expression technology is not an easy task and must be carried out by a series of key stages known as microarray data analysis procedure.

The whole procedure for analyzing gene expression technology data is composed of five steps: data generation, statistical data treatment, analysis of differentially expressed genes, classification of genes as well as data interpretation and knowledge discovery. FIGURE 2.1 shows the five-step procedure.

Data generation was fully explained in chapter 1 for the two main expression technologies microarray and SAGE. In microarray experiments, this procedure consists in: manufacture, sample preparation and labeling, hybridization, image scanning and image processing. In the case of SAGE, it consists in four steps: sample preparation, building tags and concatemers, concatemers sequencing and tags-gens correspondence. The output is raw data as stated in FIG. 2.1.

Statistical data treatment involves statistical manipulations as: data transformation, missing value estimation and data normalization for cleaning, preprocessing and processing microarray data. Its output data are well-cleaned and ready to be analyzed (see FIG. 2.1).

Analysis of differentially expressed genes identifies those genes which demonstrate a significant change in expression level under the impact of certain experimental conditions, such as in cancer studies, reference= normal and test=cancer (explained in section 1.2). In this step, statistical and data mining tools are used to distinguish between the genes that are over-expressed or under-expressed from the genes that are constant or not expressed over all the biological experiments, or even only one biological condition. The output at this stage is a list of genes ordered by rank or a selection of differentially expressed genes (as seen in FIG. 2.1).

Classification of the genes involves using unsupervised learning techniques such as "clustering" for identification of groups of co-expressed genes* with coherent gene expression patterns*. The output is a classification of genes by similarity of their expression profile*.

Knowledge discovery and interpretation consist in interpreting the microarray data via integration of gene expression profiles with corresponding biological knowledge. The output is knowledge discovery (as seen in FIG. 2.1). This step represents our target in this thesis.

In the following, we focus on microarray data analysis, specially cDNA chips technology. In fact, this five-step analysis procedure is valid for all technologies, including SAGE data. We use cDNA technology as example because it often contains more inherent noise than other technologies, so it is the most general procedure.

The next four sections explain in detail the steps of the gene expression data analysis procedure: statistical data treatment, analysis of differentially expressed genes, class discovery and data interpretation and knowledge discovery. We do not include the first step, data generation, because it has been fully explained in chapter 1 as an independent five-step procedure of microarray experiments (section 1.2).

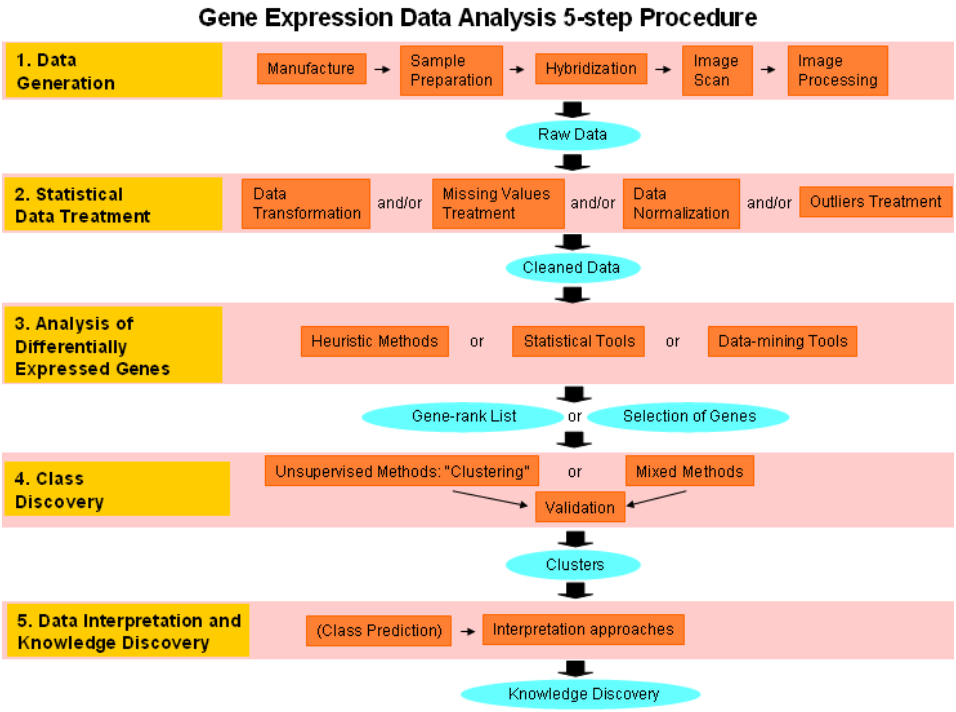

FIG. 2.1: Gene expression data analysis five-step procedure

2.1 Second Step: Statistical Data Treatment

All five sources of variation on the microarray experimentation process introduce systematic bias and errors into intensity measurements. The purpose of statistical data treatment is to remove the effects of any systematic source of variation or bias to the extent possible. In other words, this step receives raw noisy intensities and intends to return well-cleaned gene expression data.

Statistical data treatment involves several statistical data manipulations for data cleaning and preprocessing for obtaining cleaned data ready to be analyzed in further steps. In this section, we explain briefly the four most common issues in microarray data cleaning and preprocessing: data transformation, missing values treatment, outliers treatment and normalization.

As stated in the introduction, the cDNA chips technology was chosen as representative example of the gene expression technologies. The explained issues could be generalized for other gene expression technologies as in-situ oligos chip or SAGE.

2.1.1 Data transformation

It is common practice to transform RT-PCR data from the raw intensities into log intensities before proceeding with data analysis. There are several objectives of this transformation [294]:

- There should be a reasonable even spread of features across the intensity range.

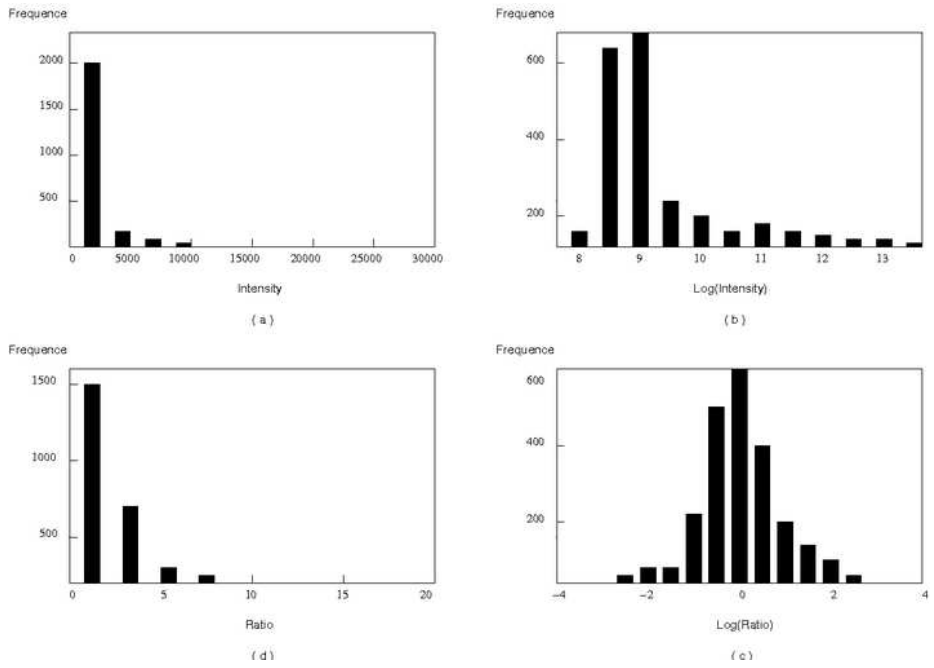

FIG. 2.2: Histogram of the gene expression intensities a) and d) before; and b) and e) after the log transformation of an example data set.

- The variability should be constant at all intensity levels
- The distribution of experimental errors should be approximately zero
- The distribution of intensities should be approximately bell-shaped.

FIGURE 2.2 shows an histogram of intensities of a typical microarray data set before and after log transformation. We can see that the raw data is very heavily arranged together at low intensities and sparsely distributed at high levels. By contrast, the data is more evenly spread over the intensity range after the log transformation. These transformation greatly reduces the skewness of the distribution and simplifies visual examination.

Microarray data analysis typically uses logarithms to base 2 [294]. In processing, the ratio of the raw Cy5 and Cy3 intensities is transformed into the difference between the logs of the intensities of the Cy5 and Cy3 channels. FIGURE 2.2 shows histograms of ratios of the intensity data set before and after log transformation

2.1.2 Missing values treatment

Microarray experiments often generate data sets with multiple missing expression values. Missing values occur for diverse reasons including insufficient resolution, image corruption, or slide contamination by dust, and so on. Missing data may also occur systematically as a result of the robotic methods employed in generating the microarrays [208]. Unfortunately, many algorithms for gene expression analysis require a complete data set as input. Therefore, methods for estimating missing data are needed before these algorithms can be applied.

Suppose a microarray data set represented by a matrix M where each row corresponds to one gene and each column represents an experimental condition. A simple approach to imputing missing values is to replace a missing entry with the average expression over the rows (*row average method*). This method is not optimal since it does not take into account the correlation structure of the entire data set. Troyanskaya et al. in [308] propose two more complex algorithms based on K-nearest neighbors (KNN impute) and singular value decomposition (SVDimpute). They also evaluated the performance of theses two algorithms and the *row average method*.

Imputation based on K-nearest neighbors (KNN): In simple terms, the KNN imputation algorithm estimates missing values by selecting K genes with expression profiles most similar to the gene of interest. Suppose that, for gene i, the expression value $x_{i,j}$ is missing in the jth experiment. The algorithm selects the K genes with non-missing values for experiment j which have closest expression profiles to gene i in the remaining experiments. A weighted average of values in experiment j from the K genes is then used as an estimate for $x_{i,j}$. In [308], the authors found that the euclidean distance was a sufficiently accurate measure for the log-transformed data.

Imputation based on Singular Value Decomposition (SVD): This method first imputes all missing values in matrix M using the row average method in a preliminary step. The Singular Value Decomposition (SVD) is then applied to produce a set of mutually orthogonal expression patterns called eigengenes. These eigengenes can be combined linearly to approximate the gene expressions in a nxm microarray data matrix M, where n and m are the number of genes and experiments, respectively. The imputation process involves a regression of the missing value $x_{i,j}$ against the selected k eigengenes (while ignoring all expression values corresponding to experiment j). That is, the missing $x_{i,j}$ is obtained from a linear combination of the k eigengenes weighted by the regression coefficients. This process is iterated until the total change in the matrix A converges to a sufficiently small arbitrary value.

Troyanskaya et al. [308] compared the performance of KNN imputation, SVD imputation and the row average method in terms of both computational complexity and estimation accuracy. They concluded that although row averaging is the fastest method, it does not perform well in terms of accuracy. They recommend the KNN imputation method as the most robust against the increasing fraction of missing data.

However the application of one of these three methodologies, they have to be practiced with caution, specially when drawing critical biological conclusions from data partially imputed. Thus, estimated data should be marked where possible, its significance to the formulation of biological conclusions should be assessed in order to avoid unwarranted assumptions.

2.1.3 Outliers treatment

Extreme values have been a source of debate among the data analysts community. The presence of extreme values in a data set can be due to systematic errors, faults in the experimental conditions, erroneous procedures, areas where a certain theory might not be valid, or it can simply be the case that some observations happen to be a long way from the center of the

data. Furthermore, these values can be taken as a source of contamination in data or it can be seen as a source of interesting information or unusual special events. Thus, it is a crucial data analysis task to interprete and characterize outliers as well as to develop statistical methods to treat them in order to decrease their impact during statistical data analysis [20].

An outlier can be defined in many ways. A statistical definition given by Grubbs [135] is: "An outlier is an observation that appears to deviate markedly from other members of the sample in which it occurs. Munoz Garcia [218] proposes another definition which states an outlier as an observation that deviates clearly of the general behavior compared to the criterion on which the analysis is carried out. Barnett and Lewis state that an outlier is an observation among a set of observations which clashes or is not in harmony with the rest of observations in the set. What characterizes an outlier is its impact on the observer [20].

Outliers treatment methods

A complete survey for concepts, tendencies and methods for treating outliers has been made by Planchon [241]. Here, we present the statistical point of view for treating outliers in relation with a probability model. We define outliers as in Barnett and Lewis and can localize them as the extremes values of a statistical distribution. However, the outliers for an exponential model and a normal model can be different, so they are model dependent.

We will explain the principal methods for treating outliers against a probabilistic model using the formalization of Barnett and Lewis [20]. They have baptized these methods *discordance test methods*.

The goal of discordance test methods is to test the outlier value in order to reject outliers of the whole of the data or to identify them as being a characteristic of a particular interest. Thus, this test is a procedure of detection that allows to decide in favor of the membership of an specific value to the data set or against it.

Supposing an univariate distribution case where the sample of a random variable X, is x_1, x_2,..., x_n. The extremes values are x_1 and x_n, for example, x_n is called an outlier if it is statistically unacceptable, in relation with the distribution of X under any distribution F. When the result of the test indicates that x_n is not acceptable in a statistical way, one can say that x_n is a discordant superior value for the level of the test. In a similar way it can be shown for the inferior value x_1 or even for the couple (x_1, x_n).

There are several discordance test methods, Barnett and Lewis have distinguished seven types of tests:

1. The excess and spreading out statistics, Dixon 1950 [94]
2. The amplitude and spreading out tests.
3. The standard deviation and spreading out test, Grubbs [134], Tietjen [306] and Cochran [77].
4. Extreme values and positions statistics.
5. Least squares statistics.
6. Superior momentums statistics.

7. Shapiro-Wilkson statistic [278] and [261].

For a more extensive list of discordance tests classified by concerned type of distribution F we can see: [60] and [20].

As we have seen in the histogram of transformed intensity measurements in FIG. 2.2, ratio gene expression measures have to be normally distributed, so we are concerned here by normal distributed discordance tests. Among these approaches, we can cite: Rosner's [129], Dixon 1950 [94], Grubbs 1950 [134], Cochran [77] and Tietjen 1972 [306].

2.1.4 Data normalization

Any of the five sources of variation (as seen in the chapter's introduction) on the microarray experimentation process introduce systematic bias into intensity measurements. The purpose of normalization is to remove the effects of any systematic source of variation to the extent possible.

For in situ oligo-chips, normalization allows direct comparison of individual gene expression levels from one chip. For RT-PCR chip, normalization can be applied to adjust the bias among multiple channels.

In general, normalization microarray methods can be divided into *global normalization* schemes and *intensity-dependent normalization* approaches. The global normalization schemes assume that the spot intensities on each pair of chips or channels being normalized are linearly related. Therefore, they can be corrected by adjusting every single spot intensity on the same chip or channel by an identical amount, called the normalization factor. By contrast, the intensity-dependent normalization methods determine the normalization factor for different spots according to their individual intensities. Normalization therefore relies on a nonlinear, intensity-dependent normalization function $X \rightarrow F(X)$ [10].

The reader is referred to the work of Stoyanova et al [295], Zhao et al [337] for alternative case-specific normalization approaches.

Global normalization approaches,

Standardization: Data sets are standardized to ensure that the mean and the standard deviation of each data set are equal. The method is simple; from each measurement on the chip, subtract the mean measurement of the chip and divide by the standard deviation. After this transformation, the mean of the measurements on each chip will be zero, and the standard deviation will be one. An alternative to using the mean and standard deviation is to use the median and median absolute deviation from the median (MAD). This has the advantage of being more robust to outliers than simply using the mean and standard deviation [334].

Iterative Linear Regression: Essentially, this method iteratively performs a linear regression on the given pair of data sets $x_{1,i}$ and $x_{2,i}$. The approach assumes that most genes in two data sets are unchanged. The variation in the data sets is caused by systematic bias and can be described by linear correspondence. For more details in the processing steps of this method see Draghici et al. [99].

Intensity-dependent normalization

Locally weighted linear regression (Lowess): Several reports have indicated that the $log_2(ratio)$ values can have a systematic dependence on the intensity [327, 329]. This most common appears as a deviation from zero for low-intensity spots. Locally weighted linear regression (Lowess) [76] analysis has been proposed as a normalization method that can remove such intensity-dependent effects in the $log_2(ratio)$ values. In essence, lowess divides the data into a number of overlapping intervals and fits a polynomial function of the form:

$$y = a_0 + a_1 x + a_2 x^2 + ...$$

More details about this methodology can be found in [76] and [327, 329]. The effects of the lowess normalization are illustrated in FIG. 2.3. In this plot (called ratio-intensity plot or R-I plot), the horizontal axis represents the sum of the log intensities $log_{10}(Cy3 * Cy5)$ which is the quantity directly proportional to the overall intensity of a given spot. the vertical axis represents $log_2(Cy3/Cy5)$ which is the usual log-ratio of the two samples. Note the strong non-linear distortion in FIG. 2.3a and how this is corrected by lowess in FIG. 2.3b.

Distribution Normalization: While the purpose of Lowess is to correct the mean of the data sets, the objective of distribution normalization is to make the distributions of the transformed spot intensities as similar as possible across the chips. A distribution normalization algorithm was proposed by Bolstad et al [41]. More detail of this method can be found in [98, 32].

Distribution normalization is an alternative to lowess normalization. It is useful where the different chips have different distributions of values. The assumption behind this method is that given a series of chips, a small number of genes may be differentially expressed, however, the overall distribution of spot intensities should not vary too much.

2.2 Third Step: Differentially Expressed Genes

One basic purpose of gene expression technology is to identify those genes which demonstrate a significant change in expression level across different classes of samples or under certain experimental conditions, for example: finding the genes affected by a specific treatment, finding marker genes that discriminate diseased from healthy subjects, or finding the genes that are active in a cancerous tissue. In other words, the goal of the third step is to identify genes that are *differentially expressed* in one set of samples relative to another[9], establishing potentially meaningful correlations between genes and specific biological conditions.

Although simple in principle, the identification of differentially expressed genes can be complex in practice, as there may be multiple experimental conditions or a lack of biological replicates. Typically, early attempts to analyze differentially expressed genes simply established a fixed cut-off k and selected those genes whose expression went through a $k - fold$

[9] In sample preparation step (section 1.2) we have seen that a microarray experiment explores a set of samples generally one relative to another i.e. reference vs test. This samples may vary in relation to the application it can be: disease vs normal state, mutated vs non mutated state, cancer vs. normal or even with more classes as: treated with this narcotics, treated with herbs, treated with placebo etc

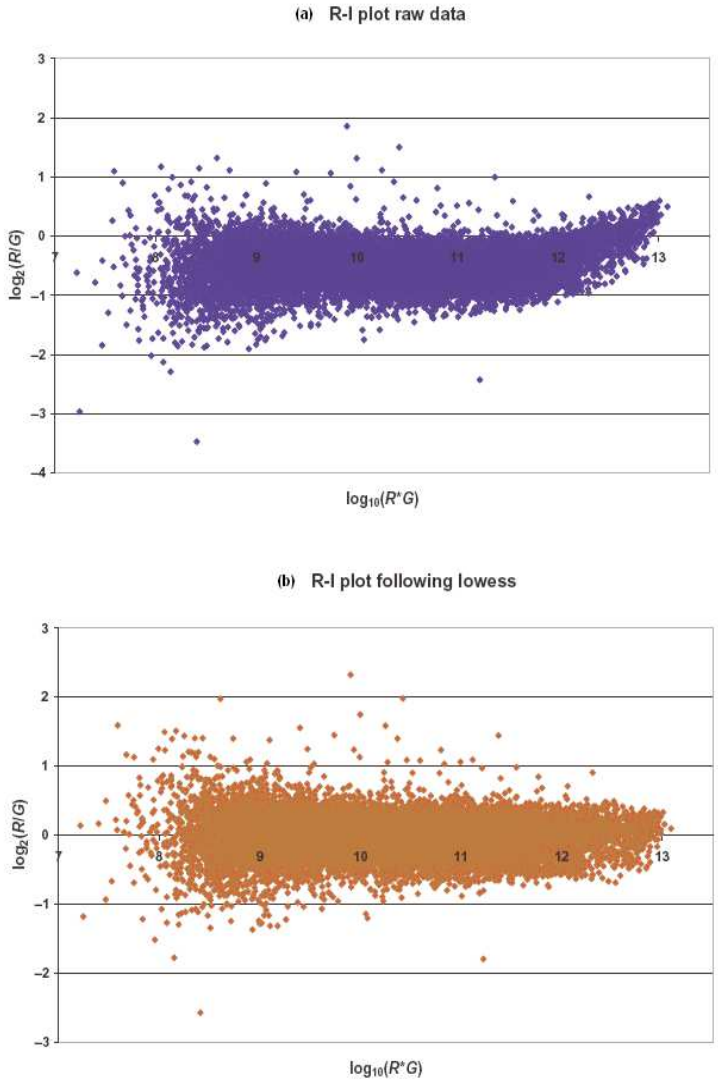

FIG. 2.3: Lowess correction

change [71, 90, 320]. However, the specification of k was often arbitrary, and it did not take into account the overall distribution of the measurements. Several variants of these simple fold change methods have been proposed to fine-tune the approach [174, 270, 271, 301].

When replicates of the samples are available, researchers can turn to some common statistical tests. For instance, the $t-test$ is a standard statistical test for detecting significant change of a variable between repeated measurements in two groups, this can be generalized to multiple groups via the $ANOVA$ F statistic [283]. Many variants of the $t-statistic$ for microarray analysis have been developed [133, 214, 309]. In addition, non-parametric based statistics are also commonly applied [25, 101, 235, 333].

Regardless of the specific approach, the significance of the statistical measure must be determined. A microarray data set typically consists of thousands of genes, and the significance test will be carried out for each gene. A drawback of this multiple testing is the increased probability of observing a false positive, which rises with the number of statistical test performed [42]. Therefore, when multiple tests are involved, methods should be applied to correct the significance level of the individual test.

This section assumes an understanding of basic statistical approaches as statistical inference, hypothesis testing and parametric and non-parametric statistical tests, the reader can see more details of this statistical issues in [113, 77, 129, 264]. However, we include in the first subsection an overview of the basic statistical definitions in these issues, as well as a general nomenclature framework for applying them in microarray data analysis.

The different statistical methods for detecting differentially expressed genes - fold change methods, parametric test, and non parametric test - will be discussed in subsections 2.2.2, 2.2.3 and 2.2.4 respectively. The problems associated with multiple testing and the available correction methods will be discussed in section 2.2.5. Finally, we introduce ANOVA in section 2.2.6.

2.2.1 General framework of statistics in microarray data analysis

In this subsection we will use four sample data sets to illustrate the problems of identifying differentially-expressed genes under various experimental design conditions.

Example data set A: Samples from human T cells grown at $37°C$ (control samples) and $43°C$ (to explore the influence of heat shock). The expression levels of 1,046 genes were monitored by cDNA chips to identify heat-shock regulated genes in human T cells. This data set is an example of *paired data* there are two related measurement for a sample: control and exposed to the shock. We are interested in the difference between the two measurements, as expressed by log ration, to determine whether a gene has been up-regulated or down-regulated by exposure to heat shock.

Example data set B: Samples from 14 multi sclerosis (MS) patients. The expression levels of 4,132 genes were measured by cDNA chips for each patient prior to and 24 hours after $interferon-\beta$ ($IFN-\beta$) treatment [222]. This data set is also paired data. However, unlike example data set A, this data set contains 14 biological replicates. Each replicate presents two related measurements corresponding to pre and post-treatment conditions. here, we wish to identify genes that were differentially expressed in multiple sclerosis following treatment.

Example data set C: Samples from 15 MS patients and 15 age and sex-matched controls [222]. The expression profiles of 4,132 genes were measured by cDNA chips. This data set is an example of unpaired data. it contains two groups of individuals (MS and Controls), our goal is to observe whether a gene is differentially expressed between the two groups. Unlike example data set B, there is not inherent relationship between the individuals in the two groups.

Example data set D: This data set is the union of example data sets B and C and contains three groups: MS, Controls, and $IFN-\beta$ treatment individuals. This data set is an example of multi-group data. Here, we intent to identify genes that are differentially expressed in one or more of these three groups.

Nomenclature

Let assume gene expression measures are presented as a matrix of m *samples* or biological conditions (columns), each sample contains n genes (rows), where $X_{i,j}$ is the expression measure of gene i in sample j. $X_{i,j} \in \mathbb{R}$, so it's continuous in all real numbers.

In statistical context, the term population denotes the entire collection of individuals or objects about which information is desired [91]. Rather, we can take a subset, called a sample, of the total population. In the case of example B, this sample contains 14 patients, which we hope (statistically) will be representative of the entire MS population.

Statistical Inference

The readout of a microarray experiment can be represented by random variables. For example, the expression level of a specific gene i in MS patients before and after $IFN-B$ treatment can be represented by two random variables x_i and y_i, respectively. A random variable does not describe the actual outcome of a particular experiment. Instead, it associates the possible but undetermined outcomes with a *probability distribution*. The probability distribution of a random variable can often be characterized by some parameters. For example, the mean μ_i of x_i is a parameter of the probability distribution of x_i.

Unfortunately, in most cases, the entire population is not available for analysis, so the actual value of the parameters remains unknown to the experimenter. however we may gain some insights into a parameter of interest by applying a numerical descriptive measure, called a statistic, to the sample. For example, we can calculate the average intensity value $\bar{x}_i$ of gene i of patients before treatment. We intend to estimate the parameter value μ_i through the *statistic* value $\bar{x}_i$. This generalized procedure is called *statistical inference*. That is, we hope to generalize our result from the small sample set of 14 patients to the entire population of MS patients.

Hypothesis Test

The problem of determining whether a gene is differentially expressed can be approached by a classical statistical procedure called the *hypothesis test*. The procedure of a hypothesis test involves the following steps:

1. Define the problem
2. Generate the hypotheses
3. Choose an appropriate statistic.

4. Calculate the statistic value based on the observed data.
5. Calculate the corresponding $p-value$ and specify the *significance level.*.
6. Reject or not reject the *null hypothesis* based on the calculated $p-value$ and the pre-specified significance level.

The first step is to define the problem. In example B, we may expect that gene i to be up or down regulated after the patient has been treated and we want to determine whether the observed data support this hypothesis.

The second step is to generate the hypotheses. These two hypotheses should be *mutually exclusive* and all *inclusive* [97]. One of the postulated hypothesis will be the named null hypothesis, H_0, which is a claim about a population characteristic that is initially assumed to be true. The other hypothesis will be the alternative hypothesis, H_A or H_1, which is the competing claim. We then consider the evidence (observed sample data), and we only *reject* the null hypothesis in favor of the competing hypothesis if there is convincing evidence against the null hypothesis [91].

The third step is to choose an appropriate statistic. Taking the example of testing whether gene i is differentially expressed, the null hypothesis is $H_0 : \bar{x}_i = \bar{y}_i$ and the alternative hypothesis $H_1 : \bar{x}_i \neq \bar{y}_i$, where $\bar{x}_i$ and $\bar{y}_i$ are the statistics refer to mean expression levels of two groups of samples, respectively.

The fourth step would be to calculate $\bar{x}_i$ and $\bar{y}_i$ based on the observed data of the 14 patients before treatment and after treatment respectively.

In order to answer to the fifth step we need to know the hypothesis testing error-prone management (illustrated in TABLE 2.1). A *Type I error* involves the rejection a null hypothesis when it is in fact true, its counterpart, the *Type II error*, refers to not rejecting H_0 when is in fact false. The probability of a Type I error is usually denoted by α, while the probability of a type II error is denoted by β [91]. Clearly, $1-\alpha$ corresponds to the probability of "true negatives", while $1-\beta$ corresponds to the probability of "true positives". All four possible outcomes of hypothesis testing are summarized in TABLE 2.1.

	Truth	
Decision	H_0 is true	H_0 is false
H_0 was rejected	false positive (Type I error) : α	true positive (correct decision): β
H_0 was not rejected	true negative (correct decision): $1-\alpha$	false negative (Type II error): $1-\beta$

TABLE 2.1: Hypothesis testing errors

The *significance level* is the probability of a Type I error [91]; simply stated, it is the quantity of errors we are prepared to accept in our studies. In the other hand the $p-value$ (sometimes called the observed significance level) is the probability, assuming that H_0 is true, of obtaining a test statistic value at least as contradictory to H_0 as what actually resulted [91]. In other words, it is the observed probability of wrongly rejecting the null hypothesis when it is actually true. Small $p-values$ suggest that the null hypothesis is unlikely to be true. In other terms, if the $p-value$ is smaller that the significance level, the null hypothesis will be rejected.

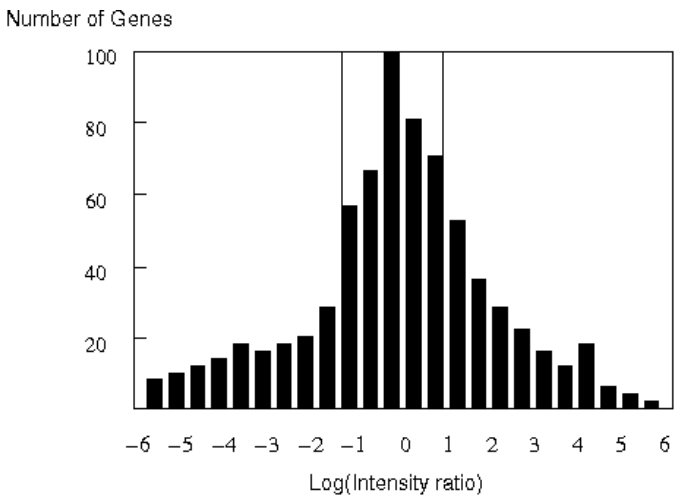

FIG. 2.4: Histogram of log ratios and selection of genes with 2-fold change($log_2 2 = 1$)

2.2.2 Fold change methods

K-fold change

In general, the fold change for a gene is calculated as the average expression over all samples in a condition divided by the average expression over all samples in another condition. Using the fold change method, finding the genes that are differentially expressed can be done by simply considering those genes which demonstrate a significant change between the experiment samples of particular interest (as cancerous samples) and controls. this approach is mot suitable for data sets without biological duplicates (as data set A).

Typically, an arbitrary threshold such as a two or three fold-change is chosen, and the difference (in log form) is considered to be significant if it is larger than the threshold (e.g., [69, 90, 320]). to facilitate the selection process, the ratio between the two expression levels for each gene is first calculated. Since most genes in a typical microarray experiment do not change, the ratios between experiment samples of particular interest and controls of most genes will be around one, and their logs will be around zero.

The experiment contained the disease/control ratios can be plotted into a histogram (as seen in FIG. 2.4). The horizontal axis of FIG. 2.4 represents the log ratio values. Using this histogram, selecting differentially expressed genes based on fold change corresponds to setting thresholds(vertical bars) at the desired minimum fold change and selecting the genes in the tails of the histogram [97].

Unusual ratios

This method considers the distribution of measurements within the data. Instead of blindly specifying the value of $k - fold\ change$, this method involves selecting those genes with experiment to control ratios at a specified distance from the mean experiment to control ratio

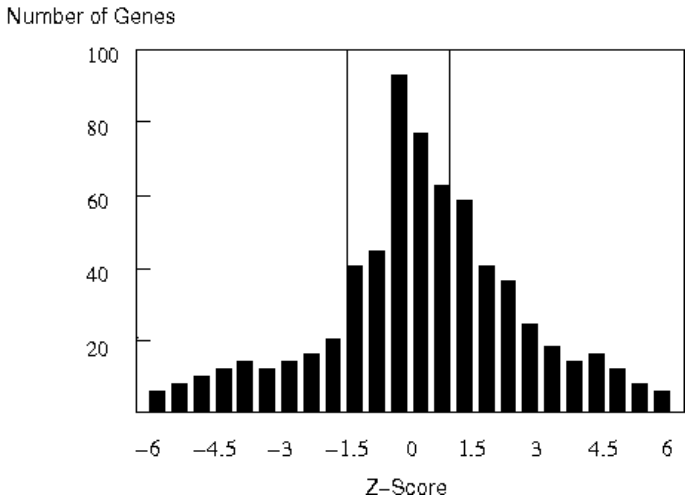

FIG. 2.5: Histogram of standardized log ratios and selection of genes with unusual ratios $\pm 1.5\sigma$

[270, 271, 301]. For example, this distance can be taken to be $\pm 1.5\sigma$ where σ is the standard deviation of the ratio distribution.

In practice, selecting genes $\pm 1.5\sigma$ away from the mean can be accomplished by standardizing the ratios and plotting them in a histogram (see FIG. 2.5) Since the standardized data will have a mean of zero and a standard deviation of one a histogram of the standardized values will be centered around zero, and the units on the horizontal axis will represent the standard deviation. Therefore, setting thresholds at $\pm 1.5\sigma$ will correspond to selecting those genes outside the vertical bars in FIG. 2.5.

Compared with the $k - fold\ change$ method, this method has the advantage of automatically adjusting the cut-off threshold. That is the thresholds determined by this method are dependent on the distribution of all ratios in the given data set, allowing a more tailored selection than the uniformed choice of a fixed threshold. However, this method also has an intrinsic drawback, in that the top $k - percent$ of most affected genes will always be selected, regardless of the number of genes regulated or the extent of regulation[98, 97].

Model-based methods

Here, we will describe briefly a model based approach [174]. for selecting differentially-expressed genes. In a model-based approach, two events are considered: E_g represents the event that gene g is expressed while $\bar{E}_g$ represents event that g is unexpressed. Let p denote the prior probability of E_g, then $1-p$ is the prior probability of $\bar{E}_g$. The model-based method assumes that the expressed genes and the unexpressed genes follow two probability distributions, respectively: the expressed genes are associated with pE_g, while the unexpressed genes are associated with $1 - p\bar{E}_g$. An observed expression y may rise from either of the two distributions.

The purpose is to determine whether the observed ratio y arises from an expressed gene of an unexpressed gene. So, we estimate the likelihood that gene g is expressed given that the observed ratio of g is y; i.e., we calculate the conditional probability $Pr(E_g \mid Y_g = y)$. From Bayes theorem, the conditional probability can be expressed as:

$$Pr(E_g \mid Y_g = y) = \frac{p * pE_g(y)}{p * pE_g(y) + (1-p) * p\bar{E}_g(y)}$$

To simplify the problem, this method assumes the normality of $pE_g(y)$ and $p\bar{E}_g(y)$, with equal variance σ. In this case the mixture model can be completely characterized by four parameters: p, σ, μ_{E_g} and $\mu_{\bar{E}_g}$. These parameters can be estimated by a maximum likelihood approach, called the EM algorithm [88]. EM algorithm searches various combinations of the parameters and converges to a local maximum-likelihood parameter setting.

Draghici [98] noted that the model-based method offers a number of advantages over the fold change and unusual ratio approaches discussed previously. Here, the maximum-likelihood estimators (MLE) become unbiased minimum-variance estimators as the sample size increases. However, the disadvantage of the maximum-likelihood estimate approach is that the results quickly become unreliable as the sample size decreases. Moreover, MLE estimates can become unreliable when the data deviate considerably from normality [98].

It should be noted that these three fold change methods: model-based, k-fold change and unusual ratio approaches are all best suited to data sets without replication (such as example data set A).For data sets with replication, the test discussed in the following sections are usually more appropriate.

2.2.3 Parametric tests

The usual method for performing an hypothesis test on data of the type exemplified by data sets B and C is $t-test$. This test was developed by W. S. Gosset [1876-1937] and was originally termed the "student's t test". Here, we explain briefly two versions of this test:*paried t-test* and *unpaired t-test*, that are applicable to data sets B and C respectively. In addition, several variants on the classical t-statistic have been proposed.

Paired t-test

The paired $t-test$ is applicable to paired data; e.g., data sets in which each data point has a pair of observations (As in example B). Here the null hypothesis is that the gene is not differentially expressed, or the mean μ of the log ratios, $log_2(\frac{x_1}{x_2})$, equals to 0, denoted by $H_0 : \mu = 0$. From the observed log ratios, we can use the following formula to calculate the $t-statistic$:

$$t = \frac{\bar{x}}{s\ /\sqrt{n}},$$

where $\bar{x}$ is the average of the log ratios, s is the standard deviation, and n is the number of the patients in the experiment. A $p-value$ can then be obtained by looking up a $t-distribution$ with $n-1$ degrees of freedom. Finally, the null hypothesis is rejected or not rejected based on the $p-value$ and a pre-specified significance level.

The t-test is more sophisticated than the fold change methods. The significance of differentially expressed genes depends not only on the average log ratio but also on both the population variability and the number of individuals in the study [294]. In general, the accuracy of the determination of the differentially-expressed genes increases with the number of individuals in the experiment.

Unusual ratio method and t-test differs fundamentally. In the ratio method, the entire set of genes is regarded as the sample set, and the most-changed genes in this sample set are considered to be differentially expressed. In contrast, the t-test takes the group of all patients as its sample set. Indeed, a conclusion based on the application of a t-test to multiple patients (biological replicates) may often be more reliable than the results of the unusual ratio method.

Unpaired t-test

The paired $t-test$ is applicable to unpaired data; e.g., where there are two unrelated groups of patients (As in example C). Here the null hypothesis states that the means of the expression levels of a given gene in the two samples will be equal: i.e., $H_0 : \mu_1 = \mu_2$. Unpaired t-test may be $equal-variance$ and $unequal\ variances$. As suggested these names, the first t-test assumes that the two samples are taken from distribution with equal variances, while the second test assumes that the two distribution have different variances. Both of these test, use the following formula to calculate the $t-statistic$:

$$t = \frac{\bar{x}_1 - \bar{x}_2}{\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}}, \tag{2.1}$$

where $\bar{x}_1$ and $\bar{x}_2$ are the means, s_1^2 and s_2^2 are the variances, and n_1 and n_2 are the sizes of the two groups, respectively. For more details in the differences of the two kinds of unpaired t-test with equal and unequal variance, the lector can be [332].

To determine whether the variances in the two distributions are equal. This can be established through the use of another hypothesis test, where the hypotheses are $H_0' : \sigma_1^2 = \sigma_2^2$ and $H_1' : \sigma_1^2 \neq \sigma_2^2$. To test the null hypothesis, the F statistic can be used, as follows:

$$F = \frac{s_1^2}{s_2^2}.$$

A $p-value$ can then be obtained on the basis of $F-statistic$ distribution with respect to the degrees of freedom, $\upsilon_1 = n_1 - 1$ and $\upsilon_2 = n_2 - 1$; this will indicate whether H_0' should be rejected.

Once we have tested for equality or inequality of variances, we applied equation 2.1 to calculate the t-test, taking into account the equality or inequality of variances to calculate de degrees of freedom of the distribution. Then, we obtain the correspondent $p-value$ on the basis of the $t-statistic$ and the corresponding degrees of freedom of the $t-student$ distribution to determine whether H_0 should be rejected or not.

Variants of t-test

Additionally to the classical t-statistics explained before, several simplified forms are also available for the identification of differentially-expressed genes.

Golub et al. [133] have proposed a method called *neighborhood analysis*. In this method, given the expression levels of gene g over all the experimental conditions, the following score is calculated:

$$P(g) = \frac{\mu_1(g) - \mu_2(g)}{\sigma_1(g) + \sigma_2(g)}, \tag{2.2}$$

where $\mu_1(g)$ and $\mu_2(g)$ are the means of the expression levels of gene g in classes 1 and 2 , respectively, and $\sigma_1(g)$ and $\sigma_2(g)$ are the standard deviation of g in classes 1 and 2, respectively. Large values of $|P(g)|$ indicate a strong correlation between gene expression and class distinction, while a positive or negative $P(g)$ indicates that g is more highly expressed in class 1 or class 2, respectively.

In another t-test variant, Pavlidis et al. [231] have adapted the *Fisher's discriminant criterion* (FDC) to define a score as:

$$F(g) = \frac{(\mu_1(g) - \mu_2(g))^2}{\left(\sigma_1^2(g) + \sigma_2^2(g)\right)^2}. \tag{2.3}$$

Genes with higher score values are selected as differentially expressed genes. Thus, equations 2.2 and 2.3 are similar to the $t - statistic$ formula and considered to be variants of that method.

2.2.4 Non parametric tests

The $t - statistic$ and its variants start from the assumption that the data will follow a normal distribution. However, the distribution of intensities of many genes may not be normal in a real data set [89]. As a result, using $p - values$ obtained from the$t - distribution$ as a test of gene expression may be meaningless in these instances.

In this subsection, we describe several non-parametric methods which do not place any assumptions on the observed data. These non-parametric methods do not rely on the estimation of parameters (such as the mean or the standard deviation) in describing the distribution of the variable of interest in the population.

Classical non-parametric statistics

There are non-parametric equivalents of both the paired and unpaired t-tests. The Wilcoxon sigh-rank test is the non-parametric equivalent of the paired t-test, while the Wilcoxon rank-sum test (also called Mann-Whitney test) is the non-parametric equivalent of the unpaired t-tests [294]. As we have noted in the previous subsection, the unpaired t-test is actually a generalization of the paired case. Therefore, we will focus here only on the Wilcoxon rank-sum test.

The Wilcoxon rank-sum test [318] organizes the observed data in value ascending order. Each data item is assigned a rank corresponding to its place in the sorted list. These ranks,

rather than the original observed values, are then used in the subsequent analysis. The major steps in applying Wilcoxon rank-sum test are as follow:

1. Merge all observations from the two classes and rank them in value-ascending order.
2. Calculate the Wilcoxon statistics by adding all the ranks associated with the observations from the class with a smaller number of observations.
3. Find the p-value associated with the Wilcoxon statistic from the Wilcoxon rank sum distribution table [149].

The use of rank-based tests of this type is appropriate when the underlying distribution is far from normal. Moreover, the rank-sum test is much less sensitive to outliers and noise, typical characteristics of gene-expression data sets, than are parametric test [89, 294]. Counterbalancing these benefits is the relative lack of sensitivity of the rank-sum tests in comparison with their parametric counterparts. Rank-sum $p-values$ tend to be higher, increasing the difficulty of detecting real differences as statistically significant [10, 294].

Other non-parametric statistics

In addition to the classical rank-sum test, several other non-parametric statistics have also been proposed. Ben-Dor et al. [25] use a threshold number of misclassification or TNoM score to select differentially expressed genes. This method assumes that a differentially-expressed gene will exhibit significantly different values in the two classes and that the values can therefore be differentiated by a threshold number. Gene values which are more clearly separated by this threshold are more likely to arise from an up-down regulated gene. Given the expression values $\vec{g}$ of gene g over all the experimental conditions, the TNoM score is defined as follows:

$$TNoM(\vec{g}) = \min_{d,t} \sum_{i} 1\left\{l_i \neq sign(d * (g_i - t))\right\}, \tag{2.4}$$

where g_i is the expression level of g in the $i-th$ experimental condition, l_i is the class label of the ith condition. $d \in \{+1, -1\}$ is used to indicate the class label, and t is the threshold to separate the expression values of g. The term $sign(d * (g_i - t))$ is called a "decision stump" which indicates the predicted class label based on d, g_i and t. The basis is that the sign of $d * (g_i - t)$ is dependent on whether the expression level of gene g in condition i is greater than the threshold value t.

Equation 2.4 seeks the best decision stump for a given gene and then counts the classification errors this decision stump makes in differentiating known class labels. Fewer errors indicate that the threshold is more successful in differentiating the two classes, and, in turn, that it is more likely that the gene is up or down-regulated. Like the classical non-parametric statistics, this method does not rely on any assumptions regarding the observed data.

2.2.5 Bootstrap analysis

In a first step, bootstrapping analysis is similar to any of the classical parametric tests; for example, this analysis can begin with a calculation of the t-statistic. In contrast, in a second

step, rather than determining the $p-value$ on the basis of the standard t-distribution (which is tabulated under the assumption of normal distribution), bootstrap analysis uses a resampling strategy to approximate the real distribution of the t-statistic.

In more detail, the bootstrap method constructs a large number of random data sets by resampling form the original data. That is, each data entry $x_{i,j}$ (measurement of gene i under experimental condition j) is randomly assigned one of the measurements from the data set. The resulting data sets resemble the original data in their values. However the correlation between genes and samples in the original data is completely disturbed through the randomization procedure.

The next step of the bootstrap method involves calculating the t-statistics for all the genes in each random data set an using the standard t-distribution to find the minimum $p - value$ among the genes. The outcome of this process is an adjusted $p - value$ for the original data set.

Bootstrap analysis is based on the concept that, it the H_0 is true, then the real (observed) data set would exhibit characteristics similar to any of the randomized data sets. In other words, the value of the selected statistic T (and thus the $p-value$) calculated from the real data would appear as a typical value in the distribution of T (and the $p - value$) from the randomized data sets. Conversely, if the value of T (and the $p - value$) from real data is "significantly abnormal", then we may be confident that the observed data are not formed by chance, and the null hypothesis should be rejected.

Comparing to other methods discussed before, bootstrap analysis has several significant advantages. As a non-parametric test, bootstrap analysis does not require that the data be normally distributed and is robust to noise. Furthermore, bootstrap analysis is more sensitive and accurate that the classical non-parametric tests, as it is able to take into account the errors arising from "multiple testing" (discussed in detail in the next subsection) Finally this method can be used with any statistical measure. That is, we can choose any statistic and evaluate its p-value using the resampling strategy. Therefore, bootstrap analysis is more appropriate for use with microarray data than either the t-test or classical non parametric tests [334].

Multiple testing

To select differentially-expressed genes, we usually apply the hypothesis test gene by gene. In practice, a microarray experiment typically involves thousands of genes. This means we have to repeatedly run the test for thousands of times. A problem with doing so many test is that the number of false positive may be increased, a phenomenon called *multiple testing* in statistics. In other words, we could make the Type I error and report false positives due to random effects. According to the definition of significance level (subsection before), the probability of committing a Type I error is exactly α. So the probability of not making a Type I error would be $1 - \alpha$. Suppose we have N genes in the data set, the probability of make correct decisions for all genes is:$Prob(globally\ correct) = (1 - \alpha)^N$, and the probability of making at least one mistake $Prob(wrong\ somewhere) = 1 - (1 - \alpha)^N$. When α is small,

the expected number of false positives is αN. For a very large N, the number of false positives may be large.

They exist four axes of approaches that deal with multiple testing: *family-wise error rate* (FWER), *false discovery rate* (FDR), *permutation-based* and one of the best known in bioinformatics community *significance analysis of microarray data* (SAM).

The principal idea in **FWER** approaches is to control the *global* significance level and the error rate of multiple test. However, these methods, are often too conservative and result in too many false negatives. They exist several FWER methods, we can mention Sidak correction for multiple comparisons (see [65]), bonferroni correction (see[42, 43]) and Holm's step wise correction ([150]).

An alternative approach, **FDR**, was proposed by Benjami and Hochberg ([28]) to control the false discovery rate (FDR) instead of the FWER. The basic idea of this approach is to control the proportion of significant results that are in fact Type I errors. However, FDR assumes all the genes in the microarray are independent, which is usually not true in reality.

FWER and FDR approaches control the global rate of false positives from different perspectives. However, neither of the approaches consider the possible correlation among data objects. For microarray data analysis this problem is particularly important since genes are often highly correlated. For example, a group of genes may participate in the same pathway. **Permutation-based** approaches take into consideration the possible correlation among genes by adjusting the $p - value$ based on the resampling theory. In [317], Westfall and Young propose a step-down correction (W-Y approach) that adjusts the $p - value$ with the consideration of the possible correlation. More details and an example of applying this method to microarray data can be found in [103]. Disadvantages of this approach is that is computationally intensive and thus very slow [98]. Also is founded in an empirical process lacking the elegance of a more theoretical approach.

Tusher et al [309] reviewed several approaches to adjusting $p - values$ for multiple testing. To address some of the defaults of the approaches cited before in microarray, they proposed the Significance Analysis of Microarrays (SAM). Basically, **SAM** assigns a score to each gene according to its change in gene expression. Genes with scores greater than a threshold are considered as "potentially" significant. To control the false positives, SAM uses permutation measurements to estimate the false discovery rate (pFDR). The score threshold for genes is then adjusted iteratively according to the pFDR until a set of significant genes have been identified. SAM method has become very popular method for the identification of differentially expressed genes in bioinformatics community. More detail of this method [309].

ANOVA (analysis of variance)

In previous sections, we have described methods for analyzing differentially expressed genes in simple data sets with only two samples. In practice, microarrays are also being used to perform more complex experiments as in example data set D, which contains three groups of samples. Thus, the problem is to identify genes that were differentially expressed on one or more groups relative to the others. There are two possible ways to make these analysis [294]:

- A straight-forward method is to apply an unpaired t-test three times, to each pair of groups in turn; genes that are significant in one or more of the t-tests are then selected. This algorithm was implemented in SAM PACKAGE.
- An alternative method is to use a statistical test that compare all three groups simultaneously and reports a single $p - value$. This method is best known as ANOVA.

Stekel [294] noted that there are two problems with the first method: increasing the false positives (becoming worst when the number of groups increase) and each of the comparisons is not independent of the other, thus it becomes very difficult to interpret the results.

Due to the above problems, we usually adopt the ANOVA strategy for Example data set D. ANOVA, performs an analysis of the data with multiple groups, and returns a single $p - value$ which suggest the level of significance whether one or more groups is different from others. if we assume the variance in gene expression comes from only one source, i.e., the different type of cancers the patients are suffering from, we actually perform the *one-way* ANOVA. Instead, if we consider the variance from multiple sources, e.g., the cancer type and the microarray experiment artifacts, we build a more general ANOVA models which include multiple correlated factors and obtain one $p - value$ for each of the factors separately. Such analysis is called the *multifactor* ANOVA.

In general ANOVA method is based on the calculation of the sum of squares, degrees of freedom, mean square deviation from the mean and F-statistics. As we present only the two different approaches of ANOVA method and its implications in microarray technology, the reader may refer to the work of Zar [332] for a full appreciation and pedagogic explanation of the two types of ANOVA algorithms.

The one-way ANOVA takes the variance in a given data set from a single source. However, the variance can be divided into two parts. First, the measurements of each group vary around their mean, which forms the within-group variance. Second, the means of each group will vary around the overall mean of the data set, which forms the inter-group variance. The essential spirit of the one-way ANOVA is to study the relationship between the inter-group and the within-group variances.

In contrast, the multifactor ANOVA builds an explicit model about the multiple, possibly correlated sources of variance that affect the measurements, and then use the data to estimate the variance of each individual variable in the model. The advantage of multifactor ANOVA is that it takes into consideration multiple sources of variance [97]. Thus, it is possible to distinguish interesting variations, such as gene regulation, from the experiment artifacts, such as differences caused by different chips or two-channels of color etc. However, the application of the multifactor ANOVA requires very careful experimental design. In most cases this requires repeating several chips with various mRNA samples, duplicating individual genes on multiple spots of a single chip, etc. In practice, due to the relatively expensive cost and intensive labor of microarray experiments, replicates are often very limited. Thus, the benefit of multifactor ANOVA may only be received in the future when sufficient replicates are available. This affirmation can be extended to all of the methods described before that needs replicates to be more accurate and high confidence.

2.3 Fourth Step: Classification of the Genes

In microarray data analysis, experts commonly said: "The genes that are co-expressed (exhibiting a common expression profile) code for the same biological function" and even more: "We may use the genes with known function to infer the function of other co-expressed genes for which information has not been previously available" [107, 303]. Furthermore, sometimes researchers wish to identify groups of biological conditions that have similar expression level patterns, and genes that are similar across samples. Thus, the need of methods for grouping data objects - genes or biological conditions - is essential for the genomic science. As seen in chapter 1, the genes are acting together for coding the cellular functions of an organism, so in the majors processes of an organism they interact among each other to code for them.

A variety of conventional and newly-developed clustering algorithms have been used to identify: co-expressed genes, coherent gene expression patterns* and samples with common patterns in gene expression technology data. *Clustering* is the data mining process of grouping objects into a set of disjoint classes, called *clusters*. The objects within a class have high degree of similarity, while objects in separate classes are more dissimilar. In this section, we focus on the classification of the genes by common expression profile or by coherent patterns. In this case, the genes are considered to be data objects, and biological conditions (either samples, time points, etc.) are seen as attributes[10]. After a clustering process has been completed, each cluster can be regarded as a group of co-expressed genes, and the corresponding coherent pattern is simply the centroid of the cluster.

Previous studies have confirmed that clustering algorithms are useful in finding co-expressed gene groups and coherent patterns [107, 303]. The identified gene groups and patterns can further help to understand gene function, gene regulation and cellular processes. Furthermore, co-expressed genes in the same cluster are likely to be involved in the same cellular processes, and a strong correlation of expression patterns between those genes indicates coregulation. Searching for common DNA sequences at the promoter regions of genes within the same cluster allows regulatory motifs specific to each gene cluster to be identified and cis-regulatory elements to be proposed [48, 303]. The inference of regulation through the clustering of gene expression data also gives rise to hypotheses regarding the mechanism of the transcriptional regulatory network [92].

In general, a clustering algorithm relies on some proximity measurement to evaluate the distance or similarity between a pair of data objects (genes) and seeks to optimize a specific object function. In this section, we first introduce several proximity measures which have been widely used with microarray data. Then, three categories of clustering algorithms, partition-based, hierarchical approaches, fuzzy logic approaches and density-based approaches, are described in sections 3.3.2, 3.3.3, 3.3.4 and 3.3.5 respectively. In the last section we discuss cluster validation techniques.

[10] In this section, we use the terms "objects" and "genes" exchangeably, and the terms "attributes", "features", and "experimental conditions" exchangeably.

2.3.1 Proximity measurement for gene expression data

A *proximity measurement* measures the similarity (or distance) between two data objects (genes). A data object or gene X_i can be formalized as a numerical vector $\vec{X_i} = \{x_{i,j} \mid 1 \leq j \leq m\}$ where $x_{i,j}$ is the value of the jth feature (sample or biological condition) for $\vec{X_i}$ and m is the number of samples. The proximity between two genes X_i and X_k is measured by a proximity function of corresponding vectors $\vec{X_i}$ and $\vec{X_k}$.

Euclidean distance

Euclidean distance is a very common used distance measurement. It's basically just the sum of the squared distances of two vector values $\vec{X_i}$ and $\vec{X_k}$. The distance between genes $\vec{X_i}$ and $\vec{X_k}$ in a $m - dimensional$ space is defined as:

$$Euclidean(X_i, X_k) = \sqrt{\sum_{j=1}^{m}(x_{i,j} - x_{k,j})^2}. \tag{2.5}$$

However, for gene expression data, the overall shapes of gene expression profiles are often of greater interest than the individual magnitudes of each sample. To solve this problem, a standardization process (as seen in section 2.2) is usually performed before calculating this distance.

Manhattan distance

Similar to euclidean distance, the Manhattan distance is the sum of the absolute distances of two vectors $\vec{X_i}$ and $\vec{X_k}$. Manhattan distance is given by this formula

$$Manhattan(X_i, X_k) = \sum_{j=1}^{m}|x_{i,j} - x_{k,j}| \tag{2.6}$$

This is a linear version of the Euclidean distance, with similar advantages and disadvantages.

Correlation coefficient

Pearson's correlation coefficient: In contrast to Euclidean distance, which measures the distance (dissimilarity) between two patterns, *Pearson's correlation coefficient* measures the extent to which two patterns are similar with each other. This measure is the most widely used measurement of association between two vectors. For two genes X_i and X_k, the linear correlation coefficient $Pearson(X_i, X_k)$ is given by the equation:

$$Pearson(X_i, X_k) = \frac{\sum_{j=1}^{m}(x_{i,j} - \mu_i)(x_{k,j} - \mu_k)}{\sqrt{\sum_{j=1}^{m}(x_{i,j} - \mu_i)^2}\sqrt{\sum_{j=1}^{m}(x_{k,j} - \mu_k)^2}} \tag{2.7}$$

where μ_i and μ_k are the means for $\vec{X_i}$ and $\vec{X_k}$ respectively. The pearson measurement have it higher value at 1 indicating stronger similarity and it ranges: $Pearson(X_i, X_k) \in (-1, 1)$. From a statistical view, each data object can be regarded as a random variable with m

observations. Pearson's correlation coefficient measures the similarity between two profiles by calculating the linear relationship of the distributions of the two corresponding random variables. The definition indicates that Pearson's correlation coefficient is invariant to linear transformations.

Pearson's coefficient is widely used and has proved effective as similarity measure for gene expression data. However, empirical study has shown that Pearson's correlation coefficient is not robust to outliers [145] and may generate *false positives*, i.e., assigning a high similarity score to a pair of dissimilar patterns. Besides, Pearson coefficient assumes an approximately Gaussian distribution of the points and may not be robust for non-gaussian distributions

Jackknife correlation: This correlation coefficient is a slight variation of pearson's correlation, specially built for data with outliers. It is defined as:

$$Jackknife(X_i, X_k) = \min\left\{\rho_{i,j}^{(1)}, ..., \rho_{i,j}^{(l)}, ..., \rho_{i,j}^{(m)}\right\}, \tag{2.8}$$

where $\rho_{i,j}^{(l)}$ is the Pearson's correlation coefficient of genes X_i and X_k with the lth sample deleted. Using the jackknife correlation avoids the "dominant effect" of single outliers. However, the generalized jackknife correlation, which would involve the enumeration of different combinations of features to be deleted, is computationally costly and is rarely used.

Spearman's rank-order correlation: D'haeseleer [92] has proposed an alternative coefficient to pearson's correlation measurement, named Spearman's rank-order correlation. Spearman coefficient does not require the assumption of Gaussian distribution and is also more robust against outliers that Pearson's correlation coefficient. The rank correlation is derived by replacing the numerical expression level $x_{i,j}$ with its rank $r_{i,j}$ among all time points. For example, $r_{i,j} = 3$ if $x_{i,j}$ is the third highest value among $x_{i,k}$, where $1 \leq k \leq m$.

The principal default of Spearman's rank-order correlation is, as a consequence of ranking, a significant amount of information present in the data is lost. Therefore, on average, Spearman's rank-order correlation coefficient may no perform as well as Pearson's correlation coefficient.

Kullback-Leibler divergence

The Kullback-Leibler divergence or *relative entropy*, is an information-theoretic approach to measuring the distance between two gene expression profiles. In general, the relative entropy between two probability mass functions $u(w)$ and $v(w)$ over the random variable W is defined as:

$$KL(u \parallel v) = \sum_{w \in W} u(w) log \frac{u(w)}{v(w)}. \tag{2.9}$$

Given a random variable W with a true distribution u, the K-L divergence measures the inefficiency of assuming the distribution of W as v. As with Pearson's correlation coefficient, the K-L divergence also regards each gene expression profile X_i as a random variable with m observations. To apply the K-L divergence, a profile X_i is converted to its probability mass function by calculating the fractional contribution of the expression level at each experimental

condition to the sum of expression levels at all conditions; i.e., $u_i(w) = \frac{X_i,w}{\sum_{j=1}^{m} x_{i,j}}$. The K-L divergence always takes non-negative values, an is zero if and only if $u = v$.

Kasturi et al. [163] have used the K-L divergence in conjunction with an unsupervised self-organizing map algorithm (explained in the next subsection). The clustering results from two gene expression data sets were found to be superior to those obtained with the hierarchical clustering algorithm using the Pearson's correlation coefficient [163].

2.3.2 Partition-based approaches

Partition-based algorithms divide a data set into several mutually-exclusive subsets based on certain clustering assumption (e.g., there are k clusters) and optimization criteria (e.g., minimize the sum of distances between objects and their cluster centroids). In other words, partitioning methods seek to minimize some measure of within-group dissimilarity of a fixed number of k groups. This is a combinatorial optimization problem, in most problems the *global optimum* will not be found, and one of the possibly many *local optima* will be instead identified. We can further divide the partition-based methods into five sub-categories: the K-means algorithm and its variations [145, 303, 190, 248, 284], K-medoids and its variations [164], Self Organizing Maps (SOM) an its extensions [145, 169, 298, 307, 206], model-based algorithms [118, 126, 209, 331] and graph theoretical algorithms [26, 140, 326, 276]

K-means and its variations

Under this method, one needs to fix the number of clusters K in advance, then the algorithm partitions the data set into K disjoint subsets which minimize the sum of squared distances from each observation to its cluster center μ_i,

$$O = \sum_{k=1}^{K} \sum_{x_i \in C_k} \|x_i - \mu_k\|^2 , \tag{2.10}$$

where x_i is the gene expression measures vector in cluster C_k, μ_k is the centroid or geometric center (mean of genes) of C_k . Thus, the objective function O tries to minimize the sum of the squares distances of objects from their cluster centers [68].

The K-means method is computationally simple and fast. The time complexity of K-means is $O(l, k, n)$ where l is the number of iterations, k is the number of clusters and n is the number of genes. However, this algorithm has several known drawbacks as a clustering algorithm for gene classifying. First, the number of gene clusters in a gene expression data set is usually unknown in advance. Second, gene expression data typically contain significant noise. The K-means algorithm forces each gene into a cluster, which may cause the algorithm to be sensitive to noise [284].

Recently, several new algorithms [145, 248, 284] have been proposed to overcome some drawbacks of the K-means algorithm. We call the variations of the K-means algorithm, since, in essence, they also are intended to minimize the overall divergence of objects from their clusters centers eq. 2.10. One common characteristic of these algorithms is that they use some thresholds to control the coherence of clusters.

In Ralf-Hewing et al. [248] introduced two parameters γ and ρ, where γ is the maximal similarity between two separate centroids, and ρ corresponds to the minimal similarity between a data point and its cluster centroid. In Heyer et al. [145] constrained the clusters to have a diameter no larger than a threshold. Furthermore, Smet et al. [284] proposed a more efficient algorithm in which each gene x_i is assigned to cluster C_k if the assignment has a higher probability than a threshold.

The $K-means$ algorithm and its variations require the initial stipulation of some global parameters such as the number of cluster or a coherence threshold. The clustering process is like a "black box"; users input the data set an the parameter values, and the cluster are generated. There is no intensive interaction between the user and the mining procedures. While this simplifies processing, they are not sensitive to the local structures of the data set and provide no opportunity to exploit user domain knowledge of the data set.

K-medoids and its variations

$K-means$ algorithm takes averages of points assigned to each cluster to define cluster centroids. Such a centroid may have little interpretative value in some problem such a when some variables are categorical or discrete. In these situations, it is more meaningful for each cluster centroid to be a representative object (i.e. the observed data points). The medoid of a cluster of points is the point with best average dissimilarity to all other points. $K-medoids$ is equivalent to $K-means$ algorithm, but it uses the dissimilarity matrix instead of data matrix means [68]. Using the medoid instead of a mean is more robust to missing values and outliers, but it conserves $K-means$ drawback of fixing a priori the number of clusters.

Kaufman and Rousseeuw have proposed a series of algorithms that solves the same problem as $K-medoids$ with some calculation differences: partitioning around medoids (PAM), clustering large Applications (CLARA) and randomized CLARA (CLARANS) [164]. PAM algorithm first finds an initial set of medoids and then exchange points so that no single switch of an observation with a medoid will decrease the objective function. CLARA algorithm it draws multiple samples of the data set, applying PAM on each sample, and giving the best clustering as the output. CLARANS draws sample of neighbors dynamically, when it founds a local optimum, CLARANS stars with new randomly selected point in search for a new local optimum. The principal drawbacks of these methods are: PAM is not robust for microarray large data sets containing thousands of genes; CLARA it depends on the sample size and is based in the sample that will not necessary represent the whole data set (can be biased); CLARANS is computationally inefficient and has to be improved [110]. Indeed, this three variations of k-medoids have been used in many microarray studies[11].

SOM and its extensions

The Self-Organizing map (SOM) was developed by Kohonen [169] on the basis of a single layered neural network. The data objects (genes) are presented as input one at a time. The output neurons are organized with a simple neighborhood structure such as two-dimensional

[11] PAM, CLARA, CLARANS are in the clustering package of open source for bioinformatics bioconductor: http://www.bioconductor.org/

$m * q$ grid (see [298] for a visual picture of this structure). Each output neuron of the neural network is associated with a m-dimensional reference vector, where m is the dimensionality of the input data objects (genes).

In the learning process, each input data point is "mapped" to the output neuron with the "closest" reference vector. Reference vectors (output nodes), which are in some neighborhood of the winning node, are updated by moving them toward the input pattern. For SOM learning each data object acts as a training sample which directs the movement of the reference vectors towards the denser areas of the input vector space. As a result, reference vectors are trained to fit the distributions of the input data set. When the training is complete, clusters are identified by mapping all data points to the output neurons.

One of the remarkable features of SOM is that it allows users to impose a partial structure on the clusters, and arranges similar patterns as neighbors in the output neuron map. This feature facilitates easy visualization and interpretation of the clusters and thus partly supports explorative analysis of gene expression patterns. Tamayo et al. [298] has applied SOM method in the context of microarray data, with a two-dimensional grid. They propose a grid-structured summary of the cluster represented by each prototype. Each summary is typically a plot of expression levels of a prototype gene across the different experiments.

Recently, several new algorithms [307, 144, 206] have been proposed based on the SOM algorithm. These algorithms can automatically determine the number of clusters and dynamically adapt the map structure to the data distribution. For example, Herrero et al. [144] extended the SOM by a binary tree structure. At first, the tree only contains a root node connecting two neurons. After a training process similar to that of the SOM algorithm, the data set is segregated into two subsets. The neuron with less coherence is then split into two new neurons. This process is repeated level by level until all neurons in the tree satisfy some coherence threshold. Other examples of SOM extensions are Fuzzy Adaptive Resonance Theory (Fuzzy Art) [307] and supervised Network Self-Organizing Map (sNet SOM) [206]. In general, they provide some approaches to measuring the coherence of a neuron (*vigilance criterion* in [307] and *grow parameter* [206]). The output map is adjusted by splitting existing neurons or adding new neurons to the map until the coherence of each neuron in the map satisfies a user-specified threshold.

SOM is a efficient and robust clustering technique. A hierarchical structure can also be built on the basis of SOM; one example is SOTA (Self Organizing Tree Algorithm) [95]. Moreover, by systematically controlling neuron splitting, SOM can easily adapt to the local structures of the data set. However, the current versions of SOM require that the splitting process be controlled by user-specified coherence threshold, something which is difficult to identify with gene expression data. Furthermore, SOM requires a user to prespecify the number of clusters, something which is typically unknown in gene expression data.

Model-based clustering

Model-based clustering approaches [118, 126, 209, 331] provide a statistical framework to model the cluster structure of gene expression data. The data is assumed to come from a finite mixture of underlying probability distributions, with each component corresponding to

a different cluster. Suppose for the *ith* observation γ_i^k gives the true, but unknown, k group level for that observation. Then letting $f_i(x_i \mid \theta_k)$ denote the conditional density function for a typical observation x_i from group k, where θ_k denotes an unknown parameter, the resulting likelihood of genes with expression profiles $x_1, x_2, ..., x_n$ is given by

$$L(\gamma, \theta) = \prod_{i=1}^{n} \sum_{k=1}^{K} \gamma_i^k f_i(x_i \mid \theta_k), \tag{2.11}$$

where the parameters $\gamma = \{\gamma_i^k \mid 1 \leq k \leq K, 1 \leq i \leq n\}$ and $\theta = \{\theta_k \mid 1 \leq k \leq K\}$. The unknown group levels γ_i^k are obtained by the method of maximum likelihood that is the method that maximizes the function L jointly in γ and θ. Usually the parameters γ and θ are estimated by the EM algorithm [88]. See more details in Banfield et al. [18].

Several early studies, including [118, 126, 331], impose a model of multivariate Gaussian distributions on gene expression data. Although the Gaussian model works well for gene-sample data (where the expression levels of genes are measured under a collection of samples, it may not be effective for time-series data (where the expression levels of genes are monitored over a continuous series of time points). Because the Gaussian model treats the time points as unordered, static samples and ignores the inherent dependency of the gene expression levels over time.

We can mention at least two models [251, 272] that have been introduced gene expression dynamics of time-series data into model-based algorithms. In Ramoni et al. [251] assumed that the time-series follow an autoregressive model, where the value of the series at time t is linear function of the values at several previous time points. Schliep et al. [272] proposed a restricted *hidden Markov model* to account for the dependencies in time-series data.

An important advantage of model-based approaches is that they provide an estimated probability γ_i^k that data object x_i will belong to cluster C_k. However, gene expression data are typically "highly-connected"; there may be instances in which a single gene has a high correlation with two or more different clusters. Thus, the probabilistic feature of model-based clustering is particularly suitable for gene expression data. However, model-based clustering relies on the assumption that the data set fits a specific distribution, which may often not be the case.

Graph theoretical algorithms

Given a data set D, we can construct a *proximity matrix* P, where $P[i, j] = proximity(X_i, X_j)$ and a weighted graph $G(V, E)$, called a proximity graph, where each data point corresponds to a vertex. For some clustering methods, each pair of genes is connected by an edge, with weight assigned according to the proximity value between the objects [276, 326]. For other methods, proximity is mapped only to either 0 or 1 on the basis of some threshold, and edges only exist between genes i and j, where $P[i, j] = 1$ [26, 140].

Graph-theoretical clustering techniques are explicitly presented in terms of a graph, thus converting the problem of clustering a data set into such graph theoretical problems as finding the minimum cut or maximal cliques in the proximity graph G.

We'll take the algorithms clustering affinity search technique (CAST) [26] and cluster Identification via Connectivity Kernels (CLICK) [276] as remarkable examples of graph theoretical algorithms.

CAST: Cluster affinity search technique: Ben-Dor et al. [26] introduced the concept of a *corrupted clique graph* data model. The input data set is assumed to come from the underlying cluster structure by "contamination" by random errors caused by the complex process of microarray experimentation. Specifically, it is assumed that the true clusters of the data points can be represented by a *clique graph* H, a disjoint union of complete sub-graphs in which each clique corresponds to a cluster. The similarity graph G is derived form H by flipping each edge/non edge with probability α. Therefore, clustering a data set is equivalent to identifying the original clique graph H from the corrupted version G with as few flips (errors) as possible.

Intuitively CAST specifies the desired cluster quality through an affinity threshold (t), average similarity between the objects within a cluster, and applies a heuristic searching process to identify qualified cluster one at a time. Therefore, CAST does not depend on a user-defined number of clusters and deals effectively with outliers. Nevertheless, CAST has the usual difficulty of determining a "good" value for the global parameter t.

CLICK: Cluster Identification via Connectivity Kernels: Motivated by HCS (highly connected subgraph) [140], Shamir et al. presented the algorithm CLICK [276]. CLICK makes the probabilistic assumption that, after standardization, pair-wise similarity values between elements (in the same or different clusters) will be normally distributed. Under this assumption, the weight $w_{i,j}$ of an edge (i, j) is defined as the probability that vertices i and j are in the same cluster. The clustering process of CLICK iteratively finds the minimum cut in the proximity graph and splits the data set recursively into a set of connected components from the minimum cut. CLICK also takes two post-pruning steps to refine the clustering results. The adoption step handles the remaining singletons and updates the current clusters, while the merging step iteratively merges two clusters with similarity exceeding a predefined threshold.

In [276] the authors compared the clustering results of CLICK with Genecluster [107] a SOM approach in two data sets. In both cases, the clusters obtained by CLICK demonstrated better quality in terms of cluster homogeneity and separation. However CLICK has the potential of going out and generating highly unbalanced partitions which separate a handful of outliers from the remaining genes. Furthermore, in gene expression data, two clusters of co-expressed genes may significantly intersect. In such situations, CLICK is unlikely to properly split the two clusters, which are likely to be reported as one highly-connected component.

2.3.3 Hierarchical approaches

These approaches produce a hierarchy of clusters rather than a set number of clusters fixed in advance (as partition-based algorithms). The nested sequence of clusters produced by these approaches makes them appealing when different levels of detail are of interest because small clusters are nested inside larger ones. Nested clusters can be graphically represented by a tree, called *dendrogram*. The branches of a dendrogram no only record the formation of the

clusters, but also indicate the similarity between the clusters. Selection of K clusters from a dendrogram corresponds to cutting the dendrogram with a horizontal line at appropriate height. Each branch cut by the horizontal line corresponds to a cluster. Hierarchical clustering algorithms can be further categorized into two kinds: agglomerative approaches and divisive approaches[68].

Agglomerative approaches

Agglomerative procedures, also called bottom-up methods, initially regard each data object as a n individual cluster (starting with n clusters) and iteratively reduces the number of clusters by merging the two most similar or closest objects or clusters, respectively, until only one "big" cluster is remaining. At the first step, when each object represents its own cluster, the distances between those objects are defined by the chosen distance measure (as seen in the section 2.3.1). However, once several objects have been linked together, a *linkage* or merging rule is needed to determine if two clusters are sufficiently similar to be linked together. There are numerous linkage rules that have been proposed. here are some of the most common [100, 164]:

- *Simple linkage* is determined by the distance of the two closest objects or nearest neighbors in the different clusters.
- *Average linkage*, which uses the average of all distances between points in the two clusters.
- *Complete linkage* is determined by the greatest or maximum distance between any two objects in the different clusters (i.e., furthest neighbors).
- *Unweighted pair-group average linkage*, where the distance between two clusters is calculated as the average distance between all pairs of objects in the two different clusters.

Other variations also exist, such as trying to minimize within-cluster distance as weighted pair-group average linkage, unweighted pair-group centroid linkage, ward's method, etc. (more details in [100, 164]).

Eisen et al. [107] applied an agglomerative algorithm called UPGMA (unweighted pair group method with arithmetic mean) and adopted a method to graphically represent the clustered microarray data set. In this method, the original input gene expression matrix is represented by a colored table (see FIG. 2.6), where large contiguous patches of color represent groups of genes that share similar expression patterns over multiple biological temporal conditions.

FIGURE 2.6 An image showing the different classes of gene expression profiles. Five hundred and seventeen genes whose mRNA levels changed in response to serum stimulation were selected. This subset of genes was clustered hierarchically into groups on the basis of the similarity of their expression profiles, using the procedure of Eisen et al. [107]. The expression pattern of each gene in this set is displayed here as a horizontal line. For each gene, the ratio

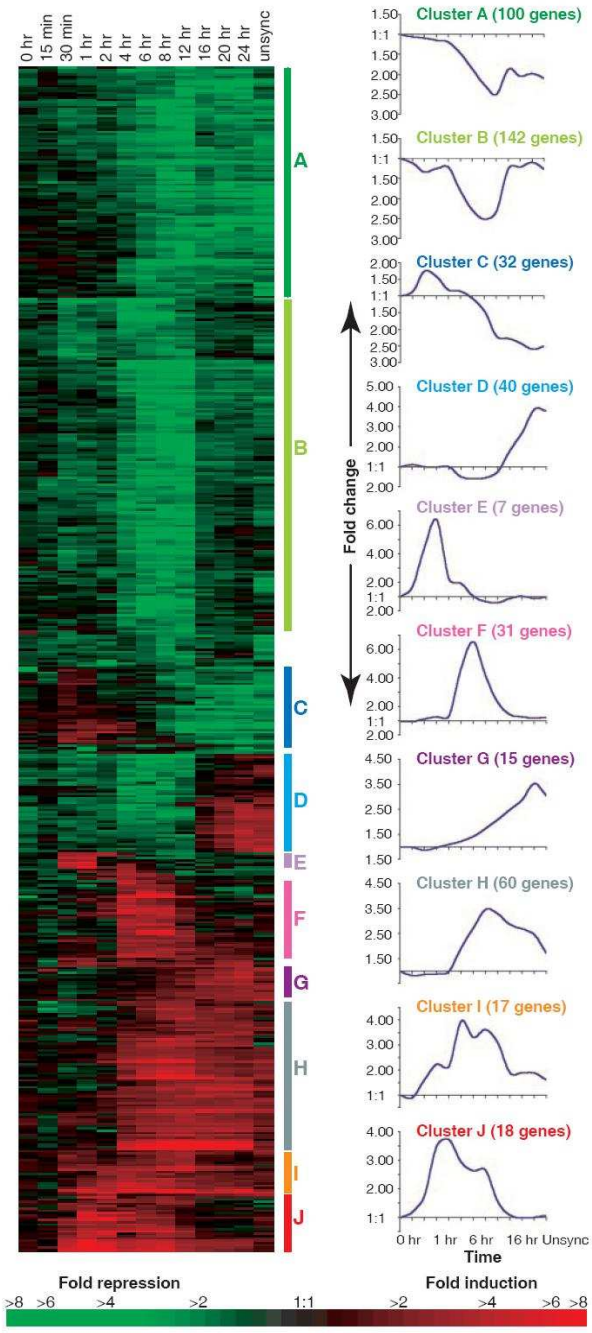

FIG. 2.6: An image showing the different classes of gene expression profiles.

of mRNA levels is represented by a color, according to the color scale at the bottom. The graphs show the average expression profiles for the genes in the biological process.

