Universidade Federal de Pernambuco
Centro de Informática
Pós-Graduação em Ciência da Computação

Comparative Analysis of Clustering Methods for Gene Expression Data

Ivan Gesteira Costa Filho

Dissertação de Mestrado

Orientador: Francisco de Assis Tenório de Carvalho

Co-Orientador: Marcílio Carlos Pereira de Souto

Recife, Abril de 2003

UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO DE INFORMÁTICA
PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Ivan Gesteira Costa Filho

Comparative Analysis of Clustering Methods for Gene Expression Data

Este trabalho foi apresentado à Pós-graduação em Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco como requisito parcial para obtenção do grau de Mestre em Ciência da Computação.

Orientador: Prof. Dr. Francisco de A. T. de Carvalho
Co-orientador: Prof. Dr. Marcílio C. P. de Souto

Recife, Abril de 2003

Contents

List of Figures

List of Tables

Acknowledgments

To all those who help me during the preparation of this dissertation.

Abstract

Large scale approaches, namely proteomics and transcriptomics, will play the most important role of the so-called post-genomics. These approaches allow experiments to measure the expression of thousands of genes from a cell in distinct time points. The analysis of this data can allow the the understanding of gene function and gene regulatory networks (Eisen *et al.*, 1998).

There has been a great deal of work on the computational analysis of gene expression time series, in which distinct data sets of gene expression, clustering techniques and proximity indices are used. However, the focus of most of these works are on biological results. Cluster validation has been applied in few works, but emphasis was given on the evaluation of the proposed validation methodologies (Azuaje, 2002; Lubovac *et al.*, 2001; Yeung *et al.*, 2001; Zhu & Zhang, 2000). As a result, there are few guidelines obtained by validity studies on which clustering methods or proximity indices are more suitable for the analysis of data from gene expression time series.

Thus, this work performs a data driven comparative study of clustering methods and proximity indices used in the analysis of gene expression time series (or time courses). Five clustering methods encountered in the literature of gene expression analysis are compared: agglomerative hierarchical clustering, CLICK, dynamical clustering, k-means and self-organizing maps. In terms of proximity indices, versions of three indices are analysed: Euclidean distance, angular separation and Pearson correlation. In order to evaluate the methods, a k-fold cross-validation procedure adapted to unsupervised methods is applied. The accuracy of the results is assessed by the comparison of the partitions obtained in these experiments with gene annotation, such as protein function and series classification.

Resumo

Abordagens de larga escala, entre as quais estão a proteômica e a transcritômica, fornecerão os desafios mais importantes na chamada "era pós-genômica". Estas abordagens permitem a medição da expressão de milhares de genes de uma determinada célula durante um número de intervalos de tempo. A análise deste tipo de dados permite a descoberta da função de genes e de redes reguladoras (Eisen *et al.*, 1998).

Vários trabalhos já foram realizados na análise computacional de series temporais de expressão gênica, cada qual usando diferentes conjuntos de dados, métodos de agrupamentos e índices de proximidade. Entretanto, o foco da maioria destes trabalhos está nos resultados biológicos. Validação de agrupamento foi empregado em alguns poucos trabalhos, porém ênfase foi dada na avaliação dos métodos de validação sugeridos (Azuaje, 2002; Lubovac *et al.*, 2001; Yeung *et al.*, 2001; Zhu & Zhang, 2000). Como conseqüência, existem poucas diretrizes fornecidas por estudos de validação de qual método de agrupamento ou índice de proximidade são mais apropriados para a análise de séries temporais de expressão gênica.

Este trabalho propõe a realização de um estudo comparativo de métodos de agrupamento e índices de proximidade para a análise de séries temporais de expressão gênica. Cinco métodos de agrupamento encontrados na literatura de análise de expressão gênica são comparados: agrupamento hierárquico aglomerativo, CLICK, agrupamento dinâmico, k-médias e mapas auto-organizáveis. Em termos dos índices de proximidade, versões de três índices são analisadas: distancia Euclidiana, separação angular e correlação de Pearson. Para avaliar os métodos, um procedimento de validação cruzada "k-fold" adaptado para métodos não-supervisionados é empregado. A acurácia dos resultados é medida através da comparação das partições obtidas nestes experimentos com dados de anotação de genes, como função de proteínas e classificação de series temporais.

Chapter 1

Introduction

Now that the sequences of genomes from several species have been or are about to be completed, researchers are looking towards the next step: the understanding of gene function and gene regulatory networks. Of the roughly 30,000-40,000 genes in the human genome sequence, the function of an estimated two thirds is likely to be unknown (Abbot, 1999). In terms of regulatory mechanisms, the knowledge is even scarcer. Large scale approaches, namely proteomics and transcriptomics, will play the most important role of the so-called post-genomics. These approaches allow experiments to measure the expression of thousands of genes from a cell in distinct time points, or in distinct conditions (such as a treated and a non-treated cell), providing to biologists the information about what gene is turned on, and in what condition (Abbot, 1999).

While proteins may yield the most important clues to cellular function, proteins are also the most difficult of the cell's components to detect on a large scale. This is not the case of ribonucleic acid (RNA), which is measured by transcriptomics approaches (D'Haeseleer *et al.*, 1999). When a gene is expressed in a cell, its code is first transcribed to an intermediary messenger RNA (mRNA), which is then translated

into a protein. The mRNA levels give a snapshot of the genome's plans for protein synthesis under the cellular conditions at that moment. Transcriptomics has the advantage over proteomics for the technology is simple and lends itself readily to automation and high throughput (D'Haeseleer *et al.*, 1999). But transcriptomics has the disadvantage that, although the expression levels it provides reflects the genome's plans for protein synthesis, it does not directly represents the final protein levels (D'Haeseleer *et al.*, 1999).

The post-genomic approaches represents a paradigm shift on the traditional biology experiments, where only a few genes were studied at a time (D'Haeseleer *et al.*, 1999). The analysis of the amount of data generated by large scale approaches makes the use of advanced statistical and computational methods, such as Machine Learning, necessary. Such methods can be used to discover trends and patterns in the underlying gene expression data (Bertone & Gerstein, 2001). The computational challenges in the analysis of gene expression are vast, and still open for further developments (Quackenbush, 2001).

Among these challenges, this work will attain to the problem of identification of meaningful subsets of genes by the use of clustering methods with the objective of finding co-expressed genes (Eisen *et al.*, 1998). This is accomplished by the analysis of data from gene expression time series (or time courses). In time series experiments, the expression of a certain cell is measured in some time points during a particular biological process. By knowing groups of genes that are expressed in a similar fashion through a biological process, it is possible to infer the function of these genes. Since these data sets consist of expression profiles of thousand of genes, this analysis cannot be carried out manually, making necessary the application of clustering methods.

One main aspect in finding co-expressed genes is the proximity (similarity or dissimilarity) index used in the clustering method. In this context, the index should give

emphasis on capturing relative magnitude proximity. There is a biological reason for this, as the absolute expression values of two genes can differ, but provided that the genes have a similar pattern of change through time (or similar series shape), they are considered co-expressed genes (Heyer *et al.*, 1999).

1.1 Problem Overview

In fact, there has been a great deal of work on gene expression analysis, each using distinct data sets of gene expression, clustering techniques and proximity indices. However, the majority of these works has given emphasis on the biological results, with no critical evaluation of the suitability of the proximity indices or clustering methods used. In the few works in which cluster validation was applied with gene expression data, the focus was on the evaluation of the proposed validation methodology (Azuaje, 2002; Lubovac *et al.*, 2001; Yeung *et al.*, 2001; Zhu & Zhang, 2000). As a consequence, so far, with the exception of (Costa *et al.*, 2002b; Datta & Datta, 2003), there is no validity study on which proximity indices or clustering methods are more suitable for the analysis of data from gene expression time series.

Based on this, a data driven comparative study of proximity indices and clustering methods used in the literature of gene expression analysis is accomplished in this dissertation. More specifically, versions of three proximity indices with support to missing values are compared (Gordon, 1999): Euclidean distance, Pearson correlation and angular separation. In terms of clustering methods, five algorithms are analysed in the experiments: agglomerative hierarchical clustering (Eisen *et al.*, 1998), CLICK (Sharan & Shamir, 2002), dynamical clustering (Costa *et al.*, 2002a), k-means (Tavazoie *et al.*, 1999) and self-organizing maps (Tamayo *et al.*, 1999). With the exception of dynamical clustering, all other methods are popular in the literature

of gene expression analysis.

All the experiments are performed with data sets of gene expression time series of the yeast *Saccharomyces cerevisiae*. This organism was chosen because there is a wide availability of public data, as well as the availability of an extensive functional classification of its genes. The functional classification will serve as an external data for the validation of the clustering results.

In order to evaluate the clustering methods and proximity indices, this dissertation proposes a validation methodology. The methodology is based on an adaptation of the k-fold cross-validation procedure to unsupervised methods. The accuracy of the results obtained in the k-fold cross-validation are assessed by an external index, which measures the agreement between the clustering results and an *a priori* classification data, such as gene functional classification or series classification (Jain & Dubes, 1988). Finally, in order to detect statistically significant differences in the results obtained by the distinct proximity indices or clustering methods, a bootstrap hypothesis test for equal means is applied (Efron & Tibshirani, 1993).

1.2 Dissertation Structure

The remainder of this dissertation is divided into five chapters. In Chapter 2, issues related to gene expression analysis are described. The aim of this chapter is to introduce the problem approached in this dissertation, as well as to put this work in perspective. In order to do so, basic biological concepts, experiments and measurements techniques of gene expression, as well as related work on gene expression are presented. Next, in Chapter 3, questions regarding Cluster Analysis are discussed. All proximity indices and clustering methods analysed in this work are described briefly. Furthermore, aspects of cluster validation relevant to the proposed methodol-

ogy are discussed. The validation methodology and the experimental design used in this dissertation are presented in Chapter 4. Then, Chapter 5 describes and analyses the results of the experiments. Finally, Chapter 6 brings the conclusions drawn from the results, some final remarks and future works.

Chapter 2

Gene Expression Analysis

This chapter gives a description of the problem approached in this dissertation, the computational analysis of gene expression data. Section 2.1 covers the basic concepts of molecular biology necessary for understanding gene expression. Then, Section 2.2 describes experiments regarding gene expression and the technologies used to measure the data. Section 2.3 overviews the computational challenges of gene expression analysis, focusing the analysis of time series data and cluster validation. Then, a discussion of related work in both analysis of time series data and cluster validation applied to gene expression data is presented in Section 2.4.

2.1 Gene Expression

Gene expression is the process explained by the central dogma of molecular biology on how the hereditary information contained in deoxy-ribonucleic acid (DNA) flows inside a cell, resulting in the protein synthesis (Silva, 2001). This process can be divided into two steps: the transcription where DNA molecules are used to build ribonucleic acid (RNA) molecules, and the translation where the RNA molecules

form proteins (see Figure 2.1). The final product of this process, the proteins, are responsible for providing the structural components of the cell and for the catalysis of biochemical reactions. Thus, it can be stated that the expression of the proteins determine the functional state of the cell (Primer on Molecular Genetics, 1992).

Figure 2.1: Molecular biology central dogma.

The central dogma of biology have been modified in recent years, since some organisms, such as viruses, do not fit in the original dogma scheme (Gentrop, 1999). But for the sake of simplicity, this section will attain to the original dogma, as it contains the basics concepts necessary for the understanding of gene expression.

2.1.1 DNA

The DNA molecules are responsible for storing the genetic information of the organisms. These molecules have a double helix structure, formed by a sequence of bases pairs. The particular order of bases in a determined sequence represents the genetic information contained in the DNA. These bases can be one of the following: adenine (A), cytosine (C), guanine (G), and thymine (T) (Gentrop, 1999). Some binding rules define the possible base pairs, which can be either A=T, T=A, C≡G or G≡C (each "-" symbolizes a hydrogen bond). As a consequence of these binding rules, the two sequences of bases that forms a DNA molecules are complementary to one another (Gentrop, 1999) (see Figure 2.2). Single stranded DNA molecules, which are only encountered in special conditions, have the capacity to bind with complementary sequences, in a process called hybridization. Such a process is used as a tool in most

of the techniques of molecular biology.

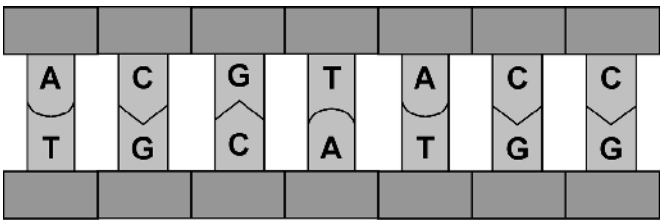

Figure 2.2: Example of a double stranded DNA molecule.

The way DNA molecules are arranged in the cells is dependent of the type of organism, which can be either a procaryote or a eukaryote. The procaryotes are the organisms without cell nucleus, while the eukaryotes are the organisms with nucleus. In the procaryotes, the DNA is arranged in a single circular DNA molecule. Whereas, in the eukaryotes, several DNA molecules, called chromosomes, are present in the cell nucleus. Each of these chromosomes is formed by billions of bases pairs.

Genes are the basic units responsible for possession and passing on a single characteristic. In other words, genes are DNA regions of the chromosomes that codes one or more proteins. In fact, only particular regions of the DNA sequences encountered in organisms represent the genes. In the region before the gene sequence (also called upstream region), regulatory regions are encountered. These regions influence the ratio of transcription (or the quantity of RNA produced) of the gene (Shamir *et al.*, 2002).

2.1.2 RNA & Transcription

In the process called transcription, the region of the DNA representing a gene is copied into a RNA molecule with the help of a enzyme called RNA polymerase (Gentrop, 1999). This enzyme binds into a upstream region called promoter sequence, which indicates where the transcription should start. Then, the enzyme slides through the

DNA sequences, building the RNA molecule base by base (see Figure 2.3). Although very similar to DNA, the RNA molecules have some distinctions: (1) RNAs are only single stranded; (2) instead of the thymine, RNAs have a uracil (U) base; and (3) RNAs degrade after some short time. The main function of the RNA is the synthesis of proteins in the cell. These molecules are divided into three groups according to their task in protein synthesis: ribosomal RNAs (rRNA) that are responsible for forming the ribosomes, transport RNAs (tRNA) that are responsible for carrying amino acids to ribosomes, and messenger RNAs (mRNA) that are responsible for encoding the genetic information contained in the genes (Gentrop, 1999).

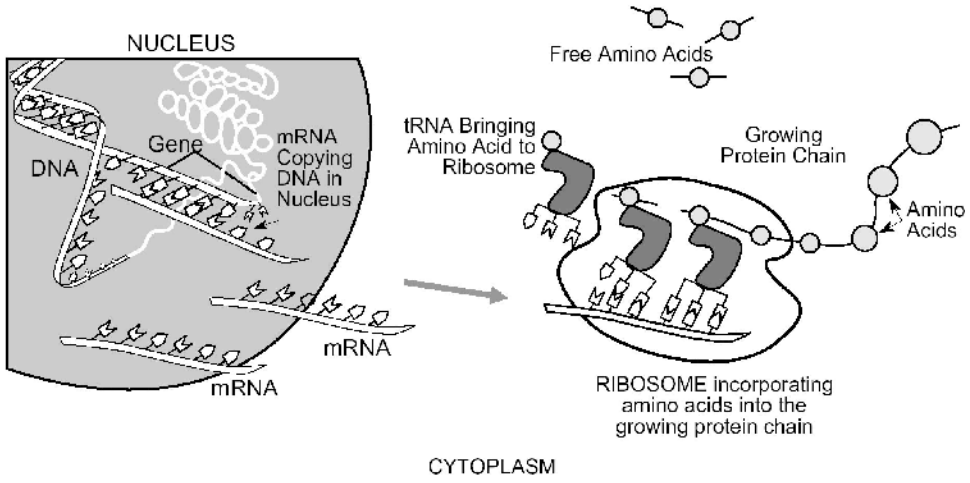

Figure 2.3: Gene expression process (Primer on Molecular Genetics, 1992).

The transcription process in eukaryotes is more complex. First, the RNA molecule is copied from a gene in the DNA, producing the primary transcript (RNA). Before leaving the nucleus, certain sequences of this primary RNA, called introns (non-coding regions), are removed by special enzymes, forming the mature RNA. In this process, certain exons sequences (codifying sequences) can also be removed, changing the final protein to be synthesised. This mechanism, called alternate splicing, plays a major role in cell differentiation. The alternate splicing makes it possible for a single gene

to codify more than one protein, all this in accordance to the cell context (Gentrop, 1999).

2.1.3 Proteins & Translation

Translation is the process of forming proteins from the information contained in the RNA. Triples of RNA bases are translated into one of the twenty amino acids, which are the building block of the proteins. The rules that map the base triplets to a amino acid is called genetic code. The translation process is coordinated by the ribosome that "reads" the mRNA molecule sequence three by three, adding the respective amino acid in the end of the synthesised protein, with the help of tRNA molecules (see Figure 2.3).

Proteins, the final product of the gene expression, are vital to the cell functioning, since they are responsible for providing the structural components of the cell and for the catalysis of biochemical reactions (Primer on Molecular Genetics, 1992). In order words, it is the presence of the proteins that dictates how the cell is working. In conclusion, it can be stated that the functional state of a cell is determined by the number of proteins available inside the cell in a certain instant (D'Haeseleer *et al.*, 1999).

2.2 Gene Expression Experiments

Traditional experiments of molecular biology were inherently local, examining one or just a few genes. These experiments were based on a reductionist view, where by explaining the parts, one could get a view of the whole. With the advent of genomics, and consequently large scale gene expressions methods, it would be a huge

effort to analise such a number of genes using the traditional approach. This amount of data requires a holist analysis of these experiments, where the data is handled in a global fashion, without the need to go down to low level details such as the biochemical reactions. Nowadays, this global analysis have been carried out with the aid of statistical and computational methods (D'Haeseleer *et al.*, 1999).

There is a number of purposes for the analysis of experiment of large scale gene expression, such as (Lubovac, 2001):

- Finding co-expressed genes, or genes that participate in the same biological process, such as the mitotic division cycle (Eisen *et al.*, 1998).
- Improving the diagnosis of diseases, such as cancer (Brown *et al.*, 2000).
- Discovery of gene regulatory networks (D'Haeseleer *et al.*, 1999).
- Analysis of cell response to drug treatment (Dopazo *et al.*, 2001).

For each specific purpose, a distinct type of experiment design is necessary. Often, there are two basic design types of experiments for gene expression. In one type, the behavior of the expression levels is observed through time, in other words, gene expression time series (or time courses) are obtained. This design is used for the finding of co-expressed genes and also for the inference of regulatory networks. In the other type of design, samples of gene expression of distinct tissues or individuals are obtained (condition experiments) (Eisen *et al.*, 1998). For example, when there is interest in disease diagnosis, the gene expression of distinct individuals, be them infected or not, are measured and compared (Brown *et al.*, 2000). This arrangement is also used for the discovery of regulatory networks, where one can be interested in comparing a normal cell to a mutated one (D'Haeseleer *et al.*, 1999). In the analysis of drug response, a mix of both arrangements can be performed, as there is interest

in comparing the time series expression of a treated individual, with the time series of non treated individual (Dopazo *et al.*, 2001).

These gene expression experiments were only possible with the development of a number of techniques capable of measuring large scale gene expression. These techniques differ in some aspects such as: the substance being measured (RNA or proteins); the process of reading the results; the way of manufacturing the artifacts; and the domain of the technology (public or private). Each of these techniques have distinct characteristics in relation to accuracy, reproducibility and cost (be it financial cost or time cost). Apart from these distinctions, most of them are based on the same molecular biology principle called hybridization, in which nucleic acids have the capacity of recognizing and combining with complementary sequences.

In the Sections 2.2.1 to 2.2.4 measurement technologies with widespread use and related to this work are be described. This work will concentrate on RNA expression based technologies, as the use of protein based techniques is not widespread, given the lack of accuracy and reproducibility of these techniques (D'Haeseleer *et al.*, 1999).

2.2.1 cDNA microarray

Complementary DNA (cDNA) microarrays, developed at the Stanford University, consists of small glass slides, where cDNA are deposited with the aid of robotics (Schena *et al.*, 1995). The idea behind the functioning of cDNA microarrays is very simple (Kain, 2001). For each gene to be measured, a sequence complementary to the gene sequence is defined (these small sequences are called probes). The probes have size ranging from 20 to 30 bases, in a way that there is a low probability of the probes hybridizing with sequence others than the target sequence. The probes are replicated a high number of times (around thousands). Then, a robot fixes the probes in a

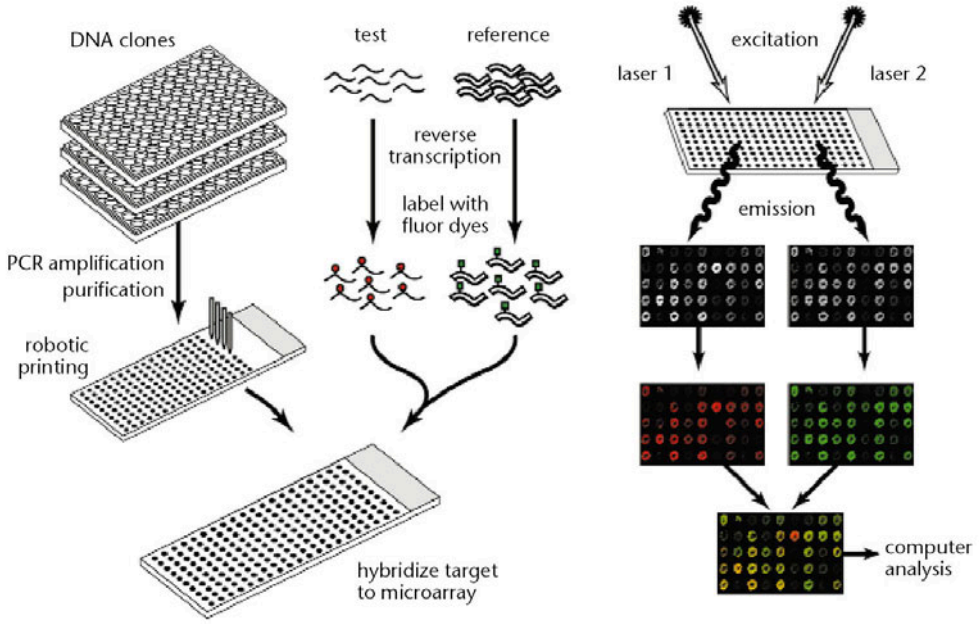

Figure 2.4: Schema of the cDNA microarray (Duggan *et al.*, 1999)

certain spot of a glass slide. At the end, the small slide will have thousands of DNA spots, placed side by side, each spot containing thousands of cDNA probes copies designed to hybridize with RNA from a certain gene.

