Using Mutual Information for extracting Biclusters from Gene Expression Data

A thesis submitted to University of Delhi

for the award of the degree of

Doctor of Philosophy

by

Seema Aggarwal

Department of Computer Science
University of Delhi
Delhi-110007, India

February, 2013

Using Mutual Information for extracting Biclusters from Gene Expression Data

A thesis submitted to University of Delhi

for the award of the degree of

Doctor of Philosophy

by

Seema Aggarwal

Department of Computer Science
University of Delhi
Delhi-110007, India

February, 2013

Declaration

The thesis entitled "*Using Mutual Information for extracting Biclusters from Gene Expression Data*", which is being submitted for the award of the degree of Doctor of Philosophy is a record of original and bona fide research work carried out by me in the Department of Computer Science, University of Delhi, Delhi, India.

The work presented in this thesis has not been submitted to any other university or institution for any academic award.

Seema Aggarwal
Department of Computer Science,
University of Delhi,
Delhi, India.

Certificate

This is to certify that the thesis entitled “*Using Mutual Information for extracting Biclusters from Gene Expression Data*” being submitted by Seema Aggarwal in the Department of Computer Science, University of Delhi, Delhi, for the award of degree of Doctor of Philosophy is a record of original research work carried out by her under the supervision of Dr. Neelima Gupta.

The thesis or any part thereof has not been submitted to any other University or institution for any academic award.

Supervisor
Neelima Gupta
Department of Computer Science
University of Delhi
Delhi, India.

Head
Department of Computer Science
University of Delhi
Delhi, India.

Acknowledgment

I thank God Almighty for his blessings so that I could complete this herculean task.

There are no words that can express my gratitude to my guide Dr. Neelima Gupta. I thank her for her invaluable guidance, continous support and patient listening. Her continous motivation, encouragement helped in getting the best out of me. Her expertise and immense knowledge acted like a steering on the circuitous road of my study.

I am thankful to Head and all faculty members, Department of Computer Science, Delhi University for thier guidance and support. My sincerest thanks also go to Dr. Pratibha Jolly, principal Miranda House. I would also like to thank Prof. Sanjay Jain and Arijeet, Department of Physics, University of Delhi, for sparing time for long discussions. Thanks are also due to Ms. Janaki Chintalapati of CDAC, Pune for her useful suggestions and timely help. I would also like to thank Amit, Ishan and Surubhi for helping me with the experimental results. I also express my sincerest thanks to Geeta Gupta who gave me both academic and emotional support.

I am highly thankful for the support and cooperation that I got from my family and all my friends.

Last but not the least, my sincere thanks also go to support staff at Department of Computer Science, Delhi Univeristy and Miranda House who helped me throughout the studies.

Seema Aggarwal

Abstract

With the advances in DNA microarray technology, expression levels of a large number of genes can be measured in parallel. This leads to generation of huge amount of gene expression data. Analysis of the microarray data to extract biologically significant information from it is a challenging task. Many people have used biclustering for this analysis. Most of these biclustering algorithms are based either on the Euclidean distance or correlation coefficient as the similarity measure. These measures capture only linear relationships between the genes but non linear dependencies may exist amongst them. Different measures are required to capture complete dependence (i.e. both linear and nonlinear). Mutual information provides a more general measure to investigate relationships (positive, negative correlation and non linear relationships as well). As it depends upon the distribution and not the actual values, no normalization of the data is required. It works well for both scaled and shifted data. The measure is also robust towards noise and outliers. In our work we propose a set of algorithms based on an approach using mutual information for extracting biclusters from gene expression data. To the best of our knowledge, none of the existing algorithms for biclustering have used mutual information as a similarity measure between two genes or conditions. Experiments were conducted on synthetic data and expression data of *Arabidopsis thaliana, Saccharomyces cerevisiae, Human breast cancer* data and *Diffuse large B cell lymphoma* data. We were able to extract the implanted biclusters from the synthetic datasets. On real datasets our experiments show the effectiveness of our algorithms in extracting biclusters from gene

expression datasets. We used DAVID, an online biological tool for the validation of our biclusters. It was found that the obtained biclusters were biologically more significant as compared to the biclusters obtained by the other existing algorithms. We also studied the promoter regions of the genes belonging to a bicluster for common patterns / Transcription Factor Binding Sites (TFBS) or motifs using another online biological tool named RSA Toolbox. Promoter regions of the genes of most of the biclusters were found to have common motif patterns. We believe our work delivers relevant information and provides a useful tool for the analysis of gene expression datasets.

Contents

List of Tables

List of Figures

x

List of Algorithms

Chapter 1

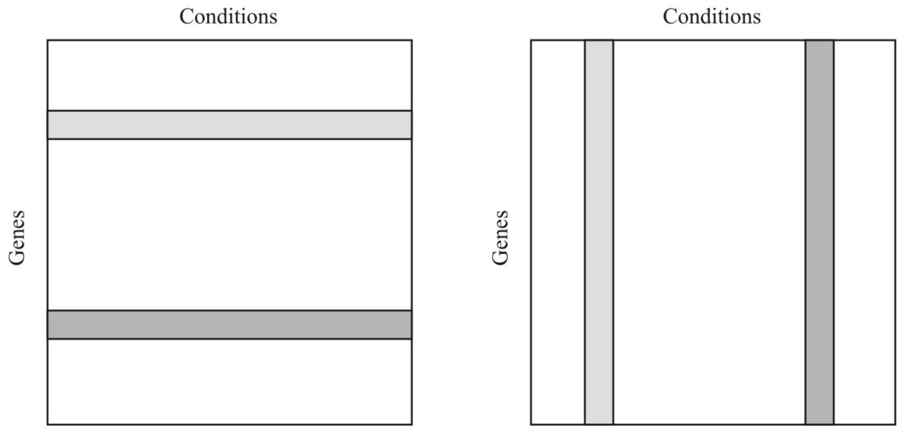
Introduction

Last two decades have seen lot of advances in the area of genomic research. Technological developments like *microarray chip* have made it possible to study the behaviour of all the genes in an organism in a single experiment in contrast to the previous methods of studying *one gene in one experiment*. Eric Lander [Lan05] and others [ZWD$^+$04, AMK00, Kau93] have also suggested that the simultaneous study of all the genes of an organism is important to decipher the logic of gene regulation in the organism. With the help of microarray experiments, expression level of thousands of genes can be monitored simultaneously [BDSY99, Dom03, DIB97, Slo02a]. The expression data has been collected for different organisms (healthy and diseased) during different developmental stages, changing environmental/chemical/clinical conditions or at different time points. This has led to generation of vast amount of data. Development of efficient computational tools for the analysis of this huge amount of data, to be able to extract biologically relevant information from it, is the need of the hour. Extracting meaningful information from this humongous data poses a great challenge to biologists as well as to the community of researchers in the field of computation. Biologists are often interested in identifying the set of genes responsible for a particular biological activity, for example the genes controlling the formation of a protein. They may also like to know the genes causing stress, high blood pressure, diabetes, heart ailment, tumor or AIDS. In plants, these activities include reproduction, growth of a particular part of a plant, photosynthesis and absorption of nutrients from soil. The genes responsible for a particular biological activity get triggered under certain conditions. For example absorption of nutrients from the soil is more in the presence of sunlight and appropriate amount of moisture in the soil. Clearly, the genes responsible for the activity get triggered under these conditions. Therefore, the expression pattern of these genes must have some sort of association amongst themselves. Vilo et al. [VBJ$^+$00] and Dhaeseleer et al. [DWFS98, DLS99] hypothesize that genes with similar expression profile may share something common in their regulatory mechanism. Once the genes responsible for a disease are known, the conditions that affect the expression of these genes and other genes affected by the same conditions can be discovered. Scientists have used the microarray technology to develop new methods of diagnosis and treatment for a number of diseases and response to drug treatments. The recognition of coordinated expression between the genes helps to draw inferences about functions of unknown genes, to form hypothesis regarding potential pathways of information flow or to infer a model of a gene network. As all the organisms are related through similarities in their DNA sequences, information from the analysis of data of one genome may also help in understanding the concepts about other organisms.