Hierarchical clustering no only groups together genes with similar expression pattern but also provides a natural way to graphically represent the data set. The graphic representation gives users a thorough inspection of the whole data set so that the users can obtain an initial impression of the distribution of data. Eisen's method is very popular by many biologists and has become one of the most widely-used tools in gene expression data analysis [107, 237, 8].

However, as pointed out in previous microarray studies [298, 19] traditional agglomerative clustering algorithms may not be robust to noise. They often base merging decisions on local information and never trace back to ensure that poor decisions made in the initial steps are corrected later. In addition, hierarchical clustering results in a dendrogram, with no guidance on cutting the dendrogram to derive clusters. Given a typical gene expression data set with thousands of genes, it is unrealistic to expect users to manually inspect the entire tree

To render the traditional agglomerative method more robust to the noise, Sasik et al [64] proposed a novel approach called *percolation clustering*. In essence, these algorithm adopts a statistical bootstrap method to merge two data objects (or two subsets of data objects) when they are significantly coherent with each other. In Bar-Joseph et al. [19] replaced the traditional binary hierarchical tree with a *k-array tree*. A heuristic algorithm was also presented to construct the k-array tree, which reduced susceptibility to noise and generated an optimal order for the leaf nodes. These two approaches increase the robustness of the derived hierarchical tree. However, neither of them indicates how to cut the dendrogram to obtain meaningful clusters.

Divisive approaches

Divisive algorithms (i.e., top-down approaches) start with a single cluster which contains all the data objects. The algorithm splits clusters iteratively until each cluster contains only one data object or a certain stop criterion is met. The various divisive approaches are primarily distinguished by the manner in which clusters are split at each step. In this subsection we explain two divisive algorithms: deterministic annealing (DAA) and super-paramagnetic clustering (SPC). Another similar divisive algorithms used for microarray data clustering is Diana (for more details see: [164]).

Deterministic annealing (DAA): Alon et al. [8] used a divisive approach, called the deterministic-annealing algorithm (DAA) to split genes. First, two initial cluster centroids C_k, $k = 1, 2, ...$, were randomly defined. The expression pattern of gene i was represented by a vector $\vec{x}_i$, and the probability of gene i belonging to cluster k was determined according to a two-component Gaussian model:

$$P_k(\vec{x}_i) = \frac{\exp(-\beta \left|\vec{x}_i - C_k\right|^2)}{\sum_k \exp(-\beta \left|\vec{x}_i - C_k\right|^2)}. \tag{2.12}$$

The cluster centroids were recalculated by $C_k = \sum_i \vec{x}_i P_k(\vec{x}_i) / \sum_i P_k(\vec{x}_i)$, and the EM algorithm [88] was then applied to solve P_k and C_k. Increasing β in small increments to a

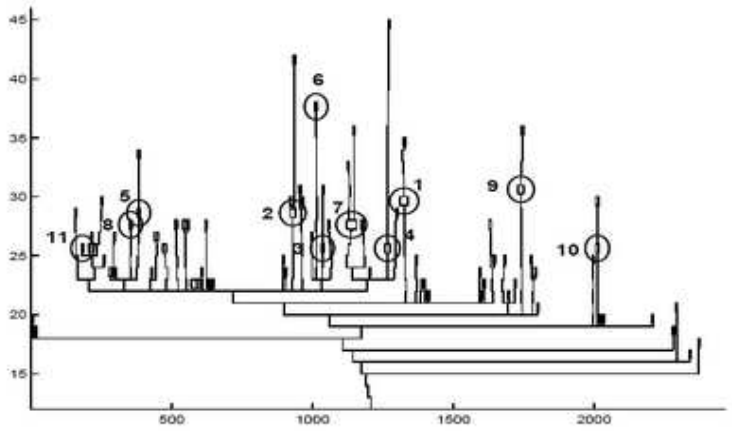

FIG. 2.7: Dendrogram of genes generated by SPC for yeast cell-cycle expression data. Figure from [125].

threshold resulted in two distinct, converged centroids. The entire data set was recursively split until each cluster contained only one gene.

Super-paramagnetic clustering (SPC): Super-paramagnetic clustering, proposed by Blatt et al. in [36] is based on the physical properties of an inhomogeneous ferromagnetic model. SPC first transforms the data set into a *distance graph* where each vertex corresponds to a data object. Two vertices V_i and V_j in the graph are connected by an edge if and only if their corresponding objects X_i and X_j satisfy the *K-mutual-neighbor criterion*, i.e., X_j is one of the $K - nearest$ objects to X_i , and vice versa. Moreover, an edge in the distance graph is associated with a weight $J_{i,j} > 0$, with a smaller Euclidean distance between X_i and X_j associated with a greater weight.

Getz et al. [125] applied SPC to a yeast cell cycle expression data. FIGURE 2.7 illustrates the dendrogram generated by SPC. Three out of the eleven identified clusters correspondent to known phases of the cell cycle, while other clusters revealed features that had not been previously identified and may serve as the basis of future experimental investigation.

Some of the advantages of SPC are its robustness against noise and initialization, a clear signature of cluster formation and splitting, and an unsupervised self-organized determination of the number of cluster at each resolution.

In summary, hierarchical clustering not only groups together genes with similar expression pattern but also provides a natural means to graphically represent the data set. The graphic representation allows users to make a throughout inspection of the entire data set and thus obtain an initial impression of the distribution of the data. Eisen's method as discussed before, is favored by many biologists and has become the most widely-used tool in gene expression data analysis [107, 237, 8]. As noted earlier, the conventional agglomerative approach suffers from a lack of robustness [298], and a small perturbation of the data set may greatly change the structure of the hierarchical dendrogram. Another drawback of the hierarchical approach is its high computational complexity. To construct a "complete" den-

drogram, the clustering process should engage in $\frac{n^2-n}{2}$ merging steps. Furthermore, for both agglomerative and divisive approaches, the "acquisitive" nature of hierarchical clustering prevents the refinement of the previous clustering steps. Thus, in "bad" decision in the initial steps, it can never be corrected in the following steps.

2.3.4 Fuzzy logic approaches

Fuzzy logic is derived from *fuzzy set theory** dealing with reasoning that is approximate rather than precisely deduced from classical predicate logic. Fuzzy logic, linguistic form uses imprecise concepts like "slightly", "quite" and "very". Specifically, it allows *partial membership* in a set and it is related to possibility theory. In this subsection we describe the Fanny approach [164], which is often in microarray data analysis tools as BIOCONDUCTOR.

Fanny: This algorithm uses fuzzy logic and produces a probability vector for each observation. A hard cluster is determined by assigning an observation to a group which has the highest probability. Like distance-based methods, one has a choice of using a general dissimilarity measure. Assuming K denote the total number of desired clusters, Fanny computes the probability vectors (called membership coefficients): $u_{x_1}, ..., u_{x_K}$ for all genes x that minimize the objective function

$$\sum_{k=1}^{K} \frac{\sum_{x,y} u_{x_k}^2 u_{y_K}^2 d(x,y)}{\sum_x u_{x_k}^2}. \tag{2.13}$$

After minimizing this equation, hard clusters are then produced, if needed by assigning genes to the group with the highest probability [164]. Typically, relatively fewer hard clusters are produced by this method. That is the major drawback of these method, if a specific number of hard clusters is desired, fanny method may not be a suitable algorithm [85].

2.3.5 Density-based approaches

Density-based approaches describe the distribution of a given data set by the "density" of data objects. The clustering process involves a search of the "dense areas" in the object space [110]. In this subsection we briefly present three algorithms: DBSCAN [110], OPTICS [11] and DENCLUE [146].

DBSCAN: The DBSCAN algorithm introduced by Ester et al. [110] is grounded on a density-based notion of clusters. To measure the "density" of data objects, DBSCAN defines the *ϵ-neighborhood* of an object p as a set o objects $N_\epsilon(P)$ such that the distance between p and each object q in $N_\epsilon(P)$ is smaller than a user-specified threshold ϵ. Intuitively, an object p has a "high" density if $N_\epsilon(P) \geq MinPts$, where $MinPts$ is a user-specified threshold.

The clustering process of DBSCAN scans the data set only once and reports all the clusters and noise. For each data object x_i, DBSCAN check the *ϵ-neighborhood* $N_\epsilon(x_i)$ of x_i. If $N_\epsilon(x_i)$ contains more than $MinPts$ data objects (i.e., x_i is a core object), DBSCAN creates a new cluster and then iteratively retrieves all data objects which are density-reachable from x_i with respect to ϵ and $MinPts$. If x_i is a border object, no points are *density-reachable* from x_i, and DBSCAN visits the next point in the data set.

While DBSCAN is able to discover clusters with arbitrary shape an is quite efficient for large data sets, the algorithm is very sensitive to input parameters. It may generate very different clustering results from slightly differences in parameter settings [11].

OPTICS: Inspired in DBSCAN, Ankerst et al. [11] introduced the algorithm OPTICS. This algorithm does not generate clusters explicitly but instead creates an ordering of the data objects and illustrates the cluster structure of the data set. in essence, this ordering contains information that is equivalent to the clustering of DBSCAN, with a wide range of parameters settings.

OPTICS algorithm generates an ordering of objects by scanning the data set once. The algorithm maintains a queue in which data objects are sorted in ascending order according to their reachability-distances. The ordering of objects is simply the sequence in which data objects are extracted from the queue. Once the ordering of the objects and their reachability-distances are obtained, clusters can be extracted using this information.

DENCLUE: Unlike the local density measures employed by DBSCAN and OPTICS, DENCLUE [146] measures object density from a global perspective. Data objects are assumed to "influence" each other, and the density of a data object is the sum of influence functions from all data objects in the data set. The incorporation of a variety of *influence functions* allow DENCLUE to be a generalization of many partition-based, hierarchical, and density-based clustering methods. More details in the mathematical foundation of DENCLUE can be seen in [146].

DENCLUE has a solid statistical foundation and allows a compact mathematical description of arbitrarily-shaped clusters. Moreover, has good clustering properties for data sets with high dimensionality and large amounts of noise. The computational efficiency of DENCLUE is significantly higher than some influential algorithms, such as DBSCAN (by a factor up to 45) [146]. However, the method requires careful selection of the density parameter σ and the noise threshold ξ, as the setting of such parameters may significantly influence the quality of the clustering results [11]. Moreover, DENCLUE outputs all clusters at the same level. Therefore, it cannot support an exploration of hierarchical cluster structures which exploits user's domain knowledge.

To our knowledge, the density-based approaches described before have not been directly applied to gene expression data for cluster analysis.

2.3.6 Cluster validation

The previous sections have reviewed a number of clustering algorithms which partition a data set on the basis of particular clustering criteria. However, different clustering algorithms, or even the use of a single clustering algorithm with different parameters, generally result in different sets of clusters. Therefore, it is important to compare various clustering results and select the one that best fits the "true" data distribution.

Cluster validity is assessed on at least three general bases. First, the quality of clusters can be measured in terms of the degree of their *homogeneity, separation and consistency*. By definition, objects within one cluster are assumed to be similar to each other, while objects in different clusters are dissimilar. The second aspect relies on a given *ground truth** of

the clusters. The "ground truth" may come from domain knowledge, such as known function families of genes. Cluster validation is based on the extent of agreement between the clustering results and this "ground truth". The third aspect of cluster validity focuses on the reliability of the clusters or on the likelihood that the cluster structure has not been formed by chance. In this subsection, we discuss these three aspects of cluster validation.

Homogeneity, separation and consistency

Here we present different measures for testing the homogeneity, separation and consistency of the obtained clusters.

The homogeneity of a cluster is defined by some measure which quantifies the similarity of data objects (genes) in the cluster C. For example,

$$H_1(C) = \frac{\sum_{X_i, X_j \in C, X_i \neq X_j} Similarity(X_i, X_j)}{\|C\| * (\|C\| - 1)}, \tag{2.14}$$

where $\|C\|$ is the cardinality of the cluster C and $Similarity(X_i, X_j)$ is the any similarity measure between gene i and gene j.

This definition represents the homogeneity of cluster C by the average pairwise object similarity within C An alternate definition evaluates the homogeneity with respect to the "centroid" of the cluster C, i.e.,

$$H_2(C) = \frac{1}{\|C\|} \sum_{X_i \in C} Similarity(X_i, \hat{X}), \tag{2.15}$$

where $\hat{X}$ is the "centroid" of C. Other definitions, such as the representation of cluster homogeneity via maximum or minimum pairwise or centroid-based similarity within C can also be useful and perform well under certain conditions.

Cluster separation is analogously defined from various perspectives to measure the dissimilarity between two clusters C_1, C_2. For example,

$$S_1(C_1, C_2) = \frac{\sum_{X_i \in C_1, X_j \in C_2} Similarity(X_i, X_j)}{\|C_1\| * \|C_2\|} \tag{2.16}$$

and

$$S_2(C_1, C_2) = Similarity(\hat{X}_1, \hat{X}_2) \tag{2.17}$$

where $\hat{X}_1$ and $\hat{X}_2$ are the centroids of C_1 and C_2, respectively.

Since these definitions of homogeneity and separation are based on the similarity between objects, the quality of C increases with higher homogeneity values within C and lower separation values between C and other clusters. Once we have defined the homogeneity of a cluster and the separation between a pair of clusters, for a given clustering result $C = \{C_1, C_2, ..., C_K\}$, we can define the homogeneity and the separation of C. For example, Sharan et al. [276] used definitions of

$$H_{ave} = \frac{1}{N} \sum_{C_i \in C} \|C_i\| * H_2(C), \tag{2.18}$$

and

$$S_{ave} = \frac{1}{\sum\limits_{C_i \neq C_j} \|C_i\| * \|C_j\|} \sum_{C_i \neq C_j} (\|C_i\| * \|C_j\|) S_2(C_1, C_2), \tag{2.19}$$

to measure the average homogeneity and separation for the set of clustering results C.

Cluster consistency ($Cons$) is analogously defined from various perspectives to measure the average distance or proportion of elements within one *reference cluster* $C^{g,0}$ and *modified cluster* $C^{g,j}$. In this particular notation we suppose that for each gene $1 \leq g \leq N$. The reference cluster $C^{g,0}$ be the cluster in the original data containing gene g and the modified cluster $C^{g,j}$ denote the cluster containing gene g in the clustering based on the data set with biological condition j deleted. The experiment contains N genes and M biological conditions. For example, Datta et al. [85] used definitions of three consistency measures, explained in the next paragraphs.

The *average proportion of non-overlap measure* computes the average proportion of genes that are not put in the same cluster by the clustering method under consideration on the basis of the full data and the data obtained by deleting the expression levels at one time point at a time. This measure is defined as:

$$Cons_1(K) = \frac{1}{NM} \sum_{g=1}^{N} \sum_{j=1}^{M} \left(1 - \frac{\left\| C^{g,j} \cap C^{g,0} \right\|}{\|C^{g,0}\|} \right), \tag{2.20}$$

where $\left\|C^{g,0}\right\|$ is the cardinality of the set $C^{g,0}$.

The *average distance between means measure* computes the (average) distance between the mean expression ratios (log transformed) of all genes that are put in the same cluster by the clustering method under consideration on the basis of the full data and the data obtained by deleting the expression levels at one specific biological condition. This measure is defined as:

$$Cons_2(K) = \frac{1}{NM} \sum_{g=1}^{N} \sum_{j=1}^{M} Similarity(\bar{x}_{C^{g,j}}, \bar{x}_{C^{g,0}}), \tag{2.21}$$

where $\bar{x}_{C^{g,0}}$ denotes the average expression profile for genes across cluster $C^{g,0}$ and $\bar{x}_{C^{g,j}}$ denotes the average expression profile for genes across cluster $C^{g,j}$.

The *average distance measure* computes the average distance between the expression levels of all genes that are put in the same cluster by the clustering method under consideration on the basis of the full data and the data obtained by deleting the expression levels of one biological condition each time. This measure is defined as:

$$Cons_3(K) = \frac{1}{NM} \sum_{g=1}^{N} \sum_{j=1}^{M} \frac{1}{\|C^{g,j}\| \, \|C^{g,0}\|} * \sum_{g \in C^{g,0}, g' \in C^{g,j}} Similarity(x_g, x_{g'}), \tag{2.22}$$

where $Similarity(x_g, x_{g'})$ is any similarity measurement between the expression profiles of genes g and g'.

Agreement with reference partition

If the *ground truth* of the cluster structure of the data set is available, we can test the performance of a clustering process by comparing the clustering results with the ground truth. Given the clustering results $C = \{C_1, ..., C_k\}$, we can construct a $n * n$ binary matrix C, where n is the number of data objects, $C_{i,j} = 1$ if X_i and X_j belong to the same cluster, and $C_{i,j} = 1$ otherwise. Similarly we can build the binary matrix P for the ground truth $P = \{P_1, ..., P_s\}$. The agreement between C and P can be disclosed via the following values:

- n_{11} is the number of object pairs $(X_i\ , X_j\ )$, where $C_{i,j} = 1$ and $P_{i,j} = 1$.
- n_{10} is the number of object pairs $(X_i\ , X_j\ )$, where $C_{i,j} = 1$ and $P_{i,j} = 0$.
- n_{01} is the number of object pairs $(X_i\ , X_j\ )$, where $C_{i,j} = 0$ and $P_{i,j} = 1$.
- n_{00} is the number of object pairs $(X_i\ , X_j\ )$, where $C_{i,j} = 0$ and $P_{i,j} = 0$.

Some commonly used validation indices [288, 138] have been defined to measure the degree of similarity between C and P:

$$Rand\ index : Rand = \frac{n_{11} + n_{00}}{n_{11} + n_{10} + n_{01} + n_{00}} \tag{2.23}$$

$$Jaccard\ coefficient : JC = \frac{n_{11}}{n_{11} + n_{10} + n_{01}} \tag{2.24}$$

$$Minkowski\ measure : Minkowski = \sqrt{\frac{n_{10} + n_{01}}{n_{11} + n_{01}}} \tag{2.25}$$

The *Rand index* and the $Jaccard\ coefficient$ measure the extent of agreement between C and P, while *Minkowski measure* illustrates the proportion of disagreements to the total number of object pairs (X_i, X_j), where X_i, X_j belong to the same set in P. It should be noted that the $Jaccard\ coefficient$ and the *Minkowski measure* do not (directly) involve the term n_{00}. These two indices may be more effective in classifying genes because a majority of pairs of objects tend to be separate clusters, and the term n_{00} would dominate the other three terms in both accurate and inaccurate solutions. Other methods are also available to measure the correlation between clustering results and the ground truth [138]. Again, the optimal measure selection is application-dependent.

Reliability of clusters

While a validation measure can be used to compare different clustering results, this comparison will not reveal the reliability of the resulting clusters; that is, the probability that the clusters are not formed by chance. In the following subsection, we will review some representative approaches to measuring the significance of the derived clusters.

P-value of a cluster

In order to answer to the probability that the clusters are or not formed by chance, one recurrent tool in microarray analysis is the parametric hypothesis testing (as seen in section 2.2.1).

Here the null hypothesis, H_0, states that the cluster of co-expressed genes is associated by chance and H_1 states the contrary, that are not formed by hazard. Many microarray data analysis approaches [303, 258, 99, 151, 228, 234] have chosen the one-tailed test with hypergeometric distribution, with a cumulative probability statistic and an user-defined threshold to answer this hypothesis proof: H_0 *vs* H_1 (clear explication of the hypothesis proof and discussion can be found at [99]. Here we describe the essentials of these hypothesis proof.

We can translate the H_0 challenge to this probability problem: We have a microarray experience with N genes, any given gene is either in the co-expressed group or not. So we have N genes in two categories: I (in the group) or O out the group. We observe that x of theses K genes are I and we want to find out what is the probability of this happening by chance. So, our question is: given N genes of which M are I and $N-M$ are O, we pick randomly K genes and we ask what is the probability of having exactly x genes of type I. Once we pick a gene from the chip, we cannot pick it again so this is clearly sampling without replacement. The probability function P that answers exactly to this problem is the hypergeometric distribution with parameters (N, M, K):

$$P(X = x \mid N, M, K) = \frac{\left(\begin{array}{c} M \\ x \end{array} \right) \left(\begin{array}{c} N-M \\ K-x \end{array} \right)}{\left(\begin{array}{c} N \\ K \end{array} \right)} \tag{2.26}$$

Based on this probability and in the cumulative distribution of X, the $p-value$ for overrepresented categories can be calculated as:

$$P(X = x \mid N, M, K) = 1 - \sum_{i=0}^{x} \frac{\left(\begin{array}{c} M \\ i \end{array} \right) \left(\begin{array}{c} N-M \\ K-i \end{array} \right)}{\left(\begin{array}{c} N \\ K \end{array} \right)}, \tag{2.27}$$

if the sum is larger than 1/2. Thus, given threshold α, for each cluster C if $p-value < \alpha$ then H_0 is rejected. In this case cluster C is not built by the effect of hazard. A smaller probability of the $p-value$ indicates a higher significance of the clustering results. If the number of clusters is "high", $p-value$ approaches have to deal with multi-testing problem (discussed in section 2.2).

Prediction strength

In [330], Yeung et al. proposed an approach to the validation of gene clusters based on the idea of "prediction strength". Intuitively, if a cluster of genes formed with respect to a set of samples (attributes) has possible biological significance, then the expression levels of the genes within that cluster should also be similar to each other in "test" samples that were not used to form the cluster.

Yeung et al. proposed a specific *figure of merit (FOM)* to estimate the predictive power of a clustering algorithm. Suppose $C_1, C_2, , ..., C_k$ are the resulting clusters based on samples $1, 2, ..., (e-1), (e+1), ..., m$, and the sample e is left out to test the prediction strength. Let $R(x, e)$ be the expression level of gene x under sample e in the raw data matrix. Let $\mu_{C_i}(e)$

be the average expression level in sample e in cluster C_i. The *figure of merit* with respect to e ant the number of clusters k is defined as

$$FOM(e,k) = \frac{1}{n} * \sum_{i=1}^{k} \sum_{x \in C_i} \sqrt{\frac{\big(R(x,e) - \mu_{C_i}(e)\big)^2}{n}} \tag{2.28}$$

Each of the m samples can be left out in turn, and the *aggregate figure of merit* is defined as $FOM(k) = \sum_{e=1}^{m} FOM(e,k)$. The FOM measures the mean deviation of the expression levels of genes in e relative to their corresponding cluster means. Thus, a small value of FOM indicates a strong prediction strength, and therefore a high level reliability of the resulting clusters.

Levine et al. [177] proposed another figure of merit M based on a resampling scheme. This scheme assumes that the cluster structure derived from the entire data set should be able to "predict" the cluster structure of subsets of the full data. M measures the extent to which the clustering assignments obtained from resamples (subsets of the full data) agree with those from the full data. A high value of M against a wide range of resampling indicates a reliable clustering result.

2.4 Fifth Analysis Step: Knowledge Discovery via Data Interpretation

The microarray data analysis fifth step is dedicated to knowledge discovery via interpretation of previous results. The goal of the interpretation step is the confrontation between two kinds of information: numeric data represented by gene expression profiles and semi-structured data represented by gene annotations extracted from different sources of biological information (the digital expression of current biological knowledge) as shown in Fig. 5.1.

One of the current challenges in gene expression technologies is to highlight the main co-expressed and co-annotated gene groups* by using prior biological knowledge [13]. In other words, the issue is the interpretation of microarray results via integration of gene expression profiles with corresponding biological gene annotations extracted from biological knowledge bases.

This section briefly introduces the fifth microarray analysis step: knowledge discovery via data interpretation. Due to the importance of the interpretation step to our research (specially for our knowledge discovery models described in chapter 6 and 7), we develop it in full detail in chapter 5.

This section briefly introduces the fifth data analysis step (elements, history and basics). Then, it presents a new and original classification in three axes of the gene expression data analysis interpretation approaches. Finally, it gives a short description of each axis.

2.4.1 Introduction

At the beginning of gene expression technologies, researches were focused on the numeric[12] side. So, there have been reported ([69, 90, 107, 298, 303, 26]) a variety of data analysis approaches which identify groups of co-expressed genes based only on expression profiles without taking into account biological knowledge. A common characteristic of purely numerical approaches is that they determine gene groups (or clusters) of potential interest. However, they leave to the expert the task of discovering and interpreting biological similarities hidden within these groups. These methods are useful, because they guide the analysis of the co-expressed gene groups. Nevertheless, their results are often incomplete, because they do not include biological considerations based on prior biologists knowledge.

In order to process the interpretation step in an automatic or semi-automatic way, the bioinformatics community is facing an ever-increasingly volume of biological information on gene annotations. Beside minimal microarray information, we have identified five different sources of biological information: general sequence databases (EBI, NCBI, DDBJ, etc.), semantic databases as thesauri, ontologies, taxonomies or semantic networks (UMLS, GO, KEGG, etc.), databases of experiments (GEO, Arrayexpress, etc.), bibliographic databases (Medline, Biosis, etc.), organism-specific nomenclature data sets (Human, Mouse, SGD) as seen in FIG. 5.1. (a more detailed description of biological sources of information will be given in chapter 3). The exploitation of these different sources of knowledge, often referred to as *Integration of biological knowledge in gene expression analysis*, is quite a complex task, so scientists developed techniques for integrating them into more complex databases [34], [217].

The interpretation step may be defined as the result of the integration between gene expression profiles analysis with corresponding gene annotations. This integration process consists in grouping together co-expressed and co-annotated genes. Based on this definition, we propose a classification of three microarray data interpretation research axes : *the prior or knowledge-based axis*, *the standard or expression-based axis* and *the co-clustering axis*. Our classification emphasizes the weight of the integration process scheduling on the final results [175, 112, 186, 139].

In prior or knowledge-based approaches, the co-annotated gene groups are built and then the gene expression profiles are integrated. In standard or expression-based approaches, co-expressed gene groups are built and then gene annotations are integrated. Finally, co-clustering approaches integrate co-expressed and co-annotated gene groups at the same time.

2.4.2 Prior or knowledge-based axis

Prior or knowledge-based approaches initially consider biological knowledge from the five sources explained before. Therefore, first they build co-annotated gene groups sharing the same biological annotations. Then, they integrate the expression profiles data for each of the genes classified into co-annotated groups, highlighting those co-expressed. Finally, the statistical significance of co-annotated and co-expressed gene groups is tested. Four remark-

[12] We understand by numeric part the analysis of the gene expression measures only, disregarding the biological annotations

able knowledge-based approaches may be mentioned: GSEA [215], iGA [53], PAGE [167] and CGGA [203].

2.4.3 Standard or expression-based axis

This axis follows the most frequently used procedure for microarray data analysis, it has been followed since the beginning of microarray technology with encouraging interpretation results [90], [107] and [69]. Expression-based approaches start by building clusters of genes sharing similar expression profiles. Then, they integrate the biological annotations of each gene from an expression cluster, building co-expressed and co-annotated subsets of genes. Finally, a selection of co-expressed and co-annotated gene groups is made by testing its statistical significance.

The most remarkable expression based approaches which integrate semantic databases (such as Gene Ontology - GO) are: FunSpec [258], OntoExpress [99], Quality Tool [128], EASE [151], THEA [228], Graph Theoretic Modeling [175] and GENERATOR [234]. Masys et al. [205] propose an unique approach using bibliographic databases (Medline, Biosis, MeSH, etc.).

2.4.4 Co-clustering axis

The challenge of co-clustering approaches is to build a clustering algorithm capable of integrating heterogeneous data as numeric gene expression profiles with textual gene annotations at once. Each co-clustering approach has its specific parameters: biological source of information, clustering method and integration algorithm. Few co-clustering approaches have been reported (Co-Cluster [139], Bicluster [186] and [202]), the main reason being the difficulty to build clustering methods fitting heterogeneous sources of information.

We explain more in detail all-three axes, their remarkable approaches, and a discussion and comparison between them in chapter 5. The lector can also see [200] for a more detailed description of these approaches and their methodology.

Chapter 3

Biological Sources of Information

In chapter 2 we have developed the five-step procedure (i.e. protocol and image analysis, statistical data treatment, gene selection, gene classification, and knowledge discovery via data interpretation) to analyze data issued from gene expression technologies. In fact, we are interested in the fifth step of this procedure which focuses on knowledge discovery via interpretation of the gene expression technology results. The goal of the interpretation step was defined as the integration of gene expression profiles and biological knowledge represented by gene annotations*.

This chapter gives a complete overview of the sources of biological knowledge available for accomplishing gene expression technology analysis. It starts with a brief introduction and a discussion of the two key concepts of our classification: source content and source structure. Then, a section is devoted to each of the six different sources of biological information: minimal microarray information, molecular sources, semantic sources, bibliographic databases, experience databases and gene-related sources.

3.1 Introduction

Nowadays, one of the main challenges in gene expression technology is to highlight the main co-expressed and co-annotated gene groups using at least one of the different sources of biological information [13]. In other words, the issue is interpretation of microarray results via integration of gene expression profiles with corresponding biological gene annotations extracted from biological data sources. In order to process the interpretation step in an automatic or semi-automatic way, the bioinformatics community is faced to an ever-increasing volume of sources of biological information on gene annotations .

We have classified the information into six sources that are: minimal microarray information, molecular sources (EMBL, GenBank,etc.), semantic sources (UMLS, GO, Taxonomy, etc.), bibliographic databases (Medline, Biosis, OMIN, etc.), gene expression databases (GEO, Arrayexpress, Microarray, etc.), and gene-related or protein-related sources (KEGG, GeneCards, etc.), as seen in Fig. 3.1. Our classification is based on two key concepts: the main content and the structure of these six sources of biological information. Here, we explain the role of these two concepts in our classification and the conventions that we will take into account along this chapter.

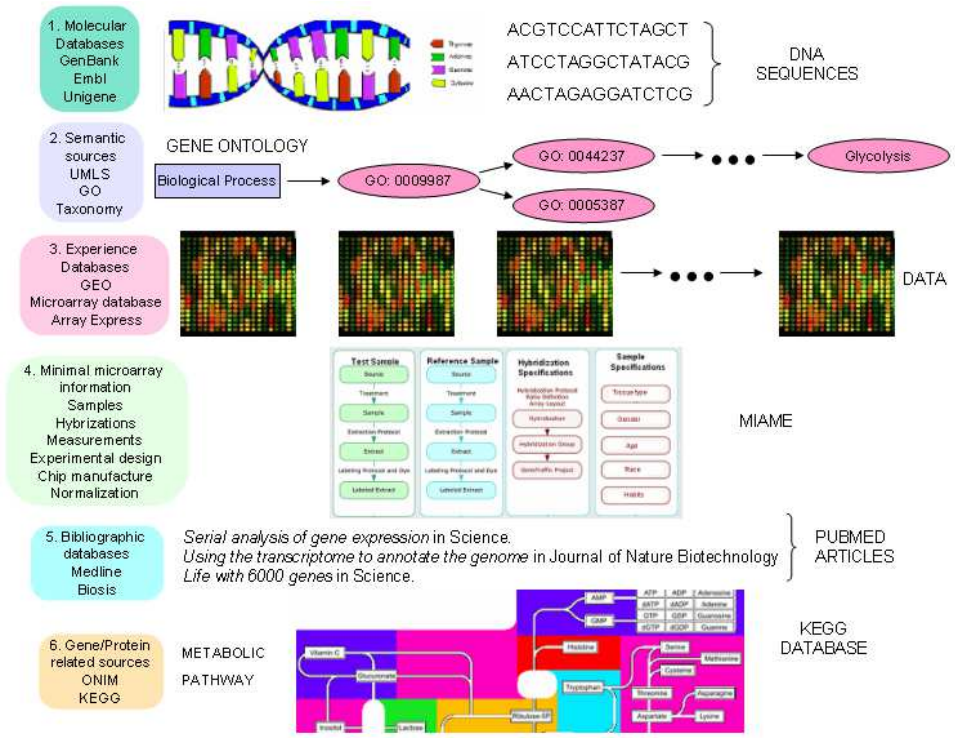

FIG. 3.1: Different kinds of biological sources of knowledge and information.

Main content of the sources of information

Since the number of gene expression technology applications (presented in section 1.3) is so vast, the related sources of information are becoming enormous. These sources are generate by different gene expression technology related domains of knowledge as: genomics, transcriptomics, proteomics, physiology, medicine.

Genomics is the study of the entire organism genome. It investigates genes or DNA regions which code for something (proteins or RNA) that could play a role or functions in the cell.

Transcriptomics identifies the expression profiles and rules of identified genes in a biological space (e.g. tissue, organ etc.), in time (e.g. while the embryonal development) and under certain conditions (e.g. cancer vs. normal, drug treatment vs. placebo etc.).

Proteomics studies the same parameters as genomics function and structure but for proteins. It also researches the interactions between the different proteins.

Physiology is the study of the mechanical, physical, and biochemical functions of living organisms.

By combining the three molecular biology sciences (genomics, transcriptomics and proteomics) and physiology we hope to understand how metabolic pathways* work for better comprehension of the physiology of the living organisms.

As we have explained in gene expression technology applications (section1.3), many of these applications are medical-related: comprehension of diseases (cancer, diabetes, schizophrenia, etc.), drug development and treatment, mutation detection, screening of disease genes, patterns in pathogens etc. Thus, the need of medical data sources of information is important.

Structure of the sources of information

One key issue in bioinformatics is the way the data and information is structured. The explosion of data and information issued from gene expression technology and related sciences in the last decade have revealed the need of structuring data. All six sources of information mentioned before may present different structures. In order to avoid the data structure confusion found in some biological sources of information, in this chapter we take into account these conventions.

Data consists of propositions that reflect reality. Such propositions may comprise numbers, words, images, measurements or observations of a variable.

Information is the result of processing, manipulating and organizing data in a way that adds to the knowledge of the person receiving it. In terms of data, it can be defined as a collection of facts from which conclusions may be drawn.

Knowledge is an acquisition (often impulsed by information) that involves complex cognitive processes: perception, learning, communication, association, and reasoning.

Repository is a central place where data is stored and maintained. This definition is very large it may be: a place where data is stored, a place where multiple databases or files are located for distribution over a network, a computer location that is directly accessible to the user without having to travel across a network etc.

Database is a structured collection of records or data which is stored in a computer so that a program can consult it to answer queries. The records retrieved in answer to queries become information that can be used to make decisions.

Knowledge can be represented by data models* as: taxonomies, thesaurus, ontologies, semantic networks, ordered respectively to their semantics weight*.

Taxonomies are vocabularies of relationship terms, often presented in a hierarchical way.

Thesaurus are controlled vocabularies of associative terms. It can be a list of semantically orthogonal topical terms.

Ontologies are semantic catalogues of the concepts* within a domain and the relationships between those concepts. It is used to reason about the objects within that domain.

Semantic networks are networks that involve semantic associations useful for human browsing to different types of biological information and knowledge data. It can be seen as a directed graph consisting of vertices, which represent concepts, and edges, which represent semantic relations between the concepts.

3.2 Minimal Experimental Biological Information

Here, we present the Minimum Information about microarray experiments (MIAME), that describe the minimum information required to ensure that microarray data can be easily interpreted and the results derived from its analysis can be independently verified. MIAME was proposed by European bioinformatics Institute (EBI) and is fully explained in [46]. The MIAME states that the minimum information about a published microarray-based gene expression experiment includes a description of the following six sections (for better comprehension of the microarray experiment steps and analysis steps the lector can see Fig. 1.17 and 2.1 respectively):

1. Experimental design: the set of hybridization experiments as a whole.
2. Chip manufacture: each chip used and each element (spot, feature) on the array.
3. Samples: samples used, extract preparation and labeling.
4. Hybridizations: procedures and parameters.
5. Measurements: images, quantification and specifications.
6. Normalization controls: types, values and specifications.

Each of these sections contains information that can be provided using controlled vocabularies, as well as fields that use free-text format. Here we discuss only the general information required in each of these sections, for a full description see the MIAME document.

Experimental design

The minimal information required in this section includes the type of the experiment (such as normal-versus-diseased comparison, time course, dose response, and so on) and the experimental variables, including parameters or conditions tested (such as time, dose, genetic variation or response to a treatment or compound). This section also provides general quality-related indicators such as usage and types of replicates.

Finally, this section specifies the experimental relationships between the chip and sample entities, that is, which samples and which arrays were used in each hybridization assay. Each of these will be assigned unique identifiers that are cross-referenced with the information provided in the following sections. This information will allow the user to reconstruct unambiguously the experiment design and to relate together information from further MIAME sections.

Chip manufacture

The aim of this section is to provide a systematic definition of all chips used in the experiment, including the genes represented and their physical layout on the chip. It consists in three parts: i) a description of the chip as a whole (such as platform type, provider and surface type); ii) a description of each type of element or spot used (properties that are typically common to many elements, such as "synthesized oligonucleotides" or "PCR products from cDNA clones") and iii) a description of the specific properties of each element, such as the DNA sequence and, possibly, quality-control indicators.

The challenge for element definition is to achieve a unique and unambiguous description of the element. Because references to an external gene index may not be stable, it is essential to physically identify each element's composition. Thus, where elements are based on cDNA clones, PCR or composite oligonucleotides, it is necessary that clone IDs are specified.

Samples

The MIAME "sample" concept represents the biological material for which the gene expression profile is being established. This section is divided into three parts which describe the source of the original sample (such as organism taxonomy and cell type) and any biological *in vivo* or *in vitro* treatments applied, the technical extraction of the nucleic acids, and their subsequent labeling.

The characteristics to accurately define a biological sample vary greatly from organism to organism. Currently, the single common feature of all samples is the organism's taxonomic definition. A list of qualifiers (qualitative sample variables) may accompany the sample description. For example in the case of humans these variables can be: gender, nationality, smoking condition, alcoholic condition etc. As for laboratory protocols for sample treatments, sample extraction and labeling, need to be specified initially. It is desirable that knowledge of these protocols are also presented in the data description for best interpreting the data.

Hybridizations

This section presents the laboratory conditions under which the hybridizations were carried out. MIAME requires that a number of critical hybridization parameters are explicitly specified: choice of hybridization solution, nature of the blocking agent, wash procedure, quantity of labeled target used, hybridization time, volume, temperature and descriptions of the hybridization instruments.

Measurements

It consists in three progressive parts from raw to processed data: a) the original scans of the array (images), b) the microarray quantification matrices based on image analysis, and c) the final gene expression matrix after normalization and consolidation from possible replicates.

Images represent the primary data form a microarray assay and the image processing algorithms used for analysis can affect the conclusions that are reached. Thus, MIAME includes a specification for image deposition, scanning protocols and image analysis methods.

For each experimental image, a microarray quantification matrix contains the complete image analysis output as directly generated by the image analysis software. These output should include the information that permits the nature and quality of individual spot measurements to be assessed.

Finally, the gene expression matrix (summarized information) consists of sets of gene expression levels for each sample. If microarray quantification matrices can be considered spot/image centric, then the gene expression matrix is gene/sample centric. At this point, expression values may have been normalized, consolidated and transformed in any number of ways by the submitter in order to present the data in a tractable form to scientific analysis.

Normalization Controls

A typical microarray experiment involves a number of hybridization assays in which the data from multiple samples are analyzed to identify relative changes in expression levels, identify differentially expressed genes and, in many cases, discover classes of genes or samples having similar patterns of expressions. In this typical experiment it exists a "reference design", in which many samples are compared to a common reference sample so as to facilitate inferences about relative expression changes between samples. For these comparisons, the reported hybridization intensities derived from image processing must be normalized (as seen in section 2.2).

In this section, MIAME standard invites to specify the parameters relevant to normalization and control elements as: normalization strategy, normalization and quality control algorithms used, the identities and location of the chip elements serving as controls and hybridization extract preparation.

The MIAME specifications are intended to draft sufficiently detailed to capture the information needed to analyze and evaluate microarray data. In effect, these specifications are only partially taken by the existent commercial and public gene expression technologies laboratories. A general consensus for presenting the information generated in a microarray experiment does not exist yet. However, is crucial to develop a general agreement among the microarray laboratories. At the moment, microarray analysts have to adapt to the minimal information provided by the specific data itself.

3.3 Molecular Databases

The goal of molecular databases is to gather collections of structural or functional or related data about nucleotides (DNA, RNA, genes) or proteins. These databases allow an easy update of new compounds or functions and powerful queries about compound annotations. Molecular databases contain data functions and structures of three molecular biology sciences: genomic, transcriptomic and proteomic (explained in section 3.1).

We can divide molecular databases in two large groups:

- Nucleotide databases which contain information of the structure and/or functions about any kind of sequence of nucleotides as: DNA/RNA sequences, genomes, EST, etc.
- Protein databases which contains information of the structure and/or functions and/or related information about proteins.

Different nucleotide and protein databases are hosted and maintained by many different organisms, laboratories and organizations. The principal molecular databases are hosted and maintained by a consortium composed of three research world groups: European Bioinformatics Institute (EBI), National Center for Biotechnology information (NCBI) and DNA data bank of Japan (DDBJ). There are mirror sites which exchange the new sequences information in real time. One essential goal of this consortium is to receive the information and to make it public, developing at once annotation tools for manipulating this information.

In the next sections we explain the contents more in detail and some of the remarkable nucleotide and protein databases. The databases mentioned above do not intend to be an exhaustive list of the existent databases, they are cited as examples of existent databases of each type. We have to take into account that they exists thousands of molecular databases, regarding the quantity of sequenced organisms or the possible functions of a whole genome or the possible proteins structure.

3.3.1 Nucleotide databases

These databases contain several information about any kind of sequence of nucleotides as: DNA/RNA sequences, genomes, EST, etc. This information is typically of one of these five types:

1. The sequence of nucleotides.
2. Annotations of these sequences.
3. The physical maps*.
4. The genetic maps*.
5. Links to more specialized databases.

Taking into account the principal source of information provided by nucleotide databases, we divide them into two main groups: sequence databases and genome databases.

Sequence databases

These databases are focused in all kinds of sequence of nucleotides as: DNA/RNA sequences, EST, etc. The information that they contain is the sequences of nucleotides and at least one of the points b)-e) mentioned above.

There exist three main sequence databases hosted and managed by the International Nucleotide Sequence Database Consortium composed by three organisms: NCBI, EBI and DDBJ, these databases are GenBank, EMBL and DDBJ respectively. Here, we present a brief explanation of these sequence databases.

GenBank, the National Institutes of Health (NIH) genetic sequence database, is an annotated collection of publicly available DNA sequences. The records within GenBank represent, in most cases, single, contiguous stretches of DNA or RNA with annotations. GenBank files are grouped into divisions; some of these divisions are phylogenetically based, whereas others are based on the technical approach that was used to generate the sequence information. Presently, all records in GenBank are generated from direct submissions to the DNA sequence databases from the original authors, who volunteer their records to make the data publicly available or do so as part of publication process. For more details see [29].

The EMBL Nucleotide Sequence Database (also known as EMBL-Bank) constitutes Europe's primary nucleotide sequence resource. Main sources for DNA and RNA sequences are direct submissions from individual researchers, genome sequencing projects and patent applications. More details in [111].

DDBJ (DNA Data Bank of Japan) is a DNA data bank which is officially certified to collect DNA sequences from researchers and to issue the internationally recognized accession number to data submitters. It also provides worldwide many tools for data retrieval and analysis developed by DDBJ and others.

These three databases are enormous, so different organisms, including the international consortium cited above, have developed more specific sequence databases. These specific sequence databases contain more specific information, which aim at special applications or knowledge domains (including sometimes analysis tools), that can be more easily exploitable by scientists around the world. We can mention some remarkable databases as examples: dbEST, Unigene, SAGEmap, STACK, Refseq, HomoloGene, Nucleotide, and there exist many others. We briefly explain some of these sequence databases.

dbEST is a division of GenBank that contains sequence data and other information on "single-pass" cDNA sequences, or Expressed Sequence Tags, from a number of organisms. A brief account of the history of human ESTs in GenBank is available. For more details see [40].

UniGene is a NCBI project which contains thousands of sequences of well-characterized genes from different living organisms. It provides a set of transcript sequences that appear to come from the same transcription locus (gene or expressed pseudogene), together with information on protein similarities, gene expression, cDNA clone reagents, and genomic location. In addition, it includes hundreds of thousands novel expressed sequence tag (EST). More details in [243].

SAGEmap is a SAGE data resource for the query and retrieval and analysis of SAGE data from any organism. This data resource supports the public use and dissemination of serial analysis of gene expression (SAGE) data. SAGEmap is a NCBI project. For more details in see [173].

STACK project aims at generating a comprehensive representation of the sequence of each of the expressed genes in the human genome by extensive processing of gene fragments to make accurate alignments, highlight diversity and provide a carefully joined set of consensus sequences for each gene. For more details see at [211]

RefSeq. The Reference Sequence collection aims to provide a comprehensive, integrated, non-redundant set of sequences, including genomic DNA, transcript (RNA), and protein products, for major research organisms. RefSeq standards serve as the basis for medical, functional, and diversity studies; they provide a stable reference for gene identification and characterization, mutation analysis, expression studies, polymorphism discovery, and comparative analyses. RefSeqs are used as a reagent for the functional annotation of some genome sequencing projects, including those of human and mouse. For more details see in [244].

Genome databases

These databases are focused in the genome of a variety of organisms containing information as: genomes, complete chromosomes, sequence maps, and integrated genetic and physical maps.

They exist several important organism-genome databases, among them we can cite: Genome Project , Genomes, Human Genome Database (GDB), Ensembl project, Mouse Genome Consortium, Flybase, TIGR, Saccharomyces Genome Database (SGD) and many others organism genomic databases. In the following, we briefly explain these genomic databases.

Genome Project database is intended to be a searchable collection of complete and incomplete (in-progress) large-scale sequencing, assembly, annotation, and mapping projects for cellular organisms. The database is organized into organism-specific overviews that function as portals from which all projects in the database belonging to that organism can be browsed and retrieved. There is also a special set of resources dedicated to Viral Genomes.

The Genome database provides views for a variety of genomes, complete chromosomes, sequence maps with contigs, and integrated genetic and physical maps. The database is organized in six major organism groups: Archaea, Bacteria, Eukaryotae, Viruses, Viroids, and Plasmids and includes complete chromosomes, organelles and plasmids as well as draft genome assemblies.

GDB: Human Genome Database is the official world-wide database for the annotation of the Human Genome. It contains the whole Human Project sequencing results: approximately 25,000 genes and more than 3 billion of chemical base pairs that make up human DNA. In 2003, when the human DNA was completely sequenced, the GDB stated to collect all worldwide generated annotations of DNA human sequences and genes. It receives direct submissions for gene annotations from scientists who volunteer their records to make

the data publicly available. It uses HUGO nomenclature and several mirror sites in all around the world to improve the submission process.

Ensembl is a joint project between European Bioinformatics Institute (EBI) and the Wellcome Trust Sanger Institute (WTSI) to develop a software system which produces and maintains automatic annotation on selected eukaryotic genomes[153]. It provides repositories, for archiving and updating genome data (sequences and annotations).

Mouse Genome Consortium is a joint project between private and public laboratories including EBI and Sanger Institute for providing the whole Mouse genome sequence, including gene annotations. It uses the Ensembl trace repository for archiving and updating data (sequences and annotations).

Flybase is a database for drosophila genes and genomes. This project is carried out by a consortium of Drosophila researchers at Harvard, Cambridge and Indiana Universities. More details in [136].

Saccharomyces Genome Database is a database of the molecular biology and genetics of the yeast Saccharomyces cerevisiae, which is commonly known as baker's or budding yeast. This project is hosted and managed by Princeton and Stanford Universities. For more details see in [147].

3.3.2 Protein databases

These databases are focused on proteins and contains structural and functional information. Most of the databases are specific and they often contain the expert's annotations. We can divide them into protein-sequence and functional databases and macromolecular structural databases.

Concerning the macromolecular protein data resources it exists the worldwide protein data bank (wwPDB). It consists of organizations that act as deposition, data processing and distribution centers for protein database data. The members are PDB[13], MSD-EBI, PDBj and BMRB. The mission of the wwPDB is to maintain a single Protein Data Bank archive of macromolecular structural data that is freely and publicly available to the global community. For more details see in [148].

Related to protein-sequence and functional databases we can mention: Uniprot, Interpro, IPI , HPI, COGs, CCD, and many others. We briefly explain some of these sequence databases.

Universal Protein Resource (Uniprot) is a catalog of information on proteins It is a central repository of protein sequence and function created by joining the informa-

[13] The RCSB PDB provides a variety of tools and resources for studying the structures of biological macromolecules and their relationships to sequence, function, and disease.

tion contained in three important databases: Swiss-Prot[14], TrEMBL[15], and PIR[16]. For more details see in [79].

InterPro is a database of protein families, domains and functional sites in which identifiable features found in known proteins can be applied to unknown protein sequences.

International Protein Index (IPI) provides a top level guide to the main databases that describe the proteomes of higher eukaryotic organisms. It provides minimally redundant yet maximally complete sets of proteins for featured species and maintains stable identifiers.

Human Proteome Initiative (HPI) aims to annotate all known human protein sequences, as well as their orthologous sequences in other mammals, according to the quality standards of UniProtKB/Swiss-Prot. In addition to accurate sequences, it provides, for each protein, plenty of information that includes the description of its function, domain structure, subcellular location, similarities to other proteins, etc.

Clusters of Orthologous Groups of proteins (COGs) were delineated by comparing protein sequences encoded in complete genomes, representing major phylogenetic lineages. Each COG consists of individual proteins or groups of paralogs from at least 3 lineages and thus corresponds to an ancient conserved domain. For more details see in [302].

3.4 Gene Expression Databases

In the past years, a myriad of gene expression technologies experiments have been conducted, and large quantities of microarray and SAGE data have been made available in public data repositories such as Gene Expression Omnibus [105], Stanford Microarray Database [132] and ArrayExpress repository [47], among other gene expression databases.

Gene Expression Omnibus is a gene expression/molecular public data repository supporting MIAME compliant data submissions and a curated online resource for gene expression data browsing, query, and retrieval. It contains gene expressions data issued from microarray and SAGE technologies. For more details see [105].

Stanford Microarray Database is a gene expression public data repository which contains microarray data submissions from several laboratories in the world. It acts as an online resource for browsing, query and retrieval. For more details see in [132].

ArrayExpress is public database of microarray gene expression data at the EBI, which is a generic gene expression database designed to hold data from all microarray platforms. ArrayExpress uses the MIAME annotation rules. This database can be queried on parameters such as author, laboratory, organism, experiment, or array types. For more details see in [47].

[14] Swiss-Prot is a curated protein sequence database which strives to provide a high level of annotation (such as the description of the function of a protein, its domains structure, post-translational modifications, variants, etc.), a minimal level of redundancy and high level of integration with other databases

[15] TrEMBL is a computer-annotated supplement of Swiss-Prot that contains all the translations of EMBL nucleotide sequence entries not yet integrated in Swiss-Prot

[16] PIR has provided many protein databases and analysis tools freely accessible to the scientific community, including the Protein Sequence Database (PSD), the first international database (see PIR-International), which grew out on Atlas of Protein Sequence and Structure.

3.5 Bibliographic Databases

The last decade has seen the explosion of millions of biomedical and life sciences literature. Only in the case of gene expression technology literature output, thousands of articles have been realized in the last years. These articles contain up-to date data that contain important insights and information about biological and medical issues. It is now important to be able to recover as much as possible of this information as it constitutes a precious source of additional information for helping to understand new genomics data.

Bibliographic databases are databases of bibliographic information containing information about books, articles or other written or online literature material. A bibliographic database is often an electronic index to journal or magazine articles, containing citations, abstracts and often either the full text of the articles indexed, or links to the full text.

The most ambitious and most used bibliographic database in life science is Medline/Pubmed[17], developed and managed by NIH's National Center for Biotechnology Information (NCBI). Pubmed is a free digital archive of biomedical and life sciences journal literature from Medline, which includes over 16 million citations of medicine and other life science journals for biomedical articles back to the 1950s [18].

Medline/Pubmed uses the **Medical Subject Headings (MeSH)** ontology. MeSH is designed to help quickly locate descriptors of possible interest and to show the hierarchy in which descriptors of interest appear. Virtually complete MeSH records are available, including the scope notes, annotations, entry vocabulary, history notes, allowable qualifiers, etc. The MeSH browser does not link directly to any Medline or other database retrieval system and thus is not a substitute for the Pubmed system.

Among other life sciences bibliographic databases we can mention: Biology browser (Biosis), Online Mendelian Inheritance in Man (OMIM), Biobase, etc.

3.6 Gene/Protein-Related specific sources

As we have seen in chapter 1 (section 1.3), gene expression technologies are employed in many life science applications. Among many others, we can mention:

- Medical usage for characterizing diseases, drug fabrication, etc.
- Pathology science for characterizing bacterias and viruses.
- Physiology issues for understanding the complex vital processes in organisms.
- Genetic topics as: mutation screening, genotypic analysis, developmental genetics

Thus, the required information and knowledge to deal with the interpretation issue (fifth step) should come from many different life sciences sources, like medicine, physiology, pharmacology, pathology, anatomy, biology among many others. Gene/protein-related specific sources are all information sources coming from different domains of knowledge which are related to gene/protein issues. They are generally databases that contain information

[17] Pubmed browser at: http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?DB=pubmed
[18] Pubmed information in: http://www.pubmedcentral.nih.gov/

about genes, proteins or even genomes, but they have to include other related information as: diseases, anatomy, pharmacology, physiology, etc.

There exists a large choice of gene/protein-related specific sources, among them we can mention: KEGG , GeneCards, Human Developmental Anatomy, Cancer Genome Anatomy Project (CGAP), BioGRID, there exists many other gene/protein-related specific sources. We briefly explain these databases

KEGG is a "biological systems" database integrating both molecular building block information and higher-level systemic information. Molecular building blocks are distinguished between genetic building blocks (KEGG genes) and chemical building blocks (KEGG ligand), while the systemic information is represented as molecular wiring diagrams (KEGG pathway) and hierarchies and relationships among biological objects (KEGG brite).

KEGG pathway are manual pathway maps representing our knowledge on the molecular interaction and reaction networks for metabolism, other cellular processes, and human diseases.

KEGG brite are functional hierarchies and binary relations of KEGG objects, including genes and proteins, compounds and reactions, drugs and diseases, and cells and organisms.

KEGG genes are gene catalogs of all complete genomes and some partial genomes with orthologous annotation (KO assignment), enabling KEGG PATHWAY mapping and BRITE mapping.

KEGG ligand is a composite database of chemical substances and reactions representing our knowledge of the chemical repertoire of biological systems and environments

More information about the foundations of KEGG can be found at [161].

GeneCards is an integrated database of human genes that includes automatically-mined genomic, proteomic and transcriptomic information, as well as orthologies, disease relationships, SNPs, gene expression, gene function, and service links for ordering assays and antibodies. For more details see [253].

Human Developmental Anatomy is a Human Atlas designed to identify those tissues present in human embryos during the first 50 or so days of the development. The Atlas can be viewed in two formats. The first, is the basic anatomical data designed to provide standards for analyzing normal and congenitally abnormal human embryos. The second is a more detailed format that includes additional tissue regions. The second format is intended to be compatible with the anatomical terminology and organization used for storing gene-expression data in the Mouse Atlas Project. For more details see in [154].

Cancer Genome Anatomy Project (CGAP) is a database of Chromosome Aberrations in Cancer relating chromosomal aberrations to tumor characteristics, based either on individual cases or associations. For more information see in [213].

BioGRID is a database of physical and genetic interactions from *Saccharomyces cerevisiae*, *Caenorhabditis elegans*, *Drosophila melanogaster* and *Homo sapiens*. For more information see in [293].

3.7 Semantic Sources: Semantic Networks, Ontologies, Thesaurus and Taxonomies.

In order to fully exploit the enormous accumulated biological knowledge, life sciences as biology have been forced to structure their information in databases and repositories (as stated in last five sources of information explained in this chapter). The complexity and high correlated information contained in these databases and repositories revealed the need of a formal representation of these information based on a well-defined semantic [282].

The thirst for knowledge in life sciences have accelerate the creation of several knowledge representation[19] sources or semantic sources. The semantic sources are well-defined and structured collections of concepts. As we have defined in the introduction of this chapter, they can be of several types: *taxonomies*, *thesaurus*, *ontologies* and *semantic networks.* There exist several biological and medical semantic sources, between the most used in gene expression technologies are Gene Ontology (GO), UMLS, NCBI Taxonomy , LinKBase, among others. We'll briefly explain some semantic sources.

Gene Ontology (GO) is a collaborative effort developed by a consortium of scientists to generate a controlled vocabulary of various genomic databases about diverse species in such a way that it can show the essential features shared by all the organisms. It can be used to annotate genes by a GO-term, with regard to its molecular functions (GO:MF), cellular localizations (GO:CL) and biological processes (GO:BP).

GO-terms are organized in structures called directed acyclic graphs (DAGs), which differ from hierarchies in that a child can have many parents, or less specialized, terms. This structure also allows annotators to assign proper-ties of genes at different levels, depending on how much is known about a gene [12].

Unified Medical Language System (UMLS). The purpose of UMLS is to enhance access to medical literature by facilitating the development of computer systems that understand biomedical language. This is achieved by overcoming two significant barriers: "the variety of ways the same concepts are expressed in different machine-readable sources by scientists" and "the distribution of useful information among many disparate databases and systems". Three main tools are used to accomplish this: metathesaurus, semantic network, and specialist lexicon.

The metathesaurus forms the base of the UMLS and is comprised of over 1 million biomedical concepts and 5 million concept names, all of which are from over 100 controlled vocabularies and classification systems used in patient records, bibliographic, administrative health data and full text databases

Semantic networks are knowledge representation schemes involving nodes and links (arcs or arrows) between nodes. The nodes represent objects or concepts and the links represent relations between nodes. This graphical representation assists in understanding the relationships of concepts.

[19] Knowledge representation is an issue that arises in both cognitive science and artificial intelligence. In cognitive science it is concerned with how people store and process information. In artificial intelligence (AI) the primary aim is to store knowledge so that programs can process it and achieve the capacities of human intelligence.

The Specialist lexicon help end users work through the variations in biomedical texts by relating words by their parts of speech, which can be helpful in web searches or searches through an electronic medical record. For more details see in [39].

The NCBI taxonomy contains the names of all organisms that are represented in the genetic databases with at least one nucleotide or protein sequence. It uses a tree for browsing the taxonomic structure or retrieve sequence data for a particular group of organisms.

LinKBase is one of the largest formal medical ontologies, i.e. a conceptual computer-understandable representation of medicine. LinKBase contains more than 1,000,000 concepts and over 7,000,000 knowledge objects. This ontology is a formal conceptual description of the medical domain and as such is rendered machine-readable by a computer.

Summary

We have fully explained the six sources of available information to achieve the fifth step of gene expression technology: the knowledge discovery via results interpretation.

Part II: Introductory Works and Knowledge Discovery Interpretation Approaches

Chapter 4

First Application Works: SAGE and Microarray Data Analysis

In this chapter, we present a full five-step data analysis of gene expression data issued from two different technologies: SAGE and Microarray with spotting oligos-chip. For each technology, we follow the five-step data analysis procedure, i.e. data generation, statistical data treatment, analysis of differentially expressed genes, classification of the genes and knowledge discovery via data interpretation (as seen in chapter 2) . Each step has been realized by applying a selection of the currently available tools and methods.

Serial Analysis of Gene Expression (SAGE) data give counts of occurrences of nucleotide sequence tags in several tissue samples and at different stages of development. Each tag can identify a gene, and the SAGE method aims at giving an overview of a cell's complete gene activity (as explained in section 1.2.2). Nowadays, SAGE data have been poorly exploited by clustering analysis due to the lack of appropriate analysis methods that consider their specific properties.

We have participated in the PKDD 2005 Discovery Challenge[20] by analyzing a gene expression data set containing expression levels of 822 SAGE tags in 74 tissue samples (biological conditions) originating from 10 different human normal and cancer tissues. We have shows that cleaning the data set (tags and experiments) is critical and that attribution of a tag to a gene is not easy.

Using several analysis techniques for the hierarchical clustering of SAGE expression data set, we confirm that, if experiments are well conducted, SAGE analyses can provide insights on understanding which genes are specifically expressed according to cell states and culture conditions. We have concluded that the comparison of cancers from various tissues is a difficult task as tissue samples cluster according to tissue origin and not as cancer versus normal. Besides many traps lay on the path to these discoveries and a careful selection and analysis of these data are required. For more details the reader can refer to [199].

Spotted oligos-chip technology measures the gene expression levels of thousands of genes via the sequencing by hybridization techniques. This technique measures the transcription of genes under tens of biological conditions (explained in section 1.2.1). However, there are many sources of variation along the whole experimentation process within this technology (explained in section 4.2.1).

[20] Principles and Practice of Knowledge Discovery in Databases (PKDD) held in Porto, Portugal in October 2005. The Discovery Challenge was the complete analysis of a Cancer SAGE Dataset for obtaining biological insights.

We have analyzed the Azerty gene expression data set provided by IPMC laboratory which contains the gene expression measures of 22739 genes taken over five time points. This experiment was replicated over 6 chips, and the biological process was intentionally hidden during the data analysis.

Applying several statistical and data mining tools we have discovered significant groups of co-expressed and co-annotated genes. We have revealed the importance of the data treatment step and showed the lack of tools for manipulating time series data. We have concluded that the integration of the biological knowledge must be present along all the data analysis procedure especially in the gene clustering step. In this manner, we can arrive at the last interpretation and knowledge discovery step with meaningful biological results.

This chapter is divided in two main parts: Discovery Challenge data set, that we named SAGE Multicancer data set, analysis, and the Azerty data set analysis.

4.1 SAGE Multicancer Data Set Analysis

This section is divided in 8 subsections. The introductory section explains briefly SAGE basics, the Multicancer data set characteristics and the Multicancer data analysis difficulties. Then, the next five sections correspond to each one of the five data analysis steps, explained in chapter 5 (including the data analysis results). The last section discusses each of the data analysis steps applied to Multicancer data set and gives a general conclusion. The information contained in this section was obtained from the publication: Exploratory Analysis of Cancer SAGE Data [199].

4.1.1 Introduction

SAGE basics

The SAGE method for detection of mRNA transcripts in eukaryotes is based on the sequencing of concatemers of short (14 base-pairs; recently 17 bp.) sequence tags* that originate from a known position (after the 3'-nearest cutting site of a restriction enzyme) to estimate transcripts abundance [311].

In contrast to microarrays, the SAGE method estimates the expression level of transcripts without prior knowledge of their sequences and is more sensitive than the EST* method [297], but requires knowledge of the complete genome. The advantage of the SAGE method is that it performs a random sampling of transcripts in a particular tissue with little sequencing effort.

The nature of the data enables the creation of large public SAGE data sets for numerous tissues, both normal and cancerous [265] as well as specific tools to analyze SAGE data. SAGE is perhaps less suitable than DNA chips for high-throughput analyses of multiple samples, but does not require the expensive equipment required to deal with DNA chips. More details in the SAGE gene expression technology characteristics were explained in section 1.2.2.

Multicancer data set description

The multicancer data set[21] contains the expression levels of 822 SAGE sequence tags, collected from 74 SAGE libraries corresponding to different tissue samples (biological conditions) originated from 10 different human normal and cancer tissues. Each of the 74 SAGE libraries contains the minimal information required for SAGE experiments as stated by the NCBI-GEO standards. We have extracted three relevant informations from each one of the 74 libraries: the type of tissue the cells come from, the state of the cells: cancer (C) or normal (N) and the source of the cells: bulk (Bu) or cell line (Ce) as shown in Table 4.1. Bulk source corresponds to tissue samples taken in vivo, and cell line source identifies cells that are indefinitely reproduced in culture.

Tissue	Cancer bulk	Cancer cell line	Normal bulk	Normal cell line	Total
Brain	8	7	5	1	21
Breast	6	3	2	0	11
Colon	2	4	2	0	8
Kidney	0	2	0	0	2
Ovary	3	4	0	2	9
Pancreas	0	3	2	2	7
Prostate	3	6	2	0	11
Peritoneum	0	0	1	0	1
Skin	1	0	0	1	2
Vessel	0	0	0	2	2
Total	23	29	14	8	74

TABLE 4.1: Repartition of the 74 SAGE libraries by cell state, kind of tissue and the cells source.

Multicancer data set difficulties

The Multicancer data set described before comprise several difficulties:

1. SAGE tags may not all be significant. PCR and sequencing errors may produce a number of errors. A single error may lead to non recognition of a transcript or wrong attribution. Some tags may be present in more than one gene. Finally, since restriction enzymes may not cut with 100 % efficiency, some tags may be wrong.
2. Each tissue sample in the data set may originate from two different sources (see Table 4.1) that may influence gene expression. Cancerous tissue samples are usually provided after surgery, a "cancer" sample may contain more healthy tissue than cancer, leading to a "wrong" condition classification.
3. Large scale analyses using DNA chips concluded that cancer cells resemble more to normal cells of the same tissue than cancer cells from a different tissue: there are many more tissue-specific genes than genes involved in cancers [259]. Thus, trying to classify all conditions in two classes, normal or cancer, in order to identify specific tags using a decision tree cannot be successful. Also, cancers may have different origins (deregulation

[21] This dataset was prepared at the Centre de Génétique Moléculaire et Cellulaire, Université Lyon I, France, using information from the SAGE map resources repository. This dataset is a sample of the original dataset that contains expression levels of 27,679 sequence tags and 90 SAGE librairies.

of oncogenes versus breakdowns of chromosomes for example), so searching for two classes only may be problematic.

Multicancer data set analysis goal and five-step procedure

The main goal of our analysis was to determine if a model that is pertinent to distinguish cancerous and non cancerous biological conditions can be generated. In the next sections, we apply the five-step procedure for the analysis of the Multicancer data set that consists of:

1. Data Generation: pruning of non-significant tags.
2. Statistical Data Treatment: normalization of biological conditions.
3. Selection of differentially expressed genes and selection of biological conditions, i.e., taking off the biological conditions which behave as outliers.
4. Clustering biological conditions.
5. Knowledge Discovery via Data interpretation.

4.1.2 First step data generation

As explained in section 1.2.2 SAGE experiments consist of four steps: sample preparation, building tags and concatemers, concatemers sequencing and tags-genes correspondence (see FIG. 1.18). The first three steps are already done by the experimenter (as stated in the 74 SAGE libraries). The last step which involves matching the sequence of each tag with the gene that has produced the transcript, has to be done by the data analyst.

Tag-transcript attribution

SAGE tags are often annotated based on the SAGE Genie principles [44] and linked to a series of expression data (often EST sequences). This step is difficult to automate and it is often complicated to understand and appreciate the methods used for tag attribution, we therefore developed specific tools.

In a first step, every human ENST sequence[22] was downloaded from Ensemble. Tags present in transcripts of a single gene were labelled as good tags (436), each was assingned its corresponding ENSG number. Tags present in transcripts originating from several genes were labeled as bad tags (219) and removed from further analysis.

Next, all EMBL human sequences (including ESTs) were downloaded to search non attributed tags (167). Every sequence recognized was blasted* for ENSG attribution. This step led to a further 80 tags attributed to a ENSG number. Reasons for tag non attribution are likely to be: i) location in a region not yet identified as a gene, ii) location in the mitochondrial genome (very few protein coding genes), which was not taken into account. and iii) tag resulting from the partial digestion of a transcript, and therefore not located in the 3' end of the sequence domain.

[22] ENST is Ensembl sequence transcript. The ENST databases can be found at the ENSEMBL site.

At this point we had clearly less tags linked to genes than if we had used a tool such as SAGE Genie or other tools. For example, we have take the first tag on the Multicancer data set, i.e. $AAAACATTCT$, it was linked to a mitochondrial sequence by SAGE Genie, while at the Global Gene Expression Group (GGEG) project it mapped to Unigene Hs.476965 (G1/S transition control protein-binding protein IEF-8502). There is clearly still a lack of precision in the procedure of tag attribution. SAGE Genie links this tag to a sequence of accession number BE874599. Blast* of this sequence provided a hit on the mitochondrial human genome, but at a position that was identified as '16S ribosomal sequence'. Such sequence has no polyA tail of any sort, and does not contain a repeat of A anywhere in the sequence [23]. Therefore, we are rather confident that every data resulting from large scale analysis using web based tools should be critically assessed either using two different public tools or ad hoc scripts and database.

The output of this first step is the recognition of 516 sequence tags as unique genes. From now, we refer to every tag as it correspondent unique ENST transcript as gene identification.

4.1.3 Second step: scaling

One of the advantages of the SAGE technology over the microarray technology is that since it relies only on a sampling process it is "self-normalized", and therefore it can directly be used for data analysis purposes. For example, a SAGE library constructed on a brain cell in France can be compared, without any specific normalization process, with a SAGE library constructed on a brain cell in the US.

However, SAGE libraries contain the expression of the gene by counting the number of occurrences in a time interval, so every library has different numbers of total genes that have been expressed. For comparison purposes, we have to avoid this bias from the data by applying a scaling procedure.

We suppose that the Multicancer data set is represented as a matrix X with 516 genes (ENST transcripts) and 74 biological conditions (SAGE libraries), where $x_{i,j}$ represents the current expression measure of gene i in biological condition j. We apply the following scaling factor to every gene expression measure (row) and for each biological condition (column) of the matrix X:

$$x^s_{i,j} = x_{i,j} * \frac{Max_j(x_{i,j})}{\sum_{i=1}^{516} x_{i,j}}, \tag{4.1}$$

where $x^s_{i,j}$ is the scaled expression measure of gene i in biological condition j and $Max_j(x_{i,j})$ is the maximum expression measure in the biological condition j.

We have applied equation 4.1 to every one of the 74 columns of the matrix X, obtaining the matrix X^s that corresponds to the matrix with the scaled gene expression measures. The implementation of 4.1 was made in R language using the SAGElyzer library from bioconductor project.

[23] This a clear demonstration that large scale tools such as proposed are not exempt of problems.

4.1.4 Third step: selecting differentially expressed genes and pertinent biological conditions

Gene selection

The goal of this step is to detect genes that are over-expressed or under-expressed, since the genes that are constant or low expressed in both conditions (cancer vs. normal) could introduce noise to our main goal. This task is gene selection, where we take out the unexpressed genes in order to find essential information about the state of the tissue (cancerous or not). The selection step and several methods to solve it was fully explained in section 2.2.

In our Multicancer data set, we have used the Significance Analysis of Microarrays (SAM) method to select differentially expressed genes. SAM is based on a modified $t - statistic$ applied for every gene i. This statistic measures the strength of the relationship between gene expression and the response variable (cancer bulk, cancer cell line, normal bulk and normal cell line) using repeated permutations of the data. The significance cutoff, δ, is determined by a tuning parameter, chosen by the user and based on the false positive rate. A brief description of this method can be found in section 2.2.5 and a full explanation in Tusher et al. [309].

We have chosen the cutoff of $\delta = 0.21$, which implies a *false discovery rate* of 5 %, which represents the standard cutoff of gene expression technologies for gene selection. The output at this stage was a matrix with 247 differentially expressed genes and 74 biological conditions.

Selection of the pertinent biological conditions

It is critical to take into account biological condition variations and in particular possible *sample outliers* which introduce noise in the clustering procedure [188]. In order to avoid this noise, we developed an algorithm for finding sample outliers among biological conditions using two tools: *Principal Component Analysis* (PCA)* and hierarchical *clustering* approaches (explained in section 2.3.3). Our original method can be resumed in four general steps:

1. Using PCA as an exploratory tool to determine the optimal number of clusters.
2. Applying an hierarchical clustering algorithm to identify sample outliers and remove them.
3. Then, applying again PCA analysis to verify that the variability level is not decreased when each of these biological conditions is removed.
4. Applying hierarchical clustering algorithm again to verify that the clustering was improved. In this step we can use different cluster validation approaches or a clustering visualization tool (explained in section 2.3.6).

Our algorithm depends on a correct selection of a clustering algorithm and their respective distance measure. In the case of the Multicancer data set we tested five clustering algorithms: K-Means, Fanny, Partial Least Squares, Unweighted Pair Groups Method Average (UPGMA) and DIvisive ANAlysis (DIANA) and five measures of distance: Euclidean,

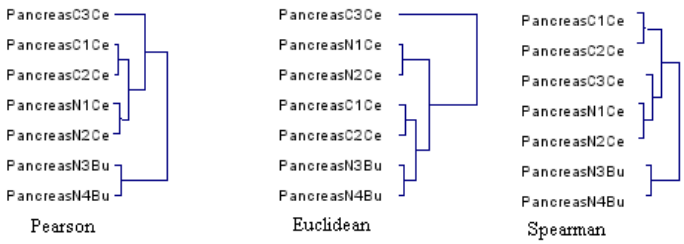

FIG. 4.1: Hierarchical clustering of the pancreas conditions

Pearson, Manhattan, Spearman and Tau according to three different consistency measures: average proportion of nonoverlap, average distance between clusters and average distance between cluster-means. These consistency measures were proposed in Datta et al.[85] (more details on these measures in section 2.3.6).

The testing results showed that the hierarchical clustering algorithms UPGMA and DIANA with Pearson, Euclidean and Spearman distance measures, were more efficient in clustering our Multicancer data set (divided by tissue) in at least two of the three consistency measures cited above (results not shown).

Thus, we decided to apply our four-step algorithm (described before) with the hierarchical algorithms UPGMA and DIANA, the average linkage measure and the Spearman, Euclidean and Pearson distances. We used these algorithms were used on tissue-specific subsets of the Multicancer data set, that is, subsets which are composed only of samples of one tissue: brain, breast, colon, pancreas etc. (see TABLE 4.1). For the sake of brevity we only explain the pancreas tissue selection procedure.

The seven pancreas conditions are distributed in 3 classes: cancer cell line (C1Ce, C2Ce and C3Ce), normal cell line (N1Ce and N2Ce) and normal bulk (N3Bu and N4Bu), see TABLE 4.1. The PCA analysis shows that the first 3 PCA components explain 98.59 % of the total variance, thus indicating that there is little noise in the data. The hierarchical trees obtained for the different distance measures (Pearson, Euclidean and Spearman) are shown in FIG. 4.1. The trees obtained with the UPGMA and the DIANA algorithms were identical, see FIG. 4.1.

When using the Pearson and Euclidean distance measures, condition PancreasC3Ce is placed in an isolated cluster, and when the spearman measure is used, it is associated with normal conditions (reducing the accuracy of the clustering). So we take out this condition and we run PCA analysis again, obtaining that the first 3 components now explain 99.03 % of the total variance (improving the total variance). Finally, a clustering on the data set without this outlier condition was conducted and the hierarchical clustering tree obtained by consensus for three distance measures as shown in FIG. 4.2.

Results obtained for other tissues are summarized in TABLE 4.2; the 16 outlier conditions are listed. These results confirm the natural division of conditions in three classes corresponding to the first 3 components of PCA analysis. Furthermore, in all cases we can see that the variance explained by the first 3 components is always improved, up to 4.31 % for Ovary conditions, when outlier conditions are removed.

FIG. 4.2: Hierarchical clustering after outlier pancreas conditions pruning

Organ/Tissue	PCA (first 3 components)	Outliers	PCA without Outliers (first 3 components)
Brain	98.46 %	{N4Ce,C1Bu,C14Bu, C5Bu, C9Ce}	99.02 %
Breast	95.57 %	{C6Bu}	97.38 %
Colon	98.56 %	{}	98.56 %
Ovary	93.60 %	{N1Ce, N2Ce, C4Ce, C6Bu}	97.91 %
Prostate	98.02 %	{N1Bu,C7Bu,C9Bu, C8Ce, C1Ce}	98.70 %
Pancreas	98.59 %	{C3Ce}	99.03 %

TABLE 4.2: PCA analysis of conditions by tissues.

We can see that the clustering of conditions gives a partition by cell source first, i.e. bulk and cell line, and then by cell state, i.e. cancerous and normal (as shown in the case of pancreas FIG. 4.2). This observation therefore confirms previous analysis that showed cell source to be of crucial influence on gene expression.

This Multicancer data set was further reduced by removing 7 conditions related to 4 different tissues: SkinC1Bu, SkinN1Ce, VesselN1Ce, VesselN2Ce, PeritoneumN1Bu, KidneyN1Ce and KidneyN2Ce (see TABLE 4.1). These conditions do not constitute sufficient information for analysis among these tissues and thus will be isolated and will act as outliers, introducing noise in the fourth analysis step: clustering the entire data set. The output at this stage is a gene expression matrix X^s with 247 differentially expressed genes and $74 - 16 - 7 = 51$ biological conditions distributed in six tissues: brain, breast, colon, ovary, pancreas and prostate.