In the next step, the RNA of the cell is separated and transcribed to cDNA, given that RNA molecules are unstable and would degrade before the experiment is over. Afterwards, the cDNA molecules are marked with green fluorescent labels. Additionally, the RNA of a control cell is also separated and transcribed, but theses molecules are marked with red fluorescent labels. The cDNA of both cells are poured in the slide. After some time, the slide is washed, removing the cDNA which has not hybridized with the probes. Next, the slide is scanned, giving as a result a image with all the spots intensities (the whole process is illustrated Figure 2.4). The digital image of the slide is then processed using computational methods, for the purpose of calculating the intensity obtained by each RNA. Figure 2.5 shows one segment of

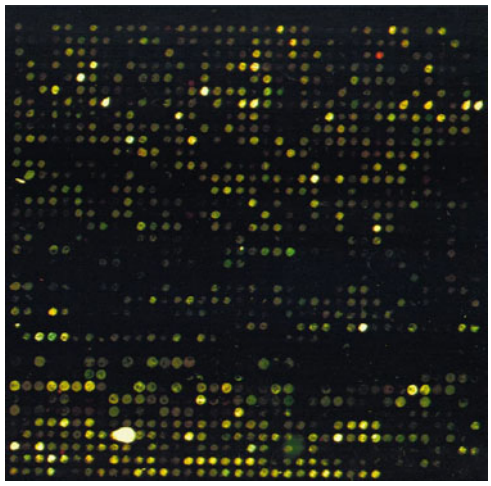

Figure 2.5: Segment of a image of a fluorescent cDNA microarray.

such images.

The advent of microarray technologies were only possible due to two main factors (Kain, 2001). First, robotics permitted the manufacturing of the slides (also called chips) with only a few centimeters, containing around 10.000 of probes spots, organised side by side as a matrix. The others factors were the sequencing of organism and the discovery of its genes, as only with this data, it was possible to construct the microarray probes (Brow & Bostein, 1999).

One problem of the cDNA microarrays is that distinct measurements with material from the same cell can obtain distinct results. Certain steps in the process are influenced by the environment and the way of execution, causing variability of the final results. Not to mention about the image processing procedure, where the lack of precision of the robots on the placement of the spots and limitations of the scanner represent additional noise on the data. This can be observed in Figure 2.5, where not all spots are uniformity placed and some neighbor spots signals are merged.

These variability problems are attacked in cDNA microarray by the use of the RNA from a control cell, as described before. The idea is to use the RNA from a single

control cell to all slides being measured in a certain experiment. The final expression level of a gene is calculated by the log ratio of the measured (Cy5) and control cell (Cy3) expression levels, as show in Equation 2.1.

$$e = \log(Cy5/Cy3) \tag{2.1}$$

Another variability factor very common in cDNA microarrays is called bleaching (this problem can not be solved by the procedure just described). In the majority of the scanners, the red and green signals are scanned separately. The problem is that the final spot intensities are influenced by the order of scanning. As a result, special procedures should be used in order to normalise the intensities of the red and green signals (for more details see Schuchhardt *et al.* (2000) and Yang *et al.* (2001)).

The cDNA technique has as advantage, among others, the high number of genes having the expression simultaneously measured, which can reach a 10.000 in a single slide (D'Haeseleer *et al.*, 1999). In fact, there is no limitation on the number of genes, as more than one slide can be used to measure the RNA of a certain cell. For some organisms, as the yeast *Saccharomyces cerevisiae*, it is possible to measure the expression of the whole genome. The technique also permits a great flexibility on the experiments design, as the probes used and consequently the genes measured can be chosen among any gene with known sequences. The main problem is the financial cost of the microarrays, which is still very high, limiting the number of conditions and replications realised in gene expression experiments.

2.2.2 Oligonucleotide array

The oligonucleotide array (or Gene Chip) is a private technology developed by Affymetrix (Lipshutz *et al.*, 1999). Its functioning is very similar to the cDNA microarray, although the technologies differs in two aspects, the manufacturing of the slides and how the variability problem is treated. The Gene Chips are constructed via a optic process, where the probes are synthesised base by base on the chip surface. The design of a chip containing a new set of probes is very expensive, but once it is done, the chip arrangement can be produced in large scale with a lower cost. In order to reduce the effect of variability, probes with same sequence are placed in 20 to 40 spots on the chips. This reduces the signal to noise ratio and improves the accuracy of the RNA quantification. Additionally, a mismatch spot (MM), containing probes with sequence with one distinct base of the original probe (PM), is placed next to each PM spot, so as to reduce the effect of cross-hybridization and background noise. The expression value of a certain gene is measured by the average of the differences of the PM probe intensity and its neighbor MM probe.

The DNA chip also allows the measurements of a high number of genes (up to 50.000 genes per chip). On the other hand, oligonucleotide chips do not offer the same flexibility as cDNA arrays, for there is a limited number of chip designs available with a fixed set of probes. On the long run, this problems should be minimised, as the number of probes packed in the chip tends to get higher, and genomes of several organisms will be fully revealed. Other advantage of this technology is that Affymetrix supply all the equipments, in contrast to cDNA microarray, where there are a number of choices, from where to buy the probes, to the software used in the image processing (Bowtell, 1999). As a consequence, experiments with DNA chips are more standardised, making it easier to compare (and analyse) experiments carried out by distinct laboratories, which is not the case of cDNA microarray.

2.2.3 SAGE

The serial analysis of gene expression (SAGE) technology is very distinct from the other methods described in this section, as SAGE uses sequencing technology to measure the expression (D'Haeseleer *et al.*, 1999). Initially, the RNA is transcribed to DNA. Then, sequences with ten bases, capable of uniquely identifying the source RNA, are extracted from the DNA molecules. Next, these small sequences are joined together in a single sequence, and then sequenced. The expression of the genes corresponds to the quantity of repeated ten bases sequences encountered at the sequencing results.

Some advantages of this method, among others, is its higher accuracy in relation to the array technologies, and the fact that the sequence of the measured RNA does not need to be known a priori. Additionally, the process uses sequencing technology that is already available in most of the molecular biology laboratories. However, the whole process consumes a lot o time. When a high number of genes are measured, the process can become quite complex, as there is the need of a lot of sequencing. As a result, experiments with this technology only measure the expression of hundreds of genes.

2.2.4 Real Time PCR

The real time PCR, also known as kinetic PCR, is an automation of the reverse transcribed polymerase reaction (RT-PCR) technique. In the RT-PCR, the RNA of the desired genes are reverse transcribed (RT) to cDNA molecules (note that the RT stands for reverse transcribed and not for real time). Then, the cDNA is replicated using the polymerase chain reaction (PCR) (D'Haeseleer *et al.*, 1999). This process has to be repeated for each target gene. Finally, with the use of high resolution gels,

the number of cDNA molecules are quantified. This process is not of a parallel nature, what can make it very time consuming. Furthermore, if the whole process in not very well controlled, there will be a high variability in the results (Bustin, 2002).

In the real time PCR, the amplification, detection and quantifications steps are carried automatically by a special machinery. All this reduces the time and complexity necessary to carry the experiments. Additionally, this automation leads to a higher sensivity, specificty and reproducibility of the experiments. As a result, real time PCR has a high precision in measuring the gene expression. However, the experiments are still time consuming. As a consequence, only a small set of genes (hundreds of genes) are measured in experiments using this technology (Bustin, 2000).

2.3 Computational Analysis

As stated before, the analysis of the amount of data generated by the approaches with large scale gene expression can only be developed with the aid of statistical and computational methods (D'Haeseleer *et al.*, 1999). There is a number of computational challenges for the analysis of gene expression data. Among them, the following should be point out (Sharan & Shamir, 2002):

- Clustering: identification of meaningful subsets of genes or conditions (Eisen *et al.*, 1998; Heyer *et al.*, 1999; Tamayo *et al.*, 1999).
- Classification: building classifiers based on the conditions or time series with objective such as disease diagnosis (Golub *et al.*, 1999) and gene function discovery (Brown *et al.*, 2000; Kuramochi & Karypis, 2001).
- Feature Selection: find a set of genes that are differentially expressed through the distinct conditions (Bo & Jonassen, 2002; Golub *et al.*, 1999; Heyer *et al.*,

1999).

- Normalization: how should the results of distinct conditions or experiments be normalised (Dopazo *et al.*, 2001; Yang *et al.*, 2001).

- Image processing: the analysis of the *micrroarray* images (Dougherty *et al.*, 1997; Yang *et al.*, 2001).

The focus of this work will be on the use of clustering methods for the analysis of time series data. In the next section, basic aspects of this type of analysis will be discussed. The other concern of this work is the validation of the clustering methods, therefore, validation issues for gene expression analysis will be covered in the subsequent section.

2.3.1 Analysis of Gene Expression Time Series

In experiments of gene expression time series, a certain cell is induced to a particular biological process. Then, the gene expression of the cell is measured in some particular time points. The analysis of this data is focused on finding co-expressed genes, more specifically, genes that have similar patterns of expression change through time. By knowing groups of genes that are expressed in a similar fashion through a biological process, biologists are able to infer gene function and gene regulation mechanisms (Eisen *et al.*, 1998).

Clustering is the main technique for the analysis of gene expression time series. In such an approach, the major aspect in finding co-expressed genes is the proximity (similarity or dissimilarity) index used (Sherlock, 2000).

Proximity Indices

Proximity indices measure the degree of alikeness between two objects. In the context of gene expression analysis, the proximity index should give emphasis on capturing relative magnitude proximity between two gene series. There is a biological reason for this, as the absolute expression values of two genes can differ, but provided that the genes have a similar pattern of change through time (the series have similar shape), they are considered co-expressed genes (Eisen *et al.*, 1998).

Figure 2.6 shows the time series of three genes during six time points (0, 30, 50, 70, 100 and 120 minutes). Apparently, all the time series are very distinct, but the genes represented by the blue and green lines can be stated to be co-expressed. Both genes behave in a similar fashion, their expression level goes up until the 70-minute time point, and than go down until the end of the process. Actually, the intensity value of the gene in green is the double of the gene in blue in most of the time points. On the other hand, these two series have a distinct behavior in relation to the gene in red, which has it expression decreasing through the whole process.

Pre-processing

Another important issue on clustering gene expression time series is the removal of uninformative time series. During a particular biological process only a few genes will be active and changing the expression levels through time. The other genes can either be housekeeping genes or not expressed during that particular process. The former represents genes that are always active, independently of the particular biological process going on. While the latter type of genes have low expression levels in all time points. These two types of genes do not need to be analysed, given that they are uninformative in relation to that particular process. In fact, the removal of

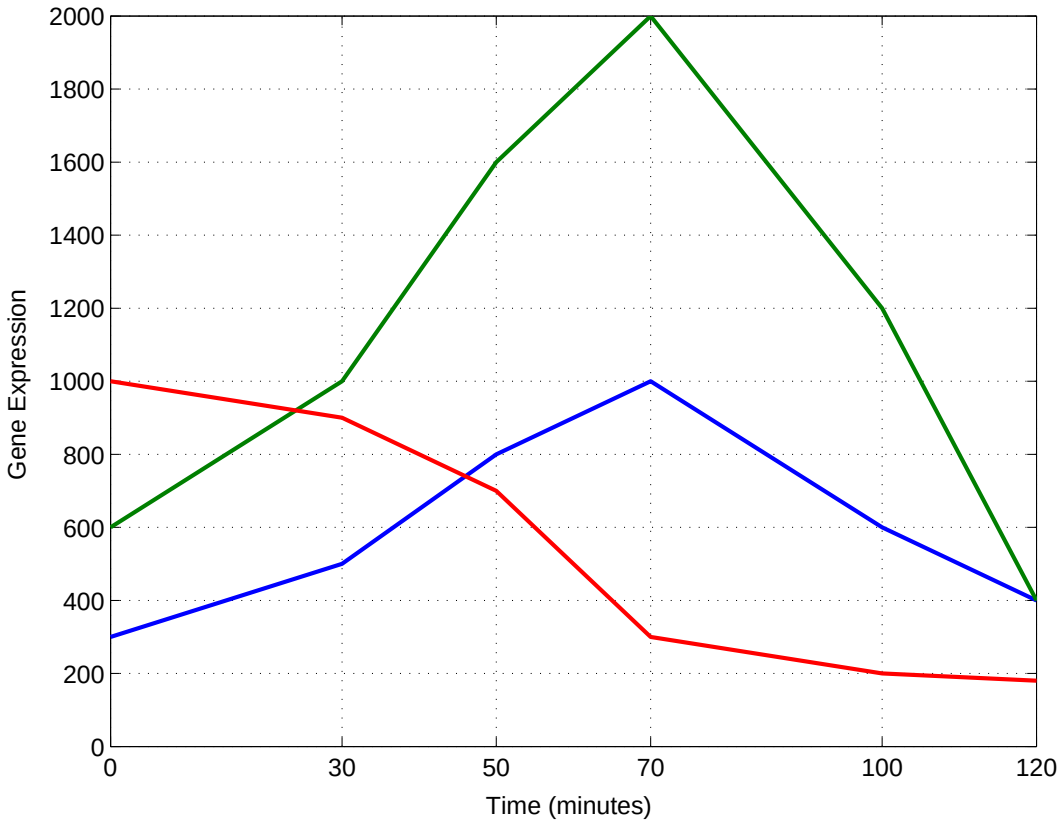

Figure 2.6: Example of three gene expression time series.

these genes reduces the computing time of the clustering methods. Furthermore, the removal can also enhance the accuracy of the results, as the presence of these genes can disturb certain clustering methods.

There are two widespread methods for dealing with such uninformative genes. The fold approach, used in Eisen *et al.* (1998) and Tamayo *et al.* (1999). In this procedure, only time series where the absolute expression levels changes for at least n folds are considered. The other approach, proposed by Heyer *et al.* (1999), genes were ranked according to their mean and variance. Then, a percentage of the genes with the lowest mean and variance values are removed (Heyer *et al.* (1999) removed the genes in the lowest 25% of both ranks).

2.3.2 Validation of Gene Expression Analysis

Most of the work on gene expression analysis relies only on ad-hoc observations to evaluate the results. As an exception there are few studies where validation issues are approached. However, these works are focused on the evaluation of the proposed validation methodologies. They do not address the results obtained by the application of the validation methodologies to compare the performance of distinct clustering methods (with the exception of Costa *et al.* (2002b) and Datta & Datta (2003)). As a consequence, so far, there are few guidelines obtained by validity studies on which proximity indices or clustering methods are more suitable for the analysis of data from gene expression time series.

One relevant issue in cluster validation, in the context of gene expression analysis, is the use of external biological data to validate the results. Some validity methodologies requires a labelling (or classification) of the elements. One common practice is to use external sources of data related to the objective of study. In validity studies of gene expression data, functional classification of the genes are largely applied as external data (Gertein & Janssen, 2000; Lubovac, 2001; Yeung *et al.*, 2001; Zhu & Zhang, 2000). One advantage of functional classifications is, among others, the availability of functional classifications schemes covering thousand of genes, such as MYGED for yeast (Mewes *et al.*, 2002), GenProtEC for E. coli (Riley, 1998), the Gene Ontology Project (The Gene Ontology Consortium, 2000), among others. There are also other types of external data used in the literature, such as, regulatory regions (van Helden *et al.*, 2001; Zhu & Zhang, 2000), enzymatic classification (Lubovac *et al.*, 2001), metabolic pathways and protein structure (Gertein & Janssen, 2000).

2.4 Related Work

2.4.1 Analysis of Gene Expression Time Series

Eisen *et al.* (1998) presented one of the first applications of clustering methods for the analysis of gene expression time series. In their study, a hierarchical unweighed pairwise average linkage method (UPGMA) was used with Pearson correlation to cluster data from seven distinct time series experiments from yeast. The results confirmed that genes with similar functions tend to cluster together. Additionally, the study proposed a graphical representation of the results, which is now widely used in the field. In this representation, the resulting dendrogram has its leaves reordered by the mean expression levels of the series, in a way that gene with similar profiles were close in the tree. In the side of the ordered tree, the expression levels of the genes are represented in a colored table, where over expressed genes have green values and under expressed genes red values (see Figure 2.7).

A number of other clustering methods have also been applied for the analysis of gene expression time series, among them, k-means (Tavazoie *et al.*, 1999), self-organizing maps (Jonsson, 2001; Tamayo *et al.*, 1999), dynamical clustering (Costa *et al.*, 2002a), graph theoretical approaches (Sharan & Shamir, 2002), principal component analysis (PCA) (Raychaudhuri, 2001) and largest first cluster3ing algorithm (Zhu & Zhang, 2000). Most of these works are often applications of distinct computational methods to a similar set of gene expression data, not proposing any new aspects in the analysis of gene expression. Thus, just some of then will be described in details in this dissertation.

In terms of proximity indices, novel proposals have been presented for the analysis of data from gene expression time series. The jackknife Pearson correlation, proposed

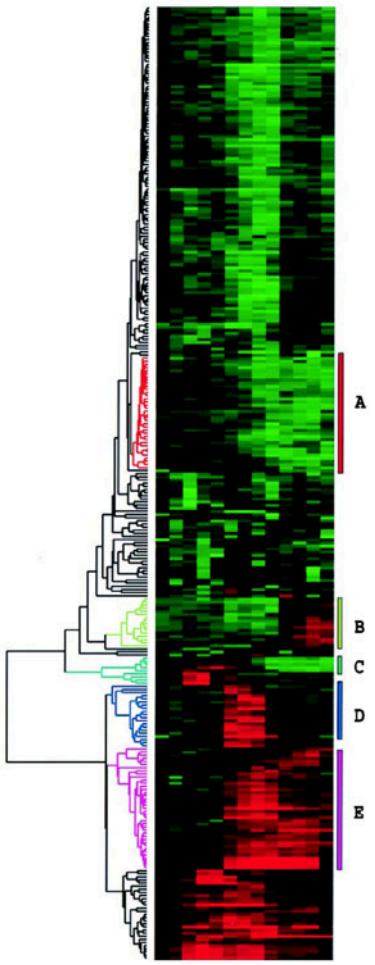

Figure 2.7: Example of the graphical representation suggested by Eisen *et al.* (1998).

in Heyer *et al.* (1999), has as objective to handle time series with outliers time points. The idea of this proximity index is to calculate the Pearson correlation between two series, not taking into consideration the values of one time point. This is repeated for all time points in the data set, excluding one distinct time point at a time. In the end, the highest value obtained is then taken as result. The work developed an analysis of the proposed proximity index. The results showed that the number of false positives decreased with the use of the jackknife Pearson correlation in relation to the original Pearson correlation.

Costa *et al.* (2002a) proposed a symbolical description of gene expression time series from multiple experiments, where each variable take as a value a time series, in conjunction with a symbolic version of a proximity measure. As shown in the beginning of this section, the proximity indices used in the analysis of gene expression time series emphasize the capture of shape proximity. However, none of them will suitably measure shape proximity with data containing gene expression time series from multiple experiments, unless special data handling is made. In the symbolic approach, the shape similarity of each time series is calculated independently, and aggregated at the end. The symbolical description was evaluated with the yeast data set (Eisen *et al.*, 1998), obtaining significant better results in comparison to the traditional approaches.

A different approach have been explored in Brown *et al.* (2000). They applied supervised methods for classifying the gene function, given data of gene expression time series. Only a subset of the expression series data used in Eisen *et al.* (1998) was employed in that work. This subset consisted of the genes belonging to one of the five functional classes, which clustered well using hierarchical clustering. The supervised methods applied obtained high precision levels, particulary in the experiments using support vector machines (SVM). Despite this, the number of false positives was high for some classes.

2.4.2 Validation of Gene Expression Analysis

The works performed in Lubovac (2001) and Lubovac *et al.* (2001) evaluated the use of internal criteria such as compactness and isolation of the clusters, as well as the use of external criteria that compare the clustering results in relation to gene annotation. More specifically, the gene annotations used were the enzymatic and functional classifications of the proteins. The results indicated that internal criteria

can be misleading as they did not show correspondence to gene annotation. The study also proposed a relative entropy criterion. This criterion compares the distribution of classes in a group with the background distribution (distribution of classes in the complete data set). The more the group differs from the background distribution, the more discriminated is the group.

A framework to find the ideal number of clusters was presented in Azuaje (2002). In order to do so, the study applied Dunn's validity index to the results of the clustering methods. The work analysed the methodology with expression data from leukemia (Golub *et al.*, 1999), but the framework can also be applied to time series data.

