Multi-Biclustering Solutions for Classification and Prediction Problems

Department of Mathematics and Informatics

Contents

List of Figures

Abstract

The search for similarities in large data sets has a relevant role in many scientific fields. It permits to classify several types of data without an explicit information about them. Unfortunately, the experimental data contains noise and errors, and therefore the main task of mathematicians is to find algorithms that permit to analyze this data with maximal precision. In many cases researchers use methodologies such as clustering to classify data with respect to the patterns or conditions. But in the last few years new analysis tool such as biclustering was proposed and applied to many specific problems. My choice of biclustering methods is motivated by the accuracy obtained in the results and the possibility to find not only rows or columns that provide a dataset partition but also rows and columns together.

In this work, two new biclustering algorithms, the Combinatorial Biclustering Algorithm (CBA) and an improvement of the Possibilistic Biclustering Algorithm, called Biclustering by resampling, are presented. The first algorithm (that I call Combinatorial) is based on the direct definition of bicluster, that makes it clear and very easy to understand. My algorithm permits to control the error of biclusters in each step, speci-

fying the accepted value of the error and defining the dimensions of the desired biclusters from the beginning. The comparison with other known biclustering algorithms is shown.

The second algorithm is an improvement of the Possibilistic Biclustering Algorithm (PBC). The PBC algorithm, proposed by M. Filippone et al., is based on the Possibilistic Clustering paradigm, and finds one bicluster at a time, assigning a membership to the bicluster for each gene and for each condition. PBC uses an objective function that maximizes a bicluster cardinality and minimizes a residual error. The biclustering problem is faced as the optimization of a proper functional. This algorithm obtains a fast convergence and good quality of the solutions. Unfortunately, PBC finds only one bicluster at a time. I propose an improved PBC algorithm based on data resampling, specifically Bootstrap aggregation, and Genetics algorithms. In such a way I can find all the possible biclusters together and include overlapped solutions. I apply the algorithm to a synthetic data and to the Yeast dataset and compare it with the original PBC method.

Dedication

I dedicate this work to my dear parents Marina Nosova e Mikhail Nosov, the most special persons in my life. Unfortunately, all my life I stay fare from you because of my study and work. But every day I feel that you love me, even I don't see you. This love helps me to overcome all difficult moments, I know that I'm not alone. You gave me my life and I'm happy to be your daughter. Thank you!

Acknowledgments

I would like to express my thanks to:

- Director of my research work, amazing person, Prof. Roberto Tagliaferri, for everything he made to me. Without him my work in the University would be impossible. Thank to him I received many significant results in the research, met a lot of interesting people. With him my research work was a pleasure to me and the period in the University I will never forget. You are one of the most important person in my life. Thank you!

- My Tutor, Prof.ssa Beatrice Paternoster, for supervision and supporting of my work.

- Prof. Giancarlo Raiconi for the help, advices and support.

- Dott. Francesco Napolitano, who helped me with many difficulties I had during my work. He discussed with me all my questions, helped to understand better MatLab, Clustering problems and many others, and recommended many winning solutions. I also thank him for being my dear friend and colleague.

- Prof. Sergio Cocozza, Dot. Roberto Amato, Prof. Gennaro Miele - the group of co-workers from University of Federico II (Naples) for collaboration and good results of my work.
- Prof. Francesco Masulli for the collaboration and valuable advices to the obtaining of a new algorithms.
- My parents for the supporting, believing in myself and love.
- My boyfriend Luca and his family - the wonderful people that accepted me like a daughter. For their moral supporting every day, encouraging worlds and help in everything.
- INdAM for the supporting of my work, especially Prof.ssa Elisabetta Strickland for help and understanding. And Mauro Petrucci for his friendship.
- I thank all my coworkers from department of Informatics and Mathematics for being friends.
- The biological part of this thesis mainly use the material of Wikipedia and [1].

Publications

Portions of the work described in this thesis has also appeared in:

Conference papers

1. E. Nosova, R. Tagliaferri, G. Raiconi, *A Multi-biclustering Combinatorial Based Algorithm*.

   This work was accepted and now is in press of CIDM 2011 post-conference proceeding.

   This paper will be presented in 2011 IEEE Symposium on Computational Intelligence and Data Mining (April 11-15, 2011 - Paris, France (CIDM 2011))

2. E. Nosova, F. Napolitano, G. Raiconi, R. Tagliaferri, S. Cocozza, R. Amato, G. Miele, *Toward an Improved Combinatoric Algorithm*, Proceedings of the Network tools and applications in biology, NETTAB-BBCC 2010, Edited by Angelo Facchiano and Paolo Romano, 2010, 71–77

This paper was presented by E. Nosova in Network Tools and Applications in Biology Workshop(November 29 - December 1, 2010, Naples, Italy (NETTAB 2010))

3. E. Nosova, R. Tagliaferri, F. Masulli, S. Rovetta, *Biclustering by Resampling*.

   This paper was accepted and now is in press of post conference volume of Lecture Notes in Bioinformatics LNBI/LNCS series of Springer Verlag. This paper was presented by E. Nosova in Seventh International Meeting on computational intelligence methods for bioinformatics and biostatistics (Palermo September 16-18 2010 (CIBB10))

4. E. Nosova, G. Raiconi and R. Tagliaferri, *A Combinatoric biclustering algorithm*, Proceedings of the 20th Italian Workshop on Neural Nets. Edited by Bruno Apolloni, Simone Bassis, Anna Esposito, Carlo Francesco Morabito, Volume 226, 2011, pp: 44 - 51

   This paper was presented by E. Nosova in WIRN 2010 workshop (Vietri sul Mare 28/05/2010).

Journals (submitted, under review)

- E. Nosova, F. Napolitano , G. Raiconi , R. Tagliaferri , S. Cocozza , R. Amato and G. Miele, *A Combinatorial Biclustering algorithm for Gene Expression Data*, prepared and ready to press in BMC Bioinformatics, 2011

Prologue

"...Just as Kepler and Newton made these predictions and discoveries by using mathematical frameworks to describe trends in astronomical data, so future predictive power, discovery, and control in biology and medicine will come from the mathematical modeling of DNA microarray data, where the mathematical variables and operations represent biological reality. The variables, patterns uncovered in the data, might correlate with activities of cellular elements, such as regulators or transcription factors, that drive the measured signals. The operations, such as data classification and reconstruction in subspaces of selected patterns, might simulate experimental observation of the correlations and possibly also causal coordination of these activities. Such models were recently created from DNA Microarray data by using singular value decomposition (SVD) and generalized SVD (GSVD), and their ability to predict previously unknown biological as well as physical principles was demonstrated..."

– Orly Alter.

Chapter 1

Introduction

1.1 Introduction to Bioinformatics and genomics.

Look around and you can see, how beautiful is life in its diversity: from the simple cells to mammals and humans. And it is strange, that this diversity depends on a linear code inside small living cells. Following the frequent rule of life: "Just a stroke of genius", like a binary code that control the computers, four DNA basis control all complicity of the genetic code. Very important discovery of the relationship between DNA and proteins, their functions and properties cames in the twentieth century and led to a revolution in the genetics understanding. Since that time we discover all the days a new information about genome. This chaos if material provides many difficulties in the analysis. One of the tasks of biologists today is organize, study and make the conclusions from all this

information. For all these problems a new Science such as Bioinformatics was proposed. I use *Bioinformatics* to get a better understanding of living systems. [1]

As it is known from Wikipedia, the term of “bioinformatics” was invented by Paulien Hogeweg in 1979 for the study of informatics processes in biological systems. And was widely used from the late 1980s in genomics and genetics. Bioinformatics includes such fields as (See Wikipedia):

- Mathematical methods of the computing analysis of comparative genomics (genomic bioinformatics).
- Development of the algorithms and software for predicting the spatial structure of proteins (structural bioinformatics).
- Researching of the strategies that correspond to computational methodologies as well as overall management of information complexity of biological systems.

Genomics is one of the fields of Bioinformatics. According to the Oxford Dictionary, Genomics is the branch of molecular biology concerned with the structure, function, evolution, and mapping of genomes, in particular, the suffix -ome means "all constituents considered collectively". An interesting areas of the Genomics is detection and analysis of the nucleic acids structures—deoxyribonucleic acid (DNA) and ribonucleic acid (RNA).

DNA contains the coding ("coding" DNA) and non-coding areas. “Scientists now estimate that humans have about 30,000 genes, located along

threadlike, tightly coiled strands of DNA called chromosomes. However, there are about three percent of genes in a human DNA; the rest consists of a "noncoding" DNA. These noncoding regions of the genome contain the information about an activity of a genes. For example, they determine in which cell types and at what stages of an organism life genes are active. Genomics is the study of the entire set of DNA sequences–both coding and noncoding DNA".[2]

1.2 Methods for gene analysis.

1.2.1 Clustering.

Follow I use the notes of [3]. Clustering is the unsupervised classification of patterns (observations, data items, or feature vectors) into groups (clusters). In other words, a cluster is a collection of objects which are "similar" between them and are "dissimilar" to the objects belonging to other clusters. Despite a simplicity of the definition, clustering is a difficult problem, used in many disciplines such that biology, psychiatry, psychology, archeology, geology, geography, and marketing. So, the goal of clustering is to determine groups in a set of unlabeled data. But how to understand, what groups provide a good clustering? It can be shown that there is no absolute criterion which would be independent of the final aim of the clustering. Consequently, the different clustering algorithm can provide absolutely different result. For example, it can be find representatives for homogeneous groups (data reduction) or unusual data

objects (outlier detection). Such, clustering involve a number of problems, providing a production and development of a clustering algorithms.

Most common clustering algorithms for the genetical data canalization are:

- K-means - an exclusive clustering algorithm [4]
- Fuzzy C-means - overlapping clustering algorithm [16]
- Hierarchical clustering [5]

Most clustering techniques identify clusters according to a distance between each pair of data points and therefore need a definition of this distance measure 1.1. It influence the shape of the clusters, as some elements may be close to one another according to one distance and farther away according to another. Common distance functions can be, for example, Euclidean distance, Hamming distance and other.

Main steps of clustering can be defined as:

- data preparation (cleaning data, data transformations, selecting subsets of records and - in case of data sets with large numbers of variables ("fields") performing some preliminary *feature selection* operations to bring the number of variables to a manageable range)
- background correction (adjustments to the data, removing of nonbiological contributions "background" to the measured signal)

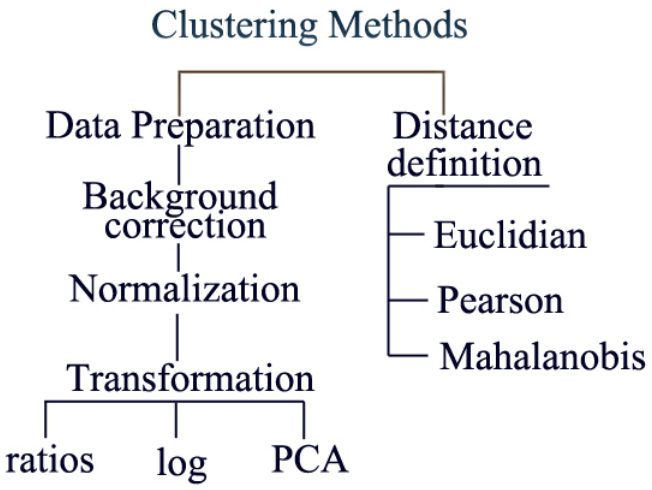


Figure 1.1: Clustering methods and Statistic.

- normalization (decomposing relations with anomalies in order to produce smaller, well-structured relations)
- transformation (application of a deterministic mathematical function to each point in a data set)

Cluster analysis can be performed not only to identify genes whose expression levels change in similar ways, but also to identify samples that have similar expression patterns. These samples could for example be different organisms or different conditions, or a combination of the two. The distance must be defined as a number, and therefore each gene or sample in the experiment requires a set of quantitative parameters. [1]

As many algorithms, clustering has some limitations. First, it is based on the assumption that related genes are similar for the most conditions.

But from studies of cellular processes, it is known that groups of genes are co-regulated and coexpressed under certain experimental conditions, and also behave almost independently under other conditions. Second, clustering solutions often divide genes into disjoint sets, implying association of each gene with a biological function or process that can simplify the biological system. To solve these problems, Biclustering technique was proposed and widely used in Bioinformatics.

1.2.2 Biclustering.

A Bicluster of a gene expression dataset is a subset of genes which shows similar trends in terms of a subset of conditions. Biclustering techniques find submatrices, which are closely regulated in accordance with some scoring criterion. In practice, it is need to build a collection of submatrices (biclusters) that fix every significant parts of gene expression data, and differently from clusters these matrices can be overlapped or cover the entire matrix.

The technique of biclustering was originally introduced by J.A. Hartigan (1972) [6] and the term was firstly introduced by Mirkin (1996) [7] (later by Cheng and Church [8] (2000) in gene expression analysis). Cheng and Church introduced a similarity score called mean squared residue as a measure of the rows coherence. They identify one bicluster at a time, mask it with random numbers, and repeat the procedure in order to eventually find other biclusters.

My choice of biclustering methods is motivated by the accuracy in the

obtained results and the possibility to find not only rows or columns that provide a partition of the dataset, but also rows and columns together.

1.3 Aim of the Thesis

In this thesis two new algorithms of biclustering *Improved PBC* (Possibilistic Biclustering algorithm) and *CBA* (Combinatorial Biclustering Algorithm) are presented. Improved PBC is based on the Possibilistic approach to biclustering, supplemented by Bagging technique and Genetic Algorithms. CBA, instead, is based on the variance of the bicluster entries, analyzed by Bimax algorithm with applying of other techniques.

These new mathematical models generalize a good result and are validated by new techniques. The proposed algorithms solve a number of Clustering problems. For example, they do not require a supervised definition of number of the clusters, separate the data with respect to a part of columns and a part the rows, do not use concept of the distance. In addition, the new algorithms solve many biclustering problems, such that data analysis, running speed and stability of results.

1.4 Structure of the thesis

In the 1st Chapter of the thesis I give main biological terms, make an introduction to Bioinformatics and genomics in the simple therms, easy understandable to non-specialists. I also discus mathematical problems in

Bioinformatics, such that, for example, classification of the data, and give possible solution of these problems (Clustering and Biclustering techniques).

In the 2d Chapter I introduce the Biological Basis, that includes the structure and functions of DNA and RNA, their link with the central dogma of molecular biology and gene expression analysis. I present a biological method of the data obtaining, such that Microarray technology and discuss the main aim of gene expression analysis.

The 3d Chapter introduces better the biclustering technology. I begin this chapter from the history of the biclustering, present some important persons and their works (Cheng and Church, Lazzeroni, Kluger and many others). Then I give a definition of Bicluster in terms of row mean, column mean and bicluster mean, that is based on the residue score and MSR. And in the end of this chapter I define the main types of biclusters.

In the 4th Chapter I present the first result of my work – Combinatorial Biclustering Algorithm (CBA). This unsupervised method solves a number of biclustering problems. It finds all biclusters together, gives a possibility to define a minimal error, minimal number of the rows and the columns. I apply different techniques to CBA, such that Biclique technique, Sorting & Deleting algorithm and Bimax Algorithm. I apply CBA to many types of data, synthetic and real biological data. Comparison with other methods and conclusion are presented.

In the Chapter 5 I present the second result of my work, such that Biclustering by Resampling. Corresponding techniques of Fuzzy logic (Fuzzy

Clustering technique and Possibilistic Clustering Paradigm) are discussed. I also introduce the Possibilistic approach to biclustering and its improvement, based on the Bootstrap aggregating and Genetic Algorithms. The Biclustering by Resampling is tasted on the different types of the data. Comparison with other methods and conclusion are presented.

Chapter 2

Biological Basis

2.1 The Nucleic Acid World.

2.1.1 The Structure of DNA and RNA

Follow I use the definitions and explanations from [1]. The main role of DNA (*deoxyribonucleic acid*) is the information storage. It is a material that holds genetic instructions in all living cells, from unicellular bacteria to multicolor plants and animals. Chemical structure of DNA transmit these instructions from generation to generation, which creates and supports new organisms. It is incredibly, that a very large amount of information about complex organisms is stored in a relatively small number of DNA molecules. This set of molecules is called the *organism's genome*. Humans have 46 DNA molecules in most cells, only one DNA molecule is located in each chromosome. Each DNA molecule is copied

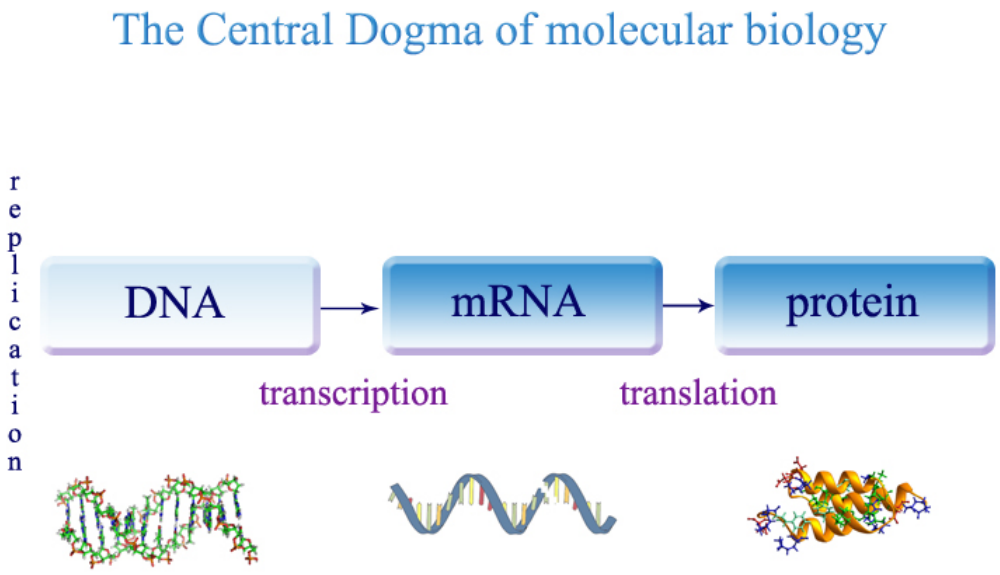
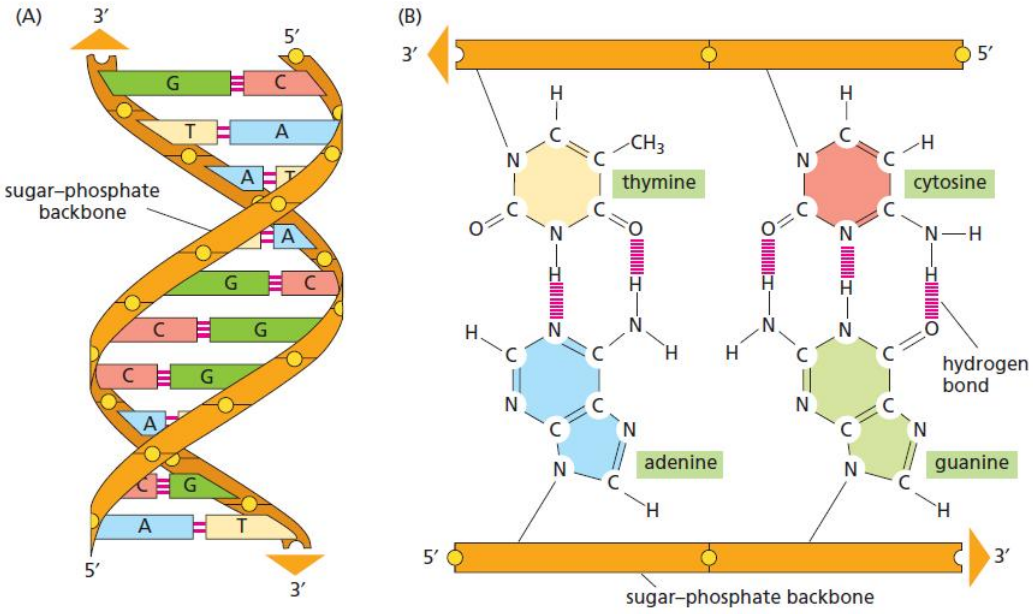


Figure 2.1: The Central Dogma of molecular biology.

and its copies are distributed in the way that each daughter cell receives a full set of genetic information (see 2.1).

Despite a complex role of the DNA, this molecule has a fairly simple chemical structure. They are linear polymers of four different nucleotide building blocks, whose differences are restricted to a substructure called *the base*. There are four bases of DNA molecule: guanine (G), adenosine (A), cytosine (C) and thymine (T). The three-dimensional structure of DNA is also relatively simple, involving regular double helices.

RNA molecules are also linear polymers, but they are much smaller than genomic DNA. Most RNA molecules also contain just four different base types, such that adenine (A), cytosine (C), guanine (G), or uracil (U). RNA



(A) DNA exists in cells mainly as a two-stranded coiled structure called the double helix. (B) The two strands of the helix are held together by hydrogen bonds (shown as red lines) between the bases; these bonds are referred to as basepairing. The figure is taken from [1].

Figure 2.2: The double helical structure of DNA.

molecules tend to have a less-regular three-dimensional structure than DNA. In most forms of RNA molecule there are also just four bases.

Each nucleic acid chain is a linear polymer of nucleotides linked together by phosphodiester linkages through the phosphate on one nucleotide and the hydroxyl group on the 3’ carbon on the sugar of another. The resulting chain has one end with a free phosphate group, which is known as the 5’ end, and one end with a free 3’ hydroxyl group, which is known as the 3’ end (see 2.2).

2.1.2 The Central Dogma of molecular biology.

The Central Dogma of molecular biology (see 2.1) was first articulated by Francis Crick in 1958 and re-stated in a Nature paper published in 1970. It deals with the detailed residue-by-residue transfer of sequential information. It states that information cannot be transferred back from protein to either protein or nucleic acid. In other words, "once information gets into protein, it can't flow back to nucleic acid".

The Central Dogma consists of 3 main position (See Wikipedia):

- *DNA Replication*. As a final step in the Central Dogma, the DNA must be replicated faithfully, to transmit the genetic information between parents and progeny. Replication is carried out by a complex group of proteins that unwinds the superhelix, unwinds the double-stranded DNA helix, and, using DNA polymerase and its associated proteins, copies or replicates the master template itself so the cycle can repeat DNA to RNA to protein in a new generation of cells or organisms.

- *Transcription*. Transcription is the process by which the information contained in a section of DNA is transferred to a newly assembled piece of messenger RNA (mRNA). It is facilitated by RNA polymerase and transcription factors. In eukaryote cells the primary transcript (pre-mRNA) is often processed further via alternative splicing. In this process, blocks of mRNA are cut out and rearranged, to produce different arrangements of the original sequence.