4.1.5 Fourth step: clustering of the biological conditions

After cleaning the Multicancer data set as explained in section 4.1.4, we then applied the UPGMA and DIANA clustering algorithms, obtaining for both algorithms and Pearson and Spearman distances identical results (shown in FIG. 4.3). However, for Euclidean distance, the distribution was similar but with longer branches to the leaves.

Comparing clustering trees obtained with the initial data set and FIG. 4.3 clearly showed that the selection process improved data quality since length of terminal branches were considerably reduced. We can observe a first degree classification by tissue that is accurate for Pancreas, Brain, Breast, Colon and Prostate tissues, but mixes Ovary tissue conditions with other tissue conditions. We can also see a clear second degree classification, among conditions of the same tissue, by cell source: bulk and cell line. Among Pancreas, Breast, Brain

and Colon condition clusters, we can observe a third degree classification by state: cancer and normal.

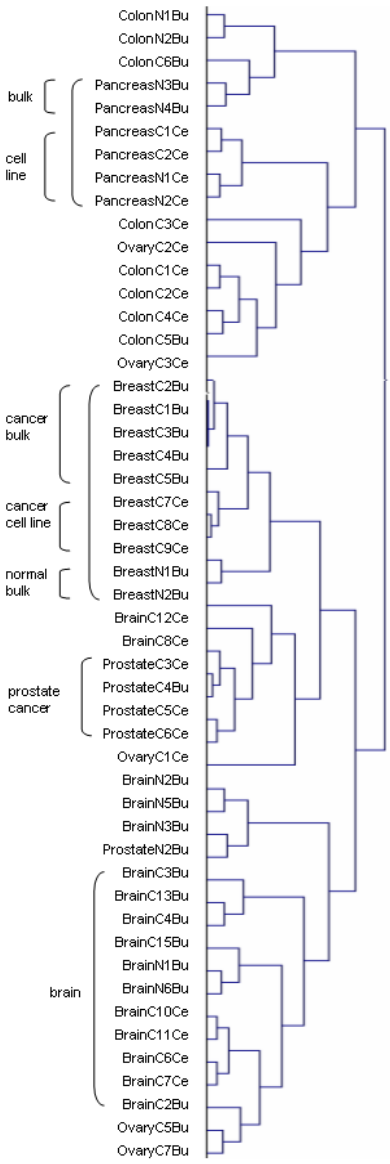

FIG. 4.3: Hierarchical clustering of conditions on the cleaned Multicancer data set.

In conclusion, clustering clearly separates cell sources corroborating previous results on SAGE and DNA chips data [221, 259]. We can conclude that there are important differences between bulk and cell line conditions that should not be ignored. When conducting studies for finding "interesting gene cancer knowledge" involving multiple tissues SAGE libraries, the study must be first oriented toward a decomposition of the conditions by tissues and then by cell sources to finally focus the analysis on cell states. Hence, two cells of the same tissue

with different states (cancerous and normal) are more similar than cells of different tissues with the same state (as seen in FIG. 4.3).

To validate our results, we applied the Quinlan's supervised algorithm C5.0 [247] to produce classification rules of biological conditions by tissue, cell state and cell type. Algorithm C5.0 generates a classifier represented either by a decision tree or a set of "if-then" rules.

Three different class attributes characterizing each condition were created: tissue type (Pancreas, Ovary, Brain, Prostate and Breast), cell source (bulk or cell line) and cell state (cancer and normal). Boosting* and cross validation* options were activated. The numbers of rules with maximal accuracy generated for each class decomposition of conditions are shown in TABLE 4.3.

Class	Number of rules	Max accuracy
Bulk	5	100 %
Cell line	5	100 %
Cancer	1	80 %
Normal	3	80 %
All 6 tissues types	1	60 %

TABLE 4.3: Rules by class and their maximal accuracy.

Using the cell source classification, 5 exact rules, i.e. with perfect accuracy, were generated. For the cell state classification, only 1 and 3 rules respectively, all with only 80 % of accuracy, were generated. Considering tissue classification, only 1 rule with 60 % accuracy was generated. This result is logical since there are 6 different tissues, thus disturbing the classification, and cells from different tissues but originating from cell lines tend to become more similar from the tag expression levels viewpoint. These results confirm that in the small cleaned data set, there is an intrinsic division of conditions by cell source that is more natural than by cell state.

Implementation

The clustering algorithms were obtained from the cluster library in the BIOCONDUCTOR open source project, the graphics for hierarchical clustering outputs were obtained by using Genesis program (more details in [296]). In the final step, the classification of biological conditions was performed using the SPSS CLEMENTINE implementation of C5.0.

4.1.6 Fifth step: knowledge discovery via data interpretation.

The resulting Multicancer data set containing 247 genes and 51 tissue samples contains an inherent decomposition first by tissue, then by sample source type and finally by tissue state. Thus, this data set is not adapted to accomplish our main goal of finding a pertinent model for distinguishing cancerous and non cancerous type.

In order to illustrate the fifth step, we have taken the cleaned Multicancer data set (containing 247 genes and 51 tissue samples) and we established an alternative goal of finding at least one co-annotated and co-expressed cluster of genes. We will briefly explain our methodology:

4.1.6 Fifth step: knowledge discovery via data interpretation.

First, we applied the three more consistency clustering algorithms for Multicancer data set: UPGMA, DIANA and Self Organizing Maps (SOM) with pearson distance (this choice was explained above in the fourth analysis step). Here, the goal is to find clusters of genes which contains similar expression profiles. Thus, genes are objects and biological conditions are attributes (inversely of the four analysis steps realized above).

Then, we have realized several scenarios, varying the possible number of clusters $k \in \{1, 2, 3, ..., 20\}$. For each one of the three algorithms, we have found the optimal number of clusters k applying the two measures proposed by Sharan et al. [276]: average homogeneity, H_{ave}, and average measure S_{ave} (fully explained in section 2.3.6). Hence, the k that gives the biggest average homogeneity within each cluster H_{ave} and the largest average separation S_{ave} among clusters was taken as optimal for each algorithm. In this case: UPGMA was $k = 10$, DIANA was $k = 8$ and SOM $k = 13$.

The next step concerns finding the co-expressed *hard clusters*, that is groups of co-expressed genes that appear in the intersection of two or three clusters among all the clusters or each of the three methods. Then, we fix a pruning factor for hard cluster selection of 75%. Let us look at two clusters: cluster five obtained by UPGMA, C_5^{Upgma}, and cluster three obtained by SOM, C_3^{SOM}, with cardinalities 25 and 37 respectively. If the cardinality of the intersection $\left|C_5^{Upgma} \cap C_3^{SOM}\right| = 20$, then calculating the pruning factor as:

$$Max\left(\frac{\left|C_5^{Upgma} \cap C_3^{SOM}\right|}{C_3^{SOM}}, \frac{\left|C_5^{Upgma} \cap C_3^{SOM}\right|}{C_5^{Upgma}}\right) * 100\% = 80\% \succeq 75\%, \qquad (4.2)$$

so the hard cluster $\left|C_5^{Upgma} \cap C_3^{SOM}\right|$ is chosen for further analysis.

The last step is to build the co-annotated gene groups from the selected co-expressed hard clusters and to test the significance of the co-expressed and co-annotated gene groups. For building the co-annotated groups, we used the web tool FATIGO [5], which finds significant associations of Gene Ontology terms with groups of genes.

As an illustrative example we present one of these selected co-expressed hard clusters: $HC_7 = \{C_5^{Upgma} \cap C_3^{SOM} \cap C_2^{Diana}\}$, which contains 7 human genes. In order to build the co-annotated gene groups, we run the FATIGO tool with two gene lists: the reference list containing 247 human genes and the list containing 7 genes of HC_7 cluster. We used only the biological process ontology contained in the semantic source Gene Ontology (explained in section 3.7). On the first list, only 162 genes had a biological process annotation, and only 6 genes in the second list. Taking a significant level $\alpha = 9.9X10^{-02}$ and a gene ontology level* of 4 we have chosen all the subsets with $p - value$ smaller or equal than α. The results for HC_7 are resumed in Table 4.4.

The first column of Table 4.4 represents the gene ontology level, the second the biological process GO annotation, the third contains the corresponding subset with the ensembl identity, the fourth contains the percentage of annotated genes in relation to the cardinality of the cluster HC_7, and the last column shows the $p - value$ of the significant subset.

From Table 4.4 we can extract that the genes $ENSG - 111786$ and $ENSG - 147677$ are highly correlated by annotation and by expression profile. Also, they are an active set

GO	Annotation	Genes	%	p-value
4	ribonucleoprotein complex biogenesis and assembly	ENSG-111786 and ENSG-147677	33.33	6.13E02
4	macromolecule complex assembly	ENSG-111786 and ENSG-147677	33.33	7.44E-02
5	protein-RNA complex assembly	ENSG-111786 and ENSG-147677	33.33	1.18E-02
5	macromolecule biosynthetic process	ENSG-111786 and ENSG-147677	33.33	2.61E-02
5	cellular protein metabolic process	ENSG-111786 and ENSG-147677 and ENSG-114902	50	6.93E-02
6	regulation of cellular biosynthethic process	ENSG-111786 and ENSG-147677	33.33	1.69E-02

TABLE 4.4: Significant co-annotated and co-expressed genes groups of cluster HC_7 with their respective GO annotation and significance level.

of genes along the six biological process indicated in the annotation column of TABLE 4.4. However, we cannot distinguish their participation in the cancerous or normal tissues. To do so a *bi-clustering technique*, that is clustering by subsets of biological conditions needs to be implemented.

Implementation

The clustering algorithms were obtained from cluster library in BIOCONDUCTOR open source project. The consistency measures and graphics were programmed in R language.

4.1.7 Discussion

The first three steps of SAGE data analysis concerning: data generation (pruning of non-significant tags), statistical data treatment (normalization of biological conditions) and differentially expressed genes and biological conditions selection were crucial for cleaning the inherent noise in the Multicancer SAGE data set.

Replicated concatemers have to be eliminated first in order to estimate the number of replicated di-tags produced by the PCR* amplification. Most SAGE studies made use of tags of 14 bp. However, a recent study showed the clear advantage of using a tag of 15 bp [93]. Even longer tags will be better. Recently, the SAGE protocol was enhanced with a new tagging enzyme (MmeI), which produces 21-22 bases tags [265], allowing direct mapping to the transcripts [312]. When numerous tags are available, it is possible to remove tags that are present only once, and that may result from errors. Sequence errors have little effect on the quantification of moderately expressed genes but a lot for rare transcripts. About 6.7 % of long SAGE di-tags acquire mutations prior to ligation, cloning and sequencing [312], arguing for a robust tag attribution to a transcript.

Only reliably annotated tags can be included in the final analysis [280]. Annotation of SAGE tags to genes and their corresponding Unigene* cluster numbers revealed that on average only 30 % of all tags (including less abundant tags) could be reliably annotated based on

the SAGE Genie principles [44]. Annotation improved to about 70 % for tags with intermediate to abundant expression levels. Remaining tags either could not reliably be associated with a gene (e.g. annotated to unclustered ESTs) or were not present in a single gene.

Selection of differentially expressed genes and biological conditions pruning have considerably decreased the inherent noise in Multicancer SAGE data set. These steps have allowed to distinguish the natural decomposition of the biological conditions by tissues and then by cell sources to finally focus the analysis on cell states. We can conclude that there are important differences between bulk and cell line conditions that should not be ignored. When conducting studies for finding "interesting gene cancer knowledge" involving multiple tissues SAGE libraries, the study must be first oriented toward a decomposition.

Concerning clustering the biological conditions, we have to choose carefully the clustering algorithm and distance measure. This decision has to be done by testing with different clustering algorithms and distances measures and then validating the results with cluster validation methodologies (as seen in section 2.3.6). Thus, we find the "best adapted" algorithm and measure to our particular data set.

With regard to the interpretation step, it depends on two main parameters, the result of the precedent steps and the availability and flexibility of the sources of biological information. In the case of the Multicancer data set, the inherent decomposition of the set first by tissue, then by sample source type bulk or cell line and finally by tissue state cancerous or normal has blocked the possibility of interpreting this data set. The large variability of Multicancer data set make it difficult to interpret in the case of our two goals: building a model for distinguishing cancerous and non cancerous type, and finding co-expressed and co-annotated significant groups.

4.1.8 Conclusion

Algorithms used to analyze SAGE data have a strong influence on results [93], and using a single analysis scenario and a single source of sequence data (annotations) would result in a weaker analysis. We have also shown incoherence of results between different public web tools, and an obvious error of gene attribution for the first tag at least. As for DNA chips data, the Bioconductor* R package call SAGElyzer provides basic essential tools. Removing outlier experiments also decrease noise and increases reliability of clustering. Finally, we have shown that if knowledge rules for cancer are sought for, it is difficult to analyze data sets including different tissues as decisional rules of maximal accuracy are those discriminating tissue origins but not normal versus cancer tissues. Thus, more samples from a single tissue should be more efficient.

4.2 Spotted oligos-chip. Microarray Technology

This section is divided in 8 subsections. The introduction section explains briefly the spotted oligos-chip technology basics, the Azerty data set characteristics, the main goal of the analysis and the data analysis difficulties. Then, the next five sections correspond to each one of the

five data analysis steps (including the data analysis results). The last two sections concern a discussion of each one of the data analysis steps applied to Azerty data set and a general conclusion. The information contained in this section was obtained from the internal report: Azerty data Analysis [197].

4.2.1 Introduction

Spotted oligos-chip basics

The spotted oligos-chip concerns the sequence by hybridization microarray technology which manufactures the chip by robotic spotting of already synthesized oligonucleotides* (More details of this technology are explained in section 1.2.1)

For better comprehension of the whole procedure of spotted oligos-chip. experiments the reader can go to section 1.2.1 and Fig. 1.17. In section 1.2.1 we explained the spotted cDNA chips technology. The main difference between cDNA chips and oligos-chip technologies is that the second one uses already synthesized oligonucleotides as probes, instead of cDNA probes (see first column of Fig. 1.17).

The oligonucleotides have the advantage of being probes that bind easily to its complementary target sequence and they can be synthesized up to 160-200 bases [172].

Azerty data set description

Azerty data set[24] is composed of 6 chips with 27648 spots each. Every chip contains the expression levels of 22739 genes (taking out the control measures) measured at five time points 1hr., 3hr., 6hr., 9hr.and 24hr. Every gene expression measure represents the logarithmic ratio, i.e. $log_2 \frac{Cy5}{Cy3}$, of two light intensities the red one, $Cy5$, corresponding to the studied or test biological sample and the green one, $Cy3$, corresponding to the normal or reference sample (more details for this data transformation can be seen in the section 2.1.1).

The studied biological time process was intentionally unknown for us. The main goal is to discover groups of co-annotated and co-expressed genes that might give us some clues of the studied biological process.

Azerty data set difficulties

The Azerty data set described before comprises several difficulties:

1. Spotted oligos-chip presents the analysis drawbacks (explained in section 1.2.1) in relation to other microarray technologies as *in situ* oligos-chip which may contain less inherent noise.
2. The Azerty data set was already statistically treated with image correction methods. Thus, it could already be biased.

[24] Thanks to the "Institute de Pharmacologie Moleculaire et Cellulaire (IMPC)" (CNRS-UNSA) at Sophia Antipolis, France, for providing us with this set of data.

3. The studied experiment is a biological time process. The current microarray tools available to accomplish the five data analysis steps (explained in chapter 2) are not adapted to this kind of time series process. Thus, we may incur loss of information.
4. The Azerty analysis is of "blind" type, which means that the analyst does not know a priori anything of the studied biological process.
5. The data set is of huge size as it contains 6 replicates of 22739 genes and 5 time point measures each.

Azerty data set analysis goal and five-step procedure

The goal of our analysis is to determine relevant biological process appearing in Azerty data set in order to uncover the hidden (for us) studied process. This was realized by building groups of co-expressed and co-annotated gene groups and interpreting the most significative gene groups. We apply the five-step procedure for the analysis of the Azerty data set:

1. Data Generation
2. Statistical Data Treatment: Global normalization and intensity-dependent normalization, and treatment of *technical*[25] and *biological*[26] *replicates*.
3. Selection of differentially expressed genes.
4. Clustering of genes
5. Knowledge Discovery via Data interpretation: building co-expressed and co-annotated gene groups.

4.2.2 First step: data generation

The five steps of microarray data experiments: manufacture, sample preparation and labeling, hybridization, image scanning, and image processing were completely done by the data provider, IPMC (reader can see the Fig. 1.17 for more details).

The output of this first step is the raw intensity measures of 22739 genes taken at 5 time points. The raw intensity measures are given as the logarithmic ratio, i.e. $log_2 \frac{Cy5}{Cy3}$, of two light intensities: the red one, $Cy5$, corresponding to the studied or test biological sample and the green one, $Cy3$, corresponding to the normal or reference sample.

4.2.3 Second step: normalization and replicates treatment

One of the main drawbacks of spotted oligos-chip is the inherent noise contained in the raw data (after the data generation step). In order to get rid of these inherent noise, the data provider IPMC has realized three different intensity-dependent normalizations:

[25] Technical replicates are the copies of the same sequence probes arranged in different spots of the DNA chip.

[26] Biological replicates are the repetitions of the whole biological experiment or process. Often, they correspond to two or more chips containing the same elements: probes and targets.

- Normalize within arrays by *lowess* normalization applied onto the surface (x, y) of the chip. This compensates the variation related to errors of handling the chips such as stains.
- Normalize within arrays by *lowess* normalization between the two color intensities: $Cy5$, $Cy3$, in order to eliminate the variations due to the use of double coloration.
- Normalize between arrays by *quantile* distribution normalization to obtain the same distribution of the signal for all the experimental conditions.

The goal of the first point is to normalize the average value of the $log_2(ratio)$ intensities to 0 on each surface (x, y) of the chip. Lowess is a locally weighted linear regression method that takes the *log* average of all gene-intensities on a surface (x, y) of the DNA chip

The goal of the second point is to normalize the average value of the $log_2(ratio)$ intensities to 0 on each chip for avoiding the variations of double coloration (more details of lowess method in section 2.1.4 or [76]).

The effects of the lowess normalization were illustrated in section 2.1.4 and Fig. 2.3. In this plot (called ratio-intensity plot or R-I plot). The horizontal axis represents the sum of the log intensities $log_{10}(Cy3 * Cy5)$ which is directly proportional to the overall intensity of a given spot. The vertical axis represents $log_2(Cy3/Cy5)$ which is the usual log-ratio of the two samples. Note the strong non-linear distortion in Fig. 2.3a and how this is corrected by Lowess in Fig. 2.3b.

The goal of the third point is to normalize all transformed $log_2(ratio)$ intensities in order to obtain distributions across the chips as similar as possible. The quantile normalization method was proposed by Yang et al. [328] for the case of two-colored spotted chips. The assumption behind this method is that given a series of chips, a small number of genes may be differentially expressed, however, the overall distribution of spot intensities should not vary too much (More details in quantile normalization method can be found in [328]).

The data provided by IPMC laboratory have already been treated by intensity-dependent normalization techniques. Now, we are interested in preparing the data for the third analysis step (selection of differentially expressed genes). Thus, we have applied two more statistical treatments: treatment of technical replicates within a chip and global normalization for each temporal biological condition of each chip.

In order to handle the technical replicates in the Azerty data set, we have followed a simple rule: the averaging of the transformed $log_2(ratio)$ intensities of the replicates. For example, in the case of gene $FOXN4$ that appears two times in the same chip with measures: 8.44 and 8.76 respectively, we have taken the average of the intensities, that is 8.60.

The majority of the differentially expressed gene analysis methods (explained in section 2.2) needs standardized data as input. In order to achieve this requirement, we have applied a global normalization method: standardization for each temporal biological condition of each chip. The method is simple: from each temporal biological condition measurement on the chip subtract the mean measurement and divide by the standard deviation of the temporal biological condition. After this transformation, the mean of the measurements of

each temporal biological condition on each one of the six chips will be zero, and the standard deviation will be one.

It is important to note that the normalization within arrays by lowess method are supposing independence of the temporal biological conditions columns. That is certainly not true because it is a temporal process. But, as stated before, most of the microarray tools suppose independence of biological conditions even if they are clearly dependent between each other. This assumption contained in the provided data will be a source of bias along our data analysis procedure.

4.2.4 Third Step: Selecting differentially expressed genes

The goal of this step is to detect genes that are over-expressed or under-expressed along the 6 biological replicates. We want to reduce the large size of Azerty data set. This task is called gene selection, where we take out the unexpressed and constant genes in order to find essential information about the studied biological process. Several selection methods were fully explained in section 2.2.

Given the temporal characteristics of the actual Azerty data set containing 6 biological replicates which contain 22739 genes and 5 temporal conditions each, a good choice of gene selection method is the statistical method Significance Analysis of Microarrays (SAM).

SAM is based on a modified $t - statistic$ for every gene i. This statistic measures the strength of the relationship between gene expression and the response variable (reference vs. studied) using repeated permutations of the data. The significance cutoff, δ, is determined by a tuning parameter, chosen by the user and based on the *false discovery rate* (FDR). The basic idea of this approach is to control the proportion of significant results that are in fact Type I errors (in hypothesis testing language). A brief description of this method can be found in section 2.2.5 and a full explanation can be found in Tusher et al. [309].

We have run the release 2.0 of SAM open source program with the following three main parameters:

1. Response type: time series data with signed area for summarizing each time course.
2. Cutoff threshold of $\delta = 0.21$, that implies the standard *false discovery rate* of 5 %.
3. Imputation engine: 20-nearest neighbor's.

The first parameter concerns the type of data i.e. time series data, we have chosen the option "summarizing each time course by signed area". This means that the surface under the time course curve is computed, counting positive area above the line (red points in Fig. 4.4) and negative below the line. This option is useful for finding genes that rise and then level off or come back down to their baseline [309].

The second parameter is the cutoff of FDR equal to 5 %, which means that we accept that 5 % of the chosen expressed genes are statistically unexpressed. The cutoff choice of FDR=5 % is one of the most common cutoff choices in microarray gene selection [309].

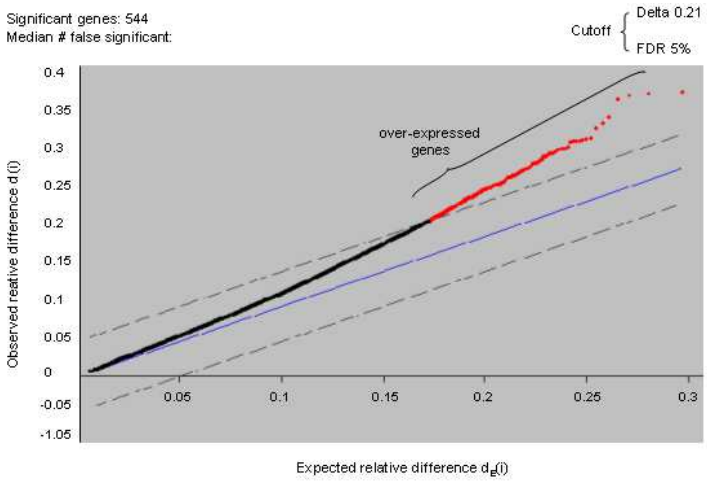

FIG. 4.4: The scatter plot of $d(i)$ vs. $d_E(i)$ to select potential significant genes (Azerty experience).

The third parameter is the choice of the missing values method, in this case 20-nearest neighbors imputation engine (explained in section 2.1.2).

After carrying out the SAM analysis we obtain 544 significant and differentially expressed genes. The SAM output results are shown in FIG. 4.4

The FIG. 4.4 plots observed values (d_i) versus its expected values (d_{E_i}). The solid blue line in the FIG. 4.4 indicates the line where $d_i = d_{E_i}$, that is where the observed relative difference is identical to the expected relative difference. The dotted lines are drawn at a distance $\delta = 0.21$ from the solid line. Given a specific δ, the procedure declares significance as follows: find the smallest upper cut point d_u such that $d_u - d_{Eu} \geq \delta$, and report all the genes i where $d_i \geq d_u$ as "significant positives" (all are marked in red color in FIG. 4.4). Similarly, find the largest d_j, called the lower cut point, where $d_k - d_{Ej} \geq \delta$, and report all the genes k such that $d_k \geq d_j$ as "significant negatives" (there are no significant negative genes in our Azerty data set, as seen in FIG. 4.4).

The output at this stage is a data set with 544 over-expressed genes and 5 temporal biological conditions in each of the 6 samples.

Strangely there are relatively few differentially expressed genes and they are all over-expressed (red line in FIG. 4.4). A possible reason of this may be the quantile distribution normalization applied to the original data because this process causes two phenomenons: i) Bringing together the gene expression measures between all biological conditions and also because a lower distribution distance between the chips. ii) If the original data set has many unexpressed or constant genes the normalization changes the original value in a way that could increase the rate of false positives.

Thus, among the 544 genes we could have more than 5% rate of false positives and surely we did not select an important number of under-expressed genes.

In order to prepare the Azerty data set for clustering techniques (fourth analysis step), we have taken the average gene expression measure over the six biological replicates or chips,

$Ave(x^c_{i,j})$, that is:

$$Ave(x^c_{i,j}) = \frac{\sum_{c=1}^{6} x^c_{i,j}}{6}, \text{ with } i = 1, 2, ..., 544.\text{and } j = 1, 2, ..., 5. \quad (4.3)$$

Here, $x^c_{i,j}$ is the gene expression measure of gene i under the temporal biological condition j in chip c. The final output of this third step is a matrix X containing 544 genes (matrix lines) measured under 5 time points: $1hr.$, $3hr.$, $6hr.$, $9hr.$ and $24hr$ (matrix columns). Each of the matrix elements $x_{i,j}$ represents the gene expression average of gene i under the time point j obtained among six biological replicates or chips. For interpretation purposes it is useful to take into account the meaning of each gene expression measure as a logarithmic ratio which compares two samples: the studied or test biological sample versus the normal or reference one.

4.2.5 Fourth step: clustering of co-expressed gene groups

The Clustering of co-expressed gene groups consists in identifying "clusters" or groups of genes which present a common expression profile among all biological conditions. In the Azerty data set it means finding clusters of genes that have similar expression profiles along the five time points of the studied biological process. (More details on this step can be seen in section 2.3).

In order to achieve this fourth step, we have chosen the two most currently used clustering approaches in microarray technology: partition-based and hierarchical approaches (as seen in [90, 69, 107, 298]). Among these approaches we have selected four unsupervised clustering techniques: Unweighted Pair Groups Method Average (UPGMA), DIvisive ANAlysis (DIANA), K-means, and Partition around Medoids[27] (PAM). These approaches were suggested by Datta et al. [85] in the case of time series data. The reader can find a full explanation of these two clustering partition-based approaches and their distance measures in section 2.3.2, 2.3.3 and 2.3.1 respectively.

K-Means and PAM are partition-based approaches, thus one needs to fix the number of clusters in advance. The K-means algorithm then assigns the observation into various clusters in order to minimize the total within-class sum of squares. The PAM algorithm first finds an initial set of medoids and then exchange points so that no single switch of an observation with a medoid will decrease the sum of squares (more details of these two methods in section 2.3.2). In the case of K-means, we have used the Pearson's correlation coefficient $Pearson(X_i, X_k)$ as distance correlation measure between the expression profiles of gene vectors X_i, X_k. In the case of PAM, we have used the Euclidean distance suggested by Kaufman et al.[164] for analyzing time series data.

UPGMA and DIANA are hierarchical approaches which produce a hierarchy of clusters rather than a set of clusters fixed in advance (as partition-based approaches).

UPGMA uses an agglomerative or bottom-up approach for building clusters. This method uses unweighted pair-group average linkage as merging rule between two clusters, which means taking the average distance between all pairs of objects in the two different

[27] A medoid can be defined as that object of a cluster, whose average dissimilarity to all the objects in the cluster is minimal i.e. it is a most centrally located point in the given data set.

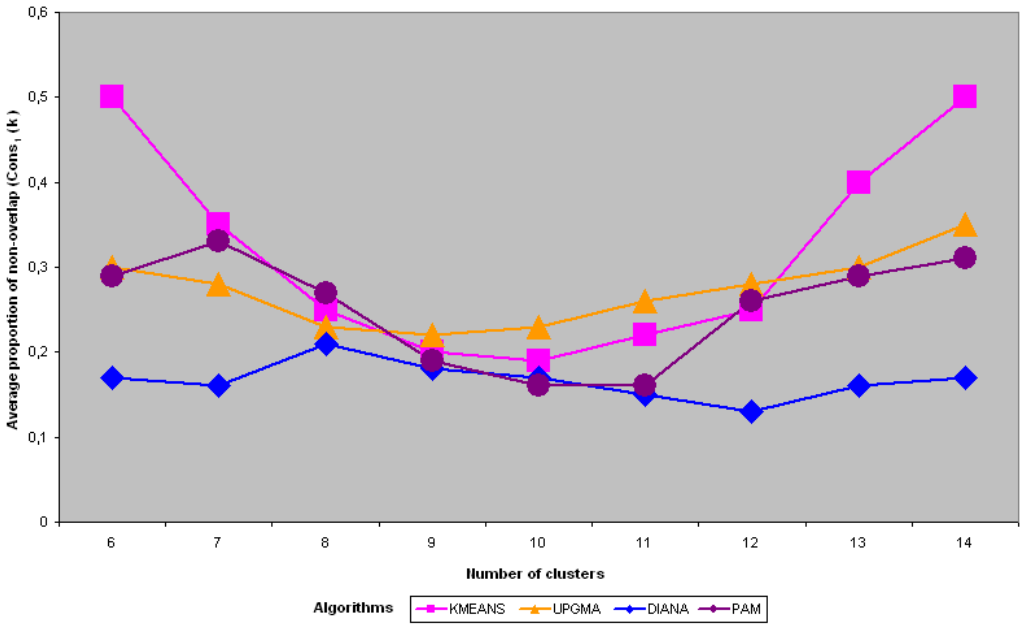

FIG. 4.5: Average Proportion of non-overlap measure, $Cons_1(k)$, for various clustering: algorithms applied to the Azerty data.

clusters. UPGMA with Pearson measure is suggested by Tamayo et al.[298] for analyzing time series data. Furthermore, UPGMA is actually the most used method ever since Eisen et al. have proposed its usage for microarray technology in 1998 [107] (more details in section 2.3.3).

DIANA uses a divisive algorithm or top-down approach for building clusters. This method using the euclidean distance $d(X_i, X_k)$ as similarity measure is suggested as a robust method for time series data by Datta et al.

Once we have defined the clustering algorithms and their respective distance and linkage measures, we need to determine the parameter k, that is, the number of clusters, and the consistency of the chosen algorithms. In order to answer to this two questions, we have used three different consistency measures: average proportion of non-overlap ($Cons_1(k)$), average distance between clusters ($Cons_2(k)$), and average distance between cluster means ($Cons_3(k)$). These consistency measures were proposed in Datta et al.[85] (more details about these measures in section 2.3.6).

The results of applying these three consistency measures over UPGMA, DIANA, PAM and K-means algorithms in Azerty data for $k \in \{6, 7, ..., 14\}$ are presented in FIGS. 4.5-4.7.

We can observe in FIG. 4.5 that concerning measure $Cons_1(k)$ UPGMA and k-means perform poorly for almost all number of possible clusters and DIANA is the most regular algorithm along all possible k values. There is a simple reason because correlation pearson measure are not appropriate here. This measure is invariant under location and scale transformations, and thus, it cannot distinguish between the patterns that are related by location and/or scale changes. In contrast, Euclidean measure works well for DIANA algorithm and

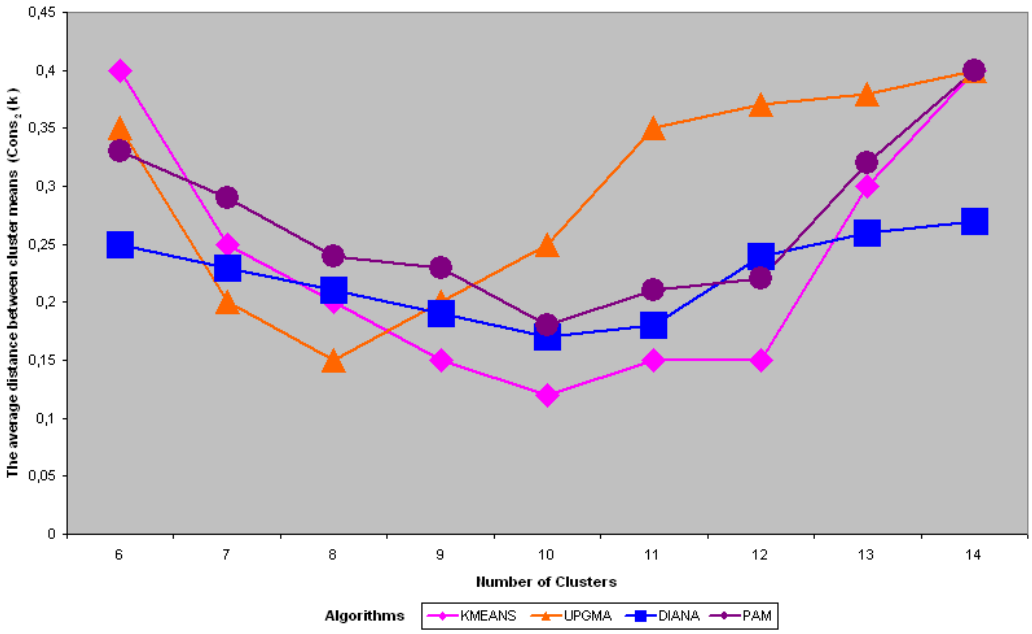

FIG. 4.6: Average distance between means measure, $Cons_2(k)$, for various clustering algorithms applied to the Azerty data.

for PAM, and shows good behavior for $k \in \{9, 10, 11\}$. Interestingly, the local minima of the four algorithms is found for values of $k \in \{9, 10, 11, 12\}$.

As illustrated in FIG. 4.6, concerning $Cons_2(k)$ the winning algorithm was k-means, in spite of the high levels of variation along different cluster values. UPGMA performs well only in the case of 8 and 9 clusters. The algorithms that use Euclidean measures do not perform well among cluster values of less than 8 or more than 11 clusters. In the case of $Cons_2(k)$ measure, Euclidean measure is less appropriate than Pearson measure. Logically, variations in the contents of the built clusters are more sensible in Euclidean distance compared to Pearson correlation distance. As well as for $Cons_1(k)$, the local minima of the 4 algorithms are found for values of $k \in \{9, 10, 11\}$.

As shown in FIG. 4.7, Diana and PAM algorithms perform quite well for $k \in \{8, 9, 10\}$. In contrast, the results for k-means and UPGMA are less optimist for all values of k. The reason could be the high variability of Azerty data set, because it represents a biological time process. A biological process could change suddenly between time points, and Pearson correlation distance is more sensible than Euclidean distance in relation to high changes between the biological time points. Interestingly the local minima of the four algorithms are situated for values of $k \in \{8, 9\}$.

Taking into account the results of the three consistency measures $Cons_1(k)$, $Cons_2(k)$, $Cons_3(k)$, we have decided to perform our clustering analysis fixing the number of cluster to $k = 10$, which represents the "best" compromise regarding the local minima shown in the three consistency measures.

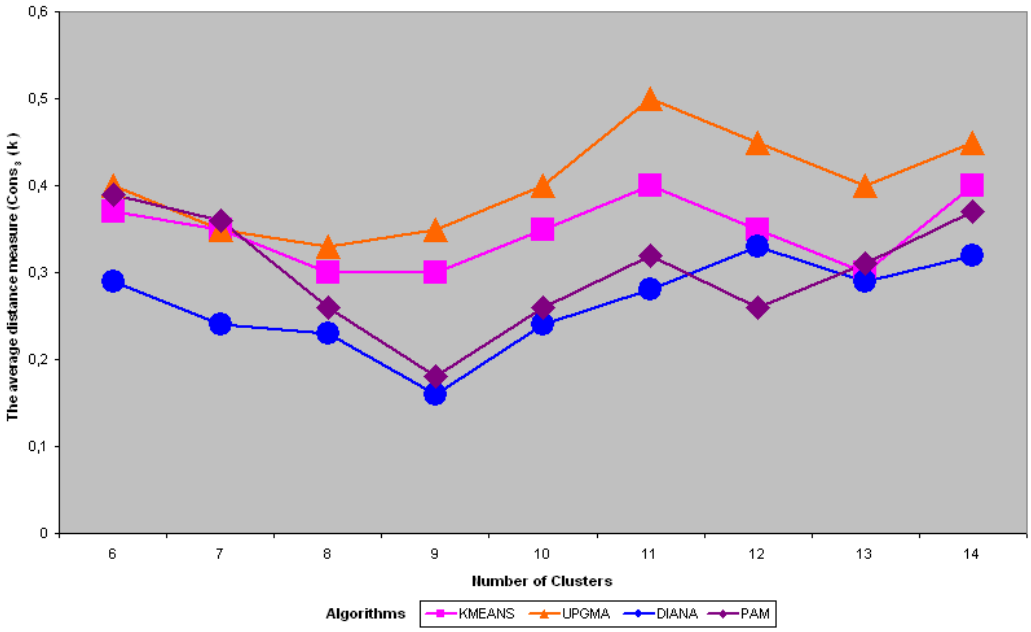

FIG. 4.7: Average distance measure, $Cons_3(k)$, for various clustering algorithms applied to the Azerty data

Concerning the consistency of the algorithm, there is no a winner between the four clustering methods presented here. So, we proceed the Azerty clustering analysis with the four clustering algorithms: UPGMA, DIANA, PAM and k-means, taking the respective parameters of distance and linkage rule and fixing the number of clusters to $k = 10$.

Applying the four clustering methods, we have obtained 10 gene groups for each algorithm. The resulting 40 clusters contain some similarities and dissimilarities between the composing elements among different methods. In other words, a group of genes could or could not appear together in two or more clusters obtained from different algorithms. Here, we are interested in finding the co-expressed genes that appear in the intersection of at least two clusters among the clusters of each of the four algorithms. We called this co-expressed gene groups as *hard clusters*.

In order to find the *hard clusters* of the Azerty data set, we have determined the intersection of the elements contained in at least two clusters among the 40 resulting clusters. There are 4 clustering methods, so a gene can appears 4 times in clusters obtained by different methods. Thus, the number of possible hard clusters is the addition of the possible combinations*: $\binom{40}{2} + \binom{40}{3} + \binom{40}{3} = 102830$. In order to reduce the number of hard clusters we have taken a pruning factor of 85%. The pruning factor is defined as:

$$Max\left(\frac{|\cap C_k^m|}{\left|C_k^{Upgma}\right|}, \frac{|\cap C_k^m|}{\left|C_k^{Pam}\right|}, \frac{|\cap C_k^m|}{\left|C_k^{K-means}\right|}, \frac{|\cap C_k^m|}{\left|C_k^{Diana}\right|}\right) * 100\% \succeq 75\%, \qquad (4.4)$$

where $|\cap C_k^m|$ is the cardinality of elements found in the intersection of clusters from method m with cluster number k, and $\left|C_k^{Upgma}\right|$ is the cardinality of the elements in cluster number k from method UPGMA. For example let us suppose three clusters obtained by three different methods: the second cluster of UPGMA clustering, C_2^{Upgma}, the fourth cluster of PAM clustering, C_4^{PAM}, and the fifth cluster of DIANA clustering $C_5^{Diana}$with cardinalities 30, 72 and 27 respectively. The cardinality of the intersection of the elements of the three algorithms is 17. Applying eq. 4.4 we found $Max\left(\frac{17}{30}, \frac{17}{72}, \frac{17}{27}\right) * 100\% = 63\% \leq 75\%$. So, the hard cluster $\left|C_2^{Upgma} \cap C_4^{PAM} \cap C_5^{Diana}\right|$ will be eliminated from our analysis.

Applying the pruning factor of equation 4.4 we have obtained as final output of the fourth analysis clustering step: 27 hard clusters containing co-expressed genes with similar expression profiles along the five time point biological process.

Implementation

All the consistency measures $(Cons_1(k))$, $(Cons_2(k))$, and $(Cons_3(k))$ were implemented in R language. The four clustering algorithm programs were obtained from the cluster library in the BIOCONDUCTOR open source project. The program for finding hard clusters and the one for obtaining the selected hard clusters were implemented in R language.

4.2.6 Fifth step: knowledge discovery via data interpretation.

This step concerns building co-annotated gene groups from the selected hard clusters and testing the significance of the resulting co-expressed and co-annotated gene groups. For building the co-annotated groups, we have used the web tool FATIGO [5], which finds significant associations of Gene Ontology terms with groups of genes.

Unfortunately for interpretation analysis, Azerty data set was partially built with human genes without any known annotation, that is only the sequence of nucleotides of these genes is known. In order to reduce the number of hard clusters containing genes without any annotation, we have run the pruning rule of: "taking only the hard clusters containing at least 50% annotated genes (an annotated gene must have at least one biological process annotation in Gene Ontology.) We have run FATIGO tool for each of the 27 hard clusters and have found only 4 hard clusters (see FIG. 4.3).

We can see in FIG. 4.8 the expression profile measures of the four selected hard clusters containing at least 50% of annotated genes with GO biological process annotations. The scale in top of FIG. 4.8 indicates that the $log_2(ratio)$ is between 3.0 and -3.0, indicating overexpression for measures between $[1, 3]$ corresponding to orange to red colors, equal expression for measures between $(-1, 1)$ corresponding to yellow colors and under-expression for measures between $[-1, -3]$ corresponding to yellow to green colors. FIG. 4.8 illustrates the similar expression profiles shown by our four chosen hard clusters: $HC_1 = \{C_5^{Upgma} \cap C_3^{SOM} \cap C_2^{Diana}\}$, $HC_2 = \{C_7^{Diana} \cap C_{10}^{PAM} \cap C_8^{Upgma}\}$, $HC_3 = \{C_7^{K-means} \cap C_4^{PAM} \cap C_8^{Diana}\}$ and $HC_1 = \{C_4^{Diana} \cap C_6^{PAM}\}$. For example in cluster HC_1 we can observe the marked overexpression (strong red) of all the 16 genes along almost all biological process. At time point nine hours

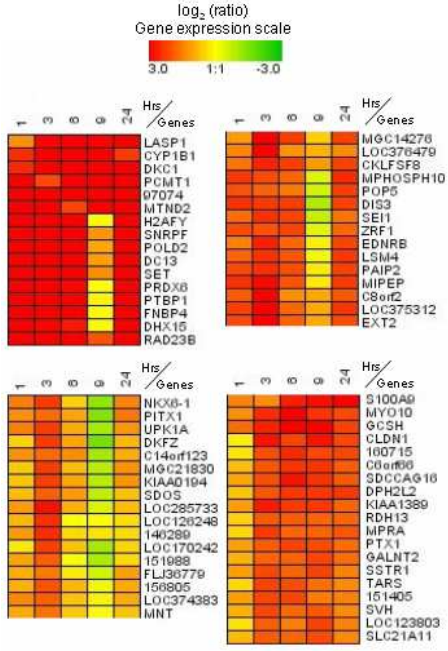

FIG. 4.8: Four selected hard clusters with at least 50% of the genes with biological process GO annotations

we can see that 10 genes have been equally expressed (yellow color) while the remaining 7 genes (top of the HC_1 cluster) continue to be overexpressed.

We have run FATIGO tool again for searching the co-annotated (by biological process GO annotations) significant groups within the four selected hard clusters HC_1, HC_2, HC_3 and HC_4. We have chosen a significant level α of $\alpha = 9.9E-02$ and a gene ontology* hierachical level of 4. Thus, the chosen gene groups have a $p-value$ smaller or equal than α. For the sake of brevity we present the results for HC_1 in TABLE 4.5:

The first column of TABLE 4.5 represents the gene ontology level. The second column shows the biological process GO annotation. The third contains the number of human genes in each subset. The fourth column contains the percentage of annotated genes in relation to the cardinality of the total number of annotated genes in cluster HC_1(in this case 12 of the 16 genes), and the last column shows the $p-value$ of the significant subset.

From TABLE 4.5 we extract that highly correlated genes (SET and H2AFY) are related not only by a similar expression profile but also that they share 6 functional annotations. Another highly correlated pair of genes is PTBP1 and SNRPF participating actively in mRNA processing and RNA splicing. The genes DKC1, SET and H2AFY participate together in two metabolic process: nucleobase and polymer, and also in chromosome organization and biogenesis (see TABLE 4.5).

Up to now, we have discovered groups of co-annotated and co-expressed genes, but at the moment we are incapable of having some clue of the unknown studied biological process. Because of the insufficient annotations obtained by GO biological process annotations concerning our four hard clusters, we did not gain any insight into the studied biological process.

GO Level	Annotation	#	Genes	%	p-value
4	biopolymer metabolic process	8	PTBP1 SNRPF RAD23B PCMT1 SET POLD2 H2AFY DKC1	67	6.24E-02
4	nucleobase, nucleoside metabolic process	7	PTBP1 SNRPF RAD23B SET POLD2 H2AFY DKC1	58	4.47E-02
4	organelle organization and biogenesis	4	LASP1 SET H2AFY DKC1	33	2.57E-02
5	DNA metabolic process	4	RAD23B SET POLD2 H2AFY	33	8.50E-02
6	Dna replication	2	SET POLD2	17	4.82E-02
6	chromosome organization and biogenesis	3	SET H2AFY DKC1	25	3.13E-02
7	mRNA processing	2	PTBP1 SNRPF	22	5.56E-02
7	RNA splicing	2	PTBP1 SNRPF	22	4.10E-02
9	chromatin assembly	2	SET H2AFY	40	3.49E-02

TABLE 4.5: Significant co-annotated and co-expressed genes groups of cluster HC_1 with their respective GO annotation and significance level.

In order to find some clues about the studied biological process we use the bibliographic source of information: PUBMED/MEDLINE. Taking the genes contained in the hard cluster, HC_1, we have searched manually 25 gene-related articles. We have obtained useful functional and relational annotations among the 16 genes contained in cluster HC_1. The results of this extracting process are presented in TABLE 4.6.

Medline-Human Annotation	Genes
Tissue Reparation	PCMT1 POLD2 SET PTBP1 RAD23B
Cancer	CYP1B1 LASP1 SET RAD23B
Defense	DKC1 SET
Regulation	DKC1 FNBP4
Damage Reparation	DKC1 PRDX6 RAD23B SET
Regulation	FNBP4 DKC1
Apoptosis	RAD23B LASP1
DNA reparation	PRDX6 DHX15 POLD2

TABLE 4.6: Co-annotated and co-expressed groups of cluster HC_1 with their respective Medline-Human annotation.

TABLE 4.6 shows the manually extracted co-annotated gene groups. For example the group of co-expressed genes PCMT1 POLD2 SET PTBP1 RAD23B plays a role in tissue reparation, and the genes CYP1B1 LASP1 SET RAD23B are active in the cancer tissue state relative to a normal tissue state. Similarly, we can read all the information in TABLE 4.6.

Resuming the information of the fourth and fifth analysis step about hard cluster HC_1 (FIG. 4.8, TABLE 4.6 and 4.5) we summarize:

- The 16 genes in HC_1 are over-expressed in all biological experiments and they contain important functional relationships among the genes within the cluster (see FIG. 4.8).

- The functional relationships obtained using GO (biological process ontology) are principally of three types: metabolic process (nucleobase, polymer and DNA), chromosome/organelle/chromatin organization and biogenesis, and RNA splicing and processing (see TABLE 4.5).
- The functional relationships obtained by extracting annotations from 25 gene-related Medline articles are: tissue reparation, cancer, defense, regulation, damage reparation, regulation, apoptosis* and DNA reparation (see TABLE 4.6).

We have found four hard and significant co-expressed clusters: HC_1, HC_2, HC_3 and HC_4. Among these clusters, we have interpreted the HC_1 cluster, obtaining the results resumed in the three last points. The Gene Ontology annotations can be accomplished by many biological processes, so the clues that we have extracted from our five-step analysis are the Medline annotations: tissue reparation, cancer, defense, regulation, damage reparation, regulation, apoptosis and DNA reparation. Indeed, we have several biological process applications containing many of these 8 annotations.

The answer to our main goal puzzle was: Azerty data set was a cicatrization process. Here, we present the complete characteristics of the cicatrization Azerty data set.

Cicatrization data set description: Cicatrization data set is composed by 6 chips with 27648 spots each. Every chip contains the expression levels of 22739 genes measured in the cicatrization of Human Bronchial Epithelial cells (HBE) at five time points $1hr.$, $3hr.$, $6hr.$, $9hr.$ and $24hr.$ Every gene expression measure represents the logarithmic ratio, i.e. $log_2 \frac{Cy5}{Cy3}$, of two light intensities the red one, $Cy5$, corresponding to the wounded HBE sample and the green one, $Cy3$, corresponding to the normal HBE sample.

Implementation

We have used the open source web tool FATIGO [5] for building the co-annotated GO gene groups. The graphics for hierarchical clustering outputs were obtained with Genesis program (more details in [296]). For obtaining the literature annotations we have used PUBMED/MEDLINE bibliographic database source .

4.2.7 Discussion

Here, we discuss the difficulties encountered along the complete analysis of spotted oligos-chip Azerty data.

The data generation step and the intensity-dependent normalization procedures were done by the provider IPMC laboratory. As explained before, the distribution normalization applied to the original data has supposed the independence of biological time conditions. This is clearly not true, because in biological process the future state depends directly on the anterior state, so they are not independent. This is a common bias in analyzing biological time series data, because there do not exist enough tools in microarray to deal with this kind of data, so analysts suppose from the beginning of the analysis the independence between the biological conditions. The consequences of this bias have not been yet studied.

The statistical data treatment is a crucial step which has been partially done by the provider as discussed before in the data generation step. The remaining statistical treatments as technical replicates treatment and global normalization for each chip were realized using standard techniques. However, this choice could be erroneous and could add more bias to the data.

For selection of differentially expressed genes we have used SAM algorithm, an statistical parametric method including a multiple testing correction feature. In spite of the robustness of this algorithm, the choice of the differentially expressed genes is finally done by an arbitrary cut-off of $FDR = 5\,\%$. Thus, we eliminated genes that are near the choice boundary (see FIG. 4.4) but they might be important for the studied biological process. Methods for differentially expressed gene selection including gene annotations in their algorithm would be advisable

Concerning the clustering of the genes, we must pay attention to the choice of clustering algorithm and distance measure. This decision has to be done by testing with different clustering algorithms and distances measures and then validating the results with cluster validation methodologies (as seen in section 2.3.6). Thus, we could find the "best adapted" algorithm and measure to our particular data set. In Azerty data we did not find a winner between the chosen clustering algorithms and distance measures using three consistency measures. Thereafter we have constructed significant "hard clusters" containing genes with similar expression profiles. Even if the clusters are reflecting co-expressed gene groups, a gene can appear only once in one group and intersections are not allowed. However, biologically genes can participate to many process at the same time, thus these clustering is unrealistic. Bi-clustering techniques explained in section 5.4 could be more adapted to this problem. Another important default is the lack of biological information integration in the clustering algorithm, so we can have co-expressed genes that participate to different biological process. Co-clustering techniques explained in section 5.4 could be more adapted to realize this task.

Finally, the interpretation step depend on two main parameters: the availability of the gene annotations and the manipulation of the different sources of biological information. In case of the Azerty data set, the existing Gene Ontology annotations for the obtained co-expressed hard clusters were insufficient or too general to obtain important information about the studied biological process. In contrast, the bibliographic literature database PubMed/Medline contains up to date and important information concerning the studied genes for targeting the main goal. However, human extraction of important annotations in millions of online gene-related articles is known to be a time consuming and difficult task. Up to now, the automatic *text mining* science algorithms in genomic areas are not effective enough because of the inherent characteristics of textual information (see section 3.5 for more information).

4.2.8 Conclusion

The inherent noise in the Azerty data set and the cumulated bias from the first four analysis steps have made the interpretation of the significant co-expressed gene groups difficult. Furthermore, the lack of annotations and automatic systems to integrate the information

contained in the gene expression profiles and gene annotations have difficult even more the achievement of our main goal.

While analyzing data issued from oligos-chip technology, we are dealing with many difficulties (see section introduction). These difficulties have to be solved carefully using the most adapted tools to tackle each of the five-analysis steps. In the case of time series data ther do not even exist tools so a guided analysis could be necessary. Selecting differentially expressed genes and clustering genes are two steps that need to be guided using at least one kind of the six biological sources (explained in chapter 3) to arrive to more realistic results. If the biological source integration is done only at the last fifth step, the conclusion might be false or unclear. Knowledge discovery and interpretation need well structured and easy handling sources of information and that is not often given in biological resources (explained in chapter 3).

Chapter 5

Biological Knowledge Interpretation approaches: Prior or Knowledge-based, Standard or Expression-based and Co-Clustering

This chapter represents the framework of our biological knowledge integration models, who are briefly introduced here. In this chapter, we discuss different approaches for integrating biological knowledge in gene expression analysis. Indeed we are interested in the fifth step of microarray analysis procedure which focuses on knowledge discovery via interpretation of the microarray results (introduced in section 2.4). We present a state of the art of methods for processing this step and we propose an original classification in three facets: **prior or knowledge based, standard or expression-based and the co-clustering**.

First we discuss briefly the purpose and usefulness of our classification. Then, following sections give an insight into each facet principles and intinsic methods. We summarize each section with a comparison between reamarkable approaches. Finally, we discuss all three facets and methods giving an outlook for future research.

5.1 Introduction

Nowadays, one of the main challenges in gene expression technologies is to highlight the main co-expressed* and co-annotated* gene groups using at least one of the different sources of biological information [13]. In other words, the issue is interpretation of microarray results via integration of gene expression profiles with corresponding biological gene annotations extracted from biological databases (presented in chapter III).

Analyzing microarray data consists in five steps: protocol and image analysis, statistical data treatment, gene selection, gene classification and knowledge discovery via data interpretation [334] (presented in chapter II). We can see in Fig. 5.1 the goal of the fifth analysis step devoted to interpretation, which is the integration between two domains, the numeric one represented by the gene expression profiles and the knowledge one represented by gene annotations issued from different sources of biological information.

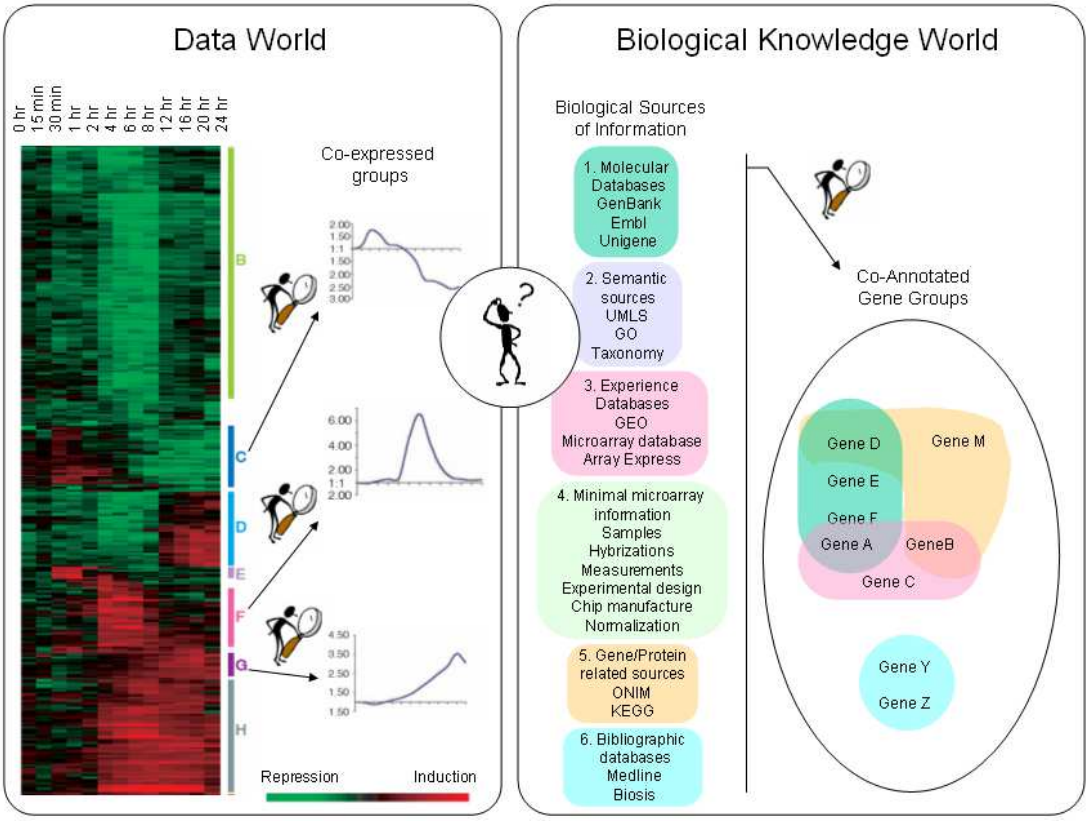

FIG. 5.1: Interpretation of microarray results via integration of gene expression profiles with corresponding sources of biological information

At the beginning of gene expression technologies, researches were focused on the numeric[28] side. So, there have been reported ([69, 90, 107, 298, 303, 26]) a variety of data analysis approaches which identify groups of co-expressed genes based only on expression profiles without taking into account biological knowledge (some of them briefly explained in section 2.3). A common characteristic of purely numerical approaches is that they determine gene groups (or clusters) of potential interest. However, they leave to the expert the task of discovering and interpreting biological similarities hidden within these groups. These methods are useful, because they guide the analysis of the co-expressed gene groups. Nevertheless, their results are often incomplete, because they do not include biological considerations based on prior biologists knowledge.

In order to process the interpretation step in an automatic or semi-automatic way, the bioinformatics community is faced to an ever-increasingly volume of sources of biological information on gene annotations. Besides minimal gene expression technology (microarray or SAGE) experiment information, we have identified five sources of biological information:

- Molecular databases (GenBank, Embl, Unigene, etc.).
- Semantic sources as thesaurus, ontologies, taxonomies or semantic networks (UMLS, GO, taxonomy, etc.).
- Gene expression databases (GEO, Arrayexpress, Microarray database, etc.).
- Bibliographic databases (Medline, Biosis, etc.).
- Gene/protein related specific sources (ONIM, KEGG, etc.)

The reader can go to chapter 3 for a full explanation in these sources of biological information. Exploiting these different sources of biological information is quite a complex task so scientists developed several tools for manipulating them or integrate them into more complex databases [34, 217].

This chapter presents a complete survey of the different approaches for automatic integration of biological knowledge with gene expression data. A first discussion of these methods is presented by Chuaqui in [74]. Here we present an original classification of the different microarray analysis interpretation approaches.

The interpretation step may be defined as the result of the integration between gene expression profiles analysis with corresponding gene annotations. This integration process consists in grouping together co-expressed and co-annotated genes. Based on this definition, three research axes may be distinguished: *the prior or knowledge-based axis*, *the standard or expression-based axis* and *the co-clustering axis*. Our classification emphasizes the weight of the integration process scheduling on the final results [175], [112], [186], [139].

Indeed the main criteria underlying the classification we propose is the scheduling of phases which alternatively consider gene measures or gene annotations. In prior or knowledge-based approaches, first the co-annotated gene groups are built and then the gene expression profiles are integrated. In standard or expression-based approaches, first co-expressed gene

[28] We understand by numeric part the analysis of the gene expression measures only, disregarding the biological annotations

groups are built and then gene annotations are integrated. Finally, co-clustering approaches integrate co-expressed and co-annotated gene groups at the same time

This chapter is organized in the following way: each section fully explains the corresponding interpretation axis, giving an insight into their remarkable approaches and summarizing with a comparison between them. Then, we develop a discussion among the three interpretation axis. Finally, it develops a discussion analyzing each of the three interpretation axis and an outlook for future research.

5.2 Prior or Knowledge-Based Axis

Prior or knowledge-based approaches are based on biological knowledge from the sources of biological information (illustrated in Fig. 5.1). Therefore, first they build co-annotated gene groups sharing the same biological annotations. Then, they integrate the expression profiles information for each of the genes classified into co-annotated groups, highlighting those ones which are co-expressed. Later on, the statistical significance of co-annotated and co-expressed gene groups is tested. We give a detail description of this three-step methodology: co-annotated gene groups composition, gene expression profiles integration and significant co-annotated and co-expressed gene groups selection.

5.2.1 Prior or knowledge-based methodology

1.-Co-Annotated gene groups composition

There exist several ways to build co-annotated gene groups. We present here one structured way of building them. First, we need to choose among different sources of biological information. Each kind of information is stored in a specific format (xml, sql, etc.) and has intrinsic characteristics. In each case, the analysis process needs to deal with each biological source format. Another issue is to choose a nomenclature for each gene identity that has to be coherent with the sources of information and thereafter with the expression data. Next, all the annotations of each gene are to be collected in one or more sources of information. Finally, we gather in a subset of genes that share the same annotation. Thus, we obtain all the co-annotated gene groups as shown in first step of Fig. 5.2.

2.-Gene expression profiles integration

There are different ways to integrate gene expression profiles with previously built co-annotated gene groups. Here we present one current way to do it. First, expression profiles measures are taken for each gene. Then, a variability measure, as $fold\ change$ or $t-statistic$ or $f-score$ [257] is used to build a sorted list of gene-ranks based on expression profiles. Finally, this measure is incorporated gene by gene into the co-annotated groups. Thus, we obtain co-annotated gene groups with the expression profiles information within, as shown in second step of Fig. 5.2.

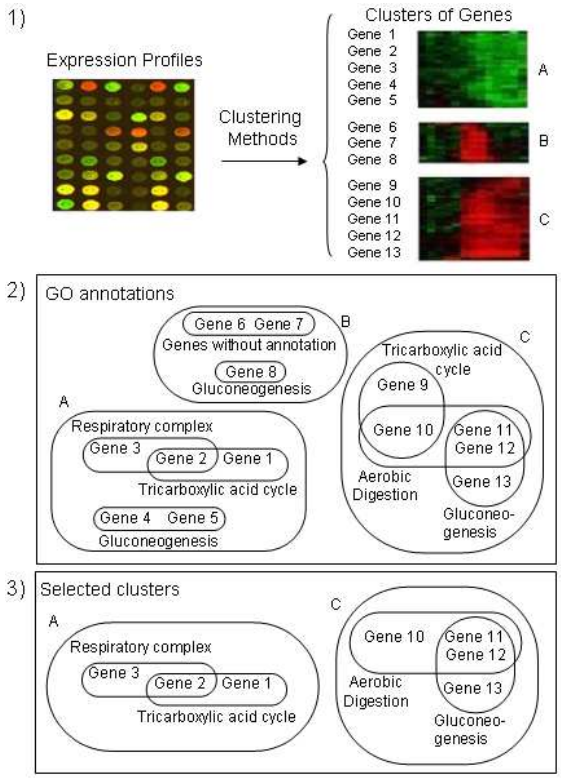

FIG. 5.2: Gene expression profiles integration into previously co-annotated groups

3.-Selection of the significant co-annotated and co-expressed gene groups

At this stage all co-annotated and co-expressed gene groups are built. The next step is to reveal which of these groups or subgroups are statistically significant. To tackle this issue the most frequent technique is the statistical hypothesis testing. Here, we present the four steps for statistical hypothesis testing:

1. Formulate the null hypothesis, H_0,
   H_0 : *Commonly, that the genes that are co-annotated and co-expressed were expressed together as the result of pure chance.* versus the alternative hypothesis, H_1
   H_1 : *Commonly, that the co-expressed and co-annotated gene groups are found together because of a biological effect combined with a component of chance variation.*
2. Identify a test statistic: The test is based on a probability distribution that will be used to assess the truth of the null hypothesis.
3. Compute the $p - value$: The $p - value$ is the probability that a test statistic at least as significant as the one observed would be obtained assuming that the null hypothesis were true.
4. Compare the $p - value$: This consists in comparing the $p - value$ to an acceptable significance value α. If $p - value \leq \alpha$ we can consider that the co-annotated and co-expressed gene group is gathered by a biological effect and thus is statistically significant. Consequently, the null hypothesis is ruled out, and the alternative hypothesis is valid

At the end of the four-step methodology explained before, the prior approaches present the interpretation results as significant co-expressed and co-annotated groups of genes (seeFig. 5.2 third step). The next section will present ones of the most remarkable approaches and methods of the prior or knowledge-based axis.

5.2.2 Remarkable prior or knowledge-based approaches

We present here four representative approaches: GSEA [215], iGA [53], PAGE [167] and CGGA [203]. In the following we describe each of them and emphasize some parameters particularly: the source of biological information, the profiles expression measure, the expression variability measure, the hypothesis testing parameters and details (type of test, test statistic, distribution, corrections etc.).

1. Gene Set Enrichment Analysis, GSEA

This approach [215] proposes a statistical method designed to detect coordinated changes in expression profiles of pre-defined groups of co-annotated genes. This method is born from the need of interpreting metabolic pathways results, where a group of genes is supposed to move together along the pathway.

In the first step, it builds a priori defined gene sets using specific sources of information which are the NetAFFX and GenMapp metabolic pathways databases.

In the second step, it takes the Signal to Noise Ratio (SNR) to measure the expression profiles of each gene within the co-annotated group. Then it builds a sorted list of genes for each of the co-annotated groups.

Third, it uses a non-parametric statistic: enrichment score, ES, (based in a Kolmogorov-Smirnoff normalized statistic) for hypothesis testing. It takes as null hypothesis:

$$H_0 : \textit{The rank ordering of genes is random with regard of the sample.}$$

Then, it assesses the statistical significance of the maximal ES by running a set of permutations among the samples. Finally, it compares the $max\ ES$ with a threshold α, obtaining the significant co-expressed and co-annotated gene groups.

2. Parametric Analysis of Gene Set Enrichment, PAGE

TThis approach [167] detects co-expressed genes within a priori co-annotated groups of genes like GSEA, but it implements a parametric method.

In first step, it builds a priori defined gene sets from Gene Ontology (GO), NetAFFX and GenMapp metabolic databases.

In second step, it takes the $fold\ change$ to measure the expression profiles of each gene within the co-annotated group. Then, it builds a $z-score$ from the corresponding $fold\ change$ of the two comparative groups (normal versus non normal) as variability expression measure.

Third, it uses the $z - score$ as parametric test statistic. Then, it uses the central limit theorem [113] to argue that when the sampling size of a co-annotated group is large enough, it would have a normal distribution. Using the null hypothesis:

$$H_0 : \textit{The } z - \textit{score within the groups has a standard normal distribution.}$$

Thus, if the size of the co-annotated gene groups is not big enough to reach normality, then it would be significantly co-expressed.

3. Iterative Group Analysis, iGA

This approach [53] finds co-expressed gene groups within a priori functionally enriched groups, sharing the same functional annotation.

In a first step, it builds a priori functionally enriched groups of genes from Gene Ontology (GO) or other sources of biological information.

In a second step, it uses the $fold\ change$ gene expression measure to build a complete sorted list of genes. Then, it generates a reduced sorted list specific to the functionally enriched group.

In a third step, it calculates iteratively the probability of change for each functionally enriched group (based in the cumulative hypergeometric distribution). It states the null hypothesis:

$$H_0 : \textit{The top } x \textit{ genes are associated by chance within the functionally enriched group.}$$

Then, it assesses the statistical significance of each group comparing the probability of change

$p - value$ against a user-determined α value.

4. Co-expressed Gene Group Analysis, CGGA

This approach [203] automatically finds co-expressed and co-annotated gene groups.

In a first step, it builds a priori defined gene groups from one source of biological information for instance Gene Ontology (GO) and KEGG.

In a second step, it uses the $fold\ change$ as a gene expression measure. Then, it composes the $f - score$ from the corresponding gene's $fold\ change$. Using the $f - score$ on each gene it builds a sorted list of gene ranks. Then, it generates a reduced list of gene ranks specific to the co-annotated enriched group.

In a third step, it states the null hypothesis:

H_0 : *x genes from a co-annotated gene group are co-expressed by chance.*

A hypergeometric distribution and $p-value$ are calculated from the cumulative distribution is assumed. This $p-value$ is compared against α to reveal all the significant co-expressed and co-annotated gene groups, including all the possible subgroups.

5.2.3 Comparison between prior or knowledge-based approaches

Table 5.1 presents the brief summary of the four prior approaches described in last section. For each approach the four following parameters are presented: sources of biological information used, expression profile measure, variability expression measure and hypothesis testing details (test statistic, distribution and particular characteristics).

First of all, the four approaches are concerned by metabolic pathways within biological processes, but they use different sources of information: iGA, PAGE and CGGA uses Gene Ontology and GSEA uses manual metabolic annotations, GENMAPP and NetAffx. CGGA is the only one which uses KEGG database combined with Gene Ontology.

For expression profiles parameters, GSEA is the only one which choice is the SNR measure while the others opted for the $fold\ change$ measure. PAGE and CGGA use respectively $z - score$ and $f - score$ variability measures to detect the changes in gene expression profiles.

For hypothesis testing, GSEA is the only one which uses a non parametric method based on a maximal ES statistic and sampling to calculate the $p - value$. In the contrary, PAGE (normal distribution), CGGA (hypergeometric distribution) and iGA (hypergeometric distribution) chose a parametric approach. iGA chose a hypothesis proof based in the most over-expressed or under-expressed genes (in the rank list) of a co-annotated group, while CGGA searches all the possible co-expressed subgroups within a co-annotated group (the internal sub-group position in the group does not matter).

Approach	Biological Source of Information	Expression Profile Measure	Variability Expression Measure	Hypothesis Testing Details
GSEA (Mootha et al. 2003)	Manual Annotations, NetAffx and GENMAPP	SNR (Signal to Noise Ratio)	Mean Expression Difference	One-tailed test. Test statistic: Maximal ES. Non-parametric distribution.
iGA (Breitling et al. 2004)	GO	Fold Change	Fold Change	One-tailed test. Modified Fisher's exact statistic: The most over or Under expressed Genes in a group. Hypergeometric distribution.
PAGE (Kim et al. 2005)	GO	Fold Change	z-score	One-tailed test. z-score statistic. Normal distribution.
CGGA (Martinez et al. 2006)	GO: (MP, BP, CC) and KEGG	Fold Change	F-score	One-tailed test. Modified Fisher's exact statistic: All over or under expressed genes in a group. Hypergeometric distribution. Binomial distribution for N large. Bonferroni Correction.

Table 5.1: Knowledge-based integration approaches

5.3 Standard or Expression-Based Axis

This axis is called standard because it follows the more frequent procedure for microarray data analysis, which consists of five steps: image analysis, statistical data treatment, genes selection, genes classification and results interpretation via biological knowledge integration. This axis has been used since the beginning of microarray technology with encouraging interpretation results [90], [107] and [69]. Thereafter, it has been used as the reference methodology in microarray data analysis. Expression-based approaches start by building gene groups or clusters of genes sharing similar expression profiles. Then, they integrate the biological annotations of each gene contained inside the expression cluster, building co-expressed and co-annotated subsets of genes. Later on, the statistical significance of co-expressed and co-annotated gene groups is tested. In the following section, we explain in detail this three-step methodology: gene expression profiles classification, biological annotations integration and significant co-expressed and co-annotated gene groups selection.

5.3.1 Standard or expression-based methodology

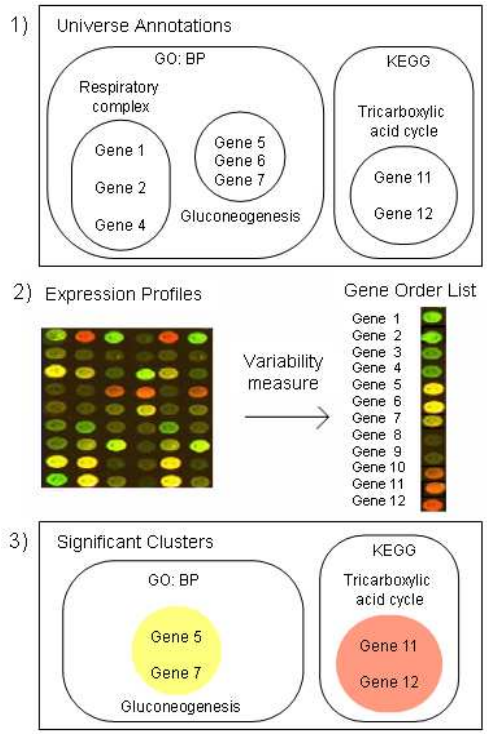

FIG. 5.3: Interpretation of microarray results via integration of gene expression profiles with corresponding sources of biological information

1.-Gene expression profiles classification

There exist several methods for classifying gene expression profiles from cleaned microarray data, i.e. data matrix of thousands of genes measured in tens of biological conditions. Various supervised methods and non supervised methods tackled the gene classification issue. Between the most common methods, we can mention: hierarchical clustering, k-means, Diana, Agnes, Fanny [164], model-based clustering [18] support vector machines SVM, self organizing maps (SOM), and even association rules (see more details in [75]).

The target of these methods is to classify genes into clusters sharing similar gene expression profiles, as shown in the first step of FIG. 5.3.

2.-Biological annotations integration

Once clusters of genes are built by similar expression levels, each gene annotation is extracted from sources of biological information. As in prior axis, this step deal with different formats of information. A list of annotations is composed for each gene, and then all the annotations are integrated into the clusters of genes (previously built by co-expression profiles). Thus, subsets of co-annotated and co-expressed gene groups are built within each cluster. FIGURE 5.3 illustrates this process: three clusters of similar expression profiles are first built, and then all the individual gene annotations are collected to be incorporated in each cluster. For example in the first under-expressed green group we have found three subsets of co-annotated

genes. These subsets are respiratory complex: Gene E and Gene D, gluconeogenesis: Gene G and Y and tricarboxylic acid cycle Gene E and Gene T. We can observe intersections of genes within the under-expressed cluster because of the different annotations that each gene may have. Thus, we obtain all the co-annotated gene groups.

3.-Selection of the significant co-annotated and co-expressed gene groups

At this stage all the co-expressed and co-annotated gene groups are built and the issue is to reveal which of these groups or the possible subgroups are statistically significant. The most current technique in use is the statistical hypothesis testing (see FIG. 5.3).

Afterward, this full three-step methodology the expression-based approaches present the interpretation results as significant co-expressed and co-annotated groups of genes.

The next section presents some of the most representative approaches and methods of the expression-based axis. Since these approaches are quite numerous, we have classified them according their main source of biological information. Thus, we have the following classification: minimal information approaches, ontology approaches and bibliographic source approaches.

5.3.2 Remarkable expression-based semantic approaches

Expression-based semantic approaches integrate fundamentally semantic annotations (contained in ontologies, thesaurus, semantic networks etc.) into co-expressed gene groups. Nowadays, semantic sources of biological information i.e. structured and controlled vocabularies are one of the best available sources of information to analyze microarray data in order to discover meaningful rules and patterns [13] (as explained in section 3.7).

Actually, expression-based semantic approaches are widely exploited. In this section we present seven among them: FunSpec [258], OntoExpress [99], Quality Tool [128], EASE [151], THEA [228], Graph Theoretic Modeling [175] and GENERATOR [234]. Each approach uses Gene Ontology (GO) as source of biological annotation, sometimes combined with another gene/protein-related specific source of information as: KEGG, MIPS, and SwissProt (all explained in chapter 3).

During last years, GO has been chosen preferably over other sources of information, because of its non ambiguous and comprehensible structure. That is the reason of the recent explosion of many more expression-based GO approaches. Among these approaches, we can cite the integration tools which integrate gene expression data with GO as GoMiner [114], FatiGO [5], Gostat [23], GoToolbox [196], GFINDer [204], CLENCH [275], BINGO [191], etc. This up to date GO COMPENDIUM gives more integration methods, GO searching tools, GO browsing tools and related GO tools.

In the next section, we describe seven remarkable expression-based semantic solutions.

1. FunSpec: web-based cluster interpreter

This approach [258] proposes a statistical evaluation of groups of co-expressed genes and proteins with respect to existing annotations.

It takes as input clusters of genes previously built by similarity in expression. Then it searches for all gene and protein annotations in four biological sources of information: Gene Ontology (GO), Munich Information Center for Protein Sequences (MIPS), Nucleotide sequence database (EMBL), Protein families of alignments and HMMs (Pfam). It builds all the subsets of co-annotated and co-expressed gene and protein groups within each cluster. It makes the selection of the significant subsets (really functionally enriched) via hypothesis testing. It states the null hypothesis:

$$H_0: \quad \textit{A functionally enriched group of genes is associated by chance within the cluster of co-expressed genes.}$$

This one-tailed hypothesis is solved on the basis of an hypergeometric distribution and using a $p - value$ calculated from the cumulative distribution as in Fisher's exact test [116]. A Bonferroni correction is applied to compensate for multiple testing. Finally, it assesses the statistical significance of each group comparing the $p - value$ against a user-determined α value (more details in [165]).