Another validation methodology was proposed by Yeung *et al.* (2001). In proposed approach, the data is clustered using all but one condition, which is used to access the accuracy of the results. This is accomplished in a jackknife fashion, where for each step, one of the conditions is hold out. This procedure is repeated for the total number of conditions. The work compared some clustering methods using distinct data sets, but the authors refrained to draw conclusions from the results, given that only a small number of data sets were available.

Costa *et al.* (2002b) applied replication analysis for the purpose of evaluating the cluster stability in the analysis of gene expression data. More specifically, the work evaluated Self Organizing Maps (SOM), dynamical clustering and UPGMA hierarchical clustering with data from yeast time series. The preliminary results showed that both SOM and dynamical clustering obtained stable results.

So far, the most complete comparative analysis of clustering methods for gene expression data was performed in Datta & Datta (2003). They proposed a validation methodology based on the jackknife procedure, similar as the one in Yeung *et al.* (2001), in conjunction to three novel relative validation indices. The work evaluated:

UPGMA hierarchical clustering, k-means, Diana, Fanny, Model-based clustering and hierarchical clustering with partial least squares. Only two data sets were used in this evaluation, the sporulation data set (Chu *et al.*, 1998) and a simulated data set. In the results, Diana achieved the best performance, followed closely by the Model-based and k-means methods. Both hierarchical methods obtained the poorest results.

Zhu & Zhang (2000) investigated the relation of gene expression clustering with gene function and promoter regions. The study used the yeast time series from Eisen *et al.* (1998) as the gene expression data set, and thirteen major classes from the Munich Information Center for Protein Sequences Yeast Genome Database (MYGED) (Mewes *et al.*, 2002) as the functional classification. The results showed that genes with similar expression levels do not necessarily share the same promoter regions and functions, even though both gene function and promoter regions do help to gain overview of the expression data.

A similar and broader study was performed in Gertein & Janssen (2000). In that work, a set of yeast data sets were compared with the MYGED functional classification. Only some functional classes had a strong relation to the gene expression profiles. The reason for this could be, among others, the vague definitions of some functions and the great overlap of the classification. The study suggested that other types of data should be used, such as protein structure and regulatory sequences.

A feasibility study was also performed in the context of supervised methods for the classification of gene function. Kuramochi & Karypis (2001) evaluated the classification precision of SVM for the fifty biggest MYGED classes, given the data of gene expression from the yeast (Eisen *et al.*, 1998). The results showed that only in eight classes the classifiers obtained reasonable accuracy. Such a study concluded that the number of gene expression data sets available is not enough to build classifiers for all functional classes.

Chapter 3

Cluster Analysis

Relevant issues in cluster analysis are covered in this chapter. Initially, Section 3.1 describes characteristics and the basic functioning of all clustering methods analysed in this dissertation, while Section 3.2 presents the proximity indices. Section 3.3 covers issues on cluster validity relevant to this work. More specifically, validity indices and related validation methodologies are described in details.

3.1 Clustering Methods

Five distinct clustering methods are analysed in this dissertation. These methods are: agglomerative hierarchical clustering (Eisen *et al.*, 1998), k-means (Tavazoie *et al.*, 1999), self-organizing maps (Tamayo *et al.*, 1999), dynamical clustering (Costa *et al.*, 2002a) and CLICK (Sharan & Shamir, 2002). All of them, with the exception of dynamical clustering, have a widespread use in the gene expression analysis. The dynamical clustering was included because it was utilized in previous work by the author (Costa *et al.*, 2002a). With the exception of the hierarchical clustering, all the other methods yield partitions as results. In the following subsections the

characteristics of these clustering methods are described.

3.1.1 CLICK

CLICK (Cluster Identification via Connective Kernels) (Sharan & Shamir, 2002) is a recently developed method based on graph theory. Such a method is robust to outliers and does not make assumptions on the number or structure of the clusters. Although CLICK does not take the number of classes as an input, by the use of the homogeneity parameter, one can force the generation of a larger number of clusters.

The method initially generates a fully connected weighted graph, with the objects as vertices and the the similarity between the objects as the weights of the edges. Then, CLICK recursively divides the graph in two, using minimum weight cut computations, until a certain kernel condition is met. The minimum weight cut divides the graph in two in a way that the sum of the weights of the discarted vertices is minimum. If a partition with only one object is found, the object is put apart in a singleton set.

The kernel condition tests if a cluster formed by a given graph is highly coupled, and consequently, if it should not be further divided. In order to do so, CLICK uses a statistical model, assuming that the similarities between objects (the weights of the edges) are normally distributed. An EM (Expectation-Maximization) method is used to build two similarity distributions, one containing similarity between mates edges (objects that should be clustered together) and other for non-mates edges (objects that should not be clustered together). The Kernel test consists in verifying if the probability of containing only mate edges exceeds the probability of containing non-mate edges in a given graph. If the test is true, them, the tested graph is taken as a final cluster, otherwise, it will be divided in two (using minimum cut computations).

More formally, let E be the objects data set; G be an fully connected graph, where each vertice represents an object in E, and the weight of the edges are the similarity between the two connected edges; minWeightCut(G) be the function that finds the minimum weight cut, returning two fully connected graphs; and S be the singleton set; the method is defined by the following recursive function (Sharan & Shamir, 2002):

```
function formKernel(G, S)
begin
    if G = {v} then
       S = S ∪ {v};
    else
        if G is a kernel then
            output G;
        else
            (H, V) = minWeightCut(G);
            formKernel(H, S);
            formKernel(V, S);
        end;
    end;
end;
```

3.1.2 Dynamical Clustering

Dynamical Clustering is a partitional iterative algorithm that optimises the best fitting between classes and their representation, using a predefined number of classes (Diday & Simon, 1980). Starting with prototypes values from random selected individuals, the method works on two alternates steps: an allocation step, where all individuals are allocated to the class with the prototype with lower dissimilarity, followed by a representation step, where a prototype is constructed for each class. A major problem of this algorithm is its sensitivity to the selection of the initial partition. As a consequence, the algorithm may converge to a local minimum (Jain &

Dubes, 1988). In order to prevent the local minimum problem, a number of runs with different initialisations are executed. Then, the best run, based on some cohesion measure, is taken as the result (Jain & Dubes, 1988). Another characteristic of this method is its robustness to noisy data. In addition, when particular proximity index and prototype representations are used, the method guarantees optimisation of local criterion (Diday & Simon, 1980). With respect to the proximity indices investigated in this work, only the use of the Euclidean distance version with data containing no missing data guarantees the minimisation of the squared error.

More formally, this method looks for a partition P of k classes from an object set E and a vector L of k prototypes, where each prototype represents one class of P. This search is done by minimising the criterion Δ of fitting between L and P (Verde *et al.*, 2000):

$$\Delta(P^*, L^*) = \min \{\Delta(P, L) | P \in P_k, L \in L_k\} \tag{3.1}$$

where P_k is the set of partitions of E in k classes and L_k is the set of prototypes associated to the classes.

More specifically, let D be a given dissimilarity function; and e_j be the j^{th} object in the set E, where $j = 1, \ldots, n$; the method works as follows:

```
1. Initialisation

    P = (C_1, ..., C_i, ..., C_k) is initialised by allocating one random
    object from E to each class;
    All individuals are allocated to the class with the closer
    representative object;

2. Representation Step

    for i = 1 to k do
        prototype G_i is set to the centroid of objects from C_i;
    end;

3. Allocation Step

    test = 0;
    for j = 1 to n do
        find class C_m of e_j;
        find class C_l such that C_l min_{i=1, ..., k} D(e_j, G_i);
        if m ≠ l then
            test = 1;
            C_l = C_l ∪ {e_j} and C_m = C_m − {e_j} ;
        end;
    end;

4. Termination Test

    if test = 0 stop, else go to 2;
```

The criterion $\Delta(P, L)$ is defined as:

$$\Delta(P, L) = \sum_{i=1}^{k} \sum_{x \in C_i} D(x, G_i) \tag{3.2}$$

3.1.3 *k*-means

k-means is another type of iterative relocation algorithm, which is widely used in cluster analysis studies (Jain *et al.*, 1999). This method is a special case of the dynamical clustering (Jain *et al.*, 1999). Thus, they share some characteristics, such

as robustness to outliers, the use of a predefined number of classes and the sensitivity to the initial partition. Furthermore, like the dynamical clustering method, k-means also optimises the squared-error criterion when the Euclidean distance is used and there is no missing data. The main distinctions from the dynamical clustering method are that k-means only works with centroids representations of the classes (Jain *et al.*, 1999), and only one object is reallocated in each allocation step (dynamical clustering reallocates all objects in each allocation step). As a result, a strategy on how the objects are considered to reallocation has to be defined. One of such strategies is to generate a random order of the input objects (Jain & Dubes, 1988).

More formally, this method looks for a partition P of k classes from an object set E and a vector L of k prototypes, where each prototype represents one class of P. Let D be a dissimilarity function; and O be a random ordering of the objects, where O_j represents the jth object in O, for $j = 1, \ldots, n$; the method works as described below:

```
1. Initialisation

    P = (C_1, ..., C_i, ..., C_k) is initialised by randomly allocating
    each object in E to one class in P;
    Generate a random order O of the objects in E;
    for i = 1 to k do
        prototype G_i is set to the centroid of objects from C_i;
    end;

2. Allocation and Representation Step

    test = 0;
    for j = 1 to n do
        find class C_m of 0_j;
        find class C_l such that C_l = min_{i=1, ..., k} D(0_j, G_i);
        if m ≠ l then
            test = 1;
            C_l = C_l ∪ {x} and C_m = C_m − {x} ;
            recalculate prototypes G_m and G_l;
        end;
    end;

3. Termination Test

    if test = 0 stop, else go to 2;
```

3.1.4 Self-Organizing Map

The Self-Organizing Map (SOM) is a type of neural network suitable for unsupervised learning (Kohonen, 1997). SOMs combine competitive learning with dimensionality reduction by smoothing the clusters with respect to an *a priori* grid. One of the main characteristics of these networks is the topological ordering property of the clusters generated. Clusters objects are mapped in neighbour regions of the grid, delivering an intuitive visual representation of the clustering. SOMs are reported to be robust and accurate with noisy data (Mangiameli *et al.*, 1996). On the other hand, SOM suffers from the same problems such as those of dynamical clustering: sensibility to the initial parameters settings and the possibility of getting trapped in local minimum

solutions (Jain *et al.*, 1999).

The SOM method works as follows. Initially, one has to choose the topology of the map, for example a 3 x 3 grid as in Figure 3.1. All the nodes are linked to the input nodes by weighted edges. The weights are first set at random, and then iteratively adjusted. Each iteration involves randomly selecting an object x and moving the closest node (and its neighbourhood) in the direction of x. The closest node is obtained by measuring the Euclidean distance or the dot product between the object x and the weights of all nodes in the map. The neighbourhood to be adjusted is defined by a neighbourhood function, which decreases through time.

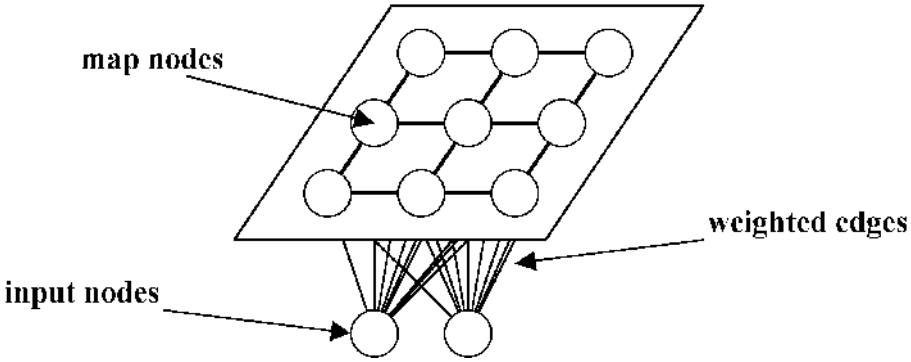

Figure 3.1: Example of a SOM with topology 3 x 3 and two input variables

Let E be the set of objects; F be the neighbourhood function; l be the learning rate; D be a proximity index; k x o be the topology of the map; N_{ij} be the node in position i x j, for $i = 1, \ldots, k$ and $j = 1, \ldots, o$; the method is defined as (Haykin, 1994):

1. Initialisate randomly the weights of the edges between the input nodes and the map;

2. Choose randomly an object x from E;

3. Find node N_{lm} such that $N_{lm} = \min_{i=1, \ldots, k;\ j=1, \ldots, o} D(x, N_{ij})$;

4. Update the weights of the node N_{lm} and its neighbourhood towards the object x by the function F in accordance to a learning rate l;

5. Go to Step 2 until the number of interactions is reached;

One problem with SOM is the high number of parameters to be selected, which includes, the topology, the learning rate, the neighbourhood function, neighbourhood radius, among others. The success of the map is dependent on selection of these parameters (Kohonen, 1997). Although there is no analytical procedure to select the parameters, there are some guidelines on how one should proceed in these selections. In terms of the learning rate, one current practice is to decrease the value towards 0 in the final iterations. The form of variation is not critical, but one popular practice is to divide the training in two phases. In the first phase, the ordering phase, a large initial radius and learning rates are used. Then, in the convergence phase, smaller initial radius and learning rate are selected (Haykin, 1994).

As stated before, one of the main characteristics of SOM is to create a visual topological map of the clusters. Such maps should have a number of nodes well above the number of real clusters in the data (Vesanto & Alhoniemi, 2000). By a visual inspections of the map, one can select the neighbour nodes that represents each cluster. However, this process is time consuming and open to subjectivity. In this study, there is the need of an objective way to assign the nodes to the final clusters, as a high number of experiments are necessary, and it is not a good practice to include subjective procedures in the validation process.

One way to overcome the problem just described is to cluster the nodes after training the map, by the use of another clustering method (the weights of each node represents the node input pattern). In this latter clustering, the number of cluster should be equal to the number of clusters in the data. The resulting partition will state what nodes are related to each cluster. In Vesanto & Alhoniemi (2000), k-means and hierarchical clustering are employed for this task, all of them obtaining good recovery accuracies. For the sake of simplicity, this study will only employ the average linkage hierarchical clustering to the SOM nodes.

Another alternative is to use maps with a unidimensional layer, where the number of nodes is equal to the number of clusters (Mangiameli *et al.*, 1996). With this type of topology, the SOM method becomes very similar to k-means. But as k-means is already analysed in this study, there would be no use of analysing SOM with this type of topology.

3.1.5 Agglomerative Hierarchical Clustering

Agglomerative hierarchical methods are procedures for transforming a distance matrix into a dendrogram (Jain & Dubes, 1988). These algorithms start with each object representing a cluster, then the methods gradually merge theses clusters into larger ones. Among the different agglomerative methods, there are three broader used variations: complete linkage, average linkage, and single linkage. These variations differ in the way cluster representations are calculated (see Jain & Dubes (1988) for more details). Depending on the variation used, the hierarchical algorithm is capable of finding non-isotropic clusters, including well-separated, chain-like, and concentric clusters (Jain *et al.*, 1999). However, since such methods are deterministic, individuals can be grouped based only on local decisions, which are not re-evaluated once decisions are made. As a consequence, these methods are not robust to noisy data (Mangiameli *et al.*, 1996).

Due to the fact that the methodology applied in this work is only adequate for the evaluation of partitions, the hierarchies are transformed into partitions before being evaluated. One way to perform this, is to cut the dendrogram in a certain level, as shown is Figure 3.2. Additionally, the hierarchical method are also used as initialisation to other partitional methods. This practice improves the initial conditions of the partitional method that receives the hierarchical results as input (Jain & Dubes, 1988).

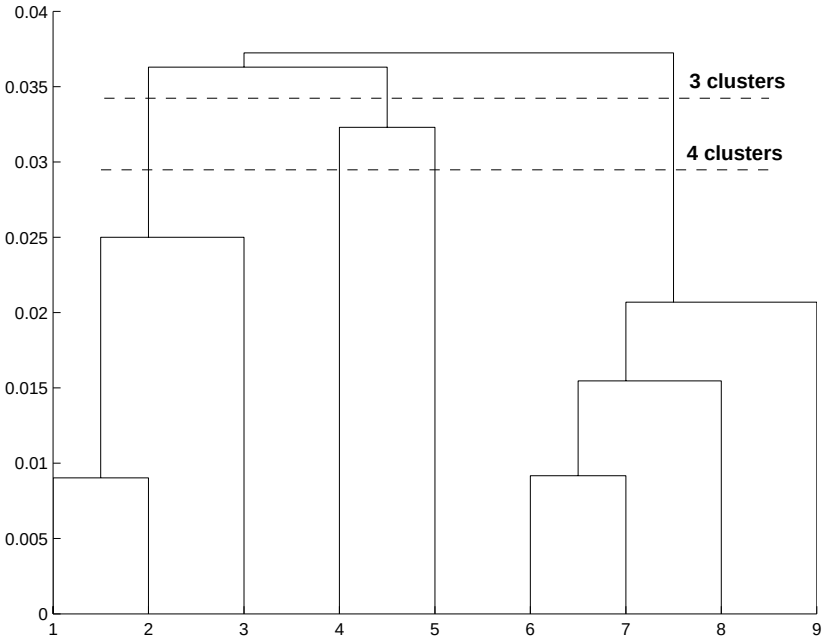

Figure 3.2: Example of two cuts in a dendrogram with nine objects. The two dashed lines represent respectively cuts with three and four clusters.

This dissertation will focus on the average linkage hierarchical clustering method or UPGMA (unweighed pair group method average), as it has been extensively used in the literature of gene expression analysis (Eisen *et al.*, 1998). In this method, the proximity between two cluster is calculated by the average proximity between the objects in one group and the objects in the other group. Given the object set E, the average linkage hierarchical clustering works as follows (Jain *et al.*, 1999):

1. Calculate a proximity matrix containing the proximity between all the objects in E. Each object is treated as a cluster;

2. Find the most similar pair of clusters and merge these two clusters in a single one;

3. Update the proximity matrix, by recalculating the proximities of the new cluster formed by the merge operation. The proximity between two cluster is calculated by the average proximity between the objects in one group and the objects in the other group;

4. Go to step 2 until only one cluster is left;

3.2 Proximity Indices

In order to cluster a set of objects, clustering methods need an index of alikeness or association between the data objects. This can be achieved by the use of proximity (similarity or dissimilarity) indices that calculate the alikeness of two objects. For the choice of a suitable index, the type of the variables and the characteristics of the index should be taken into consideration. For example, in the case of quantitative variables, the use of an Euclidean distance captures the proximity between objects considering the absolute magnitude of the values, while correlation-type indices measure the proximity in relation to the relative magnitudes of the values (Gordon, 1999).

As this dissertation focuses on measures used in the literature of gene expression analysis, proximity indices for quantitative variables with emphasis on relative magnitude proximity are investigated. Additionally, as gene expression data sets often contain missing data, the proximity indices studied need also to support missing data (Gower, 1971). Based on this, versions of the following proximity indices are studied: Euclidean distance, Pearson correlation and angular separation (Gordon, 1999). The Euclidean distance does not capture the relative magnitude proximity, unless the objects have their values normalised or standardised. Because of this, for the Euclidean

distance version, the effect of normalisation and standardisation are also investigated.

The indices studied can be formally defined as follows. Let x_{ik} denote the k^{th} quantitative value (expression value of time point k) of the i^{th} object (gene) where $i = 1, \ldots, n$ and $k = 1, \ldots, p$; the modified version of the Euclidean distance between the i^{th} and j^{th} objects is defined as:

$$d_{ij} = \frac{\sum_{k=1}^{p}(x_{ik} - x_{jk})^2 \delta_{ijk}}{\sum_{k=1}^{p} \delta_{ijk}} \tag{3.3}$$

where

$$\delta_{ijk} = \begin{cases} 0, & \text{if } x_{ij} \text{ or } x_{ik} \text{ is missing} \\ 1, & \text{otherwise} \end{cases}$$

Such a version of the Euclidean distance (Eq. 3.3) - ED, for short - is a dissimilarity index, with values near zero representing similar objects. As this version is based on the Euclidean distance, it shares the desirable characteristics of the original distance such as the ability to detect compact and isolated clusters. However, attributes with high scale values can dominate the others. This can be solved by the normalisation of the data attributes (Jain & Dubes, 1988).

The equation for the version of the Pearson correlation - PC, for short - is as follows:

$$s_{ij} = \frac{\sum_{k=1}^{p}(x_{ik} - \overline{x}_i)(x_{jk} - \overline{x}_j)\delta_{ijk}}{(\sum_{k=1}^{p} \delta_{ijk}) \sqrt{\sum_{k=1}^{p} (x_{ik} - \overline{x}_i)^2 \vartheta_{ik} \sum_{k=1}^{p} (x_{jk} - \overline{x}_j)^2 \vartheta_{jk}}} \tag{3.4}$$

where

$$\overline{x}_i = \frac{\sum_{k=1}^{p} x_{ik} \vartheta_{ik}}{\sum_{k=1}^{p} \vartheta_{ik}};$$

$$\vartheta_{ik} = \begin{cases} 0, & \text{if } x_{ik} \text{ is missing} \\ 1, & \text{otherwise} \end{cases};$$

and δ_{ijk} as in Eq. 3.3.

PC is a correlation type index that measures the angle similarity of two data vectors, yielding values between -1 and 1, where 1 represents similar objects and -1 dissimilar objects.