The task of a biologist is greatly simplified if one can extract a couple of hundreds or fewer genes showing a pattern or association in their expression values from a data consisting of tens of thousands of genes. Clustering has been found to be a useful tool in the field of analysis of gene expression data. The traditional clustering algorithms [BDSY99, ESBB98, THC$^+$99, Cla99] cluster the genes based on their expression under all the conditions as shown in Figure 1.1(a). The resultant gene clusters consists of genes with similar expression, whereas genes with dissimilar expression fall in different clusters. This helps in finding meaningful associations in microarray data and the underlying biological processes. Clustering of related genes helps to understand regulatory inputs and functional pathways. Genes belonging to a cluster are expected to perform similar biological tasks and enriched with functional categories. Thus clues to unknown gene functions may be inferred from the function of known genes in the same cluster. Genes belonging to the same cluster not only perform similar functions but they are also controlled by similar control factors. Thus, they may have common regulatory elements **(motifs)** in their promoter regions. Once these regulatory elements are identified, the entire transcriptional regulatory network may be understood [ZTOT04]. Similarly, clustering conditions over the expression levels of all genes as shown in Figure 1.1(b) helps in defining new disease subclasses. Grouping of patients samples based on disease subtype or response to treatment may lead to distinguishing similar looking diseases.

As the problem of clustering is NP hard most of the algorithms for clustering are heuristic in nature. Tavazoie et al. [THC$^+$99] used clustering to form gene clusters in the data set of Saccharomyces Cerevisiae. Further they searched for upstream DNA sequence patterns specific to each cluster. They identified 18 biologically significant DNA motifs in the promoter region of gene clusters. Eisen et al. [ESBB98] used correlation based hierarchical algorithm to analyse a 12 point time course of the serum response of 8600 human genes and a 75 condition expression study of the yeast genome. They concluded that clustering of gene expression data results in groups of genes with similar functions. Califano et al. [CST00] analyzed the human cancer cell data to identify patterns of gene expression that can be used to predict cell phenotype. They related the gene expression data with different cancer cell morphologies.

These traditional clustering algorithms work well for small data sets but fair poorly when the number of experimental conditions is large. All genes appear to be equidistant from each other for a large number of conditions. These algorithms give equal weights to all the conditions while computing the similarity amongst the genes. However, the cellular processes are generally affected only by a small subset of conditions [IFB$^+$02, BIB03]. Most of the other conditions which do not contribute to the cellular process add to the



(a) Gene clusters (b) Condition clusters

Figure 1.1: Traditional Clustering

background noise. Consider an expression data consisting of expression of genes present in tissues of different parts of the plant like flower, root, leaf and stem. Certain genes may exhibit good associations under tissues of one particular part say flower whereas the same set of genes may have poor associations when considered over the entire set of tissues drawn from different parts of plant. Thus, some relationships and associations may be overlooked while clustering over all the conditions. Moreover, these traditional clustering algorithms compute non-overlapping clusters i.e. a gene belongs to at most one cluster whereas in fact a gene may be responsible for several cellular activities and hence may belong to more than one cluster [CC00, IFB$^+$02, BIB03, GLD00, KTW05, LW07, KBCG03, JTZ04].

Cheng and Church in [CC00] introduced the notion of **biclustering** for gene expression data. **Biclustering** refers to simultaneous clustering of genes and experimental conditions in the gene expression matrix. It allows biclusters to overlap both on genes as well as on conditions as shown in Figure 1.2. Relationships amongst genes which are active over some but not all the conditions can be uncovered with the help of biclustering.

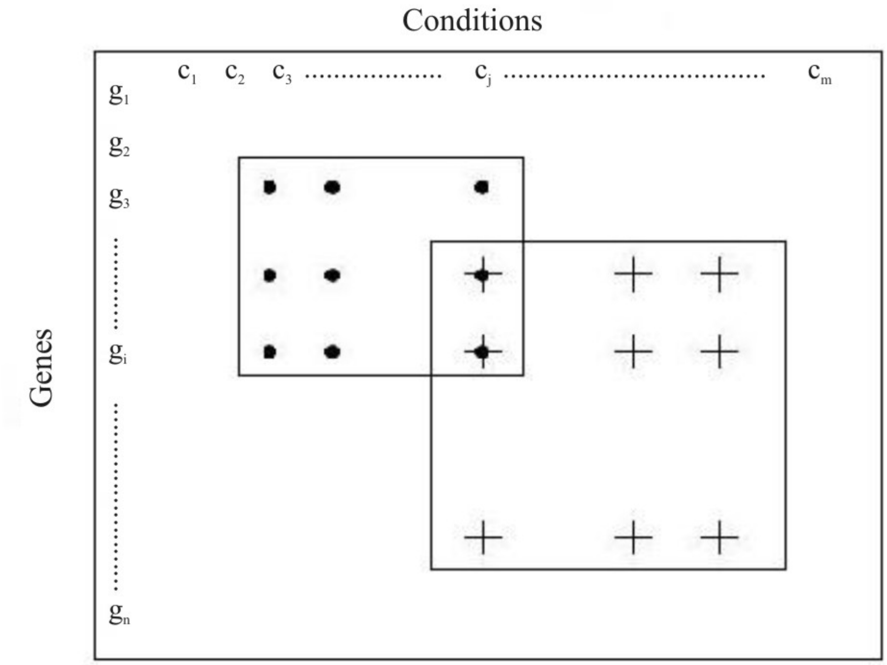


Figure 1.2: Overlapping Biclusters

It aims to identify groups of genes that show **similar expression pattern under a subset of experimental conditions** that cannot be found by classical clustering approaches ([PBZ$^+$06, MO04, HBH$^+$10]). Infact, biclustering is emerging as a standard tool for extracting knowledge from gene expression data as it fits better to biological behavior.

Biclustering provides a better visualization of the gene expression data. All the information inferred from traditional gene clusters can also be inferred from the biclusters. However, biclusters provide us more information as they also give the samples under which the genes of the biclusters behave similarly. For eg. in drug design, researchers want to study the effect of various compounds on gene expression. The effects may be similar only in a small subset of genes and may be dissimilar for others. As biclustering provides two way clustering it can be used to identify compounds having similar effect on a small group of genes.