2. Onto-Express: global functional profiling of gene expression

This approach [99] proposes several statistical evaluations of co-expressed gene groups with respect to GO existing annotations. It takes as input clusters of genes previously built by similarity in expression. In a second step, it takes all the existing GO annotations included in three ontologies, molecular function, cellular component and biological process. Then, it builds all the subsets of co-annotated and co-expressed gene groups within each cluster.

In a third step, it makes the selection of the significant subsets rejecting the null hypothesis:

$$H_0: \quad \textit{A GO annotated group of genes is associated by chance within the cluster of co-expressed genes.}$$

This one-tailed hypothesis is solved using a probability distribution and using a $p-value$ calculated from the cumulative distribution. Finally, it assesses the statistical significance of each group comparing the $p - value$ against a user-determined α value. Onto-Express gives the following test options: binomial distribution [113] (when the number of genes is very large), Fisher's exact test [193] (when the number of genes is not too important), and χ^2 test for equality of proportions [115].

3. Quality tool: judging the quality of gene expression-based clustering methods

This approach [128] proposes a measure for testing the quality of clusters of gene expression profiles based on mutual information between cluster membership and known gene annotations. In a first step, it takes clusters of co-expressed genes. In a second step, it takes all the existing GO annotations included in the three ontologies: molecular function, cellular component and biological process. Then, it builds a wide matrix of GO attributes for all genes containing 1 if the gene matches the attribute and 0 if not. It builds a contingency table for each cluster-attribute pair, from which it computes cluster-attribute entropy and mutual in-

formation [80]. In a third step, it compares this measure with clusters grouped by chance from the same microarray experiments, to check if they are better than random clusters.

This approach uses the same one-tailed hypothesis as seen before (Onto-Express and FunSpec), but it supposes a normal distribution and uses $z - score$ statistic for calculations. Finally, it obtains co-expressed and co-annotated significants groups of genes.

4. EASE: identifying biological themes within lists of genes

This approach [151] provides a friendly interface for quick annotation of genes within a cluster, giving a selection method for co-expressed and co-annotate gene groups. In a first step, it takes clusters of co-expressed genes (previously made by classification algorithms). In a second step it takes the available gene annotations from GO, KEGG, Swiss-Prot, PFAM, SMART. Then, it builds all the subsets of co-annotated and co-expressed gene groups within each cluster. In a third step, it shows the statistically significant co-expressed and co-annotated gene groups.

This approach uses the same one tailed hypothesis testing assumptions: null hypothesis, hypergeometric distribution, fisher's exact test, $p - value$ and α as used in Onto-Express and FunSpec. The only difference is the use of an alternative statistic named $ease - score$, which is a conservative adjustment that weights statistical significance in favor of co-annotated groups supported by more genes.

5. THEA: tools for high-throughput experiments analysis

This approach [228] proposes a set of tools designed for manipulating microarray results obtained by hierarchical clustering trees. It integrates gene annotations from biological sources of information and evaluates co-expressed and co-annotated groups of genes.

It takes as input clusters of genes obtained by a hierarchical clustering algorithm. Then, it queries a database in order to obtain all the possible gene annotations from the ontologies in GO on biological process, molecular function and cellular component. Then, it shows all the possible subsets of co-annotated and co-expressed gene groups within each cluster. It displays graphically the statistical evaluation of the co-expressed and co-annotated gene groups. This approach uses the same one tailed hypothesis: H_0, Fisher's exact test, $p - value$ and α set of values as used in Onto-Express and FunSpec.

6. Graph-theoretic modeling

This approach [175] extracts common GO annotations of the genes within a cluster of co-expressed genes through the modified structure of gene ontology called GO tree.

In a first step, it takes as input clusters of co-expressed genes obtained with any clustering technique. In a second step, it annotates all genes in a cluster with GO terms, taking into account the hierarchical nature of GO. It proposes a quantitative measure for estimating how well gene clusters of expression profiles are gathered together along with known GO categories. This measure is based in a graphical distance between nodes in the directed acyclic graph (DAG) of GO. In a third step, it compares this quantitative measure with the same measure taken from random clusters to see if it is better or not. Thus, it obtains co-expressed and co-annotated significants groups of genes.

7. GENERATOR: theme discovery from gene lists for identification and viewing of multiple functional groups

This approach [234] takes co-expressed gene groups and it splits them into homogeneous co-annotated significant groups within each group.

In a first step, it takes co-expressed gene groups. In a second step, it takes all GO annotations (studying each GO ontology separately) for each gene group. Then, it runs a clustering algorithm based in a Non-negative Matrix Factorization (NMF) to create a k-means (begins with k=2) partition of co-annotated groups within each gene group. This process is repeated, applying k-means algorithm (increasing each time the number of k clusters) and building a non-nested hierarchical clustering tree. At each step, it tests for significant co-expressed and co-annotated groups. For this purpose, it uses one-sided test hypothesis with the same assumptions: null hypothesis: H_0, hypergeometric distribution, fisher's exact test, $p - value$ and α as used in Onto-Express.

5.3.3 Remarkable expression-based bibliographic approach

Nowadays bibliographic databases represent one of the richest update sources of biological information. This type of information, however, is under-exploited by researchers because of the highly unstructured free-format characteristics of the published information and because of its overwhelming volume. The main challenges coming up with bibliographic databases integration are to manage interactions with textual sources (abstracts, articles etc.) and to resolve syntactical problems that appears in biological language like synonyms or ambiguities. At the moment, some text mining methods and tools have been developed for manipulate this kind of biological textual information. Among these methods we can mention Suiseki [35] which focuses on the extraction and visualization of protein interactions, MedMinder [299] takes advantage of GeneCards as a knowledge source and offers gene information related to specific keywords, XplorMed [236] which presents specified gene-information through user interaction, EDGAR [255] which extracts information about drugs and genes relevant to cancer from the biomedical literature, GIS [67] which retrieves and analyzes gene-related information from PubMed abstracts. These methods are useful as stand-alone applications but they do not integrate gene expression profiles.

We define expression-based bibliographic approaches as methods that integrates at least one of the bibliographic databases (Medline, Biosis, MeSH, etc.) annotations into co-expressed gene groups. Only a small number of approaches have integrated this kind of biological information into co-expressed gene groups. Masys et al. [205] proposed to use keyword hierarchies to interpret gene expression patterns for integrating bibliographic databases.

In a first step, his method proposes to take as input clusters of genes grouped by similarity in expression (previously built by any of the supervised or non supervised methods). Second, it searches for gene indexing terms contained in some PubMed articles. Then, it translates these indexing terms to MeSH "keywords" terms. Later, it combines the UMLS knowledge, the enzyme code nomenclature and MeSH terms to build hierarchical groups of genes classified by annotation. Third, it makes the selection of the significant groups of co-

annotated genes in each co-expressed cluster. For this purpose, it states the following null hypothesis: H_0 : *Keyword would appear at or above the observed frequency by chance in a group of keywords of the same size within the cluster of co-expressed genes.*

This hypothesis test is solved by comparing the observed versus the expected frequency of each keyword retrieved in association with a set of genes and a $p - value$ estimate of the likelihood under the null hypothesis. Finally, it obtains co-expressed and co-annotated significant groups of genes.

5.3.4 Comparison between several expression-based approaches

TABLE 5.2 presents a brief summary of eight expression-based approaches. The comparison is based on four characteristics: the source of biological information, the hypothesis-testing type and statistics, the hypothesis-testing distribution and a distinctive characteristic.

Approach	**Biological Source of Information**	**Hypothesis-testing Type and Statistics**	**Hypothesis-testing Distribution and details**	**Distinctive Characteristic**
FunSpec (Robinson et al. 2002)	GO, MIPS, EMBL and Pfam	One-tailed test Fisher's exact statistic	Hypergeometric Bonferroni Correction	Online integration of 4 different sources of biological information
OntoExpress (Draghici et al. 2002)	GO (MP, BP and CC)	One-tailed test Fisher's exact statistic χ^2 statistic	Binomial Hypergeometric χ^2	Choice of 3 different statistical methods
Quality Tool (Gibbons et al. 2002)	GO (MP, BP and CC)	One-tailed test z-score	Normal	Measure based in cluster-attribute Entropy and mutual information
EASE (Hosack et al. 2003)	GO, KEGG, Pfam, Smart, and SwissProt	One-tailed test Fisher's exact statistic	Hypergeometric Ease correction	Friendly interface for quick gene annotation
THEA (Pasquier et al. 2004)	GO (MP, BP and CC)	One-tailed test Fisher's exact statistic	Hypergeometric Binomial Bonferroni Correction	Friendly interface for quick annotation and cluster's analysis
Graph Theoretic Modeling (Sung 2004)	GO (MP, BP and CC)	One-tailed test Average PD statistic	Non-Parametric	Graphical method who proposes an Average statistic for cluster's significance
GENERATOR (Pehkonen et al. 2005)	GO (MP, BP and CC)	One-tailed test Fisher's exact statistic	Hypergeometric	Non-negative matrix factorization to create k-means partition. Results presented as a non-nested hierarchical tree
Annotation-Tool (Masys et al. 2001)	Medline (abstracts), Mesh (keywords), UMLS	One-tailed test Estimated likelihood Vs. Observed likelihood	Semi-Parametric: Empirical Likelihood	Hierarchical groups of co-annotated groups within co-expressed clusters

TABLE 5.2: EXPRESSION-BASED APPROACHES

All the approaches appear in chronological order, the first one integrates bibliographic sources of information i.e. Medline abstracts and the seven others integrate semantic sources of information principally GO, sometimes combined with another gene/protein related specific source as MIPS, KEGG, Pfam, Smart, etc. or molecular database as Embl, SwissProt, etc.

Concerning selecting co-expressed and co-annotated gene groups all the approaches have chosen a one-tailed test. FunSpec, OntoExpress, EASE, THEA and Generator have opted for Fisher's exact statistic, and their statistical evaluation methods have small variations. FunSpec, THEA, EASE, Generator have used the typical fisher's test with hypergeometric distribution. The first two of these have chosen bonferroni correction against multi-testing problem and EASE has used an ease-score correction against the over-representation weight given in bigger gene groups by Fisher's test. Only two approaches Graph Theoretic Modeling and AnnotationTool have chose non-parametric and semi-parametric statistical evaluation models respectively.

The last column in TABLE 5.2 contains an important distinctive feature. For example GENERATOR uses a particular method based on $k - means$ that builds a non-nested hierarchical tree, as final result.

5.4 Co-Clustering Axis

From the beginning of gene expression technologies, clustering algorithms were focused on grouping gene expression profiles with biological conditions [257]. Sources of biological information and well structured ontologies as GO and KEGG particularly, are constantly growing in quantity and quality and have opened the interpretation challenge of grouping heterogeneous data as numeric gene expression profiles and textual gene annotations. Co-clustering approaches focus their effort to answer this challenge. Each co-clustering approach has its specific parameters: biological source of information, clustering method and integration algorithm. They generally follow a three-step methodology described in the following.

New co-clustering integration approaches are currently one of the interpretation challenges in gene expression technologies. At the moment, few co-clustering approaches have been reported since the principal barrier is the difficulty to build clustering methods fitting heterogeneous sources of information. Among the co-clustering approaches we can cite Co-Cluster [139], Bicluster [186], ARD [61] and GENMINER [201] described in remarquable algorithms section.

5.4.1 Co-clustering methodology

In a first step, they state two different measures: one measure to manipulate gene expression profiles and the other one for gene annotations in an independent manner.

In a second step, they apply an integration criterion (merging function, graphical function etc.) within the co-clustering algorithm for building the co-expressed and co-annotated gene groups simultaneously.

Finally they select the significant co-expressed and co-annotated gene groups. In the last step, they apply either hypothesis significance test (explained in section 5.3) or testing the quality of the resulting clusters as in: [336], [138], [15], [27], [85], [130] and [285].

5.4.2 Remarquable co-clustering methods

1. Co-cluster: co-clustering of biological networks and gene expression data

This approach [139] constructs a merging distance function which combines information from gene expression data and metabolic networks, computing a joint clustering of co-expressed genes and vertices (annotations from KEGG database) of the network.

In a first step, it computes two distances: a network distance obtained from the proximity of enzymes in the metabolic pathway network beneath undirected graph form, and a gene expression distance obtained from Pearson correlation coefficients of expression matrix [109].

In a second step, it builds a merging function that consists in a mapping that relates genes to enzymes nodes in the undirected graph. Then, it applies hierarchical average linkage clustering algorithm using the merged (enzyme-gene) distance.

Finally, it evaluates the significant co-expressed and co-annotated clusters using the silhouette coefficient [260]. This quality cluster method determines the number of optimal clusters in a hierarchical dendrogram.

2. Bi-cluster: gene ontology friendly bi-clustering of expression profiles

This approach [186] directly incorporates Gene Ontology information into the gene expression clustering process, using Smart Hierarchical Tendency Preserving clustering algorithm (SHTP). HTP is a bi-clustering algorithm capable of discovering gene expression patterns embedded in only a subset of conditions. It becomes "Smart" when it integrates the GO functional annotations.

In a first step, it calculates two trees, the Tendency Preserving (TP) Cluster tree obtained from gene expression matrix (rank measures) and the Gene Ontology tree decomposition obtained from GO gene annotations.

In a second step, it builds a hierarchical structure by mapping the TP cluster tree onto GO Hierarchy.

While applying HTP clustering algorithm, the GO annotations tree is useful for two purposes: assessing functional enrichments of a cluster (using one-tailed Fisher's test as shown in OntoExpress) and selecting the subset of conditions critical to a function category (building the α threshold). Finally, the subset of co-expressed genes contained in the subset of the GO annotations tree becomes the selected significant group of co-annotated and co-expressed genes by tendency.

3. ARD: integrated analysis of gene expression by association rules discovery

This approach [61] combines gene annotations and expression profils data to discover intrinsic associations among both data sources based on co-ocurrence patterns. It uses association rules discovery mining technique for patterns extraction. The gene annotations are obtained from semantic sources of information as GO and KEGG; nomenclature databases as SGD and HUMAN; and transcriptional regulators for saccharomyces cerevisae data [176]

In a first step, it prepares the data: collecting all gene annotation from the mentionned sources of information and discretizing in three values: over-expressed, under-expressed and

no expressed the expression profiles measures. So it builds a big matrix containing all genes and theirs characteristics (gene discretized profiles and gene annotations).

In a second step, it stablishes an antecedent constraint: gene annotations would be at the left part of the rule and the expression discretized measures at the right part of the rule. Then it applies a priori algorithm, Agrawal [4], fixing the support parameter and generating the association rules itemset.

Finally, it selects significant association rules applying two filters: redundant filter and single antecedent filter and taking those rules which confidence is greater than the user-specified minimum threshold value α. The redundant filter refers to taking (when support and confidence are equal) those rules with the longest consequent or antecedent. Single antecedent filter refers to taking only rules whose antecedent contains more than one item. As informative value, it calculates the statistical significance of an association between antecedent and consequent, using χ^2 test for statistical independence [115]

4. GENMINER: Gene-integrated analysis using association rules discovery

As ARD method, this approach [201] integrates gene annotations and expression profils data to discover intrinsic associations among both data sources based on co-ocurrence patterns. It uses association rules discovery mining technique for patterns extraction. The gene annotations are obtained from any of the six sources of biological information explained before, including qualitative variable microarray information of biological conditions as: age, sex, state etc.

In a first step, it prepares the data: collecting all gene annotation from the mentionned sources of information and discretizing expression profiles measures. An adapted discretization algorithme is created taking into account the variabilities between biological conditions. It builds several scenarios of possible matrices containing all genes and theirs characteristics: gene discretized profiles and gene annotations. An scenario corresponds to a discretization method (voir chapter VII section discretization).

In a second step, it applies CLOSE algorithm [229] for generating the association rules itemset (at one support parameter value). Close algorithm allows the use of all available matrix information without limiting constraints for rules extraction. In calculation issue is more efficient than *a priori* algorithm when the items are dependent to each other (that is the genes case) because it reduces the problem of finding frequent itemsets to finding frequent closed itemsets. We can traduce this to non negliable calcul time reduction produced by a reduction in rule research space.

Finally, it selects significant association rules applying a redundant filter and taking those rules which confidence is greater than the user-specified minimum threshold value α. The redundant filter refers to taking (when support and confidence are equal) those rules with the smallest antecedent and biggest consequent. This filter is shown to give the most informative rule, pruning redondant items.

5.4.3 Comparison between co-clustering approaches

TABLE 5.3 presents a brief summary of the four co-clustering approaches explained in last subsection. It is based on four parameters: source of biological information; expression profile measure and gene matrix measure; co-clustering details; co-expressed and co-annotated gene groups selection details as seen in TABLE 5.3.

Approach	**Biological Source of Information**	**Gene Expression Profiles Measure and Gene Matrix Distance**	**Co-clustering Details**	**Co-expressed and Co-annotated gene group Selection Details**
Co-Cluster (Hanisch D. et al. 2003)	KEGG	Fold Change Pearson Correlation distance	Hierarchical Average Linkage	Silhouette Coefficient
GO Bi-clustering (Liu J. et al. 2004)	GO: (MP, BP, CC) and KEGG	Fold Change Rank between conditions	SHTP: Smart Hierarchical Tendency preserving	One-tailed Fisher's test Alfa threshold construction
ARD (Carmona et al. 2006)	KEGG and transcriptional regulators.	Fold change. Discretization in: $\{-1, 0, 1\}$ values.	ARD method with Apriori algorithm.	Support and Confidence Thresholds with Redundant (minimal at right, maximal at left) and Single Antecedent Filter.
GENMINER (Martinez et al. 2006)	GO: (MP, BP, CC), KEGG, Pubmed/Medline, transcriptional regulators, phenotype and protein interactions.	Several discretization scenarios: Fold Change, Equal Frequencies, Fixed thresholds and Nordi Algorithm.	SHTP: Smart Hierarchical Tendency preserving	Support and Confidence Thresholds with CLOSE non-redundant filter.

TABLE 5.3 CO-CLUSTERING INTEGRATION APPROACHES

The four approaches have use the gene-protein related KEGG database. Bi-cluster and GENMINER have also chosen the well-structured semantic source: Gene Ontology. ARD and GENMINER have introduced another sources of information as the transcriptional regulator information that bind to promoter regions (see [176]). GENMINER algorithm has also chosen the bilbiographic datasource Medline/PubMed and several qualitative variables contained in the MIAME data source of the microarray experiment. The generalized choice of Gene Ontology and KEGG database is because they present an implementation advantage: they are well-structured and they have a graph-based representation for gene or protein annotations. In one hand we have that co-cluster algorithms [186] and [139] are biological source dependent (because of the intrinsic relations between algorithm and biological source. On the other hand association rules algorithms [61] and [201] present more suppleness corcerning the source of biological information used. This is clearly seen in GENMINER approach [201] that also introduces bibliographic sources of information and qualitative variables as gender, state, family etc.

Concerning gene expression measures input, all four methods use *fold change* expression measures. Nevertheless, they make differents choices for manipulating this gene expression measures. Co-cluster algorithm chooses Pearson's correlation coefficient as gene matrix distance calculation tool. Bi-Cluster chooses a gene tendency measure based in the gene-rank between biological conditions. ARD and GENMINER have chosen discretization methods. ARD takes a fixed threshold discretization in three intervales$\{-1, 0, 1\}$ corresponding

to underexpressed, no expressed and overexpressed genes. GENMINER choose different discretization scenarios: an original equal frequencies intervals method and also fixed threshold intervals method.

Related to co-clustering details, both co-cluster and bi-cluster have chosen a hierarchical clustering method. However, co-cluster has opted for typical hierarchical average linkage algorithm and bi-cluster has developed the Smart Hierarchical Tendency preserving (SHTP) algorithm. In the case of the association rules algorithms, they have taken different asociation rules discovery methods: ARD has choose the "typical" a priori algorithm [4] and GENMINER has prefered CLOSE algorithm [229] based in frequent closed itemsets selection.

Related to gene group selection, co-cluster uses the silhouette coefficient for determining the quality of the clusters built (selecting the significant ones). In the other hand, bi-cluster states for a selection in two different stages. First it uses standard one-tailed Fisher's test for calculate the $p - value$ for the co-annotated and co-expressed gene groups and then it builds a particular α threshold for each of them. Finally, as seen in the previous approaches, it compares $p - value$ against α to select or not the co-expressed and co-annotated gene group (this methodology was explained in section 2.2.3).

Association Rules algorithms have taken support and confidence thresholds, but ARD have use two filters: single antecedent filter and redundant filter (minimal items at right maximal at left) and GENMINER have use a redundant filter that represents the opposite choose of ARD (minimal at left and maximal at right). Concerning redundant filter GENMINER assures the chosen rules only contains the minimal necessary information (getting ride of superfluous rules). GENMINER has also integrated the one tailed Fisher's test for calculate the $p - value$.

5.5 Discussion

Here we present a discussion on all three interpretation axes about their main methodology characteristics: biological source of information, expression profiles measurement and selection of significant gene groups.

Biological source of information

The improvement of the interpretation approaches come hand in hand with biological information source development, without glance of the interpretation axis. Interpretation approaches have to tackle the integration task for best exploiting the biological source of information. Here, we see the current state of the interpretation axis approaches facing each of the six sources of biological information described in chapter 3.

In almost all-three interpretation axis approaches the sources of biological information used were: semantic sources such as GO and gene/protein related specific sources such as KEGG combined with molecular databases (see Table 5.1, 5.2 and 5.3). This comes from the need of integrating well-structured sources of information. The "ideal" source of biological information should be: well-structured, easy-handling, clearly-explained and up to date.

Bibliographic databases, alternative choice and one of the richest and up to date, unfortunately present a strong integration barrier: the natural language textual format. Dealing with this kind of information is the goal of the *text-mining* field which extracts high-quality information from text (discussed in section 2.4). Nowadays, this field is becoming intensively studied and is a wide-open research field.

Gene expression databases represent another important source of information poorly exploited. Last years, meta-analysis techniques have been dealing with this source of biological information. Meta-analysis is a classical statistical methodology for combining results from different studies (contained in the gene expression databases) addressing the same scientific questions. Meta-analysis algorithms have focused to integrate only the expression profiles data among different studies. Some of them have recently been applied to the microarray data analysis: [254], [127], [70], [227], [158] and [240]. The difficulties faced to integrate biological knowledge to one study (as seen before) and inter-studies specifications have limited the meta-analysis results.

Concerning Minimal microarray information integration contained in the biological conditions specification such as: quantitative variables, such as gene expression or time, etc., and qualitative variables, such as tissue, gender, age etc. The gene expression data analyists have focused in the integration of gene expression profiles with gene annotations (as seen in section 2.4). Therefore all-three interpretation axis have somehow denied the existence of qualitatif data provided often by the experimenters in their gene expression data analysis. The most important reasons are: restricted access to this kind of information and algorithmic difficulties of incorporating them. Several Statistical methods applied in epidemiology science have been tackled similar qualitative and quantitative source of information, a survey can be found in [84].

Gene expression databases represent another important source of information poorly exploited. The integration of bibliographic databases, experiment databases, and minimal microarray information is thus an open research field.

Gene expression measures handling

This is one of the most crucial issues in microarray data analysis, it answers to the question: How can I best manipulate raw gene expression measures avoiding lose of information?

Before any type of gene expression measure manipulation the gene expression data analyst receives the gene expression data in a raw form. Often in microarray experiments the data comes as the $fold\ change$ measure as fully explained in section 2.1, or it can come in another intensity measurement as the Signal to Noise Ratio* (SNR).

Fifteen of the sixteen interpretation algorithms described among the three interpretation axis has chosen the commonly used fold change as gene expression measure, only the prior algorithm GSEA [215] has opted for SNR measure. The choose of a gene expression measure is determinant in the final interpretation results.

Once the raw gene expression measure is chosen for data analysis, it would be constantly manipulated along the data analysis five steps. The handling of raw gene expression measures have to fit next expression data analysis tasks such as clustering or prediction tasks. Here,

we have distinguished two different gene expression measures manipulation among all-three interpretation approaches: gene-distance matrix and differentially-expressed genes.

Distance gene matrix calculation: This is currently a clustering issue [164]. Clustering algorithms rely on proximity measurements to evaluate the distance or similarity between a pair of objects or genes. For more details in clustering algorithms and distance measure the reader can see section 2.3.

This gene expression measure manipulation is currently use in clustering techniques [164], [18], [75]. It exists a non negliable choice between proximity measures: euclidean, manhattan pearson, spearman, kullback-lieber divergence, kramer-tau, ... [334]. The distance choice is a crucial parameter for the distance gene matrix calculation that must be adapted to the data properties (time series, cancer studies, multi-tissue studies etc.). Almost all expression-based approaches (as FunSpec [258], OntoExpress [99], EASE [151], THEA [228], Graph Theoretic Modeling [175], GENERATOR [234] Annotation Tool [205]) and one co-clustering approaches [139] contain a clustering algorithm within the method. Thus, the need of a careful and validated choice of a distance measure for gene matrix calculation.

Differerentially expressed genes analysis: Differentially-expressed genes calculation identify those genes which demonstrate a significant change in expression level under the impact of certain experimental conditions. Several statistical methods have been applied for measuring this variability such as: fold change, parametric test, and non parametric test methods [257] (explained in section 2.2).

All prior axis approaches have chosen differentially expressed gene analysis as main gene expression measure manipulation. PAGE [167] and CGGA [203] have opted for parametric test methods: $z - score$ and $F - score$ repectively. iGA [53] and GSEA [215] have chosen a fold change and non-parametric methodology respectively for differentially expressed gene analysis.

Furthermore, some approaches use the results of differentially expressed genes analysis to rank the genes. Then, it takes advantage of this ranking to integrate gene expression measures in their algorithms as stated in prior approaches iGA [53] and CGGA [203], and also the co-clustering approach Bi-Cluster [186].

Discretization of gene expression measures: Discretization concerns the process of transferring continuous data into discrete counterparts. This process is usually carried out as the fiirst step toward making gene expression measures suitable for several utilizations as applying supervised algorithms as decision trees, SVM, neural networks or association rules [75].

Several methods have been applied to deal with discretization issue in bioinformatics as: biological methods, statistical methods and mining methods (fully explained in section 7.5).

Two co-clustering approaches have used discretization methods: ARD[61] and GENMINER[201]. The first one have used the commonly use biological method known as 2-fold change cut-off which determines fixed boundaries for discretize, which are supposed to be "experimentally correct" (even if it is not the case experimentally by the studied data set). The second one use the mixed biological-statistical method NORDI algorithm which

takes adavantage of two methods: 2-fold change cut-off and the statistical z-score method for calculating the discretization cut-offs (explained in section 7.5).

Discretization is a risky step, where we need to known the particular characteristics of the data: time series, cancer studies, mutation studies, as discussed in section 7.5. Discretizated measures have to summarize the "best" the information contained in the original gene expression measures. Evidently, it would exist a loss of information, but this lack can be overwhelm carrying out different scenarios of discretization and comparing the interpretation results.

Gene expression measures manipulation is a non-obvious step. We have to be very careful in the expression data characteristics as well as the whole biological process for taking the best adapted gene expression handling choice. It's so important that the final results can be severely biased if we make an incorrect choice or assumptions. A wrong choice can generate final results severely biased and a wasteful lose of information.

Significant gene groups selection

One common point in all-three interpretation axes is the step of selection of significant co-annotated and co-expressed gene groups. Once co-annotated and co-expressed gene groups are built we need to select the significant groups. This question can be reformulate in different ways: testing the significance of the clusters, testing the reliability of obtained clusters or measuring the quality of the resulting clusters etc. The reader can see section 2.2 and 2.3 for a description of several methods for realizing each of these three tasks.

In order to answer to this question, standard and prior axes preferably choose statistical *hypothesis testing* for selecting the significant gene groups (explained in chapter 2). A survey in statistical methods for microarrays concerning statistical approaches for *hypothesis testing* was made by Sokal et al. [287] and Zhang et al. [334].

Almost all expression-based and prior interpretation approaches choose parametric tests for gene group selection as: the *Fisher exact* test* (used in: [53], [203], [258], [99], [151], [228], [234]), including one of the co-clustering axis ([186]) and the $z - score$ test* which is used in: [151] and [167]. Only two integration approaches have preferred non-parametric selection methods ([215] and [175]) or semi-parametric methods [205].

Otherwise, co-clustering approaches have opted for *cluster quality* techniques, which assess the results of a clustering technique. Some cluster quality techniques can be found in [336, 138, 15, 27, 85, 130, 285]. That is the case of the Co-Cluster approach [139] which uses the *silhouette coefficient* graphical measure (more details in [260]) to validate the resulting clusters.Whereas [201] and [61] have used the support and confidence measures for gene groups selection.

Selection of significant gene groups is indeed a significant part of the interpretation step. Analysts have to be aware of the characteristics of the gene groups and the whole approach for doing the best choice between hypothesis testing (parametric, non-parametric and semi-parametric) or quality clusters methods. Once the method is chosen, all assumptions, hypothesis and parameters have to be stated carefully. All variables have to fit data characteristics, avoiding at most all possible biases.

5.6 Conclusion and Outlook

The bioinformatics community has developed many approaches to tackle the interpretation microarray challenge. In this chapter, we classify them in three different interpretation axes: prior, standard and co-clustering.

Standard or expression-based approaches give importance or weight to gene expression profiles measures. However, microarray history has revealed intrinsic errors in microarray measures and protocols that increases during the whole microarray analysis process. Thus, the expression-based interpretation results can be severely biased [112], [185]. Although widely used, standard approaches, they are based in "typical" clustering expression techniques that shown some well-known drawbacks:

1. The underlying assumption in clustering analysis step is that genes sharing similar expression profiles also share similar biological properties. Nevertheless, simultaneously expressed genes may not always share the same function or regulatory mechanism. Even when similar expression patterns are related to similar biological roles, discovering these biological connesctions among co-expressed genes is not a trivial task and requires a lot of additional work [276].
2. Current clustering algorithms (except bi-clustering techniques) group genes whose expression levels are similar across all conditions. However, a group of genes involved in the same biological process might only be co-expressed in a small subset of experimental conditions [9].
3. Many genes can be conditionally co-expressed with different sets of genes, which may refelct the different biological roles that a gene product can play in the cell. Most of the commonly used clustering algorithms group genes into single clusters, which mask these complex relationships among different sets of conditionally regulated genes [122]

On the other hand, prior or knowledge-based approaches give importance or weight to biological knowledge. Nevertheless, all sources of biological information fix many integration constraints: the database format or structure, the weak quantity of annotated genes, the availability of data, the maintainance of up-to-date and well revised annotations for instance. Consequently, the *knowledge-based* interpretation results can be poor or somewhat quite small or limitated in relation to the whole studied biological process.

Co-clustering approaches represent the best compromise in terms of integration, giving the "same" weight to expression profiles and biological knowledge. But, they have to deal with the algorithmic issue of integrating these two elements at once. So, they are often forced to give more weight to one of these elements.

In co-clustering section, we have seen two approaches: co-cluster algorithm [139] which gives more weight to knowledge, and expression profiles were used to guide the clustering analysis while bi-cluster algorithm [186] gives more weight to tendency in expression profiles and GO annotations are used to guide the clustering analysis. At this issues association rules discovery approaches [201] and [61] are the best weight deal between expression profiles measures and biological knowledge, it can be unequal to one side, to the other or almost equal.

Indeed, with the constant improvement of microarray data quality, microarray data process analysis and the completion of biological information sources; interpretation results would become better and better, independently of their interpretation approach.

As long as there is not enough reliability on these main elements, the choice of the interpretation axis approach and its parameters (source of biological information, profiles expression measure manipulation, significant gene group selection) remains of crucial importance for the final microarray interpretation results.

Part III: Data Mining Models for Knowledge Discovery via Biological Interpretation

Chapter 6

Co-expressed Gene Groups Analysis (CGGA): An automatic Tool for the Interpretation of Gene Expression Experiments

Gene expression technology produces vast amounts of data by measuring simultaneously the expression levels of thousands of genes under hundreds of biological conditions. Nowadays, one of the principal challenges in bioinfor-matics is the interpretation of huge data using different sources of information.

We propose a novel data analysis method named CGGA (Co-expressed Gene Groups Analysis) that automatically finds groups of genes that are functionally enriched, i.e. have the same functional annotations, and are co-expressed.

CGGA automatically integrates the information of microarrays, i.e. gene expression profiles, with the functional annotations of the genes obtained by the genome-wide sources of information such as Gene Ontology (GO) .

By applying CGGA to well-known microarray experiments, we have identified the principal functionally enriched and co-expressed gene groups, and we have shown that this approach enhances and optimizates the interpretation of DNA microarray experiments.

6.1 Introduction

One of the main challenges in microarray data analysis is to highlight the main co-expressed* and co-annotated* gene groups using at least one of the different sources of biological information [13]. In other words, the issue is interpretation of microarray results via integration of gene expression profiles with corresponding biological gene annotations extracted from biological databases (presented in chapter 3).

This challenge concerns the interpretation fifth step of microarray data analysis procedure which focuses on knowledge discovery via interpretation of the microarray results (fully explained in chapter 5). In other words, the goal of the fifth analysis step is the integration between two domains, the numeric one represented by the gene expression profiles and the knowledge one represented by gene annotations issued from different sources of biological information (see Fig. 2.7).

In order to process the the interpretation step in an automatic or semi-automatic way, the bioinformatics community is faced to an ever-increasingly volume of sources of biological information on gene annotations.These sources of information, constantly growing by an ever-increasingly volume of genomic data, are:

- Minimal Microarray Information (genes, biological conditions, gender, age etc.).
- Molecular databases (GenBank, Embl, Unigene, etc.).
- Semantic sources as thesaurus, ontologies, taxonomies or semantic networks (UMLS, GO, taxonomy, etc.).
- Gene expression databases (GEO, Arrayexpress, Microarray database, etc.).
- Bibliographic databases (Medline, Biosis, etc.).
- Gene/protein related specific sources (ONIM, KEGG, etc.).

A variety of statistical and data analysis approaches, identifying groups of co-expressed genes based only on the expression profiles, i.e. without taking into account prior knowledge, have been reported:[69, 90, 107, 298] (some of them explained in section 2.3). A common characteristic of purely numerical approaches is that they determine gene groups (called clusters) of potential interest; however, they leave to the expert the task of discovering and interpreting biological similarities hidden within these groups.

These methods are useful, because they guide the analysis of the co-expressed gene groups. Nevertheless, their results are often incomplete, because these approaches do not include biological considerations and also, they reject groups of genes which are expressed only under some biological conditions and not under all the biological conditions [185]. Actually, one of the major goals in bioinformatics is the automatic integration of biological knowledge from different sources of information with gene expression data [13]. A first assessment of the methods developed to answer this challenge was proposed by Chuaqui [72].

Nowadays, one of the richest sources of biological annotations is contained on structured and controlled vocabulary such as ontologies. These annotations can be functional, relational and syntactic information on genes. We present here the enrichment of two recently developed interpretation research orientations, *standard* or *expression-based* and *a priori or knowledge-based*, that exploit multiple sources of annotations such as Gene Ontology.

The standard or expression-based methods build co-expressed gene groups. Then they detect co-annotated gene groups Afterwards, the statistical significance of these co-annotated gene subsets is tested (see Fig. 5.3). Among the methods in this axis let us quote FunSpec [258], OntoExpress [99], Quality Tool [128], EASE [151], THEA [228], Graph Theoretic Modeling [175] and GENERATOR [234]. The reader can see chapter 5 for a full explanation of this axis and respective approaches.

The a priori or knowledge-based axis methods first finds co-annotated groups.Then they integrate the information contained in the profiles of expression. Later on, the statistical significance of the co-annotated groups is tested by an enriched score [215], a $p-value$ based on a hypergeometric distribution [53], or a z-score test [167] (see Fig. 5.2). The reader can see chapter 5 for a full explanation of this axis and respective approaches.

Our approach, called CGGA (Co-expressed Gene Groups Analysis), is inspired by the a priori or knowledge-based axis: the functionally enriched groups (FEG), i.e. groups of co-annotated genes by function, are initially formed from the Gene Ontology, next a function, which synthesizes the information contained in the expression data, is applied in order to obtain an arranged gene list. In this list, the genes are sorted by decreasing expression variability. The statistical significance of the FEG obtained is then tested using a parametric test approach, similar to the hypothesis testing presented in *Onto Express*[99]. Finally, we obtain co-expressed and statistically significant FEG.

Our that finds all subsets FEG of significant co-expressed genes with similar level of expression. In contrast the IGA method, limits itself to find only the subsets of FEG of the most or least expressed genes. Thus it eliminates all the co-expressed FEG who are not in ranked in the first places of the ordered list of genes, in descending or increasing order according to the case.

CGGA method is an extension of the IGA algorithm [53]. Indeed CGGA makes it possible to obtain all the possible subsets FEG of significant co-expressed genes. In contrast to IGA method which is limited to find the FEG of most expressed genes or least expressed genes. IGA leaves out all the co-expressed FEG that there are not in the first or in the last ranks of the list of genes ordered by the increasing or decreasing order of expression levels. On the other hand CGGA makes it possible to select FEG that are classified in the middle of the ordered list. For that CGGA takes into account the relative level (relative rank) of the list ranking. The search of all the possible subsets of co-expressed and co-annotated genes in any biological experiment, without limiting to only some groups, increases the chances biological phenomenon comprehension by the expert.

This chapter is organized in the following way: in section 6.2 we describe the validation data as well as the tools used: databases, ontologies, statistical packages; our algorithm CGGA is described in section 6.3; the results obtained are presented in section 6.4 and the last section presents our conclusions.

6.2 Data and Methods

6.2.1 Data set and statistical pretreatment

In order to evaluate our approach, the CGGA algorithm was applied to the DeRisi data set which is one of the most studied in this field [90]. This data set measures the variations in gene expression profiles during the cellular process of diauxic shift for the yeast *Saccharomyces Cerevisiae*. When inoculated into a glucose-rich medium (anaerobic growth), the budding yeast can convert the glucose to ethanol (aerobic respiration), the shift from anaerobic fermentation of glucose to aerobic respiration of ethanol is the so-called *diauxic shift*.

The microarray technique used is spotted cDNA chips, obtained by two color fluorochromes with distinct emission spectra Cy3 and Cy5 (this technique was explained in section 1.2.1). The DeRisi data set contains the expression levels of 6199 ORF's, opening reading

frame, of the yeast (an entirely sequenced organism), for 7 temporal points that correspond to samples harvested at successive two-hour intervals after an initial nine hours of growth.

The data set was pretreated by taking the log_2 ratios (to consider cellular inductions and repressions in a numerically equal way) and applying the imputation algorithm of k-nearest neighbors [184] in order to treat the missing values (1.9% of the total).

6.2.2 Ontology and functionally enriched groups (FEG)

In order to fully exploit data, knowledge discovery systems rely on a formal representation of information based on a well-defined semantic [282]. These formal requirements have led to the utilisation of the well structured semantic source of biological information: Gene Ontology (GO) and the molecular database SGD (Saccharomyces Genome Database). Sructure of Gene Ontology (GO) and the annotations of Saccharomyces Cerevisiae Genome with GO terms were retrieved from the GO database web site on may 2006. Automatic annotations not reviewed by curators (IEA evidence code) were discarded. For each gene product, we have stored all the functional annotations of the gene product and his parents preserving the hierarchical structure of GO.

Gene Ontology (GO): GO is a controlled vocabulary developed by a consortium of scientists to address the need for consistent descriptions of gene products in different databases. It can be used to annotate a gene or gene product by a *GO-term*, with regard to its molecular functions (GO:MF), cellular localizations (GO:CL) and biological processes (GO:BP).

GO-terms are organized in structures called directed acyclic graphs (DAGs), which differ from hierarchies in that a child, or more specialized, term can have many parent, or less specialized, terms. Annotators can assign properties of gene products at different levels, depending on how much is known about a gene [12].

Genome data: In order to be congruent with GO annotations files and among the multiple yeast gene identifiers, we have used the yeast *Saccharomyces cerevisiae* database. SGD is a scientific database of the molecular biology and genetics of the yeast [316].

Functionally enriched groups (FEG): Queries carried out on the GO database have built the whole set of the FEG: each FEG corresponds to a couple made up of a *GO-term* and of the list of genes annotated by this one.

6.2.3 Expression profile measure of the genes

In order to incorporate the expression profile of the genes, we have used a measurement of their variability of expression, *modified t-statistic*, which is more robust than other measurements such as *fold change*[257]. This measurement enables us to build a list of genes, *g-rank*, ordered by decreasing expression variability. We have used the SAM program [309] to calculate the *modified t-statistic* associated with each gene (SAM method was explained in section 2.2.5).

The SAMs modified t-statistic is a t-statistic which adds a corrector term in the denominator. Once SAM have calculate the modified t-statistic for each gene, it choose the genes with scores greater than a threshold as "potentially" significant. To control the false posi-

tives, SAM uses permutation of measurements to estimate the false discovery rate (FDR). The score threshold for genes is then adjusted iteratively according to the FDR until a set of significant genes have been identified. More details of SAM can be found in [309].

6.2.4 Implementation

CGGA program is fully developed in Perl language and it is hosted at:

http://www.i3s.unice.fr/~rmartine/CGGA.

6.3 Co-expressed Gene Groups Analysis (CGGA)

The CGGA is based on the idea that any resembling change (co-expression) of a gene subset belonging to an FEG is physiologically relevant. We say that two genes are co-expressed if they are close in the sense of the metric given by the expression variability (*modified t-statistic*). The CGGA algorithm computes a probability of change named $pc-value$ for each FEG that estimates its coherence (according to the *g-rank*) and thus allows to detect the statistically significant groups.

In order to understand CGGA algorithm we explain first the basics of hypothesis one-tailed test using modified Fisher's exact statistic best known as hypergeometric distribution test, briefly explained in section 2.2.3.

6.3.1 Hypergeometric distribution one-tailed test

Here, we present the basics of the hypergeometric distribution one-tailed test applyied to co-expressed and co-annotated gene groups significance testing (briefly explained in section 2.2.3). A survey in statistical methods for microarrays concerning statistical approaches for *hypothesis testing* was made by Sokal et al. [287].

In section 2.2.1 we have explained the six steps of *hypothesis testing* methodology : defining the problem, generate hypotheses (null hypothesis H_0 and alternative hypothesis, H_1), choose the test statistic, calculate the statistic value, calculate the $p-value$ and fix the significance level α, and finally reject or accept the null hypothesis H_0.

The question of selecting the significant co-annotated and co-expressed gene groups was discussed in section 5.5. In this section, we have seen that the half of the 16 analyzed algorithms misregarding their membership to one interpretation axis approach have used the *Fisher exact* or *hypergeometric test* method.

Significant gene group selection in the case of microarrays concerns in testing the significance of a group of co-expressed genes for a special annotation F. Thus, we have the null hypothesis, H_0 be the hypothesis that a group of co-expressed genes were associated by chance within annotation F, and the alternative hypothesis H_1 be these group of co-expressed genes is not associated by chance within annotation F.

We can translate the H_0 challenge to this probability problem: We have an microarray experiment with N genes, any given gene is either in the co-annotated group or not. So we

have N genes in two categories: F (in the group annotated by F) or F^c non in the group annotated by F. We observe that x of these K genes are F and we want to find out what is the probability of this happening by chance. So, our question is: given N genes of which n are F and $N-n$ are F^c, we pick randomly K genes and we ask what is the probability of having exactly x genes of type F. Once we pick a gene from the chip, we cannot pick it again so this is clearly sampling without replacement.

The probability function that a certain category occurs x times just by chance in the list of differentially expressed genes is appropriately modeled by a hypergeometric distribution [303]:

$$p(X = x|N, K, n) = \frac{\left(\begin{array}{c} K \\ x \end{array} \right) \left(\begin{array}{c} N-K \\ n-x \end{array} \right)}{\left(\begin{array}{c} N \\ n \end{array} \right)}. \tag{6.1}$$

Based on this, the $p - value$ of having x genes or fewer in F can be calculated by summing the probabilities of a random list of K genes having $1, 2, ..., x$ genes of category F [303, 62]:

$$p(X \leq x|N, K, n) = \sum_{i=0}^{x} \frac{\left(\begin{array}{c} K \\ i \end{array} \right) \left(\begin{array}{c} N-K \\ n-i \end{array} \right)}{\left(\begin{array}{c} N \\ n \end{array} \right)}. \tag{6.2}$$

this corresponds to a one-sided test in which small $p - values$ correspond to under-represented categories. The p value for overrepresented categories can be calculated as: $p - value(x) = 1 - p(X \leq x|N, K, n)$ when the sum is larger than 0.5 [99].

In order to accept or reject H_0 we have to fix a significant value α. Thus, if $p - value(x) < \alpha$ then H_0 is rejected, i.e. the co-annotated F group containing co-expressed genes is statistically significant for an α threshold and if $p - value(x) \geq \alpha$ then H_0 is accepted.

An equivalent method for answering this question was created by Fisher in 1922 [116], who has generated an intuitive statistic for contingency tables* known as fisher's exact test (curiously it is equivalent of applying the hypergeometric distribution function, statistic and $p - value$). Another methods that have solved the significant gene groups selection problem were: the χ^2 test for equality of proportions [115], binomial distribution test [113] used for avoiding numeric explosion when calculating hypergeometric probabilities and the $z - score$ statistic that supposes a normal gene group distribution (more details in [257]). The reader can go to section 2.2.3 for more a brief information in these hypothesis testing methods.

All these tests are particularly useful and can be used in different circumstances. One straight option for answering our hypothesis testing problem H_0 is the Fisher's exact test, but when the number of genes, N, is too large to be calculated we can use the binomial distribution test (that is a good approximation of hypergeometric distribution for N large [113]). An alternative and more robust option is χ^2 test for equality of proportions, but we cannot used it for N too small or when their assumptions are not accomplish (see [115]). Under normality assumptions we can choose $z - score$ test. In addition to this test methods, a correction for multiple experiments may be useful since repeated test are conducted to

determine the significance or each co-annotated and co-expressed group. We can mention several useful corrections for resolving this problem: Holm, Bonferroni and bootstrapping [287].

The exact biological meaning of the calculated $p-value's$ depends on the list of genes submitted as input. For example, if the list contains genes that are over-expressed and mitosis appears more often than expected, the conclusion might be that the condition under study stimulates mitosis (or more generally, cell proliferation) in a significant way. If the list contains genes that are underexpressed and mitosis appears more often than expected, exactly as before, the conclusion might be that the condition significantly inhibits mitosis.

A correction for multiple experiments testing may be useful since repeated tests are conductod to determine the significance of a given gene group annotation. Several methods have been developed to answer this question as Holm, Bonferroni and bootstrapping methods (briefly explained in section2.2.5).

6.3.2 CGGA algorithm

The CGGA algorithm first builds the *g-rank* list from the expression levels and the FEG from the GO database. For each FEG of n genes, the algorithm determines the $n(n+1)/2$ gene subsets that we want to test for co-expression. For each subset we compute the $pc-value$ corresponding to the test described below in order decide whenever the genes of the subset are co-expressed.

Let H_0 be the hypothesis that x genes from one of these subsets were associated by chance, given their place on the *g-rank* list. If H_0 is rejected, there are good chances that the genes belonging to the subset are improbably close on the list because they have a very similar expression profile.

To compute the probabilty that H_0 is true for a fixed subset FEG or class, let us ask the question, how likely is to find x members from the class placed this way on the *g-rank* list? The answer to this question is given by the following hypergeometric distribution:

$$p(X=x|N,R_{g(x)},n)=\frac{\left(\begin{array}{c}R_{g(x)}\\x\end{array}\right)\left(\begin{array}{c}N-R_{g(x)}\\n-x\end{array}\right)}{\left(\begin{array}{c}N\\n\end{array}\right)} \tag{6.3}$$

where:

$$p(X=0|N,R_{g(x)},n)=0$$

with:

- N : total number of genes in the data set,
- n : number of genes in the FEG,
- x : position of the gene in the FEG (previously ordered by rank),
- $r_{g(x)}$: absolute rank of the gene of position x in the *g-rank* list,

- $R_{g(x)}$: number of ranks (in the *g-rank* list) between the gene of position x from its FEG predecessor. $R_{g(x)}$ is calculated from the absolute ranks $r_{g(x)}$ according to the formula:

$$R_{g(x)} = r_{g(x)} - r_{g(x-1)} + 1 \; where \; R_{g(0)} = r_{g(x)} = 1. \tag{6.4}$$

Our $pc - value$ corresponds exactly to the hypothesis testing $p - value$ (refer to [99] for further details) that is:

$$p - value(x) = 1 - \sum_{k=1}^{x} p(X = k|N, R_{g(k)}, n). \tag{6.5}$$

In order to accept or reject H_0 we will use the following significance threshold:

$$\alpha = \frac{1}{|\Omega|}, \tag{6.6}$$

where $|\Omega|$ is the number of FEG obtained from all the functional annotations over the N genes of the biological experiment.

Thus for each subset of FEG genes, we test the inequality

$$p - value(x) < \alpha, \tag{6.7}$$

to reject H_0, i.e. the hypothesis that the FEG is statistically significant.

An alternative to fix the α threshold could be to ask a significant value to the expert of the studied biological process. However, this election has to avoid the choice of a relaxed threshold which corresponds to chossing "almost" by chance the significant groups.

Pseudo-code for CGGA algorithm is presented on Fig. 6.1. The algorithm has been implemented in Perl (language). It takes as input the list of annotations for each gene (generated by a query on the database GO database containing all the GO annotations) and the ordered *g-rank* list of the N genes. It returns as output the list of the groups of significant co-expressed genes.

The algorithm begins by computing the α (stage 2) and generating the FEG from the GO annotations (stages 3 to 9). Then it considers successively each FEG (stages 10 to 18). For each FEG, it takes all non-empty subsets and computes the $p - value$ for each of them (stages 11 to 16). If the computed $p - value$ is less than α, the subset is added to the FEG results list (stages 13 to 15). The added subsets that are non-maximal according to the inclusion are deleted (stage 17).

For example, let the FEG_A annotated set,

$$FEG_A = \{g_1, g_2, g_3\},$$

thus we have:

$$results(FEG_A) = \{\{g_1\}, \{g_2\}, \{g_3\}, \{g_1, g_2\}, \{g_2, g_3\}, \{g_1, g_2, g_3\}\}.$$

Then, all the subsets of $\{g_1, g_2, g_3\}$ are deleted from results(FEG_A). Finally, the total result consists of all the groups of co-expressed and significant genes (stage 19).

Input: List of annotations for each gene G: annotations(G).
Ordered list of N genes: *g-rank*.
Output: Results set containing the FEG of co-expressed genes: $results(FEG_A)$.

```
1   Begin
2       compute p-value
3       for each annotation A of the GO do
4           for each gene G do
5               if A ∈ annotations(G) then
6                   FEG_A ← FEG_A U G
7               end if
8           end for
9       end for
10      for each FEG_A do
11          for each subset S of FEG_A do
12              compute pc − value(S)
13              if pc − value(S) < p − value then
14                results(FEG_A) ← results(FEG_A)US
15              end if
16          end for
17          delete from results(FEG_A) the non maximal S according to inclusion
18      end for
19      results ← U_{i=A} results(FEG_i)
20  End
```

FIG. 6.1: CGGA algorithm

6.3.3 Example

An example of the CGGA applied to a group of co-annotated genes is presented in TABLE 6.1. The data used in the example is from the experiment carried out by DeRisi (see section 2.1), where the diauxic shift process of the yeast, *Saccharomyces Cerevisiae*, was analyzed.

The ordered *g-rank* list was computed using the *modified t-statistic* obtained with the SAM program (see section 2.3). The data of the FEG, annotated "vacuolar protein catabolism", was obtained from the GO database (see section 2.2). This FEG contains 4 genes ($n = 4$) whose rows in the total *g-rank* list vary from 6 to 424.

In TABLE 6.1 we show the values of the parameters needed to determine the significant gene subsets within the FEG. We have highlighted the subset of genes: {1, 3}, from vacuolar protein catabolism FEG, found significantly co-expressed by CGGA.

CGGA tested for H_0 the (4*5)/2=10 FEG subsets computing their $p - value$ and comparing it to the α. For example, the $p - value$ corresponding to the subset {S000000490, S000001586} of rank 6 and 8 in *g-rank* is $2.63E^{-05}$ (cf. TABLE 6.2). This $p - value$ being lower than α, fixed at $6.88E^{-04}$ (cf. section 3.1), CGGA rejected H_0 and the group of genes {S000000490, S000001586} is then labelled statistically significant and co-expressed. We see that the subset with genes of rank 6 and 8 is very close and then co-expressed. On the other hand the genes of rank 69 and 424 are rather distant from their closer neighbours, i.e. the groups that contain them are not co-expressed significantly.

List *g-rank*	x	Gene ID (SGD)	GO Annotation	$r_{g(x)}$	$R_{g(x)}$
1				1	
2				2	
■				■	
6	1	S000000490	VACUOLAR PROTEIN CATABOLISM	6	1
7				7	
8	2	S000001586	VACUOLAR PROTEIN CATABOLISM	8	3
■				■	
69	3	S000000786	vacuolar protein catabolism	69	62
■				■	
424	4	S000006075	vacuolar protein catabolism	424	356
■				■	
N				N	

TABLE 6.1: CGGA Analysis for the FEG of genes annotated "vacuolar protein catabolism"

6.4 Results

In order to evaluate our method, we compared the results obtained by DeRisi [90], IGA [53] and CGGA[203]. The results obtained using CGGA for the over-expressed and under-expressed genes are presented in TABLE 6.2 and TABLE 6.3 respectively. As expected, almost all groups identified as significantly co-expressed by the DeRisi method have also been identified by the CGGA. The groups identified by CGGA and DeRisi are in **bold**, the ones identified only by CGGA are in *italics*, and the only group identified also by IGA is in SMALL CAPS.

In the case of over-expressed genes (TABLE 6.2), CGGA found seven of the nine groups obtained manually by DeRisi [90]. The two annotated groups "glycogen metabolism" and "glycogen synthase" have not been identified by CGGA because they are expressed only at the initial phase of the process. However CGGA identified eight other statistically significant and coherent groups. Only one of these eight other groups has also been identified by IGA and none of them by DeRisi.

For the case of under-expressed genes (TABLE 6.3), CGGA has found seven of the eight gene groups selected manually by DeRisi. As for over-expressed genes, the group annotated "ribosome biogenesis" was not identified by CGGA, because it was only expressed during the final phase of the process. CGGA have also identified seven other statistically significant and coherent groups which were not identified on the DeRisi analysis nor by IGA.

The three groups identified by DeRisi that CGGA did not identify, namely the over-expressed groups "glycogen metabolism" and "glycogen synthase", and the under-expressed group "ribosome biogenesis" share two important properties. First, they contain genes belonging to a heterogeneous structure, i.e genes that appertain to several functional groups. Second, these FEG are not expressed throughout the entire process but only during a specific phase. Detect these groups will only be possible by integrating information on the metabolic pathways ontologies such as: KEGG, EMP, CFG, etc.

Functionally Enriched GO Group	n genes	x Over-expressed genes	$pc-value$
proton-transporting ATP synthase com-plex	2	2	$4.38E^{-06}$
invasive growth (sensu Saccharomyces)	5	3	$6.13E^{-06}$
signal transduction during filamentous growth	2	2	$8.77E^{-06}$
respiratory chain complex II	4	4	$3.75E^{-05}$
succinate dehydrogenase activity	4	4	$3.75E^{-05}$
mitochondrial electron transport	4	4	$3.75E^{-05}$
aerobic respiration	36	10	$3.30E^{-05}$
tricarboxylic acid cycle	14	5	$5.09E^{-05}$
tricarboxylic acid cycle	14	5	$6.54E^{-05}$
gluconeogenesis	12	2	$9.64E^{-05}$
response to oxidative stress	10	3	$1.55E^{-06}$
filamentous growth	8	4	$9.06E^{-05}$
VACUOLAR PROTEIN CATABOLISM	4	2	$2.63E^{-05}$
respiratory chain complex IV	8	2	$4.05E^{-04}$
cytochrome-c oxidase activity	8	2	$4.05E^{-04}$

TABLE 6.2: Over-expressed FEGs obtained by CGGA with a $p-value = 6.88E^{-04}$

Functionally Enriched GO Group	n genes	x Under-Expressed genes	$pc-value$
chromatin modification	6	5	$2.35E^{-06}$
mitochondrial inner memb. prot. inser. complex	3	2	$3.60E^{-06}$
regulation of nitrogen utilization	4	2	$7.20E^{-06}$
acid phosphatase activity	4	2	$7.20E^{-06}$
histone acetylation	4	4	$7.95E^{-06}$
nucleolus	52	10	$3.41E^{-05}$
rRNA modification	10	3	$2.75E^{-05}$
transcription initiation from RNA poly. II prom.	14	3	$1.00E^{-05}$
mitochondrial matrix	15	3	$1.25E^{-05}$
processing of 20S pre-rRNA	11	2	$1.97E^{-04}$
ribosomal large subunit biogenesis	9	4	$3.17E^{-04}$
small nucleolar ribonucleoprotein complex	20	3	$2.52E^{-04}$
cytosolic large ribosomal subunit	69	13	$2.87E^{-04}$
ribosomal large subunit assembly and maint.	21	2	$2.52E^{-04}$

TABLE 6.3: Under-expressed FEGs obtained by CGGA with a $p-value = 6.88E^{-04}$

6.5 Discussion

The CGGA algorithm presented in this chapter makes it possible to automatically identify groups of significantly co-expressed and functionally enriched genes without any prior knowledge of the expected outcome. CGGA can be used as a fast and efficient tool for exploiting every source of biological annotation and different measure of gene variability.

We analyze CGGA concerning three important methodology parametres: biological source of information, profiles expression manipulation, significant gene groups selection (as stated in section 5.5).

The automated functional annotation provided by our algorithm reduces the complexity of microarray analysis results and enables the integration of any of the six sources of genomic information such as ontologies (explained in chapter 3).

Concerning gene expression measure handling, CGGA is rank-based, so they need an ordered gene-rank list, for testing the co-annotated and co-expressed gene groups. Although, this could be a dangerous simplification and can be a big loss in terms of gene expression profiles information, correctly used can serve positively as guide of the interpretation algorithm process and results. Indeed it represents a non mesurable loss of the original information contained in the raw gene expression measure.

In contrast to expression-based interpretation approaches such as [99], [128], [151], [228] and [175], CGGA analyze all the possible subsets of each FEG and does not depend on the availability of fixed lists of expressed genes. Thus, it can be used to increase the sensitivity of gene detection, especially when dealing with very noisy data sets. CGGA can even produce statistically significant results without any experimental replication. It does not need that all genes in a significant and co-expressed group change, so it is therefore robust against imperfect class assignments, which can be derived from public sources (wrong annotations in ontologies) or automated processes (naming errors, spelling mistakes, etc.).

6.6 Conclusion and Outlook

CGGA can be used as a tool for platform-independent validation of a microarray experiment and its comparison with the huge number of existing experimental databases and the documentation databases. Experimental results show the interest of our approach and make it possible to identify relevant information on the analyzed biological processes. In order to identify heterogeneous groups of genes expressed only in certain phases of the process, we plan to integrate the information concerning the metabolic pathways ontologies for future work.

Chapter 7

GENMINER: Gene-Integrated Analysis by Association Rules Discovery

In this chapter, we completely develop our GENMINER algorithm: gene-integrated analysis of gene expression profiles and gene annotations by association rules discovery (presented in chapter 5). This co-clustering interpretation approach integrates at once gene annotations and gene expression profiles to discover intrinsic associations among both data sources based on frequent patterns. Gene expression data profiles are taken from cleaned microarray measures and gene annotations are obtained from any of the sources of biological information presented in chapter 3.

GENMINER algorithm is a smart adaptation of the Association Rules Discovery mining technique that fulfills the requirements for data obtained from gene expression technologies. It takes advantage of the Close[229] mining algorithm to generate low support, high confidence and non-redundant rules in an efficient way.

GENMINER method facilitates the integration of any of the seven sources of biological information (presented in chapter 3). It can even integrate easily qualitative variable information of biological conditions as age, sex, state etc.

In contrast to current clustering approaches that group genes whose expression levels are similar across all conditions, GENMINER can find subsets of genes that participate in any particular cellular process, even if the cellular process takes place only in a subset of the biological samples (this technique was also called bi-clustering in chapter 5). Actually, bi-clustering techniques could detect the genes that are conditionally co-expressed and co-annotated within different sets of genes, reflecting the different biological roles that a gene can play in the cell.

We have validated the proposed methodology analyzing two microarray data sets, the DeRisi data set [90] and the Eisen data set [107]. The gene annotations were obtained from different sources of biological information: semantic sources as Gene Ontology (all-three ontologies: BP, MF, CC), gene-protein specific databases as: KEGG and BioGRID, the nomenclature database SGD, the bibliographic database PubMed/Medline and transcriptional regulators information reported in [176]. Automatically extracted associations obtained by GENMINER revealed significant co-annotated and co-expressed gene patterns, signifying important biological relationships between the genes and their attributes. Several of these relationships are supported by recent biological literature.

This chapter is organized in the following way: it starts with a brief introduction of the interpretation challenges in gene expression technologies, it states the main data interpretation target, and it presents the GENMINER algorithm (section 1). Then, it gives a global view of association rules basics and it explains two rule extraction algorithms Apriori and Close (section 2 and 3 respectively).In section 4, it describes the GENMINER algorithm concepts and implementation aspects. Subsequently, it validates and evaluates GENMINER method by analyzing two spotted cDNA chips data sets: DeRisi and Eisen (section 5 and 6 respectively). The last two sections give a discussion and an outlook for future research.

7.1 Introduction

Gene expression technologies are powerful methods for studying biological processes through a transcriptional point of view. Since many years these technologies have produced vast amounts of data by measuring simultaneously the expression levels of thousands of genes under hundreds of biological conditions. One of the great potentials of these technologies is that the generated data contain hidden information about the biological processes that govern cell behavior. Nowadays, one of the main goals of these technologies is to discover this hidden information to achieve biological knowledge. In other words, we want to interpret gene expression technology results via integration of gene expression profiles with corresponding biological gene annotations extracted from biological databases (presented in chapter 3). Consequently, the key task in the interpretation step is to detect the present co-expressed (sharing similar expression pattern) and co-annotated (sharing the same properties such as function, regulatory mechanism, etc.) gene groups

In order to process the interpretation step in an automatic or semi-automatic way, the bioinformatics community faces an ever-increasing volume of sources of biological information on gene annotations that are:

- Minimal Microarray Information (genes, biological conditions, gender, age etc.).
- Molecular databases (GenBank, Embl, Unigene, etc.).
- Semantic sources as thesaurus, ontologies, taxonomies or semantic networks (UMLS, GO, taxonomy, etc.).
- Gene expression databases (GEO, Arrayexpress, Microarray database, etc.).
- Bibliographic databases (Medline, Biosis, etc.).
- Gene/protein related specific sources (ONIM, KEGG, etc.).

A variety of approaches recently reported have already dealt with the interpretation problem. We have classified these in three different axes [201]: *expression-based approaches* as FunSpec [258], OntoExpress [99], Quality Tool [128], EASE [151], THEA [228], Graph Theoretic Modeling [175] and GENERATOR [234], *knowledge-based approaches as* GSEA [215], iGA [53], PAGE [167] and CGGA [203] and *co-clustering approaches* as Co-Cluster [139], Bicluster [186], ARD [61]. These approaches were fully explained in chapter 5.

The most currently used interpretation axis is the *expression-based* axis which gives more importance or weight to gene expression profiles, than the other two interpretation approaches. However, it presents many well-known drawbacks:

1. Most of these approaches cluster genes by similarity expression profile levels across all conditions. Nevertheless, gene groups involved in one biological process might be only co-expressed in a small subset of biological conditions.[9]
2. Many genes may be conditionally co-expressed with different sets of genes, this can reflect the different biological roles that genes can play in the cell. Most of the commonly used clustering methods group only genes into single clusters, masking more complex relationships between different sets of conditionally regulated genes [122].
3. There is a lack of generality in the assumption: "Genes sharing similar expression profiles also share similar biological properties". Actually, simultaneously expressed genes may not always share the same function or regulatory mechanism. Even when similar expression patterns are related to similar biological roles, discovering these biological connections among co-expressed genes is not a trivial task and requires a lot of additional work [279].

Knowledge-based approaches give more importance to biological knowledge. Nevertheless, all sources of biological information fix many integration constraints: the database format or structure, the weak quantity of annotated genes, the availability of data, the maintainance of up-to-date and well revised annotations for instance. Consequently, the *knowledge-based* interpretation results can be poor or somewhat quite small or limitated in relation to the whole studied biological process.

Co-clustering approaches represent the best deal in terms of integration, giving the "same" weight to expression profiles and biological knowledge. However, they can be complex as Bi-cluster [186] and Co-cluster [139], algorithms, they can not integrate in a simple manner several sources of biological information.

To overcome all the drawbacks mentioned above, we propose the use of the Association Rules Discovery (ARD) technique. ARD is an unsupervised data-mining technique used to discover associations among subsets of items (gene expression profiles and/or gene annotations) from very large transaction databases (gene expression profiles matrix and gene annotation matrix). The ARD technique identifies groups of elements that frequently co-occur in a transaction database, establishing relationships among them of the form of an association rule: $A \Longrightarrow B$, which means when A occurs it is likely that $B.occurs$. The ARD technique has the following advantages:

1. ARD clusters the genes by frequency in patterns of expression profiles and annotations, regardless of the position in the transaction matrix. So, it represents a *bi-clustering* technique which can generate rules containing genes that are co-expressed only in a subset of the biological conditions.

2. Any gene can be assigned to any number of rules as long as its expression fulfills the assignation criteria. This means that a gene involved in many co-expressed groups will appear in each and every one of those groups, without limitation.
3. ARD is orientated ($if\ A\ then\ B$), describing the direction of a relationship. Thus, any type of relationship between expression measures and gene annotations can be discovered. For example, a gene encoding a transcription factor should appear in the left portion of the rule and its over- or under-expressed profile measures in the right part of the rule.
4. ARD favors the integration of all six biological sources of information cited before. All gene attributes (annotations and profiles) can be added to the transaction matrix by indicating the presence or the absence of the attribute (for gene annotations) or by discretizing the gene expression profiles.

As mentioned in the introduction, ARD has been previously used to mine gene expression data sets in order to discover associations among subsets of genes based only on their gene expression profiles [22], [81], [170], [310], [224], [263] and [124].

Recently, an ARD algorithm, Carmona et al. 2006 [61], has been used to integrate gene expression profiles and gene annotations with rules of the form: $Annotation \Longrightarrow [\downarrow] C1, [\uparrow] C2$. This means when a set of genes annotated with the characteristic *Annotation* occurs, the set of genes is likely to be under-expressed in biological condition $[\downarrow] C1$ and over-expressed in biological condition $[\uparrow] C2$. Thus, the gene annotations are always in the antecedent of the rule, and the gene expression measures are always in the consequent of the rule.

In contrast to past approaches, GENMINER is applied to identify sets of gene expression measures and gene annotations that frequently co-occur in a data mining context, $\mathcal{DG}$, establishing relationships among them of the form:

$$\begin{aligned} [\downarrow] C1 &\Longrightarrow [\uparrow] C2, [C], [D] \\ C &\Longrightarrow [\downarrow] C1 \end{aligned} \tag{7.1}$$

The first case states that when a set of under-expressed genes in biological condition $C1$ occurs, these genes are likely to be over-expressed in biological condition $C2$, and they are annotated by characteristics C and D. The second case means that a set of genes annotated with the characteristic C were under-expressed in biological condition $C1$.

In the past years, the ARD technique has been used in gene expression technologies in order to find the frequent gene patterns among a subset of biological conditions: [22], [81], [170], [310], [224], [263] and [124]. The association rules generated by these approaches are of the following form: $[\downarrow]$ gene $g1 \Longrightarrow [\uparrow]$ gene $g2$, $[\downarrow]$ gene $g3$, meaning that in a significant number of conditions when gene $g1$ is under-expresssed, it is likely to observe an over-expression of gene $g2$ and under-expression of gene $g3$. They have been successfully applied in gene expression profile classification, avoiding some drawbacks of standard clustering techniques (see section 2.3). However, these algorithms concern exclusively gene expression measures without taking into account biological knowledge, therefore.leaving to the expert the task of discovering and interpreting the biological similarities hidden within gene groups. Recently, ARD

has been used to integrate gene expression profiles and semantic databases as GO and KEGG with rules of the form: $Annotation \Longrightarrow [\downarrow] C1$, $[\uparrow] C2$ which means that a group of genes annotated by "Annotation" is likely to be under-expressed in biological condition $C1$ and over-expressed in condition $C2$. This approach [61] is an ingenious attempt for using ARD to integrate gene profile measures and annotations, but it presents several weaknesses:

1. The utilization of the Apriori algorithm [4] as ARD method is time-consuming (in the case of correlated data), so it limits the number of possible gene-attributes (contained in the biological sources of annotations), and it generates a lot of redundant rules.
2. It uses a redundant filter that is not minimal against inclusion
3. It uses only annotations in the left part of the rule and gene expression profiles in the right part of the rule. Indeed, often the experts search associations with annotations in the right part of the rule and gene expression profiles in the left part of the rule, or even mixing this informations either in one side or the other of the rule.
4. It uses the discretization method for expression profile measures in three intervals, $(-\infty, -1] = under - expressed$, $(-1, +1) = no - expressed$ and $[1, +\infty) = over - expressed$, applying the fold change cut-off method. This is a dangerous simplification that presents many drawbacks (as explained in section 8.3).

In order to avoid these weaknesses and to exploit optimally the ARD capacities, we have developed the GENMINER approach. GENMINER is capable of integrating gene annotations and gene expression profile data to discover intrinsic associations among both data sources based on co-ocurrence pattern extraction. This data mining method can generate rules containing all existent associations among the transaction matrix elements: gene expression profiles and annotations. Gene annotations can be integrated from any of the six sources of biological information cited before. Concerning gene expression profiles, we propose several discretization scenarios including the use of an innovative discretization method for gene expression data, NORDI (explained in section 8.3.2.1). In this manner, GENMINER can construct rules of the form: R_1 : *Plays role in fermentation* (Pubmed annotation), *Anaerobic respiration* (GO annotation), $[\uparrow] C2 \Longrightarrow [\uparrow] C1$, $[\uparrow] C3$, $[\downarrow] C7$, *Digestion* (UMLS term)}. This means that a set of yeast genes which plays a role in fermentation, anaerobic respiration, and is over-expressed in condition 2 is likely to participate in the digestion process, to be over-expressed in biological conditions 1 and 3, and to be under-expressed in condition 7.

The GENMINER algorithm fulfills the requirements for data obtained from gene expression technologies. It takes advantage of the Close [229] association rules mining algorithm that generates low support, high confidence and non-redundant rules in an efficient way. The Close algorithm allows the use of all available matrix information without limiting constraints for rules extraction. Concerning the calculation, it is more efficient than the often used Apriori algorithm [4] when the items are dependent on each other (which is the case of gene expression data) because it reduces the problem of finding frequent itemsets to finding frequent Closed itemsets [239] and [229]. We can translate this to a non-negligible time reduction calculation

produced by a reduction in the rule research space, which directly enhances the expert's data results interpretation.

One of the major limitations of ARD is the large amount of rules generated, which easily becomes a problem in many applications. This fact has been studied before. In the context of gene expression technologies, Creighton et al. imposed constraints on the size of the rules [81] while Tuzhilin [310] proposed several post-processing operators for selecting and exploring interesting rules.

The GENMINER algorithm takes profit of Close rule mining algorithm which extracts a set of rules that are minimal against inclusion. Close algorithm already integrates a *pruning* method based on the closed itemset lattice (Wille's concept lattice [319]) for rule extraction. The intuitive idea behind this method, can be seen as a redundant filter which takes the less information at the left side of the rule and the most at the right side of the rule. Thus, this filter will choose the minimal rule, that is the one which compacts best the important rule information.

7.2 Association Rules Basics

ARD is an unsupervised data-mining technique oriented towards finding associations or correlation relationships among items from very large transaction data sets. This method extracts sets of items that frequently occur together in the same transaction, and then formulate rules that characterize these relationships. Here, we will introduce some basic semantics currently used in ARD and the association rules extraction process steps.

7.2.1 Association rules semantics

Let $\mathcal{I} = \{i_1, i_2, ..., i_m\}$ be a set of m literals called *items*. Let the *transaction database* $\mathcal{T} = \{t_1, t_2, ..., t_n\}$ be a set of n transactions* (also called tuples or objects), each t_i consisting of a subset of items $I \subseteq \mathcal{I}$ and associated with a unique identifier called its OID. I is called a *k-itemset*, where k is the number of items in I. A transaction $t_i \in \mathcal{T}$ is said to contain an itemset I if $I \subseteq t_i$. Let T be a subset of the transactional database $\mathcal{T}$, that is $T \subseteq \mathcal{T}$. Intuitively we can say that:

- Transaction databases, $\mathcal{T}$, are often tables or matrices organized in horizontal rows and vertical columns. A table is the lay term for relation in databases, i. e. a subset of the Cartesian product of a set of attribute domains.
- T is any subset of objects or transactions, where $T \subseteq \mathcal{T}$
- I is any subset of items, where $I \subseteq \mathcal{I}$.
- Item i is a triple $\{A, Cop, V\}$ where: A is an attribute, Cop is a comparison operator $\{<, \leq, =, \geq, >\}$ and V is a value.
- An attribute is an inherent property of an object or an entity in a database or associated with that entity for database purposes.

Thus, the association rules discovery (ARD) technique identifies itemsets that frequently co-occur in a transaction database, establishing relationships among them of the form of an association rule: $I_1 \Longrightarrow I_2$, which means when I_1 occurs, I_2 is likely to occur too. The left side of the rule is the antecedent and the right side the consequent. Given the association rule $I_1 \Longrightarrow I_2$, there are two basic measures that define the quality of the rule: support and confidence.

The support of the rule $I_1 \Longrightarrow I_2$ in a transaction database $\mathcal{T}$ is the proportion of transactions (lines) of the table $\mathcal{T}$. In other words $supp(I_1 \Longrightarrow I_2)$ is the percentage of transactions T in $\mathcal{T}$ where I_1 and I_2 appear together:

$$supp(I_1 \Longrightarrow I_2) = \frac{|\{T \mid (I_1 \cup I_2) \subseteq T, T \in \mathcal{T}\}|}{|\mathcal{T}|}.$$

The confidence of the rule $I_1 \Longrightarrow I_2$ in the transaction database $\mathcal{T}$ is the proportion of transactions (lines) of the table $\mathcal{T}$ that have I_2 taken from those that contain I_1. In other words the $conf(I_1 \Longrightarrow I_2)$ is the percentage of transactions T in $\mathcal{T}$ that contain I_1 also contain I_2, i.e.,

$$conf(I_1 \Longrightarrow I_2) = \frac{supp(I_1 \cup I_2)}{supp(I_1)} = \frac{|\{T \mid (I_1 \cup I_2) \subseteq T, T \in \mathcal{T}\}|}{|\{T \mid I_1 \subseteq T, T \in \mathcal{T}\}|}.$$

Support and confidence are the most common quality measures related to a rule. However, sometimes both of these measures are high, indicating a rule which could be good, but in reality has an association that is not useful. This is the case which the elements of the consequent are very frequent in the transaction database [54]. Therefore, associations among uncorrelated elements can be generated using the support-confidence framework [157].

Thus, a correlation measure between the antecedent and the consequent of the rule is needed to assess the quality of the rule. The $lift$ or improvement measure is the most used correlation measure in association rules and it is defined as follows:

The $lift$ or improvement of the rule $I_1 \Longrightarrow I_2$ can be used as an independence measure between antecedent I_1 and consequent I_2 of the rule, that is, the percentage of transactions T in $\mathcal{T}$ where I_1 and I_2 appear together divided by the percentage of transactions T in $\mathcal{T}$ containing I_1 or I_2, i.e.,

$$lift(I_1 \Longrightarrow I_2) = \frac{supp(I_1 \cup I_2)}{supp(I_1)supp(I_2)}.$$

Any rule with improvement equal to 1 means independence, between consequent and antecedent of the rule. On the contrary, if the improvement is greater than one the consequent and antecedent of the rule are related, and the antecedent may predict the consequent.