- *Translation*. Eventually, this mature mRNA finds its way to a ribosome, where it is translated. In prokaryotic cells, which have no nuclear compartment, the process of transcription and translation may be linked together. In eukaryotic cells, the site of transcription (the cell nucleus) is usually separated from the site of translation (the cytoplasm), so the mRNA must be transported out of the nucleus into the cytoplasm, where it can be bound by ribosomes. The mRNA is read by the ribosome as triplet codons, usually beginning with an AUG, or initiator methionine codon downstream of the ribosome binding site. Complexes of initiation factors and elongation factors bring aminoacylated transfer RNAs (tRNAs) into the ribosome-mRNA complex, matching the codon in the mRNA to the anti-codon in the tRNA, thereby adding the correct amino acid in the sequence encoding the gene. As the amino acids are linked into the growing peptide chain, they begin folding into the correct conformation. This folding continues until the nascent polypeptide chains are released from the ribosome as a mature protein. In some cases the new polypeptide chain requires additional processing to make a mature protein. The correct folding process is quite complex and may require other proteins, called chaperon proteins. Occasionally, proteins themselves can be further spliced; when this happens, the inside "discarded" section is known as an intein.

2.2 Gene Expression analysis.

Recently, different methods are used to produce biological databases, for example, RNA interference (RNAi), different methods of gene expression and protein expression analysis. In my work I use only common data bases, such as Microarrays.

Materials of this section are taken from Wikipedia and [1].

Gene expression begins when the gene is transcribed into messenger RNAs (mRNAs), which are then translated to produce proteins. One of the evaluation of gene expression is the detection and quantification of total RNA transcript using DNA *Microarray technology* (see 2.3). In this case, a single experiment produces an enormous amount of the data.

DNA microarrays and chips are composed of short fragments of DNA attached to a surface or synthesized directly on the surface, such as a glass microscope slide, in a predetermined arrangement, so that the sequence of the DNA fragment at any position is known. In the most basic form of a Microarray experiment, the testing mRNAs in the sample are labeled with fluorescent tags and mixed with the array. RNAs in the samples, that are complementary to fragments on the array, will base-pair or hybridize with the fragments. Unbound sample is washed away, and the Microarray is scanned with a fluorescence imager. RNAs that have bound their complementary array fragment are detected as fluorescent spots at specific positions, which give their identities, while the intensity of the fluorescence measures the level of the RNAs in the original sample. In

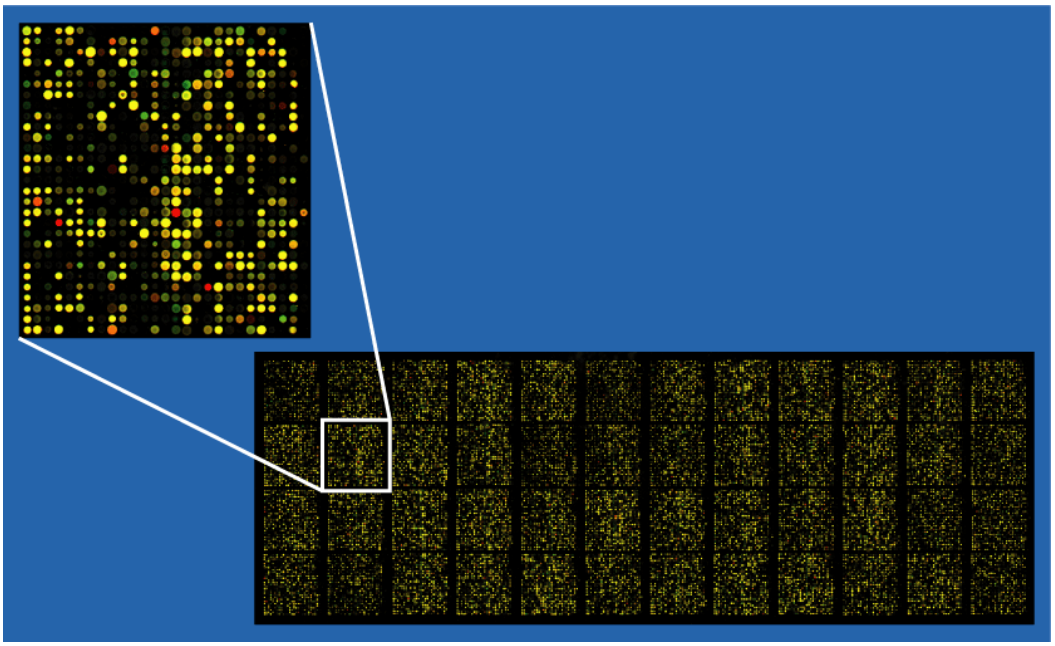
practice, the sample mRNA is first converted to cDNA by reverse transcription, and is also labeled in this reaction, or the RNA is amplified by in vitro transcription and then labeled, and this labeled RNA is hybridized to the array. For small-scale DNA arrays where high sensitivity is required, sample RNA can also be directly labeled with a radioactive tag without amplification.

A DNA array can contain from tens or hundreds to hundreds of thousands of different sequences, depending on the purpose for which it is to be used.

Most gene expression microarray experiments are intended not only detect the expressed genes at a given time, but also to detect differences in gene expression under different conditions.

There are two basic approaches in microarray technology: a one-color technique, where a single sample is hybridized to each microarray after it has been labeled with a single fluorophore; and the two-color procedure, where two samples are labeled with different fluorophores and hybridized together on a single microarray, as described above.

The main aim of gene expression analysis is the identification of common patterns of gene expression; for example, which genes are being co-expressed, and which genes have been downregulated or upregulated in one sample compared to the other.



Example of an approximately 40,000 probe spotted oligo microarray with enlarged inset to show detail. This figure is taken from Wikipedia.

Figure 2.3: Example of an approximately 40,000 probe.

Chapter 3

Biclustering: definition, history, problems.

3.1 History of the Biclustering.

The emergence of DNA microarray technology has revolutionized experimental studies of gene expression. Clustering is the most popular approach for the analysis of gene expression data, whose main objective is the identification of genes with same functions or regulatory mechanisms. As all algorithms, clustering has some limitations (that was discussed before).

A Bicluster of a gene expression dataset is a subset of genes which shows similar trends in terms of a subset of conditions. It finds submatrices, which are closely regulated in accordance with some scoring criterion. In practice, one wants to build a collection of submatrices (biclusters) that

fix every significant parts of gene expression data, and differently from clusters these matrices can be overlapped or cover the entire matrix.

Cheng and Church [8] proposed the concept of bicluster, based on a high similarity score as a measure of coherence of the genes and conditions (mean squared residue). The mean squared residue (MSR) is the variance of the set of all elements in the bicluster, plus the mean row variance and the mean column variance. For a good bicluster the value of MSR must be lower than a defined threshold. The method of Cheng and Church finds one bicluster at a time and is based on the removing and adding the rows and columns with a larger residue or a lower residue than a threshold, respectively. After determining the first bicluster they fill it by substituting the expression values with random numbers to find the second bicluster by the same way. The algorithm of Cheng and Church works well but makes impossible to find overlapped biclusters.

Based on the previous idea, Lazzeroni (2000) [9] presents the PLAID models, in which the data matrix is described as a linear function of layers corresponding to its biclusters and shows how to estimate a model using an iterative maximization process. Plaid models are a form of two-sided cluster analysis that allows clusters to overlap; they also incorporate additive two way ANOVA models within the two-sided clusters.

PLAID [9] consists of a series of additive layers intended to capture the underlying structure of a matrix. Each layer corresponds to a bicluster. Each element of the data matrix a_{ij} is modeled by

a_{ij}=$\sum_{k=0}^{K} \theta_{ijk}\rho_{ik}\kappa_{jk}$,

where K is the number of biclusters, ρ_{ik} and κ_{jk} are binary variables that represent the membership of row i and column j in layer k. Plaid uses the standard 2-way Anova decomposition for each layer k:

$$\theta_{ijk} = \mu_k + \alpha_{ik} + \beta_{jk}$$

where a μ_k is introduced to serve as a general mean, α_i is the membership for the row i, β_j is the membership for the columns j and θ_{ijk} is the plaid contribution for the element a_{ij} of the data matrix. Plaid is a form of overlapping two-sided clustering with a good speed.

The SAMBA algorithm (Statistical-Algorithmic Method for Bicluster Analysis) [12, 13] searches the gene-properties graph for statistically significant subgraphs. It defines a bicluster as a subset of genes that jointly respond across a subset of conditions. A gene respond to some condition if its expression level changes significantly at that condition with respect to its normal level. The input data is modeled as a bipartite graph with the two parts corresponding to conditions and genes respectively and edges for significant expression changes. Then the assignation of the weights to the edges of the graph occurs following two statistical models defined by the authors.

Prelic et al., (2006) [11], proposed a divide-and-conquer algorithm (BIMAX) for finding constant biclusters after discretizing the input expression matrix into a binary matrix. This discretization makes it harder to determine coherent biclusters.

Spectral [14] is a method that simultaneously clusters genes and con-

ditions, finding distinctive "checkerboard" patterns in matrices of gene expression data. It finds checkerboard structures in matrices of expression data by using eigenvectors, corresponding to characteristic expression patterns across genes or conditions. These eigenvectors are identified by singular value decomposition (SVD). Spectral algorithm depends much on the normalization over genes and conditions and existence of checkerboard structure.

The Possibilistic Biclustering algorithm, proposed by M. Filippone et al. [15], is based on the Possibilistic Clustering paradigm [16], and finds one bicluster at a time, assigning a membership to the bicluster for each gene and for each condition. The biclustering problem, in which one would maximize the size of the bicluster and minimizing the residual, is faced as the optimization of a proper functional. This algorithm obtains fast convergence and good quality solutions. PBC finds only one bicluster at time.

There are many other Biclustering techniques in literature. But no of them can find perfect biclusters in definitive time. Many of these algorithms find only one bicluster at a time, or being the graph technique, use a lot of time for calculation. In my work I try to present novel algorithms that finds better results than all known algorithms.

3.2 Bicluster definition.

Following Cheng and Church [8] let me to give a definition of bicluster. Let *A* be the expression matrix with elements a_{ij}, X be the set of genes and Y - the set of conditions. Let $I \subseteq X$ and $J \subseteq Y$ be subsets of genes and conditions. The pair (I, J) specifies a submatrix A_{IJ}.

A bicluster with coherent values identifies a subset of the genes and a subset of the conditions with coherent values on both rows and columns. I consider the additive model to find biclusters, but it can be also changed to the multiplicative one. In the case of the additive model, each element a_{ij} can be uniquely defined by its row mean, a_{iJ}, its column mean, a_{Ij}, and the bicluster mean, a_{IJ}. The difference $a_{Ij} - a_{iJ}$ is the relative bias held by the column j with respect to the other columns in the bicluster. The same reasoning applied to the rows leads to the definition that, in a perfect bicluster, the value of an element, a_{ij}, is given by a row-mean plus a column-mean minus the matrix mean:

$$a_{ij} = a_{iJ} + a_{Ij} - a_{IJ}.$$

In gene expression data, due to noise, biclusters may not always be perfect. The concept of residue was thus introduced to quantify the difference between the actual value of an element a_{ij} and its expected value predicted from the corresponding row mean, column mean, and bicluster mean. The residue score of an element a_{ij} in a submatrix A_{IJ} is defined

as:

$$RS_{IJ}(i,j) = a_{ij} - a_{Ij} - a_{iJ} + a_{IJ}.$$

In order to assess the quality of a bicluster, the mean squared residue, H, of a bicluster (I, J) is defined as the sum of the squared residues used to measure the coherence of the rows and columns in the bicluster:

$$H(I,J) = \frac{1}{|I| \times |J|} \sum_{i \in I, j \in J} RS_{IJ}(i,j)^2 = \frac{1}{|I| \times |J|} \sum_{i \in I, j \in J} (a_{ij} - a_{Ij} - a_{iJ} + a_{IJ})^2,$$

where

$$a_{iJ} = \frac{1}{|J|} \sum_{j \in J} a_{ij}, a_{Ij} = \frac{1}{|I|} \sum_{i \in I} a_{ij}, a_{IJ} = \frac{1}{|I| \times |J|} \sum_{j \in J, i \in I} a_{ij}.$$

A submatrix A_{IJ} is called a δ-bicluster if $H(I,J) \leq \delta$ for some $\delta \geq 0$.

The residue score of a_{ij} gives an idea of how the value a_{ij} fits into the data in the surrounding matrix A. The mean squared residue score gives an indication of how the data is correlated in the submatrix, whether it has some coherence or it is random. A high value of H signifies that data is uncorrelated.

Cheng and Church proved that the problem of finding the largest square δ-bicluster ($|I| = |J|$) is NP-hard. Following the authors, I am thus interested in heuristics for finding a large δ-bicluster in the reasonable time.

Explanation of the MSR:

In the case of the perfect bicluster, follow the definition I have:

$$a_{ij} = \mu + \alpha_i + \beta_j.$$

So,

$$a_{iJ} = \frac{1}{|J|}\sum_{j\in J} a_{ij} = \frac{1}{|J|}(|J| \times \mu + |J| \times \alpha_i + \sum_{j\in J}\beta_j) = \mu + \alpha_i + \frac{1}{|J|}\sum_{j\in J}\beta_j =$$

$$= \mu + \alpha_i + \frac{1}{|J|}\beta,$$

$$a_{Ij} = \frac{1}{|I|}\sum_{i\in I} a_{ij} = \frac{1}{|I|}(|I| \times \mu + |I| \times \beta_j + \sum_{i\in I}\alpha_i) = \mu + \beta_j + \frac{1}{|I|}\sum_{i\in I}\alpha_i =$$

$$= \mu + \beta_j + \frac{1}{|I|}\alpha,$$

$$a_{IJ} = \frac{1}{|I| \times |J|}\sum_{j\in J, i\in I} a_{ij} = \mu + \frac{1}{|I| \times |J|}|I|\sum_{j\in J}\beta_j + \frac{1}{|I| \times |J|}|J|\sum_{i\in I}\alpha_i =$$

$$= \mu + \frac{1}{|J|}\beta + \frac{1}{|I|}\alpha,$$

where $\alpha = \sum_{i\in I}\alpha_i$ is a constant and $\beta = \sum_{j\in J}\beta_j$ is a constant. Such that:

$$H(I,J) = \frac{1}{|I| \times |J|}\sum_{i\in I, j\in J}(a_{ij} - a_{Ij} - a_{iJ} + a_{IJ})^2 =$$

$$= \frac{1}{|I| \times |J|}\sum_{i\in I, j\in J}((\mu + \alpha_i + \beta_j) - (\mu + \beta_j + \frac{1}{|I|}\alpha) - (\mu + \alpha_i + \frac{1}{|J|}\beta) +$$

$$+(\mu + \frac{1}{|J|}\beta + \frac{1}{|I|}\alpha))^2 = 0.$$

In the case of the error $\epsilon_i j$ I have:

$$a_{ij} = \mu + \alpha_i + \beta_j + \epsilon_{ij},$$

$$a_{iJ} = \mu + \alpha_i + \frac{1}{|J|}\beta + \frac{1}{|J|}\sum_{j \in J} \epsilon_{ij},$$

$$a_{Ij} = \mu + \beta_j + \frac{1}{|I|}\alpha + \frac{1}{|I|}\sum_{i \in I} \epsilon_{ij},$$

$$a_{IJ} = \mu + \frac{1}{|J|}\beta + \frac{1}{|I|}\alpha + \frac{1}{|I| \times |J|}\sum_{j \in J, i \in I} \epsilon_{ij}.$$

In such case I have the value of MSR:

$$H(I,J) = \frac{1}{|I| \times |J|}\sum_{i \in I, j \in J} (a_{ij} - a_{Ij} - a_{iJ} + a_{IJ})^2 =$$

$$= \frac{1}{|I| \times |J|}\sum_{i \in I, j \in J} (\epsilon_{ij} - \frac{1}{|I|}\sum_{i \in I} \epsilon_{ij} - \frac{1}{|J|}\sum_{j \in J} \epsilon_{ij} + \frac{1}{|I| \times |J|}\sum_{j \in J, i \in I} \epsilon_{ij})^2 =$$

$$= \frac{1}{|I| \times |J|}\sum_{i \in I, j \in J} (\epsilon_{ij} - \frac{1}{|I|}\epsilon_{Ij} - \frac{1}{|J|}\epsilon_{iJ} + \frac{1}{|I| \times |J|}\epsilon_{IJ})^2.$$

My aim is to find biclusters with minimal value of MSR.

The model, proposed by Cheng and Church is an additive model of the biclusters with coherent values. Generally, the biclusters of three major classes can be bound (See Fig. 3.1):

- Bicluster with constant values (see 3.1 (a)), where $a_{ij} = \mu$,
- Bicluster with constant values on rows (see 3.1 (b)), $a_{ij} = \mu + \alpha_i$, or columns (see 3.1 (c)), $a_{ij} = \mu + \beta_j$,
- Bicluster with coherent values (see 3.1 (d, e)):

a) bicluster with constant values

1	1	1	1
1	1	1	1
1	1	1	1
1	1	1	1

b) bicluster with constant values on rows

1	1	1	1
2	2	2	2
3	3	3	3
4	4	4	4

c) bicluster with constant values on columns

1	2	3	4
1	2	3	4
1	2	3	4
1	2	3	4

d) bicluster with coherent values (additive)

a1+b1	a2+b1	a3+b1	a4+b1
a1+b2	a2+b2	a3+b2	a4+b2
a1+b3	a2+b3	a3+b3	a4+b3
a1+b4	a2+b4	a3+b4	a4+b4

e) bicluster with cpherent values (multiplicative)

a1xb1	a2xb1	a3xb1	a4xb1
a1xb2	a2xb2	a3xb2	a4xb2
a1xb3	a2xb3	a3xb3	a4xb3
a1xb4	a2xb4	a3xb4	a4xb4

a)Bicluster with constant values b-c) Bicluster with constant values on rows/columns d-e) Bicluster with coherent values: additive/multiplicative models.

Figure 3.1: Five different bicluster models.

- ✧ additive model: $a_{ij} = \mu + \alpha_i + \beta_j$,
- ✧ multiplicative model: $a_{ij} = \mu \times \alpha_i \times \beta_j$,

Chapter 4

The Combinatorial model

4.1 Definition of the Difference Matrix

The main idea of the combinatorial model comes from the definition of perfect bicluster. So a perfect bicluster $I \times J$ is defined as a subset of rows and a subset of columns, whose values a_{ij} are predicted using the following expression:

$$a_{ij} = \mu + \alpha_i + \beta_j$$

where μ is the typical value within the bicluster, α_i is the adjustment for row i and β_j is the adjustment for row j.

Given the data matrix A, it can be defined a matrix $G(k)$ as the difference between the k-th row and all the others. In particular, an entry of such a matrix reads $g_{ij}(k) = a_{kj} - a_{ij}$ where $k = 1, ..., N-1$, $i > k$ and $j = 1, ..., M$. It worth stressing that in case of a perfect bicluster, all $G(k)$ will have constant rows.

It is possible to combine all $G(k)$ by rows into a new matrix T whose entries t_{mj} are defined as follows:

$$t_{mj} \equiv a_{kj} - a_{hj} \tag{4.1}$$

where $k = 1, ..., N-1$, $m = l + i$, $h = k + i$, $i = 1, ..., N - k$ and

$$l = \begin{cases} 0 & k = 1 \\ \sum_{t=1}^{k-1}(N-t) & k > 1 \end{cases} \tag{4.2}$$

Explanation:

- *k=1,. . . ,N-1*
- *h=k,. . . ,N = (1,. . . ,N-k)+k = i + k, i=1,. . . ,N-k*
- *k=1 => m=1,. . . ,N-1*
- *k=2 => m=(1,. . . ,N-2)+(N-1)*
- *k=3 => m=(1,. . . ,N-3)+(N-1)+(N-2)*
- *k=4 => m=(1,. . . ,N-4)+(N-1)+(N-2)+(N-3) => ...=> m=(1,. . . ,N-k)+$\sum_{t=1}^{t=k-1}$(N-t)=i+$\sum_{t=1}^{t=k-1}$(N-t)*

Of course, starting from the indexes of a particular entry t_{mj} it is possible to trace back the corresponding entries of A that yield such a difference.

4.2 Analysis of the Difference Matrix

4.2.1 The pre-combinatorial matrix obtaining. Case of the perfect biclusters.

Starting from the matrix T defined as above, I aim to construct a binary matrix C which contains the information about the location of a certain number of entries ($\geq u_{min}$) belonging to the same row and sharing the same value. For the time being, I assume two entries being *equivalent* when they are exactly identical. As a further refinement of the algorithm, I will later discuss the case when the equivalence is admitted also for entries whose difference is smaller than a fixed error. This idea implements the concept of noise affecting the experimental results reported in the entries of the data matrix.

Let me consider the m-th row of T. There are three possible cases: (i) on the row there are no u_{min} entries with equivalent values; (ii) there are q_m groups, each of them with at least u_{min} entries with equivalent values; (iii) case of the overlapped biclusters. In the first case, to the row of T corresponds one row in C only with all vanishing entries. In the second case, to each set of entries sharing the same value corresponds a row in C which has unitary values on the columns corresponding to the elements of the group, and vanishing all the other entries. In this way, from the single row of T one gets q_m rows in C. This procedure is applied to all the rows of T.

1. Case of non overlapped biclusters, simple case (see 4.1). For all the

rows of the matrix T I find the constants with equal values. In this case it is easy to see that for every difference of two rows it will be only 1 group of the constants. I obtain matrix C , that have the same dimension, as the matrix T. C has ones on the places of each constant of T and zeros in the other cases.

The algorithm of this procedure can be seen following:

for *i = 1* to *end*

for *j = 1* to *end*

if *T(i, j) = constant* so *Z(i, j) = 1*

else *Z(i, j) = 0*

end end

2. Case of non overlapped biclusters, complex case (see 4.2). It is the most difficult case. Again, for all the rows of the matrix T I find all constants with equal values. In this case it can be q_i groups of constants in every row i of the matrix T. I obtain the matrix C, with number of the rows larger than T. From every row of T with q_i groups of constants I create q_i rows of C. Every of q_i rows of C has ones on the position of the specific group of the constants and 0 in other case.