Biclustering has its applications in areas other than computational biology like machine learning, pattern recognition, text mining and market data analysis. In the market data, the products bought by the customers are stored in a customer product table. There would rarely be customers who would have similar preferences over all the products. However, there would certainly be groups of customers who are more fond of skimmed milk products, sprouts, vegetables and fruits. These people may however differ drastically in their choice over other products like clothes. Traditional clustering may put these people in different clusters as they differ in lot of other choices. However, biclustering on this data will identify a group of customers with similar choices for a subset of products rather than all the products. Having identified a group of products and a group of people interested in these products, promotional schemes and advertising may be targeted for them to increase sales.

Various similarity measures have been used in literature to quantify the similarity in the expression level of two genes. Different similarity measures extract different patterns in the expression data. In other words, result of any clustering/biclustering algorithm depends on the choice of the similarity measure. According to many researchers [DWFS98, DLS99, DPFS00, VBJ$^+$00], the choice of similarity measure is as important as the choice of the algorithm itself. As different elements are influenced by different aspects of regulatory mechanism, there is no single choice of similarity measure that provides all the relevant biclusters in the data. Each measure produces a unique clustering of the expression patterns and one must be selected according to the type of interactions user would like to capture from the expression data [Cla99]. The interaction may vary from a simple linear association to more complex like quadratic, sinusoidal or exponential etc. Similarity measures can be classified into three classes each indicating different regularities in the data: (a) similarity according to positive correlations which identifies similar or upregulated set of genes (b) similarity according to negative correlations which identifies downregulated set of genes (c) similarity based on mutual information which detects more complex relationships. Though (a) and (b) have been successfully and satisfactorily used for several years they extract only linear relationships. Though, some work on traditional clustering [DWFS98, DLS99, BK00, PMBG07, MCA+98] have suggested the use of mutual information for clustering, none of the existing biclustering algorithms have used mutual information as a measure of similarity for biclustering.

In this work we propose the use of mutual information as a measure of similarity amongst the genes for biclustering gene expression data. In [DMM03] Dhillon et al. and others [ST00, BDG+07] have used mutual information for co-clustering word document data. Their approach is entirely different from our approach. They do not use mutual information as a similarity measure between two pair of genes. They partition the input matrix into non overlapping partitions of both rows and columns such that the loss of information between the original and the partitioned matrix is the least. They produce coclusters which are strictly non overlapping. Also, their paper has its limitations especially with reference to gene expression data. Firstly the entries in the gene expression data cannot be treated as a measure of co-occurrence. Secondly, to treat the input matrix as a joint probability distribution the entries must be all positive which may not be the case in gene expression data as down-regulation may be represented by negative values. Banerjee et al. in [BDG+07] propose a generalized co-clustering algorithm which works for negative entries in the input matrix as well. They assume that the probability distribution of the input data is either predefined or follows uniform distribution. Both Banerjee and Dhillon identify non-overlapping biclusters whereas a gene may be responsible for more than one cellular function and thus may belong to more than one bicluster. Similarly biclusters may overlap on conditions as well.

We present a set of algorithms which use mutual information to extract biclusters from the expression data. We conduct experiments on both synthetic and real datasets. Through extensive experimentation on four datasets of *Arabidopsis thaliana*, *Saccharomyces cerevisiae*, *Human breast cancer (HBC)* and *Diffuse large B cell lymphoma (DL-*

BCL), we show that biclusters extracted using mutual information are biologically more significant than those extracted by other existing biclustering algorithms which use linear similarity measures.

1.1 Motivation to use Mutual Information as a similarity measure

With advances in experimental technology, increasing methodologies are available for unveiling more complex relationships in gene expression data [KBG+07]. Biologists are interested in studying how change (an increase or decrease) in the expression of a gene affects the expression pattern of other genes. If the increase or decrease in the expression value of a gene viz a viz the increase or decrease in the expression value of another gene is linear then similarity measures like Euclidean distance or correlation coefficient will be able to recognize it. However, if the changes in the expression data are related non-linearly say by a quadratic, exponential or a sinusoidal function as shown in Figure 1.3, then these measures would fail. For example, consider a case of two genes x and y whose expression values are related as $y = sin(x)$. Even though y is completely determined by x yet the correlation coefficient between them turns out to be zero. Thus one needs different measures to identify such complex associations. Mutual information (MI) between two random variables is a measure of the amount of information one random variable contains about the other. It is zero when the two variables are totally independent [CT91, Bis06, ZWD+04]. Thus mutual information is a more general measure to capture linear as well as non linear associations or dependencies amongst genes. Besides this, mutual information has several other advantages over linear similarity measures. Since mutual information uses the distribution in the expression level of the genes rather than their actual values, data need not be normalized. It works well for both scaled and shifted data. Also, measures like correlation coefficient are biased towards outliers

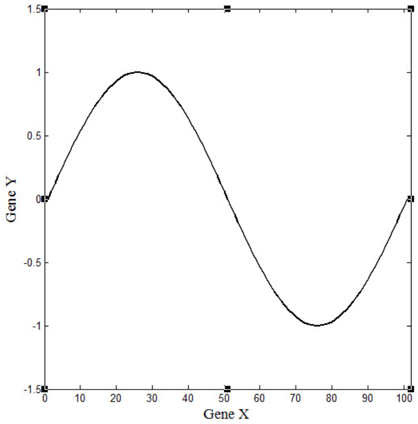


Figure 1.3: Nonlinear relationship between expression of two genes.

whereas mutual information is not. Mutual information is also robust towards noise. We discuss these points in more detail in Chapter 4.

Many researchers [SKD+02, BK00, MCA+98, ZWD+04, SATB05, PMBG07] have used mutual information for one way clustering (clustering of genes on the entire set of conditions). They have shown that the information theoretic measure is responsive to any type of dependencies including strongly non linear structures as compared to traditional measures like Euclidean distance and correlation coefficient which search only for linear relations.

Kraskov et al. [KSAG05] found that even though correlation coefficient between some gene pairs was zero, the mutual information between them was non zero thus indicating that non linear dependencies may exist between the genes. Butte and Kohane [BK00] constructed networks of various genes having high mutual information between them. They found that each network corresponded to some biological activity. According to Michaels et al. [MCA+98] some genes may share some common control inputs (and hence are related to each other) but respond differently to these inputs and only mutual information is able to identify their coordinated behavior.

1.2 Our Contribution

In this work we propose a set of biclustering algorithms using mutual information as a measure of similarity. To the best of our knowledge, none of the existing algorithms for biclustering have used mutual information as a measure of association between two genes or conditions.

In the first algorithm [GA09], to see the impact of using mutual information (MI), we simply plugged MI as a similarity measure in Maximum Similarity Biclustering algorithm (MSB) by Liu et al. [LW07] to obtain what we call as Maximum Related Biclustering algorithm named **MRB**. In this work the problem of biclustering is presented as an optimization problem and a polynomial time solution is provided for it. Score of a bicluster is defined to be the minimum of the gene scores and the condition scores where the gene score is the average mutual information of a gene with the seed gene and the condition score is the average contribution of a condition to the similarity of a set of genes in the bicluster. The aim is to minimize the bicluster score.