Let us take an example where n is the number of transactions in the studied table $\mathcal{T}$ and $n(I)$ is the counting function of the number of transactions of the itemset I in the table. We want to calculate the support, confidence and lift of the rule $A \Longrightarrow B$ where $A \subset \mathcal{I}$ and $B \subset \mathcal{I}$. Thus, we can calculate $n(A)$ as the number of transactions containing A, $n(B)$ is the number of lines that contain B and $n(AB)$ the number of lines which contains both itemsets A

and B. Using the equations above for support 7.2.1, confidence 7.2.1 and lift 7.2.1 we obtain: $supp(A \Longrightarrow B) = \frac{n(AB)}{n}$, $conf(A \Longrightarrow B) = \frac{n(AB)}{n(A)}$ and $lift(A \Longrightarrow B) = \frac{n(AB)}{n(A)n(B)}$ respectively.

7.2.2 Association rules extraction process

The association rules extraction process consists of four steps: data selection and pretreatment, *frequent itemsets* extraction, association rules generation and interpretation of extracted rules (see FIG. 7.1).

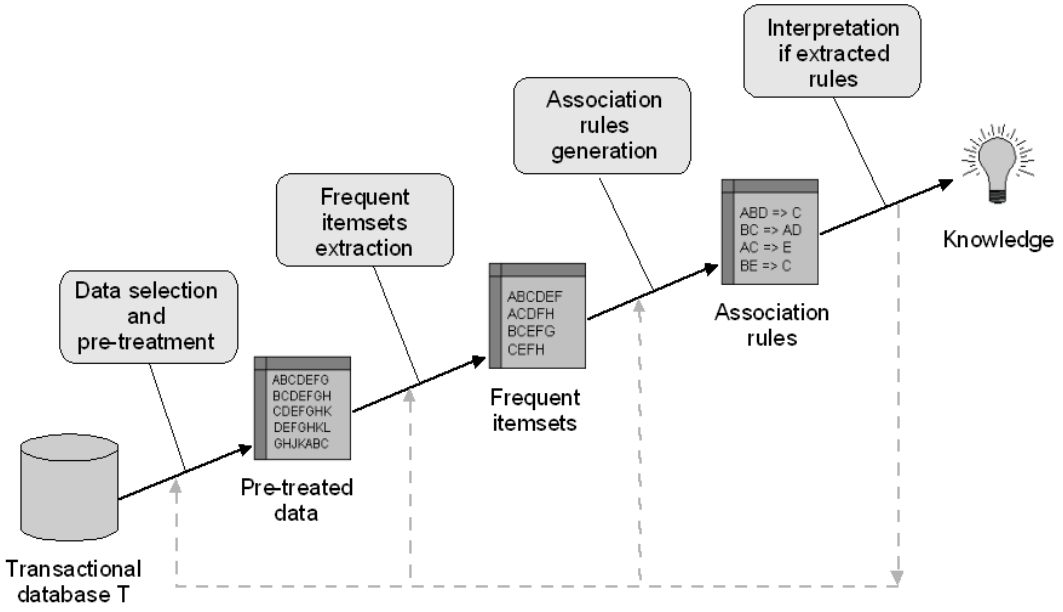

FIG. 7.1: Association rules extraction process

The data selection involves choosing useful attributes to optimize the association rule extraction. The data pretreatment issue concerns the discretization of the quantitative continuous variables into discrete variables and the determination of categorical classes for either qualitative or quantitative variables.

The frequent itemsets extraction has to find all frequent itemsets contained in the transactional database T, i.e. itemsets with support greater or equal to a given minimum support.

The association rules generation consists of producing all association rules for each frequent itemset found with confidence greater or equal to a given minimum confidence.

The interpretation of extracted rules consists of analyzing the pertinence of all extracted rules in terms of our studied problem.

In the next section, we develop the problem of mining association rules in a transactional database T, i.e. the frequent itemsets extraction and association rules generation steps. The

sectiojn is organized as follows: First, we describe the general framework for association rules mining algorithms. Then, we explain the most current approach for mining association rules: the Apriori algorithm. Finally, we explain the Close algorithm based in the *frequent Closed itemset extraction* methodology.

7.3 Association Rules Extraction

The first algorithm for mining association rules was the Apriori algorithm, presented by Agrawal in 1993 [4]. Coincidentally, a similar algorithm was developed by Mannila's [194] some months later, but in association rules experts field the credit is given to the first, that is the *Apriori* algorithm.

In this section, we explain the Agrawal's *Apriori* algorithm, which has become the reference methodology and theoretic framework for the problem of mining association rules [4]. Then, we present the Close methodology [229], which optimizes the ARD performances and the size of the results. Our GENMINER algorithm is based on the Close method for extracting association rules.

7.3.1 Framework of Agrawal's association rules extraction

Agrawal's framework [4] for the problem of extracting association rules was introduced in association rules semantic section 7.2.1. Here, we focus in the Agrawal's framework for mining association rules.

The task of mining association rules in a transactions database $\mathcal{T}$ is traditionally defined as follows: given user-defined thresholds for the permissible minimum support and confidence, find all association rules that hold with more than the given *minsupp* and *minconf*. *Minsupp* and *minconf* are two parameters fixed beforehand by the analyist, they represent the minimal support and confidence that the analyst is disposed to accept for rule extraction. Thus, the problem can be broken into two sub-problems [4] (as seen in FIG. 7.1):

1. Extracting all frequent itemsets in T, i.e. itemsets with support greater or equal to *minsupp*.
2. Generating association rules from these frequent itemsets with confidence above *minconf*.

The second problem can be solved in *main memory*[29] in a straightforward manner once all frequent itemsets and their support are known. Hence, the problem of mining association rules is reduced to the problem of finding frequent itemsets. Agrawal's Apriori algorithm was the first methodology that formalizes the frequent itemsets extraction and association rule generation steps. Here, we present Agrawal's formalization concerning association rules discovery.

[29] Main memory is computer memory that is accessible to the central processing unit of a computer without the use of computer's input/output channels. This kind of memory is used to store data that is likely to be in active use.

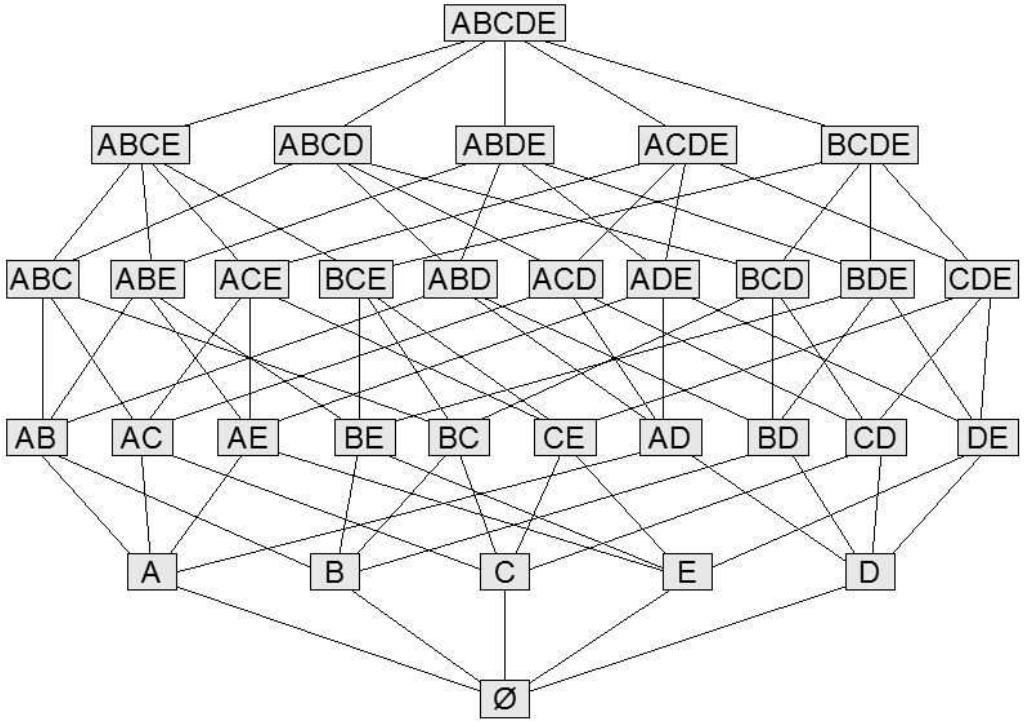

FIG. 7.2: Itemset lattice associated to a data mining context $\mathcal{D}$

A data mining context is defined as a triple $\mathcal{D}$ =$(\mathcal{T}, \mathcal{I}, \mathcal{R})$ where $\mathcal{T}$ and $\mathcal{I}$ are finite sets of objects and items respectively.$\mathcal{R} \subseteq \mathcal{T} \times \mathcal{I}$ is a binary relation between objects and items. Each couple $(t, i) \in \mathcal{R}$ denotes the fact that the transaction $t \in \mathcal{T}$ is related to the item $i \in \mathcal{I}$

Object	Items			
1	A	C	D	
2	B	C	E	
3	A	B	C	E
4	B	E		
5	A	B	C	E
6	B	C	E	

TABLE 7.1: Association rules extraction context $\mathcal{D}$

Given the data mining context $\mathcal{D}(\mathcal{T}, \mathcal{I}, \mathcal{R})$, discovering frequent itemsets is not a trivial problem, because the number of possible frequent itemsets is exponential, as huge as the size of the set of items of the context $\mathcal{D}$, i.e. $2^{\|\mathcal{I}\|}$ (see FIG. 7.2).

Having a data mining context $\mathcal{D} = (\mathcal{T}, \mathcal{I}, \mathcal{R})$ and the minimal threshold, the set of frequent itemsets F in $\mathcal{D}$ is:

$$F = \{I \subseteq \mathcal{I} : | : I \neq \emptyset \wedge supp(I) \geq minsupp\}$$

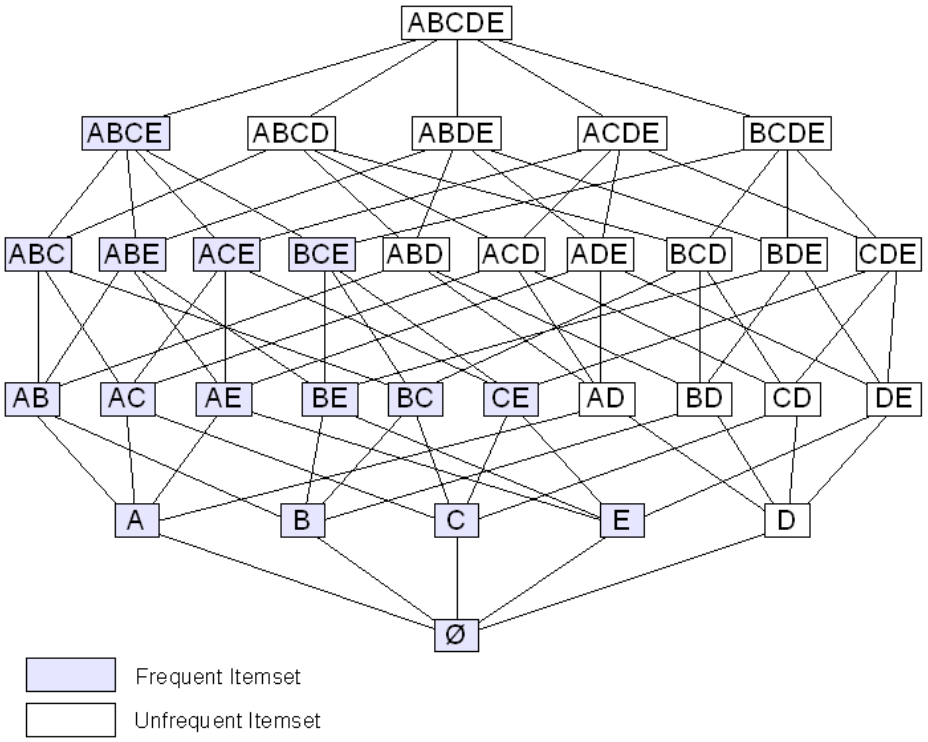

FIG. 7.3: Frequent itemset lattice F, containing the frequent itemsets associated to a data mining context $\mathcal{D}$ with *minsupp=2/6*.

Therefore, if the set of items $\mathcal{I}$ is of size m, then the number of possible frequent itemsets is 2^m. Taking all this possible itemsets we can build the itemset lattice of $\mathcal{I}$, having height of $m+1$. For example, the itemset lattice of the set of items $\mathcal{I}$ in a context $\mathcal{D}$ (TABLE 7.1) contains $2^5 = 32$ itemsets and the itemset lattice has a height of six. (see FIG. 7.2). If we take a *minsupp=2/6* and we apply equation 7.2, then the frequent itemset lattice, F, contains $2^4 = 16$ frequent itemsets (see FIG. 7.3).

Discovering frequent itemsets is the most time-consuming stage in association rules extraction because of the exponential size of the research item space and the need of continuous scanning of the context $\mathcal{D}$. A trivial method would be testing the support of each of the itemsets in the lattice, but this is impracticable when the number of items is huge.

Let F be the lattice of frequent itemsets in a data mining context $\mathcal{D}$ with a minimal support threshold: *minsupp*, the association rules generation for a minimal confidence threshold *minconf* is an exponential problem of size $\|F\|$ Given *minsupp* and *minconf,* the set of valid $\mathcal{AR}$ in $\mathcal{D}$ is:

$$\mathcal{AR} = \{r : I_2 \Rightarrow (I_1 - I_2) \quad | \quad I_1, I_2 \in F \wedge I_2 \subset I_1 \wedge \frac{supp(I_1)}{supp(I_2)} \geq minconf\}$$

In practice, the association rules generation is done straightforward without taking into accout the extraction context $\mathcal{D}$, and the execution time-cost of this phase is relatively low in comparison with the frequent itemsets extraction cost. Thus, for each frequent itemset I_1 in

F, all subsets I_2 of I_1 are determined and the value of $supp(I_1)/supp(I_2)$ is calculated. If this value is equal or higher than the *minconf* threshold, then the association rule: $I_2 \Rightarrow (I_1 - I_2)$ is generated.

7.3.2 Apriori frequent itemsets extraction algorithm

The first algorithm for finding frequent itemsets extraction was SETM [4] proposed in 1993. The authors have introduced the idea of searching the frequent itemsets by levels instead one by one. Agrawal [4] and Mannila [194] have both presented similar algorithms, but in ARD domain the credit is done to the first, that is a priori algorithm. Nowadays, Agrawal's Apriori algorithm is used in many ARD applications, and it also has inspired some others well known ARD algorithms as Partition [226], Sampling [267] and DIC [55].

Here, we briefly explain the *Apriori* algorithm, that is, the reference method in association rules discovery At the beginning, frequent itemsets are computed iteratively, in ascending size order (see FIG. 7.2). The process takes k iterations, where k is the size of the largest frequent itemsets. For each iteration $i < k$, the database $\mathcal{D}$ is scanned at once and all frequent itemsets of size i are computed. The first iteration computes the set I_1 of frequent 1-itemsets. A subsequent iteration i consists of two phases. First, a set of candidates or potentially frequent i-itemsets, C_k, is created by joining the frequent $(i-1)$ itemsets in I_{i-1} found in the previous iteration. Then, the database is scanned for determining the support candidates and the frequent $i-$itemsets are extracted from the candidates. This process is repeated until no more candidate can be determined. (see FIG. 7.3)

Apriori and several Apriori inspired algorithms as the ones cited before are specially robust for non-correlated data. The data issued from gene expression technology is highly correlated data (as explained in chapter 1 and 2), so the Apriori algorithm would be time consuming and inefficient for most of the microarray data cases. The principal association rule extraction drawbacks using Apriori and Apriori inspired algorithms are:

- Execution time problems
  - Execution times of several hours mostly (and sometimes several days).
  - Data sets are large (cannot fit in main memory).
  - Data set must be scanned (entirely read) several times during the process.
- Relevance of extracted association rules
  - There are several tens of thousands of extracted rules (sometimes millions).
  - Among these rules many are redundant (i.e. they represent the same information).
- Correlated data constitute a challenge for extracting association rules

In order to deal with each one of these drawbacks we propose the use of the Close algorithm [229] based on frequent Closed itemset extraction using a *Closed itemset lattice*, rather than the Apriori itemset lattice. The Close algorithm performs well specially for highly correlated data, it reduces considerably the execution times, and it proposes a pruning

methodology to generate minimal non-redundant rules. The following section presents the basics of this algorithm.

7.3.3 Close algorithm

Pasquier [229] proposes a new approach for association rule extraction based on the closure of the Galois connection [121]. The connection closure is used to define a condensed representation for association rules. This representation is characterized by Closed itemsets, which build the Closed itemset lattice that contains the frequent Closed itemsets. In this algorithm, it is proven that the set of frequent Closed itemsets constitutes a generator set for the concerned frequent itemsets. This representation is a basis, i.e., a generating set for all association rules, their supports and their confidences, and all of them can be retrieved without accessing the data. In the next section, we present a brief explanation of the Close algorithm emphasizing the obtained condensed representation for association rules.

Here, we develop the problem of extracting frequent Closed itemsets and generators in a data mining context $\mathcal{D}$ with the Close algorithm. This section is organized as follows: First, we give the general framework for the Close algorithm. Then, we explain the *frequent Closed itemset extraction* methodology of Close Then, we explain the fundamentals of *min-max rules* generation. Finally, we show the interest in using the Close generators in the case of microarray data.

Framework of Close association rules extraction methodology

The Galois connection of a finite binary relation [121] is a couple of functions (ϕ, ψ). ϕ associates the items related to all transactions $t \in T$ with a set of transactions $T \subseteq \mathcal{T}$. ψ associates the transactions related to all items $i \in I$.with an itemset $I \subseteq \mathcal{I}$ When a transaction t is related to all items $i \in I$, we say that t *contains* I. We denote *minsupp* and *minconf as* the minimal support and confidence thresholds.

Definition 1 (Frequent itemsets) *The support of an itemset I is the proportion of objects in the data mining context containing I: $supp(I) := |\psi(I)| \,/\, |\mathcal{T}|$. I is a frequent itemset if $supp(I) \geq minsupp$.*

Definition 2 (Frequent Itemsets Equivalence Classes) *Frequent Itemsets Equivalence class, EC, is a set of frequent itemsets, that is $I_i \subseteq \mathcal{I}$, where the support of each one of the itemsets belonging to the class is the same, i.e. $supp(I_i) := k$ for all i where $I_i \subseteq EC$.*

Definition 3 (Association rules) *An association rule AR is an implication between two frequent itemsets $I_1, I_2 \subseteq \mathcal{I}$ with the form $I_1 \rightarrow (I_2 \setminus I_1)$ where $I_1 \subset I_2$. The support and confidence of AR are defined by: $supp(AR) = supp(I_2)$, $conf\,(AR) := supp(I_2) \,/\, supp(I_1)$.*

The closure operator $\gamma = \phi \circ \psi$ associates with an itemset I the maximal set of items common to all the transactions containing I: The closure of an itemset is equal to the in-

tersection of all the transactions containing it. Using this closure operator, we define the frequent Closed itemsets.

Definition 4 (Frequent closed itemsets) *A frequent itemset $I \subseteq \mathcal{I}$ is a frequent closed itemset if $\gamma(I) := I$. The minimal closed itemset containing an itemset I is its closure $\gamma(I)$.*

The set of frequent Closed itemsets and their supports is a minimal non-redundant generating set for all frequent itemsets and their supports and thus for all association rules, their supports and their confidences. This theorem relies on the properties that the support of a frequent itemset is equal to the support of its closure and that maximal frequent itemsets are maximal frequent Closed itemsets [230]. In order to improve the efficiency of frequent Closed itemset extraction, the Close algorithms compute generators of frequent Closed itemsets.

Definition 5 (Generators) *An itemset $g \subseteq \mathcal{I}$ is a generator of a closed itemset I if $\gamma(g) := I$ and $\nexists g' \subseteq \mathcal{I}$ with $g' \subset g$ such that $\gamma(g') := I$. A generator of cardinality k is a k-generator.*

Generators are the minimal itemsets to consider for discovering frequent Closed itemsets, by computing their closures. Close performs a breadth-first search for generators in a levelwise manner.

Frequent closed itemsets and generators extraction

The Close algorithm is an iterative algorithm for extracting generators and frequent Closed itemsets in a levelwise manner (considering all itemsets of a level in the itemset lattice at the same time). During an iteration k, a list of candidate $k - generators$ is considered; their closures and their supports are computed from the data set and infrequent generators are discarded. Frequent generators are then used to construct candidate $(k+1) - generators$. The closures of frequent generators are the frequent Closed itemsets and the support of a generator is also the support of its closure.

During the k^{th} iteration, a set FC_k is considered. Each element of this set consists of three information: a $k - generator$, its closure and their support. The algorithm first initializes the candidate $1 - generators$ in FC_1 with the list of $1 - itemsets$ and then carries out some iterations. During each iteration k the following procedure takes place:

1. Closures of all candidate $k - generators$ and their supports are computed: The number of objects containing a generator determines its support, and their intersection generates its closure. Each object is considered once, and this phase requires only one scan of the data set.
2. Infrequent $k - generators$, i.e., generators with support lower than *minsupp*, are removed from FC_k.
3. The set of candidate $(k+1) - generators$ is constructed by joining the frequent $k - generators$ in FC_k as follows.

(a) Two $k-generators$ in FC_k that have the same first $k-1$ items are joined to create a candidate $(k+1)-generator$. For instance, the $3-generators$ {ABC} and {ABD} will be joined in order to create the candidate 4-generator {ABCD}.

(b) Candidate $(k+1)-generators$ that are infrequent or non-minimal are removed. One of the k-subsets of such a generator is either infrequent or non-minimal and thus does not belong to the set of frequent k-generators in FC_k.

(c) The third phase removes $(k+1)-generators$ whose closures were already computed. Such a generator g is easily identified as it is included in the closure of a frequent $k-generator$ g' in FC_k: We have $g' \subset g \subseteq \gamma(g')$.

The algorithm stops when no new candidate generator can be created. Then, each set FC_k stores the frequent k-generators, their closures and their supports.

We illustrate the Close algorithm, taking the data mining context $\mathcal{D}$ example shown in TABLE 7.1 for $minsupp = 2/6$. The corresponding itemset lattice of the set of items $\mathcal{I}$ in a context $\mathcal{D}$ (TABLE 7.1) is shown in FIG. 7.2.

The set FC_1 is initialized with the list of all 1-itemsets.The algorithm computes supports and closures of the $1-generators$ in FC_1 and infrequent ones are discarded, which is the case for generator $\{D\}$ with $minsupp = 1/6$. Then, joining the frequent generators in FC_1, six new candidate $2-generators$ are created: $\{AB\}$, $\{AC\}$, $\{AE\}$, $\{BC\}$, $\{BE\}$ and $\{CE\}$ in FC_1.The $2-generators$ $\{AC\}$ and $\{BE\}$ are removed form FC_2 because we have $\{AC\} \subseteq \gamma(\{A\})$ and $\{BE\} \subseteq \gamma(\{B\})$. The algorithm determines supports and closures of the remaining $2-generators$ in FC_2 and suppresses infrequent ones. Then, the candidate $3-generator$ $\{ABE\}$ is created by joining the frequent generators in FC_2 but is removed because the $2-generator$ $\{BE\} \subset \{ABE\}$ is not in FC_2 and the algorithm stops. This process is shown in FIG. 7.4.

Scan D $\longrightarrow$

FC_1

Generator	Closed itemset	Supp
{A}	{AC}	3/6
{B}	{BE}	5/6
{C}	{C}	5/6
{D}	{ACD}	1/6
{E}	{BE}	5/6

Pruning infrequent itemsets $\longrightarrow$

FC_1

Generator	Closed itemset	Supp
{A}	{AC}	3/6
{B}	{BE}	5/6
{C}	{C}	5/6
{E}	{BE}	5/6

Scan D $\longrightarrow$

FC_2

Generator	Closed itemset	Supp
{AB}	{ABCE}	2/6
{AE}	{ABCE}	2/6
{BC}	{BCE}	4/6
{CE}	{BCE}	4/6

Pruning infrequent itemsets $\longrightarrow$

FC_2

Generator	Closed itemset	Supp
{AB}	{ABCE}	2/6
{AE}	{ABCE}	2/6
{BC}	{BCE}	4/6
{CE}	{BCE}	4/6

FIG. 7.4: Extracting frequent closed itemsets and generators in the context $\mathcal{D}$ with CLOSE.

The reduced research space itemset lattice after determining the frequent itemsets with Close, including the Closed itemsets, the generators and the equivalence classes in context $\mathcal{D}$ is illustrated in FIG. 7.5.

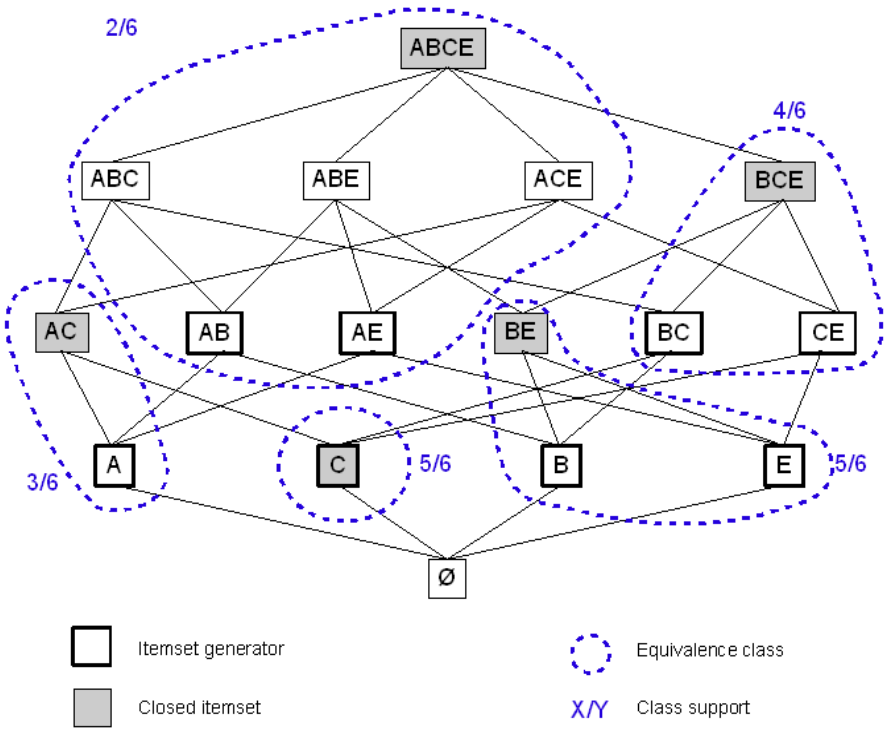

FIG. 7.5: Frequent itemset lattice generated by Close containing the closed itemsets, the generators and the classes of equivalence associated to a data mining context $\mathcal{D}$ with *minsupp=2/6*.

Association rule generation

The association rules generation concerns the composition of association rules with confidence greater or equal to a given minimum confidence, *minconf,* as stated in definition . Once the frequent Closed itemset extraction, generators and closure was done (see FIG. 7.5), Close generates the min-max association rules. We understand the min-max association rules as the most general non-redundant association rules according to their semantic. Informally, an association rule is redundant if it brings the same information or less information than is brought by another rule of same support and confidence. Then, the min-max association rules are the non-redundant association rules having minimal antecedent and maximal consequent: r is a min-max association rule if no other association rule r' has the same support and confidence, an antecedent that is a subset of the antecedent of r and a consequent that is a superset of the consequent of r.

Definition 6 (Min-max association rules) *Let $\mathcal{AR}$ be the set of association rules extracted. An association rule $r : I_1 \rightarrow I_2 \in \mathcal{AR}$ is a min-max association rule iff $\nexists\, r' : I'_1 \rightarrow I'_2 \in \mathcal{AR}$ with* $\text{supp}(r') = \text{supp}(r)$, $\text{conf}(r') = \text{conf}(r)$, *$I'_1 \subseteq I_1$ and $I_2 \subseteq I'_2$.*

Based on this definition, Close characterizes exact and approximate min-max association rules that constitute respectively the *min-max exact basis* and the *min-max approximate basis.* We understand as *exact association rules*, noted $I \Rightarrow I'$, such rules that have a 100% confidence, and *approximate association rules*, noted $I \Rightarrow I'$, such rules which have a confidence lower than 100%. Exact association rules are valid for all objects in the data set whereas approximate association rules are valid for a proportion of objects equal to their confidence.

Exact min-max association rules

First, note that exact association rules, with the form $r : I_1 \Rightarrow (I_2 \setminus I_1)$, are rules between two frequent itemsets $I_1 \subset I_2$ having the same closure: $\gamma(I_1) = \gamma(I_2)$. Since $conf(r) = 1$ we have $supp(I_1) = supp(I_2)$, and as $I_1 \subset I_2$ we see that $\gamma(I_1) = \gamma(I_2)$. We define min-max association rules among these exact rules.

Let g be the generator of $\gamma(I_1) = \gamma(I_2)$ such that $g \subseteq I_1$. Since g is minimal, we have $g \subseteq I_1 \subset I_2 \subseteq \gamma(I_2)$. Furthermore, all itemsets in the interval $[g, \gamma(I_2)]$, defined by inclusion[30], have the same closure $\gamma(I_2)$ and thus the same support. The min-max association rule among all rules with the form $r : I_1 \Rightarrow (I_2 \setminus I_1)$ with $I_1, I_2 \in [g, \gamma(I_2)]$ is the rule $g \Rightarrow (\gamma(I_2) \setminus g)$. This rule has a minimal antecedent, g, and a maximal consequent, $\gamma(I_2)$, among all rules that have the same support.

We generalize this definition to all generators of the frequent Closed itemset $\gamma(I_2)$. Let $Gen_{\gamma(l_2)}$ be the set of these generators. All exact min-max association rules constructed with $\gamma(I_2)$ are rules with the form $g \Rightarrow (\gamma(I_2) \setminus g)$ with $g \in Gen_{\gamma(l_2)}$. The extension of this property to all frequent Closed itemsets defines the min-max exact basis containing all exact min-max association rules characterized in definition .

[30] The interval $[I_1, I_2]$ contains all the supersets of I_1 that are subsets of I_2.

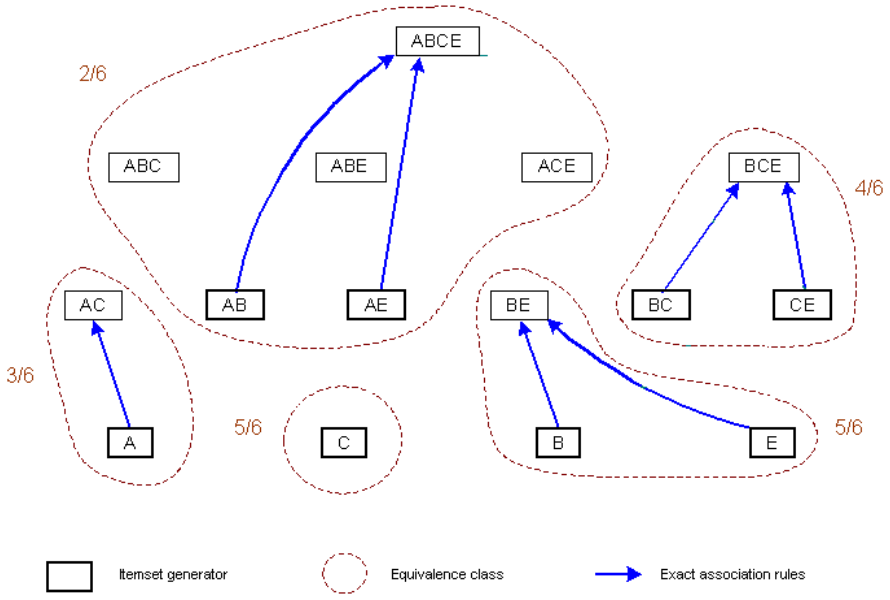

FIG. 7.6: All exact association rules extracted from $\mathcal{D}$.

Definition 7 (Min-max exact basis) *Let Closed be the set of frequent closed itemsets extracted from the context and, for each frequent closed itemset f, let's denote Gen_f the set of generators of f. The min-max exact basis is:*

$$MinMaxExact = \{r : g \Rightarrow (f \setminus g) \mid f \in Closed \ \wedge \ g \in Gen_f \ \wedge \ g \neq f\}.$$

The condition $g \neq f$ discards rules with the form $g \Rightarrow \varnothing$; it is equivalent to the condition $I_1 \subset I_2$ in the definition of association rules. We state in the following proposition that the min-max exact basis does not lead to information loss.

Example The min-max exact basis extracted from context $\mathcal{D}$ for *minsupp* $= 2/6$ is presented in TABLE 7.2. It contains seven rules whereas the set of all exact association rules contains fourteen rules.

Generator	Closure	Exact rule	Supp
{A}	{AC}	A ⇒ C	3/6
{B}	{BE}	B ⇒ E	5/6
{C}	{C}		
{E}	{BE}	E ⇒ B	5/6
{AB}	{ABCE}	AB ⇒ CE	2/6
{AE}	{ABCE}	AE ⇒ BC	2/6
{BC}	{BCE}	BC ⇒ E	4/6
{CE}	{BCE}	CE ⇒ B	4/6

TABLE 7.2: Min-max exact basis extracted from $\mathcal{D}$.

All exact association rules and their supports can be deduced from the min-max exact basis, TABLE 7.2, as presented in the FIG. 7.6. The FIG. 7.6 shows the exact association rules obtained from the min-max exact basis, including the equivalence classes, the Closed itemsets

and generators. We can see that the exact rules are the result of all possible non-redundant combinations of exact association rules within an equivalence class.

Approximate min-max association rules

Approximate association rules, with the form $r\colon I_1 \Rightarrow (I_2 \setminus I_1)$, are rules between two frequent itemsets $I_1 \subset I_2$ such that $\gamma(I_1) \subset \gamma(I_2)$. Since $conf(r) < 1$ we have $supp(I_1) > supp(I_2)$, and we deduce that $\gamma(I) \subset \gamma(I_2)$.

We deduce the definition of approximate min-max association rules. Let g_1 be a generator of the frequent Closed itemset f_1 and g_2 be a generator of the frequent Closed itemset f_2 such that $f_1 \subset g_2 \subseteq I_2 \subseteq f_2$. All rules with the form $r\colon I_1 \Rightarrow (I_2 \setminus I_1)$ where $I_1 \in [g_1, f_1]$ and $I_2 \in [g_2, f_2]$ have the same confidence and the same support since g_1, I_1 and f_1 have the same support as well as g_2, I_2 and f_2. We then deduce that the min-max association rule among all these rules is $g_1 \Rightarrow (f_2 \setminus g_1)$. Indeed, g_1 is the minimal itemset in $[g_1, f_1]$ and f_2 is the maximal itemset in $[g_2, f_2]$.

The generalization of this property to all couples of frequent itemsets I_1 and I_2 such that $I_1 \subset I_2$ and $supp(I_1) \neq supp(I_2)$ defines the min-max approximate basis containing all approximate min-max association rules characterized in definition .

Definition 8 (Min-max approximate basis) *We denote Gen the set of generators of the frequent closed itemsets in Closed. The min-max approximate basis is:*

$$MinMaxApprox = \{r\colon g \rightarrow (f \setminus g) \mid f \in Closed \ \wedge \ g \in Gen \ \wedge \ \gamma(g) \subset f\}.$$

Example The $min - max$ approximate basis extracted from context $\mathcal{D}$ for $minsupp = 2/6$ and $minconf = 2/5$ is presented in Table 7.3. It contains ten rules whereas the set of all approximate association rules, presented in Table 7.4, contains thirty-six rules.

Generator	Closure	Closed superset	Approximate rule	Supp	Conf
{A}	{AC}	{ABCE}	A → BCE	2/6	2/3
{B}	{BE}	{BCE}	B → CE	4/6	4/5
{B}	{BE}	{ABCE}	B → ACE	2/6	2/5
{C}	{C}	{AC}	C → A	3/6	3/5
{C}	{C}	{BCE}	C → BE	4/6	4/5
{C}	{C}	{ABCE}	C → ABE	2/6	2/5
{E}	{BE}	{BCE}	E → BC	4/6	4/5
{E}	{BE}	{ABCE}	E → ABC	2/6	2/5
{AB}	{ABCE}				
{AE}	{ABCE}				
{BC}	{BCE}	{ABCE}	BC → AE	2/6	2/4
{CE}	{BCE}	{ABCE}	CE → AB	2/6	2/4

TABLE 7.3: Min-max approximate basis extracted from $\mathcal{D}$.

All approximate association rules can be deduced, with their supports and confidences, from the min-max approximate basis. Table 7.3, as presented in Table 7.4.

Indeed, the resulting rules in Table 7.4 can be further reduced without losing the ability to deduce all approximate association rules, by removing *transitive min-max association rules*

Approx. rule	Supp	Conf	Approx. rule	Supp	Conf	Approx. rule	Supp	Conf
BCE → A	2/6	2/4	B → ACE	2/6	2/5	B → CE	4/6	4/5
AC → BE	2/6	2/3	C → ABE	2/6	2/5	C → BE	4/6	4/5
BC → AE	2/6	2/4	E → ABC	2/6	2/5	E → BC	4/6	4/5
BE → AC	2/6	2/5	A → BC	2/6	2/3	A → B	2/6	2/3
CE → AB	2/6	2/4	B → AC	2/6	2/5	B → A	2/6	2/5
AC → B	2/6	2/3	C → AB	2/6	2/5	C → A	3/6	3/5
BC → A	2/6	2/4	A → BE	2/6	2/3	A → E	2/6	2/3
BE → A	2/6	2/5	B → AE	2/6	2/5	E → A	2/6	2/5
AC → E	2/6	2/3	E → AB	2/6	2/5	B → C	4/6	4/5
CE → A	2/6	2/4	A → CE	2/6	2/3	C → B	4/6	4/5
BE → C	4/6	4/5	C → AE	2/6	2/5	C → E	4/6	4/5
A → BCE	2/6	2/3	E → AC	2/6	2/5	E → C	4/6	4/5

TABLE 7.4: Approximate association rules extracted from $\mathcal{D}$.

Non-transitive approximate min-max association rules

We can further reduce the number of redundant approximate association rules extracted without losing the ability to deduce all approximate association rules, with support and confidence, by removing *transitive min-max association rules*.

A min-max association rule $g \Rightarrow (f \setminus g)$ with $\gamma(g) \subset f$ is transitive if it exists a frequent Closed itemset f' such that $\gamma(g) \subset f' \subset f$. Let g' be the generator of f' such that $\gamma(g) \subset g' \subseteq f' \subset f$. Then, we have the two following approximate min-max association rules: $g \Rightarrow (f' \setminus g)$ and $g' \Rightarrow (f \setminus g')$. The rule $g \Rightarrow (f \setminus g)$ is the transitive composition of the two previous rules; its support is equal to the second rule's support and its confidence is equal to the product of their confidences.

We generalize this characterization to all triplets consisting of a generator g, its closure f and a Closed superset f' of f to define the *non-transitive min-max approximate basis*, that is the transitive reduction of the min-max approximate basis. Let us denote $I_1 \lessdot I_2$ where an itemset I_1 is an immediate predecessor of an itemset I_2, i.e. $\nexists l_3$ such that $I_1 \subset I_3 \subset I_2$. The non-transitive min-max approximate rules are of the form $g \Rightarrow (f \setminus g)$ where f is a frequent Closed itemset and g a frequent generator such that $\gamma(g)$ is an immediate predecessor of f.

Definition 9 (Non-transitive min-max approximate basis) *The non-transitive min-max approximate basis is:*

$$MinMaxReduc = \{r : g \rightarrow (f \setminus g) \mid f \in Closed \ \wedge \ g \in Gen \ \wedge \ \gamma(g) \lessdot f\}.$$

Remark 1 *This transitive reduction decreases the number of approximate rules extracted, by selecting the most precise rules, i.e. with the highest confidences, since transitive rules have lower confidences than non-transitive rules.*

Example

The non-redundant min-max approximate basis extracted from context $\mathcal{D}$ for *minsupp* = 2/6 and *minconf* = 2/5 is presented in TABLE 7.5. It contains only seven rules, that is three rules less than the approximate min-max basis. These three rules are B → ACE, C →

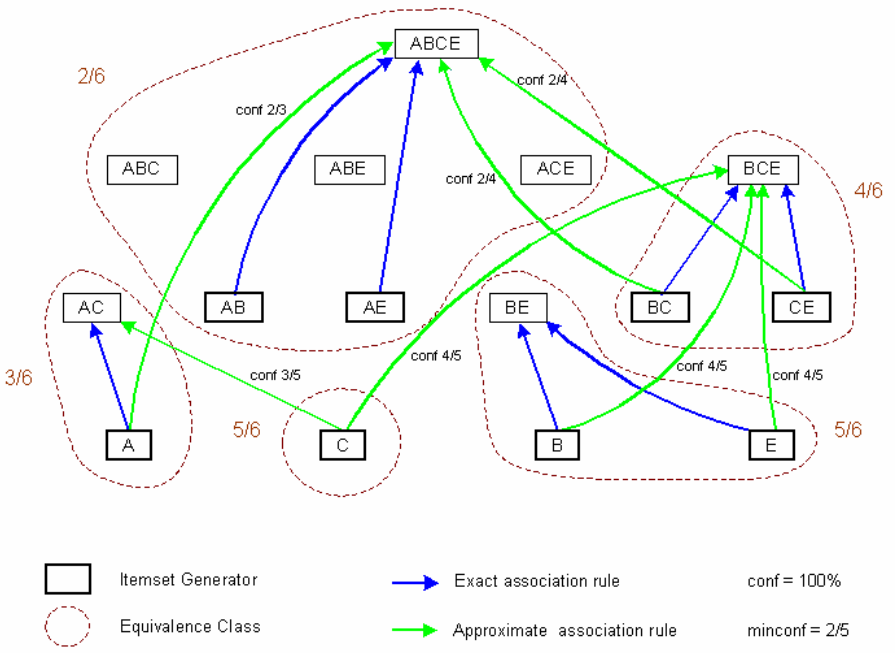

FIG. 7.7: Non redundant extracted exact and approximate association rules for *minsupp*= 2/6 and *minconf*= 2/5

BE and E → ABC that have minimal support and confidence measures among the ten rules of the approximate min-max basis.

Generator	Closure	Closed superset	Approximate rule	Supp	Conf
{A}	{AC}	{ABCE}	A → BCE	2/6	2/3
{B}	{BE}	{BCE}	B → CE	4/6	4/5
{B}	{BE}	{ABCE}			
{C}	{C}	{AC}	C → A	3/6	3/5
{C}	{C}	{BCE}	C → BE	4/6	4/5
{C}	{C}	{ABCE}			
{E}	{BE}	{BCE}	E → BC	4/6	4/5
{E}	{BE}	{ABCE}			
{AB}	{ABCE}				
{AE}	{ABCE}				
{BC}	{BCE}	{ABCE}	BC → AE	2/6	2/4
{CE}	{BCE}	{ABCE}	CE → AB	2/6	2/4

TABLE 7.5: Non-transitive min-max approximate basis extracted from $\mathcal{D}$.

All approximate association rules, with support and confidence, can be deduced from the non-transitive min-max approximate basis.

Generated exact association rules and non-transitive approximate association rules extracted from context D for at least *minsupp*= 2/6 and *minconf*= 2/5 are presented in FIG. 7.7

Extracted association rules are presented in FIG. 7.7, the dark arrows represent the exact rules and the dashed arrows the approximate rules. The set generated from all these rules represents the minimal number of non-redundant rules with at least $minsupp = 2/6$ and $minconf = 2/5$. This set has already pruned all the redundant and transitive association rules, obtaining the minimal set without losing the ability to deduce all approximate and exact association rules in context D.

7.4 GENMINER Algorithm

GENMINER association rules discovery approach [201] is a co-clustering and bi-clustering method that integrates gene annotations and gene expression measures to discover intrinsic associations among both data sources based on co-ocurrence patterns. It is a co-clustering approach which integrates co-expressed and co-annotated gene groups at the same time (the general methodology of co-clustering algorithms was explained in section 5.4). Furthermore, it is a bi-clustering algorithm that finds co-annotated and co-expressed gene groups even in a small subset of biological conditions.

As an association rules discovery approach, GENMINER, follows the four steps of the ARD process: data selection and pre-treatment, frequent itemsets extraction, association rules generation, and interpretation of extracted rules. Here, we explain more in detail each one of these four steps, and we end this section with the implementation details of GENMINER algorithm.

7.4.1 Data selection and data pretreatment

In the present work, ARD is applied to extract associations among gene annotations and gene expression patterns, integrating in this way biological sources of information with experimental numeric data (see FIG. 5.1). In order to extract these associations, we define the data mining context for gene expression technologies as a triple $\mathcal{DG} = (\mathcal{T}, \mathcal{I}, \mathcal{R})$. Here the transactions $t \in \mathcal{T}$ are represented by genes, and the items $i \in \mathcal{I}$ are each one of the gene characteristics as the gene expression measures at each biological condition or the gene annotations. $\mathcal{R} \subseteq \mathcal{T} \times \mathcal{I}$ is a binary relation between genes and their respective gene expression measures and/or gene annotations. In TABLE 7.6 we illustrate the gene expression data context $\mathcal{DG} = (\mathcal{T}, \mathcal{I}, \mathcal{R})$ for a given example. The first column contains the transactions, $t \in \mathcal{T}$, represented by the genes and their corresponding identity. The next five columns are the items, $i \in \mathcal{I}$, that are divided between two groups: gene expression measures (columns 2-3) and gene annotations (columns 4-6).

Gene annotations are issued from one or more of the six sources of biological information as explained in chapter 3: Minimal Microarray Information (genes, biological conditions, gender, age etc.), molecular databases (GenBank, Embl, Unigene, etc.), semantic sources as thesaurus, ontologies, taxonomies or semantic networks (UMLS, GO, taxonomy, etc.), gene expression databases (GEO, Arrayexpress, Microarray database, etc.), bibliographic databases (Medline, Biosis, etc.), and gene/protein related specific sources (ONIM, KEGG, etc.). The

Transactions: t	Items: i				
Genes	Expression Measures		Annotations		
	C1	C2	C	D	E
g_1	-1	0	1	1	0
g_2	0	1	1	0	1
g_3	-1	1	1	0	1
g_4	0	1	0	0	1
g_5	-1	1	1	0	1
g_6	0	1	1	0	1

TABLE 7.6: Example of association rules extraction context $\mathcal{DG}$ containing heterogeneous information: gene expression measures and gene annotations

characteristics of each one of the sources of biological information were explained in chapter 3. A discussion on the integration characteristics of each one of the six sources of information was discussed in section 5.5.

Gene annotations are boolean variables, i.e. $i \in \{0, 1\}$, indicating if an annotation pertains, $i = 1$ or not, $i = 0$, to a given gene g. For example in TABLE 7.6 we can see that the gene with identity g_1 contains two annotations represented by C, D.

Gene expression measures are cleaned gene expression data obtained from the biological gene expression technology experiment, they must be already statistically treated as seen in section 2.1 and section 2.2. Generally, gene expression measures are continuous quantitative variables, so they have to be discretized. Several discretization methods were proposed in section 8.3, we have classified them as *biological*, s*tatistical* and *mining* methods. The choice of discretization method depends on the kind of gene expression data to analyze (time series, cancer studies, multi-tissue studies etc.) and the main goal of the study.

In the case of independence between each one of the biological conditions or when the dependent biological conditions are already pre-treated as independent conditions (as seen in section 2.1-2.3), we suggest the use of our discretization algorithm: [200] (explained in section 8.3.2.1).

NORDI is based on a statistical detection of outliers and the continuous application of normality tests for transforming the initial distribution "almost normal" [31] to a "more normal" one. Once the distribution of the matrix is "more normal", it calculates the cutoffs as seen in $z-score$ methodology (explained in section). In the example of TABLE 7.6, we see that for each one of the biological conditions $C1$, $C2$, we have applied the NORDI algorithm. Thus, the final cutoffs threshold were calculated using the $z - score$ equation 8.4. We have taken the following conventions: gene over-expression is equal to 1 or $\uparrow$, gene under-expression is stated as -1 or $\downarrow$ and gene unexpression as 0 (as seen in TABLE 7.7).

7.4.2 Frequent itemsets extraction

GENMINER uses the Close [229] association rules mining algorithm for frequent Closed itemsets extraction. The frequent Closed itemsets constitute a generator set for all concerned frequent itemsets. This representation is a basis, i.e., a generating set for all association rules,

[31] By "almost" we mean that the sample S_j can be normally distributed without the outliers presence.

their supports and equivalence classes associated to a given data mining context with minsupp and their confidences, and all of them can be retrieved without accessing the data. The process of extracting frequent Closed itemsets, generators and equivalence classes was explained in section 7.3.3 The case of data mining context $\mathcal{D}$ was illustrated in FIG. 7.5. We can interpret the same FIG. 7.5 in the data mining context $\mathcal{DG}$ by simply replacing the itemsets A and B in data mining context $\mathcal{D}$ by the biological conditions $C1$ and $C2$ in context $\mathcal{DG}$ (where $C1$ is under-expressed if expressed and $C2$ is over-expressed if expressed) and taking C, D and E as gene annotations.

7.4.3 Association rules generation

As mentioned in the introduction, ARD has been previously used to mine gene expression data sets in order to discover associations among subsets of genes based only on their gene expression profiles [22], [81], [170], [310], [224], [263] and [124]. The association rules generated by these approaches are of the following form: $[\downarrow]$ *gene* $X \Longrightarrow [\uparrow]$ *gene* Y, $[\downarrow]$ *gene* Z, meaning that in a significant number of conditions when gene X is under-expressed it is likely to observe an over-expression of gene Y and under-expression of gene Z. These algorithms concern exclusively gene expression measures without taking into account biological knowledge.

Recently, an ARD methodology [61] has been used to integrate gene expression profiles and gene annotations with rules of the form: *Annotation* $\Longrightarrow [\downarrow] C1$, $[\uparrow] C2$ which means when a set of genes annotated with the characteristic *Annotation* occurs, the set of genes is likely to be under-expressed in biological condition $[\downarrow] C1$ and over-expressed in biological condition $[\uparrow] C2$. Thus, the gene annotations are always in the antecedent of the rule and the gene expression measures are always in the consequent of the rule.

In contrast to past approaches, GENMINER is applied to identify itemsets of gene expression measures and gene annotations that frequently co-occur in a data mining context, $\mathcal{DG}$, establishing relationships among them of the form:

$$\begin{aligned} [\downarrow] C1 &\Rightarrow [\uparrow] C2,\ C,\ D \\ C &\Rightarrow [\downarrow] C1 \end{aligned} \tag{7.2}$$

The first case states that when a set of under-expressed genes in biological condition $C1$ occurs, these genes are likely to be over-expressed in biological condition $C2$, and they are annotated by characteristics C and D. The second case means that a set of genes annotated with the characteristic C were under-expressed in biological condition $C1$. Thus, GENMINER is interested in obtaining all rules without regarding if the gene annotations or gene expressions are in the antecedent of the rule or the consequent of the rule. The main reason is that all possible combinations of association rules taking either gene annotations or gene expression measures at any of the two parts of an association rule, antecedent or consequent, or even mixed as stated in the first rule of equation 7.2 can be biologically meaningful, so they cannot be eliminated a priori.

In order to generate rules containing either gene annotations or gene expression profiles at any of the two parts of an association rule, the GENMINER algorithm uses again the

Close [229] mining algorithm for generating minimal non-redundant rules to a given minimum confidence *minconf*. Once the frequent Closed itemsets, generators, and closures are built (as see in section 7.3.3), Close generates the $min-max$ association rules, i.e. non-redundant rules according to their semantic. The process of generating exact and approximate $min-max$ association rules for a given support, *minsupp*, and a 100% confidence (exact rule) or a minium confidence *minconf* (approximate rule), was explained in section 7.3.3. The idea behind Close rule generation is the extraction of a min-max basis (as defined in and 7.3) of a given context $\mathcal{D}$, from which all possible generation rules can be deduced at certain *minsupp* and *minconf*. Once all non-redundant exact and approximative rules are obtained, Close removes all approximate and transitive min-max association rules by extracting a min-max non-transitive basis (as defined in). This transitive reduction decreases the number of approximate rules extracted, by selecting the most precise rules, i.e. with the highest confidences, since transitive rules have lower confidences than non-transitive rules.

As an example of GENMINER association rule extraction, the lector can follow the example given in section 7.3 concerning the data mining context $\mathcal{D}$, as seen in FIG. 7.6, TABLE 7.4 and FIG. 7.7 for exact rules, approximate rules, and the rules extracted after transition reduction respectively. The lector can interpret these results in the data mining context $\mathcal{DG}$ by simply replacing the itemsets A and B in data mining context $\mathcal{D}$ by the biological conditions $C1$ and $C2$ in context $\mathcal{DG}$ (where $C1$ is under-expressed if expressed and $C2$ is over-expressed if expressed, concerning this specific example) and taking C, D and E as gene annotations as stated in TABLE 7.6.

Generated non-redundant exact and non-transitive association rules extracted from context $\mathcal{DG}$ for at least *minsupp*$= 2/6$ and *minconf*$= 2/5$ are presented in FIG. 7.8:

Extracted association rules are presented in FIG. 7.7, the dark arrows represent the exact rules and the dashed arrows the approximate rules. The set generated from all these rules represents the minimal number of non-redundant and non-transitive rules with at least *minsupp*$= 2/6$ and *minconf*$= 2/5$. This set has already pruned all the redundant and transitive association rules, obtaining the minimal association rule set without losing the ability to deduce all approximate and exact association rules in context DG.

For example taking the exact rule of $E \Longrightarrow [\uparrow] C2$ means that when a set of genes annotated with characteristic E occurs, the same genes are likely to be over-expressed in biological condition $C2$ with *minsupp*$= 5/6$ and 100% confidence. Taking the approximate rule of $[\downarrow] C1 \Longrightarrow [\uparrow] C2$, C, E means that when a set of under-expressed genes in biological condition $C1$ occurs, these genes are likely to be annotated with characteristics C, E and to be over-expressed in biological condition $C1$ with *minsupp*$= 2/6$ and *minconf*$= 2/3$. We can see these rules illustrated in FIG. 7.7.

Gene expression technology data sets contain thousands of genes and thousands of gene attributes, producing a huge quantity of association rules. This problem becomes crucial when these data sets are highly correlated, such as data sets issued from gene expression technologies [21, 54, 281]. So, the data analyst is confronted with the following problems: How to handle such a list of association rules ? Is it possible to reduce its size without losing information ?

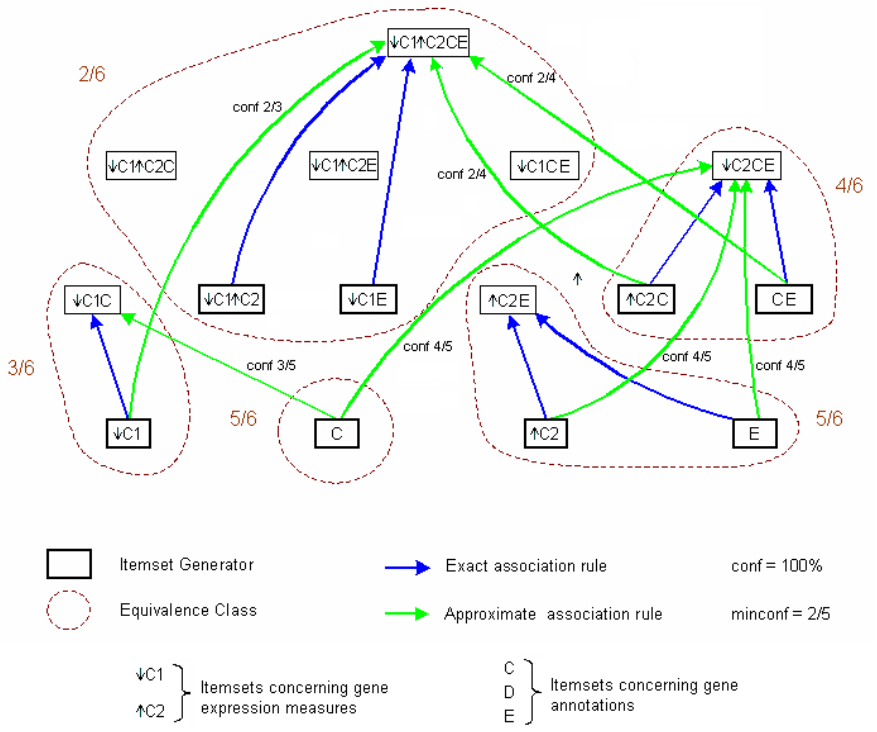

FIG. 7.8: Non-redundant and non-transitive extracted exact and approximate association rules in context DG for $minsupp = 2/6$ and $minconf = 2/5$

Moreover, the inspection of extracted association rules shows that redundant rules represent the majority of them. Their suppression will thus considerably reduce the number of rules to be handled by the analyst. In addition, redundant rules can be misleading as will be discussed in the next example. Thus, the following question arises: How to reduce extracted association rules to a smaller list containing only non-redundant association rules

The GENMINER solution to this problem was explained here included in the third mining step: generating the non-redundant and non-transitive exact and approximate rules for a given *minsupp* and *minconf* using the $min - max$ basis Close methodology explained in section 7.3. The importance of the adopted pruning methodology will be illustrated in the next example.

In order to show the importance of our pruning methodology, we have treated a sample of itemsets, I_1, containing the itemsets of three equivalence classes: *minsupp*= 2/6, *minsupp*= 3/6, and *minsupp*= 5/6 (see FIG. 7.8). All approximate association rules extracted from the sample of itemsets , I_1, in a data mining context $\mathcal{DG}$ (see TABLE 7.6) are illustrated in TABLE 7.7 (the exact association rules obtained from the sample I_1 are not taken into account.

Number	Approximate rule	Supp	Conf
1	$[\downarrow] C1, C \Rightarrow [\uparrow] C2, E$	2/6	2/3
2	$[\downarrow] C1, C \Rightarrow [\uparrow] C2$	2/6	2/3
3	$[\downarrow] C1, C \Rightarrow E$	2/6	2/3
4	$C, E \Rightarrow [\downarrow] C1$	2/6	2/4
5	$[\downarrow] C1 \Rightarrow [\uparrow] C2, C, E$	2/6	2/3
6	$[\downarrow] C1 \Rightarrow [\uparrow] C2, E$	2/6	2/3
7	$[\downarrow] C1 \Rightarrow C, E$	2/6	2/3
8	$[\downarrow] C1 \Rightarrow [\uparrow] C2, C$	2/6	2/3
9	$[\downarrow] C1 \Rightarrow E$	2/6	2/3
10	$[\downarrow] C1 \Rightarrow [\uparrow] C2$	2/6	2/3
11	$C \Rightarrow [\downarrow] C1 , [\uparrow] C2, E$	2/6	2/5
12	$C \Rightarrow [\downarrow] C1$	3/6	3/5

TABLE 7.7: Sample of approximate association rules extracted from $\mathcal{DG}$.

After applying the $min - max$ basis for extracting the minimal non-redundant approximate rules to the sample I_1 (see TABLE 7.7) with *minsupp*= 2/6 and *minconf*= 2/5, we obtain the following minimal rules:

Approximate rule	Supp	Conf
$[\downarrow] C1 \Rightarrow [\uparrow] C2, C, E$	2/6	2/3
$C \Rightarrow [\downarrow] C1, [\uparrow] C2, E$	2/6	2/5
$C \Rightarrow [\downarrow] C1$	3/6	3/5

TABLE 7.8: Approximate non-redundant association rules extracted from $\mathcal{DG}$.

The three approximate rules shown in TABLE 7.8 are the minimal basis for generating the 12 rules illustrated in TABLE 7.7 with a *minsupp*= 2/6 and *minconf*= 2/5. For example, considering the minimal non-redundant approximate rule $[\downarrow] C1 \Rightarrow [\uparrow] C2, C, E$ in TABLE 7.8

,we deduce the rules:

$$\begin{aligned}
[\downarrow] C1 &\Rightarrow [\uparrow] C2 \\
[\downarrow] C1 &\Rightarrow E \\
[\downarrow] C1 &\Rightarrow [\uparrow] C2, C \\
[\downarrow] C1 &\Rightarrow [\uparrow] C2, E \\
[\downarrow] C1 &\Rightarrow C, E
\end{aligned}$$

The rule $[\downarrow] C1 \Rightarrow C$ is not generated because it is an exact rule belonging to the same Close interval. In the same way we can deduce the remaining redundant approximate rules

$$\begin{aligned}
[\downarrow] C1, [C] &\Rightarrow [\uparrow] C2, E \\
[\downarrow] C1, C &\Rightarrow [\uparrow] C2 \\
[\downarrow] C1, C &\Rightarrow E \\
C, E &\Rightarrow [\downarrow] C1
\end{aligned}$$

from the three non-redundant approximate rules illustrated in TABLE 7.8.

After pruning the redundant rules, we applied the transitive rules pruning, obtaining from the $min - max$ basis () for extracting the minimal non-transitive approximative rules to the sample $I_1$with $minsupp = 2/6$ and $minconf = 2/5$. We obtain the minimal non-redundant rules and non-transitive approximate rules $[\downarrow] C1 \Rightarrow [\uparrow] C2$, C, E and $C \Rightarrow [\downarrow] C1$ illustrated in FIG. 7.8 in the itemset sample I_1. The transitive rule $[C] \Rightarrow [\downarrow] C1$, $[\uparrow] C2$, E was eliminated because it can be derived from rules $[\downarrow] C1 \Rightarrow [\uparrow] C2$, C, E and $C \Rightarrow [\downarrow] C1$.

Therefore, the data analyst has to focus on rules 5 and 12, because none of the 10 redundant or transitive association rules 1 to 4 or 6 to11 (see TABLE 7.7) adds any information. Furthermore, if the analyst takes one of these 10 redundant rules, for example $[\downarrow] C1$, $C \Rightarrow [\uparrow] C2$, he can come to wrong conclusions, because the analyst will believe that a set of genes has $\frac{2}{3}\%$ chances to be over-expressed in biological condition $C2$ if these genes are annotated with characteristic C and under-expressed in biological condition $C1$. As a matter fact, this set of genes has $\frac{2}{3}\%$ chances to be over-expressed in biological condition C and annotated with characteristics C and D if they are under-expressed in biological condition $C1$ (explained by the non-redundant and non-transitive rule $[\downarrow] C1 \Rightarrow [\uparrow] C2$, C, E). Thus, redundant rules can be misleading and cause misinterpretations of the results. We believe that extracting only rule 5 and 12 (for the approximative rule case) will improve the result relevance and usefulness. We conclude that the most relevant rules from the analyst's point of view are the rules that have minimal antecedent (left-hand side) and maximal consequent for a given *support* and *confidence*. In our example, these are the rules 5 and 12 of the TABLE 7.7, i.e. $[\downarrow] C1 \Rightarrow [\uparrow] C2$, C, E and $C \Rightarrow [\downarrow] C1$.

7.4.4 Interpretation of extracted rules

At this stage, GENMINER has extracted all non-redundant and non-transitive approximate and exact association rules containing information about the relationships among gene ex-

pression measures and gene annotations for gene groups. The number of extracted rules in highly correlated gene expression technology data is generally enormous and the interpretation task becomes considerably complex to analysts, even if thedata were already pruned. Our algorithm delivers the minimal basis containing only minimal non-redundant rules without information loss, and it suggests two paths for selecting the "most interesting" rules. These paths are:

1. Rule volume reduction concerns the creation of filters to select only the kind of association rules that the expert expect to obtain in order to achieve the main goal of the biological experiment. For example, choosing only the rules that have gene annotations as antecedent and gene expression measures as consequent, or vice versa and so on.
2. Rule expert selection concerns the extraction of the rules containing items that constitute the inherent knowledge of the expert. Experts knowledge can also be understood as the past knowledge concerning the main goal of the biological experiment. For example, if the goal is to find the group of genes that may be the cause of brain cancer, the rules concerning the items that represent accurate knowledge in this subject have to be chosen.

Some approaches have dealt with the automatic or semi-automatic knowledge extraction problem represented by the second path of research. These knowledge extraction approaches can be divided into objective and subjective. The objective approaches build statistical interest measures, which evaluate the accuracy and the quality of extracted motifs. In contrast, the subjective approaches construct subjective interest measures focused on the user, expert or analyst interest. More precisions concerning these two axes of research can be obtained in Brisson [56]. A survey concerning the objective approaches has been made by Azé [14] and another survey concerning both approaches objective and subjective has been made by McGarry [207]. .

GENMINER suggests the utilization of a new objective-subjective approach named Keops [56], which differentiates interesting rules from non-interesting rules taking into account two kinds of interest measures, the subjective measure represented by a prior knowledge (issued from one of the six sources of biological information or the experts knowledge in the field) and the objective measure using the support, confidence and lift of a rule.

7.4.5 Implementation

The GENMINER algorithm was implemented as a C++ program containing the Close extraction algorithm (More details see in [229]). The NORDI discretization algorithm was implemented using the R language and using several libraries of the BIOCONDUCTOR open source project. The programs for selecting the different kinds of rules from the GENMINER extracted rules file were implemented in python language.

7.5 Results of DeRisi Data Set

In order to evaluate our approach, the GENMINER algorithm was applied to the DeRisi data set which is one of the most studied in this field [90]. The DeRisi data set measures the variations in gene expression profiles during the cellular process of diauxic shift for the yeast *Saccharomyces Cerevisiae*. When inoculated into a glucose-rich medium (anaerobic growth), the budding yeast can convert the glucose to ethanol (aerobic respiration), the shift from anaerobic fermentation of glucose to aerobic respiration of ethanol is the so-called *diauxic shift*.

This section is organized as follows, first we present the DeRisi data set selection and pretreatment. Then, we present and validate the results obtained with GENMINER, comparing with ARD algorithm of Carmona et al. [61]. Finally, we underline the principal biological interpretations obtained with GENMINER.

7.5.1 DeRisi data set selection and pretreatment

The DeRisi data set was principally used to validate and compare our approach with ARD approach of Carmona et al. [61]. Thus, we have used the same parameters for gene expression measures (including discretization cut-offs) and gene expression annotations.

Gene expression measures

The microarray technique used is spotted cDNA chips obtained by two color fluorochromes with distinct emission spectra Cy3 and Cy5 (this technique was explained in section 1.2.1). The DeRisi data set contains the expression levels of 6199 ORF's (opening reading frames) of the yeast (an entirely sequenced organism) for 7 temporal points that correspond to samples harvested at successive two-hour intervals after an initial nine hours of growth.

Data treatment and discretization

The data set was pre-treated by taking the log_2 ratios (to consider cellular inductions and repressions in a numerically equal way) and applying the imputation algorithm of k-nearest neighbors [184] in order to treat the missing values (1.9% of the total).

In order to compare with the ARD algorithm of Carmona et al., we have used the $2 - fold\ change$ cutoff method (explained in section 7.4) proposed by Carmona et al.[61] Let us assume that the gene expression measures are presented as a matrix: X with n genes (rows) and m biological conditions (columns) where $X_{i,j}$ is the expression measure of gene i in biological condition j. $X_{i,j} \in \mathbb{R}$. Applying the $2 - fold$ threshold cutoffs, shown in the equation 8.2 and explained in section 7.4, the discretization intervals for every $X_{i,j} \in X$ are:

$$\begin{aligned} X_{i,j} &\geq 1 \Longrightarrow X_{i,j} \quad \text{over-expressed or } \uparrow \\ X_{i,j} &\leq -1 \Longrightarrow X_{i,j} \quad \text{under-expressed or } \downarrow \\ -1 &< X_{i,j} > 1 \Longrightarrow X_{i,j} \quad \text{Unexpressed} \end{aligned} \tag{7.3}$$

Gene annotations

Yeast genes were annotated using three sources of biological information:

- The gene/protein related specific database KEGG [161] containing the metabolic pathways in which each gene is involved (see section 3.6 for more details).
- The information of transcriptional regulators that bind to promoter regions, these data were\ reported in the article of Lee et al. [176]. This information was used to annotate yeast genes whose promoter regions were bound by at least one transcriptor regulator (with a p-value threshold of 0.0005).
- The semantic source of information Gene Ontology, which contains annotations from three different ontologies: biological processes, molecular functions and cellular annotations (explained in section 3.7).

All gene annotations were taken as boolean variables, i.e. $i \in \{0, 1\}$, indicating if an annotation belongs, $i = 1$ or not, $i = 0$, to a given gene (similarly as stated in TABLE 7.6).

Data mining context

The DeRisi data set and the corresponding annotations were transformed into a data mining context, $\mathcal{DG}$ =($\mathcal{T}$, $\mathcal{I}$, $\mathcal{R}$), as illustrated in TABLE 7.6. The transactions $t \in \mathcal{T}$ are represented by the yeast genes, and the items $i \in \mathcal{I}$ represent each one of the gene characteristics, i.e. discretized gene expression measures and gene annotations (as seen in TABLE 7.6).

We have validated our method using two different data mining contexts: mining the data using gene expression measures and KEGG gene annotations, $\mathcal{DGK}$ =($\mathcal{T}$, $\mathcal{I}$, $\mathcal{R}$); and mining the data using gene expression measures, KEGG gene annotations, transcriptional regulators, and GO annotations $\mathcal{DGKPG}$ =($\mathcal{T}$, $\mathcal{I}$, $\mathcal{R}$) as used in the ARD method presented by [61].

7.5.2 DeRisi results in mining $\mathcal{DGK}$ context

Discovering that most of the genes involved in a specific metabolic pathway are over -or under-expressed in the same experimental conditions provides clues about the biological processes that can be acting under these experimental circumstances [61]. A set of 1126 yeast genes of the whole 6199 genes included in the analysis were associated with at least one pathway from KEGG database.

Association rules extracted using the ARD algorithm (Carmona et al [61]) with *minsupp=.44%* (at least 5 transactions) and *minconf=40%* for mining $\mathcal{DGK}$ context are presented in TABLE 7.9. Since it is usual to analyze information only about individual pathways, the TABLE 7.9 presents the 21 rules, after applying a single antecedent filter.

Every rule showed in TABLE 7.9 contains a single antecedent corresponding to the KEGG annotation (second column) and a consequent composed of the differentially under-expressed (↓) or over-expressed (↑) gene expression measures at a certain biological condition (in this

Rule	Antecedent	Consecuent		Genes	Supp.	Conf.
	Annotation	C6	C7	#	%	%
1	Ribosome	↓	↓	95	8.40	72.52
2	Ribosome		↓	121	10.75	92.37
3	Ribosome	↓		96	8.53	73.28
4	Oxidative phosphorylation		↑	34	3.02	57.63
5	Citrate cycle (TCA cycle)		↑	23	2.04	76.67
6	Oxidative phosphorylation	↑	↑	29	2.58	49.15
7	Citrate cycle (TCA cycle)	↑	↑	18	1.60	60.00
8	Oxidative phosphorylation	↑		31	2.75	52.54
9	Reductive carboxylate cycle (CO2 fixation)	↑	↑	7	0.62	63.64
10	Pyruvate metabolism		↑	14	1.24	42.42
11	Glyoxylate and dicarboxylate metabolism		↑	8	0.71	57.14
12	ATP synthesis		↑	10	0.89	41.67
13	RNA polymerase		↓	17	1.51	60.71
14	Propanoate metabolism		↑	5	0.44	45.46
15	Aminoacyl-tRNA biosynthesis		↓	18	1.60	48.65
16	Methionine metabolism		↓	8	0.71	57.14
17	Selenoamino acid metabolism		↓	10	0.89	52.63
18	Cysteine metabolism		↓	5	0.44	50.00
19	Valine leucine and isoleucine biosynthesis		↓	7	0.62	43.75
20	Pantothenate and CoA biosynthesis		↓	5	0.44	45.46
21	Riboflavin metabolism		↓	5	0.44	41.67

TABLE 7.9: Rules extracted from mining $\mathcal{DGK}$ context with *minsupp=.44%* and *minconf=40%* with ARD algorithm

case only C6 and C7). Each of the rules contains the number of genes concerned by these rule (column 4), the support and the confidence (column 5 and 6) respectively.

Association rules extracted using GENMINER algorithm under the same parameters *minsupp=.44%* and *minconf=40%* and a similar data mining context $\mathcal{DGK}'$ context are presented in TABLE 7.10. The slight differences between $\mathcal{DGK}$ and $\mathcal{DGK}'$ context consists in the update of the KEGG and SGD nomenclature databases from 2006 to 2007 respectively. GENMINER has generated the closure, the generators, the exact rules, and the approximative rules. TABLE 7.10 presents the approximative rules generated by GENMINER after applying the single antecedent filter.

Rule	Antecedent	Consecuent		# Genes	Supp.	Conf.
	Annotation	C6	C7		%	%
1	Ribosome	↓	↓	96	8.53	73.28
2	Ribosome		↓	121	10.75	92.37
3	Ribosome	↓		97	8.61	74.05
4	Oxidative phosphorylation		↑	34	3.02	57.63
5	Citrate cycle (TCA cycle)		↑	23	2.04	76.67
6	Oxidative phosphorylation	↑	↑	30	2.66	50.85
7	Citrate cycle (TCA cycle)	↑	↑	19	1.69	63.33
8	Oxidative phosphorylation	↑		32	2.84	54.24
9	Reductive carboxylate cycle (CO2 fixation)	↑	↑	7	0.62	63.64
10	Pyruvate metabolism		↑	15	1.33	45.45
11	Glyoxylate and dicarboxylate metabolism		↑	8	0.71	53.33
12	ATP synthesis		↑	10	0.89	41.67
13	RNA polymerase		↓	17	1.51	60.71
14	Propanoate metabolism		↑	5	0.44	35.71
15	Aminoacyl-tRNA biosynthesis		↓	18	1.60	48.65
16	Methionine metabolism		↓	8	0.71	57.14
17	Selenoamino acid metabolism		↓	10	0.89	52.63
18	Cysteine metabolism		↓	5	0.44	50.00
19	Valine leucine and isoleucine biosynthesis		↓	7	0.62	43.75
20	Pantothenate and CoA biosynthesis		↓	5	0.44	45.46
21	Riboflavin metabolism		↓	4	0.36	36.36
22	Galactose metabolism	↑		10	0.89	31.00
23	Purine metabolism		↓	37	3.29	43.00
24	Pyrimidine metabolism		↓	28	2.49	41.00
25	Glycine. serine and threonine metabolism		↓	13	1.15	31.00
26	Starch and sucrose metabolism	↑		17	1.51	35.00

TABLE 7.10: Rules extracted from mining $\mathcal{DGK}$ context with *minsupp=.44%* and *minconf=40%* with Genminer algorithm

As we have seen, GENMINER has extracted the same 21 rules than ARD algorithm. The small differences concerning support and confidence are caused by the database update from year to year as explained before. For example the rule 1 is valid for 95 genes in ARD and 96 genes in GENMINER, that means that a new gene (of the 6199 total genes) was annotated as Ribosome in the course of one year (2006 vs. 2007) in the KEGG database.

GENMINER has detected five more rules (22-26) that were not extracted by ARD algorithm with high support and a non-negligible confidence. These rules are biologically important, and they indicate an activation of a group of genes for biological condition 6 in the case of galactose, starch and sucrose metabolism and an inhibition of a gene group for biological condition 7 in the case of purine, pyrimidine, glycine, serine and threonine metabolism. These results correspond to the yeast diauxic shift process that produces energy via metabolism while fermenting the sugar in the last part of the process.

7.5.3 DeRisi Results in mining $\mathcal{DGKPG}$ context

Another common approach used to derive biological knowledge is to extract information about transcriptional mechanisms. Promoter regions of co-expressed genes can be analyzed in order to find common upstream sequence motifs [9]. In data mining context $\mathcal{DGKPG}$ we integrated multiple types of biological information: transcriptional regulator, metabolic pathways and GO annotations. 3882 genes on the DeRisi data set were properly annotated and used for the analysis.

Association rules extracted using ARD algorithm (Carmona et al [61]) with *minsupp=.44%* (at least 5 transactions) and *minconf=100%* for mining $\mathcal{DGKP}$ (without taking into account GO annotations) context are presented in Table 7.11. After applying the redundant filter proposed in the ARD algorithm, 21 exact rules were obtained containing transcriptional regulators (written in capital letters) and KEGG annotations, as seen in Table 7.11.

We have applied the GENMINER algorithm using the same support and confidence: *minsupp=.44%* (at least 5 transactions) and *minconf=100%*, for mining $\mathcal{DGKPG}$. We have worked also with GO annotations to validate well known relationships between the KEGG database and GO annotations. Table 7.12 presents a selection of the GENMINER $min-max$ *exact basis* (explained in section 7.3) extracted from $\mathcal{DGKPG}$.