   The algorithm of this procedure can be seen following:

   for *i = 1* to *end*

   for *j = 1* to *end*

   find $\{c_k\}_{k=1}^{q(j)}$ groups of constants

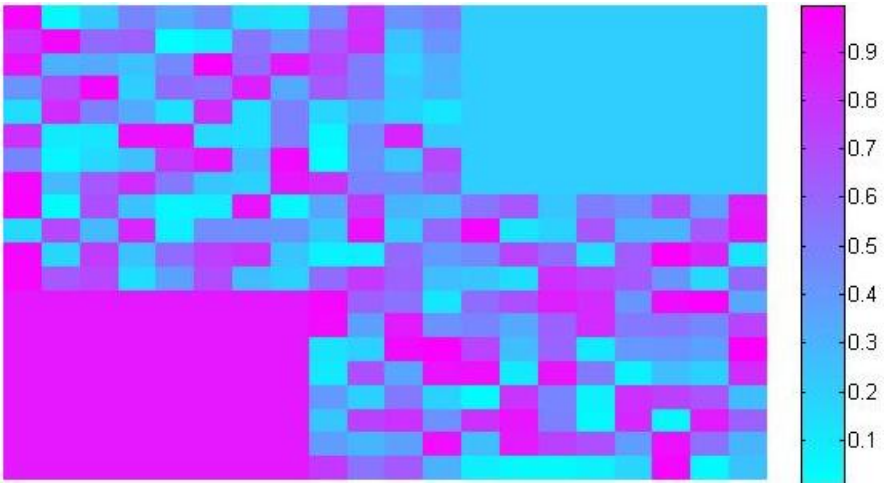


Figure 4.1: Case of non overlapped biclusters, simple case

end_groups = 1

for *k = 1* to *q(j)*

if *T(i,j) =* c_k *Z(end_groups+k - 1, j) = 1*

else *Z(end_groups+k - 1, j) = 0*

end end

end_groups = end_groups+q(j)

end end

3. Case of overlapped biclusters (see 4.3). This case is similar to the first one. For all the rows of the matrix TI find the constants with equal values and obtain the matrix C , that have the same dimension, as the matrix T. Matrix C has ones on the places of the constants of T and zeros in other cases.

The matrix CI name *pre-Combinatorial*.

To the binary matrix C it can be applied Bimax algorithm which allows to

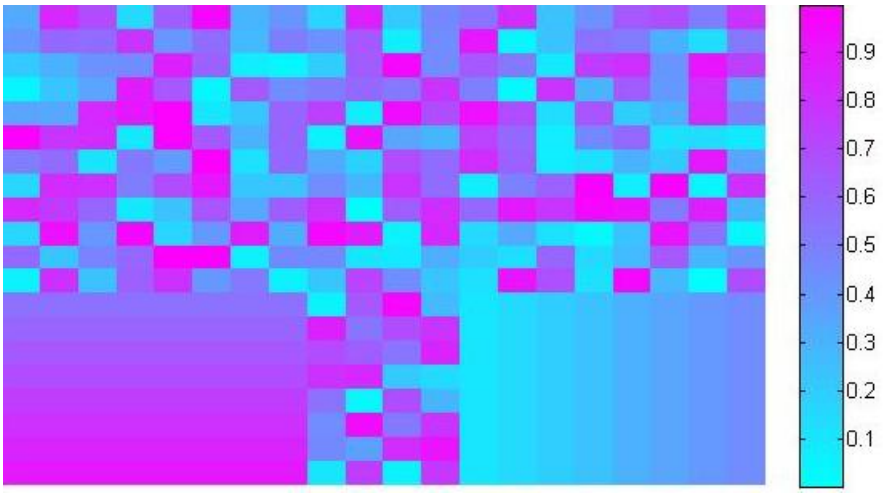


Figure 4.2: Case of non overlapped biclusters, complex case

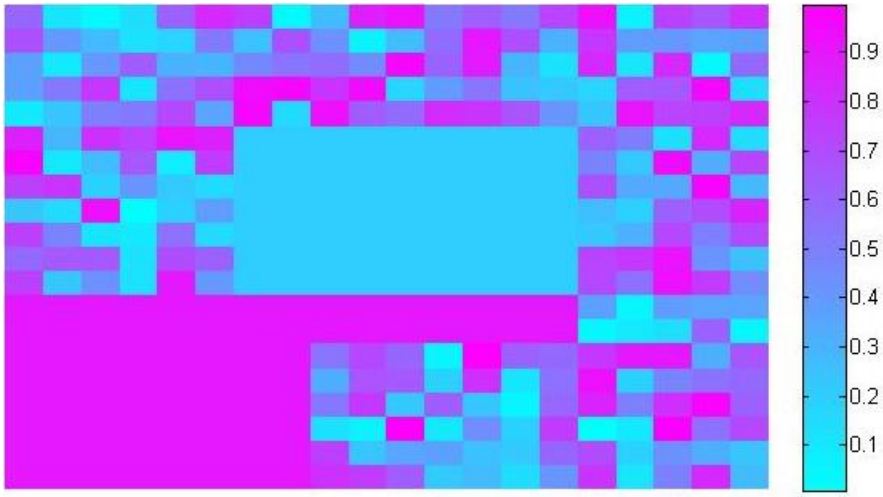


Figure 4.3: Case of overlapped biclusters

find submatrices of unitary entries[11]. Once such submatrices have been determined, I can trace back for each of them the corresponding elements of data matrix A. Such sets will provide the final desired biclusters.

So far, I considered the case where no noise affects the values of the entries of A. Since this case can be unrealistic when dealing with experimental data, I have to extend the definition of *equivalence* to the general case where values are considered to be equal within an error defined as follows.

4.2.2 Combinatorial matrix. Error definition and initial conditions.

For $i = 2, ..., N$ and $j = 2, ..., M$ (where N and M are the number of rows and of columns of A, respectively), I randomly extract a set S of submatrices A^s_{PQ} where $|P| = i$ and $|Q| = j$. Let

$$r^s_q \equiv \max_{p \in P} a^s_{pq} - \min_{p \in P} a^s_{pq}$$

be the range of the column q for each $q \in Q$ and

$$r^s \equiv \max_{q \in Q} r^s_q - \min_{q \in Q} r^s_q$$

the error of the random bicluster s. It is worth stressing that for a perfect bicluster r^s is vanishing. I consider as maximum tolerable error for a bicluster of dimension $i \times j$ the extreme of this null random distribution,

namely

$$\epsilon_{ij} \equiv \min_{s \in S} r^s$$

In this step I also define the minimum number of columns u_{min} and of rows v_{min} for the biclusters.

The matrix, which meets all these requirements I call Combinatorial C_{comb}. The matrix C_{comb} now can be analysed by different ways. So I consider some of them.

4.2.3 Cost of the initialization

Let me fix the values P and Q and extract the matrix A_{PQ}. For every of Q columns I find the value of $\epsilon_{pq} = max_{p \in P}(a_{pq}) - min_{p \in P}(a_{pq})$ of cost $O(Q \times P)$. I continue such a procedure for K random matrices A^k_{PQ}. So the cost of such a calculation is $O(K \times Q \times P)$. Now I release the values of P and Q, so: $P = 1, \ldots, N$ and $Q = 1, \ldots, M$ and obtain the initialization cost Ic:

$$Ic = O(\sum_{P=1}^{N} \sum_{Q=1}^{M} K \times P \times Q) = O(K \times \sum_{P=1}^{N} P \times \sum_{Q=1}^{M} Q) = O(K \times \frac{N(N+1)}{2} \times \frac{M(M+1)}{2}).$$

4.3 Part I.

4.3.1 Biclique

Definition: Given a bipartite graph $B = (V_1 \cup V_2, E)$, a biclique $C = U_1 \cup U_2$ is a subset of the node set, such that $U_1 \subseteq V_1$, $U_2 \subseteq V_2$ and for every $u \in U_1, v \in U_2$ the edge $(u, v) \in E$. In other words, a biclique is a complete bipartite subgraph of B (See Fig. 4.1). Maximum edge cardinality biclique (MBP) in B is a bicliquie C with a maximum number of edges. In ad edge weighted bipartite graph B, there is a weight w_{uv} associated with each edge (u, v). A maximum edge weight (MWBP) biclique is a bicliquie C, where the sum of the edge weights in the subgraph induced by C is maximum.

Such, a biclique [17] is defined to be a sub-graph of the bipartite graph where all the nodes are connected. A graph can have many bicliques, however a maximal biclique is defined to be a biclique that cannot be extend any further. In other words, that biclique cannot be a sub-graph of an even larger biclique. The largest maximal biclique is the maximal biclique with the largest number of nodes.

Some known results for related problems. The maximum node weight bicliquie problem is polynomially solvable [25]. (In a node weighted bipartite graph B, there is a weight w_v associated with each node v.) Hence, the maximum node cardinality biclique problem is also polynomially solvable. A restricted version of these problems, where there is an additional

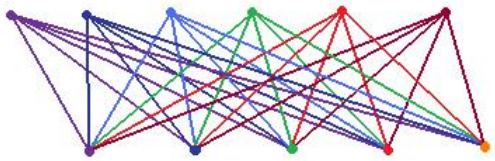


Figure 4.4: Complete bipartite graph.

requirement that $|U_1| = |U_2|$, is called the maximum balanced node cardinality bicliquie problem (MBBP). MBBP is shown to be NP-complete.

Theorem: MWBP is NP-complete [26].

Proof: Let $G = (V, E)$ be a graph with node set V end edge set E. Create a bipartite graph $B(G) = (V_1 \cup V_2, E')$ from G, such that $V_1 = V_2 = V$ and $(i, j) \in E'$ (for $i \in V_1$ and $j \in V_2$) if and only if $i = j$ or $(i, j) \in E$. Let the edges (i, i) of $B(G)$ have weight 1 and all the other edges have weight zero.

With the edge weights as defined, there is a maximum weight biclique $U_1 \cup U_2$ in $B(G)$, such that $i \in U_1$ if and only if $i \in U_2$ (i.e. $|U_1| = |U_2|$ and the biclique is "symmetric"). Such a maximum weight "symmetric" biclique can be obtained easly by deleting the nodes $i \in U_1, i$ do not $\in U_2$ and $i \in U_2, i$ do not $\in U_1$ from a maximum weight biclique. It follows that if C is a maximum clique in G, then $U_1 \cup U_2$, where $U_1 = U_2 = C$, induces a maximum weight biclique in $B(G)$. Similarly, if $U_1 \cup U_2$ is a symmetric maximum weight biclique in $B(C)$, then $C = U_1 = U_2$ is a masimum clique in G.

The Maximal Biclique Generation Problem (MBGP) consistes in generating all the maximal bicliques of a given graph. The MBGP cannot be

solved in polynomial time with respect to the input size, since the size of the output can be exponentially large.

The consensus algorithm [18].

Definition:

Let G be a graph and let X, Y be two disjoint non-empty subsets of the vertex set, with the property that every vertex in X is adjacent to every vertex in Y. The biclique of G having the bipartition sets X and Y will be denoted by (X, Y). (Note that $(X, Y) = (Y, X)$). Let $B_1 = (X_1, Y_1)$ and $B_2 = (X_2, Y_2)$be two bicliques of G. In such a case B_1 absorbs or contains B_2 if $X_2 \subseteq X_1$ and $Y_2 \subseteq Y_1$, or if $X_2 \subseteq Y_1$ and $Y_2 \subseteq X_1$.

Definition:

If $Y_1 \cap Y_2 \neq 0$,Icall $(X_1 \cup X_2, Y_1 \cap Y_2)$ one of the consensuses of B_1 and B_2. Similarly, each of those pairs of subsets $(X_1 \cap X_2, Y_1 \cup Y_2)$, $(Y_1 \cup X_2, X_1 \cap Y_2)$, $(X_1 \cup Y_2, Y_1 \cap X_2)$ which define bicliques (i.e. which involve two non-empty subsets) are consensuses of B_1 and B_2. In this way, a pair of bicliques may have 0, 1, 2, 3, or 4 consensuses.

A consensus approach to MBGP starts with a collection of C of bicliques which covers the edge set of a graph G. Such a collection is easily available, for instance by simply considering all the individual edges of the graph, viewed as bicliques. A similar straightforward way of obtaining C is to define it as the collection of all the stars centered in the vertices of the graph G.

Using the above terminology I can now define a consensus algorithm as a sequence of transformations on the collection C. The method applies repeatedly two transformations, the absorprion and the consensus adjunction - described below - and stops when none of these steps can be applied.

- *Absorption*: If the biclique B_1 in C absorbs the biclique B_2 in C, then remove B_2 from C.
- *Consensus adjunction*: For any two bicliques B_1 and B_2 in C, if any of the consensuses of B_1 and B_2 exists and is not absorbed by a biclique already in C, then add it to C.

Two trivial observations are in order. First, if the collection C covers the edge set (i.e. every edge of G is contained in at least one of the bicliques of C), then this property will be observed by both of the transformations above will always produce collections consisting only of bicliques of G.

The validity of the consensus approach is based on the following result:

Theorem:

If C is a collection of bicliques of the graph G which covers the edge set of G, and if C' is the collection of bicliques obtained from C by repeating the transformations in the consensus algorithm described above as many times as possible, then C' consists of all the maximal bicliques of G.

In this work the method of G. Alexe et al. [18], that generates all maximal bicliques (i.e. complete bipartite, not necessarily induced subgraphs) of

a graph was used. The algorithm is inspired by, and is quite similar to, the consensus method used in propositional logic. The total complexity of algorithm is polynomial in the input size, and only linear in the output size.

In this step I have all possible biclusters and the final step is to find biclusters that are different between them.

4.3.2 Sorting & Deleting algorithm

In this method I sort rows and columns such as in the left top of the matrix I have the max possible number of ones, and delete the rows and columns whose sum of ones is smaller than desired value. Then I continue this process until I have number of zeros greater than a threshold (practically I want a submatrix with only ones).

4.3.3 Results

I apply these methods to two simulated data sets and one real NCI60 genome data (see 4.5):

- the simple matrix 7×6, that includes 2 biclusters 3×4 and 4×4
- the matrix 50×200, that contained 3 biclusters such that:

$$bicluster_1(15 \times 100) = \frac{j}{100}, \textit{j}\text{=1:100};$$

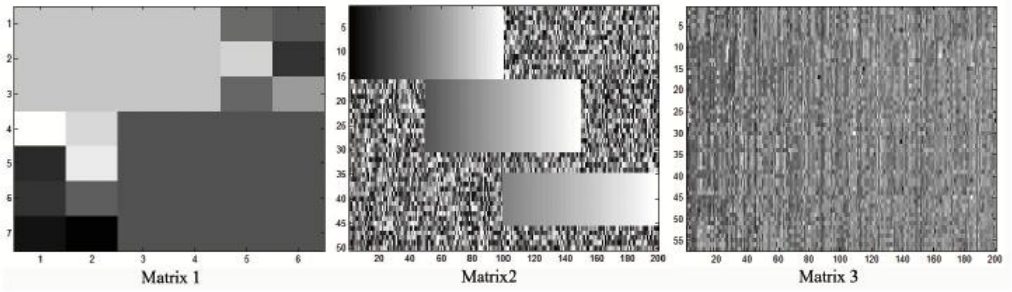


Figure 4.5: The data matrices.

$$bicluster_2(14 \times 100) = \frac{2j}{300},\ j\text{=50:150};$$

$$bicluster_3(10 \times 100) = \frac{3j}{600},\ j\text{=100:200};$$

- the real data set NCI60, that contains the processed version of cDNA microarrays used to examine the variation in gene expression among the 57 cell lines from the National Cancer Institute's (NCI60) anti-cancer drug screen. The matrix consists of 57 samples on 200 features. The dataset has been standardized to mean zero and variance 1.

I analyze these three matrices by using the following algorithms: Lazzeroni and Owen (Plaid), Bimax, Cheng and Church (CC), Spectral (all these algorithms by using the package R) and my algorithm Combinatorial. The validation of biclusters has been done by using 3 different methods:

- The value of mse, that was described before.
- The *a-priory* information on the data or the GO term data base information which is useful to identify if some agglomeration of genes

in a cluster is significant with respect to a specific annotation [19]. I analyze the biclusters relatively to genes (rows), and consider them as clusters. Specifically, to formalize this characteristic, I define the indicator S as: for each annotation i, for each cluster j

$$S_{ij} = \frac{a_{ij}}{a_{ij} + b_{ij}} \frac{A_i + B_i}{A_i}$$

where a_{ij} is the number of positive annotations in the cluster j, b_{ij} is the number of negative annotations in the cluster j, $A_i = \sum_j a_{ij}$ and $B_i = \sum_j b_{ij}$.

- Fisher's exact test is a statistical significance test used in the analysis of contingency tables where sample sizes are small [20]. The hypergeometric distribution is used to model the probability of observing at least k objects from a cluster of n objects by chance in a category containing f objects from a total database size of g objects. The P-value is given by:

$$P = \frac{\frac{f!}{k!(f-k)!} \frac{(g-f)!}{(n-k)!(g-f-n+k)!}}{\frac{g!}{n!(g-n)!}}.$$

  a significant P-value for a cluster is smaller than 0.01.

The results are shown below.

- *Matrix 7×6* How can be seen from the Table 4.1. the best separation is done by Combinatorial algorithm that gives 2 biclusters with error that equals zero; Plaid algorithm does not give any result; Bimax algorithm finds one of the two biclusters; CC algorithm with

δ=0.01 finds 25 biclusters, two of them are significant (mse equals to 0 and 0.0073, respectively); Spectral algorithm finds 25 biclusters with two significant ones with mse equals to 0.0043 and 0.017, respectively .

- *Matrix 50$\times$200 random* For Combinatorial model specify error equal zero. So I find 3 perfect biclusters contained in my matrix. Plaid algorithm finds only one perfect bicluster. Bimax algorithm finds 25 biclusters and two of them are perfect. CC algorithm does not give any result. Spectral algorithm finds 25 biclusters, one of them is perfect and 2 of them are significant (See Table 4.2.).

- *NCI60* For NCI60 that contains 8 biclusters, Combinatorial algorithm finds 5 perfect biclusters and one significant with a very small P-value (0.0115); Plaid algorithm finds 3 perfect biclusters and one significant with P-value 0.0008; Bimax finds two perfect clusters; CC does not give any result; Spectral algorithm finds 2 biclusters with P-value equals to 0.0082 and 0.0012, respectively (See Table 4.3.). In the Table 4.3. the value " - " shows that the bicluster was not found.

4.3.4 Conclusion

As shown by the experiments, Combinatorial algorithm gives always better and more accurate results than the other algorithms, because it reaches the maximal precision in the data sets analysis. In every experiment

Table 4.1: Results of the analysis on Matrix 1 data.

	Theoret	**Plaid**	**Bimax**	**CC**	**Spectral**	**CBA**
num. of bicl	2	-	1	25	25	2
dim1	3× 4	-	3× 4	3× 3	5× 4	3× 4
dim2	4× 4	-	-	6× 4	5×5	4× 4
msr1	0	-	0	0	0.004	0
msr2	0	-	-	0.01	0.02	0
P-value1	0.03	-	0.03	0.11	0.57	0.03
P-value2	0.03	-	-	0.57	0.57	0.03
Enrich1	2.33	-	2.33	1.75	1.05	2.33
Enrich2	1.75	-	-	1.17	1.05	1.75

Table 4.2: Results of the analysis on Matrix 2 data.

	Theoret	**Plaid**	**Bimax**	**CC**	**Spectral**	**CBA**
N of bic	3	1	25	1	25	3
dim1	15× 100	15× 54	15× 60	50× 200	15× 44	15× 100
dim2	14× 100	-	11×80	-	18× 44	14× 100
dim3	10× 100	-	-	-	8× 44	10×100
msr1	0	0	0	0.07	0.05	0
msr2	0	-	0	-	0.07	0
msr3	0	-	-	-	0.05	0
P-value1	0	0	0	1	0	0
P-value2	0	-	0	-	0.0005	0
P-value3	0	-	-	-	0.01	0
Enrich1	2.73	3	2.72	-	2.73	2.73
Enrich2	2.73	-	3.73	-	1.88	2.73
Enrich3	3.73	-	-	-	2.32	3.73

I *a-priori* decided the maximal error and the minimal dimension of the desired biclusters. In the case of NCI60 data set I used the half of the conditions for separating the genes.

The next step (see Part II) of my research was to use this algorithm to separate overlapped biclusters and to find the best methods to analyze the Difference Matrix.

4.4 Part II

As can be seen in Part I Combinatorial Algorithm permits to control the error of biclusters in every step, specifying this error from the begin and to define the dimensions of the desired biclusters. But it has some difficulties: the possibility to find overlapped biclusters, analysis of the Difference Matrix and defining of the initial conditions. In this part I solve these problems and then taste the algorithm in the real biological data: Gastric cancers (GC) with defective mismatch repair (MMR) comprise 10–25\% of all GC. These tumors accumulate DNA replication errors at short-repeat sequences that are identified by the presence of microsatellite instability (MSI) [21]. The objective of my study was to determine if and how MSI phenotype in GC could be distinguished from the microsatellite stable (MSS) phenotype using microarrays.

As first, I need some technique that permits to find submatrices of ones for my matrix $Comb_{bin}$. In this part I use a *Bimax algorithm*. [11] After finding the matrices of ones I turn to my initial matrix and extract the rows and columns that give these submatrices of ones. In this step I have final desired biclusters.

4.4.1 Reference method Bimax

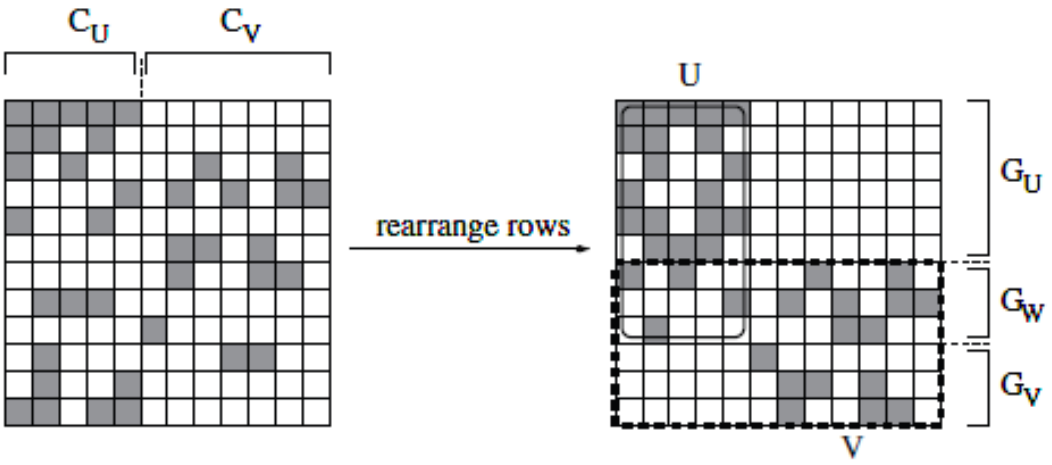
Bimax uses a simple data model, reflecting the fundamental idea of biclustering, and allows to determine all optimal biclusters in a reasonable time [11].