The algorithm was tested both on synthetic data and real data sets. The main idea behind construction of the synthetic data was to model nonlinear relationships between the genes of the bicluster over a subset of conditions. We were able to extract the implanted biclusters in the synthetic data. The algorithm was also studied on expression data sets of *Saccharomyces cerevisiae*, *Arabidopsis thaliana*, *Human breast cancer* and *Diffuse large B cell lymphoma* datasets. The biclusters were validated using external biological information by determining the functionality of the genes of the biclusters from Gene Ontology databases using *Database for Annotation, Visualization and Integrated Discovery (DAVID)* [DWHL08], an online bioinformatics tool. These databases store gene annotations i.e. biological knowledge related to genes from its sequence to function. The biclusters obtained from the expression data sets were found to be biologically significant. We also extracted common patterns (motifs) from the promoter regions of the genes belonging to the same bicluster using another online bioinformatics tool named *Retrieve*

Sequence Analysis [RSA] and they were also found to be biologically significant.

Our second approach [GA10] is based on the hypothesis that if a set of genes are related to each other then the conditions under which the genes are associated would also be related to each other. The conditions involving response to stimuli to similar pharmaceutical agents, nutritional source will be related to each other. Expression of genes of normal tissues will be related in contrast to expression of genes of diseased tissues. Also study of genes under similar environmental condtions like varying temperatures, at different time points or replicates of same experiment will all be related to each other.

In the third algorithm [GA08] we drop the assumption that the conditions belonging to a bicluster are related. In this approach we extract general biclusters where the conditions may or may not be interdependent. Instead, here we find the conditions which contribute most to the pairwise mutual information of the genes in the reduced set.

These algorithms were tested both on synthetic data and real data sets. Experimental results were compared with other algorithms namely Iterative Signature Algorithm (ISA) by Ihmels et al. [IFB$^+$02] , Binary Inclusion MAXimal (BIMAX) biclustering algorithm by Prelic et al. [PBZ$^+$06], biclustering algorithm by Cheng and Church [CC00], Order Preserving Submatrix algorithm (OPSM) by Ben Dor et al. [BDCKY02] and Maximum Similarity Biclustering algorithm (MSB) by Liu et al. [LW07]. All these algorithms extract biclusters with linear relationships. Biclusters obtained by our algorithm were biologically more relevant than those obtained by others.

In most of the clustering or biclustering algorithms requiring a gene seed or a set of genes, the gene seeds are either chosen randomly or are taken as input from the user. We have incorporated an intelligent method in our algorithms for generating the seeds from the data itself. The number of biclusters to be extracted is also determined by the algorithm itself.

Rest of the thesis is organized as follows: In Chapter 2 an overview of the essential concepts in biology are provided for better understanding of our work. Biclustering and related work are described in Chapter 3. Concepts of mutual information are explained in Chapter 4. Our algorithms along with the experimental results are presented in Chapters 5, 6 and 7. Finally, Chapter 8 concludes our work and discusses the future work.

Chapter 2

Biological Background

This chapter briefly discusses some basic and relevant biological concepts and terms required for proper understanding of our work [APS06, MDPM08, RJLS10, GSS91].

2.1 Basic concepts of a cell

Cell is the basic structural and functional unit of all living organisms. It may be regarded as a basic unit of biological activity. Each cell consists of jelly like material called protoplasm surrounded by a cell membrane. The protoplasm further has two components: the nucleus that contains the genetic material and the cytoplasm containing various other cell organelles like mitochondria, ribosomes etc. Cells of primitive organisms (such as bacteria) which do not have a nucleus are called prokaryotic cells and those of higher organisms which have a well defined nucleus are called eukaryotic cells.

There are four types of basic molecules present in a cell: sugars, lipids, amino acids and nucleotides. Sugar molecules besides being a source of energy for the cell, also play a structural role by forming the cell wall of the plant cell. Lipids are important constituent of the membrane of the cell and other cell organelles. Aminoacids are the structural units of proteins which are responsible for most of the biological processes in a cell. There are twenty naturally occurring amino acids from which all proteins are composed. The linkage of one aminoacid with another is through a peptide bond. Proteins, ‘also called polypeptide chains’ are long chains of aminoacids. Each protein has a unique sequence of aminoacids which determines its functionality. Proteins perform a variety of structural and dynamic functions in a cell.

2.2 DNA and RNA

The genetic information in a cell is stored in a long sequence of nucleotides [1] called the **Deoxy-ribose Nucleic Acid (DNA)**. DNA plays a key role in the transfer of genetic information from parent to its offspring.

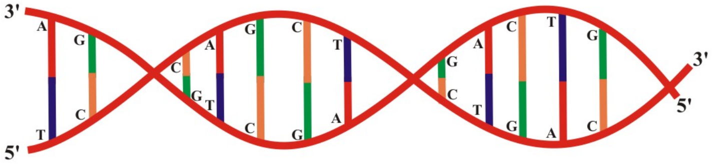


Figure 2.1: Deoxyribose Nucleic Acid

It is made up of two strands wound together in a double helical structure. Each strand consists of sequences of nucleotides with bases adenine (A), thymine (T), cytosine (C) and guanine (G). These bases are repeated throughout the DNA strand millions of times. The bases present in one DNA strand forms hydrogen bonds with the bases in the other strand in a specific manner as shown in the Figure 2.1. ‘A’ pairs with ‘T’ only and ‘G’ pairs with ‘C’ only. Once the sequence of bases in one DNA strand is known, the sequence of the other strand can be constructed because of the specific base pairing. The two strands of DNA are thus said to be complementary and this property of DNA enables it to transmit genetic information. Further, each strand has a polarity, from head called

[1] Nucleotides are molecules consisting of a nitrogenous base, a sugar group and a phosphate ion and are responsible for storage of information about an organism’s inherited characterstics

the 5’ end and a tail called the 3’ end. The two strands are anti parallel (i.e. one strand runs from the 5’ to 3’ direction and the other from 3’ to 5’ direction). DNA of eukaryotes contain noncoding sequences called **introns (or interveaning sequences)** that separate coding sequences called **exons**. Combinations of exons and introns form a **gene** which determine the sequence of aminoacids on a protein as shown in Figure 2.2. The part of DNA that codes for the formation of proteins is called an **open reading frame (ORF)**. When read from the head region (5’ end) to the tail region (3’ end) the portion of DNA before the ORF is called the **upstream region** and the portion of DNA that comes after the ORF is called the **downstream region** of the DNA.

RNA or ribose nucleic acid is a molecule which is chemically similar to DNA. It plays a key role in the synthesis of various proteins in a cell. In some lower organisms it also acts as the carrier of genetic material. Unlike DNA, RNA is a single strand structure, constructed from a DNA strand having base Uracil (U) instead of the base Thymine. There are several types of RNA molecules. The RNA that codes for proteins is called messenger RNA (mRNA). Transfer RNA (tRNA) helps in aligning the amino acids according to the sequence present in the mRNA for the formation of the protein. Ribosomal RNA (rRNA) on the other hand is a major structural component of the protein synthesizing cell organelle called ribosome.