All exact association rules, supports and confidences contained in Table 7.11 can be deduced from this selection of the $min-max$ *exact basis*. The lector can construct the numbered rules in Table 7.11 by taking the correspondent Generator and Closure (with the corresponding rule number) and generate the same rule. For example, the second rule, r2, in Table 7.11 FHL1, RAP1, YAP5, Ribosome $\Rightarrow [\downarrow]C7$ can be generated by taking the second line of Table 7.12 the generator {YAP5, Ribosome}and closure $\{[\downarrow]C6, [\downarrow]C7$, RAP1, YAP5, FHL1, Translation, Ribosome} at the same *supp=.67%* and *conf=100%*. We have seen in section that the generator and its closure build the correspondent *min-max* association rule among all rules with the form YAP5, Ribosome$\Rightarrow [\downarrow]C6, [\downarrow]C7$, RAP1, FHL1, Translation.

Rule	Antecedent	Consecuent		Genes	Supp.	Conf.
	Annotation	C6	C7	#	%	%
1	FHL1, GAT3, RAP1, Ribosome		↓	34	0.88	100
2	FHL1, RAP1, YAP5, Ribosome		↓	26	0.67	100
3	FHL1, GAT3, RAP1, YAP5, Ribosome	↓	↓	21	0.54	100
4	FHL1, RAP1, PDR1, Ribosome		↓	17	0.44	100
5	SFP1, Ribosome	↓	↓	12	0.31	100
6	FHL1, GAT3, RAP1, RGM1, Ribosome	↓	↓	11	0.28	100
7	FHL1, RAP1, SFP1, Ribosome	↓	↓	11	0.28	100
8	FHL1, RAP1, PDR1, YAP5, Ribosome		↓	10	0.26	100
9	FHL1, GAT3, RAP1,YAP5, RGM1, Ribosome	↓	↓	9	0.23	100
10	FHL1, GAT3, RAP1, PDR1, YAP5, Ribosome	↓	↓	9	0.23	100
11	FHL1, RAP1, SMP1, Ribosome	↓	↓	8	0.21	100
12	FHL1, GAT3, RAP1, SFP1, Ribosome		↓	7	0.18	100
13	FHL1, RAP1,YAP5, SFP1, Ribosome		↑	6	0.15	100
14	FHL1, RAP1, PDR1, SFP1, Ribosome		↓	6	0.15	100
15	YAP6, Ribosome		↓	6	0.15	100
16	FHL1, RAP1, PDR1, YAP5, SFP1, Ribosome	↓	↓	5	0.13	100
17	FHL1, GAT3, RAP1, PDR1, YAP5, SFP1, Ribosome	↓	↓	5	0.13	100
18	FHL1, RAP1, PDR1, SMP1, Ribosome	↓	↓	5	0.13	100
19	FHL1, RAP1, MET31, Ribosome		↓	5	0.13	100
20	FHL1, YAP6, Ribosome		↓	5	0.13	100
21	HAP2, HAP3, HAP4, Oxidative phosphorylation		↓	5	0.13	100

TABLE 7.11: Rules extracted from mining $\mathcal{DGKP}$ context with *minsupp=.44%* and *minconf=100%* with ARD algorithm

Rule	Generator	Closed Itemset	Genes #	Supp. %	Conf. %
1	{GAT3, Ribosome}	{C7 [↓], GAT3, RAP1, FHL1, Translation, Ribosome}	34	0.88	100
2	{YAP5, Ribosome}	{C6 [↓], C7 [↓], RAP1, YAP5, FHL1, Translation, Ribosome}	26	0.67	100
3	{GAT3, YAP5, Ribosome}	{C6 [↓], C7 [↓], GAT3, RAP1, YAP5, FHL1, Translation, Ribosome}	21	0.54	100
4	{PDR1, Ribosome}	{C7 [↓], PDR1, RAP1, FHL1, Translation, Ribosome}	17	0.44	100
5	{Ribosome, SFP1}	{C7 [↓], SFP1, Translation, Ribosome}	12	0.31	100
6	{RGM1, Ribosome}	{C7 [↓], GAT3, RAP1, RGM1, FHL1, Translation, Ribosome}	11	0.28	100
7	{Ribosome, FHL1, SFP1}	{C7 [↓], RAP1, FHL1, SFP1, Translation, Ribosome}	11	0.28	100
8	{PDR1, YAP5, Ribosome}	{C6 [↓], C7 [↓], PDR1, RAP1, YAP5, FHL1, Translation, Ribosome}	10	0.26	100
9	{RGM1, YAP5, FHL1, C6 [↓]}	{C6 [↓], C7 [↓], GAT3, RAP1, RGM1, YAP5, FHL1, Translation, Ribosome}	9	0.23	100
10	{GAT3, PDR1, FHL1, C6 [↓]}	{C6 [↓], C7 [↓], GAT3, PDR1, RAP1, YAP5, FHL1, Translation, Ribosome}	9	0.23	100
11	{Ribosome, SMP1}	{C6 [↓], C7 [↓], RAP1, FHL1, SMP1, Translation, Ribosome}	8	0.21	100
12	{GAT3, C6 [↓], SFP1}	{C6 [↓], C7 [↓], GAT3, RAP1, FHL1, SFP1, Translation, Ribosome}	7	0.18	100
13	{YAP5, Ribosome, SFP1}	{C6 [↓], C7 [↓], RAP1, YAP5, FHL1, SFP1, Translation, Ribosome}	6	0.15	100
14	{PDR1, C6 [↓], SFP1}	{C6 [↓], C7 [↓], PDR1, RAP1, FHL1, SFP1, Translation, Ribosome}	6	0.15	100
15	{Ribosome, YAP6}	{C7 [↓], YAP6, Translation, Ribosome}	6	0.15	100
16	{C7 [↓], PDR1, RAP1, SMP1}	{C6 [↓], C7 [↓], PDR1, RAP1, FHL1, SMP1, Translation, Ribosome}	5	0.13	100
17	{GAT3, YAP5, Ribosome, SFP1}	{C6 [↓], C7 [↓], GAT3, RAP1, YAP5, FHL1, SFP1, Translation, Ribosome}	5	0.13	100
18	{PDR1, Ribosome, SMP1}	{C6 [↓], C7 [↓], PDR1, RAP1, FHL1, SMP1, Translation, Ribosome}	5	0.13	100
19	{Ribosome, MET31}	{C7 [↓], RAP1, FHL1, MET31, Translation, Ribosome}	5	0.13	100
20	{Ribosome, FHL1, YAP6}	{C7 [↓], FHL1, YAP6, Translation, Ribosome}	5	0.13	100
21	{HAP3, Oxidative phosphorylation}	{C7 [↑], HAP2, HAP3, HAP4, Transport, Metabolites and energy generation, Oxidative phosphorylation}	5	0.13	100

TABLE 7.12: Selection of the min-max exact basis extracted from $\mathcal{DGKPG}$ with *minsupp=.44%* and *minconf=100%* with GENMINER algorithm

This rule has a minimal antecedent and a maximal consequent among all rules that have the same support and confidence. Thus, each one of the elements of the consequent, even all ($[\downarrow]C6, [\downarrow]C7$, RAP1, FHL1, Translation) can be indifferently at the antecedent part of the rule conserving the same support and confidence. We have considered all these rules redundant in relation to the correspondent *min-max* association rule. So, the rule r2 of TABLE 7.11 can be obtained from the $min - max$ rule by passing the elements RAP1 and FHL1 to the antcedent part and leaving only the element $[\downarrow]C7$ in the consequent part. In a similar way, the lector can build the 21 rules presented by ARD algorithm in TABLE 7.11 by taking the respective generator and closure sets obtained with GENMINER in TABLE 7.12.

GENMINER algorithm gives three kinds of results: the $min-max$ *exact* and *approximative basis*, the exact rules and the approximative rules. So, the analyst could choose between these three outputs for guiding their analysis. As seen in the last example, ARD algorithm gives their results only in the form of association rules, without any knowledge of the characteristics of the extracted rule. On the other hand, GENMINER gives to the analyst not only the extracted rules, but also the $min - max$ *basis*, which enables to understand some of the characteristics of the obtained rules. Taking as example the second rule (obtained by ARD algorithm) presented in TABLE 7.11, r2: FHL1, RAP1, YAP5, Ribosome $\Rightarrow [\downarrow]C7$, the analyst will interpret that when a set of 26 genes annotated with the promoters: FHL1, RAP1, YAP5 and the metabolic pathway Ribosome occurs, the set of genes is likely to be under-expressed in biological condition seven $[\downarrow]C7$ with a support of .67% and 100% confidence. Indeed, taking the corresponding $min - max$ rule obtained by GENMINER YAP5, Ribosome $\Rightarrow [\downarrow]C6, [\downarrow]C7$, RAP1, FHL1, Translation; the analyst knows that the only two necessary itemsets are the gene annotations YAP5 and Ribosome and the ones that are in the consequent part could be or not in the antecedent part or consequent part of the rule. Thus, he will interpret, when a set of 26 genes annotated with YAP5 and Ribosome (necessary) and possibly ($[\downarrow]C6$, RAP1, FHL1, Translation) occurs, that the set of genes is likely to be under-expressed in biological condition seven $[\downarrow]C7$ and possibly ($[\downarrow]C6$, RAP1, FHL1, Translation) with a support of .67% and 100% confidence. This new interpretation tool enables the analyst to know more about the characteristics of the extracted rules, detecting the main relations and new relations among gene annotations and gene expression measures.

Concerning the GO annotations, we can see in TABLE 7.12 that the KEGG annotation: Ribosome and the GO annotation Translation appear always together in the closures set (R1-R20). The reason is that they are equivalent terms referring to the same biological process: production of proteins explained in section 1.1. Similarly, the KEGG annotation 'oxidative phosphorylation' and the GO annotation 'Transport' refer to the same biological process which is the terminal process of cellular respiration.

7.5.4 Biological significance of the discovered associations

In order to evaluate the biological significance of the associations provided by GENMINER, it is important to analyze two parameters: support and confidence. As we have stated, the support of a rule is the percentage of transactions (annotated genes) that shows co-occurrences of a given annotation or a similar expression pattern and the confidence represents

the percentages of genes of a given category (represented by the antecedent) that show the expression patterns or gene annotations appearing in the consequent of the rule.

In the case of DeRisi results and many biological experiments, the confidence is a crucial parameter. If only a small set of genes are annotated into a very specific category, the support value of the rules containing this annotation will be quite low. However, if these rules have a high confidence value, they reveal that this specific biological property is highly associated with the expression pattern that appears in the consequent.

In this section, we present the most marked discovered biological association extracted from TABLES 7.10 and 7.12. The lector can find a more extended discussion concerning the biological significance issue of TABLES 7.9 and 7.11 in their publication [61].

As noted in the TABLES 7.10 and 7.12, extracted rules from DeRisi data set only revealed marked alterations at biological conditions C6 and C7, which is in agreement with the curve of glucose concentration reported in the original paper [90].

The rule r3 of TABLE 7.10 shows that more than 70% of all genes annotated as "ribosome" were under-expressed at time point 6, while the rule r2 of the same TABLE shows that more than 90% of the genes annotated within this category were under-expressed at time point 7. This increase in confidence (and also in support) value from time point 6 to 7 indicates that an increasing number of ribosomal genes were significantly under-expressed. The association of this pathway and the under-expression pattern of some genes involved in pathways related to protein and nucleic acid biosynthesis is in agreement with the observation that yeast cells enter into a non proliferating stationary phase in response to glucose depletion [192].

The rule r8 of TABLE 7.10 shows that 60% of genes involved in "TCA cycle" were mainly over-expressed at time points 6 and 7, when more than 76% of all genes annotated as "TCA cycle" were over-expressed (as seen in r5 of TABLE 7.10).

Additionally, extracted rules as r11 and r14 show that the genes involved in "glyoxylate and dicarboxylate metabolism" and "propanoate metabolism" were also mainly over-expressed at time point 7 which reflects the main metabolic changes associated to the diauxic shift in yeast, manually identified by DeRisi [90].

In rules r17 and r18 of TABLE 7.11 states that the transcriptional regulators FHL1, GAT3, SMP1, PDR1 and YAP5 are related to under-expression in biological conditions C6 and C7 and with the metabolic pathway "ribosome". This reveals that promoter regions were bound by this set of transcriptional regulators and, in addition, they were highly repressed in response to glucose depletion.

The rule that can be obtained from the first generator of TABLE 7.12, that is the rule GAT3, FHL1, RAP1, Ribosome $\Rightarrow C7[\downarrow]$ shows that ribosomal genes whose promoter regions were bound by RAP1, FHL1 and GAT3 gene products presented an inhibition pattern in response to nutrient starvation. These associations were extracted with relatively high support values and suggest a connection among GAT3, FHL1 and RAP1 and the decrease in ribosomal gene transcription in response to glucose depletion. The connection among RAP1 and ribosomal gene transcription is well-known [216].

In rule r21 of TABLE 7.11 we can note that the under-expressed genes whose promoters regions were bound by the products of HAP2, HAP3, HAP4 were mainly involved in "oxidative phosphorylation", the biological process in which cytochrome correlated genes are involved (as stated by DeRisi in [90]).

The rule r5 of TABLE 7.11 showed that 100% of genes whose promoter regions were bound by the SFP1 gene product were annotated as "ribosome" were inhibited in response to nutrient starvation. In a recently published work, Marion et al. [195] have demonstrated that this transcription factor released from ribosomal protein gene promoters and ribosomal protein gene transcription is down-regulated in response to changes in nutrient availability.

7.6 Results of Eisen Data Set

In order to evaluate and obtain meaningful relationships between gene annotations and gene expression profiles with GENMINER, it was applied to the most famous study in bioinformatics field: Eisen data set [107]. This data set contains the expression measures of 2465 yeast genes under 80 biological conditions extracted from a collection of four independent microarray studies about the *Saccharomyces Cerevisiae* during several biological processes: cell cycle experiments (Spellman et al.[290]), sporulation experiments (Chu et al. [71]), temperature shock experiments (Eisen et al. [107]) and diauxic shift (DeRisi et al.[90].

Spellman et al. have studied three cell cycle yeast processes: alpha factor arrest and release (18 time points), cell size selection and release selected by elutriation (14 time points) and cdc15 arrest and release (15 time points) as seen in lines 1-3 of TABLE 7.13 respectively. More details on these experiments can be found in Spellman et al. [290].

Chu et al. have studied the transcriptional responses during the biological process of sporulation which is the production and release of spores. They have varied the sporulation process over three different experimental conditions: sporulation (6 time points), sporulation 5 hour timepoint (3 time points) and the effect measures of transcription factor Ndt80 knockout on sporulation (2 measures) as seen in lines 4-6 of TABLE 7.13 respectively. More details on these experiments can be found in Spellman et al. [71].

Eisen et al. measured the responses of yeast while growing at different temperatures and conditions: yeast cells exposed to heat shock $25 - 37°C$ (6 time points), yeast cells exposed to dithiothrietol (DTT) and a temperature of $25°C$ or DTT shock (4 time points), and yeast cells exposed to cold shock (4 time points) as seen in lines 7-9 of TABLE 7.13 respectively. More details can be found in Eisen et al. [107].

The last line of TABLE 7.13 corresponds to the diauxic shift experiment of DeRisi et al. [90] (fully explained in the last section). More details on these experiments can be found in DeRisi et al. [90]. TABLE 7.13 shows the composition of the 79 biological conditions contained in the whole Eisen data set as explained before.

This section is organized as follows, first we present the Eisen data set selection and pre-treatment. Then, we present the selected biological results obtained with GENMINER,

Biological Conditions	Biological Process	Experiment	References
C00-C17	Cell-cycle	Alpha factor arrest and release	Spellman et al, 1998
C18-C31	Cell-cycle	Elutriation	Spellman et al, 1999
C32-C46	Cell-cycle	cdc15 arrest and release	Spellman et al, 2000
C47-C52	Sporulation	-	Chu et al., 1998
C53-C55	Sporulation	5 hour timepoint	Chu et al., 1999
C56-C57	Sporulation	In ndt80 knockout	Chu et al., 2000
C58-C63	Cell Response	Heat Shock 25-37C	Eisen et al., 1998
C64-C67	Cell Response	DTT Shock	Eisen et al., 1999
C68-C71	Cell Response	Cold Shock	Eisen et al., 2000
C72-C78	Diauxic Shift	-	DeRisi, et al., 1997

TABLE 7.13: Eisen data set during several biological process: cell cycle experiments, sporulation experiments, temperature shock experiments and diauxic shift.

emphasizing the different sources of biological information used in the extraction process. Finally, we underline the principal biological interpretations obtained with GENMINER.

7.6.1 Eisen data set selection and pretreatment

In order to show the advantages presented by GENMINER algorithm, we have exploited several sources of biological information (for gene annotations) as well as the NORDI discretization algorithm (for gene expression measures treatment).

Gene expression measures

The microarray technique used is spotted cDNA chips obtained by two color fluorochromes with distinct emission spectra Cy3 and Cy5 (this technique was explained in section 1.2.1). The Eisen data set contains the expression levels of 2465 ORF's (opening reading frames) of the yeast (an entirely sequenced organism) for 79 biological conditions (as shown in TABLE 7.13).

Data treatment and discretization

The data set was pre-treated by taking the log_2 ratios (to consider cellular inductions and repressions in a numerically equal way) and applying the imputation algorithm of k-nearest neighbors [184] in order to treat the missing values (1.9% of the total).

The studied biological processes of the yeast (sporulation, diauxic shift, heat shock etc.) are independent from each other, and they are supposed to be normally distributed (as explained in section 7.4). In this manner, all the hypotheses of NORDI discretization algorithm are accomplished (see section 7.4.2). Let us assume that the gene expression measures are presented as a matrix: X with n genes (2465 rows) and m biological conditions (79 columns),

where $X_{i,j}$ is the expression measure of gene i in biological condition j. $X_{i,j} \in \mathbb{R}$. Applying the *NORDI* discretization algorithm explained in section 7.4.2, we obtain the following three discretization intervals for every $X_{i,j} \in X$:

$$\begin{aligned} X_{i,j} &\geq 1 \Longrightarrow X_{i,j} \quad \text{over-expressed or } \uparrow \\ X_{i,j} &\leq -1 \Longrightarrow X_{i,j} \quad \text{under-expressed or } \downarrow \\ -1 &< X_{i,j} > 1 \Longrightarrow X_{i,j} \quad \text{Unexpressed.} \end{aligned} \tag{7.4}$$

Gene annotations

We have used the *Saccharomyces cerevisiae* Database (SGD) nomenclature for naming the yeast genes (SGD was explained in section 3.3). All yeast genes were annotated using seven sources of biological information:

- The semantic source of information Gene Ontology, containing annotations from biological processes, molecular functions and cellular annotations (explained in section 3.7).
- The bibliographic source of information from SGD's MANUALLY CURATED PAPERS of PubMed/Medline (see more details in section 3.5). The bibliographic source of information
- The gene/protein related specific repository BioGRID [293] containing the information about physical and genetic interactions. (see more details in section 3.6).
- The gene/protein related specific database KEGG [161] containing the metabolic pathways in which each gene is involved (see section 3.6 for more details).
- The phenotype information of given yeast genes extracted from SGD's FILE.
- The information of transcriptional regulators that bind to promoter regions, these data were reported in the article of Lee et al. [176]. This information was used to annotated yeast genes whose promoter regions were bound by at least one transcriptor regulator (with a p-value threshold of 0.0005).

All gene annotations were taken as boolean variables, i.e. $i \in \{0, 1\}$, indicating if an annotation pertains, $i = 1$, or not, $i = 0$, to a given gene (similarly as stated in TABLE 7.6).

Data mining context

The Eisen data set and their corresponding annotations were transformed into a data mining context, $\mathcal{DGAL} = (\mathcal{T}, \mathcal{I}, \mathcal{R})$ where the transactions $t \in \mathcal{T}$ are represented by the yeast genes, and the items $i \in \mathcal{I}$ are each one of the gene characteristics, i.e. discretized gene expression measures and gene annotations (as seen in TABLE 7.6).

The resulting data mining context $\mathcal{DGAL} = (\mathcal{T}, \mathcal{I}, \mathcal{R})$ contains 2465 transactions or genes measured over 79 biological conditions (each value discretized by the NORDI algorithm). The obtained gene annotations over the 2465 genes were: 24 Gene Ontology annotations, 15 KEGG annotations, 25 transcriptional regulators, 20 protein interactions, 14 phenotypes and

20 pubmed keywords. Thus, the resulting matrix of genes measures and annotations is of 2465 columns and 197 lines.

7.6.2 Eisen results in mining $\mathcal{DGAL}$ context

In order to complete the analysis, we explore the full potential of GENMINER method integrating gene expression profiles with multiple types of biological information: transcriptional regulator, metabolic pathways, GO annotations, interactions between proteins, phenotype information and the links to the PUBMED articles. Furthermore, we have taken into account all possible kinds of rules having either gene annotations or gene expression measures indifferently as antecedent or consequent.

The results presented here correspond to extracted rules using GENMINER algorithm with *minsupp=.41%* (at least 10 transactions) and *minconf=50%* for mining $\mathcal{DGAL}$ context. We have selected and described meaningful biological rules, emphasizing the form of the rule in order to show the potentials of our algorithm.

Exploring associations of the type Gene annotations $\Longrightarrow$ Gene expression patterns

In the case of the yeast diauxic shift process (biolocal conditions 72-78 in Table 7.13) we have found all the rules presented in Table 7.10 for data mining context $\mathcal{DGK}$ and Table 7.11 for data mining context $\mathcal{DGKP}$. The difference in these rules are the support and confidence measures, because the Eisen data contain only a selection of 2465 genes of the 6199 genes used in DeRisi data. However, the biological interpretation of the results given in the DeRisi Results section is also valid here.

Exploring associations of the type Gene expression patterns $\Longrightarrow$ Gene annotations

Here, we analyze association rules of the kind: gene expression Patterns $\Longrightarrow$ gene Annotations, that means when a group of genes is over-expressed or under-expressed in a set of biological conditions, these genes are likely to have the correspondent gene annotations. Selected association rules extracted for mining $\mathcal{DGAL}$ with *minsupp=.41%* (at least 10 transactions) and *minconf=50%* are presented in Table 7.14-7.18. The antecedent of the rule contains the over-expression or under-expression in a set of biological conditions and the consequent is composed by their correspondent gene annotations (first three columns of Tables 7.14-7.18). The support is given in terms of number of transactions and the confidence is a percentage (columns 4 and 5 of Tables 7.14-7.18). The resulting rules are presented in five different Tables corresponding to five different biological processES: elutriation on Table 7.14 (see C18-C31 on Table 7.13), sporulation Table 7.15 (see C47-C57 on Table 7.13), heat shock on Table 7.16 (see C58-C63 on Table 7.13), cold shock on Table 7.17 (see C68-C71 on Table 7.13) and diauxic shift on Table 7.18 (see C72-C78 on Table 7.13).

Concerning the elutriation process (C18-C31), we have found an over-expression of the responsible genes of the protein synthesis (GO:0006412 BP translation), an under-expression of the genes responsible of the cellular organization (GO:0006996 BP organelle organization

and biogenesis), an under-expression of the genes responsible for the ribosomal organization (GO:0042254 BP ribosome biogenesis and assembly) and an over-expression of the genes which play an role in the response to stress (GO:0006950 BP response to stress). See TABLE 7.14.

Rule	Antecedent		Consequent	Supp. #	Conf. %
1	C22[↑], C24[↑], C23[↑]	⇒	Translation (protein formation)	26	87
2	C21[↑]	⇒	Translation (protein formation)	39	52
3	C21[↑], C22[↑], C24[↑]	⇒	Translation (protein formation)	18	86
4	C25[↑], C23[↑]	⇒	Translation (protein formation)	21	68
5	C19[↓]	⇒	Organelle organization and biogenesis	12	55

TABLE 7.14: Selected elutriation rules extracted from mining $\mathcal{DGAL}$ context with *minsupp=.41%* (at least 10 transactions) and *minconf=50%* with GENMINER algorithm

In the sporulation experiments, we note an over-expression of the genes intervening in the sugar formation (GO:0005975 BP: carbohydrate metabolic process) and the protein synthesis (GO:0006412 BP translation). This claim is confirmed by the under-expression of the genes belonging to the process of sugar transformation into energy (sce00010 Glycolysis / Gluconeogenesis pathway). See TABLE 7.15.

Rule	Antecedent		Consequent	Supp. #	Conf. %
1	C50[↓], C51[↓], C52[↓]	⇒	Carbohydrate metabolic process	12	52
2	C49[↓], C50[↓], C52[↓]	⇒	Carbohydrate metabolic process	12	55
3	C49[↓]	⇒	Translation (protein formation)	42	52
4	C48[↓], C49[↓]	⇒	Translation (protein formation)	27	57
5	C49[↑], C51[↑], C52[↑]	⇒	Organelle organization and biogenesis	18	51
6	C49[↓], C50[↓], C51[↓]	⇒	Glycolysis / Gluconeogenes	13	52
7	C55[↑]	⇒	Translation	50	54

TABLE 7.15: Selected sporulation rules extracted from mining $\mathcal{DGAL}$ context with *minsupp=.41%* (at least 10 transactions) and *minconf=50%* with GENMINER algorithm

A notable exception is to be noticed at the last timepoint of the sporulation (C55) process where an over-expression of the genes is playing a role in the protein synthesis (GO:0006412 BP translation). We can also remark an over-expression of the genes responsi-

ble for the cellular organization (GO:0006996 BP organelle organization and biogenesis). See TABLE 7.15.

In the Heat Shock(C58-C63) process, we note an under-expression of the genes responsible for the protein synthesis (GO:0006412 BP translation), an under-expression of the genes responsible for the cellular organization (GO:0006996 BP organelle organization and biogenesis), an under-expression of the genes responsible for the ribosomal organization (GO:0042254 BP ribosome biogenesis and assembly) and an over-expression of the genes related to stress response (GO:0006950 BP response to stress). See TABLE 7.16.

Rule	Antecedent		Consequent	Supp. #	Conf. %
1	C63[↓], C62[↓], C60[↓]	⇒	Translation (protein formation)	16	80
2	C61[↓], C62[↓], C60[↓]	⇒	Translation (protein formation)	35	88
3	C59[↑], C60[↑], C62[↑]	⇒	Response to stress	15	52
4	C59[↓]	⇒	Organelle organization and biogenesis	41	69
5	C59[↓]	⇒	Ribosome biogenesis and assembly	39	66

TABLE 7.16: Selected heat shock rules extracted from mining $\mathcal{DGAL}$ context with *minsupp=.41%* (at least 10 transactions) and *minconf=50%* with GENMINER algorithm

In the Cold Shock experiment (C68-C71), there is an under-expression of the genes responsible for the protein synthesis (GO:0006412 BP translation).See TABLE 7.17.

Rule	Antecedent		Consequent	Supp. #	Conf. %
1	C72[↓], C70[↓]	⇒	Translation (protein formation)	16	84
2	C69[↓], C71[↓]	⇒	Translation (protein formation)	14	67

TABLE 7.17: Selected cold shock rules extracted from mining $\mathcal{DGAL}$ context with *minsupp=.41%* (at least 10 transactions) and *minconf=50%* with GENMINER algorithm

Concerning the diauxic shift process (C72-C78), there is an over-expression of the genes responsible for the energy generation (GO:0006091 BP generation of precursor metabolites and energy) and an under-expression of the genes responsible for the protein synthesis (GO:0006412 BP translation). As seen in the fourth block of TABLE 7.18.

Exploring associations of the type Gene annotations $\Longrightarrow$ Gene annotations

Independently of the gene expression levels, it is also possible to highlight existent relationships among gene annotations. Selected *annotation* $\Longrightarrow$ *annotation* association rules extracted for

Rule	Antecedent		Consequent	Supp. #	Conf. %
1	C76[↑], C78[↑]	⇒	Generation of precursor metabolites and energy	24	52
2	C77[↓], C78[↓]	⇒	Translation (protein formation)	21	66

TABLE 7.18: Selected diauxic shift rules extracted from mining $\mathcal{DGAL}$ context with *minsupp=.41%* (at least 10 transactions) and *minconf=50%* with GENMINER algorithm

mining $\mathcal{DGAL}$ with *minsupp=.41%* (at least 10 transactions) and *minconf=50%* are presented in TABLE 7.19. The antecedent and the consequent of the rule contain gene annotations issued from one of the seven sources of biological information explained before (first 3 columns of TABLE 7.19). The support is given by the number of concerned genes, and the confidence is given as a percentage (columns 4 and 5 of TABLE 7.19).

Rule	Antecedent		Consequent	Supp. #	Conf. %
1	PATHWAY=sce04111 (Cell cycle)	⇒	GO:0007049 (Cell cycle)	59	69
2	PATHWAY=sce00190 (Prune metabolism)	⇒	GO:0005737 (Cytoplasm)	52	96
3	PROMOTER=FHL1	⇒	PROMOTER=RAP1	114	86
4	PROMOTER=RAP1	⇒	PROMOTER=FHL1	114	61
5	PROMOTER=RAP1, PROMOTER=FHL1	⇒	GO:0005737, GO:0006412, GO:0005840	93	82
6	PUBMED=16155567	⇒	PHENOTYPE=inviable	96	94
7	GO:0005737, GO:0045333	⇒	GO:0006091	54	100
8	GO:0016192	⇒	GO:0006810	167	100
9	GO:0005739	⇒	GO:0005737	503	100
8	GO:0005740	⇒	GO:0005737, GO:0005739	167	100

TABLE 7.19: Selected Annotation $\Longrightarrow$ Annotation rules extracted from mining $\mathcal{DGAL}$ context with *minsupp=.41%* (at least 10 transactions) and *minconf=50%* with GENMINER algorithm

We identify an association between annotations carried out by groups of independent experts who state as a Close concept the KEGG term sce04111 (Cell cycle) and the Gene Ontology term GO:0007049 (cell cycle) with corresponding support of 59 and confidence of 69% (r1 of TABLE 7.19). We have also identified less obvious associations, but nevertheless several strong ones like the relationship between the KEGG term sce00190 (purine metabolism) and the GO term GO:0005737 (cytoplasm) with a support of 52 and a confidence of 96% (7.10 of TABLE 7.19).

Concerning the transcriptional regulators, extracted rules enable to state strong relationship between promoters: $FHL1$ and $RAP1$. For example the rule $FHL1 \Longrightarrow RAP1$ with

a high support of 14 and a confidence of 0,86 (r3 of TABLE 7.19) and the rule $RAP1 \Longrightarrow FHL1$ with the same support and a confidence of 0,61 (r4 of TABLE 7.19). One can deduce from these rules that the genes activated by $FHL1$ are also activated by $RAP1$ The reverse is less true since there is a considerable proportion of genes activated by $RAP1$ and not by $FHL1$. This information is already known and was described in many articles. For example Zhao et al. [338] state that "$RAP1$ binding is essential for the recruitment of $FHL1$", and they explain the association between them in the following phrase: "based on recent work, a simple model for the transcription of RP (ribosomal proteins) genes is that $RAP1$ recruits $FHL1$, which in turn recruits the transcriptional activator $IFH1$". The last phrase confirms the results obtained in rule r5 of TABLE 7.19 where the promoters $RAP1$ and $FHL1$ are closely related to the Gene Ontology terms GO:0005737 (cytoplasm) and GO:0006412 (translation) and GO:0005840 (ribosome). The two last terms are closely related to protein synthesis, and the cytoplasm activity shows us the transcriptional cellular activity while $RAP1$ and $FHL1$ transcription factors are activated.

We have also detected rules which relate scientific articles with phenotypes as the rule r6 of TABLE 7.19 where PUBMED:16155567 $\Longrightarrow$ inviable with a support of 96 genes and a confidence of 94%. The PubMed article 1615567 'The synthetic genetic interaction spectrum of essential genes' [86] presents a review of the essential yeast genes. These genes are for the majority annotated as inviable, i.e. the organism does not survive when the correspondent gene is removed. It could be interesting to examine what about the few genes quoted in the article which are not annotated inviable. This could be a lapse of memory in this article [86].

When the analyzed data represent a hierarchy, it is possible, by examining the obtained rules, to reconstitute the original hierarchy. For example, the rule r9 of TABLE 7.19, i.e. GO:0005739 => GO:0005737 with supp=503 conf=100%, means that there are 503 genes annotated by GO:0005739 and also by GO:0005737. GO: 0005739 is a sub-term of GO: 0005737 or it represents the same concept exactly. In Eisen data set, we have more than 1500 genes annotated with GO: 0005737. Therefore, GO: 0005739 is a sub-term or child of the parental term GO: 0005737.

The rule r10 of TABLE 7.19, i.e. GO:0005740 => GO:0005737, GO: 0005739 with supp=167 conf=100%, means that the terms annotated GO:0005740 are also annotated by GO:0005737 and GO: 0005739. Thus, we continue the unfolding of the hierarchy GO:0005740 is a sub-term of GO:0005739 containing 167 genes.

7.7 Discussion

In this work we present the GENMINER algorithm, an association rules discovery approach that fulfills the requirements of data obtained from gene expression technologies. Our approach integrates at once gene expression profiles with gene annotations to discover intrinsic associations among both data sources based on frequent patterns. That means it is a co-clustering and bi-clustering approach integrating gene expression patterns and gene annotations at once and finding patterns of co-expressed genes in subsets of biological conditions. Opposite to the majority of gene expression interpretation approaches, defined as *expression-*

based and *knowledge-based* (see chapter 5), in which biological information and gene expression profiles are incorporated in an independent manner, our approach integrates both data sources in a single framework. More advantages of co-clustering approaches have been discussed in the discussion part of chapter 5.

GENMINER is an original approach that takes advantage of the Close association rules extraction algorithm reducing considerably the execution time for rule extraction and generating a minimal basis for association rules extraction, enhancing the expert interpretation of the extracted rules. As we have shown in TABLE 7.12, the use of the *min-max basis* for rule generation allows the expert to know more about the inherent characteristics of the desired rule like the minimal rule which synthesizes all the rule information, that means the rule with a minimal antecedent and a maximal consequent. The expert can build next to this minimal rule a set of rules with meaningful biological implications with same support and confidence (as we have shown building the rules of TABLE 7.9 after the generator and closure itemsets in TABLE 7.12).

The analysis of the DeRisi data set has permitted to validate the GENMINER algorithm and to compare it with the ARD association rules algorithm. Concerning the DeRisi data set results obtained with the same parameters used in the ARD algorithm, GENMINER has found all the association rules presented in ARD publication [61] for data mining contexts $\mathcal{DGK}$ and $\mathcal{DGKP}$ with *minsupp=.44%* and *minconf=40%*.

In the DeRisi data set result section we have shown that the redundant filter applied by ARD is not minimal against inclusion. ARD algorithm states "that the rule with the longest antecedent or consequent summarizes all information, and the rest of the rules can be discarded". This assertion is not precise, because the minimal rule is the one with minimal antecedent and maximal consequent. So, the prunning method used in ARD will eliminate the minimal and non-redundant rules. In TABLE 7.12 we have shown that the generator and closure of the rule generates the minimal rule from which we can generate all the rules obtained by the ARD algorithm in TABLE 7.11. Furthermore, the use of the min-max basis for rule generation enhances the expert interpretation and knowledge of the extracted rules (as explained before).

The analysis of the Eisen data set for data mining context $\mathcal{DGAL}$ shows the potential of our method to integrate several heterogeneous sources of information as GO, BioGRID, KEGG, phenotype information, transcriptional regulators information, information of selected articles with gene expression profiles measured over 79 biological conditions. That means a table $\mathcal{T}$ with 2465 objects or genes and 197 attributes. alowing us to evaluate the GENMINER algorithm using several sources of information with a large data set. This table is only an example of the possibilities of our algorithm, which can easily be extended to integrate any kind of gene annotation obtained from any source of biological information (explained in chapter 3). Therefore, the integration of different types of biological information is an essential consideration to fully understand the underlying biological processes. In addition, qualitative variables as gender, tissue, age and so on, contained in the minimal gene expression experimental information could easily be added to the analysis in order to extract association rules among these features and gene expression patterns.

GENMINER constructs association rules containing either gene expression patterns or gene annotations as antecedent or consequent of the extracted rule. In the results obtained with the DeRisi data set in TABLE 7.10 and TABLE 7.12 we have shown that we can extract meaningful association rules of the form Gene Annotations $\Longrightarrow$ Gene expression Patterns as stated in [61]. In the results obtained with Eisen data set we have shown the importance of taking into account all possible combinations among itemsets composed either of gene annotations or gene expression patterns as antecedent or consequent of the rule. In the TABLES 7.14-7.18 we found important biological known statements of the form: Gene Expression Patterns $\Longrightarrow$ Gene Annotations and Gene Annotations. Rules of the form Gene Expression Patterns $\Longrightarrow$ Gene Expression Patterns have been studied before [22], [81], [170], [310], [224], [263] and [124].

The analysis of two famous gene expression data sets DeRisi and Eisen in different data mining contexts: $\mathcal{DGK}$, $\mathcal{DGKPG}$ and $\mathcal{DGAL}$ has proven the capacity of GENMINER to extract meaningful associations among gene expression profiles and gene annotations (as seen in TABLES: 7.10, 7.12, 7.14-7.18). Several interpretations of these tables can be found in the respective result sections 7.6 and 7.7.

GENMINER uses the Close algorithm for extracting rules, which is specifically designed for highly correlated data, that is the case of gene expression technology data where several genes groups are expressed together in different biological conditions. In comparison with the Apriori algorithm, the calculation time of the Close rule extraction algorithm is considerably smaller in case of correlated data. More information about this issue - comparison between Apriori and *Close* algorithm performances - can be found in [229]. The smaller calculation time shown by the GENMINER algorithm in comparison with the ARD algorithm allows GENMINER to deal with very large and correlated data sets, which is the case of the data mining context $\mathcal{DGAL}$ in the Eisen data set. Furthermore, it allows to use several sources of annotation, including the semantic source of annotation GO that contains thousands of gene annotations. In TABLE 7.12 we show the results of mining data context $\mathcal{DGKPG}$, which contains thousands of gene ontology attributes.

GENMINER has implemented a new discretization algorithm, NORDI, specially designed for discretizing data issued from gene expression technologies in the case of independent biological conditions. We have obtained satisfactory results using NORDI algorithm in the case of Eisen data (as showed in TABLES 7.14-7.18). However, the discretization issue is a delicate step when using supervised methods as ARD. We propose the use of several discretization scenarios (as the ones proposed in section 7.4), and then analyze the pertinence of the obtained results against the expected results to validate the discretization algorithm. In a recent work, Pan et al. [225] have suggested that "the robustness of biological conclusions made by using microarray analysis should be routinely assessed by examining the validity of the conclusions by using a range of threshold parameters issued from different discretization algorithms". Unfortunately, there do not exist any discretization algorithms specially designed for time process data, which integrate the time variable without an important loss of the temporal information.

Another delicate issue in association rules discovery is the threshold for selecting significative rules. Support and confidence are the most common measures related to a rule, in many cases, the only ones used to point out the relevance of it. However, it is important to note that sometimes both of these measures are high, indicating a rule which could be good, and yet still produce a association that is not useful. In other words, associations among uncorrelated elements can be generated using this support-confidence "framework" [157]. In the case of significant rules proposed in this work, perhaps confidence is the most significant value from a biological point of view. For example, if only a small set of genes are annotated into a very specific category, the support value of the rules containing this annotation will be quite low. Nevertheless, if these rules have a high confidence value, they reveal that this specific biological property is highly associated with an expression pattern of another gene annotation that appears in the consequent. GENMINER uses the support-confidence framework, providing the lift or improvement of the rule 7.2.1 which measures the correlation or independence between consequent and antecedent of the rule for avoiding the selection of association among uncorrelated elements.

7.8 Outlook and Conclusion

Outlook

One known drawback of association rules discovery is the number of generated rules that is generally very high, even if large values of *minsupp* and *minconf* are used. This huge amount of information is difficult to process manually, and it requires a conscious examination of the generated rules to extract those that are more interesting than others for a particular application goal. GENMINER proposes the generation of non-redundant rules by the construction of a *min-max basis* for rule generation. Even if we generate only the minimal rules against inclusion, the number of rules can still be very high for expert interpretation. There is a real need for a post-treatment of the generated rules in order to help the expert to obtain meaningful biological associations. We have distinguished five different aspects concerning rule post-treatment:

- Code a program that organizes the information taking into account useful biological criterions as:
  - Form of expected rule, i.e.: Gene Annotations $\Longrightarrow$ Gene expression Patterns, Gene expression Patterns $\Longrightarrow$ Gene Annotations, Gene Annotations $\Longrightarrow$ Gene Annotations, etc.
  - Special type of gene annotation contained in the rule: semantic sources of information, bibliographic sources, gene-protein related specific sources of information etc.
  - Special subset of biological conditions contained in the rule.
- Code a program that generates all the possibles rules from a given generator and closure.

- Develop a program that could generate association rules from any of the sources of biological knowledge and could search this existing rule over all obtained rules using GENMINER.
- Develop an interactive semi-automatic program that allows the expert to find "interesting" rules taking into account his knowledge and the existing knowledge in the state-of-the-art.
- Create an automatic tool for building associations between genes and relative information from sources of biological knowledge as articles (text mining goal).

Concerning the threshold issue for selecting significative rules, GENMINER uses the support-confidence framework providing also the lift of the rule 7.2.1 in order to avoid the selection of association among uncorrelated elements. Although support and improvement values provide information about the association between the antecedent and the consequent parts of the rule, they do not inform about their statistical significance [55]. The statistical significance of an association rule could be evaluated using one of the statistical significance tests explained in section 2.2. For this purpose, we plan to use the one-tailed hypergeometric test used to find significant co-expressed and co-annotated groups in the CGGA algorithm (explained in section 6.3.1).

Conclusion

We have developed the GENMINER methodology which integrates several sources of biological information with gene expression patterns in an automatic way. This approach is based on an association rules discovery technique, and it enables the knowledge discovery via the interpretation of associations between expression data issued from a gene expression technology and gene annotations issued from any of the sources of biological information. The presented results obtained with GENMINER show that it is a promising tool for finding meaningful relationships between gene expression patterns and gene annotations.

The analysis of the DeRisi data set has permitted to validate the GENMINER algorithm and to compare it with the ARD association rules algorithm. Concerning the DeRisi data set results obtained with the same parameters used in ARD algorithm, GENMINER has found all the association rules presented in ARD publication [61] for data mining contexts $\mathcal{DGK}$ and $\mathcal{DGKP}$ with *minsupp=.44%* and *minconf=40%*.

Chapter 8

GENETREE: GENE-Integrated Analysis using a Decision Tree Algorithm

In chapters 6 and 7 we have proposed two novel class discovery mining algorithms: CGGA for identifying co-expressed and co-annotated groups of genes, and GENMINER for finding coherent gene expression patterns within gene groups. In this chapter we deal with the class prediction issue within the fifth step of the gene expression data analysis procedure (see FIG. 2.1). Class prediction refers to the assignment of biological samples or conditions to known classes such as disease-type, drug-response, toxicity-reponse etc. Among the main applications of class prediction we can mention the classification of tumors for medical diagnosis and treatment of cancer [133, 7, 166] and the classification of new molecules as toxic or not toxic via the gene expression patterns they exhibit [87, 315].

In contrast with class discovery problem (as seen in chapter 5-7) where the classes are unknown and need to be discovered from the data; in predictive methods, the classes are predefined and the task is to understand the basis of their classification from a set of class-labeled objects. This information is used to build a classifier which will then be used to predict the class of unlabeled objects.

In the data mining field, class prediction is often recognized as a typical "supervised learning problem or classification" and class discovery an "unsupervised learning problem or clustering". In many situations, for instance deciphering complex diseases as cancer, the two data mining problems are related, as the classes which are discovered from clustering methods are often used later on in a predictive method setting. Here, we focus on supervised learning, and we use the simpler term "classification".

In this chapter, we propose a decision tree based algorithm: GENE-integrated analysis for biological sample prediction using decision TREEs (GENETREE) for solving the class prediction problem discussed above.

We start this chapter with a brief introduction concerning the prediction challenges in bioinformatics, the most used supervised methodologies to tackle this issue. Then, we define the predictive goal in gene expression technologies and explain the process for building a predictive model. Since discretization is a key problem for predictive variables, we present in section 8.3 several discretization techniques used in bioinformatics emphasizing in our novel discretization algorithm: NORDI specially designed for gene expression data. In section 8.4 we explain decision tree basics, making emphasis on Quinlan's C5.0 algorithm [247]. In section

8.5 we explain the GENETREE algorithm. This chapter ends with a brief discussion and it gives an outlook for future research.

8.1 Introduction

As seen in previous chapters, one of the most current goals in gene expression technologies is the classification of biological samples using gene expression data. By allowing the monitoring of expression levels in cells for thousands of genes simultaneously, gene expression technologies may lead to a more complete understanding in several applications as: distinguish among known tumor classes, predict clinical outcomes such as survival or response to treatment, and identify previously unrecognized and clinically significant subclasses of tumors [101]. Several studies had shown the growing research interest on these important gene expression technologies applications [7, 166, 242].

More generally, classification of biological samples is a class prediction problem to which supervised learning techniques are ideally suited. Assuming a number of training data corresponding to known biological conditions labeled as "reference=healthy" or "test=disease" and the associated gene expression measures, then they can predict the class (healthy or disease) of a new patient on the basis solely of their gene expression levels. This asseveration is partially true, as stated in Golub et al. [133] who concludes that: "The leukemia diagnosis remains imperfect and could benefit from a battery of expression-based predictors for various cancers". Actually, by applying the existent gene expression technology and known analytic tools, they obtain predictive genes which can distinguish at some confidence level among several classes in a given experiment, but they can not differentiate the same classes against other similar classes. For instance, in the case of leukemia, Golub can not differentiate in an efficient way the cases of leukemia, because the predictive genes that he had found could be expressed in many other cancers. One solution to this prediction problem can be the integration of gene annotations within the supervised learning algorithm. Gene annotations can be integrated from any of the six biological sources of information: MIAME, molecular databases, semantic sources, gene expression databases, bibliographic databases or gene/protein related specific sources (explained in chapter 3).

Since the beginning of gene expression technologies, a variety of supervised learning algorithms have been used to solve the prediction problem specially for disease-type applications. These algorithms take into account only gene expression profiles without integrating any other source of knowledge within the algorithm. Among the most remarkable supervised methods we can list:

- A linear discriminant analysis (LDA) in Dudoit [101] for predicting cancer tumors and in Hakak [137] for predicting schizophrenia types.
- A modified K-nearest neighbor (KNN) method in Pomeory [242] to predict embryonal tumors.
- A Support Vector Machine (SVM) in Ramaswamy [250] for classifying tumors and Furey [119] in predicting organ classes
- A weighted voting technique in Golub [133] to predict leukemia classes.

- Decision trees in Zhang [335] for tumor prediction and Ramanathan [249] for disease-type prediction.

These methods are useful for predicting the studied class or disease-type at a certain degree of effectiveness. However, the actual use of predictive algorithms in gene expression technology field present several weaknesses :

- Error in prediction because the sample classification could be produced by a set of generic expression-based predictors concerning several classes as related diseases, tumors or toxics according to the studied case.
- Error estimators are biased, thus error estimator procedures should be applied independently of the gene selection process, and not as part of it as it is commonly done [102].
- A low number of biological samples tends to overfit the solutions. This is produce by the dimensionality effect of gene expression data where the number of objects or sample experiments is very low (tens of biological conditions) and the number of attributes is extremely high (thousands of genes).
- Lack of optimization techniques on data learning parameters in past data prediction studies. Thus, the classifier can not be robust enough to treat similar gene expression data sets [33].
- Lack of biological knowledge as an inherent part of the classifier building procedure.
- Lack of a discretization algorithm specially adapted to gene expression data.

The use of supervised algorithms for solving prediction problems in gene expression technologies is a relatively new field compared to their use in other domains. It is necessary to include the best available tools of matching learning methods in gene expression technology to close the gap between machine learning field and bioinformatics [33]. The wide research problem of the integration of biological knowledge (as the available gene annotations) in any supervised algorithm remains open.

In order to tackle several of the drawbacks cited before, we propose a GENE-integrated analysis for biological sample prediction using decision TREEs (GENETREE). This algorithm takes advantage of the well known decision tree algorithms ID3, C4.5 and C5.0 proposed by Quinlan [247] and it extends the entropy splitting criterion to more complex one which takes into account several sources of gene annotations.

We have chosen the decision tree supervised methodology because it can be easily interpreted and it is able to model fairly complex functions [283]. However, a known drawback of this methodology is that they are prone to overfitting* on training samples. This problem may be particularly sever with gene expression technologies which contain a large number of genes and a limited number of samples. We proposed "pruning" the tree, i.e., restricting the height of the tree and the applications of bagging* or boosting* techniques to avoid overfitting.

GENETREE automatically integrates gene expression profiles, with gene annotations obtained by genome-wide sources of information such as Gene Ontology and gene/protein

related specific sources such as: KEGG, BioCarta, UniProt, Cancer Gene Census etc.Cancer Genome Anatomy Project (CGAP) (explained in chapter 3).

In addition, we propose two novel algorithms: an outlier sample detection method (see section 4.1.4), and the normal discretization (NORDI) algorithm (section 8.3.2.1) specially designed for gene expression data sets. NORDI enables any number of discretization intervals, used within the decision tree algorithm.

8.2 Prediction in Gene Expression Technologies

Supervised algorithms have found widespread application in bioinformatics [16]. The diverse range of rapidly expanding data produced by modern gene expression technologies has fuelled a need for accurate classification algorithms.

Class prediction methods are techniques specifically designed to classify objects into known classes. For gene expression technologies data, prediction generally refers to the classification of patient samples with unknown class type by using gene expression data concerning known class patient samples. Biological experiments may be: disease-type, drug-response for instance. Thus, the goal may be to predict a diagnostic, offering a new way to distinguish similar-looking diseases [133, 166], or it may be predict clinical outcomes [7, 242].

The process of building a prediction model in bioinformatics is not an easy task, it may need a full seven-step procedure (as seen in Fig. 8.1) of its own which consists in:

1. Class discovery from the full gene expression data set.
2. Data Selection of biological samples and *informative genes*[32] from the full gene expression data set.
3. Data Pretreatment or discretization of continuous variables.
4. Data Partition into training and test sets.
5. Model choice and construction using a supervised learning algorithm.
6. Prediction accuracy evaluation and refining the model [33, 102, 334].
7. Biological Interpretation of the predictive model results.

The Fig. 8.1 shows the whole process for building a predictive model. It starts by defining the full gene expression technology data set containing the gene expression measures of N genes taken over M biological conditions or samples (the gene space is often in thousands of genes and the sample space is in tens of biological conditions). The first stage is optional and it consists in applying class discovery algorithms for clustering the biological samples in K classes (see Fig. 8.1). Once, the biological classes K are known, The second stage is data selection that involves two issues: selection of the pertinent biological conditions or samples, and selection of informative genes. Selection of the pertinent biological conditions must be done to get ride of sample outliers, that might introduce noise to our analysis. Selection

[32] Informative Genes: Group of genes which are strongly correlated with the groups of organisms or individuals. In medical applications, the biological samples are often individuals.

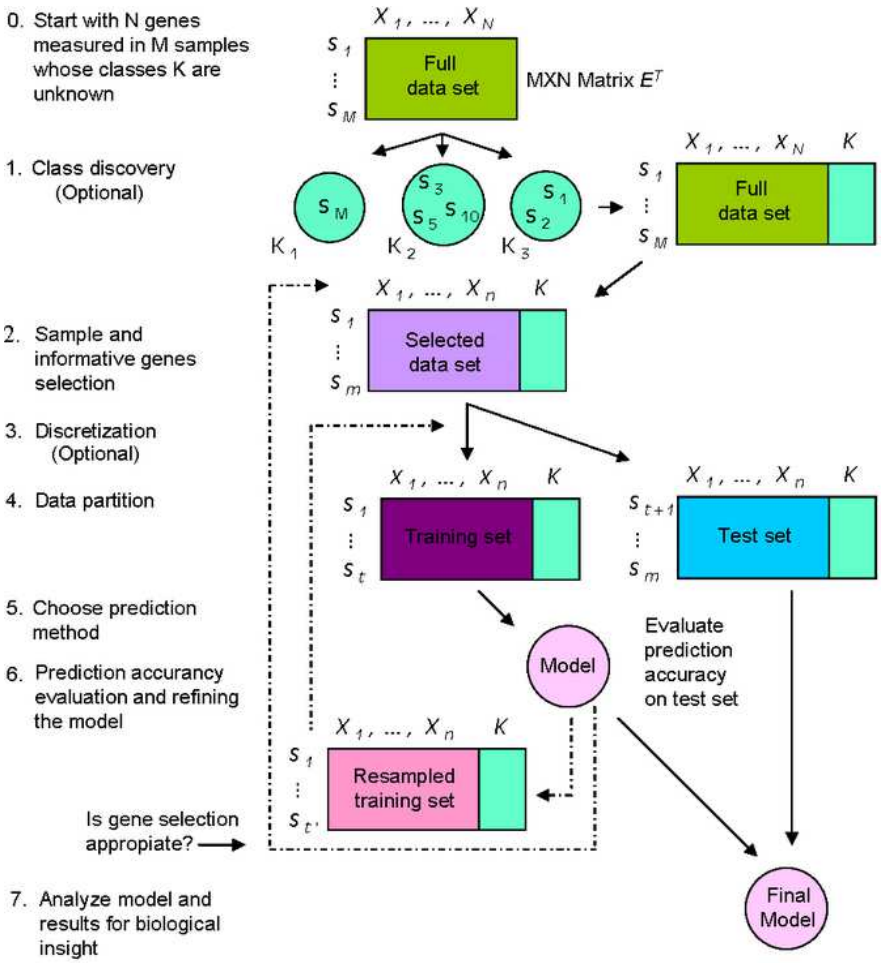

FIG. 8.1: An overview of a full seven-step process for building a predictive model to classify samples in gene expression technologies application.

of informative genes is a delicate step concerning the analysis of differential expressed genes for gene selection. This step is necessary to reduce the *curse of dimensionality* problem in prediction problem. As stated by R. Bellman [24], in the context of prediction problem, it refers to the exponential growth of the hypothesis space with respect to the number of features, in this case of genes. In contrast, the number of samples is already too low in gene expression data, that often the sample selection is not done (even if it is important). The output at this stage is a data set containing the gene expression measures of n selected genes over m biological samples (see FIG. 8.1).

The third step is a necessary condition for using supervised learning techniques, where the continuous variables, as often gene expression measures, have to be discretized. The discretization procedure has to be done just before applying the predictive model because the discretized values introduce some loss of information compared to the original values.

The fourth step concerns the data partition into training set and test set (a seen in FIG. 8.1 where we divide the selected data set into t training samples and $m - t$ test samples). The fifth step consists in choosing the prediction model from the wide-range of supervised algorithms and building it using the training set containing m samples. Once the classifier has been built, the evaluation phase starts, it concerns the classifier estimation error applied on test data set. In FIG. 8.1 we show the resampling of the training set in order to evaluate prediction accuracy. Many prediction methods require tuning some parameters (such as number of genes, number of nearest-neighbors to consider, or the number of decision trees to be built) for refining the prediction model. This phase is illustrated in FIG. 8.1 taking the gene selection parameter example: the number of genes is it appropriate to the prediction model? if not we can return to the gene selection phase and start again the prediction model process. Once the final model is built, the last phase consists in analyzing the predictive model results and interpreting this results for biological insight. In the next sections, we explain more in detail every step and we present several methods to deal with.

First step: class discovery

The first "optional step", class discovery, is a classical "unsupervised learning" problem, in the sense that no-predefined class labels and training samples are provided. In other words, the biological samples are automatically partitioned into several groups exhibiting greater within-group than between-group similarities. Distinguishing classes within gene expression technologies data is of special practical interest, since the identification of phenotypes from samples through traditional pathological or clinical methods is usually slow. Moreover, there may be unknown subtypes of tumors which respond differentially to drug treatment as in [133] or lead to heterogeneous clinical outcomes as in [7].

The clustering techniques discussed in section 2.4 are useful to find the class structure hidden in the sample space, using samples as the data objects and genes as the attributes. Unfortunately, if the entire set of genes (thousands of them) in a gene expression technology data set is adopted as the feature vector, the performance of most conventional clustering algorithms will be degraded by the dimensionality problem.

The goal of class discovery in gene expression technologies is therefore to identify the structure of the phenotypes of samples or sample classes K, and it can be viewed as involving two sub-problems:

1. A K-partition of the samples matching their empirical phenotype distinction.
2. The detection of the informative genes that manifests the phenotype distinction

In fact, these two issues are closely interrelated. Once the class structure has been correctly identified and the samples have been appropriately assigned to classes, the gene selection methods described in data selection step can be used to rank the genes according to their relevance to the classification. Conversely one the informative genes have been identified, we can apply the clustering techniques discussed in section 2.4 to partition the samples into K classes.

Recently, two strategies for class discovery which exploit this dynamic relationship between genes and samples have been developed: CLIFF in Xing et al. [324] and ESPD in Tang et al. [300]. They combine clustering and supervised gene selection processes in an iterative manner. Both are based on the intuition that a valid approximate sample partition can be obtained using the entire set of genes. The approximate partition allows the selection of a moderately-valid gene subset, which will in turn draw the approximate partition closer to the target partition in the next iteration. After several iterations, the sample partition may converge to the true class structure, and the selected genes will be feasible candidates for the set of informative genes.

Second step: data selection

The data selection involves two issues: selection of pertinent biological conditions or samples, and selection of informative genes.

Sample selection

It is critical to take into account biological condition variations and in particular possible *sample outliers* which may introduce noise in the classification procedure [33]. Most approaches for sample selection are unsupervised, which reduce the dimensionality of biological conditions space in gene expression technologies. Principal component analysis* (PCA) [160] is a classical method which projects the original data set along a few directions in an attempt to capture the major variations in the data. However, the results obtained through PCA are often difficult to interpret.

In order to solve this problem, we have developed an algorithm for finding sample outliers among biological conditions using two tools: *Principal Component Analysis* (PCA)* and hierarchical *clustering* approaches (explained in section 2.3.3). This methodology was explained in section 4.1.4.

Gene Selection

The selection of informative genes is an important task [133] that help biomedical researchers to understand disease mechanisms. They can also be used to resolve levels of heterogeneity among cells that are not apparent by eye and to provide a more accurate prognosis and prediction of response to therapy [73]. This step consists in determining which genes to use in the classification procedure. Gene selection is not only necessary to reduce data dimensionality but also to identify those genes that are closely related to tissue types.

In general the approaches to gene selection can be categorized as statistical supervised approaches or unsupervised methods, depending on whether the class labels of the samples are given a priori. A full review of statistical approaches for selecting differentially expressed genes is presented in section 2.2.

Concerning supervised algorithms for gene selection we can mention three novel approaches. The first, *gene-pairs* [38], evaluates how well a pair of genes in combination distinguishes two experimental classes. The second, *virtual genes*, [325], employs the inherent correlation among n genes to predict the class labels. The third approach, [179], considers the combined discriminative power of a subset of genes by integrating a genetic algorithm (the gene selection process) with a KNN algorithm (classification process) to achieve a better set of informative genes.

The PCA unsupervised approach [160] can also be used in the case of gene selection, for reducing the dimensionality of microarray data. Another algorithm is *Gene shaving* [141], a PCA-based approach developed specifically for microarray data. After shaving the gene dimension for several iterations, the algorithm reports a set of informative gene groups.

Third step: data pretreatment

The data pretreatment issue concerns the discretization of the quantitative continuous variables into discrete variables and the determination of categorical classes for either qualitative or quantitative variables.

Specifically in gene expression technologies we have two kinds of data information: gene annotations that are often textual, which can be treated as qualitative variables, and gene expression measures, which are generally quantitative continuous variables. In several supervised learning methodologies, gene expression measures must be converted into discrete values through a discretization process technique.

We have classified gene expression technologies discretization methods in three approaches: *biological* basis, *statistical* basis and *mining* basis. These approaches, as well as the our novel method named normal discretization or NORDI, are explained in section 8.3.2.1.

Fourth step: training and test sets selection

The partition into training and test data sets is done over the pretreated data set and can be done by several algorithms as: re-substitution method, hold-out, cross-validation, bootstrap and non-random method. Let n be the total number of available samples.

The *re-substitution method* consists in using all n samples for building the classifier and used again as test set for estimating its performance.

The *hold-out* method divides the data set containing n samples in two parts: the training set, on which the hypothesis is trained (with s samples), and the hold-out set, on which its performance is measured (with t samples). The sum of the train set and test sets has to be n and the choice of the samples is done randomly. For example a well-known hold out is using 1/3 of the samples as test set.

Cross-Validation Method. It is also called rotation estimation [168], is the statistical technique of partitioning a sample of data by choosing s out of n samples as the training set and estimating its error rate using $n - s$ sample observations (test set). This process is repeated for all distinct choices of s patterns, and finally the average of the error rate, known as $s - fold$ estimate, is computed. The choice may be $s = 1$, which is also know as *leave-one-out* method. This procedure is also known as cross-validation $s - fold$ procedure, for more details see in [33, 168, 37].

Bootstrapping* method. A bootstrap design sample of size n is formed from the n observations by sampling with replacement. The classification rule is designed using the bootstrap sample and tested twice: n observations of the bootstrap design sample are used to obtain the bootstrap re-substitution estimate, E_R^β, and the original design set is used to obtain the bootstrap estimate of conditional error P_n^β. This procedure is repeated r times (typically r lies between 10 and 200). An arithmetic mean of the differences is used to reduce the bias of the re-substitution estimate. More details can be found in [33].

A novel *non-random approach* is suggested by Sprevak et al. [291] when dealing with small data sets (with a few number of samples like gene expression technology case). This method divides the data into two partitions based on similar statistical characteristics. This algorithm reduces the variability of predictive accuracies and provide consistent results across different classification models.

Fifth step: prediction model choice and construction

As with clustering, choosing a prediction method requires selecting from a vast range of techniques. Some of the most straightforward linear and quadratic discriminant methods, LDA and QDA respectively, are very well described by Dudoit et al. in [101]. Related methods include weighted voting in Golub et al. [133], shrunken centroids [305] and compound covariates [142]. A deceptively simple but powerful approach is *k-nearest neighbor prediction*, in which the prediction for a test sample S is the most common class label among the k training samples most similar to S, for more details see [242, 101, 250, 212].

Simple *neural network** [212] may be effective at learning the complex functions often inherent in multi-class diagnostic problems [166]. Also, novel pattern pattern-discovery algorithms such as Splash [58] have shown some success at learning non-linear functions of the input variables. Two other well-studied classes of algorithms are growing interest for gene expression technologies prediction problem: support vector machines (SVMs) and decision tree classifiers.

SVMs are a family of statistical machine-learning methods that have been proposed as a particularly suitable to the dimensions of microarray learning problems [57, 119, 250]. Intuitively, SVM's try to draw a hyperplane in n-dimensional gene-expression space between the training examples from two classes. If no separating hyperplane exists, the samples are mapped into a higher-dimensional space where such a separator does exist. The algorithms minimize the potential over-fitting problems by choosing the separator farthest from the training samples, thus leaving room for generalization. More complex mapping functions provide non-linear mapping into higher dimensional spaces, resulting in a non-linear classifier for the original data. While these models may be difficult to interpret, they are potentially quiet powerful.

Decision tree algorithms classify samples by filtering them through a tree-like structure, testing at each branchpoint (called a *node*) some simple attribute of that sample, such as whether of the expression of $p53$ is greater or lesser than 100, as see in [212]. We can cite some examples of the use of decision trees in gene expression prediction problems as the tumor prediction in Zhang [335] and for disease-type prediction in [249]. Single decision trees* are particularly prone to overfitting*. However, as tree models are easily built, easily understood, and able to model quite complex functions, there are many modified tree-based techniques for avoiding overfitting include pruning the tree; that is, restricting the number of consecutive branches so that is forced to generalize. More powerful solutions are possible by repeatedly sampling the data to build many trees and combining these trees into a single predictive model using techniques known as *bagging** [50] and *boosting** [269, 268]. Combined tree models may be harder to interpret that single trees, but standard approaches allow determination of which genes contributed most heavily to the models predictive powers [51].

To decide how to best approach a prediction problem, it is important to first consider the desired outcome. Are there just two classes to be distinguished, or many? Is it desirable to find the minimal number of predictive genes, in order to minimize the number of leads or to provide a simple diagnostic tool? Would it be better to have an easily interpretable model, which may help provide new medical insights, or is the only goal the greatest prediction accuracy possible? If the output will ultimately affect patients treatment, it may be essential to have an accurate confidence estimate for each prediction. All of these issues can influence the choice of a prediction method.

Sixth step: prediction accuracy evaluation and refining the model

The accuracy evaluation problem concerns in estimating the classification error on test data set. Thus, this step is directly linked with the fourth step: training and test sets selection explained above. Besides, many predictions methods require to refine the model by tuning some parameters such as the number of genes, the number of nearest-neighbors to consider, or the number of decision trees built etc. Thus, the model refinement step can also be evaluated by prediction accuracy estimators.

Evaluation prediction accuracy

Prediction accuracy evaluation refers to error estimation of the classifier on the test set. The error rate of a classifier is the proportion of incorrectly classified samples. The true error rate depends on the class distribution. If the class distribution was known, the true error could be computed exactly [339]. However, the distribution is often unknown in practice, and the error rate has to be estimated from the given data set . Most of the commonly used error estimators result from combining one of the five methods proposed in the fourth step (re-substitution, hold-out, cross-validation, bootstrapping, non-random method etc.) with an error function criterion [33]. Among the most known error criteria we can mention four methods: *error counting*, EC, *smooth modification of error counting*, SM, *posterior probability estimate*, PP, and *quasi-parametric estimate*, QP.

Error counting is the typical scheme where the output below a given threshold is hard thresholded to belong to one class, otherwise to the other class. For multiple classes, the winner takes all.

Smooth modification of error counting consists in taking the part of the correctly classified sample observations for estimate the misclassification probability.

Posterior probability estimate is the probability of a sample belonging to a class. An advantage of this estimate is that the test data can be unlabeled.

Quasi-parametric estimate assumes that the values of the discriminant function have a normal distribution. Error rate is found analytically from sample means and variances of the output of discriminant functions for different observations.

Hence, by combining this error functions criteria with the training-test selection approaches, we can have at least 20 different methods of error estimation. More details in this issue can be found in [339].

The choice of any particular combination for the error estimation function depends in several factors, here we make the following recommendations for choosing one of them as stated in Raudys and Jain [252]:

- The re-substitution method gives in optimistically biased estimates of asymptotic error rates. Hence, it should only be used when the sample size is large.
- The hold-out error counting estimate results in an unbiased estimate of the expected error rates. the disadvantage of this method is that not all observations of the design sample take part in the learning process and only a part of them are used for calculating classification error.
- The leave-one-out estimate produces a practically unbiased estimate of the expected error rate if the sample observations are statistically independent. For dependent observations, the estimate approaches that of re-substitution method. the main disadvantage is that for some classifiers it is extremely computationally expensive.
- Bootstrap methods and their variants appear to be more accurate than leave-one-out estimates only when the classification error is large.

- The variance of SM, PP and QP estimates can be less than the variance of the EC estimate. the first three estimates are also biased depending on the data type.

Refining the model

One of the critical issues in applying supervised algorithms for solving bioinformatics problems is the expert's ability in understanding of machine learning algorithms and setting the parameters contained along the first six steps procedure described here. A good experimental design for data prediction and adequate optimization of algorithms is critical to the successful application of machine learning techniques. Refining the model consists in tuning and optimizing the parameters along the prediction model process.

One of the major problems is the parameter optimization of different machine learning algorithms used in bioinformatics. Classifiers, clustering methods, gene and sample selection methods use a number of parameters that must be optimized. Optimization can be a tedious task especially when these parameters are continuous variables. A general practice for parameter optimization is to use a validation set on which the impact of tuning parameters can be judged and optimized. One of the underlying assumptions of this process is that the validation set closely mirrors the test set, which is often found to be unreliable. Concerning the experimental design problem, the most delicate issue in machine learning is the amount of data that is necessary for building reliable and robust machine learning. Other related problems are how to sample for a validation set, how to choose classifiers, how to describe a cost matrix and a rejection threshold, how to develop machine learning systems that can automatically determine the optimal parameters. For further details in this issue the lector can see [33].

Seventh step: biological interpretation of the prediction model results

The resulting prediction model, as well as the prediction results, the informative genes selection and the discovered classes, have to be interpreted by experts and may yield new biological insights. In order to achieve this stage, the expert realizes the interpretation step manually or in a primitive semi-automatic way. As well as seen in the class discovery problem (see chapter 5), the expert analyses the results taking into account either his personal experience or by searching in at least one of the several sources of biological information explained in chapter 3. Sometimes he uses semi-automatic tools for extracting information from this heterogeneous sources (as explained in chapter 3).

The use of machine learning for solving bioinformatics problems is a relatively new field compared to the use of supervised learning algorithms in other domains. Thus, the interpretation of the predictive results via the integration of biological knowledge is an open field of research. Even the integration of biological knowledge in any of the 7 steps process for building a prediction model has not yet been tackled.

Here, we propose the integration of gene annotations at some delicate steps in building a predictive model as class discovery, data selection, building a predictive model and biological

interpretation of predictive model results. This task is a recent issue in machine learning application in bioinformatics.

8.3 Discretization issues in Gene Expression Technologies

In order to answer to the discretization question, an essential requirement in many supervised algorithms, we have developed a novel algorithm named **NORDI** (normal discretization), specially fitted to gene expression technologies. NORDI is based on statistical detection of outliers and the continuous application of normality tests for transforming the initial distribution "almost normal" to a "more normal" one. The term "almost" means that the sample S_j can be normally distributed without the outlier's presence. Conception, implementation [33], experimentation and validation are presented in this document.

As seen in section 8.2, a crucial step in the process of building a classifier is the discretization step (see FIG. 8.1). In order to build a decision tree model all values in T must be qualitative (known also as categorical). Specifically in gene expression technologies, we have two kinds of data information: gene annotations that are often textual and which can be treated as qualitative variables, and gene expression measures, which are generally quantitative continuous variables. So, gene expression measures must be converted into discrete values through a discretization process technique.

Discretization of gene expression measures consists in determining qualitative values which reflect the degree of gene expression. This question is directly related to the third analysis step of gene expression technologies: analysis of differentially expressed genes discussed in section 2.2. However, the main difference is that in discretization techniques, we focus on establishing fixed intervals to build qualitative values representing the degree of expression or the expression or unexpression of gene expression measures [246]. Answering this question is a difficult task, but a variety of discretization approaches have been applied in gene expression technology. Several supervised and non-supervised discretization methods have been reported in the literature [63], [96] .Here, we focus on the most commonly used discretization approaches applied in gene expression technologies, and we present a novel discretization method specially adapted to microarray data named NORDI.

Gene expression technologies allow us to measure simultaneously gene expression profiles of thousands of genes in different biological conditions (time, different tissues, etc.). Let us assume the expression data measures presented as a transposed matrix E^T: mXn explained in section 8.4.1.2, where E^T is a matrix with n genes (columns) and m biological conditions (rows), as illustrated in FIG. 8.1. The columns of this matrix, X_j, are gene vectors and the rows are the samples S_j. Each matrix entry, $e_{j,i}$ represents the gene expression measure of gene i (variable) in biological condition or sample j where $e_{j,i} \in \mathbb{R}$, so it is a continuous in all real numbers. The question is: What's the degree of expression of each gene i in sample j in a qualitative manner? We can answer this question by applying discretization methods. Here,

[33] NORDI program is available by request, and soon it will be available in bioconductor project: http://www.bioconductor.org/.

we will classify the gene expression technologies discretization methods in three approaches with *biological* basis, *statistical* basis and *mining* basis respectively.

8.3.1 Biological methods:

In the origin of microarray experiments, scientists realized the difficulty of obtaining cleaned (without noise) microarray data. They have established a methodology for getting rid of most of the noise, but keeping the representative part of the expression measure [246](explained in chapter 2 as the first two analysis microarray steps). Once they obtain the cleaned expression measure, they make several tests for qualifying the obtained measure [246] and they report this expression measure in easily handled measures as the fold change (FC), Signal to Noise Ratio (SNR) and others [246] for analysis purposes. Here, we present the most common discretization methods for determining if each matrix entry of E^T is telling us the expression or non-expression of gene i in sample j.

Two-fold change cutoff

Taking the cleaned expression measure as the Fold change measure defined as:

$$Fold\ Change = \log\ Ratio \quad \text{where} \quad Ratio = \frac{Cy5}{Cy3} \tag{8.1}$$

This *Ratio* is the relationship between the two color variables: $Cy5$ (red) and $Cy3$ (green) which means the response of one gene in a state labeled with red relative to a state labeled in green for each biological condition (as seen in chapter 1, microarray basics). Applying log function to the *Ratio* reduces the big quantities problem and makes data more easily handled.

Fold change measure has the advantage of conserving the biological homeostasis principle of an organism: "Cellular inhibitions and repressions must be compensated" because it considers cellular inductions and repressions in a numerically equal manner [219].

We suppose that each element of the matrix, $e_{j,i}$, for every $i = 1, 2, ...n$ and $j = 1, 2, ...m$ is given in terms of fold change measure of gene i in biological condition j. If the whole matrix E^T accomplishes the following characteristics:

1. All data is well cleaned (minimal noise).
2. No outliers.
3. Number of genes is largely enough.
4. The rows of the matrix S_j for every $j = 1, 2, ...m$ are independent from each other and are normally distributed $S_j \sim N(\mu_j, \sigma_j)$.
5. Missing values are no significant in relation to the number of genes.

Supposing this five characteristics and using the central limit theorem [113], we can say that the matrix E^T is distributed as normal distribution $N(0, 1)$ where $\mu = 0$ and $\sigma = 1$.

This assumption is "approximately" proven in practice by gene expression experiments. Thus, biologists have adopted the $2 - fold$ threshold as an intuitive measure for determining

the non-expression of a gene expression measure. They have defined $2-fold$ threshold cutoffs as:

$$\begin{aligned} e_{j,i} &\geq \mu + \frac{k}{2}\sigma = Ot = 1 \Longrightarrow e_{j,i} \quad \text{over-expressed} \qquad \text{where } k = 2 \\ e_{j,i} &\leq \mu - \frac{k}{2}\sigma = Ut = -1 \Longrightarrow e_{j,i} \quad \text{under-expressed} \quad \text{where } k = 2 \\ -1 &= Ut < e_{j,i} > Ot = 1 \Longrightarrow e_{j,i} \quad \text{unexpressed} \qquad \text{where } k = 2 \end{aligned} \tag{8.2}$$

The absolute distance that separates the over-expressed threshold from the under-expressed threshold, i.e. $|Ot - Ut| = 2$, gives us the unexpressed interval of size 2, which determine the name of $2 - fold$ threshold cutoff.

Although this measure has not a rigorous statistical definition, and it could depend on many factors as type of gene expression technology, gene expression intensities etc., it is one of the most used measures [225]. For resolving the gene expression intensity problems, biologists have used several similar thresholds defining the $k - fold$ change cutoffs. This cutoffs can be calculated by using the above equation 8.2 for an specific predefined k, i.e. $k = 2, 3, 4, ...K$.

All the $k - fold$ change cutoffs present several weakness:

- They have to accomplish the five characteristics cited above.
- In most of the cases gene expression technologies have outliers and a $\sigma \neq 1$. Therefore, data containing most of genes with very low fold change intensities but with high standard deviation σ would be interpreted as unexpressed genes. On the other hand, data containing most of genes with very high fold change intensities but with low standard deviation σ would be taken as expressed genes.
  Several others weaknesses and misuses have been analyzed and reported in the literature [225], [83], [219], [66], [59].

Equal number of expressed genes

This method was used mainly in cancer studies when the biologists seek for the signature of genes, that is, the specific genes that participate in the disease. These genes can be either inhibitors or activators, so they have to be in equal number for conserving the biological homeostasis principle (seen above). Thus, let E^T be the profiles expression matrix, the equal number of over-expressed and under-expressed genes EN is user-determined (taking into account user specific disease knowledge). For constructing the discretization intervals, we need an ordered list from each row S_j of the profiles expression matrix E^T in a descending way. Then, we define the discretization intervals for each row S_j as:

$$\begin{aligned} \text{First } EN \text{ genes of ordered column } S_j &\Longrightarrow EN \text{ over-expressed genes} \\ \text{Last } EN \text{ genes of ordered column } S_j &\Longrightarrow EN \text{ under-expressed genes} \\ \text{All genes in } S_j \text{ except first and last } EN &\Longrightarrow \text{unexpressed genes} \end{aligned} \tag{8.3}$$

This fixed and somewhat arbitrary threshold present several disadvantages, principally the low statistical significance [225]. However, it could be useful as a comparative discretization method or in some disease-specific studies. This discretization methodology is useful for comparing the effectiveness of other discretization methodologies in order to predict the state of a tissue [250].