Model. The model assumes two possible expression levels for a gene: without changes and with changes with respect to a control experiment. Let a set of *m* experiments and *n* genes can be represented by a binary matrix $E^{n \times m}$, where a cell e_{ij} is 1 if a gene i responds to the condition j and it is 0 otherwise it. A bicluster (G, C) corresponds to a subset of genes $G \subseteq 1, ..., n$ that jointly respond to a subset of conditions $C \subseteq 1, ..., m$. The pair (G, C) defines a submatrix of E with all the elements equal to one. I would like to find all maximal biclusters.

DEFINITION 1. *The pair* $(G, C) \in 2^{\{1,...,n\}} \times 2^{\{1,...,m\}}$ *is called an inclusion-maximal bicluster if and only if (1)* $\forall i \in G, j \in C : e_{ij} = 1$ *and (2) do not* $\exists (G', C') \in 2^{\{1,...,n\}} \times 2^{\{1,...,m\}}$ *with (a)* $\forall i' \in G', j' \in C' : e_{i'j'} = 1$ *and (b)* $G \subseteq G' \wedge C \subseteq C' \wedge (G', C') \neq (G, C)$.

Algorithm Bimax is a binary inclusion-maximal biclustering algorithm, a fast divide-and-conquer approach, that requires much less memory resources than many other algorithms. Its running-time complexity is $(O(nm\beta$ $min\{n, m\}))$, where β is the number of all inclusion-maximal biclusters in data matrix,n and m are the binary matrix dimensions. This algorithm provides a worst-case running-time complexity for matrices that contain disjoint biclusters. The complete algorithm and the proof of the general upper bound for the running-time complexity can be found in the Supplementary Material [11].

Bimax tries to identify areas of E that contain only 0s and therefore can be excluded from further inspection. The idea of Bimax algorithm, which is illustrated in 4.6, is to divide E into three submatrices, one of which



(1) divide the input matrix into two smaller, possibly overlapping submatrices U and V; (2) divide the set of columns into two subsets C_U and C_V, by taking the first row as a template; (3) resort the rows of E : first all genes that respond only to conditions given by C_U, then those genes that respond to conditions in C_U and in C_V and finally the genes that respond to conditions in C_V only. Figure is taken from [11].

Figure 4.6: Illustration of the Bimax algorithm.

contains only 0-cells and therefore can be disregarded. Then the algorithm is applied to the remaining two submatrices U and V; the recursion ends if the current matrix represents a bicluster, i.e. contains only 1s. If U and V do not share any rows and columns of E, i.e. G_W is empty, the two matrices can be processed independently from each other.

4.4.2 Final algorithm

The algorithm to be applied to A can be summarized as follows:

1. Define the initial conditions, namely u_{min}, v_{min} and ϵ_{ij}

2. Construct the difference matrix T

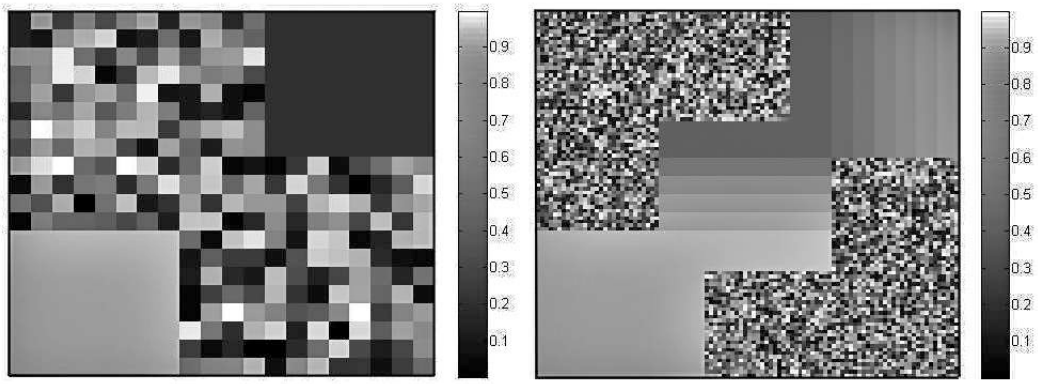


Figure 4.7: The simulated data matrices

3. Transform T in the corresponding binary matrix C, using $\epsilon_{v_{min}u_{min}}$ as error to define the equivalence between two entries

4. Analyze C with the Bimax algorithm

5. Trace back the resulting biclusters on A

6. Filter out all the biclusters of dimension $i \times j$ smaller than $v_{min} \times u_{min}$ or having an error greater than ϵ_{ij}

4.4.3 Results

Simulated data, matrix 20×20

First, I apply the algorithms to a simulated data. I create a very simple matrix 20×20 with random values from the interval (0, 1). The data matrix has two biclusters 8×8 with constant values, that are 0.2 for the first (rows (13:20) and columns (13:20)) and 0.9 for the second (rows (1:8) and columns (1:8)) (See Fig. 4.7 left).

I apply Plaid and try to find the best parameters. After running Plaid more than 200 times, I find the best parameters: row.release = 0.3, col.release = 0.5. With these best values, Plaid algorithm finds one bicluster 6×5 with the value of MSR 0.0222 and Enrichment 1.6667.

I use Cheng and Church model with the following coefficients: δ = 0.001; α = 1.5, number of biclusters = 2. And get two different biclusters with cardinalities 6×6 and 7×8, values of MSR: 0 and 0, Enrichment: 2.5 and 2.5.

I apply the Spectral algorithm with the following values: number of eigenvalues = 3, min number of rows = 5, min number of columns = 5. I find 22 biclusters that can be classified into two groups with respect to the rows. One of these groups finds the first bicluster, but no group finds the second theoretical bicluster. The best result is the following: cardinality 9×9, MSR: 0.0423, Enrichment: 1.39.

SAMBA algorithm after many runs does not give any result. No parameters for creating some bicluster were found. It can be concluded that for little data sets SAMBA algorithm does not work well.

For CBA I use minimal number of rows and columns = 2, error threshold = 0. And find two perfect biclusters.

It can be seen that for little matrices with two non-overlapped perfect biclusters CBA works better than the other algorithms. Cheng and Church algorithm gives a good result, Plaid and Spectral algorithms find only one bicluster and SAMBA does not work at all (See Table 4.4).

Simulated data, matrix 100×100

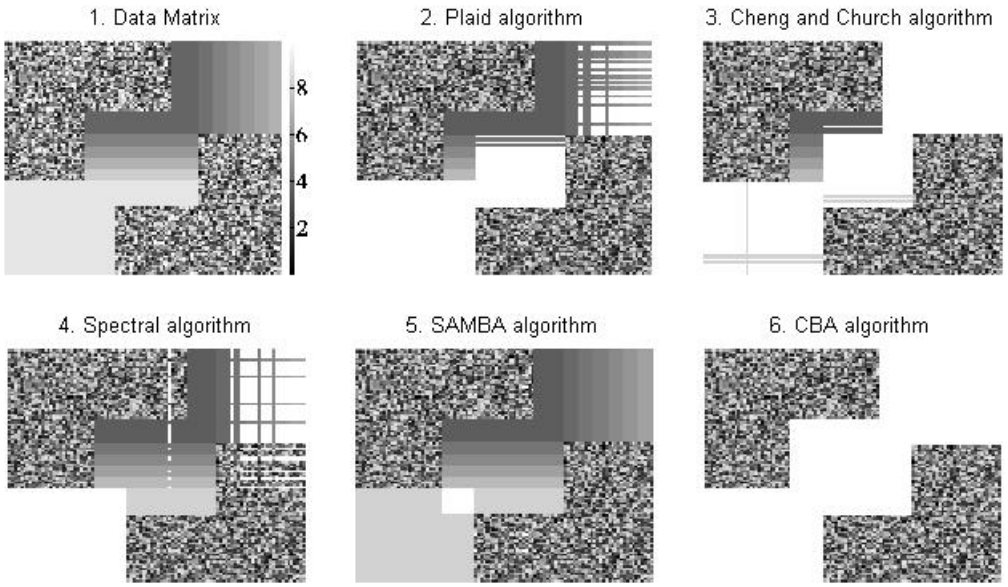
Now I apply the algorithms to larger simulated data. I created a matrix 100×100 with random values from the interval (0, 1). The data matrix has three overlapped biclusters 40×40, 41×41, 40×40 (See Fig. 4.7 right) with constant values on rows or columns. The first bicluster has the rows (1:40) and the columns (1:40) and its entries equal to 0.9. The second bicluster has the rows (30:70) and the columns (30:70) and has constant values for the rows. The third bicluster has the rows from (61:100) and columns from (61:100) and has constant values for the columns.

As before, I apply Plaid and try to find the best parameters. I run Plaid more than 50 times and find the best parameters: row.release = 0.4, col.release = 0.6. Plaid algorithm finds 6 biclusters, three of them are significant. As a result, Plaid model finds all 3 theoretical biclusters with cardinalities: 40×40, 24×21, 28×30.

I use Cheng and Church model with the following coefficients: δ = 0.001; α = 1.5, number of biclusters = 3. And it gets three different biclusters with cardinalities 37×39, 30×30 and 40×40, values of MSR: 0 , 0 and 0, Enrichment: 2.5, 2.44 and 2.5.

I apply the Spectral algorithm with the following values: number of eigenvalues = 1, min number of rows = 20, min number of columns = 20. I find two good biclusters, which have the following features: cardinalities 40×40 and 42×22, MSR: 0 and 0.0133, Enrichment: 2.5 and 2.15.

I use SAMBA with the following parameters: minimal number of genes:



Comparison of the results on the simulated 100×100 data of different algorithms. White colour shows the found biclusters for every considered algorithm.

Figure 4.8: Comparison of the results on the simulated 100×100 data.

6, minimal number of condition: 3. As a result five biclusters are found, and two of them are significant.

For the CBA I use minimal number of rows and columns = 10, error threshold = 0. And find three perfect biclusters.

It can be seen that also for larger matrices with three overlapped perfect biclusters, CBA works better than the other algorithms, Plaid and Cheng and Church algorithms find perfect biclusters with smaller cardinality, Spectral and SAMBA find two biclusters (See Table 4.5 and Fig. 4.8).

Figure 4.9: The heatmap of E.coli data matrix

Cellular Localization Sites of Proteins (E.coli)

I use Cellular Localization Sites of Proteins (E.coli) of Nakai and Kanehisa. This dataset contains 336 number of instances and 7 attributes (see Fig. 4.9).

Class Distribution can be seen in the Table 4.6. There are 8 classes with different number of elements. This data is available on

http://archive.ics.uci.edu/ml/machine-learning-databases/ecoli/

It can be seen that two classes are very small (of 2 elements), and they can not be found by the biclustering techniques. That is why I cancel them. As a result I have only 6 classes.

After running Plaid algorithm 50 times, I obtain the parameters: row.release

= 0.4, col.release = 0.6. Plaid algorithm finds 2 biclusters with following properties: cardinality 110×2 and 42×2, MSR is 0 and 0 - it is clear because the biclusters have only 2 columns. Plaid model finds the classes of im+imU, om+pp. If I analyze their enrichment respect to 2 similar theoretic classes im+imU, om+pp I have: 2.0933 and 1.958.

I apply Cheng and Church model with the follow coefficients: δ = 0.001; α = 1.5, number of biclusters = 6. And get 6 different bicluster, but only 2 of them are significant with the cardinality 40×5 and 39×6, MSR: 0.0023 and 0.0034. Cheng and Church model finds the follow classes: cp and im+imU, so the Enrichment respect to the same theoretic classes is: 2.2825 and 2.2894.

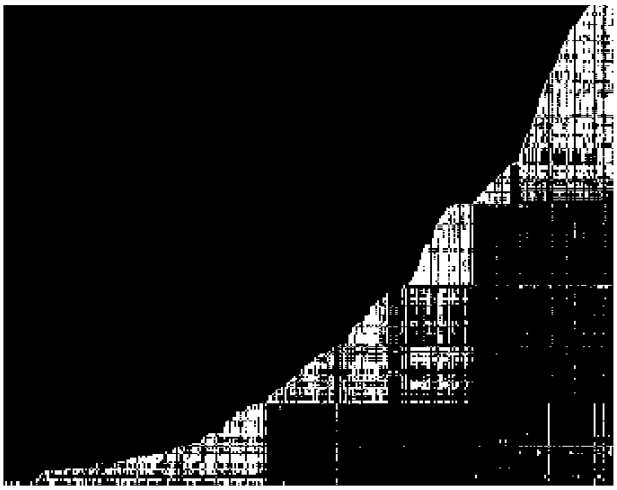
I run Spectral algorithm with the follow values: number of eigenvalues = 3, min number of rows = 2, min number of columns = 2. As a result, I receive 4 biclusters, 2 of them are not overlapped, but only one is significant. It has the follow parameters: cardinality 12×5, MSR 0.0066, and finds only part of class om with Enrichment 2.2.

I apply SAMBA algorithm many times with different conditions. As a result, SAMBA detect biclusters with only two conditions. The best initial conditions are: permitted overlap between two biclusters: 0.1, minimal number of genes: 10, minimal number of conditions: 1. I receive 9 biclusters with all the MSR equal to 0 because of only 1 and 2 columns and only 5 of them are significant. These biclusters separate 3 groups of the genes: cp with the best Enrichment 1.9091, im with the best Enrichment 1.9636 and omL with Enrichment 33.6.

For the CBA I use minimal number of rows = 10, minimal number of columns = 6, error threshold = 0.3. In the first step I find 4500 biclusters, but it can be seen that they are very overlapped and I can distinguish five non much overlapped groups (see Fig. 4.10). In the Fig. 4.10 for the abscissa I put the genes, for the ordinate biclusters. In this step the algorithm merges biclusters: (1:144)×(1:833), (145:217)×(834:1988), (224:260)×(1989:2773), (261:279)×(2774:3130) and (280:336)×(3100:4500). As a result I find 5 classes, they are cp, im, imU, om, omL+pp with cardinalities 115×6, 56×6, 46×6, 16×6 and 48×6. It can be seen that the Enrichment with respect to the theoretical classes is high, cardinality is good. So I find a very good separation. The results for all the algorithms are shown in Table 4.7.I note that the values of Enrichment for the biclusters are calculated with respect to their own classes (see the row "names bic").

Annotations

- C&C - Cheng and Church,
- Spect - Spectral
- CBA - Combinatorial
- Theor - description of the data matrix
- N of bic - total number of founded biclusters,
- no overl - number of biclusters, that are do not overlapped much (less than 10% of overlap),



This figure shows relation between the biclusters and rows of E.coli data: for abscissa all rows of a data matrix are presented, for ordinate - all biclusters, if a point of the heatmap is white so the row enter in bicluster, if it is black - no.

Figure 4.10: Relation between the biclusters and rows of E.coli data.

- the best - number of the best biclusters,
- En - Enrichment of the bicluster n (n=1:6),
- dim$n$ - dimension of the bicluster n (n=1:6),
- bic$n$ - name of the gene n (n=1:6),
- MSRn - MSR of the bicluster n (n=1:6),
- row/col.release - is a scalar in [0,1](with interval recommended [0.5-0.7]) used as threshold to prune rows/columns in the layers depending on row homogeneity (it was used for the Plaid model)
- δ and α - are the maximum of accepted score and the scaling factor for the C&C model.

Gastric Cancer data.

Data Definition

Biotinylated cRNA targets were synthesized from each sample and hybridized to Affymetrix oligonucleotide chips (GeneChip HG-U133A/B) that contain 45,000 probe sets (39,000 unique transcripts and 33,000 well-substantiated human genes). The full data set was normalized according to the invariant set method. The expression data are available at GEO (Gene Expression Omnibus) public data bank http://www.ncbi.nlm.nih.gov/geo/.

Analysis of the data.

The tissues were collected from a series of GC cases identified in an area around Florence (Italy) characterized by high GC risk in the period 2000-2005.I find different biclusters and validate them (see Methods for details on data simulation). After data selection I receive 82 genes, 31 normal and 38 tumoral (19 MSS and 19 MSI) tissues.

Case of all data.

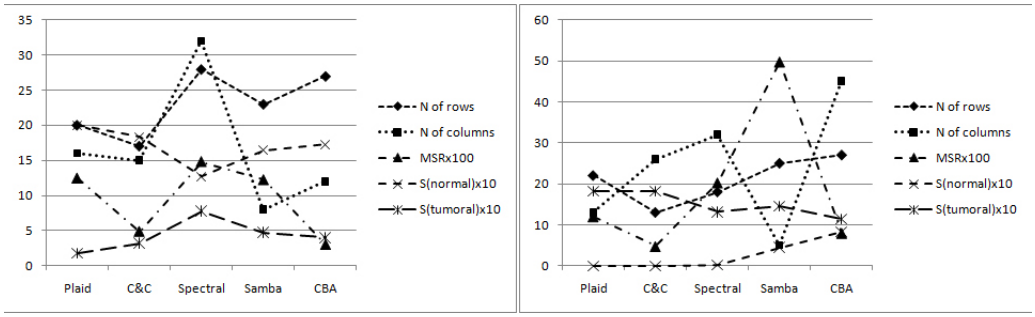
First, I analyze all data set that consist of 69 normal/tumor tissues and try to find possible separation.

The Plaid algorithm I apply with the following data: row.release = 0.1, col.release = 0.3. As a result I obtain 5 biclusters. Two of them are good respect to the supervised separation of normal and tumoral tissues and another is good respect to MSR, but this bicluster does not separate the data. So I have the first bicluster of cardinality 20×16, MSR = 0.1243, S(normal) = 2.0032, S(tumoral) = 0.1816, with 18 normal and 2 tumoral tissues. It can be seen the value of enrichment S(normal) very high. The second bicluster has the cardinality 22×13, MSR = 0.1202, S(normal) = 0, S(tumoral) = 1.8158 with 0 normal and 22 tumoral tissues. The third has the cardinality 19×10 and has a very small MSR = 0.0551 but does not give a good separation of the data: it includes 14 normal and 5 tumoral tissues. The other biclusters does not give the separation. The Cheng and Church algorithm gives 10 biclusters and two of them separate good a half of normal and tumoral tissues with very slow value of MSR for the both biclusters. So we have the first bicluster of cardinality 17×15,

MSR = 0.0486, S(normal) = 1.833, S(tumoral) = 0.3204 and includes 14 normal and 3 tumoral tissues. The second bicluster has the cardinality 13×26, MSR = 0.0477, S(normal) = 0, S(tumoral) = 1.8158 and includes 0 normal and 13 tumoral tissues. The other biclusters do not separate the tissues. The Spectral algorithm gives 22 biclusters and only 3 of them are not overlapped by rows. But no one gives a good separation of the tissues and MSR. SAMBA gives 13 biclusters, the best separation is obtain in two cases. The first is the bicluster of cardinality 23×8, MSR = 0.1221, S(normal) = 1.6452, S(tumoral) = 0.4737 and includes 17 normal and 6 tumoral tissues. The second bicluster has the cardinality 25×5, MSR = 0.497, S(normal) = 0.4452, S(tumoral) = 1.4526 and includes 5 normal and 20 tumoral tissues. It can be seen that in the second case the value of MSR is more high than in other algorithms and this bicluster can be obtained from only unsupervised analysis. It can be seen that no algorithm finds only two groups of biclustering for normal and tissues separation. In all the cases at least one biclusters exists that contains normal and tumoral tissues together in quite equal proportion. Only Cheng and Church algorithm finds biclusters with MSR less than 0.05 that suggests about a high coherence of the found biclusters, but only one bicluster has good separation of Tumoral tissues. Plaid finds a very good separation of normal and tumoral tissues in 2 cases, but the value of MSR is greater than 0.12, which indicates a high level of noise.

The comparison of the different biclustering techniques can be seen in the Fig. 4.11 (left).

Let me now apply CBA algorithm to this case. In the Fig. 4.12 (left) can be



Comparison of the different biclustering techniques. GC case. Case of all data (left), case of only tumoral data (right).

Figure 4.11: Comparison of the different biclustering techniques. GC case.

seen the dependence of 3 functions: for abscissa I sign number of genes, for ordinate I sign number of tissues, color of graph's entry means the value of error. The minimal number of tissues I choose 27. For different number of genes I run the algorithm for low value of initial error. The results can be seen in the Table 4.8. It can be seen that the value media of enrichment S for the normal tissues is always more than 1.1 (except two cases) and for the tumoral tissues it is less than 1. It means that for all cases biclusters separate only normal tissues. And the tumoral tissues don't proved some class. It is clear because the normal tissues have the high level of correlation, they are very ordered for any number of genes. More accurate result can be seen for a large number of genes. Here I avoid the possibility to find random little parts of the tissues, normal and tumoral, that are seem to have the same properties due the small number of genes.

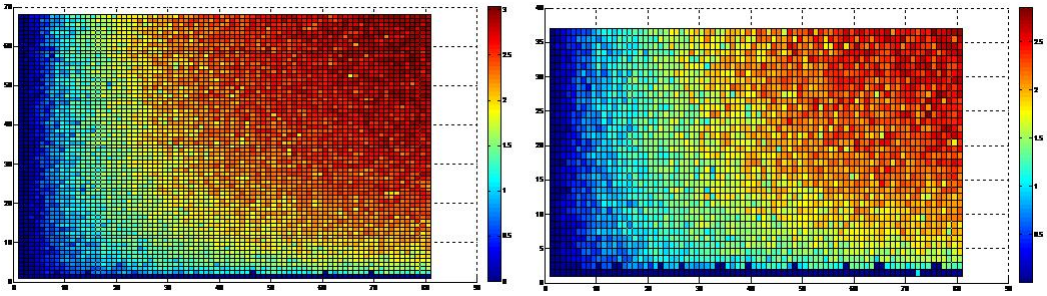
For the Table 4.8. and Table 4.9. the following indication is done: N_g - number of genes in the experiment, Err - the initial error, (Er1, Er2, Er3) the minimal, medium and maximal errors for the resulting biclus-

ters respectively, (Msr1, Msr2, Msr3) the minimal, medium and maximal values of MSR for the resulting biclusters respectively, (S_n1, S_n2, S_n3) and (P_n1, P_n2, P_n3) the minimal, medium and maximal values of enrichment and p-value for normal tissues respectively, (S_t1, S_t2, S_t3) and (P_t1, P_t2, P_t3) the minimal, medium and maximal values of enrichment and p-value for tumoral tissues respectively, (S_s1, S_s2, S_s3) and (P_s1, P_s2, P_s3) the minimal, medium and maximal values of enrichment and p-value for MSS respectively, (S_i1, S_i2, S_i3) and (P_i1, P_i2, P_i3) the minimal, medium and maximal values of enrichment and p-value for MSI respectively, Rel - the relation of number of the normal and tumoral tissues (Mss/Msi tissues) in the bicluster with minimal error, N - number of biclusters that was find.

Case of tumoral data.

Now I cancel all normal tissues and analyze only tumoral.