2.3 Gene and Gene expression

Gene refers to a functional unit of DNA. It is a sequence of bases that encodes a protein or an RNA molecule. Protein coding genes carry information for making proteins which determine various characterstics like colour of eye, hair etc. of an organism. The non protein coding genes code for RNA molecules. The physical characterstics of an organism i.e. what that organism looks like is called its **phenotype** and the genetic encoding of its phenotype is called its **genotype**. Set of all the genes in an individual is known as its

genome. The size of a genome may vary from 6000 genes in yeast to about 40,000 genes in human beings. **Gene Expression** is the process by which information from a gene (its sequence) is manifested into structure and functions of a cell. We say that a gene is expressed when the protein it codes for, is synthesized. Gene expression may also be referred to as the step in which the genotype of an organism is manifested into its phenotype i.e. the genetic information stored in the gene is expressed in the form of proteins that are responsible for the phenotype of the organism. Different subset of genes may be responsible for different phenotype of an individual. For example, a subset of genes responsible for the color of the eye may be different from the genes responsible for the height of an individual. Similarly in an apple tree, genes responsible for the shape of the fruit may be different from the ones that control its taste. Consider a gene that controls the height of an individual. The extent of expression of this gene determines the height. If the expression is within a normal range, the individual has a normal acceptable height. An over expressed gene might lead to a *giant* and an under expressed one leads to a *dwarf*.

The genotype of an organism interacts with the environment which thus influences the phenotype. In other words, the characteristics of an organism may be the result of the coordinated expression of one or several genes and their interactions with the environment. A gene may be highly expressed under certain conditions and may be suppressed under some other set of conditions. For example, an apple plant with the genotype of red juicy apple may give good quality apples in favorable conditions like significant amount of sunlight, air and water whereas the same plant under unfavorable conditions will yield bad quality apples.

Each cell of an organism contains the same set of genes. However different sets of genes are expressed in different cells. Also, within the same cell gene expression may vary with time and may also be affected by the internal and external state of the cell at that time. For instance, it is due to the coordinated action of several genes that the

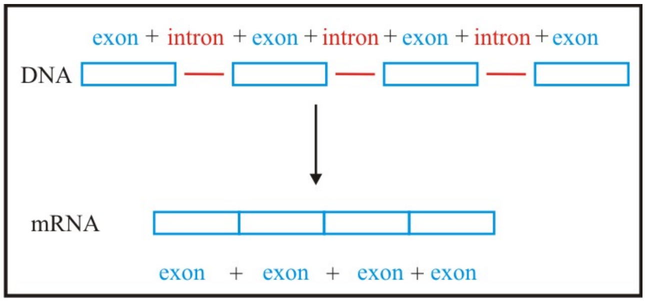

exon + intron + exon + intron + exon + intron + exon
DNA
mRNA
exon + exon + exon + exon

Figure 2.2: Gene Expression

development of the fertilized ovum and sperm grows into a normal adult.

The process of **gene expression** is a two step process that consists of transcription followed by translation. See Figure 2.3. **Transcription** is the process of transfer of genetic information from a portion of DNA into an mRNA molecule. **Translation** is the process of transfer of information from RNA to protein. **Transcription** is the first step in the formation of proteins from genes. It is carried out by an enzyme called RNA Polymerase. The process is initiated when one (or more) special proteins called **Transcription Factor (TF)** bind to one (or more) specific sequence(s) of nucleotides called the **Transcription Factor Binding Site(s) (TFBS)** on the promoter region of a gene. The enzyme RNA Polymerase moves along the strand of the DNA. As it encounters each DNA nucleotide, it adds the corresponding complementary RNA nucleotide to a growing mRNA strand. Once the stop signal is reached the newly constructed mRNA strand is released. Finally, it leaves the nucleus and serves as a template for the synthesis of protein in the cytoplasm at the ribosome.

Translation is the second step in the expression of genes. It involves reading the mRNA, conversion of the message carried in mRNA into amino acids and the synthesis of the corresponding proteins at the ribosomes. The genetic information based on the language of four bases (A, U, C, G) of mRNA is translated to a language of proteins consisting of amino acids. In other words, the sequence of nucleotides in the mRNA determine the sequence of amino acids in the synthesized protein. Each amino acid cor-

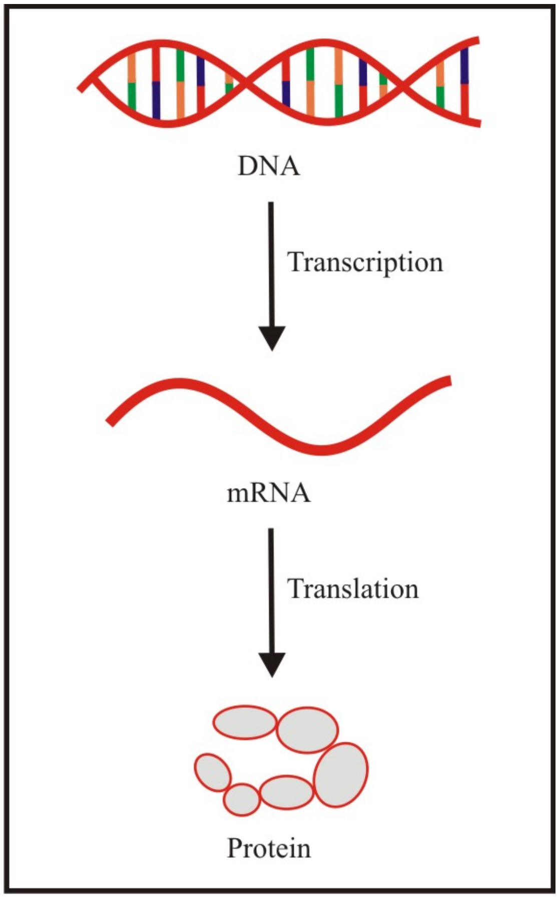


Figure 2.3: Two phases of Gene Expression

responds to a triplet of three nucleotide bases called a **codon**. 4 nucleotides can form 64 (4^3) possible codons out of which three (TAA, TAG and TGA) indicate the end of a protein sequence and are called the stop codons. All others code for a particular aminoacid. Most of the aminoacids are encoded by more than one codon. The codon AUG represents methionine and is also the translational **start signal**.

Translation begins when a tRNA (transfer RNA) molecule encounters the start codon on the mRNA. The tRNA moves up the sequence of the mRNA reading three nucleotides (codon) at a time. Each codon specifies the amino acid to be added to the protein sequence. The translation continues until a stop codon is encountered. The amino acid chain is then released. This amino acid chain is nothing but the protein the gene codes for.

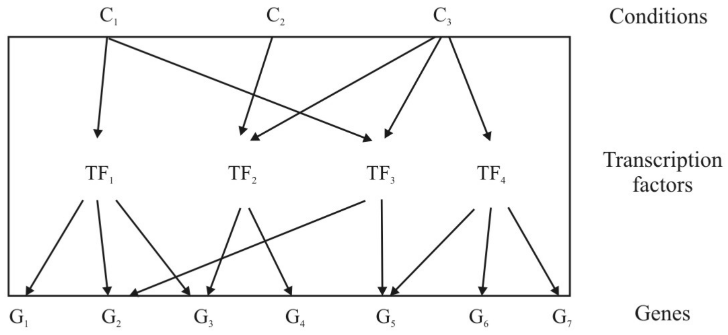


Figure 2.4: Effect of Conditions on Gene Expression

Since the expression of a gene is controlled and regulated by one or more TFs and their binding to the TFBS in the promoter regions of the gene, it is expected that the genes that are coexpressed are regulated by the same set of TFs and hence have common TFBSs. In other words genes having similar expression profiles, thus belonging to the same bicluster, are considered to have a **common regulatory mechanism or signature or motif** in their promoter region [HZGD05].