The equation for the version of the angular separation - *AS*, for short - is as follows:

$$s_{ij} = \frac{\sum_{k=1}^{p} x_{ik} x_{jk} \delta_{ijk}}{(\sum_{k=1}^{p} \delta_{ijk}) \sqrt{\sum_{k=1}^{p} x_{ik}^2 \vartheta_{ik} \sum_{k=1}^{p} x_{jk}^2 \vartheta_{jk}}} \tag{3.5}$$

where δ_{ijk} is as defined in Eq. 3.3; and ϑ_{ik} is as defined in Eq. 3.4.

AS (Eq. 3.5) is also a correlation type index, with the same characteristics as those of *PC* (Eq. 3.4). The difference between them is that *AS* measures the angle similarity from the origin, while *PC* measures the angle similarity from the mean of the data. Both correlations differ from *ED* (with no prior normalisation) in that they do not consider the vector size when measuring the proximity.

As clustering methods work either with pairwise similarity (s_{ij}) or a dissimilarity (d_{ij}) indices, sometimes, it is necessary to transform similarities in dissimilarities and vice-versa (Gordon, 1999). In the case of similarities bound to [-1,1] and dissimilarities bound to [0,1], the two following equations can be used to transform, respectively, similarities in dissimilarities (Eq. 3.6) and dissimilarities in similarities (Eq. 3.7):

$$d_{ij} = \frac{1 + s_{ij}}{2} \tag{3.6}$$

$$s_{ij} = 1 - 2d_{ij} \tag{3.7}$$

Among a number of pre-processing procedures available, two of them broadly used in the literature of gene expression are analysed (Tamayo *et al.*, 1999). The first procedure is the normalisation of the data vectors (genes) so that they have a norm equal to one. This procedure requires the values of the data vectors to be positive. The other is a standardisation procedure that makes the data vectors to have zero mean and standard deviation equals to one. The application of either procedure makes *ED* capture relative magnitude dissimilarity. As the data sets used in this work contain missing data, both procedures were adapted to support missing values.

Formally, let x_{ik} denote the k^{th} quantitative value (expression value of time point k) of the i^{th} object (gene) where $i = 1, \ldots, n$ and $k = 1, \ldots, p$; the standardised values are obtained by the following equation:

$$z_{ik} = \begin{cases} \text{missing}, & \text{if } x_{ik} \text{ is missing} \\ \dfrac{x_{ik} - \overline{x}_i}{s_i}, & \text{otherwise} \end{cases} \tag{3.8}$$

where

$$s_i^2 = \frac{\sum_{k=1}^{p} (x_{ik} - \overline{x}_i)^2 \vartheta_{ik}}{\left(\sum_{k=1}^{p} \vartheta_{ik} - 1\right)};$$

and $\overline{x}_i$ and ϑ_{ik} are defined as in Eq. 3.4.

In order to obtain the normalised values, the following equation is used:

$$y_{ik} = \begin{cases} \text{missing}, & \text{if } x_{ik} \text{ is missing} \\ \dfrac{x_{ik}}{\sum_{k=1}^{p} x_{ik} \vartheta_{ik}}, & \text{otherwise} \end{cases} \tag{3.9}$$

where ϑ_{ik} is defined as in Eq. 3.4.

3.3 Cluster Validity

The evaluation of clustering results in an objective and quantitative fashion is the main objective of cluster validity. Despite of its importance, cluster validity is rarely employed in applications of cluster analysis. The reasons for this are, among others, the lack of general guidelines on how cluster validity should be carried out, and the great need of computer resources (Jain & Dubes, 1988).

In this section, procedures and tools for cluster validity relevant to this work are described. More specifically, Section 3.3.1 describes aspects of indices for cluster validity. Next, in Section 3.3.2, methodologies used for the evaluation of clustering methods are explained.

3.3.1 Validity Indices

The aim of validity indices is to measure objectively the adequacy of a structure returned from a cluster analysis. Such indices should measure the alikeness that the structure gives true information about the data, or that the structure captures intrinsic characteristics of the data (Jain & Dubes, 1988).

The validity indices vary in two main aspects: the type of structure, and the type of criteria measured. In terms of structure, validity indices can handle hierarchies, partitions or individual clusters. With respect to the criterion, there are three types: external, internal and relative. The external criteria asses the accuracy by comparing the structure with *a priori* information. Internal criteria measure the accuracy by comparing the structure with the input data (and only the input data). The last

type, relative criteria, are used to compare two cluster structures, in order to point out which structure is better in some sense. This work will attain to validity indices appropriate for evaluating partitions and external criteria. The reasons for this choice are, among others, the fact that: most of the methods evaluated gives partitions as result and external labels are available for some data sets. Alternatively, internal criteria could be used, allowing the addition of unlabelled data sets in this experiments. However, there are a number of difficulties in applying internal indices, specially in a comparative analysis, where the choice of the index could favour some specific clustering methods (Dubes, 1998).

External Indices

External indices are used to assess the degree of agreement between two partitions (U and V), where partition U is the result of a clustering method and partition V is formed by an *a priori* information independent of partition U, such as a category label (Jain & Dubes, 1988). There are a number of external indices defined in the literature, such as Jaccard, Rand and corrected Rand (or adjusted Rand) (Jain & Dubes, 1988). One characteristic of most of these indices is that they can be sensitive to the number of classes in the partitions or to the distributions of elements in the clusters. For example, some indices have a tendency to present higher values for partitions with more classes (Rand), others for partitions with a smaller number of classes (Jaccard) (Dubes, 1987). The corrected Rand, which has its values corrected for chance agreement, does not have any of these undesirable characteristics (Milligan & Cooper, 1986). Thus, the corrected Rand index is the only external index used in the validation methodology proposed by this work. However, in order to explain the general idea of external indices, the Rand and Jaccard indices are first described.

More formally, let D be a set of objects, $U = \{u_1, \ldots, u_r, \ldots, u_R\}$ be the partition

obtained by clustering D, and $V = \{v_1, \ldots, v_c, \ldots, v_C\}$ be the partition defined by the *a priori* classification of the objects in D. Given two objects x_i and x_j from D, the external indices can be expressed in terms of the following indicator functions (Jain & Dubes, 1988):

$$I_U(i,j) = \begin{cases} 1, & \text{if } x_i \in u_r \text{ and } x_j \in u_r \text{ for } r \leq R \\ 0, & \text{otherwise} \end{cases} \tag{3.10}$$

$$I_V(i,j) = \begin{cases} 1, & \text{if } x_i \in v_c \text{ and } x_j \in v_c \text{ for } c \leq C \\ 0, & \text{otherwise} \end{cases} \tag{3.11}$$

These indicator functions define the following contingency table:

		I_U		
		1	0	
I_V	1	a	b	m_1
	0	c	d	$M - m_1$
		m_2	$M - m_2$	M

(3.12)

In this table, the agreements of the partitions are represented by a and d, where a indicates the number of individual pairs in the same classes in both partitions, and d denotes the number of individual pairs in separate classes in both partitions. The disagreements are indicated by b and c, where b represents pairs in the same class in partition V but in separate classes in partition U, and c represents pairs in separate classes in V but in the same class in U. M is the total number of element pairs, m_1 is the sum of a and b, and m_2 the sum of a and c. From these values the Rand and Jaccard indices can defined as:

$$\text{Rand} = \frac{a+d}{a+b+c+d} \tag{3.13}$$

$$\text{Jaccard} = \frac{a}{a+b+c} \tag{3.14}$$

Both Jaccard (Eq. 3.14) and Rand (Eq. 3.13) indices yield values in the interval [0,1], where the more the value approximates to 1 the higher the agreement is. The difference among them is that Jaccard does not take into consideration the agreement represented by term d. These two indices suffers from the same problem, as there is no indication of how good a partition is given the value obtained. For instance, Milligan & Cooper (1986) showed that partitions with a high number of clusters can obtain Rand index values near 1 independently of their quality. One way to overcome this problem is to correct the indices for random agreement. The corrected Rand index (Hubbert & Arabie, 1985), for example, can be described as the following equation:

$$\text{corrected Rand} = \frac{a+d-n_c}{a+b+c+d-n_c} \tag{3.15}$$

Such an index is obtained by adding to the Rand index a correcting term (n_C), which adjusts the statistic by estimating random agreement (Hubbert & Arabie, 1985). This correction considers that the baseline distributions of the partitions are fixed. The corrected Rand index can take values from -1 to 1, with 1 indicating a perfect agreement between the partitions, and values near 0 or negatives corresponding to cluster agreement found by chance. In fact, an analysis by Milligan & Cooper (1986) confirmed that corrected Rand scores near to 0 when presented to clusters generated from random data, and showed that values greater than 0.05 indicate clusters not achieved by chance.

As the n_c term cannot be defined from a, b, c and d, the exact corrected Rand equation can only be expressed in terms of the contingency table of partitions U and V, as defined below (Jain & Dubes, 1988):

	v_1	v_2	$\cdots$	v_C	
u_1	n_{11}	n_{12}	$\cdots$	n_{1C}	$n_{1.}$
u_2	n_{21}	n_{22}	$\cdots$	n_{2C}	$n_{2.}$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
u_R	n_{R1}	n_{R2}	$\cdots$	n_{RC}	$n_{R.}$
	$n_{.1}$	$n_{.2}$	$\cdots$	$n_{.C}$	n

(3.16)

where n_{ij} represents the number of objects that are in clusters u_i and v_i; n is the number of all objects in the partitions; $n_{i.}$ indicates the number of objects in cluster u_i; and $n_{.j}$ indicates the number of objects in cluster v_j. Thus, the exact corrected Rand equation is as follows:

$$\text{corrected Rand} = \frac{\sum_i^R \sum_j^C \binom{n_{ij}}{2} - \binom{n}{2}^{-1} \sum_i^R \binom{n_{i.}}{2} \sum_j^C \binom{n_{.j}}{2}}{\frac{1}{2}[\sum_i^R \binom{n_{i.}}{2} + \sum_j^C \binom{n_{.j}}{2}] - \binom{n}{2}^{-1} \sum_i^R \binom{n_{i.}}{2} \sum_j^C \binom{n_{.j}}{2}} \tag{3.17}$$

3.3.2 Validation Methodologies

Methodologies for cluster validity are inherently statistical. The task of such procedures is to find how unusual or valid a certain cluster structure is. One procedure very popular in cluster validity is the Monte Carlo test (Jain & Dubes, 1988). In this test, a number of data sets are built given a null model (usually this null model represents no structure or randomness). These data sets are clustered and evaluated by a validity index, giving as result a distribution of values (baseline distribution).

Then, the observed value (the cluster structure to be evaluated) is compared with the baseline distribution (obtained by the null model) using statistical tests.

Monte Carlo tests have been widely employed in cluster validity studies (Gordon, 1999; Jain & Dubes, 1988; Milligan, 1996). However, this test presents some problems. First, Monte Carlo consumes a lot of computer resources, as a high number of replications are needed for building the baseline distribution (from 500 to 1000 replications) (Jain & Dubes, 1988). Nowadays, this may not be a big problem, as processing time is becoming cheaper. However, for complex experiments, such a number of replications can still be a problem. Second, the definition of the null model is not a trivial task. In fact, there is a wide range of null models types, each with some advantages and disadvantages (Gordon, 1996). Indeed, Gordon (1996) suggested that more than one null model should be employed in validation analysis, what makes the validation process more complex and time consuming.

Other statistical methodology that has an increasing use in cluster validity is bootstrap. Bootstrap has been used to build consensus trees (Felsenstein, 1985), and to measure cluster stability (Jain & Moreau, 1987). In fact, bootstrap samples of an original data set could be used to build a null model (the hypothesis of no structure or randomness) (Jain & Dubes, 1988). These bootstrap samples can be obtained either by resampling the objects or the attributes in the data set. In order words, all the problems present in Monte Carlo tests related to the choice of the null model would be avoided. But still, the number of resamples necessary for building accurate test is still high (500 to 1000) (Efron & Tibshirani, 1993).

Replication analysis is another well known procedure for cluster validation (McIntyre & Blashfield, 1980). This procedure, based on cross-validation, measures the stability of a method in clustering a certain data set. This method is also based on making a number of samples from the original data set. However, it requires a small

number of replications to perform the test (at least 30). Since in this work the number of experiments necessary for comparing the proximity indices and clustering methods is high (around a 100 distinct experiments), it would be too costly to use either the Monte Carlo or the bootstrap test. Because of this, the validation methodology proposed in this dissertation is based on the replication analysis. As a consequence, only the replication analysis is described with further details.

Replication analysis

Replication analysis is a procedure based on cross-validation with the aim of measuring the stability (or replicability) of clustering methods. This is done by comparing the results obtained by clustering subsets of data randomly drawn from a single population (McIntyre & Blashfield, 1980; Morey *et al.*, 1983). The higher the similarity of the partitions obtained by clustering the distinct subsets, the higher the stability of the given method is. It is important to point out that stability and accuracy are not necessarily correlated. Even though the solution given by a method can be stated as stable, it does not mean that the solution has a good accuracy. In contrast, stability is related to the reliability of the results, being a necessary characteristic of accurate results (McIntyre & Blashfield, 1980).

In the supervised learning context, the cross-validation procedure is performed in two steps. First, a sub set of the data (training set) is used to training the method, obtaining a classifier (or function) as result. Then, another subset of the data (test set) is presented to this classifier. In contrast, clustering results are not classifiers (or functions) as in supervised learning. In fact, the results obtained with clustering methods consist of structures such as partitions or hierarchies, which cannot be directly used to classify other objects. This problem is overcome in replication analysis by the use of the nearest centroid procedure. In this procedure, for each cluster of

a partition, a centroid is calculated. In order to classify one object, the distances between the object and the centroids are calculated. The object is then assigned, given a proximity index, to the class with the nearest centroid. This method resembles steps of some clustering algorithms (SOM, k-means, average linkage hierarchical clustering) when an element is assigned to a cluster. As a consequence, the use of this procedure should not include an additional bias in the validation process (McIntyre & Blashfield, 1980).

Basically, the replication procedure works as follows. The data set is randomly divided in two disjoint data sets A and B. Then, the objects in A are clustered. The resulting partition is used in conjunction to the nearest centroid procedure to classify the objects in B (Nearest Centroid step). Next, the objects in B are clustered in the same way as the objects in A (Direct Clustering step). Finally, the partitions obtained in the Nearest Centroid Step and Direct Clustering step are compared (both partitions are obtained from the set B). The higher the agrement between these partitions, the higher the stability. This procedure is then repeated a number of times with distinct partitions A and B.

Formally, let D be the data set; n the number of clusters; A_i and B_i the two random subsets from the data set D; R_i the resulting partition of the set A_i; C_i the set of centroids of partition R_i; P_i and NC_i the resulting partitions of the set B_i, for $i = 1, \ldots, k$; then, the replication analysis procedure works as follows (McIntyre & Blashfield, 1980):

1. for $i = 1$ to k do

2. D is randomly divided into two disjoint folds A_i and B_i;

3. Apply the clustering method to the set A_i obtaining partition R_i with n clusters as result;

4. Calculate the n centroids of the clusters in R_i, forming C_i;

5. Calculate the distances between the centroids in C_i and the objects in B_i;

6. Assign the objects of B_i to the nearest centroid in C_i obtaining partition NC_i as result (Nearest Centroid step);

7. Apply the clustering method to the set B_i obtaining partition P_i with n clusters as result (Direct Clustering step);

8. Measure the agreement of partitions NC_i and P_i with an external index;

The idea behind the replication analysis is simple. The stability is measured by comparing the partition obtained by a clustering method, with the partition obtained in a independent sub set of data (via the nearest centroid procedure). Monte carlo experiments of this procedure (McIntyre & Blashfield, 1980) have shown that the replication analysis is useful for the evaluation of clustering methods. Furthermore, its was also demonstrated that there was a high correlation between the stability and the accuracy of the results.

An evaluation of variant procedures was carried out in Breckenridge (1989). Besides the nearest centroid procedure the study evaluated the nearest neighbour procedure and the quadratic discriminant analysis classification rule. Monte Carlo experiments demonstrated that the nearest neighbor procedure obtained better results for detecting instability than the other procedures. However, its was stated that the nearest centroid procedure should be used in situations were the data is clustered by relative magnitude or shape, which is the context of this work (Breckenridge, 1989).

Chapter 4

Methods and Experiments

This chapter presents in details the validation methodology and the experimental design used in the comparative analysis. Section 4.1 introduces the validation methodology proposed in this dissertation. Then, Section 4.2 describes the data sets used in the experiments. The last section describes the experimental design utilised in the experiments, as well as some implementation issues specific to each clustering method.

4.1 Validation Methodology

In this section, a methodology for cluster validity with the objective of comparing the accuracy of clustering methods and proximity indices is described. This methodology consists of the use of an adaptation of the k-fold cross-validation procedure to unsupervised methods. Such a procedure is inspired in the replication analysis (McIntyre & Blashfield, 1980). The accuracy of the results obtained in the k-fold cross-validation is measured with the use of an external index. The mean values of the external index obtained by each clustering method (or proximity index) are compared two by two

with a bootstrap hypothesis test, in order to asses the statistical significance of any difference in the results.

4.1.1 Cross-validation

The comparison of two supervised learning methods is, often, accomplished by analysing the statistical significance of the difference between the mean of the classification error rate, on independent test sets, of the methods evaluated. In order to evaluate the mean of the error rate, several (distinct) data sets are needed. However, the number of data sets available is often limited. One way to overcome this problem is to divide the data sets into training and test sets by the use of a k-fold cross validation procedure (Mitchell, 1997). This procedure can be used to compare supervised methods, even if only one data set is available. The procedure works as follows. The data set is divided into k disjoint equal size sets. Then, training is performed in k steps, each time using a different fold as the test set and the union of the remaining folds as the training set. Applying the distinct algorithms to the same folds with a k at least equal to thirty, the statistical significance of the differences between the methods can be measured, based on the mean of the error rate from the test sets.

In unsupervised learning, when there is an *a priori* classification of the data set available, the comparison between two methods can also be done by detecting the statistical significance of the difference between the mean value of a certain external index (it is important to point out that the a priori classification is not used in the training, but only to evaluate the results). But again, the number of training sets available is also limited. Monte Carlo and bootstrap tests could be used to generate additional training sets, but they have a high computational cost. This work proposes a method to overcome these problems. Such a methodology is an adaptation of the k-fold cross-validation procedure for unsupervised methods.

In the proposed unsupervised k-fold cross-validation procedure, the data set is also divided in k folds. For each iteration of the procedure, one fold is used as the test set, and the remaining folds as the training set. The training set is presented to a clustering method, giving a partition as result (training partition). Then, the nearest centroid technique is used to build a classifier from the training partition. The centroid technique calculates the proximity between the elements in the test set and the centroids of each cluster in the training partition (the proximity must be measured with the same proximity index used by the clustering method evaluated). A new partition (test partition) is then obtained by assigning each object in the test set to the cluster with nearest centroid. Next, the test partition is compared with the *a priori* partition (or *a priori* classification) by using an external index (this *a priori* partition contains only the objects of the test partition). At the end of the procedure, a sample with size k of the values for the external index is available.

Formally, let D be the data set; n the number of clusters; F_i the i^{th} test fold (or set); R_i the resulting partition of the training set $D - F_i$; C_i the set of centroids of partition R_i; T_i the resulting partition of test fold F_i; and P_i the *a priori* partition with the objects from F_i, for $i = 1, \ldots, k$; then, the unsupervised k-fold cross-validation procedure works as follows:

1. D is randomly divided into k equal and disjoint folds F_i;

2. for $i = 1$ to k do

3. Apply the clustering method to the set $D - F_i$ obtaining partition R_i with n clusters as result;

4. Calculate the n centroids of the clusters in R_i, forming C_i;

5. Calculate the distances between the centroids in C_i and the objects in F_i;

6. Assign the objects of F_i to the nearest centroid in C_i obtaining partition T_i as result;

7. Measure the agreement of partitions T_i and P_i with an external index;

The general idea of the k-fold cross-validation procedure is to observe how well data from an independent set F_i is clustered, given the training results. If the results of a training set have a low agreement with the *a priori* classification, so should have the results of the respective test set. In conclusion, the objective of the procedure is to obtain k observations of the accuracy of the unsupervised methods with respect to an *a priori* classification, all this with the use of independent test folds.

The proposed procedure is an adaptation of the replication analysis described in Section 3.3.2. However there are two main differences between these procedures. In the replication analysis, the nearest centroid technique is used to analyse the stability of the results. In order to do so, the test set is also clustered with the same method from the Step 3, obtaining one partition of the test set as result. The stability is measured by comparing this partition with the partition obtained in Step 6 (partition T_i) (see algorithm in Section 3.3.2) . On the other hand, the unsupervised k-fold cross validation is used to analyse the accuracy of the results. This is done by comparing the partition from Step 6 with an *a priori* classification. Second, the "test folds"

(fold B) of the replication analysis are not drawn independently from the others, in contrast to the independent test folds of the unsupervised k-fold procedure.

4.1.2 Hypothesis Test

Two-sample hypothesis tests are applied to measure the significance of the difference between the sample means of two random variables. In this work, these two samples are formed by the values of the external index provided by the unsupervised k-fold cross-validation procedure for the two clustering methods (or proximity indices) to be compared. The test indicates if a sample mean of a clustering method can be stated to be superior to the other.

The hypothesis test used in this work is based on bootstrap resampling. Bootstrap is a data based method used to measure the accuracy of statistical estimates (Efron & Tibshirani, 1993). The idea behind bootstrap is simple; given a sample, elements are randomly drawn with replacement, forming a bootstrap sample. The estimate is build by calculating a desired statistics from a large number of bootstrap samples. The bootstrap method was chosen due to its capacity to build accurate estimates when a limited number of elements are available in the samples. Furthermore, the bootstrap method has the advantage of not making parametric assumptions about the sample distributions. However, such a method is not so accurate as other tests such as the t-test (Efron & Tibshirani, 1993).