Plaid algorithm finds in only one case bicluster that separate MSI tissues and has the following characteristics: the cardinality 11×17, MSR 0.1027 and includes 0 MSS and 11 MSI tissues. Cheng and Church algorithm finds 10 biclusters with low value of MSR (in media 0.0479) but does not give a good separation of the tissues. SAMBA and Spectral algorithms do not give a separation. MSS and MSI tissues have a large level of noise so that it is more difficult to find their separation. From all considered techniques only Plaid finds one bicluster of MSI with larger value of MSR than 0.1. As in the first case, Cheng and Church algorithm finds biclusters with low level of MSR (0.04). But in this case no good separation of MSS and MSI is found. Samba and Spectral algorithms find biclusters with a great value of MSR and no good separation of the data.



(left) A dependence of 3 functions: for abscissa I sign number of genes, for ordinate I sign number of tissues, color of graph's entry means the value of error for normal/tumoral tissues. (right) A dependence of 3 functions: for abscissa I sign number of genes, for ordinate I sign number of tissues, color of graph's entry means the value of error for tumoral tissues.

Figure 4.12: Dependence of initial conditions

The comparison of the different biclustering techniques can be seen in the Fig. 4.11 (right).

Like was describe first I construct Fig. 4.12 (right) to analyze the dependence for number of genes, tissues and error. Minimal number of tissues I assign 17. The result can be seen the Table 4.9. It can be seen that for the low number of genes I always receive the separation of MSI (see value of enrichment). For the large number of the biclusters of MSS. It can be seen that MSS tissues form the biclusters for the large number of genes and MSI tissues form the biclusters for small number of genes because of different behavior of the disease in these tissues. In difference from MSI, the tissues MSS are more stable and can be detected correctly for large number of genes. MSI are very unstable and have the similar behavior only for the small number of genes.

Analysis of the overlap

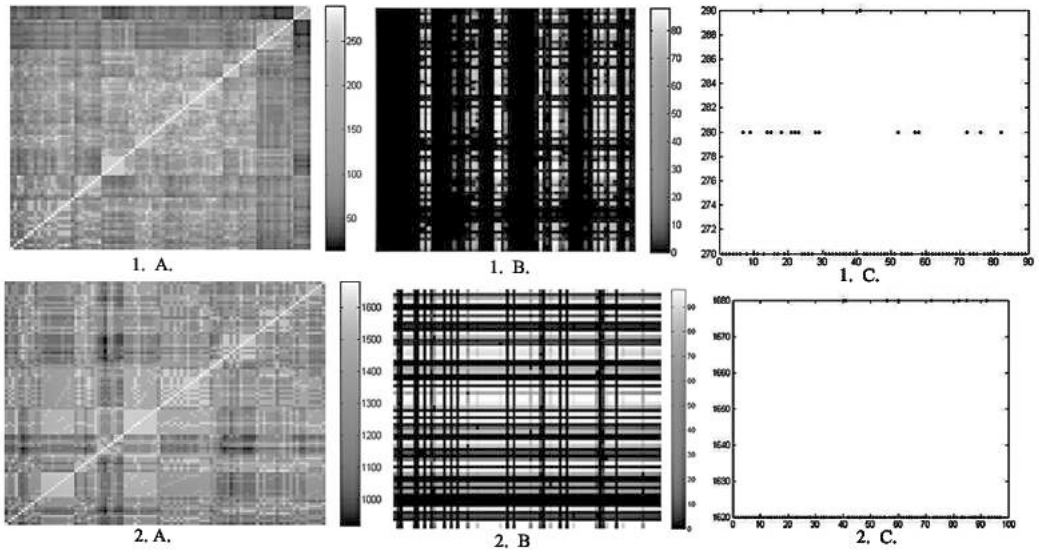
I analyze 3 cases to understand the overlap of biclusters (the best cases of unsupervised separation). They are:

1. all tissues, biclusters of 10 genes (see Fig. 4.13 cases 1.A,1.B,1.C),
2. all tissues, biclusters of 60 genes (see Fig. 4.13 cases 2.A,2.B,2.C),
3. tumoral tissues, biclusters of 30 genes (see Fig. 4.14).

I obtain three results for every case:

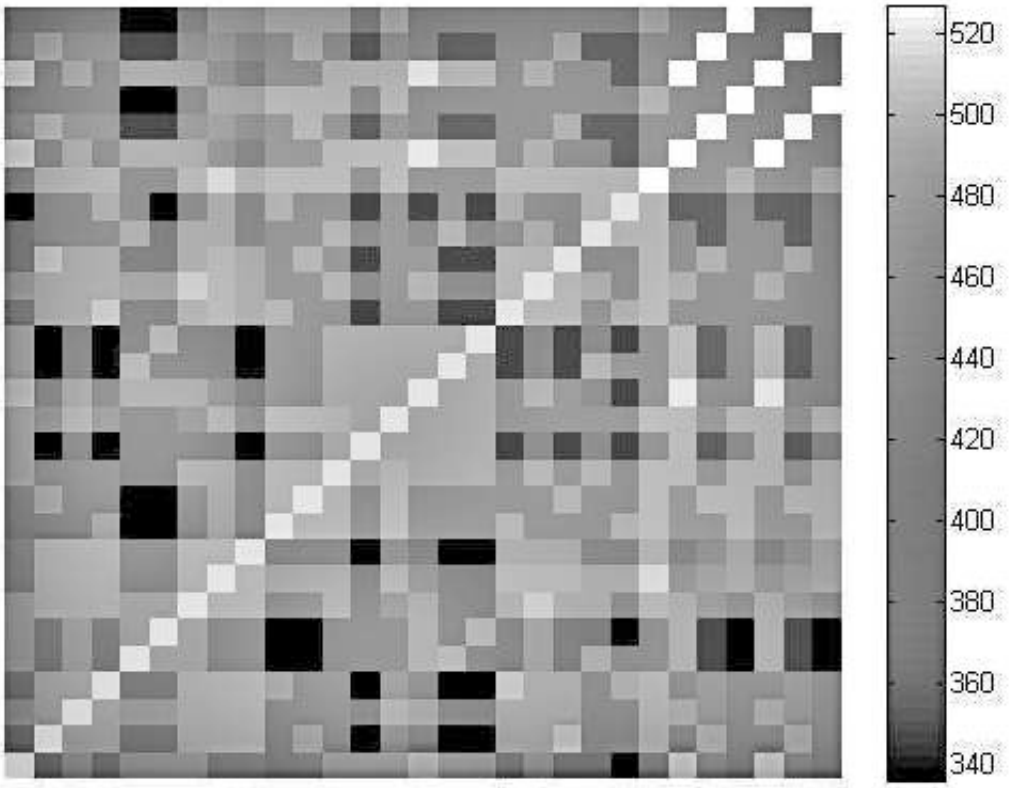
1. the matrix with entries that equal to a number of elements of intersection of every pair of biclusters. This is symmetric matrix (Fig. 4.13 cases A);
2. for every element of a data matrix I calculate a number of biclusters in which this element is contained. This result as a matrix is presented (Fig. 4.13 cases B);
3. dimension of the biclusters (Fig. 4.13 cases C).

The resulting matrices can be seen in the Fig. 4.13 In the third case the biclusters has a very high overlap. And can be considered as a unique bicluster. The heatmap of intersection in the third case can be seen in a Fig. 4.14.



A. the matrix with entries that equal to a number of elements of intersection of every pair of biclusters; B. for every element of a data matrix, a number of biclusters in which this element is contained; C. dimension of the biclusters; Case 1: all tissues, biclusters of 10 genes Case 2: all tissues, biclusters of 60 genes.

Figure 4.13: Result for overlap



The matrix with entries that equal to a number of elements of intersection of every pair of biclusters, tumoral tissues, biclusters of 30 genes.

Figure 4.14: The heatmap of intersection

Table 4.3: Results of the analysis on NCI60 data.

	Theor	**Plaid**	**Bimax**	**CC**	**Spectral**	**CBA**
N	8	8	82	100	9	6
dim1	8×200	5×45	-	-	-	7× 120
dim2	6×200	4×56	-	-	-	6× 106
dim3	7×200	10×36	6× 20	-	-	10× 120
dim4	9×200	-	-	-	7× 42	5× 80
dim5	8×200	12× 44	9× 23	-	7× 42	9× 124
dim6	7×200	-	-	-	-	-
dim7	6×200	-	-	-	-	-
dim8	6×200	-	-	-	-	5× 120
msr1	0.48	0.46	-	-	-	0.25
msr2	0.43	0.25	-	-	-	0.25
msr3	0.34	0.4	0.22	-	-	0.38
msr4	0.7	-	-	-	0.81	0.3
msr5	0.36	0.32	0.16	-	0.65	0.3
msr6	0.72	-	-	-	-	-
msr7	0.59	-	-	-	-	-
msr8	0.47	-	-	-	-	0.42
P-value1	0	0.001	-	-	-	0
P-value2	0	0	-	-	-	0
P-value3	0	0	0	-	-	0
P-value4	0	-	-	-	0.01	0
P-value5	0	0	0	-	0.001	0
P-value6	0	-	-	-	-	-
P-value7	0	-	-	-	-	-
P-value8	0	-	-	-	-	0.012
Enrich1	7	5.6	-	-	-	7.13
Enrich2	9.33	9.33	-	-	-	9.33
Enrich3	8	5.09	8	-	-	5.6
Enrich4	6.22	-	-	-	3.56	6.22
Enrich5	7	4.08	5.44	-	5.33	5.44
Enrich6	8	-	-	-	-	-
Enrich7	11.2	-	-	-	-	-
Enrich1	9.33	-	-	-	-	4.66

Table 4.4: Results for the Matrix 20×20

	Plaid	C&C	Spect	SAMBA	CBA	Theor
N of bic	1	2	22	-	2	2
no overl	1	2	2	-	2	2
the best	1	2	1	-	2	2
MSR1	0.02	0	-	-	0	0
MSR2	-	0	0.04	-	0	0
E1	1.67	2.5	-	-	2.5	2.5
E2	-	2.5	1.39	-	2.5	2.5
dim1	6×6	8×7	9×9	-	8×8	8×8
dim2	-	7×8	-	-	8×8	8×8

Table 4.5: Results for the Matrix 100× 100

	Plaid	C&C	Spect	SAMBA	CBA	Theor
N of bic	6	3	2	5	3	3
no overl	3	3	2	3	3	3
the best	3	3	2	2	3	3
MSR1	0	0	0	-	0	0
MSR2	0	0	-	0.005	0	0
MSR3	0	0	0.013	0.011	0	0
E1	2.5	2.5	2.5	-	2.5	2.5
E2	2.44	2.44	-	2.5	2.5	2.5
E3	2.5	2.5	2.15	2.5	2.5	2.5
dim1	40×40	37×39	40×40	-	40×40	40×40
dim2	28×30	30×30	-	16×11	41×41	41×41
dim3	24×21	40×40	42×22	6×3	40×40	40×40

Table 4.6: Class Distribution of E.coli

class	N of elements
cp (cytoplasm)	143
im (inner membrane without signal sequence)	77
pp (perisplasm)	52
imU (inner membrane, uncleavable signal sequence)	35
om (outer membrane)	20
omL (outer membrane lipoprotein)	5
imL (inner membrane lipoprotein)	2
imS (inner membrane, cleavable signal sequence)	2

Table 4.7: Results for E.coli data

	Plaid	C&C	Spect	SAMBA	CBA	Theor
N of bic	2	6	4	9	4500	8
no overl	2	6	2	3	5	8
the best	2	2	1	3	5	6
MSR1	-	0.0023	-	0	0.005	0.006
MSR2	0	0.0034	-	0	0.007	0.01
MSR3	-	-	-	-	0.006	0.006
MSR4	0	-	0.007	-	0.005	0.006
MSR5	-	-	-	0	0.005	0.002
MSR6	-	-	-	-	-	0.006
E1	-	2.28	-	1.91	2.15	2.4
E2	2.09	2.29	-	1.96	4.36	4.36
E3	-	-	-	-	6.89	9.6
E4	1.96	-	2.2	-	16.8	16.8
E5	-	-	-	33.6	5.89	67.2
E6	-	-	-	-	-	6.46
bic1	-	cp	-	cp	cp	cp
bic2	im+imU	im	-	im	im	im
bic3	-	-	-	-	imU	imU
bic4	om+pp	-	om	-	om	om
bic5	-	-	-	omL	omL+pp	omL
bic6	-	-	-	-	-	pp
dim1	-	70×5	-	31×3	115×6	143×8
dim2	110×2	39×6		41×1	56×6	77×8
dim3	-	-	12×5	-	46×6	35×8
dim4	42×2	-	-	-	16×6	20×8
dim5	-	-	-	10×2	48×6	5×8
dim6	-	-	-	-	-	52×8

Table 4.8: Results of the analysis on normal/tissues data.

N_g	5	10	11	12	15	20	25	30	35	40	45	55	60	65
Err	0.6	0.71	0.75	0.8	0.88	0.98	1.1	1.2	1.28	1.4	1.53	1.8	2.0	2.2
Er1	0.14	0.36	0.39	0.49	0.48	0.78	0.77	0.91	0.85	1.02	1.17	1.45	1.68	1.84
Er2	0.43	0.57	0.58	0.69	0.72	0.83	1.01	1.07	1.05	1.17	1.34	1.64	1.87	1.93
Er3	0.59	0.7	0.75	0.79	0.87	0.97	1.09	1.19	1.15	1.38	1.42	1.78	1.93	2.12
Msr1	0.01	0.02	0.02	0.03	0.03	0.04	0.04	0.05	0.05	0.06	0.06	0.07	0.08	0.1
Msr2	0.02	0.03	0.03	0.03	0.03	0.04	0.05	0.05	0.05	0.06	0.07	0.08	0.09	0.12
Msr3	0.03	0.03	0.03	0.03	0.04	0.04	0.05	0.06	0.06	0.07	0.08	0.09	0.12	0.12
S_n1	0.83	1.15	0.95	1.07	1.07	1.03	0.99	0.95	0.99	0.99	0.82	0.91	0.91	1.07
S_n2	1.25	1.41	1.38	1.43	1.4	1.22	1.28	1.06	1.13	1.13	0.99	1.17	1.37	1.24
S_n3	1.6	1.73	1.73	1.73	1.65	1.57	1.57	1.32	1.32	1.24	1.24	1.27	1.57	1.46
P_n1	0.79	0.12	0.43	0.25	0.25	0.25	0.43	0.43	0.43	0.43	0.79	0.62	0.62	0.25
P_n2	0.05	0.00	0.00	0.00	0.00	0.05	0.02	0.25	0.12	0.12	0.43	0.12	0.00	0.05
P_n3	1^{-4}	1^{-6}	1^{-6}	1^{-6}	1^{-6}	1^{-6}	1^{-6}	0.02	0.02	0.05	0.05	0.05	1^{-4}	7^{-4}
S_t1	0.4	0.4	0.4	0.4	0.47	0.54	0.54	0.74	0.74	0.81	0.81	0.78	0.54	0.63
S_t2	0.76	0.67	0.69	0.65	0.67	0.82	0.77	0.95	0.89	0.89	1.01	0.87	0.7	0.81
S_t3	0.92	0.87	1.03	0.94	0.94	0.97	1.01	1.03	1.01	1.01	1.14	1.07	1.07	0.94
P_t1	1	1	1	1	1	0.99	0.99	0.95	0.95	0.88	0.88	0.88	0.99	0.99
P_t2	0.95	0.99	0.98	0.98	0.98	0.88	0.95	0.57	0.75	0.75	0.38	0.75	0.98	0.88
P_t3	0.57	0.75	0.38	0.57	0.57	0.57	0.38	0.38	0.38	0.38	0.09	0.21	0.21	0.38
Rel	$\frac{20}{9}$	$\frac{22}{5}$	$\frac{17}{10}$	$\frac{21}{6}$	$\frac{15}{12}$	$\frac{16}{11}$	$\frac{14}{18}$	$\frac{14}{13}$	$\frac{14}{13}$	$\frac{14}{13}$	$\frac{13}{14}$	$\frac{15}{12}$	$\frac{18}{9}$	$\frac{15}{12}$
N	187	89	163	24	168	35	59	25	98	54	36	7	97	44

Table 4.9: Results of the analysis on MSS(s)/MSI(i) data.

N_g	10	15	20	25	30	35	40	45	50	55	60	65	70	19t70g
Err	0.85	0.95	1.05	1.15	1.21	1.35	1.45	1.55	1.65	1.95	2.1	2.35	2.45	2.6
Er1	0.37	0.57	0.79	0.75	0.73	0.98	0.82	1.05	1	1.09	1.18	1.27	1.61	1.48
Er2	0.58	0.83	0.93	0.93	0.86	1.04	1.14	1.16	1.2	1.72	1.62	1.75	1.65	2.05
Er3	0.84	0.91	1.04	1.12	0.96	1.14	1.44	1.55	1.52	1.92	2.05	2.33	1.67	2.53
Msr1	0.02	0.04	0.05	0.05	0.06	0.06	0.06	0.07	0.07	0.08	0.09	0.11	0.12	0.13
Msr2	0.04	0.05	0.05	0.06	0.06	0.06	0.07	0.08	0.08	0.09	0.11	0.12	0.13	0.14
Msr3	0.05	0.05	0.05	0.06	0.06	0.07	0.08	0.08	0.09	0.12	0.13	0.14	0.13	0.14
S_s1	0.47	0.59	0.47	0.47	0.35	0.82	0.59	0.82	0.82	0.71	0.94	0.82	1.38	1.06
S_s2	0.92	0.75	0.75	0.64	0.52	0.96	0.86	1.01	1.08	1.02	1.27	1.17	1.46	1.49
S_s3	1.29	0.94	0.94	1.06	0.71	1.06	1.18	1.18	1.29	1.53	1.53	1.53	1.5	1.58
P_s1	0.99	0.99	0.99	0.99	1	0.74	0.99	0.74	0.74	0.9	0.74	0.74	0.02	0.5
P_s2	0.74	0.9	0.9	0.98	0.99	0.5	0.74	0.5	0.26	0.5	0.09	0.26	0.00	0.00
P_s3	0.09	0.74	0.74	0.26	0.9	0.26	0.09	0.09	0.09	0.00	0.00	0.00	0.00	0.00
S_i1	0.71	1.06	1.06	0.94	1.29	0.94	0.82	0.82	0.71	0.47	0.47	0.47	0.5	0.42
S_i2	1.08	1.25	1.25	1.36	1.48	1.04	1.14	0.99	0.92	0.97	0.73	0.83	0.54	0.52
S_i3	1.53	1.41	1.53	1.53	1.65	1.18	1.41	1.18	1.17	1.29	1.05	1.17	0.62	0.94
P_i1	0.9	0.26	0.26	0.74	0.09	0.74	0.74	0.74	0.9	0.99	0.99	0.99	0.02	0.5
P_i2	0.26	0.09	0.09	0.02	0.00	0.5	0.26	0.5	0.74	0.5	0.9	0.74	0.00	0.00
P_i3	0.00	0.00	0.00	0.00	0.00	0.09	0.00	0.09	0.25	0.09	0.26	0.26	0.00	0.00
Rel	$\frac{7}{11}$	$\frac{8}{9}$	$\frac{6}{12}$	$\frac{4}{13}$	$\frac{4}{14}$	$\frac{8}{9}$	$\frac{7}{10}$	$\frac{10}{9}$	$\frac{9}{10}$	$\frac{11}{8}$	$\frac{12}{5}$	$\frac{10}{7}$	$\frac{13}{5}$	$\frac{15}{5}$
N	34	10	38	77	29	11	74	83	79	146	89	124	3	21

Chapter 5

Biclustering by Resampling

5.1 Fuzzy clustering.

The popularity of fuzzy set methods in fields such as control and rule-based reasoning is due to the fact that they are able to represent ill-defined classes and concepts in a natural way [16]. In Zadeh's formulation of fuzzy set theory, the representation of such ill-defined classes or concepts is achieved by means of membership functions defined over the appropriate domain of discourse [27]. These memberships are absolute, and denote degrees of belonging or typically. Zimmermann and Zysno shown [28] that a good model for membership function that model vague concepts or classes is:

$$u(x) = \frac{1}{1+d(x,x_0)},$$

where $d(x, x_0)$ is the distance of a point x in the domain of discourse from

the prototypical member x_0 of the class. In other words, in this formulation, membership values are solely a function of the "distance" of a point from a prototypical member [28]. The FCM algorithm and its derivates are not really suitable for generating such membership functions from training data, since they do not generate memberships that can be interpreted as degree of compatibility. Many other researches was made for the best definition of the memberships and objective functions.I follow by [16].

Let U denote a fuzzy partition matrix generated by the FCM algorithm. Then the elements $u_{ij}$of U satisfy the following conditions [29]:

$$u_{ij} \in [0, 1] \text{ for all } i \text{ and } j,$$

$$0 < \sum_{j=1}^{N} u_{ij} < N \text{ for all } i, \quad (1)$$

$$\sum_{j=1}^{C} u_{ij} = 1 \text{ for all } j.$$

Here, $u_{ij}$is the grade of membership of the feature point x_j in cluster β_i, C is the number of classes, and N is the total number of feature points. It follows that the symbol β_i will be used to denote the i-th cluster and its prototype, since the prototype contains the parameters that characterize the cluster.

The last condition confines the memberships to lie on the hyperplane defined by $\sum_{j=1}^{C} u_{ij} = 1$.

The original FCM formulation minimizes the objective function given by

$$J(L,U) = \sum_{i=1}^{C} \sum_{j=1}^{N} (u_{ij})^m d_{ij}^2 \text{ subject to } \sum_{i=1}^{C} u_{ij} = 1 \text{ for all } j,$$

where $L = (\beta_1, ..., \beta_C)$ is a C-tuple of prototypes d_{ij}^2 is the distance of feature point x_j to prototype β_i, N is the total number of feature vectors, C is the number of classes and $U = [u_{ij}]$ is a $C \times N$ matrix, call es the fuzzy C-partition matrix, satisfying the conditions in (1). Here, u_{ij} is the grade of membership of the feature point x_j in cluster β_i, and $m \in [1, \infty)$ is a weighting exponent called the fuzzifier.