Binding of Transcription factors with the TFBS may be regulated by many conditions as shown in Figure 2.4. Infact expression of one gene may be governed by the expression of another gene. Some genes may code for a protein which in turn may act as a transcription factor and regulate the expression of some other genes. The entire network is quite complex. Also, the gene to protein correspondence is not one to one. There are genes that may code for more than one protein. Ideally measurement of gene expression should be done by measuring the amount of protein produced. However, it is often easier to measure one of the intermediate product like mRNA to infer the gene's expression level.

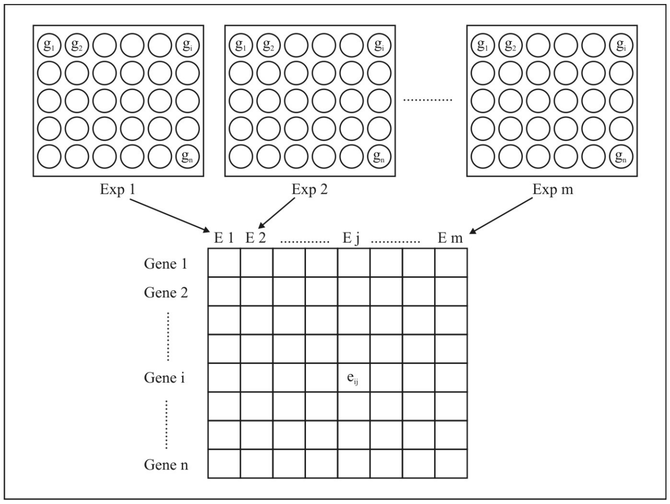


Figure 2.5: Microarray Experiments: each experiment corresponds to a condition

2.4 Microarray experiments and Expression matrix

A microarray experiment allows us to determine the expression levels of thousands of genes by measuring the amount of mRNA bound to each site of the microarray. Data resulting from a microarray experiment is represented as a **gene expression matrix**. Microarray experiments are based on the principle of hybridization. Linking of a DNA strand with an mRNA strand by hydrogen bonds is called **hybridization**. More the number of complementary bases present between the DNA and the mRNA strand, stronger is the hybridization. A microarray chip is a glass or a silicon slide [CQB04] which consists of an array of spots, where each spot contains multiple copies of a gene sequence known as probe. Thus in one experiment with a microarray chip with n spots, we can study the expression levels of n genes (probes) simultaneously. There may be thousands of such spots on a slide each containing millions of identical DNA molecules. From a sample of interest, e.g. a tumor biopsy, mRNA molecules (known as target) extracted and labeled with fluroscent dye are washed over the entire slide and is allowed to hybridize to the complementary gene sequences on the array. The sample contains more concentration of the mRNA molecules corresponding to the genes highly expressed under the tumorous condition as compared to other genes. Thus the degree of hybridization will be more for such genes. By measuring the intensities of the fluroscent light emitted from each spot of hybridization, one can measure the amount of mRNA present in the sample. The images of the array are taken and analysed using image analysis software resulting in an intensity matrix.

Each intensity matrix is converted into a vector which corresponds to a column of the gene expression matrix with n rows or genes. Repeating the experiment for m conditions results in a $n \times m$ gene expression matrix as shown in Figure 2.5. Each row in the matrix corresponds to the expression profile of a gene and each column corresponds to a sample or a condition [CST00]. The $(ij)^{th}$ entry of the expression matrix represents the expression of i^{th} gene under j^{th} sample.

Chapter 3

Biclustering

Clustering algorithms have often been used to reduce the complexity of humongous expression data made available with the help of microarray technology. The traditional clustering algorithms are not suitable for all applications especially gene expression data analysis. These traditional clustering algorithms group the genes over all the conditions whereas cellular processes are affected only under a small subset of conditions. Most of the conditions which do not contribute to the cellular process add to the background noise. In gene expression data, it is desired to search for groups of genes which show some compatibility under a small group of conditions. Also, a single gene may belong to more than one group as a gene may be involved in more than one biological process.

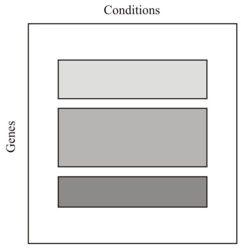


Figure 3.1: Principal Component Analysis

Various dimensionality reduction techniques like **principal component analysis**

(PCA), **singular value decomposition** and **feature selection** have been used to filter out the irrelevant conditions. However, these techniques compute clusters on the same set of few relevant conditions (as shown in Figure 3.1) whereas different sets of conditions trigger different biological processes. Biologists are not only interested in identifying sets of coexpressed genes but also the group of conditions responsible for the co-expression of a group of genes. For example, when cancer patients are treated with different drugs, one is interested in determining the genes causing cancer and also the set of drugs to which a patient responds positively.

Cheng and Church [CC00] introduced the notion of biclustering for gene expression data. **Biclustering** refers to simultaneous clustering of both genes and experimental conditions of the expression data. A bicluster can be viewed as a submatrix of the gene expression matrix such that the rows of the bicluster show a similar behavior under the columns of the bicluster [MO04].

Unlike traditional clustering algorithms where clustering is carried out over all the conditions to produce nonoverlapping clusters, PCA/feature selection algorithms where clustering is performed over the *same* set of reduced samples or coclustering algorithms which again extract nonoverlapping clusters, *biclustering* is a more general framework as the biclusters may overlap both on genes as well as on conditions. That is, they allow genes and conditions to belong to more than one bicluster and be responsible for more than one biological activity. Thus biclustering algorithms fit better to biological behavior in contrast to traditional clustering/feature selection/coclustering.

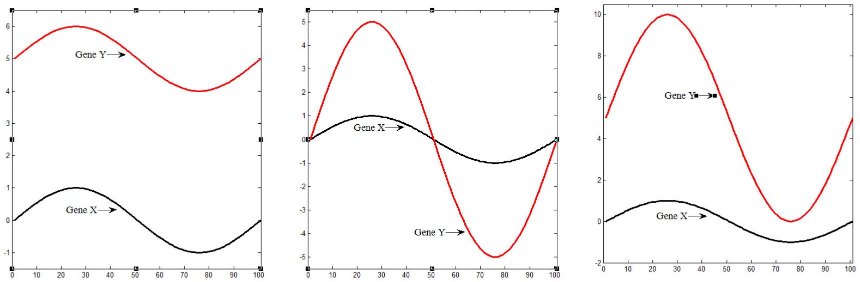
3.1 Problem Definition

Let G be a set of N_g genes and C be a set of N_c samples/conditions. Let E be an $N_g \times N_c$ expression matrix where each row represents the expression of a gene under N_c samples. E is subjected to a biclustering algorithm which delivers a biclustering scheme π_i consisting of k_i biclusters. $\pi_i = (BC_1, BC_2, ..., BC_{k_i})$, BC_j is a tuple (G_j, C_j), G_j being a subset of genes and C_j a subset of conditions such that G_j show similar behaviour under C_j. Different biclustering schemes may contain different number of biclusters. Let $\lambda : (G \times C) \rightarrow 2^{\{0...k\}}$ be a function that yields a set of labels for each gene condition pair (g_l, c_r). Note that since the biclusters may overlap both on genes and conditions, a (gene, condition) pair may be assigned more than one label. Also, there may be a (gene, condition) pair which does not belong to any bicluster, such a pair is assigned a special label 0.