More formally, let r be the number of bootstrap samples replicates; $\mathbf{y}$ be the sample $\mathbf{y} = (y_1, \ldots, y_i, \ldots, y_n)$; $\mathbf{z}$ be the sample $\mathbf{z} = (z_1, \ldots, z_j, \ldots, z_m)$; and $\overline{y}$ and $\overline{z}$ be two sample means. The hypothesis of the test are:

$$H_0 : \overline{y} = \overline{z}$$

$$H_1 : \overline{y} < \overline{z}$$

Then, the bootstrap procedure to compare samples $\mathbf{y}$ and $\mathbf{z}$ is defined as (Efron & Tibshirani, 1993):

1. Form samples $\tilde{\mathbf{y}}$ and $\tilde{\mathbf{z}}$ by applying the following transformation to $\mathbf{y}$ and $\mathbf{z}$: $\tilde{y}_i = y_i - \overline{y} + \overline{x}$ for $i = 1, \ldots, n$ and $\tilde{z}_j = z_j - \overline{z} + \overline{x}$ for $j = 1, \ldots, m$; where $\overline{y}$ and $\overline{z}$ are the samples means, and $\overline{x}$ is the mean of the combined sample
2. for $k = 1$ to r do
3. Form the bootstrap samples $\mathbf{y}^{*k}$ and $\mathbf{z}^{*k}$ from $\tilde{\mathbf{y}}$ and $\tilde{\mathbf{z}}$
4. Calculate $t(\mathbf{y}^{*k}, \mathbf{z}^{*k})$, given the statistic defined by Equation 4.1

$$t(A, B) = \frac{\overline{a} - \overline{b}}{\sqrt{\frac{\overline{s}_a^2}{n} + \frac{\overline{s}_b^2}{n}}} \tag{4.1}$$

where

$$A = (a_1, \ldots, a_i, \ldots, a_n);$$
$$B = (b_1, \ldots, b_j, \ldots, b_m);$$
$$\overline{s}_a^2 = \textstyle\sum_{i=1}^{n} (a_i - \overline{a})^2/(n-1)$$
$$\text{and } \overline{s}_b^2 = \textstyle\sum_{i=1}^{m} (b_i - \overline{b})^2/(n-1).$$

5. Calculate the statistic $t(\mathbf{y}, \mathbf{z})$ with the original samples $\mathbf{y}$ and $\mathbf{z}$ (Eq. 4.1), and find the achieved significance level (ASL) (Eq. 4.2), given $W = \{(\mathbf{y}^{*1}, \mathbf{z}^{*1}), \ldots, (\mathbf{y}^{*k}, \mathbf{z}^{*k}), \ldots, (\mathbf{y}^{*r}, \mathbf{z}^{*r})\}$

$$ASL = \frac{\left| \left\{ \forall (\mathbf{y}^{*k}, \mathbf{z}^{*k}) \in W | t(\mathbf{y}^{*k}, \mathbf{z}^{*k}) \geq t(y, z) \right\} \right|}{r} \tag{4.2}$$

6. If ASL is smaller than a defined significance level (α) then H_0 is rejected.

4.2 Data Sets

The yeast *Saccharomyces cerevisiae* is one of the most well studied biological organism, in fact, it is one of the first organisms to have the whole genome known (Heyer *et al.*, 1999). Since there is a wide availability of public data from the yeast, as well as the availability of an extensive functional classification of its genes, allowing the validation of the clustering results, this dissertation focus on data from this organism. More specifically, one classification scheme and two data sets from the Yeast are used. The *Yeast Functional Classification* consists of a classification scheme of half of the known yeast genes. The two data sets contain data of gene expression time series: the *Yeast All* and the *Mitotic Cell Cycle* data sets. From these expression data sets, only genes belonging to a certain classification scheme are used to form the final data sets. More specifically, from the *Yeast All* expression data, two data sets are formed by the use of two distinct functional classifications schemes devised from the *Yeast Functional Classification*. In terms of the *Mitotic Cell Cycle* data set, also two data sets are formed, one is formed by the *Yeast Functional Classification* scheme and the other is formed by a series shape classification performed in Cho *et al.* (1998).

4.2.1 Yeast Functional Classification

Munich Information Center for Protein Sequences Yeast Genome Database (MYGD) is the main scheme for classifying protein function of the yeast organism (Mewes *et al.*, 2002). This classification scheme is currently composed of a tree with 249 classes spread in five levels. The genes are catalogued in accordance to information from biochemical and genetic studies, where genes with a large amount of information tend to be classified in higher levels of the tree (the number of classes in each level is shown in Table 1). Genes can be assigned to more than one class, consequently

the overlap of classes is large, with genes being assigned to an average of 2.9 classes. Out of the 6200 known yeast ORFs (Open Reading Frames), around 3900 belong to at lest one of the MYGD classes. (Original data available at: http://mips.sf.de/proj/yeast/catalogues).

Level	NUMBER OF CLASSES
1	16
2	107
3	85
4	39
5	2

Table 4.1: Number of classes in the five levels of the MYGD classification.

This data is used as the external category label in order to evaluate the accuracy of the clustering results. In other words, this classification data does not contain any gene expression data, but it is used in conjunction with expression data sets, supplying a label for the genes contained in the expression data sets. In fact, two classifications schemes were obtained from this data, the *FC* and the *REDUCED FC*.

The *FC* classification scheme is formed by thirteen first level classes of the MYGD, as in (Zhu & Zhang, 2000). These classes are expected to show similar expression profiles. Table 2 shows theses classes and the number of genes in each class.

The *REDUCED FC* (Table 3) is composed of five MYGD classes that have shown a high tendency to cluster together (Eisen *et al.*, 1998). Furthermore, genes belonging to these classes have been successfully used for building function prediction classifiers using supervised methods (Brown *et al.*, 2000).

CLASS	NUMBER OF GENES
Metabolism	1215
Energy	258
Cell Cycle and DNA Processing	815
Transcription	847
Protein Synthesis	363
Protein Fate	655
Cellular Transport	537
Cellular Communication	60
Cell Rescue, Defense and Virulence	287
Regulation of Cellular Environment	216
Transposable Elements, Viral, Plasmid Proteins	116
Control of Cellular Organisation	217
Transport Facilitation	363

Table 4.2: MYGD classes from the *FC* scheme with theirs respective number of genes.

CLASS	NUMBER OF GENES
Tricarboxylic acid cycle	17
Respiration	22
Cytoplasmic ribosome	121
Proteasome	35
Histones	11

Table 4.3: MYGD classes from the *REDUCED FC* scheme with theirs respective number of genes

4.2.2 Yeast All

This data set contains data from five yeast experiments, where 6200 ORFs had their expression profiles measured using cDNA microarrays. The ORF profiles contain 71 time points, observed during the following five biological processes: the mitotic cell division (cycle alpha, cdc15, elutration) (Spellman *et al.*, 1998), sporulation (Chu *et al.*, 1998) and diauxic shift (DeRisi *et al.*, 1997). These processes contained, respectively, 18, 25, 14, 7 and 7 time points. The expression value of each ORF in a time point is the *log* transformation (base 2) of the ratio between the measured expression level and the control expression level (Eisen *et al.*, 1998). Some of the genes contain missing values, either because insignificant hybridisation levels were detected, or because the genes were not measured in certain processes. (Data available at: http://genome-www.stanford.edu/clustering).

As stated in Section 3.2, the normalisation procedure requires the data vectors to contain only positive values, which is not the case of the log ratio values obtained in cDNA microarrays. In order to overcome this problem, the data sets applied to experiments that use normalisation are raised to the power of two, returning to the original measure-control ratio.

Two data sets were devised from the original *Yeast All* data set, the *FC Yeast All* and the *Reduced FC Yeast All*. The *FC Yeast All* data set contains only genes in the *FC* classification. A missing data filter was applied to this data set, excluding profiles with more than 20% of missing data. As in Heyer *et al.* (1999), a final filtering was employed in order to remove uninformative genes with low expression levels or with low variance between the time points. In these removed ORFs, the expression level did not vary across time, thus these profiles were considered uninformative in relation to gene function. In order to apply this filtering, genes were ranked according to their variance and mean, where the ones within the 25% lowest values (Heyer *et al.*,

1999) in each rank were removed. At the end, the *FC Yeast All* data set contained 1765 genes. The *Reduced FC Yeast All* data set contains only genes from the *Reduced FC* classification. Since there is a reduced number of genes in this data set, only the missing filter was applied, ending up with 205 genes.

4.2.3 Mitotic Cell Cycle (CDC 25)

This data set was obtained in an experiment from the Yeast organism during the mitotic cell division cycle (Cho *et al.*, 1998). The set contains the expression profiles measured with oligonucleotides arrays during 17 time points, with a similar set of ORFs as the one used in the *Yeast All* data set. In oligonucleotides arrays, there are 20 pairs of probes for each ORF. These pairs are composed of perfect match (PM) and mismatch (MM) probes, where the latter works as a specificity control. The expression of a gene is measured by the average of the difference of the PM and MM probes (Lipshutz *et al.*, 1999).

Two data sets were also devised from the *Mitotic Cell Cycle*, the *FC CDC 25* and the *Series CDC 25*. In the *FC CDC 25* dataset, only genes in the *FC* classification were considered. A variance filtering was employed in order to remove the 25% of the genes with lowest variance and mean. This data sets did not contained any missing data. The final number of genes in this data set was 1869. The *Series CDC 25* data set contains genes belonging to a visual classification of the series shape performed by Cho *et al.* (1998). In this classification, 420 genes were assigned to one of five known phases of the cell cycle (some of the genes were assigned to a multiple phase class). There was no need to pre-process this data set, as only informative gene profiles were included in the classification.

4.3 Experiments

The experiments are divided in two parts. In the first part, only the proximity indices are compared, while in the second one the comparison of the clustering methods is accomplished. The results obtained in the former are used to choose the proximity indices (with the best accuracy given a clustering method) to be used in the latter part. In the following two sections, both experiments are described. In the last section, implementation issues specific to each each clustering method are described.

4.3.1 Comparison of Proximity indices

The first part of the experiments compare versions of three proximity indices: angular separation (AS), Pearson correlation (PC) and Euclidean distance (ED). With respect to the Euclidean distance version, experiments are performed with the data vectors in three forms, namely, original (ED_1), normalised (ED_2) and standardised (ED_3) values. This yields five distinct settings of proximity indices and pre-processing. Each of these settings was implemented in the following clustering methods : CLICK, SOM, hierarchical clustering, dynamical clustering, k-means, and dynamical clustering and k-means with initialisation from the hierarchical method. The experiments were accomplished by presenting the four data sets (*FC Yeast All*, *Reduced FC Yeast All*, *FC CDC 25* and *Series CDC 25*) to all these methods and indices settings. More specifically, for each method, proximity index, and data set; a thirty-fold unsupervised cross-validation was applied. Afterwards, the mean values of the corrected Rand index (CR) for the test folds were measured. Finally, the mean of CR obtained by the five settings of proximity indices and pre-processing were compared two by two, using the bootstrap hypothesis test with 1000 bootstrap samples. As the interest of this experiment is in comparing the proximity indices, the hypothesis

tests only compared the results of experiments performed with the same clustering methods and data sets.

4.3.2 Comparison of Clustering Methods

The second part of the experiments compare the following clustering methods: CLICK, SOM, hierarchical clustering, dynamical clustering, k-means, and dynamical clustering and k-means with initialisation from the hierarchical clustering. Each clustering method was evaluated with the proximity index that obtained the higher accuracy in the first part of the experiments. The experiments were accomplished by presenting the same four data sets (*FC Yeast All*, *Reduced FC Yeast All*, *FC CDC 25* and *Series CDC 25*) to all methods. More specifically, for each method and data set; a thirty-fold unsupervised cross-validation was applied. Afterwards, the mean values of the corrected Rand index (CR) for the test folds were measured. Finally, the mean of CR obtained by the seven clustering methods were compared two by two, using the bootstrap hypothesis test with 1000 bootstrap samples. As the interest of this experiment is in comparing clustering methods, the hypothesis tests only compared the results of experiments developed with the same data sets.

In order to evaluate the usefulness of the validation methodology, a random assignment method was also included in this evaluation. This method simply assigns randomly the objects in the input data set to a cluster. Its is important to notice that this method is evaluated in the same manner as the other methods. In brief, the method is used to cluster the training sets in the k-fold cross-validation procedure. The nearest centroid procedure used to cluster the test set is then performed normally given the random partition. The only distinction of the evaluation of the random assignment method is that the final results are taken from the mean corrected Rand values obtained in 100 different runs. These mean results obtained by the random

assignment method are taken as the worst case. All other clustering methods should obtain values significantly higher than it.

4.3.3 Clustering Method Implementations

In order to perform the experiments with dynamical clustering and $k-$means methods, an implementation from (Costa *et al.*, 2002a) was used. In terms of the parameters of these two methods, the number of clusters was set to the number of *a priori* classes (the number of clusters was also set to the number of *a priori* classes in the other methods), and the number of distinct initialisations used was 100.

In relation to the CLICK method, an implementation available in the software Expander was utilised. (Expander available at: http://www.cs.tau.ac.il/∼rshamir/expander/expander.html). The implementation did not support the Euclidean distance version, so only the Pearson correlation and angular separation versions are compared with this method. Missing data was not supported as well, so only the *CDC 25* data sets were used in the CLICK experiments. The homogeneity, the other algorithm parameter, was set to its default value.

The SOM Toolbox for Matlab was used to run the SOM experiments (SOM Toolbox available at: http// www.cis.hut.fi/projects/somtoolbox.). The original implementation only supported the Euclidean distance. Thus, in order to include Pearson correlation and angular separation, modifications were done in the code. SOM requires parametrisation experiments, in order to tune its performance (see Section 3.1.4). Due to the number of parameters available, and the complexity of choosing them, only the topology will be varied. This choice is based in a previous study with gene expression data, where its was found that the topology was the parameter with highest impact on the results (Jonsson, 2001).

In order to set the other parameters from SOM, a method of the toolbox that uses a number of heuristics to set the parameters was employed (for more details see the description of the method *som_make* in Vesanto *et al.* (2000)). As not all the results obtained by this parametrisation were satisfactory, another parametrisation based on the one used in Vesanto & Alhoniemi (2000) was employed (this parametrisation is refereed as *VESANTO*, while the former is refereed as *DEFAULT*). The *VESANTO* parametrisation used 10 epochs and a learning rate of 0.5 during the ordering phase. The initial radius was set to the topology highest dimension and the final radius to half the highest dimension. In the convergence phase, 10 epochs and a learning rate of 0.05 were used. The initial radius was set to half the highest topology dimension minus 1 and the final radius to 1. In both phases, the neighbourhood function was the Gaussian. In relation to the topology, the following procedure was applied. An initial topology is chosen. Additionally, experiments with a larger and smaller topology are also performed. If the initial topology obtain the best results then no more experiments are done. Otherwise, the same process is repeated for the topology with best result.

R software was used with the hierarchical clustering experiments (software available at: http://www.r-project.org/). Only experiments with the average linkage method were performed, since this method has been extensively used in the gene expression literature (Eisen *et al.*, 1998). As the external index used in this work is suitable only for partition comparison, the resulting hierarchies were cut in a given level in order to provide partitions (see Section 3.1.5). In the experiments with gene expression data, sub-trees with less than 5 objects were ignored.

Chapter 5

Results

The results of the comparative analysis are presented and analysed in this chapter. Section 5.1 describes the results achieved in the comparative analysis of the proximity indices, while Section 5.2 describes the results achieved in the comparative analysis of the clustering methods. The results of the experiments for selection for parameters for SOM are reported in Appendix A, while Appendix B presents detailed statistics of all experimental results.

5.1 Proximity Indices

5.1.1 Experiments

Figure 5.1 shows the mean values of corrected Rand for the experiments performed with the *FC Yeast All* data set (the higher the corrected Rand, the higher the accuracy). Regarding the experiments with SOM, ED_3 and PC obtained higher values than the other proximity indices. In these cases, the hypotheses of no difference (null hypothesis) were rejected in favour of ED_3 and PC (at significance level α of 0.01).

In respect to the hierarchical clustering, ED_1 and ED_2 achieved lower values than the other proximity indices. In these cases, the hypotheses of no difference were rejected in favour of ED_3, AS and PC at α of 0.01. For all other methods, except for the SOM and hierarchical clustering, AS obtained a higher accuracy than the other proximity indices. In these cases, the hypotheses of no difference between AS and the other proximity indices were rejected in favour of AS at $\alpha = 0.05$. In fact, with all clustering methods, except for SOM and hierarchical clustering, the hypotheses of no difference between ED_1 and AS were rejected in favour of AS at even a lower significance level ($\alpha = 0.01$).

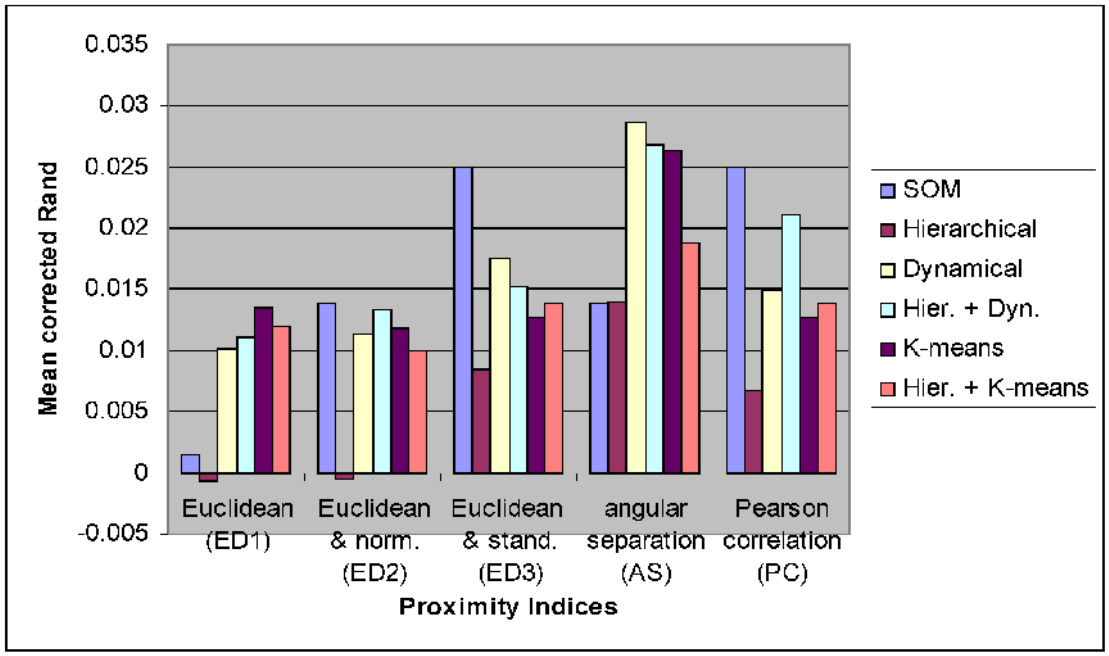

Figure 5.1: Mean of corrected Rand values from the *FC Yeast All* experiments

In Figure 5.2, the mean values of corrected Rand with the *Reduced FC Yeast All* data set are illustrated. In the experiments with SOM, ED_3 obtained a higher accuracy than the other proximity indices. In these cases, the null hypotheses were rejected in favour of ED_3 in comparison to ED_1 ($\alpha = 0.01$), AS and PC ($\alpha = 0.05$). In terms of hierarchical clustering, ED_1 and ED_2 achieved lower values than the

other proximity indices. In these cases, the hypotheses of no difference were rejected in favour of ED_3, AS and PC at $\alpha = 0.01$. Furthermore, still in the hierarchical clustering, PC obtained the higher accuracy than the other proximity indices. The null hypotheses were rejected in favour of PC at $\alpha = 0.05$. For the dynamical clustering and k-means, both with or without hierarchical initialisations, AS achieved a lower accuracy in comparison to the other proximity indices. For these four experiments, the hypotheses of no difference between AS and all other proximity indices were reject in favour to ED_1, ED_2, ED_3 and PC at $\alpha = 0.02$. Furthermore, still with these four methods, ED_1 obtained an accuracy as high as ED_2, ED_3 and PC, with no significant difference detected among these proximity indices.