5.2 Possibilistic Clustering Paradigm

For PCM the obtained evaluations of membership to clusters are interpretable as a *degree of typicality*. The possibilistic approach to clustering proposed by Keller and Krishnapuram [22], assumes that the membership function of a data point in a fuzzy set (or cluster) is absolute, i.e. it is an evaluation of a degree of typicality not depending on the membership values of the same point in other clusters.

$$u_{ij} \in [0,1], \forall i, j;$$

$$0 < \sum_{j \in C} u_{ij} < n_c, \forall i;$$

$$\bigvee_{i} u_{ij} > 0, \forall j.$$

The task of the objective function is to find the highest memberships for representative feature points, while unrepresentative points should have

low membership in all clusters. In the following function the distance from the features to prototypes is made as low as possible while u_{ij} is as large as possible.

$$J(L,U) = \sum_{j=1}^{N}(u_{ij})^m d_{ij}^2 + \eta_i \sum_{j=1}^{N}(1-u_{ij})^m.$$

where $L = (\beta_1, ..., \beta_C)$ is a C-tuple of prototypes d_{ij}^2 is the distance of feature point x_j to prototype β_i, N is the total number of feature vectors, C is the number of classes and $U = [u_{ij}]$ is a $C \times N$ matrix, call es the fuzzy C-partition matrix, satisfying the conditions in (1). Here, u_{ij} is the grade of membership of the feature point x_j in cluster β_i, and $m \in [1, \infty)$ is a weighting exponent called the fuzzifier. The parameter η (that the authors term *scale*) depends on the average size of the k-th cluster, and must be assigned before the clustering procedure starts and $\eta \sim d_{ij}^2$.

Keller and Krishnapuram proposed a theorem.

Theorem: suppose that $X = x_1, x_2, ..., x_N$ is a set of feature vectors, $L = (\beta_1, ..., \beta_C)$ is a C-tuple of prototypes, d_{ij}^2 is the distance of feature point x_j to the cluster prototype $\beta_{ij}(i = 1, ..., C; j = 1, ..., n_g)$, and $U = [u_{ij}]$ is a $C \times N$ matrix of possibilistic membership values. Then U may be a global minimum for $J(L,U)$ only if:

$$u_{pq} = [1 + (d_{pq}^2/\eta_p)^{\frac{1}{m-1}}]^{-1}.$$

Proof:

In order to derive the necessary conditions and the membership up grating equations, it can be noted that the rows and columns of U are independent of each other. Hence, minimizing $J_m(L,U)$ with respect to U is equivalent to minimizing the following individual objective function with respect to each of the u_{ij}(provided that the resulting solution lies in the interval [0,1]):

$$J_m^{ij}(\beta_i, u_{ij}) = u_{ij}^m d_{ij}^m + \eta_i(1-u_{ij})^m. \text{ (2)}$$

Differentiating (2) with respect to $u_{ij}$and setting it to 0 leads to the equation

$$u_{ij} = \frac{1}{1+(\frac{d_{ij}^2}{\eta_i})^{\frac{1}{m-1}}}. \text{ (3)}$$

It is obvious from (3) that u_{ij}lies in the desired range.

Solving the equation for d^2 in terms of u_{ij} from (3), can be obtained:

$$d_{ij}^2 = \eta_i(\frac{1-u_{ij}}{u_{ij}})^{m-1}. \text{ (4)}$$

It can be now eliminated d_{ij}^2 from the objective function using (4):

$$J(L,U) = \eta_i \sum_{j=1}^{N}(1-u_{ij})^{m-1}.$$

For a given value of η_i, minimizing $J(L,U)$ is equivalent to maximizing

$$J'(L,U) = \eta_i \sum_{j=1}^{N}(1-(1-u_{ij})^{m-1}) = \eta_i \sum_{j=1}^{N} u'_{ij}, \text{ (5)}$$

where $u'_{ij} = 1 - (1 - u_{ij})^{m-1}$ can be interpreted as a modified membership. It is to be noted that u'_{ij} is obtained from u_{ij} via a monotonic mapping since

$$\frac{d}{du_{ij}} u'_{ij} = (m-1)(1-u_{ij})^{m-2} > 0 \text{ for } m > 1.$$

Hence, u'_{ij} varies the same way as u_{ij}, i.e. $u_{ij} = 0 \Rightarrow u'_{ij} = 0$; $u_{ij} = 1 \Rightarrow u'_{ij} = 1$; both are monotonically decreasing functions of d^2_{ij}. Furthermore, for the special case of $m = 2$, (5) reduces to

$$J'(L, U) = \eta_i \sum_{j=1}^{N} u_{ij}. \quad (6)$$

From (5) and (6), It can be seen that for a given value of η_i, each of the C-subobjective functions is maximized by choosing the prototype location such that the sum of the (modified) memberships is maximized. This is achieved if the prototype is located in a dense region since the (modified) membership is a monotonically decreasing function of the distance to the prototype. If there are indeed C-dense regions in feature space (corresponding to C-distinct clusters), then, with proper initialization, each prototype will converge to a dense region. In such a situation, even if all η_i are equal (and, hence, all subobjective functions become identical) each of them will still have C-distinct minima corresponding to the C-dense regions.

5.2.1 The meaning of the scale parameter η and the fuzzifier parameter m

Krishnapuram et al. [22] in their work described the choice of the parameters η and m. They noted that that η determines the relative degree to which the second term in the objective function is important compared with the first. If the two term are to be weighted roughly equally, then η should be of the order of d_{ij}^2. And give the definition:

$$\eta_p = Q\frac{\sum_{q=1}^{K} u_{pq}^m d_{pq}^2}{\sum_{q=1}^{K} u_{pq}^m}.$$

Typically Q is chosen to be 1. In this Thesis following Krishnapuram et al. I use the same definition of the scale parameter with some coefficients, depending from the dimension of the biclusters that I want to find. The authors assume that the "fuzzifier" m, determines the rate of decay of the membership value. When m = 1, the memberships are crisp, i.e., all points with $d^2(x_j, \beta_i)$ greater than η_i will have zero memberships. When $m \to \infty$, the membership function does not decay to zero at all. Note that a good choice for the value of the fuzzifier m for the PCM seems to be around 1.5. Then the authors eliminate m altogether by choosing alternative formulations of the PCM and define:

$$J(U, Y) = \sum_{p\in K}\sum_{q\in c} u_{pq}E_{pq}^2 + \sum_{p\in K}\frac{1}{\beta_p}\sum_{p\in c}(u_{pq}\log u_{pq} - u_{pq}),$$

where $E_{pq} = ||k_q - y_p||^2$ is the squared Euclidean distance, and the parameter β_p (that I can term *scale*) depends on the average size of the p-th

cluster, and must be assigned before the clustering procedure. Note that $(u_{pq} \log u_{pq} - u_{pq})$ is a monotonically decreasing function in [0,1], similar to $(1 - u_{pq})^m$. Thanks to the regularizing term, points with a high degree of typicality have high u_{pq} values, and points not very representative have low u_{pq} values in all the clusters. Note that if I take $\beta_p \to \infty \; \forall p$ (i.e., the second term of $J_m(U, Y)$ is omitted), I obtain a trivial solution of the minimization of the remaining cost function (i.e., $u_{pq} = 0 \; \forall p, q$), as no probabilistic constraint is assumed.

The pair (U, Y) minimizes J_m, under my constraints only if:

$$u_{pq} = e^{-E_{pq}/\beta_p}, \forall p, q,$$

and

$$y_p = \frac{\sum_{q=1}^{r} x_q u_{pq}}{\sum_{q=1}^{r} u_{pq}}, \forall p.$$

These conditions can be interpreted as formulas for recalculating the membership functions and the cluster centers (Picard iteration technique), as shown, e.g., in [23].

A good initialization of centroids must be performed before applying PCM (using, e.g., Fuzzy C-Means [16], [22], or Capture Effect Neural Network [23]). The PCM works as a refinement algorithm, allowing us to interpret the membership to clusters as cluster typicality degree, moreover PCM shows a high outliers rejection capability as it makes their membership very low.

5.3 The possibilistic approach to biclustering

In this section following Filippone et al. [23]I represent the concept of biclustering in a fuzzy set theoretical approach. For each bicluster they assign two vectors of membership, one for the rows and one for the columns, denoting them **a** and **b** respectively. Such that if a_i and b_j equal to one(zero) then row i and column j belong(or not) to the bicluster. For an element x_{ij} of XI assign its membership u_{ij} such that:

$$u_{ij} = \textbf{and}(a_i, b_j).$$

The cardinality of the bicluster is then defined as:

$$n = \sum_i \sum_j u_{ij}.$$

The membership u_{ij} can be obtained like:

$$u_{ij} = a_i b_j, (\textbf{product})$$

or

$$u_{ij} = \frac{a_i + b_j}{2}, (\textbf{average}).$$

So the equations:

$$H(I,J) = \frac{1}{|I||J|} \sum_{i \in I, j \in J} RS_{IJ}(i,j)^2 = \frac{1}{|I||J|} \sum_{i \in I, j \in J} (a_{ij} - a_{Ij} - a_{iJ} + a_{IJ})^2,$$

and

$$a_{iJ} = \frac{1}{|J|} \sum_{j \in J} a_{ij}, a_{Ij} = \frac{1}{|I|} \sum_{i \in I} a_{ij}, a_{IJ} = \frac{1}{|I||J|} \sum_{j \in J, i \in I} a_{ij}$$

can be generalized as:

$$d_{ij}^2 = \frac{(x_{ij} + x_{IJ} - x_{iJ} - x_{Ij})^2}{n},$$

where:

$$x_{IJ} = \frac{\sum_i \sum_j u_{ij} x_{ij}}{\sum_i \sum_j u_{ij}}, x_{iJ} = \frac{\sum_j u_{ij} x_{ij}}{\sum_j u_{ij}},$$

$$x_{Ij} = \frac{\sum_i u_{ij} x_{ij}}{\sum_i u_{ij}}, G = \sum_i \sum_j u_{ij} d_{ij}^2.$$

To maximize the bicluster cardinality n and minimize the residual G using the fuzzy possibilistic paradigm Filippone et al. make the following assumptions:

- one bicluster at a time is considered;
- the fuzzy memberships a_i and b_j are interpreted as typicality degrees of gene i and condition j with respect to the bicluster;
- the membership u_{ij} is computed.

All these requirements are fulfilled by minimizing the following func-

tional J_B with respect to **a** and **b**:

$$J_B = \sum_i \sum_j (\frac{a_i + b_j}{2}) d_{ij}^2 + \lambda \sum_i (a_i \ln(a_i) - a_i) + \mu \sum_j (b_j \ln(b_j) - b_j).$$

As in the Possibilistic C-means model, the parameters λ and μ control the size of the bicluster by penalizing too small values of the memberships. Their values can be estimated by simple statistics over the training set, and then possibly hand-tuned, for instanced to incorporate a-priory knowledge.

Setting the derivatives of J_B with respect to the memberships a_i and b_j to zero:

$$\frac{\partial J}{\partial a_i} = \sum_j \frac{d_{ij}^2}{2} + \lambda \ln(a_i) = 0,$$

$$\frac{\partial J}{\partial b_j} = \sum_i \frac{d_{ij}^2}{2} + \mu \ln(b_j) = 0,$$

the following solutions can be obtained:

$$a_i = \exp\left(-\frac{\sum_j d_{ij}^2}{2\lambda}\right), \qquad b_j = \exp\left(-\frac{\sum_i d_{ij}^2}{2\mu}\right).$$

5.3.1 The Possibilistic Biclustering (PBC) algorithm

- Initialize the memberships **a** and **b**
- Compute d_{ij}^2 for all i, j
- Update a_i for all i
- Update b_j for all j

- If $||\mathbf{a}' - \mathbf{a}|| < \varepsilon$ and $||\mathbf{b}' - \mathbf{b}|| < \varepsilon$ then stop
- else jump to step 2.

The parameter ε is a threshold controlling the convergence of the algorithm. The memberships initialization can be made randomly or using some a priory information about relevant genes and conditions.

5.3.2 Bootstrap aggregating (Bagging)

In this section I follow to L. Breiman [24] and explain the Bootstrap aggregating (Bagging) technique. A learning set L consists of data $(y_n, \mathbf{x}_n), n = 1, ..., N$ where the y's are either class labels or a numerical response. There is a procedure for using this learning set to form a predictor (in my case a bicluster) $\varphi(\mathbf{x}, L)$ - if the input is $\mathbf{x}$it can be predicted y by $\varphi(\mathbf{x}, L)$. Now, suppose that there is a sequence of learning sets L_k each consisting of N independent observations from the same underlying distribution as L. The aim is to use the L_k to get a better predictor then the single learning set predictor $\varphi(\mathbf{x}, L)$. The restriction is that it is allowed to work with the sequence of predictors $\varphi(\mathbf{x}, L_k)$.

If y is numerical, an obvious procedure is to replace $\varphi(\mathbf{x}, L)$ by the average of $\varphi(\mathbf{x}, L_k)$ over k. i.e. by $\varphi_A(\mathbf{x}) = E_L\varphi(\mathbf{x}, L)$ where E_L denotes the expectation over L, and the subscript A in φ_A denotes aggregation. If $\varphi(\mathbf{x}, L)$ predicts a class $j \in 1, ..., J$, then one method of aggregating the $\varphi(\mathbf{x}, L_k)$ is by voting.

If there is a single learning set L without the luxury of replicates of L, an imitation of the process leading to φ_A can be done. Taking repeated bootstrap samples $L^{(B)}$ from L form a $\varphi(\mathbf{x}, L^{(B)})$. Breiman [24] call this procedure "*bootstrap aggregating*" or bagging.

$L^{(B)}$ forms replicate data sets, each consisting of N cases, drawn at random, but with replacement, from L. Each $(y_n, \mathbf{x}_n)$ may appear repeated some times or not at all in any particular $L^{(B)}$. The $L^{(B)}$ is a replicate data set drawn from the bootstrap distribution approximating the distribution underlying L.

5.4 Improved Possibilistic Clustering Algorithm

As shown in [15], the PBC algorithm finds the larger bicluster of the data matrix with small MSR, when compared with other methods.

Different runs of the PCB algorithm on the same data matrix find very similar biclusters with high overlapping.

In order to find further biclusters, in this paper I study the effect of resampling techniques. In particular, I use Bootstrap for generating new versions of a data matrix and after that I apply the PBC model. The new multiple versions of data matrix are obtained by making bootstrap replicates of the biclustering set. In such a way all possible biclusters I can found.

5.4.1 Applying Bootstrap aggregating to a PBC model

Let X be the data matrix $M \times N$ with elements x_{ij}, $i \in M$, $j \in N$. As first step, following the Bootstrap aggregating [24], I create l new data matrices M_{bag}. Every matrix M_{bag} has a random number of column copies from X, such that the dimension of the matrices M_{bag} is $M \times N$.

Then, for every Bagging matrix I apply the PBC algorithm and analyze the result by F, MSR, and the value of enrichment S, that can be seen follow, i.e. the *a-priori* information on the data or the GO term data base information which is useful to identify if some agglomeration of genes in a cluster is significant with respect to a specific annotation [19]. I analyze the biclusters relatively to genes (rows), and consider them as clusters.

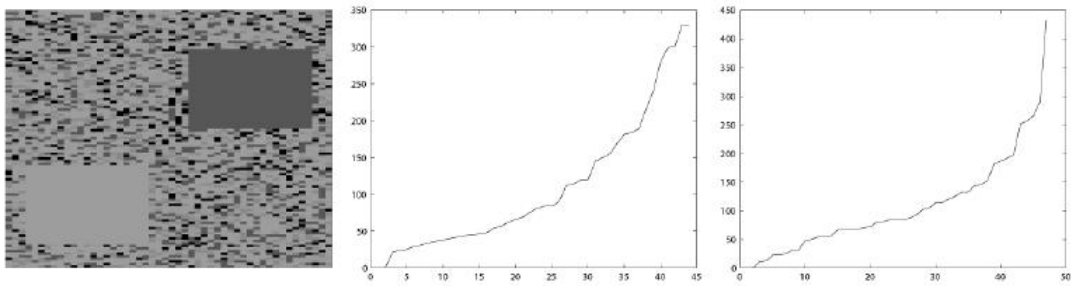
5.4.2 Results

The analysis of the synthetic data matrix

First, I apply my algorithm to the synthetic data matrix X $M \times N$, that consists of 100×50 whose elements values are from 1 to 10 (5.1 A) . There are two biclusters A and B of size 30×18 each one. The MSR value of the matrix M is 6.8. I choose the value of the coefficients λ and μ such that:

$$\lambda = \frac{\sum_{i=1}^{N} \lambda_i}{N \times 1.5}, \text{ and } \mu = \frac{\sum_{i=1}^{N} \mu_i}{N \times 1.3}$$

The threshold ε is defined as 0.001.



A) The synthetic data matrix X. B) Number of the elements from A on the ordinate respect to number of the elements from B on the abscissa. C) Number of the elements from B on the ordinate respect to number of the elements from A on the abscissa.

Figure 5.1: Result for a synthetic data matrix

PBC. I apply the PBC method for separating my data. As the result I find one bicluster (49×37) that contains (25×17) elements from the bicluster A and (22×16) elements from B, MSR = 3.8198.

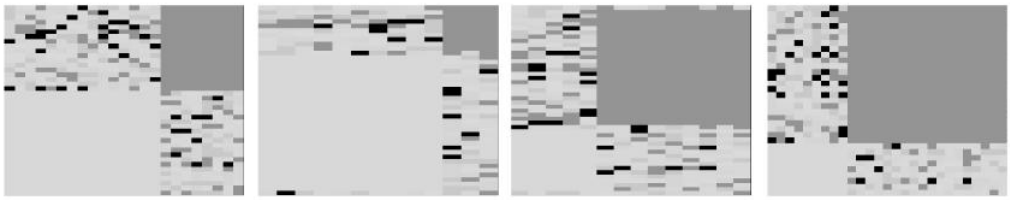
PBC Bagging algorithm.I run this algorithm 200 times and find 200 Bagging Matrices M_{bag}; then I apply PBC to these matrices and find the MSR and the value of S for every bicluster. After that I cancel all biclusters that have the MSR value more than 3.39 (half of the MSR of the data matrix X). Then for the remaining biclusters I analyze their matrix F (the first and the second columns of this matrix show how many elements from the biclusters A and B, respectively, enter in the current bicluster). As a result I obtain in many cases the separation to the biclusters as in the first case (PBC).

However, I also obtain separated biclusters A and B. For separating the bicluster A I choose the biclusters with rows that have a value in the first column of F greater than the value of the rows in the second column (size

of $A >$ size of B). And *viceversa* for the bicluster B. In 5.1 B. I have on the abscissa I have the number of elements from B that can be accepted in the bicluster with elements from A, while on the ordinate the number of elements from A. In 5.1 C. *viceversa*.

In the both cases I choose only biclusters that have large size. I can see from the graphics that in the first case the jump of the size values is from 189 to 299 while in the second case the jump is from 182 to 252. So I take all the biclusters with entry size of A greater than 299 in the first case and with entry size of B greater than 252 in the second case. As a result I have:

- I found two best cases of the separation of the bicluster A (5.2 A, B):
  - ✧ MSR = 3.2401 size = 920 (40×23), 330 elements from A, 156 elements from B;
  - ✧ MSR = 2.4048 size = 546 (42×13), 300 elements from A, 30 elements from B;
- Two best cases for the separation of the bicluster B (5.2 C, D):
  - ✧ MSR = 2.9643 size = 644 (46×14), 252 elements from B, 75 elements from A;
  - ✧ MSR = 2.8298 size = 891 (33×27), 432 elements from B, 81 elements from A;



The Heatmaps of the result of the separation of the biclusters A and B

Figure 5.2: The Heatmaps of the result.

Analysis with the PBC bagging method of the real data (Yeast)

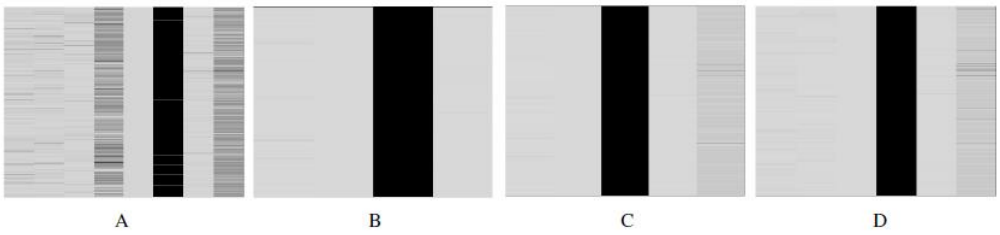
I consider the real data set Yeast (5.3 A), created by Kenta Nakai, Institute of Molecular and Cellular Biology, Japan http://archive.ics.uci.edu/ml/datasets/Yeast. This data matrix consists of 8 attributes and 1484 instances (see Table 10).

class	class definition	number of elements
CYT	(cytosolic or cytoskeletal)	463
NUC	(nuclear)	429
MIT	(mitochondrial)	244
ME3	(membrane protein, no N-terminal signal)	163
ME2	(membrane protein, uncleaved signal)	51
ME1	(membrane protein, cleaved signal)	44
EXC	(extracellular)	37
VAC	(vacuolar)	30
POX	(peroxisomal)	20
ERL	(endoplasmic reticulum lumen)	5

Table 10. Yeast data.

The matrix has a MSR = 0.0089. There are 10 classes with number of rows of (463, 5, 35, 44, 51, 163, 244, 429, 20, 30 respectively); for the PBC and PBC Bagging analysis I consider the initial conditions:

$$\lambda = \frac{\sum_{i=1}^{N} \lambda_i}{N \times 3}, \text{ and } \mu = \frac{\sum_{i=1}^{N} \mu_i}{N}.$$



The Heatmaps of A) data matrix Yeast B-D) some resulting biclusters.

Figure 5.3: The Heatmaps of Yeast and some results.

For each of these four cases for PBC Bagging I made 300 runs and built F. I also made an analysis by calculating the enrichment S. For every bicluster I kept the cases with $S \geq 1.1$.

I have the follow results (see Table 11):

- **PBC** MSR = 0.0019, size: 631×6, I found the good separation of the first bicluster.
- **PBC Bagging** The next three classes were found(results for the average for all the cases):
  - ✧ Msr = 0.0024, size: 612×6 - the separation of the bicluster 1.
  - ✧ MSR = 0.0029, size: 276×5 - the separation of the Bicluster 6.
  - ✧ MSR = 0.0015, size: 269×5 - the separation of the Bicluster 8.

  Together with this results the biclusters that contain some classes together were found. Some of them are:
  - ✧ MSR = 0.0028, size: 239×5 - the separation of the Biclusters 1 and 6 together.

✧ MSR = 0.0034, size: 622×6 - the separation of the Biclusters 1, 6 and 10 together.

The results for the average of enrichment (e) and the matrix F (in %) can be seen in the Table 11. Heatmaps of some biclusters can bee seen in the 5.3 (B, C, D).