Many biclustering algorithms also define a score for a bicluster and aim to discover biclusters that optimize the score. Biclustering algorithms are interested in identifying $B_{opt} = argmax\{f(B)\}/argmin\{(f(B)\}$ where $f(B)$ denotes the score of a bicluster B,. Bicluster scores have been defined in various ways in literature. Cheng and Church used average Mean Square Residure (MSR) as a bicluster score and aimed to minimize the score. The **residue** of an element a_{ij} in the bicluster denoted by $A(I, J)$ is defined as $r_{ij} = (a_{ij} - a_{Ij} - a_{iJ} - a_{IJ})$ where a_{Ij}, a_{iJ} and a_{IJ} are the row, column and bicluster mean respectively. The mean square residue $H(I, J)$ of a bicluster $A(I, J)$ is then given by $\frac{1}{|I||J|} \sum_{i \in I, j \in J} r_{ij}^2$. In [LW07] authors define a bicluster score $s(I, J)$ as the minimum similarity score of any gene with the seed gene ($min_{i \in I}\ s(i, J)$) or the minimum similarity score of any condition ($min_{j \in J}\ s(I, j)$) whichever is minimum i.e. $s(I, J) = min\{min_{i \in I}\ \{s(i, J)\}, min_{j \in J}\ \{s(I, j)\}\}$ where the similarity is based on Euclidean distance. Liu et al. aimed to maximize the bicluster score. In one of our approaches we use the same definition of the bicluster score but our similarity of a row with the seed row is based on **mutual information** rather than **Euclidean distance**.

3.2 Types of Biclusters

Genes in a bicluster have expression values varying in a similar manner or having some relationship under the conditions of the bicluster. Different biclustering algorithms define this similarity/ relationship differently. We classify biclusters into two broad categories based on the type of relationship that exists between the genes as follows:



(a) Additive relationships

(b) Multiplicative relationships

(c) Additive and multiplicative relationships

Figure 3.2: Linear relationships between expression of two genes

1. **Biclusters with linear relationships:** These biclusters consist of genes having linear relationship between their expression values i.e. the expression levels of genes show linear coherence. Most general form of linear relationship may be described as $y = mx + c$. Different types of biclusters resulting from different relationships like additive, multiplicative or a combination of both additive and multiplcative as shown in Figure 3.2 respectively fall under this category as shown in Tables (3.1, 3.2 and 3.3). Most of the biclustering algorithms extract such biclusters having linear relationships that may overlap both on genes and conditions.

2. **Biclusters having nonlinear relationships:** Whenever the expression of a gene is a nonlinear function of the expression of another gene we say that there exists a

	c1	c2	c3	c4	c5	c6
g1	1	2	3	4	5	6
g2	2	3	4	5	6	7
g3	1.5	2.5	3.5	4.5	5.5	6.5
g4	2.5	3.5	4.5	5.5	6.5	7.5

Table 3.1: Bicluster with additive relationships

	c1	c2	c3	c4	c5	c6
g1	1	2	3	4	5	6
g2	2	4	6	8	10	12
g3	.5	1	1.5	2	2.5	3
g4	3	6	9	12	15	18

Table 3.2: Bicluster with multiplicative relationships

	c1	c2	c3	c4	c5	c6
g1	1	2	3	4	5	6
g2	3	5	7	9	11	13
g3	4	7	10	13	16	19
g4	5	9	13	17	21	25

Table 3.3: Bicluster with general linear relationships

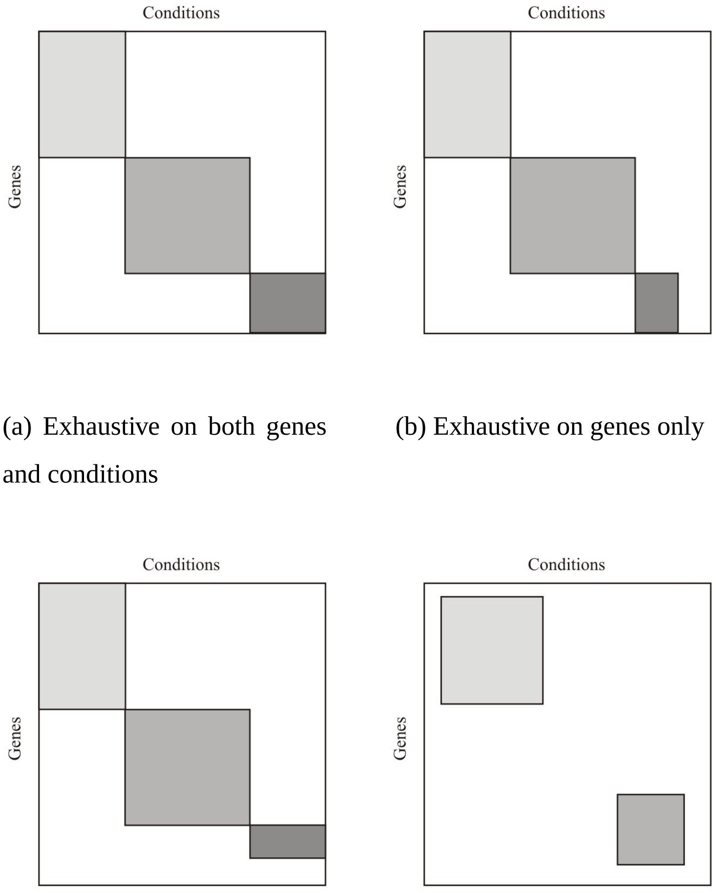
	c1	c2	c3	c4	c5	c6
g1	-3	-2	-1	1	2	3
g2	9	4	1	1	4	9
g3	27	8	1	1	8	27
g4	4.5	2	.5	.5	2	4.5

Table 3.4: Bicluster with nonlinear relationships

nonlinear relationship between the two genes as shown in Table 3.4. The expression level of gene $g2$ is obtained as square of the expression level of gene $g1$ and that of gene $g3$ is obtained as absolute of cube of the expression level of gene $g1$ and that of $g4$ is obtained as half of square of the expression level of gene $g1$. Algorithms using similarity measures like distance or correlation coefficient are unable to extract such biclusters.

3.3 Organization of Biclusters in the expression data

Different biclustering algorithms extract different number of biclusters of varying sizes. A scheme of biclusters can be classified into **exhaustive** or **nonexhaustive** depending on whether all genes or conditions belong to some bicluster or not as shown in Figure 3.3. A scheme of biclusters can also be classified into **overlapping** or **nonoverlapping** depending upon whether genes or conditions can be a part of one or more than one bicluster at the same time as shown in Figure 3.4.