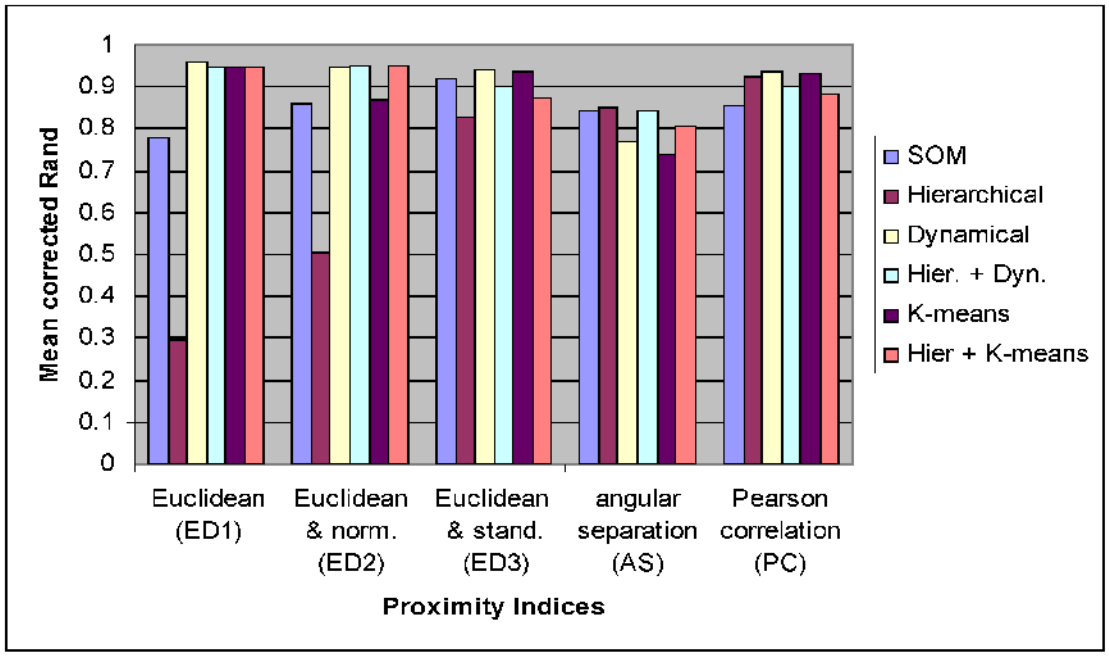

Figure 5.2: Mean of corrected Rand values from the *Reduced FC Yeast All* experiments

The results for the *FC CDC 25* data set are summarised in Figure 5.3. In the dynamical clustering (with or without hierarchical initialisation), ED_2 and AS achieved a higher accuracy than ED_1. In both cases, the null hypotheses were rejected in favour

of ED_2 and AS at $\alpha = 0.02$. In the results with k-means (with or without hierarchical initialization), ED_2 had a higher accuracy in comparison to ED_1. In this case, the null hypothesis was rejected in favour of ED_2 at $\alpha = 0.02$. In terms of CLICK, SOM and hierarchical clustering, no significant difference was detected among the results.

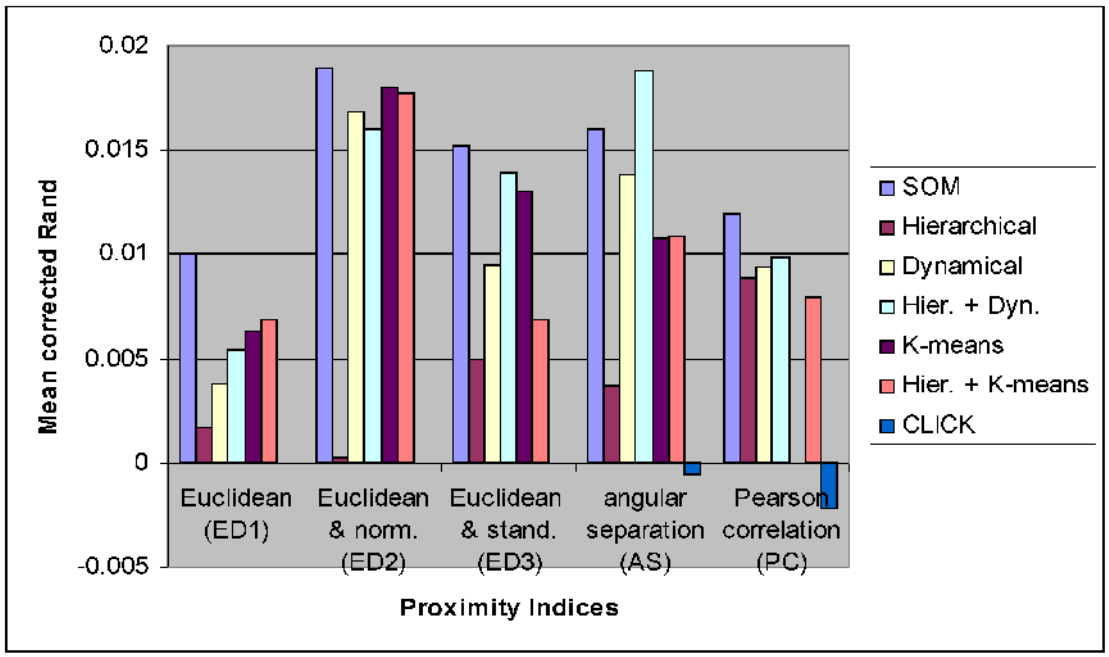

Figure 5.3: Mean of corrected Rand values from the *FC CDC 25* experiments

Figure 5.4 shows the mean values of corrected Rand with the *Series CDC 25* data set. ED_2, ED_3, AS and PC achieved a higher accuracy in comparison to ED_1 in all clustering methods, except for CLICK. The hypotheses of no difference between ED_1 and the other proximity indices were rejected in favour of ED_2, ED_3, AS and PC at $\alpha = 0.01$. The accuracy of ED_3, AS and PC were also higher than ED_2 in the dynamical and k-means methods (with or without hierarchial initialisations). In these cases, the null hypotheses were rejected in favour of ED_3, AS and PC at $\alpha = 0.02$. In respect to the experiments with CLICK, the PC obtained values significantly higher than AS (null hypothesis rejected at $\alpha = 0.01$).

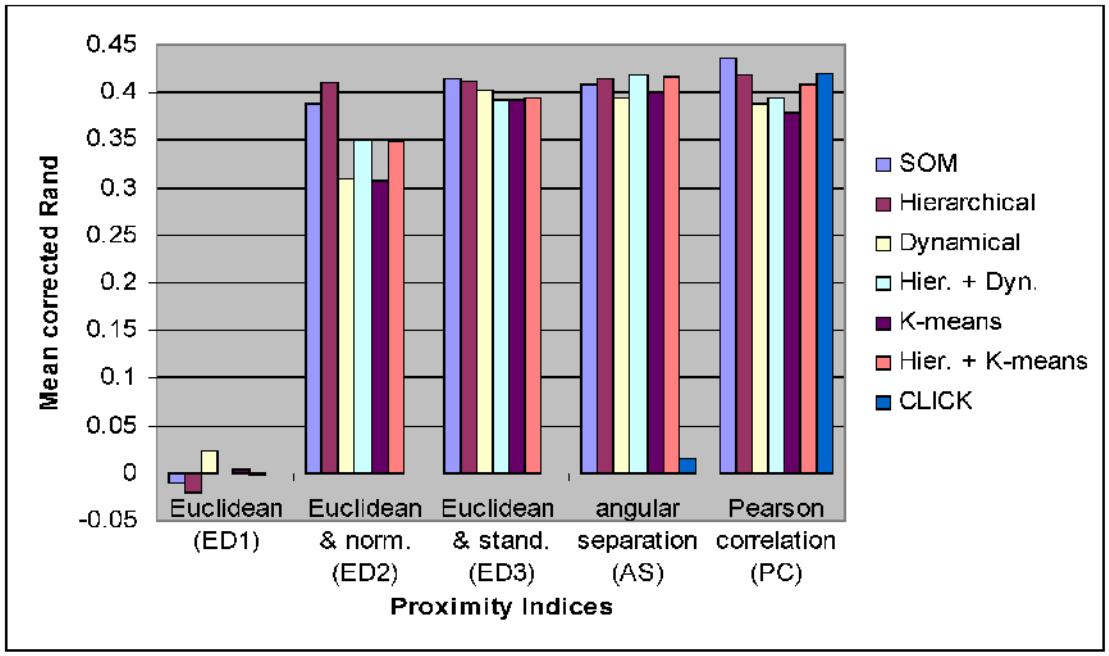

Figure 5.4: Mean of corrected Rand values from the *Series CDC 25* experiments

5.1.2 Discussions

ED_1 lead to the lowest accuracies in all but in the *Reduced FC YeastAll* data set. These results were already expected, due to the fact that this proximity index is not suitable for capturing relative magnitude dissimilarity (or shape dissimilarity). In the *Series CDC 25* data set, which is the only data set with the classification directly related to the series shape, the difference of ED_1 and the other proximity indices, which capture relative magnitude, was even larger.

With respect to the *Reduced FC YeastAll* data set, ED_1 had values as high as other proximity indices, but not showing a significant advantage over them. This is actually a rather interesting result that shows that the data set, by having a reduced set of genes, has distinct characteristics of data sets with a higher number of genes such as the *FC Yeast All* data set. Recalling Section 4.2.1, the *Reduced FC* classification was devised from the results achieved in Eisen *et al.* (1998). In other words, this

reduced set of classes were the ones more easily classified in previous studies, so one could argue that these genes profiles are so well separated that even ED_1 is capable of discriminating them. Furthermore, it also can be said that this data set is biased. In the experiments carried out in Eisen *et al.* (1998), the hierarchical clustering was used with the Pearson correlation to cluster the results. Not surprisingly, in the hierarchical clustering experiments, the Pearson correlation got a higher accuracy than the other proximity indices.

In relation to the proximity indices capable of measuring shape similarity (ED_2, ED_3, AS and PC), no proximity index can be stated to be superior to the others. In the *FC Yeast All* data set, AS obtained significant higher values than the others. On the other hand, in the *FC CDC 25* data set, ED_2 obtained the highest values. ED_2, ED_3 and PC achieved the highest values in the *Reduced FC Yeast* data set, while ED_3, AS and PC had the highest values in the *Series CDC 25* data set. One possible reason for these contrasting results is that the data sets were captured with different microarray technologies and use different expression level representations. As mentioned in Section 4.2, the elements in the *Yeast All* data set was captured with cDNA microarrays, where the expression levels were log ratios between the measure and control expression levels. In contrast, the elements in the *CDC 25* data set were captured with oligonucleotide arrays, where the expression values represent the mean of the absolute values from twenty distinct probes.

5.2 Clustering Methods

5.2.1 Experiments

The results of Section 5.1 were used to select the proximity indices for the comparative analysis of the clustering methods. Indeed, only the proximity indices with best accuracy for a given clustering method and data set were selected. Table 5.1 shows these proximity indices.

	FC Yeast All	*Red. FC Yeast All*	*FC CDC 25*	*Series CDC 25*
SOM	PC	PC	ED_2	PC
hierarchical	AS	PC	PC	PC
dymanical	AS	ED_1	ED_2	ED_3
k-means	AS	ED_1	ED_2	AS
hier. + dym.	AS	ED_2	AS	AS
hier. + k-me.	AS	ED_2	ED_2	AS
CLICK	-	-	AS	PC

Table 5.1: Proximity indices with best accuracy in the experiments of Section 5.1 for a given clustering method and data set.

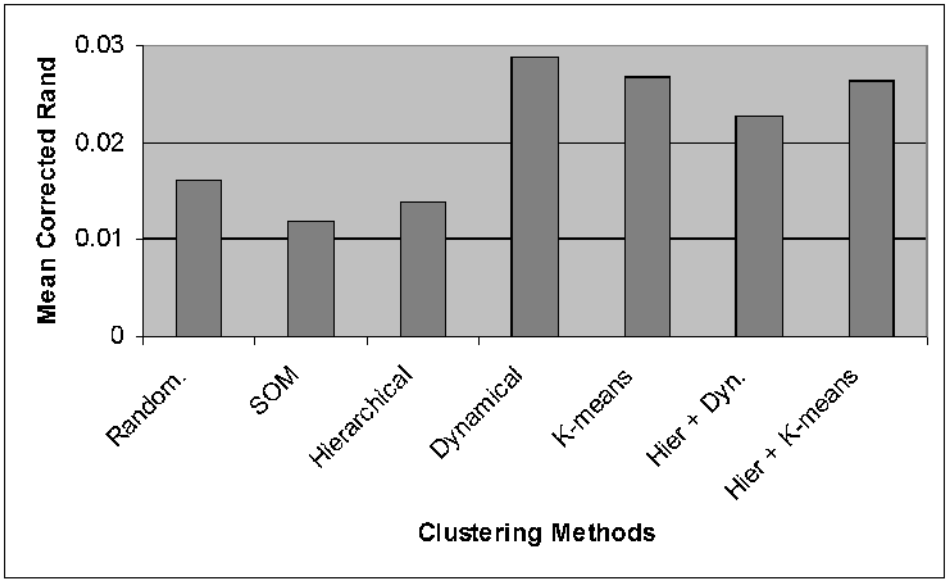

Figure 5.5: Mean of corrected Rand values from the *FC Yeast All* experiments

In Figure 5.5, the mean values of corrected Rand for the experiments with the *FC Yeast All* data set are shown. The dynamical clustering obtained a higher accuracy than the other clustering methods. The null hypotheses were rejected in favour to the dynamical clustering in comparison to random assignment and hierarchical clustering at $\alpha = 0.01$. SOM and k-means also achieved a significant higher accuracy than random assignment and hierarchical clustering. In these cases, the null hypotheses were rejected in favour to k-means and SOM in comparison to random assignment ($\alpha = 0.02$) and hierarchical clustering ($\alpha = 0.05$). Dynamical clustering and k-means both with hierarchical initialization also achieved a significant higher accuracy than random assignment and hierarchical clustering. In these cases, the null hypotheses were rejected in favour to dynamical clustering and k-means in comparison to random assignment ($\alpha = 0.05$) and hierarchical clustering ($\alpha = 0.05$).

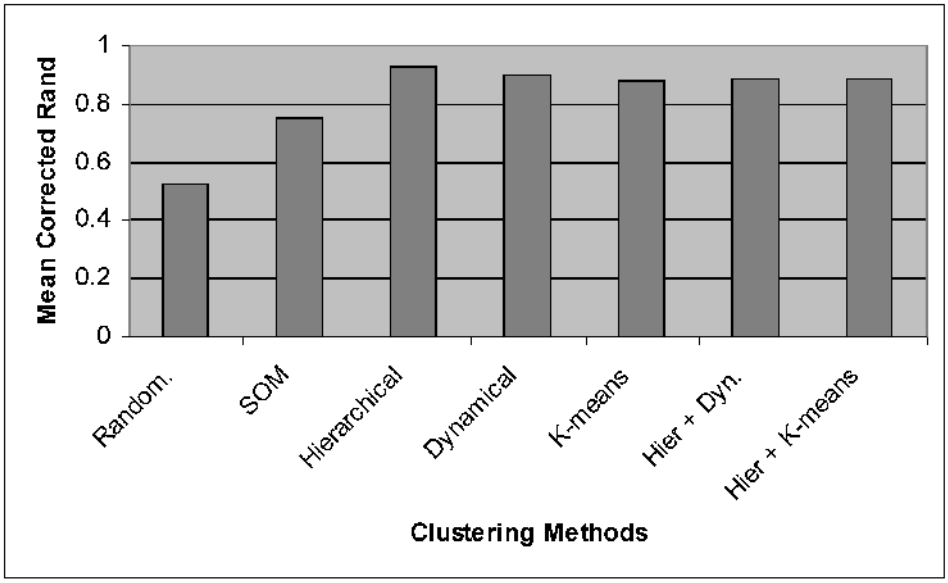

Figure 5.6: Mean of corrected Rand values from the *Reduced FC Yeast All* experiments

The mean values of corrected Rand for the experiments with the *Reduced FC Yeast-all* data set are presented in Figure 5.6. The random assignment method obtained the lowest accuracy in comparison to all other methods. The null hypotheses were rejected

in favour to SOM, hierarchical clustering, dynamical clustering and k-means (with or without hierarchical initialization) in relation to the random assignment method at a $\alpha = 0.01$. No other significant difference were detected among the methods.

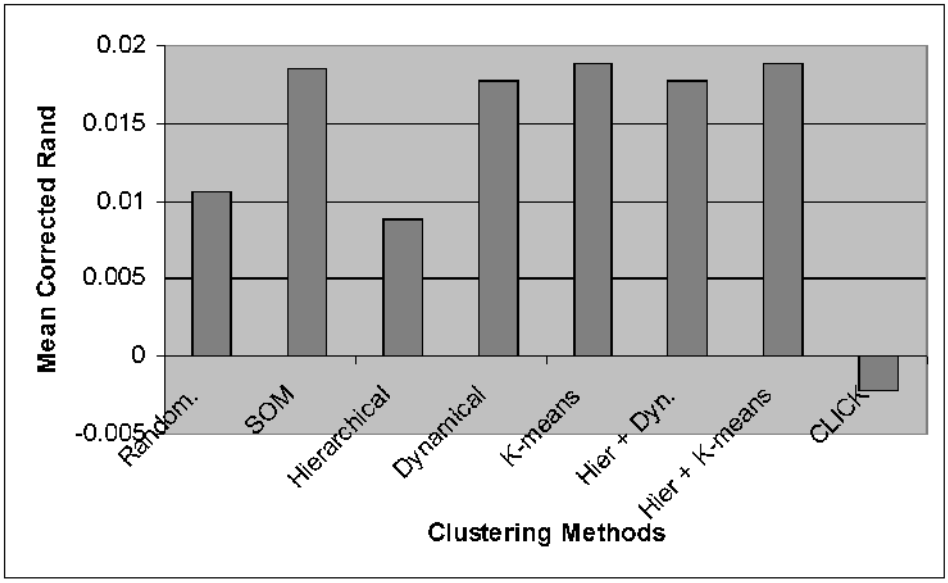

Figure 5.7: Mean of corrected Rand values from the *FC CDC 25* experiments

Figure 5.7 ilustrastes the mean values of corrected Rand of the experiments with the *FC CDC 25* data set. The CLICK method obtained a lower result than all others methods, including the random assignment. In these cases, the null hypotheses were rejected in favour to all other methods at $\alpha = 0.01$. k-means (with or without hierarchical initialization) and SOM obtained significant higher accuracy than random assignment and hierarchical clustering. The null hypotheses were rejected in favour to SOM and k-means at a $\alpha = 0.01$. Dynamical clustering (with or without hierarchical initialization) also obtained significant higher accuracy than random assignment and hierarchical clustering. The null hypotheses were rejected in favour to dynamical clustering at a $\alpha = 0.5$.

Figure 5.8 shows the mean values of corrected Rand for the experiments performed with the *Series CDC 25* data set. The random assignment method obtained the lowest

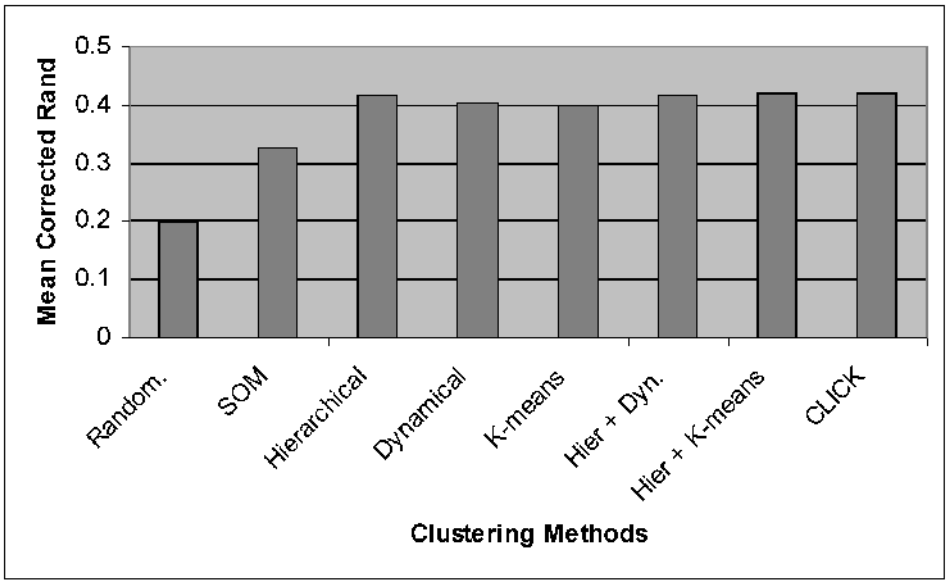

Figure 5.8: Mean of corrected Rand values from the *Series CDC 25* experiments

results in comparison to all other methods. In these experiments, the null hypotheses were rejected in favour to SOM, hierarchical clustering, CLICK, dynamical clustering and k-means (with or without hierarchical initialization) at a $\alpha = 0.01$. No other significant difference were detected among the methods.

5.2.2 Discussions

In terms of hierarchical clustering, low accuracies were achieved in experiments with the *FC CDC 25* and *FC Yeastall 25* data sets. This was not the case of the two other data sets (*Reduced FC Yeastall* and *Series CDC 25*), as the hierarchical clustering got accuracies as high as other methods. It can be concluded that hierarchical clustering has some problems in clustering larger data sets formed by the complete *Functional Classification (FC)* scheme. The clusters in the data sets based on the *FC* scheme are not so compact and isolated, as the ones with the *Reduced FC* and the series shape classification. The *FC* data sets have a higher number of genes and their classification

were not devised from gene expression analysis. Given the lack of robustness of the hierarchical clustering methods to outliers and noisy data (see Section 3.1.5), the low accuracies in the *FC* data sets are expected. These results are also compatible to other comparative analysis of clustering methods for gene expression . In Datta & Datta (2003), the average hierarchical clustering also obtained worse results than other clustering methods, such as k-means and model-based methods. The hierarchical methods also showed a low stability in the experiments presented in Costa *et al.* (2002b).

Some comments about the results of the CLICK method also should be made. In the *Series CDC 25* experiments, CLICK achieved the highest mean corrected Rand in relation to all other methods. On the other hand, CLICK obtained negative values in the *FC CDC 25* data set. As mentioned before, the CLICK method encounters the number of clusters automatically. This task was perfectly performed in the *Series CDC 25*, where 6 clusters were encountered in most of the experiments. This was not the case in the *FC CDC 25* experiments, where the number of clusters varied around 20 and 26 with the *PC*; and around 5 to 7 with the *AS*. These results suggest that CLICK showed instability in clustering the *FC CDC 25* gene expression data set. It can be the case that CLICK presented similar problems as the hierarchical clustering, however, only one data set with the complete *Functional Classification* was used in the experiments. Further experiments are necessary to investigate this issue properly.