Case		1 bic	2 bic	3 bic	4 bic	5 bic	6 bic	7 bic	8 bic	9 bic	10 bic
1.	F	60	0	10	0	15	41	28	44	30	40
	e	1.4	0	0.23	0	0.4	0.96	0.66	1.04	0.7	0.94
2. i.	F	60	0	10	0	10	40	30	40	20	40
	e	1.45	0	0.2	0	0.24	0.96	0.7	0.96	0.5	0.96
2. ii.	F	20	0	10	0	9	35	10	20	10	20
	e	1.07	0	0.53	0	0.5	1.88	0.53	1.07	0.53	1.07
2. iii.	F	20	0	1	0	4	7	7	33	7	8
	e	1.1	0	0.06	0	0.22	0.38	0.38	1.82	0.38	0.44
2. iv.	F	20	0	3	0	5	21	12	17	8	17
	e	1.24	0	0.18	0	0.31	1.3	0.74	1.05	0.49	1.05
2. v.	F	50	0	10	03	15	49	36	45	20	48
	e	1.19	0	0.24	0.007	0.35	1.16	0.85	1.07	0.47	1.15

Table 11. Results of the analysis on Yeast data.

5.4.3 Conclusion

In this Chapter I presented a new method for the biclustering analysis. My PBC Bagging algorithm is a very fast algorithm, gives a good separation of the data set with respect to the value of MSR and enrichment and permits to find all the possible biclusters of the desired size (overlapped or not), that can be seen from the results. I decided to calculate the λ and μ values as the mean of the values in the method of Krishnapuram [22],

and found a very good separation. Finally, further analysis and biological validation of the obtained results is under study.

5.5 Improving by the Genetic Algorithms.

I noted, that Genetic Algorithm (GA), applied directly to the data set, does not solve a problem of the multi-solution, but gives a good improvement to the solutions obtained previously. Following I explain the GA technique and its initialization by the Bagging technique.

The GA technique was firstly proposed by John Henry Holland [30] and permits the analysis of the multi-objective functions. GA is based on the evolutionary ideas of a natural selection and genetics. Algorithm is started with a set of solutions (represented by chromosomes), called *population*. Solutions from one population are taken and used to form a *new population*. It is supposed that a new population will be better than the old one. Solutions which are selected to form new solutions (*offspring*) are selected according to their fitness. Offspring is more suitable and has more chances to being reproduced. The algorithm consists of the following steps:

1. Randomly generate an initial population $M(0)$
2. For each individual m of the current population $M(t)$ compute and obtain the fitness functions$f(m)$
3. For each individual m in $M(t)$ define selecting probabilities $p(m)$, so that $p(m)$ is proportionally to $f(m)$

4. Use a probabilistic selection of the individuals from $M(t)$ to produce offspring via genetic operators and generate $M(t+1)$,

5. Repeat step 2 until satisfying solution is obtained.

Single-objective GA are very important in the optimization problems, but most real search typically involve multiple objectives. The MOEA (Multi-Objective Evolutionary Algorithms) tries to optimize more then one conflicting characteristics represented by fitness function by generating a set of Pareto-optimal solutions [31]. I will use MOEA in my case.

As can be seen, the first step includes random generating of the initial population. The result obtained in such a way can be evidently improved by the definite initialization, that can be seen in the final algorithm.

So the final algorithm will be following:

1. create a new bagging matrix $L^{(B)}$

2. apply the technique of [22] to initialize the values of μ and λ for PBC algorithm

3. apply PBC algorithm and obtain the vectors a and b

4. create the vector $c = [ab]$ and use it as initial population for GA

5. apply MOEA with the following parameters:

$$f_1 = \sum_i \sum_j u_{ij} d_{ij}^2, \tag{5.1}$$

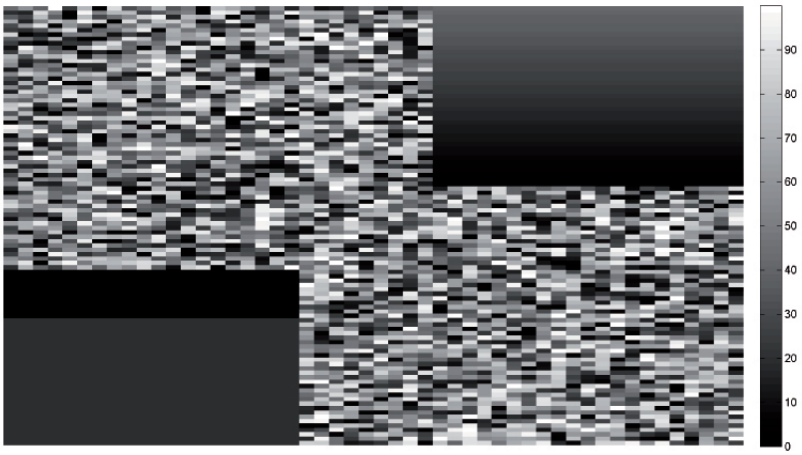


Figure 5.4: The simulated Data Matrix

$$f_2 = \sum_{ij}(1 - u_{ij}), \tag{5.2}$$

The function (5.2) maximizes the bicluster cardinality and the function (5.1) minimizes the bicluster error. So I solve the task of the parameters λ and μ initialization.

6. repeat the steps 1-5 q times.

5.5.1 Results

Simulated data set

First, I analyze simulated data set. I apply PBC, PBC with Bagging and my algorithm to simulated data matrix 100 $\times$ 50, with two biclusters. The data matrix has values from 0 to 100 see Fig. 5.4

I run PBC with following threshold: for a = 0.5, for b = 0.6 and find one bicluster of a cardinality 60×25 and MSR 559.1255. It can be seen that

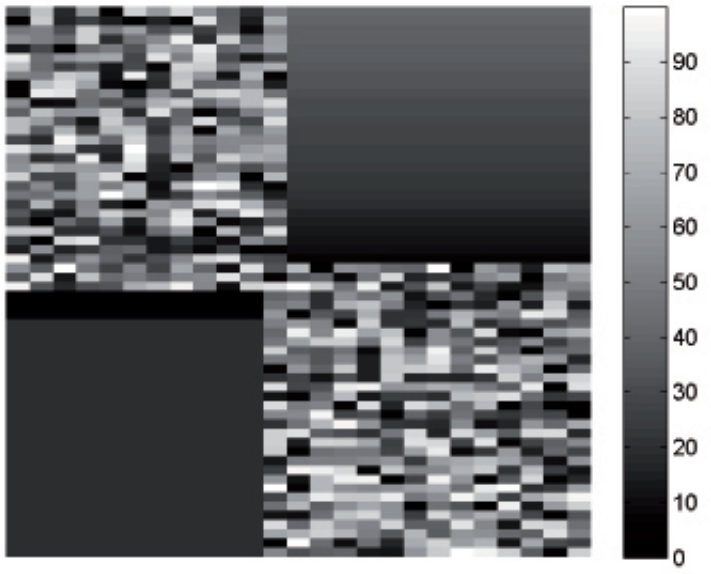
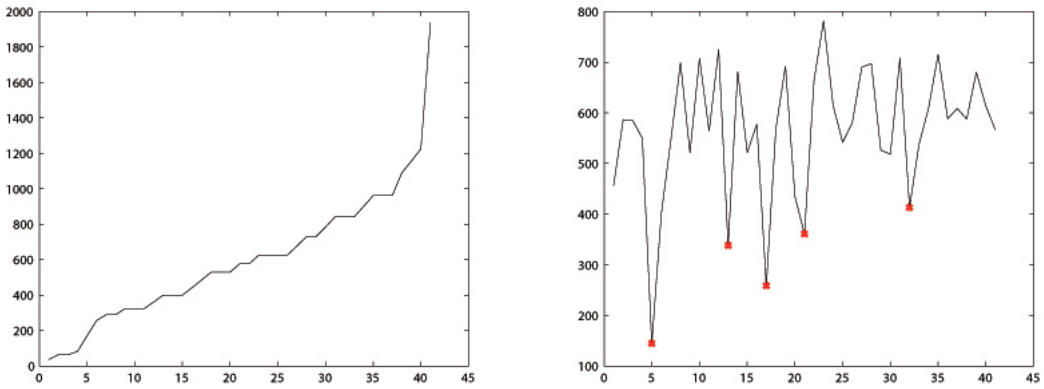


Figure 5.5: PBC algorithm. Heatmap of the resulting bicluster.

resulting bicluster of PBC contains both theoretical biclusters and does not separate them. Heatmap of the result can be seen in the Figure 5.5.

Now I apply PBC with Bagging. I run this algorithm 100 times with the thresholds for a = 0.5, b = 0.6 and ε = 0.001. Then I get the biclusters with overlap less than 70\%. As a result 41 bicluster remind. Now I sort biclusters respect to the number of elements and save those with a better MSR. Such, I get 5 overlapped bicluster that describe only one of the theoretical biclusters, see Fig.5.6 and Fig.5.7

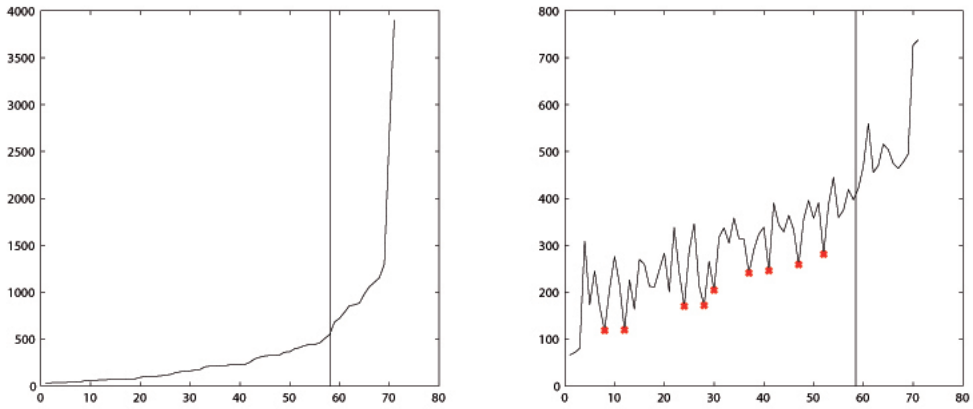
I run my algorithm 100 times with the following parameters: ε = 0.001, threshold for memberships b = 0.6, threshold for membership a = 0.5. The crossover and mutation probabilities I select as 0.75 and 0.03. In such case I obtain 72 biclusters with more than 30 elements. Then I sort biclusters with respect to the number of elements. Resulting number of



Number of elements (left), MSR (right) for the biclusters after sorting, PBC with Bagging

Figure 5.6: Result for PBC with Bagging

Figure 5.7: PBC with Bagging. Resulting bicluster.



Number of elements (left), MSR (right) of the biclusters after sorting

Figure 5.8: Result of PBC with Resempling

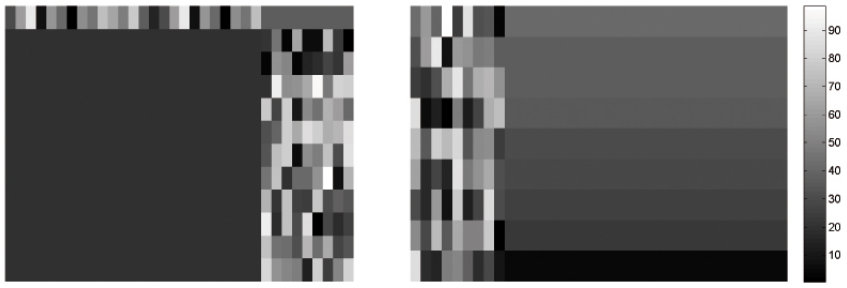
elements and MSR can be seen in the Fig.5.8.

As a result, I obtain 9 significant biclusters with MSR less than 280. It can be seen from the Fig.5.8 (left) that cardinality has a jump on the point (58; 546) (it is shown by a vertical line on the figure). From the first 58 bicluster I choose the best one with respect to MSR. These biclusters have the value of overlap less than 70\%. As a result two biclusters with the following parameters are found: for the first one number of elements is 324, MSR = 258.4387; for the second one number of elements is 408, MSR = 280.6964, see Fig.5.9.

Yeast data

I analyze Yeast microarray data, available on [8]

It can be seen that Yeast data is a collection of 2884 genes (attributes) under 17 conditions (time points), with three pairs of equal rows. The data have 34 null entries with -1 indicating the missing values. All entries are



Result for the first (left), and second (right) biclusters. Simulated data matrix.

Figure 5.9: PBC with resempling. Simulated data matrix.

integers lying in the range of 0 to 600. The missing values are replaced by random number between 0 and 800. For my algorithm I use following parameters: number of runs = 50, ε = 0.001, threshold for membership b = 0.6, and analyze the cases with a threshold of membership a equal to 0.8, 0.85 and 0.9. The crossover and mutation probabilities I select as 0.75 and 0.03, but it was noticed that these parameters had insignificant effect on the results. For every run I find one bicluster with some value of overlap and save the biclusters with the value of overlap less than 70%. For that biclusters I find all IDs that have known name of the genes and obtain new reduced biclusters. I analyze them by using DAVID tool, that is available on http://david.abcc.ncifcrf.gov/. DAVID provides typical batch annotation and gene-GO term enrichment analysis to highlight the most relevant GO terms associated with a given gene list. For every bicluster I make Functional Annotation Chart Report, that lists annotation terms and their associated genes, with the minimal number of genes equal to 10 and threshold for Fisher Exact P-Value 0.05. Next I obtain results for such term: Category, Term, Count (number of genes that en-

ter to GO class), PValue, Genes List, Fold Enrichment, Benjamini. From these results I save the biclusters with values of Benjamini less than 0.05. Results for any threshold can be seen following.

Threshold 0.8

After the first running I obtain 34 biclusters with the value of overlap less than 70%. The number of known genes vary in the interval from 23 to 141 for every bicluster. DAVID tool discovers 22 significant GO classes with Benjamini$<= 0.05$. From the biclusters that define the same GO class I get those with the smallest value of Benjamini. And obtain 8 different biclusters that describe 22 significant GO classes. The results can be seen in the Table 5.1. To visualize the Arbitrary GO Graphs containing the imputed GO terms and their closure to the root I use AmiGO tool, that is able on http://www.geneontology.org/. The Graphic for the threshold 0.8 is presented in the

Fig. 5.11.

Threshold 0.85

First, I obtain 31 biclusters with the value of overlap less than 70%. The number of known genes vary in the interval from 6 to 52. DAVID tool discovers 8 significant GO classes with Benjamini $<= 0.05$. From the biclusters that define the same GO class I get those with the smallest value of Benjamini. And obtain 5 different biclusters that describe 8 significant

GO classes. The results can be seen in the Table 5.2. The Graphic for the threshold 0.85 is presented in the

Fig. 5.12.

Threshold 0.9

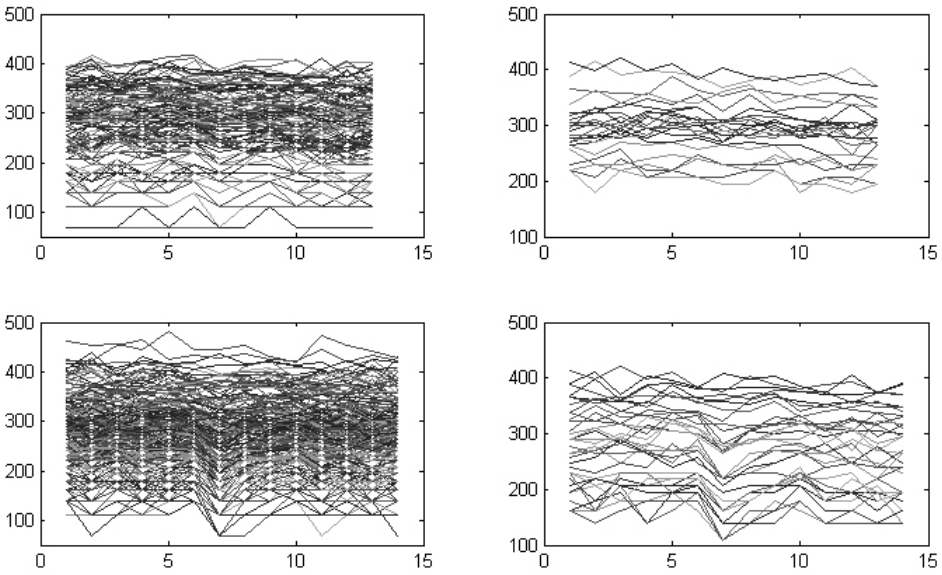
After running of the algorithm I obtain 21 biclusters with the value of overlap less than 70%. The number of known genes vary in the interval from 2 to 24. DAVID tool discovers 4 significant GO classes with Benjamini $<= 0.05$. From the biclusters that define the same GO class I get those with the smallest value of Benjamini. And obtain 2 different biclusters that describe 4 significant GO classes. The results can be seen in the Table 5.3. The Graphic for the threshold 0.9 is presented in the

Fig. 5.13.

As can be seen, the threshold 0.8 gives the better result and discover more significant classes. In the Fig.5.10 two random biclusters from the 22 discovered are shown by plots of biclusters and plots of best significant GO class values.

Comparison with other methods

I compare my methods with two algorithms: PBC [15] and PBC with Bagging to see a good improvement.



Two random biclusters from the Yeast data: plots of biclusters (left), plots of best significant GO class values (right)

Figure 5.10: Two random biclusters from the Yeast data

PBC algorithm is good to find one bicluster of large cardinality and small MSR. I run this algorithm 50 times with the following parameters: threshold for a is 0.8 and threshold for b is 0.6. Overlap of all the biclusters is more than 70% so as a result I obtain only one bicluster of 542 genes. This bicluster contains 17 GO classes with Benjamini value less than 0.05. The larger of them is GO:0050789 regulation of biological process that contains 179 elements of my bicluster with Benjamini 0.041. It can be seen that PBC obtains the same six groups as my algorithm, but the values of Enrichment of PBC are always smaller. Results can be seen in the Table.5.4 and

Fig.5.14.

I run PBC with Bagging algorithm 50 times with the following parame-

ters: threshold a 0.8 and threshold of b 0.6. After 70% overlap controlling only 17 biclusters remind. Number of genes in these biclusters are in the interval from 67 to 208. I apply DAVID and get all the resulting groups with Benjamini less than 0.05. As a result I obtain 9 biclusters that describe 17 GO classes. 14 of these GO classes are the same that in my algorithm, but have always a smaller value of Enrichment. Results can be seen in the Table.5.5 and Fig.5.15.

5.5.2 Conclusion

I introduced a new algorithm, based on the PBC, GA and Bagging techniques. My algorithm is able to solve many biclustering problems, such that, for example, multi-biclustering solutions and initialization. It also helps to find the biclusters of appreciable dimension and high correlation. I can choose the values of a and b threshold to change the dimension of biclusters in all the cases after running the algorithm. It gives us the chance to select biclusters of desired cardinality. As can be seen, in the case of Simulated data with two biclusters I find both biclusters with little error. In the case of Yeast data the best result is found in the case of the 0.8 threshold of a. In such case I discover 22 significant GO classes. My algorithm finds biclusters better than PBC and PBC with Bagging, because it better avoids a blocking in local minimum and such permits to find biclusters of smaller dimension.

Table 5.1: Results for the threshold 0.8. Where SPK - SP PIR KEYWORDS, GC2 - GOTERM CC 2, USF - UP SEQ FEATURE, KP - KEGG PATHWAY, Enr - Fold Enrichment, Bon - Bonferroni, Ben - Benjamini, Cat - Category, N - Count, P - PValue.

Cat	Term	N	P	Enr	Bon	Ben
SPK	atp-binding	24	2e-5	2.6	3e-3	3e-3
GC2	0005622 intracellular	89	4e-3	1.1	0.1	0.035
GC2	0044424 intracellular part	89	3e-3	1.1	0.07	0.03
USF	nucleotide phosphate-binding region:ATP	18	3e-5	3.2	9e-3	9e-3
SPK	nucleotide-binding	26	2e-5	2.5	4e-3	2e-3
GC2	0044422 organelle part	25	5e-5	1.8	1e-3	1e-3
GC2	0044446 intracellular organelle part	25	5e-5	1.8	1e-3	1e-3
GC2	0043227 membrane-bounded organelle	30	3e-3	1.3	0.05	0.02
GC2	0043233 organelle lumen	11	9e-3	2.4	0.17	0.04
GC2	0030529 ribonucleoprotein complex	15	5e-4	2.8	9e-3	3e-3
GC2	0043229 intracellular organelle	46	3e-3	1.2	0.06	0.015
GC2	0030427 site of polarized growth	13	6e-3	2.5	0.14	0.05
SPK	activator	12	1e-3	3.3	0.17	0.05
SPK	transcription regulation	21	1e-3	2.2	0.19	0.04
SPK	nucleus	34	2e-4	1.8	0.03	0.01
SPK	Transcription	16	5e-4	2.7	0.07	0.03
GC2	0043234 protein complex	45	6e-5	1.7	2e-3	2e-3
SPK	protein biosynthesis	15	1e-4	3.4	0.02	7e-3
SPK	ribosomal protein	12	5e-4	3.5	0.08	0.02
SPK	ribosome	12	7e-5	4.4	0.01	6e-3
KP	sce03010:Ribosome	10	1e-3	3.4	0.04	0.04
GC2	0043228 non-membrane-bounded organelle	37	7e-4	1.7	0.02	6e-3
SPK	phosphoprotein	69	8e-6	1.52	1e-3	1e-3

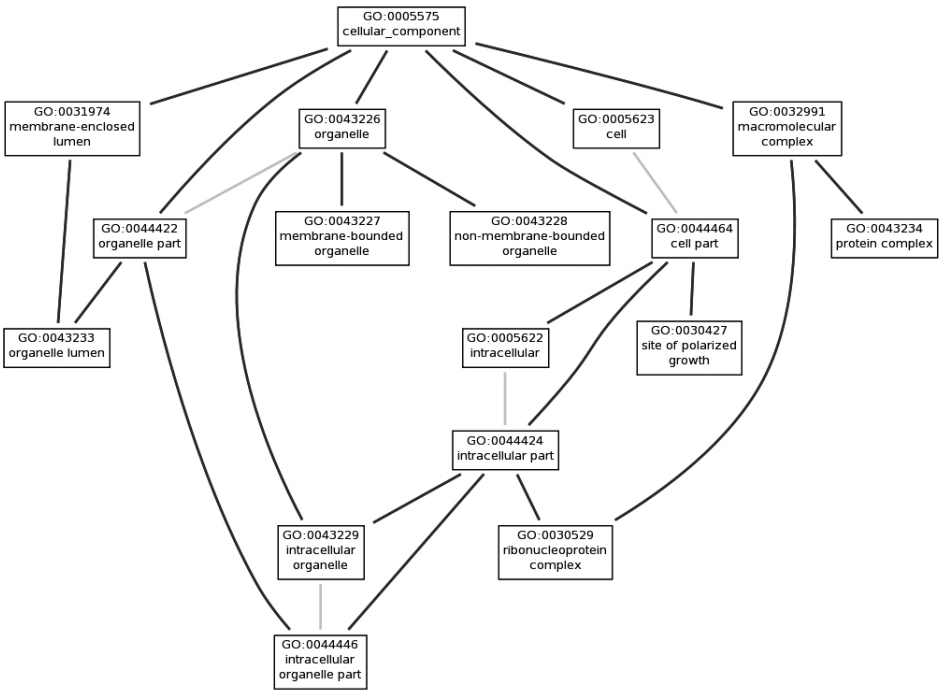


Figure 5.11: Arbitrary GO Graph for the case 0.8

Table 5.2: Results for the threshold 0.85. Where SPK - SP PIR KEYWORDS, GC2 - GOTERM CC 2, USF - UP SEQ FEATURE, KP - KEGG PATHWAY, Enr - Fold Enrichment, Bon - Bonferroni, Ben - Benjamini, Cat - Category, N - Count, P - PValue.