8.3.2 Statistical methods

Statistical methods are generally based on the normal distribution of each row S_j of the profiles expression matrix E^T. Each individual measure $e_{j,i}$ is often taken as the fold change (as defined before). An assessment of the principal statistical methods was made by [83]. The objective of most of these methods was to distinguish the genes which were differentially expressed in one biological condition or all around the biological process. In the last six years, a multiplicity of statistical methods has been applied and developed to solve this question, we can mention: *z-score* [327], [304], *t-test* applications [59],[309], [17], ANOVA (*F-test* applications) [323], [159], [183], [106], [321], [66] and *mixed models.*[220]. The majority of these methods has been used for selecting differentially expressed genes between two conditions or in the whole biological experiment. However, it could be used for discretization purposes. Here we will focus on the most generalized methodology for obtaining discrete intervals from the expression matrix E^T: the $z-score$ methodology.

Z-score method

The named $z-score$ method is based in the commonly known z-statistic: [113]. The first that has used it in a discretization gene expression context was [327] and it can be seen as an statistical formalization of the $k-fold$ change method.

We suppose that each element of the matrix E^T is given in terms of fold change measure of gene i in biological condition j. Let the expression matrix E^T accomplish the 5 characteristics cited before in $2-fold$ change method. Thus, for each row, $S_j \sim N(\mu_j, \sigma_j)$ (normally distributed) and a certain level of predetermined confidence $1-\alpha$, we define the $z-score$ threshold cutoffs as:

$$\begin{aligned} Z &= \frac{e_{j,i}-\mu_j}{\sigma_j} \geq z_{\alpha/2} = Ot \Longrightarrow e_{j,i} \text{ over-expressed,} \\ Z &= \frac{e_{j,i}-\mu_j}{\sigma_j} \leq z_{\alpha/2} = Ut \Longrightarrow e_{j,i} \text{ under-expressed,} \\ Ut &< e_{j,i} > Ot \Longrightarrow e_{j,i} \text{ unexpressed,} \end{aligned} \tag{8.4}$$

where $Z \sim N(0,1)$ and $z_{\alpha/2} = \Phi^{-1}(1-\alpha/2)$ if the cumulative distribution function is

$\Phi(z_{\alpha/2}) = P(Z \leq z_{\alpha/2}) = 1-\alpha/2$.

Supposing that every row $S_j \sim N(0,1)$ and $z_{\alpha/2} = 1$ we obtain the same threshold as the $2-fold$ change method. The equality $z_{\alpha/2} = 1$, means that we have labelled about 68.26%

of the genes as unexpressed and 15.85% as over-expressed and under-expressed respectively (see normal distribution percentiles and characteristics [113].

The z-score method is more robust than $2-fold$ because we can take into account the row variance, σ_j^2, of the sample, and we can choose the α parameter. However, as well as the $2-fold$ method, all five expression matrix characteristics have to be accomplished, which is not often the case in microarray data. Some variation of the $z-score$ method have been proposed [327].

Next, we will present the Yang method that tries to surpass some limitations established by the normality distribution and the existence of outliers.

Yang discretization method

Based on the $z-score$ methodology Yang [327] has proposed the use of a raw matrix E'^T before cleaning process of the gene expression profiles. Let us take a look at each row, S'_j, of the expression raw matrix E'^T. Yang's method, first realizes a fine statistical treatment of the gene expression measures of the raw matrix E'^T applying missing data algorithms and *lowess* algorithm [264] (explained in section 2.1). Then it makes the detection and filtration of the outliers contained in each row of the data, using an intensity based technique [327]. Finally, it applies $z-score$ discretization methodology, establishing the Ot and Ut cutoffs.

This method is an attempt of pre-analyzing the expression matrix profiles before applying the $z-score$ method, but eliminating definitely the outliers could imply high risks. There are two principal types of outliers: technical ones as measure errors and highly expressed genes. Outlier elimination is an erroneous procedure because that are precisely the outliers which can contain exceptional information about the biological process, thus they are crucial for it [219]. The NORDI discretization method, proposed by Martinez [198], surpasses some limitations established by the $z-score$ and Yang discretization method, specially concerning the outliers treatment and the normality test implemented tools.

8.3.2.1 NORDI discretization method

In order to resolve some of the $z-score$ and Yang's methodology drawbacks, Martinez has proposed the normal discretization method (Nordi) [198]. This method supposes that the gene expression matrix $\boldsymbol{E}^T$ already contains well-cleaned expression measures $e_{j,i}$, where the number of genes is large enough and the missing values are not representative (as viewed in characteristics 1, 3 and 5 of $2-fold$ change method).

NORDI is based on a statistical detection of outliers and the continuous application of normality tests for transforming the initial distribution "almost normal" [34] to a "more normal" one. Once the distribution of the matrix is "more normal", it calculates the cutoffs as seen in the $z-score$ methodology. Here, we present some basics in outlier treatment methods and normality tests, underlying the methods used by NORDI algorithm.

[34] Almost normal means that the sample S_j could be normally distributed by removing sample outliers.

Outliers

Extreme values have been a source of debate among the data analysts community. The presence of extreme values in a data set can be due to systematic errors, faults in the experimental conditions,erroneous procedures, areas where a certain theory might not be valid, or it can simply be the case that some observations happen to be a long way from the center of the data. Furthermore, these values can be taken as a source of contamination in data or they can be seen as a source of interesting information or unusual special events. Hence, it is a crucial data analysis task of interpreting and characterizing outliers, thus developing statistical methods to treat them in order to decrease their impact during statistical data analysis [20].

An outlier can be defined in many ways, a statistical definition given by Grubbs [135] is: An outlier is an observation that appears to deviate markedly from other members of the sample in which it occurs. Munoz Garcia [218] propose another definition as an observation that deviates clearly of the general behavior compared to the criterion on which the analysis is carried out. Barnett and Lewis state that an outlier is an observation among a set of observations, which clashes or is not in harmony with the rest of the observations in the set. What characterizes an outlier is its impact on the observer [20].

Outlier's treatment methods

A complete survey for concepts, tendencies and methods for treating outliers has been made by Planchon [241]. Here, we will only focus on the statistical point of view for treating outliers in relation with a probability model as normal distribution. In this section, we see outliers as in the Barnett and Lewis definition, and we can localize them in the extremes values of a statistical distribution.

The principal methods for treating outliers against a probabilistic model, known as discordance test methods [20] were presented in section 2.1.3. The goal of discordance test methods is to test the outlier value in order to reject it of the whole of the data or to identify it as being a characteristic of a particular interest. Thus, this test is a procedure of detection that allows to decide in favor of the membership of a specific value to the data set or not.

Supposing an univariate distribution case where the sample of a random variable X, is x_1, x_2,..., x_n. The extreme values are x_1 and x_n, both of them, or one of the two for example x_n can be outliers if it is *statistically unacceptable*, in relation with the distribution de x_n under F. When the result of the test indicates that x_n is not acceptable in a statistical way, one can say that x_n is a discordant superior value for the level of the test. In a similar way it can be shown for the inferior value x_1 or even for the couple (x_1, x_n).

As we have seen in section 2.1, the gene expression measures taken at each biological condition have an "almost" normal distribution, so we are interested in normal distributed discordance tests. Among these methods, we can cite: Rosner's [129], Dixon 1950 [94], Grubbs 1950 [134], Cochran [77] and Tietjen 1972 [306].

These 5 tests present several advantages and disadvantages depending on the characteristics of the treated data set. In a study, the EPA environmental protection agency (US

EPA, 1992), has qualified the effectiveness of these 5 methods in their detection quality. The winner in correct outlier detection was Grubbs Test [3], the data sets were very large samples (the order of 10^4 values) of environmental data that were log-normally distributed. That is exactly the same case of gene expression data rows S_j, where the gene expression value is in terms of the logarithmic fold change measure and each matrix row: S_j for $j = 1, 2, ...n$ is log-normally distributed.

Grubbs' test ([134] and [135])

Grubbs definition of outliers is: "An outlier is one member that appears to deviate markedly from other members of the sample in which it occurs. Grubbs methodology provides statistical rules that lead the data set analyst to look for causes of outliers when they really exist, and hence to decide which alternative between the case of experimental errors or interesting information is the correct one.

Grubbs test is used to detect outlying observations in a univariate data set, based on a assumed underlying normal population or distribution. Let us suppose a sample population X that is ordered from the lowest value x_1 to the highest value x_n. We want to test if one of these two extreme values: x_1 or x_n or both of them: x_1 and x_n can be outliers. They would be outliers if they are statistically unacceptable in relation with the distribution de x_n or x_1 or both of them under the normal distribution F. Thus, we have the hypothesis testing:

$$H_0 : \text{There are no outliers in the data set}$$

vs

$$H_{1A} : x_1 \text{ or } x_n \text{ is an outlier in data set.}$$

or for both of them:

$$H_{1B} : x_1 \text{ and } x_n \text{ are outliers in data set.}$$

The Grubbs statistic for the first one-sided H_0 vs H_{1A} test is:

$$G_A = \frac{x_{outlier} - \bar{X}}{\mathcal{S}} \tag{8.5}$$

where $x_{outlier}$ can be x_1 or x_n depending in the maximal deviation from the sample mean $\mathcal{X}$

and $\mathcal{S}$ is the standard deviation estimator of the population sample, i.e.

$$\mathcal{S} = \frac{\sum_{i=1}^{n}(x_i - \bar{X})^2}{n-1}. \tag{8.6}$$

Grubbs statistic for one outlier 8.5 is equivalent in testing power to the statistic U:

$$U_A = \frac{V_1^2}{V^2}$$

where V_1^2 is the variance of the sample with one suspicious value excluded, and V^2 is the

variance of the whole population sample X.

If the estimators in equation 8.5 are biased (with n in denominator), then simple dependence occurs between V^2 and U_A, i.e. U_A: $U_A = 1 - \frac{1}{n-1}G_A^2$ and it makes U_A and G_A statistics are equivalent in their testing power. The G_A distribution for one outlier test is $t - student$ with $n-2$ degrees of freedom [232], and the following formula can be used to approximate the critical value:

$$CV_A = t_{\alpha/n,n-2}\sqrt{\frac{n-1}{n-2+t^2_{\alpha/n,n-2}}} \tag{8.7}$$

In the case of the detection of both outliers, that is the two-sided test H_0 vs H_{1B}, the Grubbs statistic is:

$$G_B = \frac{x_n - x_1}{\mathcal{S}} \tag{8.8}$$

where x_1 or x_n are the extreme values of the sample X. Grubbs statistic for both outliers 8.8 is equivalent to the statistic U_B:

$$U_B = \frac{V_2^2}{V^2},$$

where V_2^2 is the variance of the sample with two suspicious values excluded, and V^2 is the variance with two values excluded.

In the case of G_B and U_B statistics, Grubbs gives only G_B distribution values because U_B was too complicated to calculate[134]. For approximating the critical values for α threshold of G_B David and Pearson 1954 [233] have given the next distribution

$$CV_B = t_{\alpha/n,n-2}\sqrt{\frac{2(n-1)t^2_{\alpha/n(n-1),n-2}}{n-2+t^2_{\alpha/n(n-1),n-2}}}. \tag{8.9}$$

So, we have presented two tests for detecting outliers: the G_A and U_A statistics for one outlier and G_B and U_B for two outliers at the same time. At a given α threshold the correspondent critical values can be calculated by the CV_A and CV_B formulas 8.7 and 8.9 respectively. In this manner we can reject or accept the corresponding hypothesis H_{1A} and H_{1B} for accepting or rejecting the existence of one or both outliers. More details on this outliers detecting method can be seen at Grubbs publications: [134] and [135].

Normality test methods

The normal distribution is the most widely used family of distributions in statistics, and many statistical tests are based on the assumption of normality. In probability theory, normal distributions arise as the limiting distributions of several continuous and discrete families of distributions. The fundamental importance of the normal distribution as a model of quantitative phenomena in the natural and behavioral sciences is due to the central limit theorem [113]. A variety of psychological test scores and physical phenomena like photon counts can be well approximated by a normal distribution. In our case, as a consequence of the psychological homeostasis principle, gene expression profiles matrix E^T and each one of the rows S_j of E^T is assumed to be normally distributed $S_j \sim N(\mu_j, \sigma_j)$. Nevertheless, several rea-

sons like the existence of outliers, experimental or manipulation errors etc., causes variations in the distributions of the studied data sets. Thus, normality tests are necessary to assure a given data set distribution being similar to the normal distribution.

The null hypothesis, H_0 , states that the data set is similar to the normal distribution, therefore a sufficiently small $p - value$ indicates non-normal data. Several tests have been made for answering the normality question posed by the H_0 , we can mention: graphical methods as: $Q - Q$ plots and *rainkit* plots and several statistical tests: Kolmogorov-Smirnov , Lilliefors [1], [180] , Anderson-Darling, Shapiro-Wilkinson [277] and Jaeque-Bera test [30]. For more details on these tests see [223]. In these section, we will be specially interested in $Q - Q$ plots, Lilliefors test and Jarque-Bera tests which have particular characteristics that are useful for our NORDI algorithm.

Lilliefors test

Lilliefors test [180] is an adaptation of the Kolmogorov-Smirnov test [1]. It is used to test the null hypothesis H_0 : Data is normally distributed. The null hypothesis does not specify which normal distribution, i.e. it does not specify the expected value and variance. The test proceeds as follows:

- First estimate the population mean μ and population variance σ based on the data.
- Then find the maximum discrepancy between the empirical distribution function and the cumulative distribution function (CDF) of the normal distribution with the estimated mean μ and estimated variance σ. Just as in the Kolmogorov-Smirnov test, this will be the test statistic.
- Finally, we confront the question of whether the maximum discrepancy is large enough to be statistically significant, thus requiring rejection of the null hypothesis. This is where this test becomes more complicated than the Kolmogorov-Smirnov test. Since the hypothesized CDF has been moved closer to the data by estimation based on those data, the maximum discrepancy has been made smaller than it would have been if the null hypothesis had singled out just one normal distribution. Thus we need the "null distribution" of the test statistic, i.e. its probability distribution assuming the null hypothesis is true. This is the Lilliefors distribution. To date, tables for this distribution have been computed only by Monte Carlo methods.

The test is relatively weak for samples when n is small, but it can be robust when a large amount of data is available. Lilliefors test is effective for normal distributions with large amount of data and when we do not know the expected mean μ and variance σ of the distribution. This is the case of our gene expression profiles matrix rows S_j. If that is not the case it is preferable to choose Kolmogorov-Smirnov or Anderson-Darling test that are generally more robust and more easy for calculating resulting distributions [1].

Jarque-Bera test

The Jarque-Bera test [30] is a goodness-of-fit measure of departure from normality that answers to the null hypothesis H_0 stated before. It is based on statistical third and fourth standardized momentums $\gamma_3 = \mu_3/\sigma^3$and $\gamma_4 = \mu_4/\sigma^4$ that are best known as: *skewness* and *kurtosis* measures respectively.

Skewness, γ_3, can be seen as a measure of the asymmetry of the probability distribution of a real-valued random variable X. Roughly speaking, a distribution has positive skew (right-skewed) if the right (higher value) tail is longer or fatter and negative skew (left-skewed) if the left (lower value) tail is longer or fatter.

Kurtosis, γ_4, is a measure of the "peakedness" of the probability distribution of a real-valued random variable X. Higher kurtosis means more of the variance is due to infrequent extreme deviations, as opposed to frequent modestly-sized deviations.

The Jaeque-Bera test is defined as:

$$J = \frac{n}{6}\left(\gamma_3^2 + \frac{\gamma_4^2}{4}\right) \tag{8.10}$$

where n is the number of observations (or degrees of freedom) of the distribution. J statistic has an asymptotic chi-squared χ_2^2 distribution with two degrees of freedom. Intuitively, any sample from a normal distribution has an expected skewness $\gamma_3 = 0$ and expected kurtosis $\gamma_4 = 0$. In this manner, as seen in the equation 8.10, any deviation from this two measures increases the J statistic, and its $p - value$ calculated at a given threshold α will be lower, so it will be more probable to reject H_0 in favor of H_1.This method is particularly effective for samples distributed almost as normal distributions, because this method is the most sensible in detecting the existence of outliers from the ones cited before: $Q - Q$ plots, *rainkit* plots, Kolmogorov-Smirnov, Lilliefors, Anderson-Darling, Shapiro-Wilkinson. The characteristic of the Jaeque-Bera test is given because it is based on *skewness* and *kurtosis* measures, that can be seen as asymmetry and peakedness measures, which can be severely damaged by presence of outliers. So, this method is very sensible to outliers in normal distributed samples.

Q-Q Plot

$Q - Q$ *plot* ("Q" stands for quantile) [223] is a tool for diagnosing differences in distributions,such as non-normality, of a population from which a random sample S has been taken. It consists in plotting the $\frac{k}{(n+1)}$ quantiles of the comparison distribution (i.e. the normal distribution) on the horizontal axis (for $k = 1, ..., n$), and the order statistics of the sample S on the vertical axis. Quantiles are essentially points taken at regular intervals from the cumulative distribution function (as normal distribution) of a random variable X. Furthermore, the k^{th} order statistic of a sample S is equal to its k^{th}-smallest value.

$Q - Q$ *plot* tries to answer the hypothesis testing stated above: H_0 : The sample distribution is similar to a normal distribution versus H_A : The sample distribution does not looks alike graphically to a normal distribution.

In a normal distribution population the $Q-Q$ *plot* approximates a straight line, especially near the center. In case of substantial deviations from that appearance, the analyst rejects the null hypothesis H_0 of resemblance. More details can be seen in [223].

$Q-Q$ *plots* are similar to *rankit plots*, also called normal probability plots. The difference is that in normal probability plots, instead of the $\frac{k}{(n+1)}$ quantiles of the normal distribution, one plots the expected value of the k^{th} order statistic from a normal distribution with $\mu = 0$ and $\sigma = 1$. Only when n is small there is a substantial difference between a $Q-Q$ *plot* and *rankit plot*.

$Q-Q$ *plots* are an easy exploratory and graphical tool for visualizing a given sample versus another distribution as the normal one It can be useful as well for: detecting possible outliers, detecting similar tail behavior and similar distribution shapes and to see common location and scale between distributions.

8.3.2.2 NORDI algorithm

Let's suppose that the gene expression matrix $\boldsymbol{E}^T$ already contains well-cleaned expression measures: $e_{j,i}$, for $i = 1, 2, ...n$ genes and $j = 1, 2, ..., m$ biological conditions, the number of genes is large enough and the missing values are not representative. Suppose that each biological condition has an "almost" normal distribution $S_j \sim N(\mu, \sigma)$ for $j = 1, 2, ...m$. The NORDI algorithm states that every sample of the expression matrix S_j can be normally distributed $S_j \sim N(\mu, \sigma)$ if all outliers of each row are removed (but keeping a list of removed outliers, i.e. L^k) by Grubbs outliers method (explained before). Each time an outlier k is removed, a Jaeque-Bera normality test has to be accomplished for the remaining sample S_j^k where k is the number of removed outliers at each step, $k = 0, 1, 2, ..., clean$ ($k = clean$ means that there are no more outliers in the sample according to Grubbs test). If the remaining sample S_j^{clean} is "more normally" distributed than the original sample S_j according to QQ-plot and Lilliefors normality tests, then we choose the cleaned sample S_j^{clean} for computing the cutoff thresholds. We then applied the $z-score$ methodology to sample S_j^{clean} in order to calculate the over-expressed: Ot and under-expressed: Ut cutoffs 8.4.

The discretization procedure is done over the whole sample S_j, within all the removed outliers in the list L^K, using the Ot and Ut cutoffs. This procedure is repeated for all m samples. It is important to notice that the threshold cutoffs are calculated over the cleaned sample S_j^{clean} that is "more normally distributed" than the initial one S_j^0. However, the elements in the final outliers list L^K, has to be taken into account for the gene expression analysis because they may contain the most relevant information, so they can not be removed from the analysis.

Pseudo-code for NORDI discretization algorithm is presented on Fig. 8.2. The algorithm has been implemented in R (language). It takes as input each one of the gene expression matrix rows S_j, the number of discretization intervals, ndi, and user's $p-value$ for outliers. It returns as output the discretization intervals for each biological sample S_j.

The NORDI algorithm begins by reading all the gene expression matrix profiles (step 2). For each one of the row samples, S_j, it orders the sample in ascending manner (step 4) and it computes the $QQ-Plot$ test and $Lillierfors$ test (step 5). Then, it computes the

Input: Gene Expression Profiles Matrix Rows: S_j for $j = 1, 2, ..., m$
Number of discretization intervals $ndi = 3$
Outlier's $p - value$
Output: Ot and Ut cutoffs and ndi intervals

Begin
lecture gene expression matrix: E^T and $ndi = 3$
for each row S_j of the matrix **do**
  order in ascending manner the row S_j
  compute QQ-Plot Test and Lillierfors Test LF for S_j
  if QQ or LF **then**
    compute Jarque Bera JB normality test: J_0
    assign Outliers: $Out = 1$ and Normality Amelioration: $Noa = 0$
    while $Out = 1$ and $Noa \geq 0$
    compute two-sided Grubbs statistic: G_B and critical value CV_B
    compute one-sided Grubbs superior statistic G_Asup and CV_Asup
    compute one-sided Grubbs inferior statistic G_Ainf and CV_Ainf
    if $CV_B < p - value$ **then**
      assign $Out = 1$
      compute JB test without k outliers J_1 for sample S_j^k
      if $Noa = J_0 - J_1 \geq 0$ **then**
      remove momentarily outliers from column S_j^k
      else if $CV_Asup < p - value$ **then**
        assign $Out = 1$
        compute JB test without superior outlier J_1
        if $Noa = J_0 - J_1 \geq 0$ **then**
        remove momentarily outliers from column S_j^k
        else if $CV_Ainf < p - value$ **then**
        assign $Out = 1$
        compute JB normality test without inferior outlier J_1
        if $Noa = J_0 - J_1 \geq 0$ **then**
        remove momentarily outliers from column S_j^k
        end if
        else assign $Out = 0$
      end if
    end if
    assign $J_0 = J_1$
    end while
  end if
  compute QQ-Plot Test and Lillierfors LF Test to S_j^{clean}
  if the sample S_j^{clean} is more "normal" than original S_j **then**
  compute Ut, Ot and $ndi = 3$ **intervals** by $z - score$ methodology
  else continue
end for
End

FIG. 8.2: NORDI algorithm

Jaeque-Bera (JB) 8.10 normality test J_0 of the sample S_j^k and it assigns two decision variables: the boolean variable of outlier's presence: Out and the normality amelioration: $Noa = J_0 - J_1$ (steps 7-8). The normality amelioration is obtaining by computing the JB test, J_1, for sample S_j^k and JB test, J_0, for sample S_j^{k-1}. In the presence of outliers $Out = 1$ and while it exists normality amelioration $Noa \geq 0$[35], NORDI will compute the Grubbs statistics G and critical values CV for two outliers, the top or superior outlier and the inferior outlier (See G statistics: 8.5,8.8 and CV equations: 8.7,8.9) (steps10-12). Then, it will test the CV of each one of the three outlier cases (both, superior and inferior) against the predefined $p - value$ (steps: 13 or 18 or 23). If it exists normality amelioration $Noa \geq 0$ (steps: 16 or 21 or 26), then it will remove the tested outlier (steps: 17 or 22 or 27). The procedure will finish when there are not outliers $Out = 0$ or there is not normality amelioration $Noa \prec 0$. For each sample S_j it would be obtained an outlier cleaned sample S_j^{clean}. NORDI will compare the normality of each of this two samples applying $QQ - Plot$ test and $Lillierfors$ test (step 36). If S_j^{clean} is "more normal" than S_j, then it will compute the Ut and Ot cutoffs as well as the 3 discretization intervals (steps 36-37), as seen in the $z - score$ methodology procedure 8.4. The procedure will be done for all the biological samples $j = 1, 2, ...m$ in the gene expression matrix.

8.3.3 Mining methods

Other discretization methods were proposed in order to help data mining methods as ARD and decision trees implementations in specific transcriptome analysis applications as cancer studies, metabolic pathway analysis etc. Among these methods we can mention three basic threshold methods by Becquet et al. [22], a mean gene expression method [286] and a complex one based on fuzzy logic by Woolf [322].

Three basic thresholds methods

They were proposed by [22] and used within the framework of ARD methodology in gene expression SAGE technology applied to cancer data. Their interest lies in transforming the SAGE expression profiles matrix E^T into a discrete boolean matrix, where 0 means no gene expression and 1 means gene expression.

These methods use SAGE [36] data where all expression profiles matrix values are null or positive natural numbers, i.e.: $e_{j,i} \geq 0$ and $e_{j,i} \in \mathbb{N}$.

Max-Minus $\delta\%$ method: Let us take an user pre-determined percentage $\delta\%$ for all of the matrix E^T.

First we calculate the maximal value of the row: $Max(S_j)$.Then using the pre-determined $\delta\%$ we calculate the row threshold as: $Et = Max(S_j)(1 - \delta\%)$. Thus, the corresponding discretization is:

[35] $Noa \geq 0$ because the Jaque Bera test states that a normal distribution has a JB coefficient equal to 0.If $Noa \geq 0$ then the sample without k outliers: S_j^k.is "more normal" than the sample without k-1 outliers: S_j^{k-1}.

[36] SAGE method do not use the fold change value currently used in microarray data.

$$\begin{aligned} e_{j,i} &\geq Et \Longrightarrow e_{j,i} \text{ expressed or 1,} \\ e_{j,i} &\leq Et \Longrightarrow e_{j,i} \text{ unexpressed or 0.} \end{aligned}$$

Applying this formula to each row $j = 1, 2, ...m$ we convert E^T into a boolean matrix.

Mid-Ranged method: First we calculate the maximal and minimal value of the row: $Max(S_j)$ and $Min(S_j)$ .Then we calculate the mid-ranged threshold as:

$$Et = \frac{Max(S_j) - Min(S_j)}{2} + Min(S_j).$$

So, we have the discretization:

$$\begin{aligned} e_{j,i} &\geq Et \Longrightarrow e_{j,i} \text{ expressed or 1,} \\ e_{j,i} &\leq Et \Longrightarrow e_{j,i} \text{ unexpressed or 0.} \end{aligned}$$

Applying this formula to each row $j = 1, 2, ...m$ we convert E^T into a boolean matrix.

Highest values at δ% level: Let us take an user pre-determined percentage δ% for all of the matrix E^T.

First, we calculate δ% of the number of genes (without counting the missing values) as:

$$\delta\% n(S_j) = K$$

where $n(S_j)$ counts the number of genes with non-null value of the row j. Then, the row S_j is ordered in a descendent way, from the highest expression profiles value to the lower. Finally we choose the first K genes or the highest expression values.

Thus, the discretization is:

$$\begin{aligned} \text{If } e_{j,i} &\in \{\text{First } K \text{ genes in the ordered row } S_j\} \Longrightarrow e_{j,i} \text{ expressed or 1,} \\ \text{If not} &\Longrightarrow e_{j,i} \text{ unexpressed or 0.} \end{aligned}$$

Applying this formula to each row $j = 1, 2, ...m$ we convert E^T into a boolean matrix.

These three measures have to be used carefully, because they are specifically made for Cancer SAGE data, some useful comparisons, advantages and drawbacks for these measures can be found in Ruggero's article [262].

Mean gene expression method

This method [286] is used in the reconstruction of metabolic pathways using supervised learning methods as decision trees. Its goal is constructing a boolean matrix that represents the change of a gene from a temporal state to another state. So, it is the answer to the problem of predictive genes (changes of temporal state) and gene expression measures.

This method consists of calculating for every line of the gene expression matrix E^T he mean of every gene:

$$MeanX_{i,\cdot} = \sum_{j=1}^{m} e_{j,i},$$

then we obtain n thresholds, corresponding to the genes applying this formula to each gene, $i = 1, 2, .., n$. Finally, we compare each expression value $e_{j,i}$ of the matrix to its corresponding mean value threshold $MeanX_{i,.}$. We assign 1 if the expression value is greater or equal than the mean line value and 0 or no expression if not. Thus, the discretization method is:

$$\begin{aligned} e_{j,i} &\geq MeanX_{i,.} \Longrightarrow e_{j,i} = 1 \text{ expressed} \\ e_{j,i} &< MeanX_{i,.} \Longrightarrow e_{j,i} = 0 \text{ unexpressed} \end{aligned}$$

This method can be useful when we want to know if an individual gene is differentially expressed over a period of time. However, here we did not take into account the other genes and the whole process. That, means that even if one gene is very highly or lowly expressed in relation with other genes, it can appear as unexpressed in all the time process.

Fuzzy logic method

This method [322] is an ingenious fuzzy logic to transform the gene expression matrix E^T, into qualitative values using heuristic rules. Its goal is to resolve the problem of saying in an absolute way that a gene is expressed. In reality, a gene is expressed or not in a relative manner, that is compared to something or from a specific point of view. Let us explain the methodology of this discretization method.

The method takes as input the expression measure matrix E^T, where $e_{j,i}$ is the fold change *ratio*, as defined in eq. 8.1 (without *log* operator). Then, it normalizes the gene expression matrix, obtaining only values that are in the interval $[0, 1]$. Following, it defines three possible expression values: Low (L), Medium (M) and High (H). Each gene expression value $e_{j,i}$ is then transcribed by a rule which indicates the percentage of L, M or H. Later on, all these heuristic rules are inserted into a decision matrix. Applying an algorithm (more details in [322]) based in the heuristic rules, we obtain the degree of expression of each value in relation to one state.

In principle this fuzzy method is very promising because it takes into account the relativity of expression of one gene in relation to the others. However, in practice its calculation is time-consuming. The Woong article mentions that it takes more than 10 days to discretize a gene matrix with 6321 genes and 5 conditions. Another big problem is the enormous quantity of possible values of discretization results. Indeed, this fine discretization would be difficult to implement in practice for the most common gene expression data and would be complicate to integrate in supervised learning algorithms.

8.3.4 Discussion

We have summarized several discretization methods applied in different gene expression technologies as microarray and SAGE. We have distinguished as well several applications as metabolic pathways, cancer studies, drugs test etc., of these methods. But what about our initial question: When can we say that an expression profile of a gene or the gene is expressed or not? It did not exist an obvious answer to this. and we have to put it in biological context (type of gene expression technology, biological application, etc.) and also take into account

the mining algorithm that would be applied thereafter. It does not exist an unique discretization method, so we have to choose the appropriate one to our particular interpretation needs. Here, we propose six parameters to considerate before applying one of the methods:

- Gene expression technology: SAGE, RT-PCR, microarray, EST etc.
- Type of state variables: cancer/normal, placebo/medicament, mutated/not mutated, active/passive, etc.
- Type of biological conditions: temporal process, tissues, several time processes etc.
- General biological application of the experiment: temporal process description, cancer studies, testing medicaments, epidemiology studies, mutation detection etc.
- Control of the statistical pre-treatment of the gene matrix expression (missing values, noise, replicates, normalizations etc.).
- Immediate application of the discretization: as input of supervised or non-supervised methods, as an analysis phases in an algorithm etc.
  Taking into account these six parameters will yield a better discretization method choice and will allow us to obtain coherent interpretation results.

8.4 Decision Trees Basics

Decision Trees is one of the most widely-used classification practical algorithm in the machine learning field [212]. Intuitively a decision tree is a classifier constructed by *internal nodes* which specifies a test regarding some attribute of the object., each branch descending from that node corresponding to one of the possible values of this attribute. As determined by the test specified by an internal node, the training examples will be divided and distributed along the descending branches. This splitting process continues until all the training examples pertaining to the node share a common label and are considered "pure". Such nodes are called leaf nodes.

Various decision trees can be built to optimize particular splitting criteria. Some of the most common decision trees algorithms are: C4.5/C5.0 [247], CART [52] and QUEST [189] among others. The main distinctive points among theses algorithms its their splitting criteria and it will be discussed later in this section 8.3.2

In this section we state the framework for decision trees classification in gene expression technologies. Here we mainly emphasize the fifth step of the model building process, i.e. predictive model construction. This section is organized as follows: the first section presents the classification in gene expression technologies framework and statistical decision theory basics, the second section gives an overview of the decisions tree framework and it explains the splitting criterion differences among several decision tree algorithms. Finally it ends with a brief discussion over decision trees important parameters.

8.4.1 Classification and statistical decision theory

8.4.1.1 Classification

Classification is a prediction or learning problem in which the variable to be predicted assumes one of K predefined and unordered values, $\{c_1, c_2, ..., c_K\}$, arbitrarily relabeled by the integers $\{1, 2, ..., K\}$ or $\{0, 1, 2, ..., K-1\}$, and sometimes $\{-1, 1\}$ in binary classification. The K values correspond to K predefined classes, e.g. disease-type, drug-response etc. Associated with each object, S, are: a *response or dependent variable* (class label), $Y \in \{1, 2, ..., K\}$ and a set of n measurements which constitute the attribute vector or vector of predictor variables, $\mathbf{X} = (X_1, ..., X_n)$. The attribute vector $\mathbf{X}$ belongs to a attribute space X, e.g. the real numbers $X \in \mathbb{R}^n$. The task is to classify an object into one of the K classes on the basis of an observed measurement $\mathbf{X} = \mathbf{x}$, i.e., predict Y from $\mathbf{X}$.

A classifier or predictor for K classes is a mapping C from X into $\{1, 2, ..., K\}$, $C : X \rightarrow \{1, 2, ..., K\}$, where $C(x)$ denotes the predicted class for an attribute vector x. That is, a classifier C corresponds to a *partition* of the attribute space X into K disjoint and exhaustive subsets, $I_1, ..., I_K$, such that a sample with attribute vector $x = (x_1, ..., x_n) \in I_k$ has predicted class $\hat{y} = k$ (modifications can be made to allow doubt or outlier classes [256]).

Classifiers are built or *trained* from past experience, i.e., from observations which are known to belong to certain classes. Such observations constitute the *learning set* (LS), $LS = \{(x_1, y_1), ..., (x_n, y_n)\}$. A classifier built from a learning set LS is denoted by $C(\cdot\ , LS)$. When the learning set is viewed as a collection of random variables, the resulting classifier is also a *random variable*. Intuitively, for a fixed value of the attribute vector x, as the learning set varies, so will the predicted class $C(x\ , LS)$. It is thus meaningful to consider distributional properties (e.g. bias and variance) for classifiers when assesing or comparing the performance of different classifiers.

8.4.1.2 Classification for gene expression data

In the case of gene expression data from disease-type experiments (as cancer), attributes correspond to genes X_i which are measured over different biological samples or tumor types S, and the K classes correspond to these biological samples types or tumor types (e.g. nodal positive vs negative breast tumors, tumors with good vs. bad prognosis etc.). Here the predictive problem is the classification of biological samples (malignancies) into known classes.

For this purpose, gene expression data of N genes measured over M tumor samples may be summarized by an expression matrix $E^T : MXN$ (as illustrated in Fig. 8.1). The columns of this matrix, X_i, are gene vectors, containing at each matrix entry the gene expression measure of gene X_i (variable i) of sample S_j, (observation j). Note that this gene expression data matrix is the transpose of the standard $E : NXM$ gene expression matrix used in precedent chapters. The NXM representation was adopted in the microarray literature for display purposes, since for very large N and small M it is easier to display this matrix than the transposed one. The real values of the expression levels must be discretized into several categories (as illustrated in third step of Fig. 8.1). When the biological samples belong to known classes, the data for each observation j consist of a *gene expression profiles* $x = (x_1, ..., x_n)$ and a class label y_j, i.e., of predictor variables x_i and response variable y_i. For

K classes, the class labels y_i are defined to be integers ranging from 1 to K, and M_k denotes the number of learning set observations belonging to class k. Data from gene expression technologies experiments present a so-called "*small* M, *large* N" problem, that is, a very large number of variables (genes) relative to the number of observations (biological samples)

8.4.1.3 Statistical decision theory

It is useful to view classification as a statistical decision theory problem. For each object, an attribute vector $\mathbf{X} = \mathbf{x}$ is examined to decide which class the object belongs to. Assume observations are independently and identically distributed (i.i.d.) from an unknown multivariate distribution. Denote the class k prior, or proportion of objects of class k in the population of interest, by $\pi_k = p(Y = k)$. Objects in class k have an attribute vectors with class conditional density $p_k(x) = p(\mathbf{x} \mid Y = k)$. Defining a loss function L, where $L(h, l)$ simply elaborates the loss incurred if a class h case is erroneously classified as belonging to class l and defining the risk function, $R(C)$, for a classifier C is the expected loss when C is used to classify, that is,

$$R(C) = E[L(Y, C(\mathbf{X}))] = \sum_k E[L(k, C(\mathbf{X})) \mid Y = k]\ \pi_k = \sum_k \int L(k, C(\mathbf{x}))p_k(\mathbf{x})\ \pi_k, \tag{8.11}$$

where $E[L(\cdot, \cdot)]$ is the correspondent expected value for loss function L. Typically, $L(h, h) = 0$, and in many cases the loss is defined to be symmetric with $L(h, l) = 1$, $h \neq l$ making an error of type I (see hypothesis testing section 2.2.1). Then, the risk is simply the *misclassification rate*, $p(C(\mathbf{X}) \neq Y) = \int_{C(\mathbf{X}) \neq k} p_k(\mathbf{x})\pi_k$. Note that here the classifier is viewed as fixed, that is, if a learning set L is used to train the classifier, probabilities are conditional on L. When (unrealistically) both π_k and $p_k(\mathbf{x})$ are known, it is possible to define an optimal classifier which minimizes the risk function. This situation gives an upper bound on the performance of classifiers in the more realistic setting where these distributions are unknown.

8.4.2 Decision trees framework

Decision Trees are structured classifiers constructed by repeated splits of subsets or *internal nodes* of the attribute space X into K descendant subsets, starting with X itself. These *internal nodes* specifies a test regarding some attribute of the object, and each branch descending from that node corresponds to one of the possible values of this attribute. As determined by the test specified by an internal node, the training examples will be divided and distributed along the descending branches. This splitting process continues until all the training examples pertaining to the node share a common label and are considered "pure". Such nodes are called leaf nodes. Each terminal subset is assigned a class label and the resulting partition of X corresponds to the classifier.

There are three main aspects for tree constructions:

1. The splitting rule
2. The decision to declare a node terminal or to continue splitting.
3. The assignment of each terminal node to a class.

We explain these three main aspects taking as examples the binary decision tree construction using two algorithms C4.5/C5.0 [247] and CART [52]. Binary decision trees categorical predictor variables $x_{i,j}$ have only two values, e.g. 1 or -1. This is the case of discretized gene expression values that can be either over-expressed or under-expressed for a given gene i and a particular sample condition j.

8.4.2.1 The splitting rule

The construction of a decision tree (the learning process) involves creation of a hierarchy of tests pertaining to attributes. Various decision trees can be built to optimize a particular splitting criteria. The simplest splits are based on the value of a single variable. The main idea is to split a node so that the data in each of the descendant subsets are "purer" than the data in the parent subset.

A number of definitions are needed in order to provide a precise definition of a node splitting rule.

An *impurity function* is a function $\phi(\cdot)$ defined on the set of all $K - tuplets$ $p = (p_1, ..., p_K)$, with $p_K \geq 0$, $k = 1, ..., K$, and $\sum_k p_k = 1$. This function has the following properties:

- $\phi(p)$ is maximal if p is uniform, i.e. $p_k = 1/K \ \forall \ k$.
- $\phi(p)$ is zero if p is concentrated on one class, i.e. $p_k = 1$ for some k.
- $\phi(p)$ is symmetric in p, i.e., invariant to permutations of the entries p_k.

For a node t, let $n(t)$ denote the total number of learning set cases in t and $n_k(t)$ the number of class k cases in t. For class priors π_k, the resubstitution estimate of the probability that a case belongs to class k and falls into node t is given by $\hat{p}(k,t) = \pi_k n_k(t)/n_k$. The resubstitution and the resubstitution estimate that a case at node t belongs to class k is $\hat{p}(k \mid t) = \hat{p}(k,t)\,\hat{p}(t)$. When data priors $\pi_k = n_k \,/\, n$ are used, $\hat{p}(k \mid t)$ is simply the relative proportion of class k cases in node t, $n_k(t) \,/\, n_k$. So we define the impurity measure $i(t)$ of node t by

$$i(t) = \phi\left(\hat{p}(1 \mid t), ..., \hat{p}(K \mid t)\right). \tag{8.12}$$

Having defined node impurities, we are now in a position to define a *splitting rule*. Suppose a split s of a parental node t sends a proportion p_R of the cases in t to the right daughter node t_R and p_L to the left daughter node t_L. Then, the goodness of split is measured by the decrease in impurity

$$\triangle i(s,t) = i(t) - p_R i(t_R) - p_L i(t_L). \tag{8.13}$$

The split s which provides the largest improvement $\triangle i(s,t)$ is used to split node t and called the *primary split*. Splits that are nearly as good as the primary split are called *competitor splits*. Finally, *surrogate splits* are defined as splits that most closely imitate the primary splits. Informally, "imitation" means that if a surrogate split is used instead of a primary split, the resulting daughter nodes will be very similar to the ones defined by the

primary split. The split that provides the best agreement of the two sets of daughter nodes is called the first surrogate split.

Some of the most common splitting rules are based in two impurity functions $\phi(\cdot)$, [10] that are:

The *entropy** impurity measure is defined as:

$$\phi_E(p) = -\sum_k p_k \log p_k \tag{8.14}$$

where 0 if $log(0) \equiv 0$.

The *Gini index* impurity measure is defined as:

$$\phi_G(p) = -\sum_{k \neq l} p_k p_l = 1 - \sum_k p_k^2. \tag{8.15}$$

The *entropy* measure is employed in the widely-used C4.5/C5 algorithms introduced by Quinlan [247] and the *Gini index* measure was used by Breiman et al. [52] in the original version of their classification and regression tree (CART) methodology.

In order to define the splitting rule of these two algorithms, we first define the needed probabilities as: p and q that are the proportions of observations going to the left and right cases of the tree; p_L and q_L are the proportions of 1's and -1's in the left sided case, and p_R and q_R are the proportions of 1's and -1's in the right sided case, respectively.

By replacing the entropy impurity measure $\phi_E(p)$ of equation 8.14 into the splitting rule $\triangle i(s,t)$ in equation 8.13 we obtain the objective function used in C4.5/C5 algorithms:

$$SC_E = \triangle i_E(s,t) = q(-q_R \log q_R - p_R \log p_R) + p(q_L \log q_L - p_L \log p_L) \tag{8.16}$$

In a similar way, by replacing the *Gini index* impurity measure $\phi_G(p)$ of equation 8.15 into the splitting rule $\triangle i(s,t)$ in equation 8.13 we obtain the objective function used in CART algorithm:

$$\triangle i_G(s,t) = q q_R p_R + p q_L p_L \tag{8.17}$$

At each splitting step the minimum of the objective functions eq. 8.16 and eq. 8.17 gives the primary split s to split node t in C5 and CART algorithms respectively. This primary split represents the largest improvement $\triangle i(s,t)$.

8.4.2.2 The decision to declare a node terminal

In both of the explained algorithms C4.5/C5 and CART the tree is fully constructed by taking the primary splits as explained before using the same criterion to declare a node terminal. The decision to declare a node terminal is when all cases on that node are of the same class, otherwise you continue to split.

This stage may also concerns obtaining the "*best-sized*" (see [52]) tree and accurate estimates of classification error (as illustrated in steps 5 and 6 of the FIG. 8.1). The correct-sized tree is obtained by cutting and replacing branches from the tree (pruning) in order to reduce the classification error. For estimating the classification error we can use any of the 20 methods explained in the evaluation accuracy prediction section.

One of the most used pruning method is the following: Once the large decision tree is grown, it is selectively *pruned upward*, yielding a decreasing sequence of subtrees. If a given branch has a higher classification error rate (estimated using the test set) than a simple leaf would, the branch is replaced with a leaf. By applying this heuristic rule from the bottom to the top of the tree, you prune back the tree for better future prediction. Cross-validation is used to identify the subtree having the lowest estimated misclassification rate.

Either C4.5/C5 or CART uses an *upward pruning method* for determining the optimal-sized tree and *cross-validation* for calculate the estimate of classification error.

8.4.2.3 Class assignment rule.

This step concerns the determination of the class at the terminal node when the stop criterion is not achieved. For each terminal node, choose the class that minimizes the estimate of the misclassification probability, given that a case falls into this node. Note that given equal costs and priors, the resulting class is simply the majority class in that node. Either C4.5/C5 or CART uses this class assignment rule criterion.

Unfortunately the problem of finding the most compact tree is known to be NP complete* [156]. Constructing an optimal binary decision tree is NP-complete. In a recent study Lim et al. [181] compare twenty-two different decision tree algorithms in terms of classification accuracy, training time and number of leaves. Among decision trees algorithms with univariate splits, C4.5/C5.0 [247], CART [52] and QUEST [189] have the best combinations of error rate and speed. For the GENETREE algorithm we have chosen the Quinlan's C4.5/C5 algorithm because of this reason and because the entropy splitting criterion has a mathematical interpretation.

8.5 GENETREE algorithm principles

Biological sample prediction is one of the most exciting areas to which gene expression technologies are currently applied. This issue concerns the main goal of several bioinformatics applications, such as determining the disease state of a tissue, effectiveness of a medicament, and toxicity of a molecule.

GENETREE is a decision tree classifier for biological sample prediction (as disease-type or drug-response etc.) integrating the information contained in gene expression profiles obtained from gene expression technologies data, and gene annotations issued from several biological knowledge sources, as explained in chapter 3.

Our algorithm takes advantage of the well-known decision tree algorithm C4.5/5.0 proposed by Quinlan [247] and extends the modified entropy splitting criterion to a richer one using the knowledge contained in several sources of biological information. GENETREE splitting criterion is composed of several independent criteria: the modified C4.5 Quinlan's entropy criterion (SC_E) applied to gene expression discretized profiles and four additional criteria, SC_1, SC_2, SC_3 and SC_4defined from gene annotations.

GENETREE intends to build decision tree models containing a set of consistent and functionally heterogeneous predictive genes specially chosen to achieve better class prediction.

These characteristics may help to avoid the problem of "imperfect disease diagnosis because of a battery of expression-based predictors for various cancers (classes)" as stated by Golub in [133].

Additionally, GENETREE methodology proposes the utilization of two novel original algorithms: a sample selection methodology (explained in section 4.1.3, for more details see [199]) and normal discretization algorithm (NORDI) specially designed for gene expression data.

As a supervised learning approach applied to gene expression technologies, GENETREE, follows the seven steps process for building predictive models: class discovery, sample and gene selection, discretization, data partition, predictive model choice, accuracy prediction evaluation and refining the model, and biological interpretation of the model (as illustrated in Fig. 8.1). Here, we explain in detail each one of these seven steps and we finish the section with the GENETREE pseudocode.

8.5.1 Class discovery

In section 8.3 we have defined the predictive bioinformatics main goal as the classification of biological samples S into K known classes (see section 8.4.1.2). In the case of gene expression data, attributes correspond to genes X_i which are measured over different biological samples S_j, and classes K correspond to these biological samples types. All the notation details concerning this prediction problem in gene expression technologies where fully developed in section 8.4.1.2 and section 8.4.1.1.

For the sake of brevity we suppose that the biological classes K are known. That is the case of most of the gene expression data sets specially built for prediction purposes, where the classes are already known biological samples. Indeed, if the classes are not known or we may want to explore the existence of more classes, we need to apply one of the class discovery algorithms explained in section 8.2, in order to find the biological classes.

8.5.2 Data selection

This is a critical stage for the success of the predictive power of the model. It concerns sample selection and selecting informative genes.

GENETREE uses a novel sample selection methodology (explained in section 4.1.3, for more details see [199]) and it suggests the utilization of a problem-adapted methodology for choosing informative genes.

Unfortunately gene selection is a necessary step for building a predictive model, because of the dimensionality problem of low number of biological samples and big number of genes.

Gene reduction is a necessary step and we suggest the utilization of algorithms that takes into account not only the genes that are differentially expressed, but also the algorithms which takes into consideration the pertinence of the selected genes with their inherent class as *gene-pairs* [38], *virtual genes*, [325] and Li et al. [179] approaches (see section 8.3).

8.5.3 Discretization

GENETREE uses a novel discretization methodology NORDI (normal discretization), specially designed to gene expression technologies (for more details see section 8.4.3).

NORDI is based on statistical detection of outliers and the continuous application of normality tests for transforming the initial distribution "almost normal" to a "more normal" one. Once the distribution of the data set is "more normal" it calculates the cutoffs as seen in $z - score$ methodology (section 8.4.2).

The number of discretization intervals can be fixed by the analyst given the characteristics of the studied data set. We suggest the utilization of several discretization scenarios to validate the prediction model results (for a more detailed discussion on discretization issues see section 8.4.4).

Implementation

The NORDI discretization algorithm was implemented using the R language and using several libraries of the BIOCONDUCTOR open source project.

8.5.4 Data partition

GENETREE uses the cross-validation method. This statistical technique of partitioning a sample of data chooses s out of n samples as the training set and estimates its error rate using $n - s$ sample observations (test set). This process is repeated for all distinct choices of s patterns, constructing $\binom{n}{s}$ classifiers, and finally the average of the error rate, known as $s - fold$ estimate, is computed.

This choice is done because the gene expression data set contains the explained "big N and small M " dimensionality problem, where there are thousands or hundreds of attributes or genes and tens of objects or samples. Thus, a way to obtain meaningful prediction results via decision trees technique is the use of cross validation, as shown with the C4.5/C5 algorithm [247].

8.5.5 Choosing and building the prediction model

For GENETREE algorithm we extended Quinlan's C4.5/5.0 algorithm, modifying the splitting rule (explained in section 8.3). In this section, we first explain different gene measures based on their biological annotations, then, we develop the main aspects for GENETREE construction: splitting rule, decision to declare a node terminal, and assignment of each terminal node to a class, we finish by illustrating the splitting GENETREE step with an example. In this section; we make special emphasis in the proposed gene measures based on gene annotations and the novel GENETREE splitting rule aspects, which integrates biological knowledge and gene expression profiles.

8.5.5.1 Gene measures based on gene annotations

First, we define the following gene measures $M_1(\cdot)$, $M_2(\cdot)$, $M_3(\cdot)$ and $M_4(\cdot)$ concerning gene annotations.

Proximity between a gene and the goal-classes $M_1(X_i, GC)$

The goal classes are all the possible K classes without taking into account the reference class (e.g. c_1=normal, c_2=leukemia A and c_3=leukemia B, c_2 and c_3 are the goal classes). In order to avoid confusion, we define the set of goal-classes as the set $GC = \{c_{2,}..., c_K\}$, where the class c_1 is taken as the reference class, so it doesn't enters in the set of goal-classes.

Let's define $\rho_1(X_i, c_k)$ as a correlation measure of annotations concerning gene X_i and the c_k goal-class. This correlation measure, $\rho_1(X_i, c_k)$, determines the degree of closeness between the goal-relative gene annotations of gene X_i and the goal-class c_k and it is bounded by $0 \leq \rho_1(X_i, c_k) \leq 1$.

Indeed, for a given gene, X_i, we are interested in a correlation measure that takes into account all possible goal-classes, so we define this measure as the average among all the respective correlation measures $\rho_1(X_i, c_k)$, i.e.:

$$M_1(X_i, GC) = \frac{\sum_{k=2}^{K} \rho_1(X_i, c_k)}{K-1}. \tag{8.18}$$

$M_1(X_i, GC)$ is an average measure that determines the degree of closeness between an specific gene X_i and all the possible goal-classes. Taking the leukemia example described above, if we have a gene X which has one annotation concerning the induction of leukemia B and no annotation for leukemia A. Their $M_1(X_i, GC) = (100 + 0)/2 = 50$ is their average measure of closeness.

Proximity between a gene and the extended goal-classes: $M_2(X_i, EGC)$**.**

An extended goal-class, ec_k^r, is a closely-related class r (diseases, toxics, medicaments etc.) of the goal-class k. We take the set of extended goal-classes of goal-class c_k as $EGC_k = \{ec_k^1, ec_k^2, ..., ec_k^R\}$. For example in the last example the extended goal-class for leukemia A could be any type of cancer. The degree of closeness among related goal-classes and the goal-classes will be determined taking into account the specific problem characteristics.

The proximity between a gene X_i and the R closely-related classes of class, c_k, is defined by equation 8.18 as:

$$M_1(X_i, EGC_k) = \frac{\sum_{r=1}^{R} \rho_1(X_i, ec_k^r)}{R} \tag{8.19}$$

Hence, taking into account the correlation measure $M_1(X_i, EGC_k)$ for all possible goal-classes c_k, we define the proximity between a gene X_i, and all the extended goal-classes as:

$$M_2(X_i, EGC) = \frac{\sum_{k=1}^{K-1} M_1(X_i, EGC_k)}{K-1}. \tag{8.20}$$

$M_2(X_i, EGC)$ is an average measure that determines the degree of closeness between an specific gene X_i and all the possible extended goal-classes. Taking the leukemia example described above, if we have a gene X and 10 related leukemia diseases. Supposing that gene X has annotations stating that the presence of gene X induces 7 of this related-leukemia disease, their average measure is equal to: $M_1(X_i, EGC) = (7 * 100 + 0)/10 = 70$ is their average measure of closeness between the gene X and ten related leukemia diseases..

Functional proximity between two genes according GO: $M_3(X_i, X_j)$ and $M_4(X_i, X_j)$

In order to calculate a measure of the functional similarity, according to Gene Ontology (GO), between two genes, X_i, X_j, we choose the so called *"principal distance"* proposed in [175], which is a modified nodal distance that use a directed acyclic graph (DAG) model for GO. This distance takes into account the hierarchical structure of GO and computes the distance between two gene annotation within the GO tree. The GO tree $T_G = (V_C, E)$ is composed by the set of nodes (GO terms) V_C and the set of edges E.

The principal distance is a metric on GO tree which measures the closeness between two GO terms (gene GO annotations) and is defined as follows:

Definition: Suppose that $v1$ and v_2 are two nodes (GO terms) in the GO tree: $T_G = (V_C, E)$[37] where

$$Pd(v1, v2) = \begin{cases} 0 \text{ if } v1 = v2, \\ W(\omega_0) \text{ otherwise,} \end{cases} \tag{8.21}$$

and the weight $W(t)$ of level t is defined as a function $W : I_H \to \Re^+$ such that $W(i) > W(i+1)$ and $I_H = 1, 2, 3, ..., H$. The weight function define the following parameters:

- H_0 is the height of T_G
- $H = H_0 + 1$.
- ω_0 is the lowest common ancestor of v_1 and v_2

Given a GO code[38] for v_i we use $W(v_i)$ in place of W(level of v_i) for notational convenience (the current modeling of GO tree, supposes the parameters $H = 15$ and $W(k) = 150 - 10(k-1)$ for $k \in I_H$). For example, the principal distance of $Pd(C_1, D_1) = W(C_1)$ and $Pd(C_3, E_2) = W(A_1)$ (see FIG. 8.2).

The principal distance defined above, $Pd(v_1, v_2)$, takes into account the importance of each annotation in the hierarchical GO structure. For example, in FIG. 8.3, we can see than the common distance, counting the branches between two nodes, of $d(B_1, B_2) = d(B_1, D_1) = 2$. Even if these two nodes above has the same path length, their relationships are quite different from each other. It is likely that B_1 and D_1 are more closely related (B_1 ancestor of D_1) than B_1 and B_2. Computing their principal distance we have $Pd(B_1, B_2) = W(A_1)$ and $Pd(B_1, D_1) = W(B_1)$, by the hierarchical structure of GO tree, we know that $W(B_1) < W(A_1)$. Thus, the principal distance takes into account the hierarchical structure of GO (see FIG. 8.3).

This distance can also be defined in an algebraic way by using GO codes. Let $\mathbb{N}_0$ be the set of natural numbers including zero. Then, given two GO codes $v_1 = a_1 a_2 \cdots a_H$ and $v_2 = b_1 b_2 \cdots b_H$ with $a_i, b_i \in \mathbb{N}_0$,

$$Pd(v_1, v_2) = \begin{cases} 0 \text{ if } a_i = b_i \text{ for all } i, \\ W(L) \text{ otherwise,} \end{cases} \tag{8.22}$$

[37] DAG model from [175]

[38] GO code is the Gene Ontology codification of each GO Term. For example the GO term $GO : 0016265$: death is the fifth child of biological process $GO : 0008150$. This can be seen in the GO code, the code for death is $250X10^{12}$ and for biological process$200X10^{12}$.

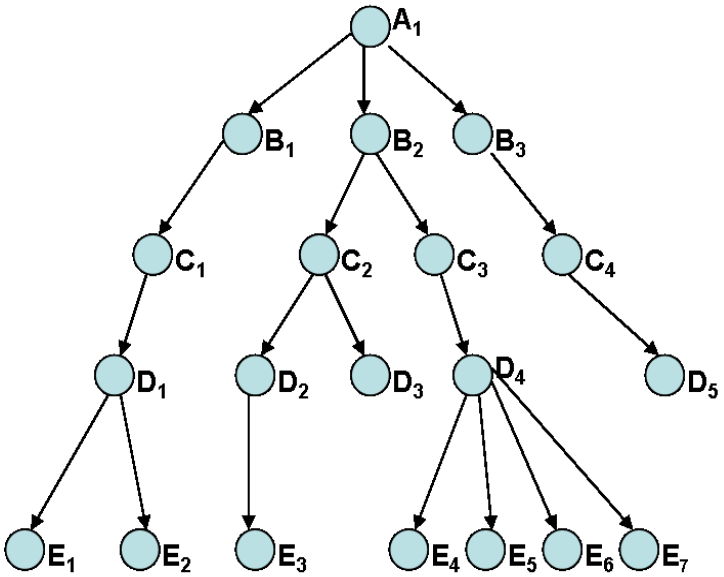

FIG. 8.3: Metric relationship of GO. The levels of the correspondent gene annotations (GO terms): A_i, B_i, C_i, D_i, and E_i are 1, 2, 3, 4 and 5 respectively.

where $L = \max_{1 \leq i \leq H} \{i \mid a_i = b_i\}$.

It is shown in [175] that the distance function Pd, eq. 8.21 or eq. 8.22, is a metric on the set V_C of all GO codes. Hence, the distance $Pd(v_1, v_2)$ has the reflexive, symmetric and transitivity properties (for more details see [175]).

Here, we define two measures $MaxPd$ and $AverPd$ based on the metric $Pd(v_1, v_2)$ explained before. First, we explain the notion of *multiset*. If we have the following three sets $\{1\}$, $\{1, 1\}$ and $\{1, 1, 1\}$ are equal in the set notation. Yet, if we want to take the number of occurrence of elements into account. In that case, such set is called as a *multiset*. Given a multiset $M = \{v_1, v_2, ..., v_n\}$ of GO codes in GO tree.

$MaxPd$ is defined as the maximum value of principal distances between any two elements in M, i.e.:

$$MaxPd(M) = \max_{1 \leq i \prec j \leq n} \{Pd(v_i, v_j)\} \tag{8.23}$$

and

$$AverPd(M) = \sum_{1 \leq i \prec j \leq n} \frac{Pd(v_i, v_j)}{Cons} \tag{8.24}$$

where $Cons = \frac{n(n-1)}{2}$.

Let X_a and X_b be a pair of genes that are annotated with GO terms, these terms are directly related to GO codes (as shown before). Assuming the gene X_a has s_i annotations and the gene X_b has s_j annotations, where $s_i + s_j = n$. The resulting multiset containing both of the s_i and s_j annotations is $M = A_{X_a, X_b} = \{v_1, v_2, ..., v_n\}$. Using the equations: 8.23 and 8.24 we define two gene measures:

$$M_3(X_a, X_b) = MaxPd(A_{X_a, X_b}) \tag{8.25}$$

and

$$M_4(X_a, X_b) = AverPd(A_{X_a,X_b}). \tag{8.26}$$

The measure $M_3(X_a, X_b)$ gives the lowest common ancestor (LCA) between the genes X_a and X_b. If the LCA is located at higher levels of the gene ontology tree, T_G (levels 1 or 2), the genes contain heterogeneous annotations. Sometimes the resultant GO code from $M_3(X_a, X_b)$ may therefore be placed at relatively higher levels on account of just one false positive. While this might be bad because it is not flexible, it can be also considered good because it informs us of the existence of some functional outliers.

The measure $M_4(X_a, X_b)$ gives the most frequent GO codes between the genes. In other words, the GO codes at which the two genes are concentrated in the GO space. This measure tries to infer the strongest meanings of the GO gene annotations from its most concentrated subcluster and hence it does not concern a few functional outliers in that cluster. If this measure has relative low values, means that they exist a strong concentration of GO annotations between the gene X_a and X_b, in other words they have homogeneous GO annotations. For further details in calculating these two measures the reader can refer to Lee et al. [175].

8.5.5.2 GENETREE splitting rule

The GENETREE splitting rule is defined at each node t and it consists in two main parts: choosing the root node and choosing the subsequent nodes. The notation is the same as the one stated in section 8.4. Let's see each of these parts in detail.

Choosing the root node

In order to choose the root node we compute the entropy splitting criterion, $SC_E = \triangle i_E(s, 1)$, (C4.5/5.0 algorithm) [247] of each gene X_u for $u = 1, ..., n$, obtained from the objective function 8.16. Then, we order the genes decreasingly by taking their information gain $IG_s^1(X_u) = \triangle i_E(s, 1)$ (As we have stated in section 8.4, the minimal of SC_E corresponds to the maximal information gain). We take the maximal information gain gene: $X^1max = \max IG_s^1(X_u)$ for $u = 1, ..., n$. Then, we look at the following (in decreasing order) genes X_i with a distance $\left|X^1max - IG_s^1(X_u)\right| < \delta$. Thus, we obtain the set of possible root node genes G_s^1, containing all the genes at a relative distance δ of the maximal information gene X^1max. In the case that G_s^1 contains more than one gene we compute the proximity between each gene in this subset and the goal-classes $M_1(X_u, GC)$, using eq. 8.18. Then, we order the genes in a decreasing way, according to the measure M_1, in a list $L1M_1$. We then choose the top gene of this list, i.e. with maximal M_1, as the root node $t = 1$. In the case of equality between two or more genes in the top of the $L1M_1$ list, we then compute $M_2(X_u, EGC)$ for each of these genes, and in a similar way we build the list $L1M_2$, an we order it decreasingly. Then, we choose the top gene of the list (with maximal M_2) as the root node $t = 1$. In the case of equality we choose the top gene in the list $L1M_2$as the *i'th* node (gene X^i) of the tree.

Consequently, the root node is the gene that maximizes the information gain and it may be directly related with the goal-classes, or even more with the extended goal-classes. This means that is the "best" discriminant between classes taking into account the entropy

information gain criterion applied to gene expression measures, and it may also contain determinant class-related or even extended class-related biological information

Choosing the subsequent nodes of the tree

Here, we explain the general procedure for choosing the subsequent nodes t of the tree. We suppose the subsequent node (gene) is $t = i$ and the set of possible nodes (genes) contains $n - (i - 1)$ genes, taking out the previously selected $i - 1$ nodes (genes), i.e. genes X^1, $X^2, ..., X^{i-1}$.

As well as for the root node, we first compute for each of the remaining $n-(i-1)$ genes their information gain $IG_s^i(X_u)$ using SC_E criterion and we order them decreasingly. We take the maximal information gain gene: $X^imax = \max IG_s^i(X_u)$ for $u = 1, ..., n-i+1$. Then, we look at the following (in decreasing order) genes X_u with a distance $\left|X^imax - IG_s^i(X_i)\right| < \delta$. Thus, we obtain the set of possible node genes G_s^i, containing all the genes at a relative distance δ of the maximal information gene X^imax. We suppose the cardinality of this set is $\left\|G_s^i\right\| = z$.

Then, we compute the average of the functional proximity measures $M_3(X_w, X_t)$ and $M_4(X_w, X_t)$, applied between each gene $X_w \in G_s^i$ and each one of all the previous nodes $t = 1, ..., i-1$. The average functional proximity measures between the gene, X_w, and the previously selected t nodes (genes X^t) respective to the measure $M_3(X_u, X_t)$ is defined as:

$$AverM_3(X_w, X^t) = \sum_{t=1}^{i-1} \frac{M_3(X_w, X^t)}{i-1}. \tag{8.27}$$

In a similar way the average functional proximity measures between the gene, X_w, and the previously selected t nodes (genes X^t) respective to the measure $M_4(X_w, X_t)$ is defined as:

$$AverM_4(X_w, X^t) = \sum_{t=1}^{i-1} \frac{M_3(X_w, X^t)}{i-1}. \tag{8.28}$$

For each of the genes $X_w \in G_s^i$, we make the following two comparisons:

- If $AverM_3(X_w, X^t) \geq \epsilon$
- If $AverM_4(X_w, X^t) \leq \zeta$

If these two comparisons are true then we take out the gene X_w of the set G_s^i, if not we leave the gene on the set G_s^i. For given parameters ϵ and ζ, if these two comparisons are true, it means that the gene X_w is, in average, functional similar concerning GO annotations to the previously selected genes X^t in the arborescence.

Thus, we have a new set of heterogeneous genes HG_s^i at an information gain distance $\left|X^imax - IG_s^i(X_i)\right| < \delta$. In the case that HG_s^i contains more than one gene we compute the proximity between each gene in this subset and the goal-classes $M_1(X_w, GC)$, using eq. 8.18. Then, we order the genes in a decreasing way, according to the measure M_1, in a list LiM_1. Then, we choose the top gene of this list, i.e. with maximal M_1, as the *i'th* node (gene X^i) of the tree. In the case of equality between two or more genes in the top of the LiM_1 list, we then compute $M_2(X_w, EGC)$ for each of these genes, and in a similar way we build

a decreasing order list LiM_2, an we order it decreasingly. We then choose the top gene of the list (with maximal M_2) as the $i'th$ node (gene X^i) of the tree. In the case of equality we choose the top gene in the list LiM_2 as the $i'th$ node (gene X^i) of the tree.

Consequently, each subsequent node X^i is a gene that maximizes the information gain, it is not functional similar to the previously selected genes in the tree X^t, and it may be directly related with the goal-classes, or even more with the extended goal-classes. This means that it is a "good" discriminant between classes which takes into account the entropy information gain criterion applied to gene expression measures, it is functionally heterogeneous in relation with the previously selected genes, and it may also contain determinant class-related or even extended class-related biological information.

8.5.5.3 Decision to declare a node terminal

GENETREE fully constructs the tree by taking the splits s determined splitting criteria explained in the last section and building a tree containing $X^1, X^2, ...$ nodes until the criterion to declare a node terminal. As stated in Quinlan's C4.5/C5 algorithm, the decision to declare a node terminal is when all cases on that node are of the same class, otherwise we continue to split.

This stage may also concerns obtaining the "*best-sized*" (see [52]) tree and accurate estimates of classification error (as illustrated in steps 5 and 6 of the Fig. 8.1).

In order to tackle this issue, GENETREE algorithm proposes the commonly used upward pruning method (which yields a decreasing sequence of subtrees) combined with 10-fold cross-validation to identify the subtree having the lowest estimated misclassification rate.

This choice is done because the gene expression data set contains the explained "big N and small M" dimensionality problem, where there are thousands or hundreds of attributes or genes and tens of objects or samples. Thus, a way to obtain meaningful prediction results via decision trees technique is the use of cross validation, as stated in C4.5/C5 algorithm.

8.5.5.4 The assignment of each terminal node to a class

This step concerns the determination of the class at the terminal node when the stop criterion is not achieved. For each terminal node, choose the class that minimizes the estimate of the misclassification probability, given that a case falls into this node. Note that given equal costs and priors, the resulting class is simply the majority class in that node. This criterion is the same as the used in the Quinlan's C4.5/C5 algorithm.

8.5.6 Prediction accuracy evaluation and refining the model

GENETREE choose the cross-validation methodology for obtaining meaningful prediction results (as stated before). It proposes the utilization of the s*mooth modification of error counting* as error function criterion (explained in section 8.2.6). The main reason is that all our data set, the training data set and the test data set is labelled, so it is a reasonable estimator of the misclassification probability. Cross-validation combined with smooth modification of

error counting function will be used for assessing the prediction accuracy of GENETREE model.

Concerning the refining model step, it is one of the most delicate steps in gene expression technologies, because of the number of important parameters contained in gene expression analysis.

Among these parameters we can mention the parameters concerning directly the decision tree model construction as:

- Optimal size of the tree.
- The number of trees to built.
- The splitting rule parameters: δ, ϵ, and ζ.
  Other important parameters to take into account are the ones concerning the whole prediction model process as:
- Number of samples.
- Number of "informative genes".
- Available sources of biological knowledge.
- Number of classes
- Parameters concerning the efficiency of the obtained classes on the class discovery step.
- Parameters concerning the whole statistical pretreatment of the original data set.

We suggest to use a validation set on which the impact of tuning parameters, at least for the direct decision tree construction parameters, can be objectively judged and optimized.

8.5.7 Biological interpretation of the prediction model results

The relevance of most of the prediction results (disease-prediction, drug-response prediction, molecule toxic-prediction etc.) in health issues, requires a predictive confidence of almost 100%. Thus, the expert needs to interpret as better as possible the predictive model outputs using the available sources of biological knowledge concerning their specific problem.

This is not an easy task, and most of this step interpretation is done by hand by the expert.

GENETREE integrates several sources at information such as GO ontology and disease-related databases for calculating four measures M_1, M_2, M_3 and M_4 concerning gene annotations. This measures can be used once again for analyzing some of the characteristics of the resulting predictive cluster of genes: $PG = \{X^1, X^2, ...\}$.

In order to analyze the cluster characteristics using only the gene expression profiles information, we can use one of the cluster validity measures (homogeneity, separation and consistency) explained in section 2.3.6.

For the analysis of the cluster numeric and biological characteristics we can use several approaches as Quality Tool [128], EASE [151], THEA [228], Graph Theoretic Modeling [175] and GENERATOR [234] among many others.

In order to analyze the cluster characteristics through a biological point of view, several related biological source tools can be used (see chapter 4).

8.5.8 Example of application of GENETREE splitting rule

In order to manually exemplify the novel splitting rule feature of GENETREE algorithm we have chosen an artificial example, inspired in the real Golub et al. data set [133]. One of the main goals of the Golub study was the assignment of particular tumor samples to already-defined classes: acute myeloid leukemia (AML) and acute lypmphoblastic leukemia (ALL), reflecting current states or future outcomes. Distinguishing ALL from AML is critical for successful differentiate leukemia treatment as chemotherapy containing corticosteroids for ALL patients and containing cytarabine. In-situ DNA chips were used to monitor the expression of 6817 human genes over 38 bone marrow samples (27 ALL, 11 AML).

From this data set we have chosen at random 4 informative genes and 5 samples (3 ALL, 2 AML). We have taken the entire data set as training set and we have applied NORDI discretization algorithm (see FIG. 8.2) to each biological sample S_i for $i = 1, 2, 3, ...5$. We have calculated the over-expressed (Ot) and under-expressed (Ut) threshold in order to discretize the selected data set into two values: 1 for gene over-expression and -1 for gene under-expression (see TABLE 8.1).

Samples S_i	*Genes* X_j				*Classes* c_k
	X_1	X_2	X_3	X_4	
S_1	1	1	1	1	AML
S_2	1	-1	-1	1	ALL
S_3	-1	-1	1	1	AML
S_4	1	-1	-1	-1	ALL
S_5	-1	1	1	-1	ALL

TABLE 8.1: Selected leukemia data set containing the discretized gene expression values of each gene over every AML or ALL sample.

TABLE 8.1 shows the selected leukemia data set L, which contains five biological samples, S_i, distributed in two classes, c_k, corresponding to 3 ALL and 2 AML, and four human informative genes: $X_1 = HOXA9$, $X_2 = Zyx$, $X_3 = CD11c$ and $X_4 = CD33$. Each numeric matrix cell corresponds to the expression (over-expression=1 and under-expression=-1) of gene X_j over a given biological sample S_i. The last column represent the correspondent leukemia classes AML and ALL (in this case there are two goal-classes and no reference class).

The numerical information needed for applying GENETREE splitting set of criteria for selecting the root node $t = 1$ and the subsecuents nodes $t = i$ is presented in TABLE 8.2.

In order to calculate the first gene node, X^1, we calculate the information gain of all the genes for $u = 1, 2, 3, 4$. For $\delta = 0$, genes X_3 and X_4 have the same information gain for the first node ($t = 1$), i.e. $IG_s^1 = .42$ (see TABLE 8.2). Thus, we use the proximity between each gene and the goal-classes , $GC = \{ALL,\ AML\}$, criteria, M_1. We compute $M_1(X_3, GC)$ and $M_1(X_4, GC)$ using the eq.8.18 and taking the gene annotations from the article of Golub et al. [133]. Golub states that: $X_4 = CD33$ is directly related with goal classes, so $M_1(X_4, GC) = 100$ and that $X_3 = CD11c$ it may be related with goal classes,

Genes X_u	IG_s^1 at $t=1$	IG_s^2 at $t=2$	$M_1(X_i, GC)$	$M_2(X_i, EGC)$	$M_3(X_i, X^1)$ *GO level*	$M_4(X_i, X^1)$ *Avg(GOcode)*
$X^2 = X_1$	.02	.55	0	100	1	125
X_2	.02	.55	100	50	3	33
X_3	.42	.55	50	100	3	52
$X^1 = X_4$	.42	-	100	50	-	-

TABLE 8.2: Extract of the splitting information process containing all the needed splitting criteria: IG_s^i, $M_1(X_i, GC)$, $M_2(X_i, EGC)$, $M_3(X_i, X^1)$ and $M_4(X_i, X^1)$ for selected leukemia data set.

so we fix this correlation measure to $M_1(X_3, GC) = 50$ (see TABLE 8.2). We have that $M_1(X_4, GC) > M_1(X_3, GC)$, so the root node chosen by our algorithm is $X^1 = X_4$. At this stage one of the branches of the tree contains all cases on that node on the same class, so it is a terminal node and we continue the splitting process from the remaining branch of root node X^1.

To obtain the subsequent node, we calculate the information gain for all the remaining genes for $u = 1, 2, 3$. Taking $\delta = 0$, we have that all the remaining genes X_1, X_2 and X_3 have the same information gain for the subsequent node ($t = 2$), i.e. $IG_s^2 = .55$ (see TABLE 8.2). In the case of the subsequent node, the second criterion for the splitting rule is obtained from two comparisons $M_3(X_w, X^t) \geq \epsilon$ and $M_4(X_w, X^t) \leq \zeta$ (it is not necessary to compute the average of this measures because $t = 2$). Each gene X_w is obtained from the set of possible node genes, i.e. $G_s^2 = \{X_1, X_2, X_3\}$ and the thresholds were fixed to $\epsilon = 3$ and $\zeta = 75$.

In order to calculate the corresponding functional proximity between two genes according to Gene Ontology M_3 and M_4 measures, we search for all GO annotations corresponding to each of the concerned genes (see TABLE 8.3).

Gene (X_i) : GO Terms
X_1 : multicellular organ development
X_1 : cell adhesion
X_2 : cell-cell signaling
X_2 : signal transduction
X_3 : cell adhesion
X_3 : organ morphogenesis
X_4 : signal transduction
X_4 : regulation of cell proliferation
X_4 : cell-cell signaling

TABLE 8.3: GO terms associated to each one of the genes X_u in the selected data set.

Using the gene-paired annotation, we compute the *Max* measure M_3 and the *Average* measure M_4 applying the eqs. 8.25 and 8.26 respectively. The corresponding results for $M_3(X_i, X^1)$ and $M_4(X_i, X^1)$ for $i = 1, 2, 3$ are presented in the sixth and seventh column of TABLE 8.2. The M_3 measure is presented in terms of the GO levels that is obtained by the correspondent resulting GO code and the measure M_4 is in $Avg(GOcode)$ form. For example, for calculating $M_3(X_1, X^1 = X_4)$, we first define the multiset of annotations $M = \{v_1, v_2, ..., v_5\}$, where v_1 ="multicellular organ development", v_2 ="cell adhesion" and so on.