As a whole, k-means, dynamical clustering (both with or without hierarchical initialization) and SOM obtained high accuracies in all experiments. The use of the hierarchical initialization does not affect the accuracy of k-means and dynamical clustering, even if the hierarchical method alone does not achieve a good accuracy. In fact, the hierarchical initialization reduces the run time of both dynamical clustering and k-means experiments, as there is no need of several random initializations (see

Section 3.1.2). SOM has one main disadvantage in relation to k-means and dynamical clustering. It required more complex experiments for selecting the parameters. On the other hand, SOM returns a topological map, where the clusters have neighboorhood relations. Such structure is much more informative than simple partitions returned by k-means and dynamical clustering. Furthermore, in the experiments performed in this dissertation, the number of clusters was already known. However, in a problem where this number is unknown, the use of k-means and dynamical clustering also requires "parametrisation" experiments for finding the ideal number of clusters.

The results of the experiments with the *Reduced FC Yeast All* data set reinforce the suggestions made in Section 5.1.2 that there is some bias in this data set. Even though, k-means and dynamical clustering (with or without hierarchical clustering initialization) achieved good results, the hierarchical clustering had the largest accuracy. Again, this is not a coincidence, as the functional classes presented in this classification scheme were the ones more easily clustered in experiments done with the hierarchical clustering.

Regarding the utilisation of gene annotation as an *a priori* classification, in both *FC Yeast all* and *FC CDC 25* data sets, where the complete functional classification was used, a low agreement with the clustering results was encountered. In these experiments, the mean values of corrected Rand were below 0.05, which indicate clustering solutions found by chance (Milligan & Cooper, 1986). A previous study (Gertein & Janssen, 2000), using similar data sets, had already indicated that the functional classification has only a weak relation to the clustering of gene expression profiles. The reasons for this are, among others, the vague definitions of some functions and the great overlap of the classes (Gertein & Janssen, 2000). These weak relations were also encountered in the context of supervised methods for building functional classifiers based on the gene expression profiles (Kuramochi & Karypis, 2001). Nevertheless, in

the context of this work, the previous issues do not represent a problem, since this work is concerned only with the comparison of the clustering methods (or proximity metrics), and not with the evaluation of the quality of the clusters generated.

In relation to the validation methodology, the results obtained by the random assignment method demonstrated the usefulness of the validation methodology employed in this work, and as a consequence, the validity of the results encountered. As expected, the random assignment method showed the lowest accuracy (or accuracies as low as other methods) in all experiments.

Chapter 6

Conclusions

The main contribution of this dissertation was to present a comparative analysis of clustering methods and proximity indices applied to the analysis of gene expression time series. In order to do so, a validation methodology based on the k-fold cross-validation procedure and the use of gene annotation was proposed. The study carried out in this dissertation is more complete than previous ones, as it used more data sets, and included methods not evaluated before, such as SOM and dynamical clustering. Furthermore, no comparative analysis of proximity indices has been performed before.

In the comparative analysis of the proximity indices, the results did not indicated the superiority of one particular index over the others. In three out of four data sets, the Euclidean distance version with original data obtained the worst results. This was already expected, since that proximity index does not capture the relative magnitude proximity. With respect to proximity indices that capture relative magnitude, in the *FC Yeast All* data set, the angular separation version achieved the highest values; while in the *FC CDC 25* data set, the Euclidean distance version with normalisation achieved the highest values. In the *Series CDC 25* data set, where series shape was directly taken into consideration in the classification labels, all relative magnitude

proximity indices achieved high values. From these results, no relative magnitude proximity index can be stated to be superior to the others.

In relation to the comparison of the clustering methods, the methods presented distinct performances. The hierarchical clustering and CLICK obtained low accuracies in the data sets with the complete *Functional Classification*. On the other hand, SOM, dynamical clustering and k-means had the best accuracies in all experiments. Furthermore, the use of the hierarchical method as initialisation to dynamical clustering and k-means resulted in a substantial reduction of run time, with no lost in the accuracy.

The comparative analysis carried out in this dissertation only compared the accuracy of the clustering methods. However, it is important to point out that other characteristics should be taken into consideration in the choice of a clustering method. For example, one should also consider the type of output of the clustering method. SOM, for instance, gives a topological map as result, a structure more informative than the partitions provided by CLICK, dynamical clustering and k-means. Another example, some methods, such as CLICK, do not require the number of clusters to be set, which is not the case of k-means and dynamical clustering. This characteristic is very important when the number of clusters in the data set is unknown.

Another contribution of this work is the proposed validation methodology. There is no report of the use of the unsupervised k-fold cross-validation procedure as approached in this work. This methodology has as advantage to others, a lower computational cost. The methodology showed consistent results, specially, with the random assignment method. Such a method obtained the lowest results (or results as low as other methods) in all data sets. Furthermore, the use of functional classification as a external classification proved to be valid in the context of a comparative analysis, despite the low relation of the gene expression data with functional classification.

The analysis of the absolute accuracy obtained in the data sets with the complete *Functional Classification* is another contribution of this work. The results reinforce the findings of previous works (Gertein & Janssen, 2000; Kuramochi & Karypis, 2001), where it was found that the functional classification of the genes has only a weak relation to gene expression data.

6.1 Future Work

The number of public data sets of gene expression time series with an external classification is undesirable low. The use of new data sets in the future is vital to answering some of the questions raised in this work. One of the questions is to investigate if a particular proximity index is more suitable for data captured with a particular type of *microarray* technology. Another issue to be further evaluated is the poor results obtained with the CLICK method in the experiments with the *FC CDC 25* data sets. It should be investigated if the poor results are related with the use of the complete *Functional Classification*.

Other types of biological information have already been used as external categories. Such data can be used in the proposed validation methodology as a complement to the use of functional classification. Among these sources there are: regulatory regions, protein structure, and metabolic pathways (Gertein & Janssen, 2000; Zhu & Zhang, 2000).

This analysis can also be enhanced with the inclusion of new clustering methods. In special, methods with good results in other comparative analysis (Datta & Datta, 2003) and of model-based clustering methods, which are now extensively applied to analysis of gene expression time series (Schliep *et al.*, 2003).

In relation to the validation methodology, one future work is to perform a detailed evaluation of the unsupervised k-fold cross-validation procedure. Such evaluation should carry out Monte Carlo experiments with the generation of artificial data sets, so as to evaluate the characteristics of this methodology.

Bibliography

Abbott, A. (1999), A post-genomic challenge: learning to read patterns of protein synthesis, *Nature*, 402:715-720.

Azuaje, F. (2002), A cluster validity framework for genome expression data, *Bioinformatics*, 18(2):319-320.

Bertone, P., Gerstein, M. (2001), Integrative data mining: the new direction in bioinformatics, *IEEE Engineering in Medicine and Biology*, 20:33-40.

Bowtell, D. D. (1999), Options available–from start to finish–for obtaining expression data by microarray, *Nature Genetics*, 21(1S):25-32.

Bo, T., Jonassen, I. (2002), New feature subset selection procedures for classification of expression profiles, *Genome Biology*, 3(4):research0017.1-0017.11.

Breckenridge J. N. (1989), Replication cluster analysis: Method, consistency, and validity, *Multivariate Behavior Research*, 24(2):147-161.

Breckenridge J. N. (2000), Validating cluster analysis: Consistent replication and symmetry, *Multivariate Behavior Research*, 35(2):261-285.

Brown, M. P., Grundy, W. N., Lin, D., Cristianini, N., Sugnet, C. W., Furey, T. S., Ares, M. JR, Haussler, D. (2000), Knowledge-based analysis of microarray gene expression data by using support vector machines, *Proc. of National Academy of Sciences USE*, 97:262-267.

Brown, M. P., Bostein, D. (1999), Exploring the new world of genome with DNA microarrays, *Nature Genetics*, 21:33-37.

Bustin, S. A. (2000), Absolute quantification of mRNA using real-time reverse transcription polymerase chain reaction assays, *Journal of Molecular Endocrinology*, 25:169-193.

Bustin, S. A. (2002), Quantification of mRNA using real-time reverse transcription PCR (RT-PCR): trends and problems, *Journal of Molecular Endocrinology*, 29:23-39.

Cho, R., Campbell, M., Winzeler, E., Steinmetz, L., Conway, A., Wodicka, L., Wolfsberg, T., Gabrielian, A., Landsman, D., Lockhart, J., Davis, W. (1998), A genome-wide transcriptional analysis of the mitotic cell cycle, *Molecular Cell*, 2:65-73.

Chu, S., Del Risi, J., Eisen, M., Mulholland, J., Botstein, D., Brown, P.O., Herskowitz I. (1998), The Transcriptional Program of Sporulation in Budding Yeast, *Science*, 282:699-705.

Costa, I. G., de Carvalho, F. A. T., de Souto M. C. P. (2002a), A Symbolic Approach to Gene Expression Time Series Analysis, *Proc. of the VII Brazilian Symposium on Neural Networks*, IEEE Computer Society, 1:24-30.

Costa, I. G., de Carvalho, F. A. T., de Souto M. C. P. (2002b), Stability Evaluation of Clustering Algorithms for Time Series Gene Expression Data, *Proc. of I Brazilian Workshop on Bioinformatics*, 2002, 88-90.

Costa, I. G., de Carvalho, F. A. T., de Souto M. C. P. (2003), Comparative Study on Proximity Indices for Cluster Analysis of Gene Expression Time Series, *Journal of Intelligent and Fuzzy Systems*, Accepted.

Datta S., Datta, S. (2003), Comparisons and validation of statistical clustering techniques for microarray gene expression data, *Bioinformatics*, 19:459-466.

DeRisi, J. L., Iyer V. R., Brown P. O. (1997), Exploring the metabolic and genetic control of gene expression on a genomic scale, *Science*, 278:680-686.

D'Haeseleer, P., Liang, S., Somogyi, R. (1999), Gene Expression Data Analysis and Modeling, *Proc. of Pacific Symposium on Biocomputing*.

Diday, E., Simon, J.C (1980), Clustering Analysis, *Digital Pattern Recognition,*, Springer-Verlag, New York, 47-92.

Dopazo, J., Zanders, E., Dragoni, I., Amplett, G., Falciani, F. (2001), Methods and aproaches in the analysis of gene expression data, *Journal of Immunological Methods*, 250:93-112.

Dougherty, E. R., Chen, Y., Batman, S., Bittner, M. L.(1997) Digital measurement of gene expression in a cDNA microarray, Proceedings of SPIE, 3034:68-73.

Dubes, R. (1987), How many clusters are best? An experiment, *Pattern Recognition*, 20(6):645-663.

Dubes, R. (1998), Cluster Analysis and Related Issues, *In Handbook of pattern recognition & computer vision*, Word Scientific Publishing, Second Edition, 3-32.

Duggan D. J., Bittner M., Chen Y., Meltzer P., Trent J. (1999), Expression profiling using cDNA microarrays, *Nature Genetics*, 21:10-14.

Efron B., Tibshirani, R. (1993), *An Introduction to the Boostrap*, Chapman & Hall, New York.

Eisen, M. B., Spellman, P. T., Brown, P. O., Botstein, D. (1998), Cluster analysis and display of genome-wide expression patterns, *Proc. of National Academy of Sciences USA*, 95:14863-14868.

Felsenstein, J. (1985), Confidence limits on phylogenies: an approach using the bootstrap, *Evolution*, 39:783-791

Fuhrman, S., Cunningham, M.J., Wen, X., Zweiger, G., Seilhamer, J.J., Somogyi, R. (2000), The application of shannon entropy in the identification of putative drug targets, *Biosystems*, 55:5-14.

GENTROP, Departamento de Genética, Universidade Federal de Pernambuco (1999), *Genética Molecular: Fundamentos e Aplicações*, http://www.biolmol.hpg.ig.com.br/index.htm.

Gerstein, M., Janssen, R. (2000), The current excitement in bioinformatics - analysis of whole genome expression data: how does it relates to protein structure and function?, *Current Opinion on Structural Biology*, 10(5):574-84.

Golub, T. R., Slonim, D. K, Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J. P., Coller, H., Loh, M. L., Downing, J. R., Caligiuri, M. A. (1999), Class Prediction and Discovery Using Gene Expression Data, *Science*, 286:531-537.

Gordon A. D. (1996), Null models in cluster validation, *From Data to Knowledge: Theoretical and Practical Aspects of Classification, Data Analysis and Knowledge Organization*, Springer-Verlag, Berlin, 32-44.

Gordon A. D. (1999), *Classification*, Chapman & Hall, New York.

Gower, J. (1971), A general coefficient of similarity and some properties, *Biometrics*, 27:857-874.

Haykin, S. (1994), *Neural Networks: A Comprehensive Foundation*, Macmillan/IEEE Press, New Jersey.

Heyer, L. J., Kruglyak, S., Yooseph, S. (1999), Exploring expression data: identification and analysis of coexpressed genes, *Genome Research*, 9(11):1106-1115.

Hubbert, L. J., Arabie, P. (1985), Comparing partitions, *Journal of Classification*, 2:63-76.

Jain A. K., Dubes, R. C. (1988), *Algorithms for clustering data*, Prentice Hall, New Jersey.

Jain A. K., Murty, M. N., Flynn, P. J. (1999), Data Clustering: a review. *ACM Computing Surveys*, 31(3).

Jain A. K., Moreau, J. V. (1987), Bootstrap techniques in cluster analysis, *Pattern Recognition*, 20:547-568.

Jonsson, P. (2000), *Improving Clustering of Gene Expression Patterns*, Master Dissertation, Department of Computer Science, University of Sködve.

Kain, K. K. (2001), Biochips for Gene Spotting, *Science*, 294:621-625.

Kohonen, T. (1997), *Self-Organizing Maps*, Springer-Verlag, Berlin.

Kuramochi M., Karypis, G. (2001), Gene Classification Using Expression Profiles: A Feasibility Study, *Proc. of the 2nd IEEE International Symposium on Bioinformatics and Bioengineering.*

Lipshutz, R., Fodor, S., Gingeras, T., Lockhart, D. (1999), High density oligonucleotide arrays. *Nature Genetics*, 21:20-24.

Lubocac, Z. (2000), *Evaluation of clustering of gene expression data*, Master Dissertation, Department of Computer Science, University of Sködve.

Lubovac, Z., Olsson, B., Jonsson, P., Laurio, K., Andersson, M. L. (2001), Biological and statistical evaluation of clustering of gene expression profiles, *Proc. of Mathematics and Computers in Biology and Chemistry*, WSES Press, 149-155.

Mangiameli, P. Chen, S. K., West, D. (1996), A comparison of SOM neural network and hierarchical clustering methods, *European Journal of Operational Research*, 93:402-417.

McIntyre, R. M., Blashfield R. K. (1980), A nearest-centroid technique for evaluating the minimum-variance clustering procedure, *Multivariate Behavorial Research*, 15:225-238.

Mewes H. W., Frishman D., Güldener U., Mannhaupt G., Mayer K., Mokrejs M., Morgenstern B., Münsterkoetter M., Rudd S., Weil B. (2002), MIPS: a database for genomes and protein sequences, *Nucleic Acids Research*, 1:30(1):31-4.

Milligan, G. W. (1996), Clustering Validation: results and implications for applied analyses, *Clustering and Classification* , World Scientific, Singapore, 341-375.

Milligan G. W., Cooper M. C. (1986), A study of the comparability of external criteria for hierarchical cluster analysis, *Multivariate Behavorial Research*, 21:441-458.

Mitchell, T. (1997), *Machine Learning*, McGraw Hill, New York.

Morey, L. C., Blashfield, R. K. and Skinner, H. A. (1983). A comparison of cluster analysis techniques within a sequential validation frame work, *Multivariate Behavioral Research*, 18:309-329.

U.S. Department of Energy. DOE human genome program (1992), *Primer on molecular genetics*, Washington, D.C.

Quackenbush, J. (2001), Computational Analysis of Microarray Data, *Nature Genetics Reviews*, 2:418-427.

Raychaudhuri, S., Sutphin, P. D., Chang, J. T., Altman, R. B. (2001), Basic microarray analysis: grouping and feature reduction, *Trends in Biotechnology*, 19(5):189-193.

Riley M. (1998), Genes and proteins of Escherichia coli K-12, *Nucleic Acids Research*, 26:54.

Schena, M., Shalon, D., Davis, R. W., Brown, P. O. (1995), Quantitative monitoring of gene expression patterns with a complementary DNA microarray, *Science*, 270:467-470.

Schliep A., Schoenhuth A., Steinhoff C. (2003), Using Hidden Markov Models to Analyze Gene Expression Time Course Data, *Proc. of the International Conference on Intelligent Systems for Molecular Biology SupplementtoBioinformatics*, Accepted.

Schuchhardt, J., Beule, D., Malik, A., Wolski, E., Eickhoff, H., Lehrach, H. and Herzel, H. (2000), Normalization strategies for cdna microarrays, *Nucleic Acids Res*, 28, E47.

Shamir R. Shtainhart A., Torok D. (2002), *Introductory Concepts: Analysis of Gene Expression Data, DNA Chips and Gene Networks*, (Lecture Notes).

Sharan, R. Shamir, R. (2002), CLICK: A Clustering Algorithm with Applications to Gene Expression Analysis. *Proc. of Intelligent Systems for Molecular Biology SupplementtoBioinformatics*, 307-316.

Sherlock, G. (2000), Analysis of large-scale gene expression data, *Current Opinion in Immunology*, 12(2):201-205.

Silva F. H. (2001), Biologia Molecular, *I Escola de Inteligência Artificial e Bioinformática*, São Carlos, 2001.

Spellman, P. T., Sherlock, G., Zhang, M. Q., Iyer, V. R., Anders, K., Eisen, M. B., Brown, P. O., Botstein, D., Futcher,B. (1998), Comprehensive Identification of

Cell Cycle-regulated Genes of the Yeast Saccharomyces cerevisiae by Microarray Hibridization, *Molecular Biology of the Cell*, 9:3273-3297.

Tamayo, P., Slonim, D., Mesirov, J., Zhu, Q., Kitareewan, S., Dmitrovsky, E., Ler, E. S., Golub, T. R. (1999), Interpreting patterns of gene expression with self-organizing maps: methods & application to hematopoietic differentiation, *Proc. National Academy of Sciences USA*, 16:96(6):2907-2912.

Tavazoie, S., Hughes, J. D., Campbell, M. J., Cho, R. J., Church, G. M. (1999), Systematic determination of genetic network architecture, *Nature Genetics*, 22:281-285.

The Gene Ontology Consortium, Gene Ontology: tool for the unification of biology (2000), *Nature Genetics*, 25:25-29.

van Helden, J., Gilbert, D., Wernisch, L., Schroeder, M., Wodak, S. (2001), Applications of regulatory sequence analysis & metabolic network analysis to the interpretation of gene expression data, *Lecture Notes in Computer Sciences* 2066: 155-172.

Verde, R., de Carvalho, F. A. T., Lechevallier, Y. A. (2000), Dynamical Clustering algorithm for Multi-nominal Data. In Kiers, H. A. L., *et al.* (Eds): *Data Analysis, Classification, & Related Metohds*, Springer, Heidelberg, 387-394.

Vesanto J., Alhoniemi, E. (2000), Clustering of the Self-Organizing Map, *IEEE Transactions on Neural Networks*, 11(3):586-600.

Vesanto, J., Himberg, J., Alhoniemi, E., Parhankangas, J. (2000), *SOM Toolbox for Matlab 5*, Technical Report, Helsinki University of Technology, Neural Networks Research Centre.

Yang, Y., Dudoit, S., Luu, P., Lin, D., Peng, V., Ngai, J., Speed, T. (2001), Normalization for cDNA Microarray Data: a robust compose method adressing single & multiple slide systematic variarion, *Nucleic Acids Research*, 30(4):e15.

Yang, Y. H., Buckley, M. J., Speed, T. P. (2001), Analysis of cDNA microarray images, *Briefings in Bioinformatics*, 2:341-349.

Yeung K. Y. Haynor, D. R., Ruzzo, W. L. (2001), Validating Clustering for Gene Expression Data, *Bioinformatics*, 17:309-318.

Zhu J., Zhang M.Q. (2000), Cluster, function & promoter: analysis of yeast expression array, *Proc. of Pacific Symponsiun of Biocomputing*, 479-490.

Appendix A

Parametrisation of SOM

This appendix ilustrastes the results of the parametrisation experiments with SOM. As stated in Section 4.3.3, SOM requires parametrization experiments in order to tune its performance. Due to the number of parameters available, and the complexity of choosing them, only a reduced set of parameters will be varied. Previous studies with gene expression data has found that topology was the parameter with highest impact of the results (Jonsson, 2001). As a result, the topology will be the only parameter to be varied.

The following procedure was applied to vary the topology. First, an initial topology is chosen. Then, experiments with a larger and smaller topology are also performed. If the initial topology obtain the best results then no more experiments are done. Otherwise, the same process is repeated for the topology with best result. In the *FC Yeast All* and *FC CDC 25* data sets, the initial topology was 10x10, while in the *Reduced FC Yeast All* and *Series CDC FC* the initial topology was 5x5.

In order to set the other parameters from SOM, a method of the toolbox that uses a number of heuristics to set the parameters was used (this parametrisation

is refereed as *DEFAULT*). The detailed description of this heuristics can be found in Vesanto *et al.* (2000). As not all the results obtained by this parametrization were satisfactory, another parametrisation based on the one used in Vesanto & Alhoniemi (2000) was employed (this parametrisation is refereed as *VESANTO*). The *VESANTO* parametrisation had 10 epochs and a learning rate of 0.5 during the ordering phase. The initial radius was set to the topology highest dimension and the final radius to half the highest dimension. In the convergence phase, 10 epochs and a learning rate of 0.05 are used. The initial radius is set to half the highest topology dimension minus 1 and the final radius to 1. The exact initial and final radius can be seen in Table A.1 and Table A.2.