Cat	Term	N	P	Enr	Bon	Ben
SPK	nucleus	23	2e-4	2.1	0.02	9e-3
GC2	0043234 protein complex	16	1e-4	2.6	3e-3	3e-3
SPK	phosphoprotein	20	3e-4	1.9	0.03	0.03
GC2	0044422 organelle part	19	8e-5	1.99	2e-3	2e-3
GC2	0044446 intracellular organelle part	19	8e-5	1.99	2e-3	2e-3
GC2	0043227 membrane-bounded organelle	22	2e-3	1.4	0.04	0.02
GC2	0043229 intracellular organelle	23	1e-3	1.4	0.02	0.02
SPK	Transcription	10	5e-4	3.9	0.045	0.045

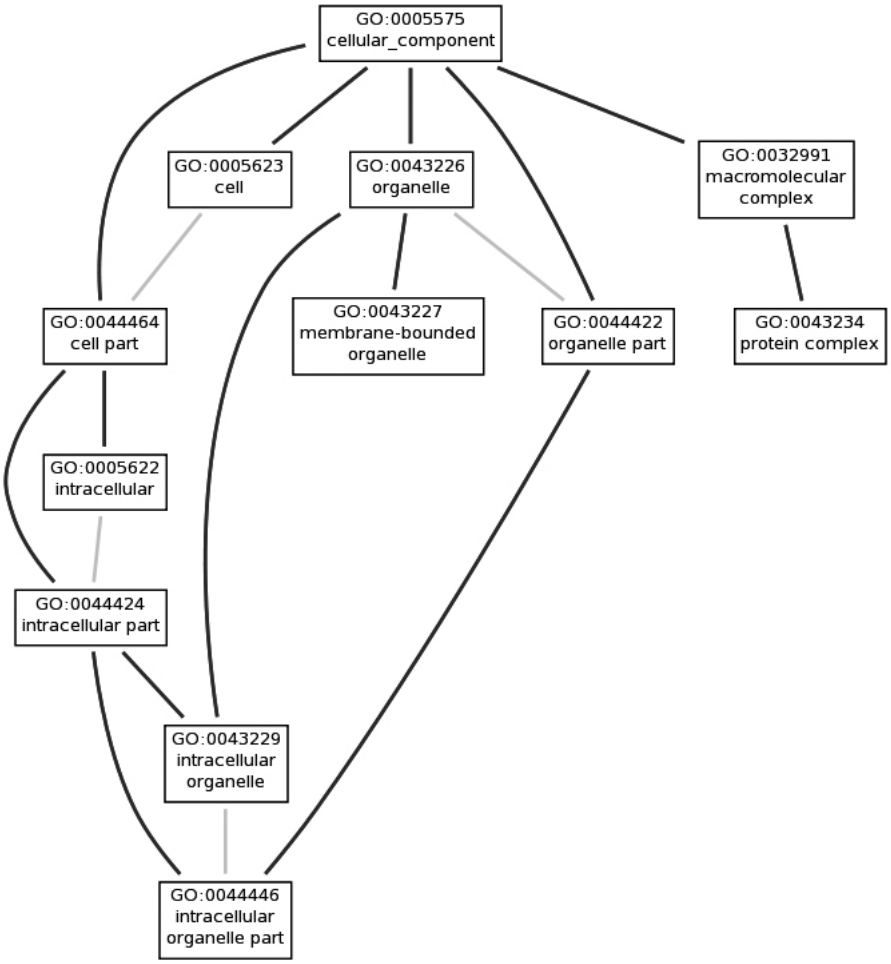


Figure 5.12: Arbitrary GO Graph for the case 0.85

Table 5.3: Results for the threshold 0.9. Where SPK - SP PIR KEYWORDS, GC2 - GOTERM CC 2, USF - UP SEQ FEATURE, KP - KEGG PATHWAY, Enr - Fold Enrichment, Bon - Bonferroni, Ben - Benjamini, Cat - Category, N - Count, P - PValue.

Cat	Term	N	P	Enr	Bon	Ben
SPK	phosphoprotein	13	7e-4	2.13	0.04	0.04
GC2	0043234 protein complex	12	3e-3	2.39	0.05	0.03
GC2	0044422 organelle part	17	9e-4	1.86	0.02	0.02
GC2	0044446 intracellular organelle part	17	9e-4	1.86	0.02	0.02

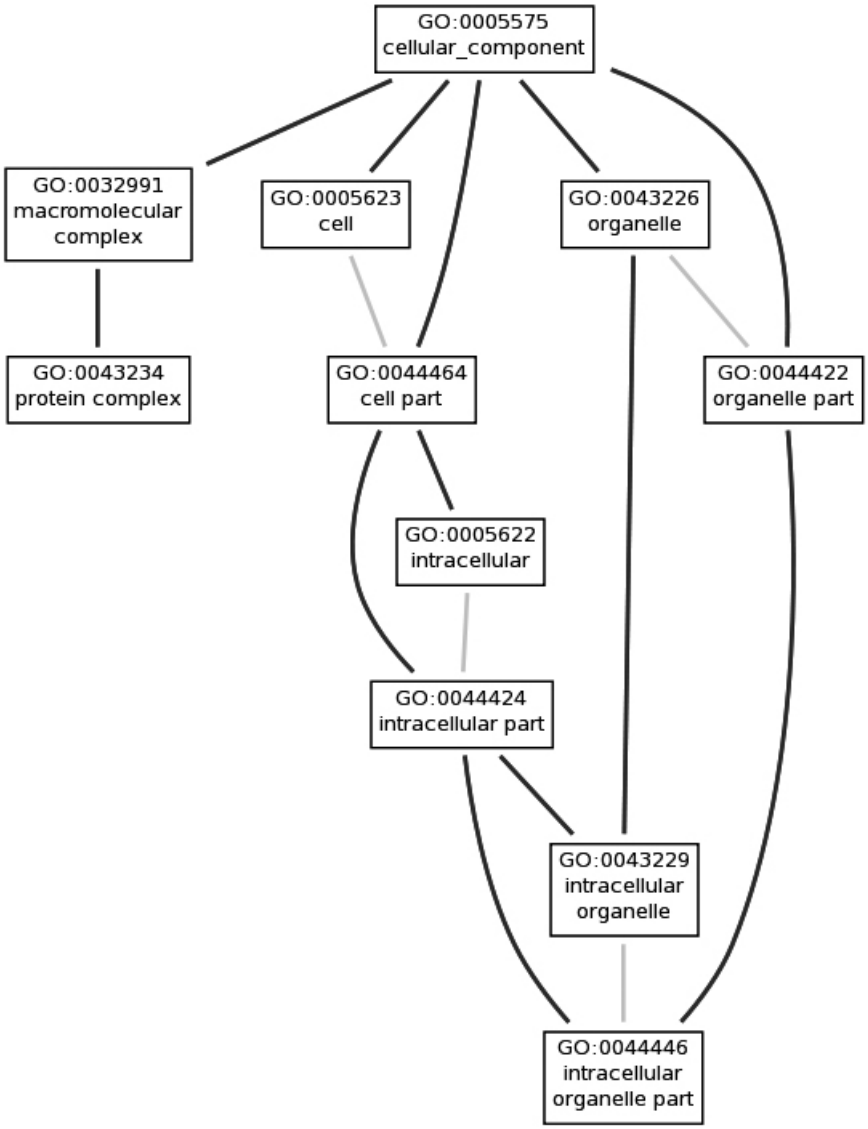


Figure 5.13: Arbitrary GO Graph for the case 0.9

Table 5.4: Results for the PBC method with the threshold 0.8. Where SPK - SP PIR KEYWORDS, GM2 - GOTERM MF 2, GC2 - GOTERM CC 2, GOTERM BP 2 - GB2, USF - UP SEQ FEATURE,Enr - Fold Enrichment, Ben - Benjamini, Cat - Category, N - Count, P - PValue.

Cat	Term	N	P	Enr	Ben
SPK	acetylation	28	1e-4	2.21	0.003
SPK	ATP	36	2e-4	1.93	0.004
SPK	dna-binding	45	0.002	1.57	0.045
SPK	dna-directed rna polymerase	11	2e-4	4.19	0.004
GM2	GO:0000166 nucleotide binding	112	0.002	1.3	0.035
GC2	GO:0005933 cellular bud	30	0.002	1.78	0.008
GB2	GO:0022613 ribonuc. complex biogenesis	60	7e-6	1.52	0.02
GC2	GO:0030427 site of polarized growth	34	0.003	1.66	0.01
GC2	GO:0043233 organelle lumen	94	6e-4	1.38	0.002
GB2	GO:0050789 regulation of bio. process	179	0.002	1.2	0.04
GB2	GO:0051236 establishment of RNA localization	24	5e-4	2.16	0.03
USF	mutagenesis site	86	3e-5	1.54	0.02
USF	nucleotide phosphate-binding region:ATP	51	2e-4	1.7	0.04
SPK	nucleotidyltransferase	19	0.001	2.32	0.02
SPK	nucleus	165	4e-5	1.3	0.001
SPK	Transcription	63	2e-4	1.59	0.005
SPK	transferase	70	0.002	1.42	0.035

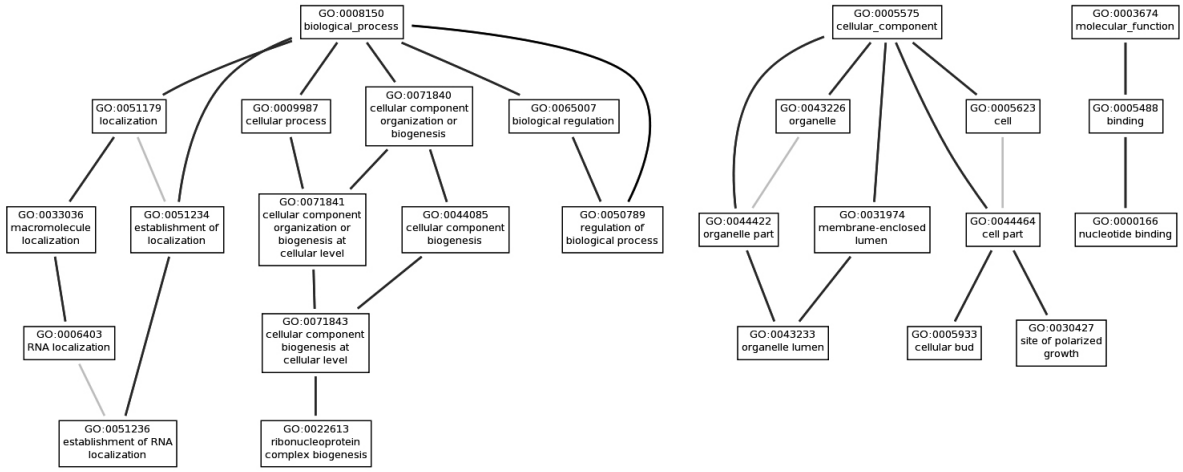


Figure 5.14: Arbitrary GO Graph for the PBC case

Table 5.5: Results for the PBC with Bagging. Where SPK - SP PIR KEYWORDS, GC2 - GOTERM CC 2, GM2 - GOTERM MF 2, Enr - Fold Enrichment, Bon - Bonferroni, Ben - Benjamini, Cat - Category, N - Count, P - PValue.

Cat	Term	N	P	Enr	Ben
SPK	atp-binding	28	1e-5	2.4	0.003
GC2	GO:0005622 intracellular	85	0.002	1.01	0.012
GC2	GO:0043229 intracellular organelle	78	3e-4	1.21	0.004
SPK	nucleus	61	6e-5	1.58	0.006
GC2	GO:0030529 ribonucleoprotein complex	29	1e-4	2.13	0.001
GC2	GO:0043234 protein complex	51	8e-5	1.66	0.001
SPK	protein biosynthesis	16	2e-4	3.02	0.014
GM2	GO:0003735 struct. constituent of ribosome	15	0.001	2.67	0.04
SPK	ribonucleoprotein	20	2e-5	3.13	0.001
SPK	ribosome	13	2e-4	3.66	0.007
GC2	GO:0043228 non-membrane-bounded organelle	44	1e-4	1.62	0.001
SPK	nucleotide-binding	29	7e-5	2.19	0.006
SPK	phosphoprotein	84	7e-6	1.46	0.001
GC2	GO:0044422 organelle part	63	5e-5	1.47	0.001
GC2	GO:0044446 intracellular organelle part	63	5e-5	1.47	0.001
SPK	cytoplasm	59	0.002	1.43	0.047
GC2	GO:0044424 intracellular part	174	0.002	1.08	0.01

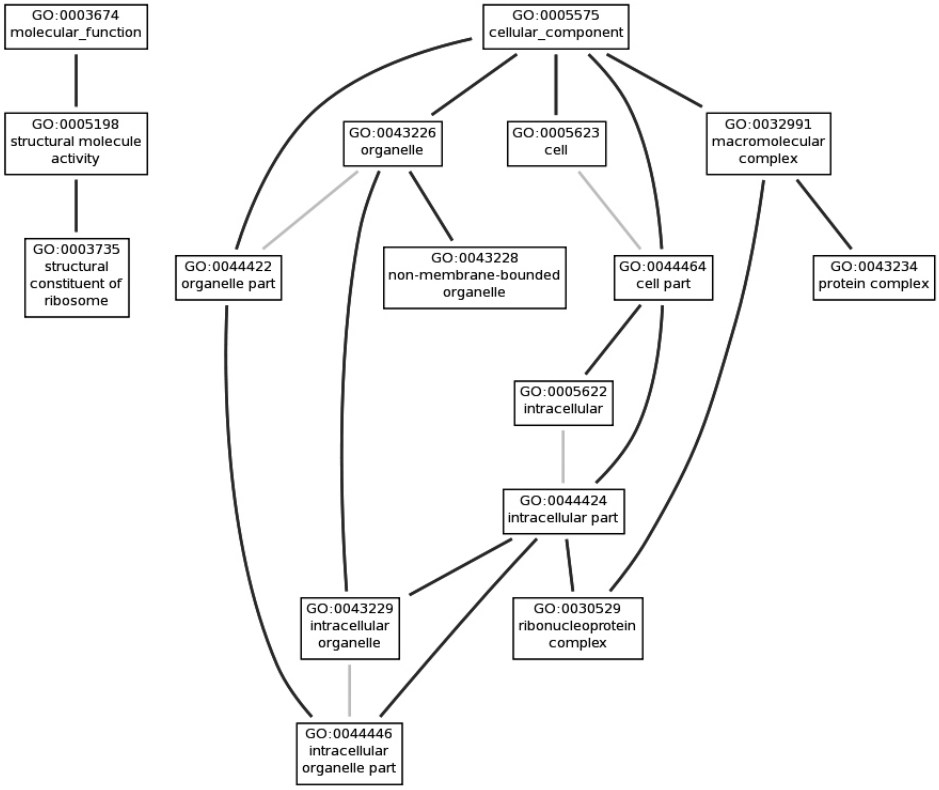


Figure 5.15: Arbitrary GO Graph for the PBC with Bagging

Chapter 6

Conclusion

In the last years a large amount of information about genomes was discovered, increasing the complexity of analysis. Therefore the most advanced techniques and algorithms are required. In many cases researchers use unsupervised clustering. But the inability of clustering to solve a number of tasks requires new algorithms. So, recently, scientists turned their attention to the biclustering techniques.

In this thesis I propose two novel biclustering techniques such that Combinatorial Biclustering Algorithm (CBA) and Improved PBC.

CBA permits to solve the following problems: 1) classification of data with respect to rows and columns together; 2) discovering of the overlapped biclusters; 3) definition of the minimal number of rows and columns in biclusters; 4) finding all biclusters together. I apply this model to synthetic and real biological data sets and show the results. CBA is an accurate technique that permits to find, in all examined cases, a good classifica-

tion of data. This algorithm reaches the maximal precision in the data sets analysis. In every experiment I a-priory decided the maximal error and the minimal dimension of the desired biclusters.

In the case of Improved PBC my aim was to find some method that permits to separate microarray data and requires low number of initial conditions. As a base of my algorithm I used a PBC algorithm, proposed in [15]. PBC finds one bicluster at a time, assigning a membership to a bicluster for each gene and condition. Filippone et al. try to maximize the size of a bicluster and minimize the residual. This algorithm blocks in the local minimum and for this reason does not give a multi-biclustering solutions. In this thesis I try to merge the Genetic Algorithms and Bagging techniques to solve the cited problems. In my case I choose an acceptable multi-objective functions to avoid the initialization of λ and μ. And use the GAs to variate the PBC bagging solutions. I apply my technique to synthetic and Yeast data and make the comparison with other techniques. It can be seen, that in all examined cases, Improved PBC shows a better results than an other algorithms.

Bibliography

[1] M. Zvelebil, J. Baum, Understanding Bioinformatics, New York : Garland Science, 2008.

[2] An Introduction to Genomics: The Human Genome and Beyond, Department of Energy's Joint Genome Institute and the Technical & Electronic Information Department, Lawrence Berkeley National Laboratory, with support from the Department of Energy Office of Science, Biological and Environmental Research Program.

[3] M. Matteo, A Tutorial on Clustering Algorithms, http://home.dei.polimi.it/matteucc/Clustering/tutorial_html/

[4] J.B. MacQueen, Some Methods for classification and Analysis of Multivariate Observations, Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability, 1967 , 281–297.

[5] J.H. Ward, Hierarchical Grouping to Optimize an Objective Function, Journal of the American Statistical Association 58 (301), 1963, 236–244

[6] J.A. Hartigan. Direct clustering of a data matrix, Journal of the American Statistical Association 67 (1972), 123–129.

[7] B. Mirkin, Mathematical Classification and Clustering, Kluwer Academic Publishers, 1996.

[8] Y. Cheng and G. Church, Biclustering of expression data, Proc. Eighth Intl Conf. Intelligent Systems for Molecular Biology(ISMB 00), 93–103, 2000.

[9] L. Lazzeroni and A. Owen, Plaid Models for Gene Expression Data, technical report, Stanford Univ., 2000.

[10] Y. Kluger, R. Basri, J. T. Chang, and M. Gerstein, Spectral biclustering of microarray cancer data: co- clustering genes and conditions, Genome Res., 13 (2003), 703–716.

[11] A. Prelic, S. Bleuler, P. Zimmermann, A.Wille, P. Buhlmann, W. Gruissem, L. Hennig, L. Thiele, and E.Zitzler, A Systematic comparison and evaluation of biclustering methods for gene expression data, Bioinformatics, 22(2006), 1122–1129

[12] A. Tanay et al., Revealing modularity and organization in the yeast molecular network by integrated analysis of highly heterogeneous genomewide data, Proc. Natl Acad. Sci. USA, 2004, pp. 2981–2986.

[13] A. Tanay, R. Sharan and R. Shamir, Discovering statistically significant biclusters in gene expression data, Bioinformatics, vol. 18, 2002, pp.136- 144.

[14] A. Prelic, S. Bleuler, P. Zimmermann, A. Wille, P. Buhlmann, W. Gruissem, L. Hennig, L. Thiele, E. Zitzler, A systematic comparison and evaluation of biclustering methods for gene expression data, Bioinformatics, 2006, 22 (9), 1122–1129,

[15] M. Filippone, F. Masulli, S. Rovetta, S. Mitra, and H. Banka, Possibilistic approach to biclustering: An application to oligonucleotide microarray data analysis, Lecture Notes in Bioinformatics, C. Priami, Ed., Springer, vol. 4210, pp. 312-322, 2006.

[16] R. Krishnapuram and J.M. Keller. "A possibilistic approach to clustering". IEEE Transactions on Fuzzy Systems, vol. 1 no. 2, pp. 98–110, 1993.

[17] J. A. Bondy, U. S. R. Murty, Graph Theory with Applications, North-Holland, 1976.

[18] G. Alexe, S. Alexe, Y. Crama, S. Foldes, P. Hammer, B. Simeone, Consensus algorithms for the generation of all maximal bicliques, Discrete Appl. Math. 145 (1) (2004), 11-21.

[19] A. Ciaramella, S. Cocozza, F. Iorio, G. Miele, F. Napolitano, M. Pinelli, G. Raiconi, R. Tagliaferri, Clustering, Assessment and Validation: an application to gene expression data, Proceedings of International Joint Conference on Neural Networks, Orlando, Florida, USA, 12–17.

[20] R. A. Fisher, Statistical Methods for Research Workers, Oliver and Boyd, 1954

[21] L. Ottini, M. Falchetti, R. Lupi, et al. Patterns of genomic instability in gastric cancer: clinical implications and perspectives. Ann Oncol 17(Suppl. 7): pp. 97–-102, 2006.

[22] R. Krishnapuram, J. Keller, The possibilistic c-means algorithm: insights and recommendations, IEEE Transactions on Fuzzy Systems, vol. 4, no. 3, pp. 385–393, 1996.

[23] F. Masulli, A. Schenone, A fuzzy clustering based segmentation system as support to diagnosis in medical imaging, Artificial Intelligence in Medicine, vol. 16, no. 2, pp. 129–147, 1999.

[24] L. Breiman, Bagging Predictors, Technical Report No. 421, 1994.

[25] M. S. Garey and D. S. Johnson, Computers and Intractability: A Guide to NP-Completeness. W.H. Freeman, New York, 1979

[26] M. Dawande, P. Keskinocak, S. Tayur, On the Biclique problem in Bipartite graphs, GSIA Working Paper 1996-04, Pittsburgh.

[27] L. A. Zadesh, Fuzzy sets as a basis for a theory of possibility, Fuzzy Sets and Systems, vol. 1, pp.3-28,1978

[28] H. J. Zimmerman and P. Zysno, Quantifying vagueness in decision models, European J. Operational Res., vol. 22, pp 148-158, 1985.

[29] J. C. Bezdek, Pattern recognition with Fuzzy Objective Function Algorithms, New York: Plenum Press, 1981.

[30] J. H. Holland. Adaptation in natural and artificial systems. University of Michigan Press, Ann Arbor, 1975.

[31] K. Deb, Multi-Objective Optimization using Evolutionary Algorithms, Wiley, London, 2001.