Then we compute all the combinations of possible principal distances $Pd(v_i, v_j)$ using the eq. 8.23. Finally we take the maximum of the resulting set, obtaining $M_3(X_1, X_4) = 125$ as shown in TABLE 8.2.

Once the M_3 and M_4 were computed for all genes, we eliminate (only at this stage) the genes that are functionally closed related with X_4. In this case X_2 and X_3 were eliminated, thus the chosen gene at node $t = 2$ was gene $X^2 = X_1 = HOXA9$.

At this stage each of the two branches of the tree contains all cases on the node on the same class, so both of them are terminal nodes. The predictor set obtained by GENETREE is $\{CD33, HOXA9\}$

Biological insights of the results

Concerning the initial four genes, Golub et al. [133] make several asseverations.

- $CD11c$ and $CD33$ are two highly predictive genes which encode cell surface proteins for which monoclonal antibodies.
- $CD11c$ and Zyx both concern the cell adhesion of the cell.
- $HOXA9$ is a known oncogene, i.e. which increases the malignancy of a tumor cell.
- $HOXA9$ shows the most highly correlation with AML outcome. This predictive force hypothesis needs to be tested in further studies.

From this conclusions stated in [133], we can resume the predictive importance of $HOXA9$ and the predictive importance and functional proximity between two pairs of genes: $CD11c$ and Zyx as well as $CD11c$ and $CD33$.

Our main goal was building a decision tree model that could choose the most predictive heterogeneous genes. In this example, the predictive set of genes $\{CD33, HOXA9\}$ is a set of predictive heterogeneous genes. If we apply directly Quinlan's algorithm C4.5/C5, the resulting predictor set can be anyone of these sets: $\{CD33, CD11c\}$, $\{CD33, Zyx\}$, $\{HOXA9, CD11c\}$ or even $\{CD11c,\ CD33\}$, $\{CD11c,\ Zyx\}$. In these cases we take predictive genes, but in many of these cases the chosen genes have the same functionality. This can be understood as redundant information on the predictive model. This results give us an optimistic glimpse of the potentialities of our gene-integrated decision trees algorithm GENETREE.

8.6 Discussion and Outlook

Gene expression profiling studies are increasingly being used to identify clusters of functionally linked co-expressed genes, as stated as a class discovery issue. However, the prediction of the state of a tissue, the toxicity of a molecule or the effectiveness of a medicament is a relative new issue concerning gene expression technology studies. Modern computational and statistical methods, such as supervised techniques, had been poorly used to reliably predict the outcome of a biological sample. In the last years, the class prediction issue has been explored mainly in the tumor classification application, but it remains a widely open research field.

In this chapter, we deal with the class prediction problem giving a general framework for applying supervised algorithms in gene expression technologies. At the moment, most of the existing approaches have used only the gene expression profiles information. We strongly suggest the utilization of the large and heterogeneous sources of biological knowledge across all the steps for building a predictive model.

In order to achieve the sample selection and discretization steps of the predictive model process, we propose two novel algorithms: an outlier sample detection method (see section 4.1.4) and the normal discretization (NORDI) algorithm (section 8.4) specially designed for gene expression data sets. For succeeding on the sample selection issue, we strongly suggests the use of all the available biological sample information contained in the gene expression technology protocol as MIAME in the case of microarray data. This qualitative sample information such as gender, age, degree of alcoholism, smoking degree etc., could be valuable information for sample selection in the pretreatment phase.

Concerning the choose of a predictive model, we suggest the utilization of an algorithm that can be handled and interpreted through biological insights. This avoids all complicated "black box" supervised methodologies that could complicate the already intrincate gene-protein interactions. Here, we have proposed an "intelligent" use of decision trees methods, taking care of their known drawbacks as overfitting or optimization failures in order to find an optimal set of functional heterogeneous genes to predict the class or classes of a given biological sample. This challenge is enormous and must be solved by ingeniously integrating the available sources of biological knowledge.

To date, this problem has received special attention in the context of disease-type applications as cancer research. A reliable and precise classification of tumors is essential for successful diagnosis and treatment of cancer. Current methods for classifying human malignancies rely on a variety of clinical, morphological and molecular variables. In spite of recent progress, there are still uncertainties in diagnosis because most cancer genes remain functionally uncharacterized in the physiological context of disease development. Gene expression technologies are novel biotechnologies which are being used increasingly to tackle this problem in cancer research [133, 7, 166, 242]. Concerning only this specific field of research, analysts dispose of a large choice of heterogeneous biological sources of information to tackle the tumor prediction issue (see TABLE 8.4).

In this chapter we explore a plausible solution to tackle this prediction challenge: GENE-integrated analysis for biological sample prediction using decision TREEs (GENETREE). The main contribution of our approach consists in proposing an original splitting criterion composed of five different criteria: the entropy criterion SC_E for manipulating gene expression profiles and four additional criteria, SC_1, SC_2, SC_3 and SC_4 obtained from gene annotations.

The construction of the annotational gene measures underlying this four criteria is not an easy task. The challenge of computing the proposed novel proximity measures, M_1 and M_2, is enormous. Here, we present a manual way of searching and extracting the principal annotations concerning a given gene and their goal-classes M_1, and between a given gene and the extended goal-classes M_2. However, in reality, this task should be done automatically by text mining tools capable of extracting high-quality information from text and databases

Biological Source of Information	Source Type
* Gene Ontology (GO)	semantic
*Munich Information Center for Protein Sequence (MIPS)	gene/protein-related
*Gene Map Annotator and Pathway Profiler (GENMAPP)	gene/protein-related
*Kyoto Encyclopedia of Genes and Genomes (KEGG)	gene/protein-related
*Biocarta	gene/protein-related
Cancer Cell Map	gene/protein-related
Module Map	gene/protein-related
*Uniprot	molecular databases
Biomolecular interaction Network Database (BIND)	protein interaction
IntAct	protein interaction
Human Protein Reference Database (HPRD)	protein interaction
Database of Interacting Proteins (DIP)	protein interaction
Online Predicted Human Interaction Database (OPHID)	protein interaction
Molecular Interaction Network Database (MINT)	protein interaction
Protein-protein interaction (PP1) of cancer proteins	protein interaction
Cancer Gene Census	cancer genes
Cancer Gene Data Curation Project	cancer genes
Cancer Gene Resequencing Resource	cancer genes
The Tumor Gene Family Databases	cancer genes
Oncomine	cancer genes
Cancer Progam Data sets	gene expression database
*Stanford Microarray Database (SMD)	gene expression database
*Gene Expression Omnibus (GEO)	gene expression database
Cancer Gene Expression Database(CGED)	gene expression database

TABLE 8.4: Cancer-related genomic data sources: semantic databases, experience databases, gene-protein related specific databases. The sources of information marked with * have an entry on the web-site glossary.

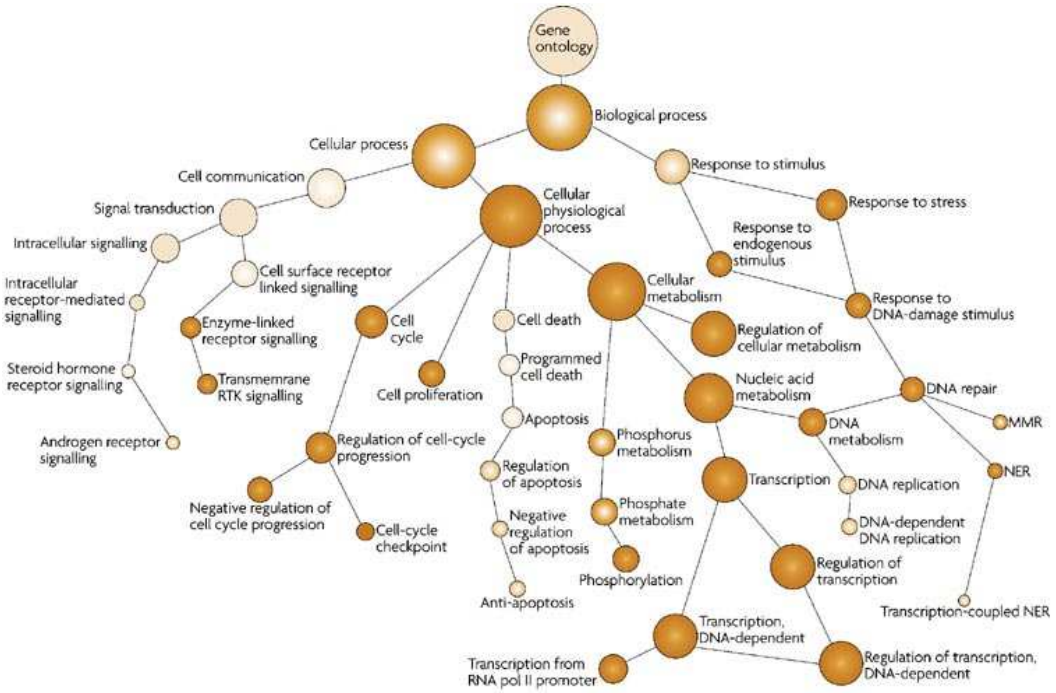

FIG. 8.4: A graphical representation of a nested GO classification showing the functional annotations of 384 known cancer genes. Image from [152] with permission of the author.

(discussed in section 2.4), in order to give a measure of a given gene and their goal classes or extended goal classes. Nowadays, this subject is becoming intensively studied and is a wide-open research field.

In order to calculate the functional proximity between two or more genes we have used a previously proposed GO tree distance between a set of annotations. The two measures M_3 and M_4 inspired from this distance, are useful rough measures of the heterogeneity and/or homogeneity of two or more genes given the set of their respective GO annotations. However, this measure has to be improved in order to reflect better the underlying multiple biological process hidden behind gene annotations.

Still, Gene Ontology is one of the valuable semantic sources of biological knowledge that intends to reflect a large image of the gene associated biological processes, cellular components and molecular functions of different species. The construction of a annotational measure that could reflect the heterogeneity of two genes given the set of their annotations can be done. A recent publication of Pingzao et al. [152] has showed the skewed functional annotations (GO terms) assigned to known cancer genes derived largely from familial syndromes or single-gene defect* cancers (see FIG. 8.4).

FIGURE 8.4 shows that GO provides a computationally accessible, organism-independent means for examining and reporting gene function and their annotational relationships. The size of the terms (circles) is proportional to total gene membership, and the color shading indicates the degree of stastical significance (darker tones denote drecreasing $p - values$).

Once the current version of GENETREE would be totally implemented, the next step is the study of the behavior of our decision model parameters: optimal size of the tree, number of trees to be build or our main splitting rule parameters δ, ϵ and ζ, across real gene expression data sets. A crucial issue would be the tuning and optimization of these parameters.

Concerning the whole prediction process parameters we'll take special attention to the informative gene selection and discretization steps, because they represent one of the largest source of error and bias in the model, and may lead to false conclusions [250].

At the moment, GENETREE constitutes a rising effort for building a gene-integrated model capable of building a set of functionally heterogeneous gene predictors for improving the prediction in many gene expression technology applications as tissue-disease diagnosis, molecule-toxicity or drug-response.

Conclusion and Outlook

Conclusion

We have developed three data mining models for knowledge discovery with genomic expression data: CGGA (Co-expressed Gene Groups Analysis), GENMINER (Gene-integrated analysis using association rules discovery) and GENETREE (GENE-integrated analysis for biological sample prediction using decision tree algorithms). These approaches are automatic tools for interpreting the data issued from any gene expression technology. They can be used by experts in the field for discovering the hidden information and knowledge contained in genomic expression data. The main idea behind these approaches is the interpretation of gene expression data via the automatic integration of biological knowledge from different sources of information with numerical gene expression profiles.

CGGA and GENMINER models deal with the **class discovery** issue in gene expression technologies stated by us as: "Highlighting the main co-expressed and co-annotated gene groups using at least one source of biological knowledge".

Recently, numerous interpretation approaches have made efforts to tackle the class discovery issue. Indeed, these heterogeneous approaches have chosen several tools and methodologies to deal with this issue. Therefore, we have proposed a new framework for interpreting gene expression data as the result of the integration of gene expression profiles and corresponding gene annotations. As a basis of our contribution, we have presented an original classification of interpretation approaches, consisting in three axes: *knowledge-based* axis, *expression-based* axis, and *co-clustering* axis.

Nowadays, most of these approaches are based on gene expression measures (*expression-based*) which are often noisy data, thus the results can be severely biased. CGGA and GENMINER are *knowledge-based* and *co-clustering* approaches respectively, which automatically integrate gene expression profiles and the biological annotations of the genes obtained by the genome-wide sources of biological knowledge such as molecular databases, semantic sources, gene expression databases, bibliographic databases, gene/protein related specific sources and Miame information..

CGGA contains an original function which synthesizes the information contained in the gene expression measures with the correspondent gene annotations, in order to highlight the main co-expressed and co-annotated gene groups. By applying CGGA to well-known microarray experiments, we identify the main functionally enriched and co-expressed gene groups, and we have shown that this approach enhances and optimizes the interpretation of microarray experiments.

GENMINER is a co-clustering and bi-clustering association rules discovery approach which automatically integrates at once gene annotations and gene expression profiles to discover intrinsic associations between both data sources based on frequent patterns. Our algorithm is an adaptation of traditional association rules mining techniques, that takes advantage of the CLOSE [229] algorithm to generate low support, high confidence

and non redundant rules in an efficient way. Automatically extracted associations reveal significant groups, meaning important biological relationships between gene attributes and patterns. Many of these relationships are supported by recently reported works.

The GENETREE model deals with the **class prediction** issue in gene expression technologies which was defined as: "Building a predictive model for disease-type classification using at once gene expression measures and the sources of biological knowledge".

The biological sample prediction applied to gene expression data is a relatively new field in bioinformatics. Nevertheless, a variety of supervised learning algorithms have been used to solve the prediction problem for disease-type applications, such as cancer. These algorithms take into account only gene expression profiles without integrating any source of biological information, thus ignoring valuable biological information.

GENETREE is a supervised algorithm for biological sample prediction that takes advantage of the well known C4.5/C5.0 decision tree algorithms, and it extends the entropy splitting criterion to a more complex one which takes into account several sources of gene annotations. Thus, it automatically integrates gene expression profiles with gene annotations obtained by genome-wide sources of information such as Gene Ontology and gene/protein related specific sources of information.

In order to accomplish all steps necessary to build a predictive model, we have developed two novel algorithms: an outlier sample detection method and the normal discretization, NORDI, algorithm; these two algorithms are specially fitted to gene expression data sets.

Our **sample selection** method is based on combining the statistical Principal Component Analysis (PCA) and the hierarchical clustering unsupervised algorithm for detecting sample outliers in gene expression data sets. The promising results obtained by applying this approach to SAGE and microarray data sets show their effectiveness for sample selection in gene expression data sets.

NORDI (normal discretization) is based on statistical detection of outliers and the continuous application of normality tests for transforming the initial sample distribution from an *almost normal* distribution to a *more normal* one. By applying this approach to several gene expression data, we have shown that NORDI, enhances the interpretation results obtained by many supervised learning and association rules algorithms.

Outlook

The work presented in this thesis delineates a clear research path for knowledge discovery within gene expression data. The main idea behind this path lies on the interpretation of gene expression data by incorporating the valuable and heterogeneous knowledge contained in the biological sources of information.

Either unsupervised (which tackle the class discovery problem) or supervised (which deal with the class prediction problem) approaches must integrate the biological sources of information at one stage of their model building process. The scientific community

studying the subject has no answer to determine the step of the process when the biological information has to be integrated.

In this section we first explain the future works concerning our three mining algorithms, and then we give a general outlook concerning the stated research path in bioinformatics.

CGGA is a practical automatic tool for constructing co-expressed and co-annotated gene groups. It can be used as a tool for platform-independent validation of a microarray experiment and its comparison with the huge number of existing experimental and documentation databases. Experimental results show the interest of our approach and make it possible to identify relevant information on the analyzed biological processes. However, it cannot identify groups of genes expressed only at certain phases of the biological process, so we plan to integrate the biological information concerning the metabolic pathways to solve this lack. CGGA is rank-based, meaning that the measure for manipulating gene expression profiles is based on the position of the respective gene in a sorted list. Thus, the genes position in the rank list are sensitive parameters that have to be optimized using several measures for testing the expression variability of the genes. Furthermore, we could integrate bi-clustering algorithms for constructing several scenarios of gene expression variability taking into account several analysis purposes as gene variability in a part of the process, gene variability at the next time point of the process, gene variability at each biological stage of the process, for instance. These extensions would turn CGGA in a more robust algorithm.

GENMINER is a practical automatic tool for constructing co-expressed and co-annotated gene groups at once. Even if our algorithm generates only the minimal rules against inclusion, the number of rules can still be very high for expert interpretation. There is a need for post-treatment of the generated rules in order to help the expert to obtain meaningful biological associations. Post-treatment of rules in bioinformatics is an open research issue, which depends on the collaboration between the expert and the developer, as well as the characteristics of the biological source of information. Among the possible rule post-treatment tools that could render GENMINER more robust we can mention for instance:

- Develop a program that could generate association rules from any of the sources of biological knowledge and could search this existing rule over all obtained rules using GENMINER.
- Develop an interactive semi-automatic program that allows the expert to find "*interesting*" rules taking into account *his* knowledge and the existing knowledge.

Concerning the threshold issue for selecting significant rules, GENMINER uses the support-confidence framework providing also the lift of the rule in order to avoid the selection of association among uncorrelated elements. Although support and improvement values provide information about the association between the antecedent and the consequent parts of the rule, they do not inform about their statistical significance. For this

purpose, we plan to integrate at least the one-tailed hypergeometric test used to find significant co-expressed and co-annotated groups in CGGA algorithm.

GENETREE is an ingenious attempt for sample prediction using a decision tree algorithm and integrating the information contained in gene expression profiles and correspondent gene annotations. The GENETREE development, evaluation, parameter tuning and interpretation of their predictive results have to be done with gene expression data sets in order to measure the effectiveness of the model.

Once the current version of GENETREE is totally implemented, the next step is the study of the direct parameter behaviors in our decision model as: optimal size of the tree, number of trees to be built or our main splitting rule parameters δ, ϵ and ζ across real gene expression data sets. A crucial issue would be the tuning and optimization of these parameters.

Another important issue concerning our splitting criteria is the construction of the three gene measures based on gene annotations. An optimal way to construct them would be to develop an automatic tool in order to extract associations between genes and their concerned information from several sources of biological knowledge such as articles, for instance. This can be done using text mining techniques.

Concerning the model prediction steps, we can mention several key stages that have to be more deeply developed in order to succeed in any predictive model applied in gene expression data sets. These key stages are: selecting informative genes, sample selection and discretization. These three stages are open fields of research, and they represent an important source of error in applying supervised algorithms in gene expression technologies. Furthermore, these stages have no unique solution, they generally depend on the characteristics of the gene expression data set and the analysis' main goal.

Concerning the three mining approaches, the improvement of our approaches comes hand in hand with biological information source development. Thus, all the ameliorations and creation of well-structured, easy-handling, clearly-explained and up-to-date sources of biological information would yield to better interpretation results. The advances in the *text-mining* field, capable to extract high-quality information from text, would be crucial for the future development of knowledge discovery gene expression data tools.

In this thesis, we have dealt with extracting the hidden information and knowledge contained in gene expression data such as Microarray or SAGE. At the moment, there are several technical limitations that turn this goal into an extremely difficult task.

Genomic science intends to understand the biological process under the optic of the genes; indeed the proteins are the ones which activate or inhibit all the biological process in an organism. Unfortunately, these two research fields - genomics and proteomics - are studied separately, and the connections between them are not evident. Thus, the main interpretation of gene expression data results has to be translated in terms of proteins to understand the underlying biological process, which is not an easy task. Furthermore, this interpretation concerns gene regulatory networks* and metabolic pathways* which are not completely understood.

The Genome project has already sequenced all the genes of several organisms, humans included. Nevertheless, the functions and characteristics of most of the genes (this is the case of humans) are unknown.

Gene expression data may contain inherent noise due to several issues as manufacture, technical pre-treatment, analytic pre-treatment etc. However, gene expression technologies are improving the technical issues day by day.

Heterogeneity and particular characteristics of each one of the sources of biological knowledge may yield a difficult integration in any supervised or unsupervised algorithm. However, the scientific community working on genomics and their related fields is improving the quality and handiness of these precious sources of biological knowledge.

An important technical restriction is that the recent computational power is not enough to compute the majority of existing algorithms with very large data sets containing thousands or even millions of genes.

In spite of all these technical and knowledge constraints, the importance of understanding the complex structure and the underlying functions of biological processes concerning the living organisms is worthy enough to continue mining for knowledge discovery in the genomic field.

Bibliography

1. Abdi, H. and P. Molin. *Encyclopedia of Measurement and Statistics: Lilliefors test of normality*. Thousand Oaks (CA): Sage, 1971.
2. Adams, M. D., et al. "The genome sequence of Drosophila melanogaster," *Science*, *287(5461)*:2185–2195 (2000).
3. Agency, Enviromental Protection. *Statistical Training Course for Ground-Water Monitoring Data Analysis*. Office of Solid Waste, 1992.
4. Agrawal, R., et al. "Mining Association Rules Between Sets of Items in Large Databases." *In Proceedings of the ACM SIGMOD international conference on Management of data*. 207–216. 1993.
5. Al-Shahrour, F., et al. "FatiGO: a web tool for finding significant associations of Gene Ontology terms with groups of genes," *Bioinformatics*, *20*(4):578–580 (2004).
6. Alberts, B., et al. *The Molecular Biology of the cell*. Garland Publishing, 1989.
7. Alizadeh, A. et al. "Distinct types of diffuse large B-cell lymphoma identifyed by gene expression profiling," *Nature*, *403* (200).
8. Alon, U., et al. "Broad patterns of gene expression revelead by clustering analysis of tumor and normal colon tissues probed by oligonucleotide array." *Proceedings of the National Academy of Sciences of the United States of America96*. 6745–6750. June 1999.
9. Altman, R. and S. Raychaudhuri. "Whole-genome expression analysis: challenges beyond clustering," *Current Opinion Structural Biology*, *11*:340–347 (2001).
10. Amaratunga, D. and J. Cabrera. *Exploration and analysis of DNA mecroarray and protein array data*. Wiley series in Probability and Statistics, Wiley, 2003.
11. Ankerst, M., et al. "OPTICS: ordering points to identify the clustering structure." *SIGMOD '99: Proceedings of the 1999 ACM SIGMOD international conference on Management of data28*. 49–60. New York, NY, USA: ACM Press, June 1999.
12. Ashburner, M., et al. "Gene Ontology: tool for the unification of biology," *Nature Genetics*, *25*:25–29 (2001).
13. Attwood, T. and C. Miller. "Which craft is best in bioinformatics?," *Computer Chemistry*, *25*:329–339 (2001).
14. Aze, Jerome. *Extraction de connaissances a partir de données numériques et textuelles*. PhD dissertation, Université Paris-Sud, 2003.
15. Azuaje, F. "A cluster validity framework for genome expression data. Bioinformatics," *Bioinformatics*, *18*:319–320 (2002).
16. Baldi, P. and S. Brunak. *Bioinformatics: The machine learning approach* (2nd Edition). MIT Press, 2001.
17. Baldi, P. and A. Long. "A bayesian framework for the analysis of microarray expression data: regularized t-test and statistical inferences of gene changes," *BMC Bioinformatics*, *17*:509–519 (2001).
18. Banfield, J. and A. Raftery. "Model-based Gaussian and non-Gaussian clustering," *Biometrics*, *49*:803–822 (1993).

19. Bar-Joseph, Z., et al. "K-ary clustering with optimal leaf ordering for gene expression data," *Bioinformatics*, *19*:1070–1078 (2003).
20. Barnett, V. and T. Lewis. *Outliers in Statistical Data* (3 Edition). Wiley and Sons, 1994.
21. Bayardo, R. J., et al. "Constraint-based rule mining in large, dense databases," *Data mining and knowledge discovery*, *4*:217–240 (2000).
22. Becquet, C., et al. "Strong-association-rule mining for large-scale gene-expression data analysis: a case study on human SAGE data.," *Genome Biology*, *3*:1–16 (2002).
23. Beissbarth, T. and T.P. Speed. "GOstat: find statistically overrepresented Gene Ontologies within a group of genes," *Bioinformatics*, *20*(9):1464–1465 (2004).
24. Bellman, R. *Adaptive control processes: A guided tour*. Princeton University Press, 1961.
25. Ben-Dor, A., et al. "Tissue classification with gene expression profiles," *Journal of Computational Biology*, *7*:559–584 (2000).
26. Ben-Dor, A., et al. "Clustering gene expression patterns," *Computational Biology*, *6*:281–297 (1999).
27. Ben-Hur, A., et al. "A stability based method for discovering structure in clustered data." *Pacific Symposium on Biocomputing 7*. 6–17. 2002.
28. Benjamini, Y. and Y. Hochberg. "Controlling the flase discovery rate: a practical and powerful approach to multiple testing," *Journal of the Royal Statistical Society*, *57*:289–300 (1995).
29. Benson, Dennis A., et al. "Genbank," *Nucleic Acids Research*, *30*:17–20 (2002).
30. Bera, A. and C. Jarque. "Efficient tests for normality, homoscedasticity and serial independence of regression residuals: Monte Carlo evidence," *Economics Letters*, *7*:313–318 (1981).
31. Bergeron, Bryan. *Bioinformatics Computing*. Prentice Hall, 2002.
32. Bhadra, D. and A. Garg. *An interactive visual framework for detecting clusters of a multidimensional dataset*. Technical Report 2001-03, Department of Computer Science and Engineering, University at Buffalo, NY, 2001.
33. Bhaskar, H., et al. "Machine learning in bioinformatics: A brief survey and recommendations for practitioners," *Computers in Biology and Medicine*, 1–21 (2005).
34. Blaschke, C., et al. "Co-clustering of biological networks and gene expression data," *Bioinformatics*, *18*:S145–S154 (2002).
35. Blaschke, C., et al. "Extracting information automatically from biological literature," *Comparative and Functional Genomics*, *2*(5):310–313 (2001).
36. Blatt, M., et al. "Superparamagnetic clustering of data," *Physical Review Letters*, *76*:3251–3254 (1996).
37. Blum, Avrim, et al. "Beating the hold-out: bounds for K-fold and progressive cross-validation." *COLT '99: Proceedings of the twelfth annual conference on Computational learning theory*. 203–208. New York, NY, USA: ACM Press, 1999.
38. Bo, T. and I. Jonassen. "New feature subset selection procedures for classification of expression profiles," *Genome Biology*, *3*:1–11 (2002).

39. Bodenreider, Olivier. "The Unified Medical Language System (UMLS): integrating biomedical terminology," *Nucleic Acids Research*, *32*:D267–D270 (2004).
40. Boguski, M. S., et al. "dbEST-database for "'expressed sequence tags"'," *Nature Genetics*, *4*:332–333 (1993).
41. Bolstad, B. M., et al. "A comparison of normalization methods for high density oligonucleotide array data based on variance and bias," *Bioinformatics*, *19*:185–193 (2003).
42. Bonferroni, C. E. *Il calcolo delle assicurazioni su gruppi di teste*. Rome, 1935. Chapter Studi in Onore del Professore Salvatore Ortu Carboni.
43. Bonferroni, C. E. *Teoria statistica delle classi e calcolo delle probabilità*, *8*. Pubblicazioni del Instituto Superiorie di Scienze Economiche e Commerciali di Firenze, 1936.
44. Boon, K., et al. "An anatomy of normal and malignant gene expression." *Proceedings of the National Academy of Sciences of the United States of America99*. 11287–11292. 2002.
45. Bowtell, D. D. "Options available from start to finish for obtaining expression data by microarray," *Nature Genetics*, *21*:25–32 (1999). Supplement.
46. Brazma, A., et al. "Minimum Information about a microarray experiment MIAME - toward standards for microarray data," *Nature Genetics*, *29*:365–371 (2001).
47. Brazma, A., et al. "Array-Express - A public repository for microarray gene expression data at the EBI," *Nucleic acids*, *31*:68–71 (January 2003).
48. Brazma, A. and J. Vilo. "Minireview: Gene expression data analysis," *Federation of European Biochemical societies*, *480*:17–24 (June 2000).
49. Brazma, Alvis, et al. "A quick introduction to elements of biology: cells, molecules, genes, functional genomics, microarrays." http://www.ebi.ac.uk/microarray/-biology_intro.html, October 2001.
50. Breiman, L. "Baggin predictors," *Machine Learning*, *24*:123–140 (1996).
51. Breiman, L. *Manual on setting up, using and understanding random forests v3.1*. University of California at Berkeley, 2002.
52. Breiman, L., et al. *Classification and regression trees*. Wadsworth, Monterrey, CA, 1994.
53. Breitling, R., et al. "IGA: A simple tool to enhance sensitivity and facilitate interpretation of microarray experiments," *BMC Bioinformatics*, *2005*(34) (2004).
54. Brin, S., et al. "Beyond Market Baskets: Generalizing Association Rules to Correlations." *Proceedings of the ACM SIGMOD Internaltional Conference on Management of Data*. 265–276. 1997.
55. Brin, S., et al. "Dynamic itemset counting and implication rules for market basket data." *Proceedings of the ACM SIGMOD Internaltional Conference on Management of Data*. 255–264. 1997.
56. Brisson, Laurent. *Intégration de connaissances expertes dans le processus de fouille de données pour l'extraction d'informations pertinentes*. PhD dissertation, Uiversité de Nice Sophia Antipolis, 2006.

57. Brown, M. P. et al. "Knowledge based analysis of microarray gene expression data by using support vector machines." *Proceedings of the National Academy of Sciences of the United States of America97*. 262–267. 2000.
58. Califano, A., et al. "Analysis of gene expression microarrays for phenotype classification." *ISM'00: Proceedings of the International Conference on Intelligent Systems for Molecular Biology8*. 75–85. 2000.
59. Callow, J., et al. "Microarray Expression Profiling Identifies genes with altered expression in HDL-Deficient mice," *Genome Research*, *10*:2022–2029 (2000).
60. Carletti, G. "comparaison empirique de méthodes statistiques de detection de vqleurs anormales a une et a plusieurs dimensions." *Proceedings de la faculté des Sciences Agronomiques de l'etat: Gembloux, Belgique*. 1–225. 1988.
61. Carmona-Saez, P., et al. "Integrated analyis of gene expression by association rules discovery," *BMC Bioinformatics*, *7*:54 (2006).
62. Casella, G. *Statistical Inference*. Duxbury series in Probability and Statistics, Duxbury, 2002.
63. Catlett, J. "On Changing continuous attributes into ordered discrete attributes." *Proceedings of the European Working session on learning*. 164–178. 1991.
64. Šášik, R., et al. "Percolation Clustering: A novel algorithm applied to the clustering of gene expression patterns in dictyostelium developement." *Pacific Symposium on Biotecomputing*. 335–347. 2001.
65. Šidák, Z. "Rectangular confidence regions for the means of multivariate normal distributions," *Journal of the American Statistical Association*, *62*:626–633 (1967).
66. Chen, J., et al. "Analysis of variance components in gene expression data," *BMC Bioinformatics*, *20*:1436–1446 (2001).
67. Chiang, J., et al. "GIS: a biomedical text-mining system for gene information discovery," *Bioinformatics*, *1*(20):120–121 (2004).
68. Chipman, H., et al. "Statistical Analysis of Gene Expression Microarray Data." *Statistical Analysis of Gene Expression Microarray Data*, edited by Terry Speed. 159–192. Chapman and Hall / CRC, 2003. Chapter 4.
69. Cho, R., et al. "A genome-wide transcriptional analysis of the mitotic cell cycle," *Molecular Cell*, *2*:65–73 (1998).
70. Choi, J., et al. "Combining multiple microarray studies and modeling inter-study variation.," *Bioinformatics*, i84–Ůi90 (2003).
71. Chu, S., et al. "The transcriptional program of sporulation in budding yeast," *Science*, *282*:699–705 (1998).
72. Chuaqui, R. "Post-analysis follow-up and validation of microarray experiments," *Nature Genetics*, *32*:509–514 (2002).
73. Chung, C. H., et al. "Molecular portraits and the family tree of cancer," *Nature Genetics*, *32*:533–540 (2002).
74. Churchill, G. "Fundamentals of experimental design for CDNA micro-arrays," *Nature Genetics*, *32*:490–495 (2002).
75. Cios, K., et al. *Data Mining Methods for Knowledge Discovery*. Boston/London: Kluwer Academic Publishers, 1998.

76. Cleveland, W. S. "Robust locally weighted regression and smoothing scatterplots," *Journal of the American Statistical Association*, *74*:829–836 (1979).
77. Cochran, W. and G Snedecor. *Statistical Methods*. Iowa State University Press, 1980.
78. Collins, F. S. "Microarrays and macroconsequences," *Nature Genetics*, *21(1)*:2 (1999).
79. Consortium, The UniProt. "The Universal Protein Resource (UniProt)," *Nucleic Acids Research*, *35*:D193–197 (2007).
80. Cover, T. and J. Thomas. *Elements of information theory*. New York, USA: Wiley-Interscience, 1991.
81. Creighton, C. and S. Hanansh. "Mining gene expression databases for association rules," *Bioinformatics*, *19*:79–86 (2003).
82. Crick, F. "Central dogma of molecular biology," *Nature*, *227*:561–563 (1970).
83. Cui, Xiangqin and Gary A. Churchill. "Statistical tests for differential expression in cDNA microarray experiments," *Genome Biology*, *4*:210 (Mars 2003).
84. D'agostino, B. *Tutorials in Biostatistics: Statistical Modelling of Complex Medical Data* (1 Edition), *2*. Tutorials in Biostatistics. Wiley InterScience, 2004.
85. Datta, Su. and So. Datta. "Comparisons and validation of clustering techniques for microarray gene expression data," *Bioinformatics*, *4*:459–466 (2003).
86. davierwala, A., et al. "The synthetic genetic interation spectrum of essential genes," *Nature Genetics*, *37(10)*:1147–52 (2005).
87. Debouck, C. and P. N. Goodfellow. "DNA microarrays in drug discovery and development," *Nature Genetics*, *21(1)*:48–50 (1999). Supplement.
88. Dempster, A. P., et al. "Maximal likelihood from Incomplete Data Via the EM Algorithm," *Journal of the Royal Statistical Society*, *39*:1–38 (1977).
89. Deng, L., et al. "A rank sum test method for informative gene discovery." *Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'04)*. 410Ű–419. Seattle, Washington, USA: ACM Press, 2004.
90. DeRisi, J., et al. "Exploring the metabolic and genetic control of gene expression on a genomic scale," *Science*, *278*:680–686 (1997).
91. Devore, J. and R. Peck. *Statistics: the exploration and analysis of data* (3rd Edition). Pacific Grove, California, USA: Duxbury, 1986.
92. D'Haeseleer, P., et al. "Mining the gene expression matrix: Inferring gene relationships from large scale gene expression data," *Information processing in cells and tissues*, 203–212 (1998).
93. Dinel, S., et al. "Reproducibility, bioinformatic analysis and power of the SAGE method to evaluate changes in transcriptome," *Nucleic Acids Research*, *33* (2005).
94. Dixon, W. "Analysis of extreme values," *Annals of mathematical Statistics*, *21*:488–506 (1950).
95. Dopazo, J. and J. M. Carazo. "Phylogenetic reconstruction using an unsupervised growing neural network that adopts the topology of a phylogenetic tree," *Journal of Molecular Evolution*, *44*:226–233 (1997).

96. Dougherty, J., et al. "Supervised and Unsupervised Discretization of Continuous Features." *Proceeding of the 12th International Conference on Machine Learning*. 194–202. 1995.
97. Drăghici, S. "Statistical intelligence: effective analysis of high-density microarray data," *Drug Discovery Today*, *11*:S55–63 (2002).
98. Drăghici, S. *Data analysis tools for DNA microarrays*. Chapman and Hall / CRS, 2003.
99. Drăghici, S. and P. Khatri. "Global functional profiling of gene expression," *Genomics*, *1*(81):1–7 (2003).
100. Dubes, R. and A. Jain. *Algorithms for clustering data*. Prentice Hall, 1988.
101. Dudoit, S., et al. "Comparison of discrimination methods for the classification of tumors using gene expression data," *Journal of the american Statistical Association*, 77–87 (2002).
102. Dudoit, S. and F. Jane. "Classification in microarray experiments." *Technical Report, University of California Berkeley*. 1–63. 2002.
103. Dudoit, S., et al. *Statistical methods for identifying differentially expressed genes in replicated cDNA microarray experiments*. Technical Report 578, University of California, Berkeley, 2000.
104. Duggan, D. J., et al. "Expression profiling using cDNA microarrays," *Nature Genetics*, *21*:10–14 (1999). Supplement.
105. Editorial, Genetics. "Coming to terms with microarrays," *Nature Genetics*, *32*:333–334 (2002). Supplement.
106. Effron, B., et al. "Microarrays and their use in a comparative experiment," *Stanford publications*, 1–100 (2000).
107. Eisen, M., et al. "Cluster analysis and display of genome wide expression patterns." *Proceedings of the National Academy of Sciences of the USA95*. 14863–8. 1998.
108. Ekins, R. P. and R. W. Chu. "Microarrays: their origins and applications," *Trends in biotechnology*, *17*:217–218 (1999).
109. Elihu, D., et al. "Mercury Exposure and Effects at a Thermometer Factory," *Scandinavian Journal of Work Environmental Health*, *8*(1):161–Ű166 (2004).
110. Ester, M., et al. "Algorithm for dicovering clusters in large spatial databases with noise." *Proceedings of 2nd International Conference on KDD*. 226–231. 1996.
111. et al., Tamara Kulikova. "EMBL Nucleotide Sequence Database in 2006," *Nucleic Acids Research*, 16–20 (2006). Database issue.
112. Fang, Z., et al. "Knowledge guided analysis of microarray data," *Biomedical Informatics*, *10*:1–11 (2005).
113. Feller, W. *An Introduction to Probability Theory and Its Applications* (3 Edition). Wiley and sons, 1971.
114. Feng, W., et al. "Development of gene ontology tool for biological interpretation of genomic and proteomic data." *AMIA Annual Symposium Proceedings*. 839. 2003.
115. Fisher, L. and G. Van Belle. *Biostatistics: a methodology for health sciences*. New York, USA: Wiley and Sons, 1993.

116. Fisher, R.A. "On the interpretation of X2 from contingency tables, and the calculation of P," *Journal of the Royal Statistical Society*, *85*(1):87–94 (1922).

117. Fleischmann, R. D., et al. "Whole-genome random sequencing and assembly of Haemophilus influenzae Rd," *Science*, *269(5223)*:496–512 (1995).

118. Fraley, C. and A. E. Raftery. "How many clusters? Which clustering method? Answers via model-based cluster analysis," *The Computer Journal*, *41*:578–588 (1998).

119. Furey, T., et al. "Support Vector Machine Classification and validation of cancer tissue samples using microarray expression data," *Bioinformatics*, *16*:909–914 (2000).

120. Gabig, M. and G. Wegrzyn. "An introduction to DNA chips : principles, technology, applications and analysis," *Acta biochimica Polonica*, *48(3)*:8 (2001).

121. Ganter, B. and R. Wille. *Formal concept analysis: Mathematical foundations*. Springer-Verlag, 1999.

122. Gasch, A. and M. Eisen. "Exploring the conditional corregulation of yeast gene expression through fuzzy k-means clustering," *Genome Biology*, *3*:1–22 (2002).

123. Gasch, Audrey P., et al. "Genomic expression responses to DNA-damaging agents and the regulatory role of the yeast ATR Homolog Mec1p," *Molecular Biology of the Cell*, *12(10)*:2987Ű–3003 (2001).

124. Georgi, E., et al. "Analyzing microarray data using quantitative association rules," *Bioinformatics*, *21*:123–129 (2005).

125. Getz, G., et al. "Superparamagnetic clustering of yeast gene expression profiles," *Physica A*, *279*:457–464 (2000).

126. Ghosh, D. amd Chinnaiyan, A. M. "Mixture modelling of gene expression data from microarray experiments," *Bioinformatics*, *18*:275–286 (2002).

127. Ghosh, D., et al. "Statistical issues and methods for meta analysis of microarray data: a case study in prostate cancer," *Functional and Integrative Genomics*, *3*:180–Ű188 (2003).

128. Gibbons, D. and F. Roth. "Judging the quality of gene expression-Based Clustering Methods Using Gene Annotation," *Genome Research*, *12*:1574–1581 (2002).

129. Gilbert, R. *Statistical Methods for Environmental Pollution Monitoring*. Van Nostrand Reinhold, 1987.

130. Giurcaneanu, C.D., et al. "Stability-based cluster analysis applied to microarray data." *Proceedings of the Seventh International Symposium on Signal Processing and its Applications*. 57–60. 2003.

131. Gofieau, A., et al. "Life with 6000 genes," *Science*, *264(5287)*:563–567 (October 1996).

132. Gollub, J., et al. "The Stanford microarray database: data access and quality assessment tools," *Nucleic Acids Research*, *31*:94–96 (January 2003).

133. Golub, T. R., et al. "Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring," *Science*, *286*:531–537 (October 1999).

134. Grubbs, F. "Sample Criteria for Testing Outlying Observations," *The Annals of Mathematical Statistics*, *21*:27–58 (1950).

135. Grubbs, F. "Procedures for detecting Outlying Observations in Samples," *Technometrics*, *11*:1–21 (1969).

136. Grumbling, G., et al. "FlyBase: anatomical data, images and queries," *Nucleic Acids Research*, *34*:D484–D488 (2006).

137. Hakak, Y., et al. "Genome-wide Expression Analyis reveals dysregulation of Myelination-related genes in chronic schizophrenia." *Proceeding Natural Academie of Science98*. 4746–4751. 2001.

138. Halkidi, M., et al. "On clustering validation techniques," *Intelligent Information Systems*, *17*:107–145 (2001).

139. Hanisch, D., et al. "Co-clustering of biological networks and gene expression data," *Bioinformatics*, *18*:S145–S154 (2002).

140. Hartuv, E. and R. Shamir. "A clustering algorithm based on graph connectivity," *Information Processing Letters*, *76*:175–181 (2000).

141. Hastie, T., et al. "'Gene shaving' as a method for identifying distinct sets of genes with similar expression patterns," *Genome Biology*, *1*:1–21 (2000).

142. Hedenfalk, I. et al. "Gene-expression profiles in hereditary breast cancer," *The New England journal of medicine*, *344*:539–548 (2001).

143. Heller, Renu A., et al. "Discovery and analysis of inflammatory disease-related genes using cDNA microarrays." *Proceedings of the National Academy of Sciences of the United States of America94*. 2150–2155. 1997.

144. Herrero, J., et al. "A hierarchical unsupervised growing neural network for clustering gene expression patterns," *Bioinformatics*, *17*:126–136 (2001).

145. Heyer, L. J., et al. "Exploring expression data: identification and analysis of coexpressed genes," *Genome Research*, *9*:1106–1115 (1999).

146. Hinneburg, A. and D. A. Keim. "An efficient approach to clustering in large multimedia database with noise." *Proceedings of the 4th International Conference on Knowledge Discovery and data mining*. 58–65. 1998.

147. Hirschman, J. E. et al. "Genome Snapshot: a new resource at the Saccharomyces Genome Database (SGD) presenting an overview of the Saccharomyces cerevisiae genome," *Nucleic Acids Research*, *34*:D442–5 (2006). Database issue.

148. H.M. Berman, K. Henrick, H. Nakamura. "Announcing the worldwide Protein Data Bank," *Nature Structural Biology*, *10*:980 (2003).

149. Hollander, M. and D. A. Wolfe. *Nonparametric Statistical Method* (2nd Edition). New York, NY: Wiley, 1999.

150. Holm, S. "A simple sequentially rejective multiple test procedure," *Scandinavian Journal of Statistics*, *6*:65–70 (1979).

151. Hosack, D. and G. Dennis. "Identifying biological themes within lists of genes with EASE," *Genome Biology*, *4*(70) (2003).

152. Hu, P., et al. "Computational prediction of cancer-gene function," *Nature Reviews Cancer*, *7*:23–34 (2007).

153. Hubbard, T. J. P., et al. "Ensembl 2007," *Nucleic Acids Research* (January 2007). Database issue.

154. Hunter, A., et al. "An ontology of human developmental anatomy," *Journal of Anatomy*, *203*:347–355 (2003).

155. Hunter, L. *Molecular Biology for Computer Scientists. In Artificial Intelligence and Molecular Biology*. AAAI Press, 1993.

156. Hyafil, L. and R. Rivest. "Constructing optimal binary decision trees is NP-complete," *Information Processing Letters*, *5*:15–17 (1976).

157. Ji, L. and K. Tan. "Mining gene expression data for positive and negative co-regulated gene clusters," *Bioinformatics*, *20*:2711–2718 (2004).

158. Jiang, H., et al. "Joint Analysis of two microarray gene-expression data sets to select lung adenocarcinoma marker genes," *BMC Bioinformatics*, *5*:81 (2004).

159. Jin, W., et al. "The contributions of sex, genotype and age to transcriptional variance in Drosophila melanogaster," *Nature Genetics*, *29*:389–395 (2001).

160. Jolliffe, I. T. *Principal component analysis*. Springer Verlag, 1986.

161. Kanehisa, M., et al. "From genomics to chemical genomics: new developments in KEGG," *Nucleic Acids Research*, *34*:D354–357 (2006).

162. Karp, G. *Cell and molecular biology: Concepts and Experiments*. John Wiley and Sons Inc, 2002.

163. Kasturi, J., et al. "An information theoretic approach for analyzing temporal patterns of gene expression," *Bioinformatics*, *19*:449–458 (2003).

164. Kaufman, L. and P. Rousseeuw. *Findings Groups in Data. An introduction to Cluster Analysis*. New York, USA: Wiley and Sons, 1990.

165. Kerr, K.M. and G. Churchill. "Statistical design and the analyissi of gene expression microarray data," *Genetics Research*, *77*:123–128 (2001).

166. Khan, J. et al. "Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks," *Nature Medicine*, *7*:673–679 (2001).

167. Kim, S. and D. Volsky. "PAGE: Parametric Analysis of Gene Set Enrichment," *BMC Bioinformatics*, *6*:144 (2005).

168. Kohavi, Ron. "A study of cross-validation and bootstrap for accuracy estimation and model selection." *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence2*. 1137Ű–1143. 1995.

169. Kohonen, T. *Self-organization and associative memory*. Berlin: Springer-Verlag, 1994.

170. Kotala, P., et al. "Gene expression profiling of DNA microarray data using peano count tree (p-trees)." *In Proceedings of the first Virtual Conference on Genomics and bioinformatics*. 15–16. 2001.

171. Lander, E. S. "The new genomics: global views of biology," *Science*, *274*:536–539 (October 1996).

172. Lander, E. S. "Array of hope," *Nature Genetics*, *21*:3–4 (January 1999).

173. Lash, A. E., et al. "SAGEmap: a public gene expression resource," *Genome Research*, *10*:1051–60 (2000).

174. Lee, M. L., et al. "Importance of replication in microarray gene expression studies: statistical methods and evidence from repetitive cDNA hybridizations." *Proceedings of the National Academy of Sciences of the United States of America97*. 9834–9839. 2000.

175. Lee, S., et al. "A graph theoretic modeling on GO space for biological interpretation of gene clusters," *Bioinformatics*, *3*:381–386 (2004).

176. Lee, T., et al. "Transcriptional Regulatory Networks in Saccharomyces cerevisiae," *Science*, *298*(5594):799–804 (2002).

177. Levine, E. and E. Domany. "Resamplimg methods for unsupervised estimation of cluster validity," *Neural Computation*, *13*:2573–2693 (2001).

178. Lewin, B. *Genes VIII*. Prentice Hall, 2003.

179. Li, L., et al. "Gene assesment and sample classification for gene expressino dara using a genetic algorithm / k-nearest neighbor method," *Combinatorial chemistry and high throughput screening*, *4*:727–739 (2001).

180. Lilliefors, H. "On the Kolmogorov-Smirnov test for normality with mean and variance unknown," *Journal of the American Statistical Association*, *62* (1967).

181. Lim, T., et al. "A Comparison of Prediction Accuracy, Complexity, and Training Time of Thirty-Three Old and New Classification Algorithms," *Machine Learning*, *40(3)*:203–228 (2000).

182. Lipshutz, R. J., et al. "High density synthetic oligonucleotide arrays," *Nature Genetics*, *21*:21–24 (1999).

183. Litteli, R., et al. "RD: SAS system for mixed models." *In Proceedings of the Cary, NC: SAS Institute Inc.*. 1–76. 1996.

184. Little, R. and D. Rubin. *Statistical Analysis with Missing Data* (2 Edition). Wiley and Sons, 2002.

185. Liu, J., et al. "Biclustering in gene expression data by tendency." *Computational Systems Bioinformatics Conference, CSB 2004 Proceedings*. 182–193. 2004.

186. Liu, J., et al. "Gene ontology friendly biclustering of expression profiles." *Computational Systems Bioinformatics Conference, CSB 2004 Proceedings*. 436–447. 2004.

187. Lockhart, D. J. et al. "Expression monitoring by hybridization to high-density oglionucleotide arrays," *Nature Biotechnology*, *14*:1675–1680 (1996).

188. Loguinov, A. V., et al. "Exploratory differential gene expression analysis in microarray experiments with no or limited replication," *Genome Biology*, *5* (2004).

189. Loh, W.-Y. and Y.-S. Shih. "Split selection methods for classification trees," *Statistica Sinica*, *7*:815–840 (1997).

190. MacQueen, J. B. "Some methods for classificaion and analysis of multivariate observations." *Proceedings of the Fifth Berkeley Symposium on Mathmatical Statistics and Probability1*. 281–297. Berkeley: Unversity of California Press, 1967.

191. Maere, S., et al. "BiNGO: a Cytoscape plugin to assess overrepresentation of Gene Ontology categories in biological networks," *Bioinformatics*, *21*:3448–3449 (2005).

192. Mager, W. H. and A. J. De Kruijff. "Stress-induced transcriptional activation," *Microbiol Rev*, *59*:506–531 (1995).

193. Man, M., et al. "POWER SAGE: comparing statistical test for SAGE experiments," *Bioinformatics*, *16*(11):953–959 (2000).

194. Mannila, H., et al. "Efficient algorithms for discovering association rules." *In Proceedings of the AAAI Workshop on Knowledge Discovery in Databases*. 181–192. 1994.

195. Marion, R. M., et al. "Sfp I is a stress-and nutrient- sensitive regulator of ribosomal protein gene expression." *Proceedings of the National Academy of Sciences of the USA101*. 14315–14322. 2006.

196. Martin, D., et al. "GOToolBox: functional analysis of gene datasets based on Gene Ontology," *Genome Biology*, *5*(12) (2004).

197. Martinez, R. *Azerty data analyis: A full microarray analysis over the cicatrization process*. Technical Report, I3s laboratory, Execo Project UNSA, 2006.

198. Martinez, R. *Normal Discretization Algorithm for Gene Expression Technologies: NORDI*. Technical Report, I3s laboratory, Execo Project UNSA, 2006.

199. Martinez, R., et al. "Exploratory Analysis of Cancer SAGE Data." *Proceedings of the PKDD'2005 conference, Discovery Challenge*. 1–12. 2005.

200. Martinez, R. and M. Collard. "Extracted Knowledge: Interpretation in Mining Biological Data: a Survey," *International Journal of Computer Science and Applications: Special issue in Research Challenges in Information Science*, *1*:1–21 (2007).

201. Martinez, R. and M. Collard. "Extracted Knowledge Interpretation in Mining Biological Data: a Survey." *Research Challenges in Information Science RCIS 07 proceedings1*. 1–18. 2007.

202. Martinez, R., et al. "GENMINER: Gene-Integrated analyis by association rules discovery," *Journal of Biochemistry* (2007). Acceptation in process.

203. Martinez, R., et al. "Co-expressed Gene Groups Analysis (CGGA): An Automatic Tool for the Interpretation of Microarray Experiments," *Journal of Integrative Bioinformatics*, *3*(11):1–12 (2006).

204. Masseroli, M., et al. "GFINDer: Genome Function INtegrated Discoverer through dynamic annotation, statistical analysis, and mining," *Nucleic Acids Research*, *32*:293–300 (2004).

205. Masys, D. "Use of keyword hierarchies to interpret gene expressions patterns," *Bioinformatics*, *17*:319–326 (2001).

206. Mavroudi, S., et al. "Gene expression data analysis with a dynamically extended self-organized map that exploits class information," *Bioinformatics*, *18*:1446–1453 (2002).

207. McGarry, Ken. "A survey of interstingness measures for knowledge discover," *Knowledge Engineering*, *20*:39–61 (2005).

208. McLachlan, G. J. *Analyzing Microarray Gene Expression Data*. Wiley, 2004.

209. McLachlan, G. J., et al. "A mixture model-based approach to the clustering of microarray expression data," *Bioinformatics*, *18*:413–422 (Mars 2002).

210. Mekalanos, J. J. "Environmental signals controlling expression of virulence determinants in bacteria," *Journal of Bacteriology*, *174*:1–7 (January 1992).

211. Miller, Robert T., et al. "A Comprehensive Approach to Clustering of Expressed Human Gene Sequence: The Sequence Tag Alignment and Consensus Knowledge Base," *Genome Research*, *9*:1143–1155 (1999).

212. Mitchell, T. M. *Machine learning*. McGraw Hill, 1997.

213. Mitelman, F., et al. "Mitelman Database of Chromosome Aberrations in Cancer." http://cgap.nci.nih.gov/Chromosomes/Mitelman, 2007.

214. Model, F., et al. "Feature selection for DNA methylation based cancer classification," *Bioinformatics*, *17*:S157–164 (2001).

215. Mootha, V., et al. "PGC-l'alpha-reponsive genes involved in oxidative phosphorylation are coordinately downregulated in human diabetes," *Nature Genetics*, *34*(3):267–273 (2003).

216. Morse, R. H. "RAP, RAP, open up! New wrinkles for RAPI in yeast," *Trends Genet*, *16*:51–53 (2000).

217. Muller, H., et al. "Textpresso: An Ontology-Based information Retrieval and Extraction system for biological literature," *PLoS Biology*, *2*(11):309 (2004).

218. MUnoz-Garcia, J. and J. Pascual-Acosta A. Moreno-Rebollo. "Outliers: a formal approach," *International Statistical Revue*, *58*:215–226 (1990).

219. Mutch, D., et al. "The limit fold change model: A practical approach for selecting differentially expressed genes from microarray data," *BMC Bioinformatics*, *3*:17 (2002).

220. Newton, M., et al. "Detecting differential gene expression with a semi parametric hierarchical mixture method," *Biostatistics*, *5*:155–176 (2004).

221. Ng, R. T., et al. "Hierarchical cluster analysis of SAGE data for cancer profiling." *Proceedings of the BIOKDD conference*. 65–72. 2001.

222. Ngyen, L. T., et al. "Flow cytometric analysis of in vitro proinflammatory cytokine secretion in peripheral blood from multiple sclerosis patients," *Journal of Clinical Immunology*, *19*:179–185 (1999).

223. NIST/SEMATECH. *e-Handbook of Statistical Methods*. http://www.itl.nist.gov/-div898/handbook/, 2007.

224. Pan, F., et al. "Carpenter: finding closed patterns in long biological datasets." *Ninth ACM SIGKDD international conference on knowledge discovery and data mining (KDD'03)9*. 637–642. 2003.

225. Pan, K., et al. "Effects of threshold choice on biological conclusions reched during analysis of gene expression by DNA microarrays," *National Academy of Sciences PNAS*, *102*:8961–8965 (2005).

226. Park, J., et al. "An efficient has based algorithm for mining association rules." *Proceedings of the ACM SIGMOD Internaltional Conference on Management of Data*. 175–186. 1997.

227. Parmigiani, G., et al. "A cross-study comparison of gene expression studies for the molecular classification of lung cancer," *Clinical Research*, *10*:2922–Ű2927 (2004).

228. Pasquier, C., et al. "THEA : Ontology-driven analysis of microarray data," *Bioinformatics*, *20*(16) (2004).

229. Pasquier, N., et al. "Efficient Mining of Association Rules using Closed Itemset Lattices," *Information Systems*, *24*:25–46 (1999).

230. Pasquier, N., et al. "Pruning closed itemsets lattices for association rules." *BDA conference*. 177–196. 1998.

231. Pavlidis, P., et al. "Gene functinoal classification from heterogeneous data." *Proceedings of the fifth annual international conference in computational biology (RECOMB 2001)*. 249–255. ACM Press, 2001.

232. Pearson, E. and C. ChandraSekar. "The Efficiency of Statistical Tools and A Criterion for the Rejection of Outlying Observations," *Biometrika*, *28*:308–320 (1936).

233. Pearson, E. and H. David. "The Distribution of the Ratio, in a Single Normal Sample, of Range to Standard Deviation," *Biometrika*, *41*:482–493 (1954).

234. Pehkonen, P., et al. "Theme discovery from gene lists for identification and viewing of multiple functional groups," *BMC Bioinformatics*, *6*:162 (2005).

235. Pei, J., et al. "CLOSET: An efficient algorithm for mining frequent closed itemsets." *Proceedings 2000 ACM-SIGMOD International Workshop in Data Mining and Knowledge Discovery (DMKD'00)*. 11–20. 2000.

236. Perez-Iratxeta, C., et al. "Exploring Medline abstracts with XplorMed," *Drugs Today*, *38*(6):381–389 (2002).

237. Perou, C. M., et al. "Distinctive gene expression patterns in human mammary epithelial cells and breast caners." *Proceedings of the National Academy of Sciences of the United States of America96*. 9212–9317. August 1999.

238. Pevsner, J. *Bioinformatics and Functional Genomics*. Wiley-Liss, October 2003.

239. Pfaltz, J. and C. Taylor. "Closed Set Mining of Biological Data." *In Proceedings of BIOKDD02: Workshop on Data Mining in Bioinformatics*. 43–48. 2002.

240. Pingzhao, H., et al. "Integrative analysis of multiple gene expression profiles with quality adjusted effect size models," *BMC Bioinformatics*, *6*:128 (2005).

241. Planchon, V. "Traitement des valeurs aberrantes : concepts actuels et tendances generales," *BASE*, *9*:19–34 (2005).

242. Pomeroy, S., et al. "Prediction central nervous system embryonal tumour outcome based on gene expression," *Nature*, *415*:436–442 (2002).

243. Pontius, J. U., et al. "UniGene: a unified view of the transcriptome," *The NCBI Handbook : National Center for Biotechnology Information* (2003).

244. Pruitt, K. D., et al. "NCBI Reference Sequence (RefSeq): a curated non-redundant sequence database of genomes, transcripts and proteins," *Nucleic Acids Research*, *33*:D501–D504 (January 2005).

245. Purves, William K., et al. *Life: The Science of Biology* (7 Edition). Sinauer Associates Inc and W. H. Freeman and Company, 2003.

246. Quackenbush, J. "Computational analysis of microarray data," *Nature Genetics*, *2*:418–427 (2001).

247. Quinlan, J. Ross. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, 1993.

248. Ralph-Herwig, P. A., et al. "Large-scale clustering of cDNA-fingerprinting data," *Genome Research*, *9*:1093–1105 (1999).

249. Ramanathan, Murali, et al. "Visualized Classification of Multiple Sample Types." *Proceedings of the Workshop of Data Mining in Bioinformatics in BIOKDD02*. 14863–8. 2002.

250. Ramaswamy, Sridhar, et al. "Multi-class Cancer Diagnosis Using Tumor Gene Expression Signatures." *Proceeding Natural Academie of Science98*. 4746–4751. 2001.

251. Ramoni, M. F., et al. "Cluster analysis of gene expression dynamics." *Proceedings of the National Academy of Sciences of the United States of America99*. 9121–9126. July 2002.

252. Raudys, S. and A. K. Jain. "Small sample size effects in statistical pattern recognition: recommendations for practitioners," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *13*:252–264 (1991).

253. Rebhan, M., et al. "GeneCards: encyclopedia for genes, proteins and diseases," *Weizmann Institute of Science, Bioinformatics Unit and Genome Center (Rehovot, Israel)* (1997). http://www.genecards.org/.

254. Rhodes, D., et al. "Meta-analysis of microarrays : inter-study validation of gene expression profiles reveals pathway dysregulation in prostate cancer," *Cancer Research*, *62*:4427–Ű4433 (2002).

255. Rindflesch, T., et al. "EDGAR: extraction of drugs, genes and relations from the biomedical literature." *Proceedings of the Pacific Symposium on Biocomputing*. 517–528. 2000.

256. Ripley, D. *Pattern Recognition and Neural Networks*. Cambridge University Press, 1996.

257. Riva, A., et al. "Comments on selected fundamental aspects of microarray analysis," *Computational Biology and Chemistry*, *29*:319–336 (2005).

258. Robinson, M. "FunSpec : a Web based cluster interpreter for yeast," *BMC Bioinformatics*, *3*:35 (2002).

259. Ross, D. T., et al. "Systematic variation in gene expression patterns in human cancer cell lines," *Nature Genetics*, *24*:227–35 (2000).

260. Rousseeuw, P. "Silhouettes: a graphical aid to the interpretation and validation of cluster analysis," *Computational and Applied Mathematics*, *20*:53–Ű65 (1987).

261. Royston, J. "An extension of Shapiro and Wilk's W Test for normality to large samples," *Application Statistics*, *31*:115–124 (1982).

262. Ruggero, G., et al. "Assesement of discretization techniques for relevant pattern discovery from gene expression data." *Workshop on data mining in bioinformatics with SIGKDD Conference*. 1–7. 2004.

263. Ruiz, C., et al. "Distance-enhanced association rules for gene expression." *In Proceedings of BIOKDD03: 3rd ACM SIGKDD Workshop on Data Mining in Bioinformatics*. 34–40. 2003.

264. Ryan, T. *Modern Regression Methods* (1 Edition). Wiley and Sons, 1997.

265. Saha, Saurabh., et al. "Using the transcriptome to annotate the genome," *Nature Biotechnology*, *20*:508–512 (2002).

266. Sauer, S. and S Sherwood. "SAGE for beginners." http://www.embl-heidelberg.de/-info/sage/, January 2002.

267. Savasere, E., et al. "An efficient algorithm for mining association rules in large databases." *Proceedings of the 21th Internaltional Conference on Very Large Data Bases*. 432–444. 1995.

268. Schapire, R. E. "The strength of weak learnability," *Machine learning*, *5*:197–227 (1990).

269. Schapire, R. E., et al. "Boosting the margin: a new explaination for the effectiveness of voting methods," *Annals of Statistics*, *26*:1651–1686 (1998).

270. Schena, M., et al. "Quantitative monitoring of gene expression patterns with a complementary DNA microarray," *Science*, *270*:467–470 (1995).

271. Schena, M., et al. "Parallel human genome analysis: Microarray-based expression monitoring of 1000 genes." *Proceedings of the National Academy of Sciences of the United States92*. 10614–10619. 1996.

272. Schliep, A. Schönhuth, A. and C. Steinhoff. "Using hidden markov models to analyze gene expression time course data," *Bioinformatics*, *19*:i255–i263 (2003). Supplement.

273. Schuler, G. S., et al. "A gene map of the human genome," *Science*, *274*:540–546 (1996).

274. Setubal, J. C. and J. Meidanis. *Introduction to Conputational Molecular Biology*. PWS Publishing, 1997.

275. Shah, N. and N. Fedoroff. "CLENCH: a program for calculating Cluster ENriCHment using the gene ontology," *Bioinformatics*, *20*:1196–1197 (2004).

276. Shamir, R. and R. Sharan. "CLICK: A clustering algorithm with applications to gene expression Analysis." *Proceedings International Conference Intelligent Systems and Mollecular Biology8*. 307–316. 2000.

277. Shapiro, S. and M. Wilk. "An analysis of variance test for normality (complete samples)," *Biometrika*, *52*:591–611 (1965).

278. Shapiro, S., et al. "A comparative study or various test for normality," *Journal American Statistical Association*, *63*:1343–1372 (1968).

279. Shatkay, H., et al. "Genes, themes, microarrays: using information retrieval for large-scale gene analysis." *Proceedings International Conference Intelligent Systems Molecular Biology11*. 340–7. 2000.

280. Shippy, R., et al. "Performance evaluation of commercial short-oligonucleotide microarrays and the impact of noise in making cross-platform correlations," *BMC Genomics*, *5*:61 (2004).

281. Silverstein, C., et al. "Beyond market baskets: Generalizing association rules to dependence rules," *Data mining and knowledge discovery*, *2*:39–68 (1998).

282. Simoff, S. and M. Maher. "Ontology-based multimedia data mining for design information retrieval." *Computing in Civil Engineering, Proceedings of the International Computing Congress100*. 212–223. 2003.

283. Slonim, D. "From patterns to pathways: gene expression data analysis comes of age," *Nature Genetics*, 502–508 (2002).

284. Smet, F. D., et al. "Afaptive quality-based clustering of gene expression profiles," *Bioinformatics*, *18*:735–746 (2002).

285. Smolkin, M. and D. Ghosh. "Cluster stability scores for microarray data in cancer studies," *BMC Bioinformatics*, *4*(36) (2003).

286. Soinov, L., et al. "Towards reconstruction of gene networks from expression data by supervised learning," *Genome Biology*, *4*:6 (2002).

287. Sokal, R. and F. Rohlf. *Biometry: The Principles and Practice of Statistics in biological research* (3 Edition). Freeman, 1995.

288. Sokal, R. R. "Clustering and classification: Background and current directions." *Classification and clustering*, edited by J. Van Ryzin. Academic Press, 1977.

289. Southern, Edwin, et al. "Molecular interactions on microarrays," *Nature Genetics*, *21*:5–9 (1999).

290. Spellman, Paul T., et al. "Comprehensive identification of cell cycle-regulated genes of the yeast Saccharomyces cerevisiae by microarray hybridization," *Molecular Biology of the Cell*, *9*:3273–97 (1998).

291. Sprevak, D., et al. "A non-random data sampling method for classification model assessment." *ICPR '04: Proceedings of the 17th International Conference on Pattern Recognition3*. 406–409. 2004.

292. Standafer, E. and W. Wahlgren. *Modern Biology*. Holt, Rinehart and Winston, 2002.

293. Stark, C., et al. "BioGRID: A General Repository for Interaction Datasets," *Nucleic Acids Research*, *34*:D535–9 (2006).

294. Stekel, D. *Microarray Bioinformatics*. Cambridge University Press, 2003.

295. Stoyanova, R., et al. "Normalization of single channel DNA array data by principal component analysis," *Bioinformatics*, *20*:1772–1784 (2004).

296. STURN, Alexander, et al. "Genesis: cluster analysis of microarray data," *Bioinformatics*, *18*:207–208 (2002).

297. Sun, M., et al. "SAGE is far more sensitive than EST for detecting low-abundance transcripts," *BMC Genomics*, *5*:1 (2004).

298. Tamayo, P. and D. Slonim. "Interpreting patterns of gene expression with self-organizing maps: Methods and application to hematopoietic differentiation." *Proceedings of the National Academy of Sciences of the USA96*. 2907–2912. 1999.

299. Tanabe, L., et al. "MedMiner: an Internet text-mining tool for biomedical information, with application to gene expression profiling," *Biotechniques*, *27*(6):1210–1217 (1999).

300. Tang, C., et al. "ESPD: A pattern detection model underlying gene expression profiles," *Bioinformatics*, *20*:829–838 (2004).

301. Tao, H., et al. "Functional genomics: expression analysis od Escherichia coli growing on minimal and rich media," *Journal of Bacteriology*, *181*:6425–6440 (1999).

302. Tatusov, R. L., et al. "The COG database: a tool for genome-scale analysis of protein functions and evolution," *Nucleic Acids Research*, *28*:33–36 (2000). http://-www.ncbi.nlm.nih.gov/COG/.

303. Tavazoie, S., et al. "Systematic determination of genetic network architecture," *Nature Genetics*, *22*:281–285 (1999).

304. Thomas, J., et al. "An efficient and robust statistical modelling approach to discover differentially expressed genes using genomic expression profiles," *Genome Research*, *11*:1227–1236 (2001).

305. Tibshirani, R., et al. "Diagnosis of multiple cancer types by shrunken centroids of gene expression." *Proceedings of the National Academy of Sciences of the United States of America99*. 6567–6572. 2002.

306. Tietjen, G. and R. Moore. "Some Grubbs-type statistics for the detection of several outliers," *Technometrics*, *14*:583–597 (1972).

307. Tomida, S., et al. "Analysis of expression profile using fuzzy adaptive resonance theory," *Bioinformatics*, *18*:1073–1083 (2002).

308. Troyanskaya, O., et al. "Missing value estimation methods for DNA microarrays," *Bioinformatics*, *17*:520–525 (2001).

309. Tusher, V. G., et al. "Significance analysis of microarrays applied to the ionizing radiation response." *Proceedings of the National Academy of Sciences of the United States of America98*. 5116–5121. 2001.

310. Tuzhilin, A. and G. Adomavicius. "Handling very large numbers of association rules in the analysis of microarray data." *In Proceedings of the Eight ACM SIGMOD Internaltional Conference on Data Mining and Knowledge Discovery*. 396–404. 2002.

311. Velculescu, V., et al. "Serial analysis of gene expression," *Science*, *270*:484–487 (1995).

312. Viatcheslav, R. A. and J. W. Clarence. "Correction of sequence-based artifacts in serial analysis of gene expression," *Bioinformatics*, *20*:1254–1263 (2004).

313. Wang, David G., et al. "Large-scale identification, mapping and genotyping of single-nucleotide polymorphisms in the human genome," *Science*, *280*:1077–1082 (1998).

314. Weaver, R. F. *Molecular Biology*. McGraw-Hill, 2001.

315. Weinstock-Guttman, B., et al. "Genomic effects of interferonBeta in multiple sclerosis patients," *Jounal of Inmunology*, *171*:2694–2702 (2003).

316. Weng, S., et al. "Saccharomyces Genome Database (SGD) provides biochemical and structural information for budding yeast proteins," *Nucleic Acids Research*, *31*:216–218 (2003).

317. Westfall, P. H. and S. S. Young. *Resampling-based multiple testing: examples and methods for p-value adjustment*. New York, NY: Wiley, 1993.

318. Wilcoxon, F. "Individual comparisons by ranking methods," *Biometrics*, *1*:80–83 (1945).

319. Wille, R. "Concept lattices and conceptual knowledge systems," *Computers and Mathematics with Applications*, *23*:493–515 (1992).

320. Wodicka, L., et al. "Genome-wide expression monitoring in Saccharomyces cerevisiae," *Nature Biotechnology*, *15*:1359–1367 (1997).

321. Wolfinger, R., et al. "Assesinng gene significance from cDNA microarray expression data via mixed models," *Journal of Computational Biology*, *8*:625–637 (2001).

322. Woolf, P. and W. Yixin. "A fuzzy logic approach to analyzing gene expression data," *Physiological Genomics*, *3*:9–15 (2000).

323. Wu, H., et al. "MAANOVA: A software package for the analysis of spotted cDNA microarray experiment." *Software package in Bioconductor*. 1–73. 2002.

324. Xing, E. P. and R. M. Karp. "Cliff: Clustering of high-dimensional microarray data via iterative feature filtering using normalized cuts," *Bioinformatics*, *17*:306–315 (2001).

325. Xu, X. and A. Zhang. "Virtual gene: Using correlations between genes to select informative genes on microarray datasets," *Transactions on computational systems biology II, LNBI 3680*, 138–152 (2005).

326. Xu, Y., et al. "Clustering gene expression data using a graphic-theoretic approach: an application of minimum spanning trees," *Bioinformatics*, *18*:536–545 (2002).

327. Yang, I., et al. "Within the fold: assesing differential expression measures and reproducibility in microarray assays," *Genome Biology*, *3*:11 (2002).

328. Yang, Y. and N. P. Thorne. "Normalization for two-color cDNA microarray data." *IMS Lecture Notes40*. 403–418. 2003.

329. Yang, Y. H., et al. "Normalization for cDNA microarray data: a robust composite method addressing simgle and multiple side systematic variation," *Nucleic Acids Research*, *30* (2002).

330. Yeung, K., et al. "Validating clustering for gene expression data," *BMC Bioinformatics*, *17*(4):309–318 (2001).

331. Yeung, K. Y., et al. "Model-based clustering and data transformations for gene expression data," *Bioinformatics*, *17*:977–987 (2001).

332. Zar, J. H. *Biostatistical Analysis* (4th Edition). Pearson Education, 1998.

333. Zhan, Fenghuang, et al. "Global gene expression profiling of multiple myeloma, monoclonal gammopathy of undetermined significance, and normal bone marrow plasma cells," *Blood*, *99*:1745–57 (2002).

334. Zhang, A. *Advanced analysis of gene expression microarray data* (1 Edition), *1*. Science, Engineering, and Biology Informatics. World Scientific, 2006.

335. Zhang, H., et al. "Recursive partitioning for tumor classification with gene expression microarray data." *Proceedings of the National Academy of Sciences of the USA98*. 6730–6735. 2001.

336. Zhang, K. and H. Zhao. "Assessing reliability of gene clusters from gene expression data," *Functional Integrative Genomics*, 156–173 (2000).

337. Zhao, R., et al. "An adaptive method for cDNA microarray normalization," *BMC Bioinformatics*, *6* (2005).

338. Zhao, Y., et al. "Fine-Structure Analysis of Ribosomal Protein Gene Transcription," *Mollecular Cellular Biology*, *26(13)*:4853–62 (2006).

339. Zhou, Xin and K. Z. Mao. "The ties problem resulting from counting-based error estimators and its impact on gene selection algorithms," *Bioinformatics*, *22*:2507–2515 (2006).

Publications

Journals

Martinez, R. and Pasquier, N. and Pasquier, C. and Collard, M. and Lopez-Perez, L. "**Co-expressed Gene Groups Analysis (CGGA): An Automatic Tool for the Interpretation of Microarray Experiments,**" *Journal of Integrative Bioinformatics*, 3(11):1-12 (2006).

Martinez, R. and Collard, M. "**Extracted Knowledge Interpretation in Mining Biological Data: a Survey**," *International Journal of Computer Science and Applications: Special issue in Research Challenges in Information Science*, 1:1–21 (2007).

Martinez, R. and Pasquier, N. and Pasquier, C. and Collard, M. and Lopez-Perez, L. "**Analyse des Groupes de Gènes Co-exprimés : un outil automatique pour l'interprétation des expériences de biopuces,**" *Numero special de la revue RNTI: Classification,* 1:1–12 (2007). To appear.

Martinez, R. and Pasquier, N. and Pasquier, C. **GENMINER: Gene-Integrated analysis by association rules discovery**, *Journal of Biochemistry,* 1–8 (2007). Acceptation in process.

International Conferences

Martinez, R. and Christen, R. and Pasquier, C. and Pasquier, N. "**Exploratory Analysis of Cancer SAGE Data**," *Proceedings of the PKDD'2005 conference, Discovery Challenge,* 1–12 (2005).

Martinez, R. and Pasquier, N. and Pasquier, C. and Lopez-Perez, L. "**Interpreting Microarray Experiments Via Co-expressed Gene Groups Analysis**," *Proceedings of the 9th international conference on Discovery Science: Lecture Notes in Computer Science*, 4265:316-320 (2006).

Martinez, Ricardo and Collard, Martine. "**Extracted Knowledge: Interpretation in Mining Biological Data: a Survey**," *Proceedings of the Research Challenges in Information Science* (RCIS) 07, 1:1-18 (2007).

National Conferences

Martinez, R. and Pasquier, N. and Pasquier, C. and Collard, M. and Lopez-Perez, L. "**Analyse des Groupes de Gènes Co-exprimés : un outil automatique pour l'interprétation des expériences de biopuces**," *Actes des XIIIème Rencontres de la Société Francophone de Classification*, 1:1-6 (Septembre 2006)

Technical Reports

Martinez, R. and Collard, M. "**Molecular Biology Basics and Gene expression technologies,**" *I3s laboratory, Execo Project (UNSA)*, 1-55 (Mars 2004).

Martinez, R. and Collard, M. "**Microarray Data Analysis Procedure**," *I3s laboratory, Execo Project (UNSA)*, 1-55 (Fevrier 2005).

Martinez, R. "**Azerty data analysis: A full microarray analysis over the cicatrization process,**" *I3s laboratory, Execo Project (UNSA)*, 1-25 (Fevrier 2006).

Martinez, R."**Normal Discretization Algorithm for Gene Expression Technologies: NORDI,**" *I3s laboratory, Execo Project (UNSA)*, 1-18 (Juillet 2006).