	4x4	6x6	8x8	10x10	12x12
initial radius (ordering phase)	4	6	8	10	12
final radius (ordering phase)	2	3	4	5	6
initial radius (convergence phase)	1	2	3	4	5
final radius (convergence phase)	1	1	1	1	1

Table A.1: Topologies and parameters used in the *VESANTO* parametrisation with the *FC Yeast All* and *FC CDC 25* data sets.

Table A.3 shows the type of parameterisation and the topology that obtained the best accuracy for each proximity index and data set. In a whole, topologies smaller than the initial one obtained the best results. In relation to the type of parametrisation, neither *VESANTO* or *DEFAULT* showed advantage over the other.

	3x3	4x4	5x5	6x6	7x7
initial radius (ordering phase)	3	4	5	6	7
final radius (ordering phase)	2	2	2	3	3
initial radius (convergence phase)	1	2	2	3	3
final radius (convergence phase)	1	1	1	1	1

Table A.2: Topologies and parameters used in the *VESANTO* parametrisation with the *Reduced FC Yeast All* and *Series CDC 25* data sets.

	FC Yeast All	*Red. FC Yeast All*	*FC CDC 25*	*Series CDC 25*
ED_1	*VESANTO* - 4x4	*VESANTO* - 3x3	*DEFAULT* - 8x8	*DEFAULT* - 5x5
ED_2	*DEFAULT* - 4x4	*VESANTO* - 3x3	*VESANTO* - 4x4	*VESANTO* - 4x4
ED_3	*DEFAULT* - 8x8	*DEFAULT* - 3x3	*VESANTO* - 6x6	*DEFAULT* - 3x3
AS	*DEFAULT* - 4x4	*DEFAULT* - 3x3	*VESANTO* - 4x4	*DEFAULT* - 3x3
PC	*DEFAULT* - 8x8	*VESANTO* - 6x6	*VESANTO* - 10x10	*DEFAULT* - 5x5

Table A.3: Type of parametrisation and topologies with best accuracy in the experiments wiht SOM.

Appendix B

Results of the Experiments

	ED_1	ED_2	ED_3	AS	PC
Minimum	-0.038955000	-0.02011100	-0.00715300	-0.02011100	-0.00715300
1st Quartile	-0.011842500	0.00107025	0.01266925	0.00107025	0.01266925
Mean	0.001434833	0.01390043	0.02492750	0.01390043	0.02492750
Median	0.003211000	0.01342950	0.01938250	0.01342950	0.01938250
3rd Quartile	0.015756500	0.02172650	0.03366200	0.02172650	0.03366200
Maximum	0.035862000	0.05571200	0.08626300	0.05571200	0.08626300
Std Dev.	0.021609145	0.01831361	0.02147350	0.01831361	0.02147350

Table B.1: Detailed results of the SOM method in the experiments with the *FC Yeast All* data set

	ED_1	ED_2	ED_3	AS	PC
Minimum	-0.00589800	-0.02209700	-0.01933300	-0.01690300	-0.02208400
1st Quartile	0.00524825	-0.00708550	-0.00276125	0.00058600	0.00207075
Mean	0.02278537	0.00994930	0.01390600	0.01878857	0.01770200
Median	0.02081150	0.00581650	0.00917000	0.01420100	0.00962450
3rd Quartile	0.03758400	0.02189675	0.02495825	0.03079775	0.02945175
Maximum	0.06536200	0.06250300	0.09621600	0.09306800	0.07647000
Std Dev.	0.02114637	0.02070905	0.02515325	0.02611256	0.02510121

Table B.2: Detailed results of the hierarchical clustering method in the experiments with the *FC Yeast All* data set

	ED_1	ED_2	ED_3	AS	PC
Minimum	-0.01307200	-0.01848900	-0.02618400	-0.01106100	-0.03525500
1st Quartile	-0.00059550	-0.00367500	0.00237350	0.00920575	0.00208275
Mean	0.01008307	0.01144753	0.01751193	0.02872277	0.01497493
Median	0.00771900	0.01449450	0.01706250	0.02627250	0.01231550
3rd Quartile	0.01433575	0.02116800	0.02799325	0.04432500	0.02454025
Maximum	0.05043400	0.08472600	0.06005700	0.09312800	0.08445500
Std Dev.	0.01573918	0.02047774	0.02196634	0.02782920	0.02434092

Table B.3: Detailed results of the dynamical clustering method in the experiments with the *FC Yeast All* data set

	ED_1	ED_2	ED_3	AS	PC
Minimum	-0.0192270	-0.0247980	-0.0254510	-0.0169030	-0.0204350
1st Quartile	0.0029902	-0.0045587	-0.0034280	0.0070192	-0.0017165
Mean	0.0134798	0.0118640	0.0128157	0.0263822	0.0143914
Median	0.0111390	0.0037545	0.0141965	0.0188280	0.0123260
3rd Quartile	0.0303467	0.0289612	0.0257190	0.0395310	0.0237000
Maximum	0.0404540	0.0654960	0.0556210	0.1120520	0.0733450
Std Dev.	0.01772847	0.0225336	0.0204279	0.030316	0.0219716

Table B.4: Detailed results of the dynamical clustering method with the hierarchical initialisation in the experiments with the *FC Yeast All* data set

	ED_1	ED_2	ED_3	AS	PC
Minimum	-0.02046400	-0.01971200	-0.01885600	-0.01741900	-0.00797700
1st Quartile	-0.00086700	-0.00036375	-0.00314075	0.00788625	0.01095450
Mean	0.01102020	0.01334937	0.01521707	0.02679240	0.02104383
Median	0.01052900	0.01063400	0.01405750	0.02154050	0.01942100
3rd Quartile	0.02598250	0.02461725	0.03280550	0.04146975	0.03269625
Maximum	0.05464900	0.10320800	0.07197200	0.10494000	0.05287200
Std Dev.	0.01913167	0.02448483	0.02423892	0.02930881	0.01719546

Table B.5: Detailed results of the k-means method in the experiments with the *FC Yeast All* data set

	ED_1	ED_2	ED_3	AS	PC
Minimum	-0.032699000	-0.039570000	-0.04498300	-0.04348100	-0.0274750
1st Quartile	-0.006754000	-0.007965750	-0.00445250	0.00350650	-0.0143537
Mean	-0.000617966	-0.000507666	0.00843563	0.01396167	0.0068099
Median	-0.001645000	0.000230000	0.00961100	0.01047700	0.0022325
3rd Quartile	0.004225250	0.006265750	0.01843375	0.02617025	0.0261447
Maximum	0.043605000	0.033034000	0.06111200	0.07309700	0.0922250
Std Dev.	0.0122873723	0.013360061	0.02609532	0.0216848	0.0280843

Table B.6: Detailed results of the k-means method with the hierarchical initialisation in the experiments with the *FC Yeast All* data set

	ED_1	ED_2	ED_3	AS	PC
Minimum	0.1666660	0.2500000	0.6500000	0.0374990	0.3170730
1st Quartile	0.6400000	0.8137305	0.8590600	0.7892855	0.7929890
Mean	0.7793728	0.8613679	0.9206397	0.8418367	0.8553418
Median	0.8739740	1.0000000	1.0000000	0.9524885	0.8969325
3rd Quartile	1.0000000	1.0000000	1.0000000	1.0000000	1.0000000
Maximum	1.0000000	1.0000000	1.0000000	1.0000000	1.0000000
Std Dev.	0.2472943	0.2154636	0.1059930	0.2424424	0.1865070

Table B.7: Detailed results of the SOM method in the experiments with the *Reduced FC Yeast All* data set

	ED_1	ED_2	ED_3	AS	PC
Minimum	0.00000000	0.0000000	0.3859640	0.3823520	0.5161290
1st Quartile	0.03749925	0.3636730	0.6425000	0.7555550	0.8888880
Mean	0.29620580	0.5056674	0.8274096	0.8506834	0.9233477
Median	0.22126400	0.5161290	0.9524885	0.8969325	1.0000000
3rd Quartile	0.43229150	0.6475000	1.0000000	1.0000000	1.0000000
Maximum	0.90497700	1.0000000	1.0000000	1.0000000	1.0000000
Std Dev.	0.29596473	0.3104467	0.2100381	0.1766119	0.1249292

Table B.8: Detailed results of the hierarchical clustering method in the experiments with the *Reduced FC Yeast All* data set

	ED_1	ED_2	ED_3	AS	PC
Minimum	0.3170730	0.3152170	0.1025640	0.1126760	0.3170730
1st Quartile	0.8665170	0.8599222	0.7519835	0.3143635	0.7681218
Mean	0.9026139	0.8711358	0.8454977	0.4953523	0.8491364
Median	1.0000000	1.0000000	0.8888880	0.4722220	0.8888880
3rd Quartile	1.0000000	1.0000000	1.0000000	0.6400000	1.0000000
Maximum	1.0000000	1.0000000	1.0000000	1.0000000	1.0000000
Std Dev.	0.1622026	0.2006912	0.2052012	0.2360171	0.1898708

Table B.9: Detailed results of the dynamical clustering method in the experiments with the *Reduced FC Yeast All* data set

	ED_1	ED_2	ED_3	AS	PC
Minimum	0.2500000	0.3170730	0.0374990	0.0869560	0.0374990
1st Quartile	0.8590600	0.8665170	0.5987232	0.4111102	0.6335225
Mean	0.8670838	0.8880683	0.7730904	0.6263922	0.7757642
Median	1.0000000	1.0000000	0.8888880	0.6290905	0.8768760
3rd Quartile	1.0000000	1.0000000	1.0000000	0.9762442	1.0000000
Maximum	1.0000000	1.0000000	1.0000000	1.0000000	1.0000000
Std Dev.	0.2252518	0.1740505	0.2870277	0.2992167	0.2748604

Table B.10: Detailed results of the dynamical clustering with the hierarchical initialisation method in the experiments with the *Reduced FC Yeast All* data set

	ED_1	ED_2	ED_3	AS	PC
Minimum	0.3170730	0.2222220	0.1666660	-0.1413040	0.1463410
1st Quartile	0.8559565	0.3678252	0.8524652	0.2108572	0.7519835
Mean	0.8808899	0.6947467	0.8533684	0.4074373	0.8366877
Median	1.0000000	0.8223870	0.8969325	0.3856065	0.8888880
3rd Quartile	1.0000000	1.0000000	1.0000000	0.6090322	1.0000000
Maximum	1.0000000	1.0000000	1.0000000	1.0000000	1.0000000
Std Dev.	0.1880846	0.2964676	0.2204303	0.2537918	0.1940716

Table B.11: Detailed results of the k-means method in the experiments with the *Reduced FC Yeast All* data set

	ED_1	ED_2	ED_3	AS	PC
Minimum	0.2500000	0.3170730	0.0374990	0.0740740	0.1249990
1st Quartile	0.8590600	0.8665170	0.5161290	0.4111102	0.5486482
Mean	0.8670838	0.8880683	0.7044799	0.5275299	0.7428550
Median	1.0000000	1.0000000	0.7181960	0.5247310	0.8619620
3rd Quartile	1.0000000	1.0000000	1.0000000	0.6362500	1.0000000
Maximum	1.0000000	1.0000000	1.0000000	1.0000000	1.0000000
Std Dev.	0.2252518	0.1740505	0.2702943	0.2545882	0.2770787

Table B.12: Detailed results of the k-means with the hierarchical initialisation method in the experiments with the *Reduced FC Yeast All* data set

	ED_1	ED_2	ED_3	AS	PC
Minimum	-0.026465000	-0.02353700	-0.02743500	-0.01259400	-0.02855800
1st Quartile	-0.013982750	0.00081600	0.00076250	-0.00166175	-0.01278025
Mean	0.002310433	0.01851277	0.01495410	0.01600093	0.01194507
Median	-0.001790000	0.01724300	0.01462300	0.01107200	0.00483600
3rd Quartile	0.012871750	0.03695675	0.03026000	0.02902600	0.03574000
Maximum	0.055510000	0.06381000	0.05800100	0.07681300	0.06948100
Std Dev.	0.023394508	0.02405881	0.02206298	0.02190193	0.02586289

Table B.13: Detailed results of the SOM method in the experiments with the *FC CDC 25* data set

	ED_1	ED_2	ED_3	AS	PC
Minimum	-0.02271000	-0.025208000	-0.03339400	-0.01066600	-0.0324360
1st Quartile	-0.00240000	-0.011236750	-0.00893425	0.00000000	-0.0073320
Mean	0.00873153	0.000210066	0.00494963	0.00373496	0.0088233
Median	0.00432750	-0.002901500	0.00347400	0.00000000	0.0062020
3rd Quartile	0.01529625	0.010414000	0.01565775	0.00000000	0.0241665
Maximum	0.08727900	0.042238000	0.05645800	0.06038600	0.0556600
Std Dev.	0.02131606	0.016459574	0.02131238	0.01360641	0.0214893

Table B.14: Detailed results of the hierarchical clustering method in the experiments with the *FC CDC 25* data set

	ED_1	ED_2	ED_3	AS	PC
Minimum	-0.02042000	-0.01499800	-0.028298000	-0.02607200	-0.023044000
1st Quartile	-0.00636575	-0.00207425	-0.005215000	0.00304500	-0.007672750
Mean	0.00380890	0.01681667	0.009513133	0.01386087	0.009426333
Median	0.00135550	0.01197550	0.010249000	0.00939550	0.010751500
3rd Quartile	0.00856975	0.02605075	0.023440000	0.02404775	0.023276500
Maximum	0.04983800	0.08040600	0.048761000	0.05221700	0.055180000
Std Dev.	0.01621921	0.02446900	0.019859272	0.01815733	0.022023515

Table B.15: Detailed results of the dynamical clustering method in the experiments with the *FC CDC 25* data set

	ED_1	ED_2	ED_3	AS	PC
Minimum	-0.04220500	-0.02267300	-0.025887000	-0.02526400	-0.02694900
1st Quartile	-0.00486150	-0.00537200	-0.012403750	-0.00460775	-0.01037450
Mean	0.01145683	0.01776647	0.006880433	0.01081817	0.00798090
Median	0.00704950	0.01355600	0.005263500	0.00797400	0.00511950
3rd Quartile	0.02735750	0.02999100	0.021991000	0.02092275	0.02144875
Maximum	0.08246900	0.10073900	0.056232000	0.08306000	0.06444600
Std Dev.	0.02483053	0.03004264	0.022410962	0.02334287	0.02294249

Table B.16: Detailed results of the dynamical clustering method in the experiments with the *FC CDC 25* data set

	ED_1	ED_2	ED_3	AS	PC
Minimum	-0.01669200	-0.02559500	-0.01802300	-0.02688100	-0.03171800
1st Quartile	-0.00783050	0.00668475	-0.00014725	-0.00384750	-0.01126800
Mean	0.00632460	0.01796753	0.01301357	0.01076463	0.00000840
Median	0.00453250	0.01533800	0.01070450	0.01100400	0.00216100
3rd Quartile	0.01306025	0.03099350	0.02238025	0.02280075	0.01329875
Maximum	0.06059100	0.05872400	0.06095200	0.05165400	0.03226200
Std Dev.	0.01902226	0.01919484	0.01794062	0.02068030	0.01654204

Table B.17: Detailed results of the k-means method in the experiments with the *FC CDC 25* data set

	ED_1	ED_2	ED_3	AS	PC
Minimum	-0.02278000	-0.01821000	-0.01615000	-0.02361600	-0.033421000
1st Quartile	-0.00908575	-0.00255700	-0.00574375	0.00138675	-0.003519500
Mean	0.00539350	0.01599837	0.01397607	0.01882283	0.009854267
Median	0.00350250	0.00681200	0.01158250	0.01779500	0.010167000
3rd Quartile	0.01591750	0.03866150	0.03133950	0.03141950	0.025452500
Maximum	0.04625500	0.07946500	0.05914800	0.08556200	0.050062000
Std Dev.	0.01922413	0.02800058	0.02096454	0.02462083	0.024383853

Table B.18: Detailed results of the k-means method with the hierarchical initialisation in the experiments with the *FC CDC 25* data set

	AS	PC
Minimum	-0.0078200000	-0.0356400
1st Quartile	0.0000000000	-0.0200780
Mean	-0.0005700667	-0.0021954
Median	0.0000000000	-0.0012555
3rd Quartile	0.0000000000	0.0093820
Maximum	0.0000000000	0.0434950
Std Dev.	0.0017106661	0.0205161

Table B.19: Detailed results of the CLICK method in the experiments with the *FC CDC 25* data set

	ED_1	ED_2	ED_3	AS	PC
Minimum	-0.14609500	0.0722890	0.1223620	0.1512950	0.1170960
1st Quartile	-0.06958725	0.2766270	0.2945575	0.2940890	0.2886520
Mean	-0.01055743	0.3892668	0.4144066	0.4092227	0.4354292
Median	-0.02492200	0.3709215	0.3839525	0.3755730	0.4472155
3rd Quartile	0.01138975	0.4829067	0.5632032	0.5510202	0.5734952
Maximum	0.26916500	0.8460230	0.8212180	0.8239840	0.8212180
Std Dev.	0.10250789	0.1884654	0.1907337	0.1816961	0.1990142

Table B.20: Detailed results of the SOM method in the experiments with the *Series CDC 25* data set

	ED_1	ED_2	ED_3	AS	PC
Minimum	-0.18181800	0.0791160	0.1053370	0.0597010	0.1845010
1st Quartile	-0.06236975	0.2116350	0.2730070	0.2848555	0.2786980
Mean	0.00057430	0.3500720	0.3929059	0.4178399	0.3949949
Median	-0.01591600	0.2871400	0.3869960	0.4483040	0.3758325
3rd Quartile	0.03626225	0.4950988	0.4861800	0.5248928	0.4945055
Maximum	0.26778200	0.7271730	0.7448380	0.7448380	0.7172150
Std Dev.	0.11307053	0.1819851	0.1618505	0.1822161	0.1504085

Table B.21: Detailed results of the hierarchical clustering method in the experiments with the *Series CDC 25* data set

	ED_1	ED_2	ED_3	AS	PC
Minimum	-0.18954200	0.0791160	0.1304340	0.1053370	0.0995670
1st Quartile	-0.05383400	0.1960662	0.2784205	0.2402258	0.2182145
Mean	0.02366617	0.3083868	0.4028769	0.3946001	0.3879385
Median	0.00926450	0.2815780	0.3813080	0.3481525	0.3813080
3rd Quartile	0.06391275	0.4138440	0.5019395	0.5353082	0.5019395
Maximum	0.31906600	0.6845250	0.7271730	0.7448380	0.7271730
Std Dev.	0.12306776	0.1470607	0.1709510	0.1723212	0.1852805

Table B.22: Detailed results of the dynamical clustering method in the experiments with the *Series CDC 25* data set

	ED_1	ED_2	ED_3	AS	PC
Minimum	-0.169714000	0.0791160	0.1845010	0.0597010	0.1512950
1st Quartile	-0.075091250	0.2108388	0.2909635	0.2848555	0.3090785
Mean	-0.002685267	0.3484938	0.3941878	0.4172044	0.4080005
Median	-0.017190000	0.2871400	0.3813080	0.4483040	0.3869960
3rd Quartile	0.041951750	0.4950988	0.4909160	0.5248928	0.5019395
Maximum	0.268274000	0.7271730	0.7283580	0.7448380	0.7172150
Std Dev.	0.114705161	0.1829879	0.1412790	0.1829587	0.1594077

Table B.23: Detailed results of the dynamical clustering method with the hierarchical initialisation in the experiments with the *Series CDC 25* data set

	ED_1	ED_2	ED_3	AS	PC
Minimum	-0.156473000	0.1340830	0.0800000	0.0597010	0.0597010
1st Quartile	-0.087649000	0.2126630	0.3038920	0.3108712	0.2411188
Mean	0.003923567	0.3068223	0.3924228	0.4000893	0.3793802
Median	-0.013970500	0.2812560	0.3667820	0.3881490	0.3759840
3rd Quartile	0.065359000	0.4051102	0.5326460	0.5274132	0.4914248
Maximum	0.367041000	0.6845250	0.7172150	0.7283580	0.7271730
Std Dev.	0.112776364	0.1417217	0.1679634	0.1724263	0.1785361

Table B.24: Detailed results of the k-means method in the experiments with the *Series CDC 25* data set

	ED_1	ED_2	ED_3	AS	PC
Minimum	-0.13576800	0.1512950	0.1007900	0.1007900	0.1294640
1st Quartile	-0.06707075	0.3039733	0.2767645	0.2799075	0.2627222
Mean	-0.01973157	0.4098164	0.4131179	0.4140051	0.4175295
Median	-0.03349250	0.3734520	0.3955645	0.4157565	0.3925210
3rd Quartile	0.02065325	0.5065978	0.5414718	0.5200700	0.5322922
Maximum	0.16428500	0.7271730	0.6976740	0.7448380	0.7663070
Std Dev.	0.07463268	0.1457338	0.1657752	0.1720371	0.1828924

Table B.25: Detailed results of the k-means method with the hierarchical initialisation in the experiments with the *Series CDC 25* data set

	AS	*PC*
Minimum	0.15189800	0.0259480
1st Quartile	0.03358100	0.3414435
Mean	0.01529393	0.4206158
Median	0.00000000	0.4075320
3rd Quartile	0.03937275	0.5755730
Maximum	0.38362000	0.7283580
Std Dev.	0.09963067	0.1914951

Table B.26: Detailed results of the CLICK method in the experiments with the *Series CDC 25* data set