(a) Exhaustive on both genes and conditions

(b) Exhaustive on genes only

(c) Exhaustive on conditions only

(d) Nonexhaustive biclusters

Figure 3.3: Biclustering schemes with different coverage

1. **Exhaustive on both genes and conditions:** A biclustering scheme is said to be **exhaustive on both genes and conditions** if all the genes or conditions present in the expression matrix belong to atleast one bicluster as shown in Figure 3.3(a).

2. **Exhaustive on genes only:** A biclustering scheme is said to be **exhaustive on genes** if all the genes present in the expression matrix belong to atleast one bicluster but there may be certain conditions which are left unclustered as shown in Figure 3.3(b).

3. **Exhaustive on conditions only:** A biclustering scheme is said to be **exhaustive on conditions** if all the conditions present in the expression matrix belong to atleast one bicluster but there may be certain genes which are left unclustered as shown in Figure 3.3(c).

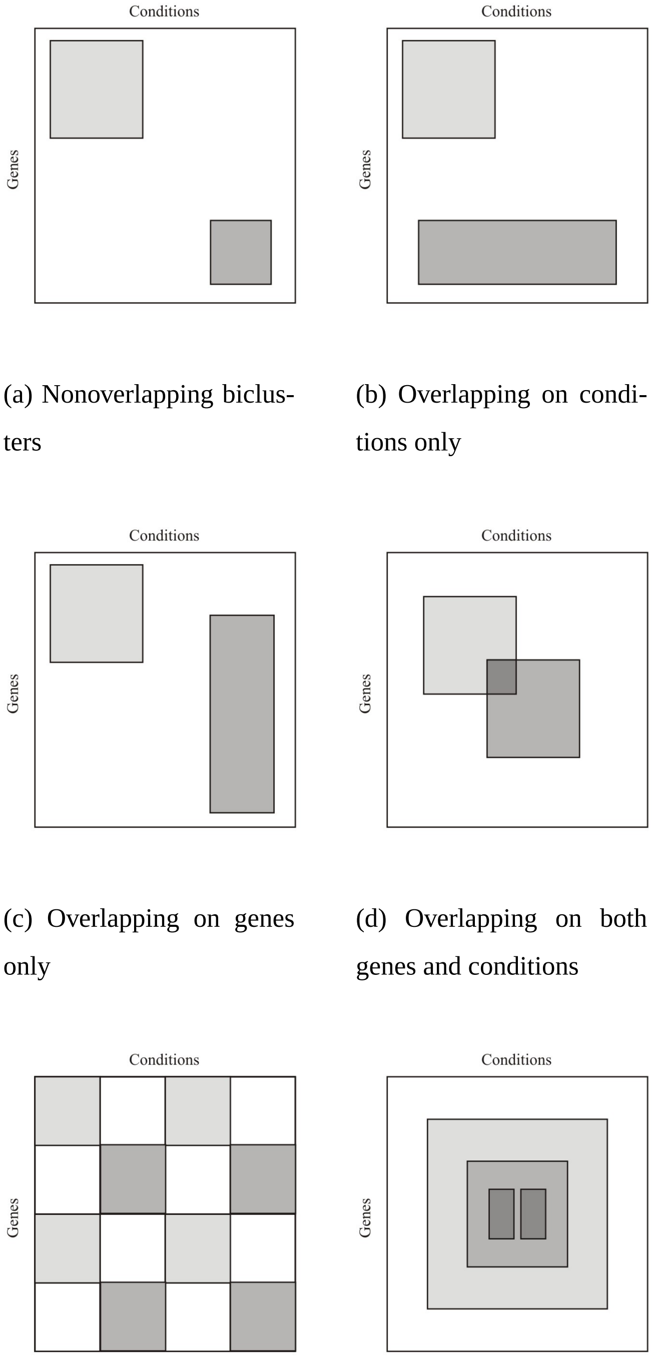
4. **Non-Exhaustive:** A biclustering scheme is said to be **non-exhaustive** if few genes or conditions present in the expression matrix are left unclustered as shown in Figure 3.3(d).

5. **Non-overlapping:** Two biclusters are said to be non-overlapping if they neither share a gene nor a condition as shown in Figure 3.4(a).

6. **Overlapping on conditions only:** Two biclusters are said to be **overlapping on conditions** if they share a condition but do not share a gene as shown in Figure 3.4(b). Traditional clusters always overlap on conditions as clustering is performed on the entire set of conditions.

7. **Overlapping on genes only:** When one or more genes may be shared between two biclusters but there are no common conditions then the biclustering scheme is said to be **overlapping on genes** as shown in Figure 3.4(c).

8. **Overlapping on both genes and conditions:** Two biclusters are said to be overlapping on both genes and conditions if they share one or more genes as well as conditions as shown in Figure 3.4(d). Checker board structure allows a gene and condition to belong to more than one bicluster as shown in Figure 3.4(e). However, a gene condition pair can together belong to atmost one bicluster.



(a) Nonoverlapping biclusters

(b) Overlapping on conditions only

(c) Overlapping on genes only

(d) Overlapping on both genes and conditions

(e) Checker board organization

(f) Hierarchical organization

Figure 3.4: Biclustering schemes with different overlap

9. **Overlapping with hierarchical organization:** When two or more bicluster combine to form another bicluster we get a hierarchical scheme of biclusters as shown in Figure 3.4(f).

3.4 Related Work

Like traditional clustering, most of the algorithms for biclustering are also heuristic in nature. Various approaches and similarity measures have been used in literature to extract biclusters. Based on the approach, an algorithm may fall in one or more of the following categories:

1. Iterative Algorithms
2. Enumerative Algorithms
3. Divide and Conquer Algorithms
4. Two way Clustering Algorithms
5. Probabilistic Algorithms
6. Graph based Algorithms
7. Other approaches

3.4.1 Iterative Algorithms

Starting from an initial solution, these algorithms iteratively improve the quality of the biclusters. **Cheng and Church** [CC00] presented the first such algorithm for the biclustering problem. Authors define the **residue** (r_{ij}) of an element a_{ij} in a bicluster denoted by $A(I, J)$ as $r_{ij} = (a_{ij} - a_{Ij} - a_{iJ} - a_{IJ})$ where a_{Ij}, a_{iJ} and a_{IJ} are the row, column and bicluster mean respectively. The mean square residue $H(I, J)$ is given by

$\frac{1}{|I||J|}\sum_{i\in I,j\in J} r_{ij}^2$. They defined δ bicluster as the one whose mean square residue score (MSR) is less than a threshold δ. They proposed a node deletion algorithm to find large sized δ biclusters. Starting with the input matrix as a whole the algorithm selects a row or a column with the highest score for deletion such that the MSR of the resultant submatrix is lowered. This is repeated until the MSR is lowered below δ. The bicluster obtained after deletion of rows and columns may not be of maximum size. The algorithm then adds the previously deleted row (column) with the lowest score such that the MSR of the bicluster is increased but remains below δ. This is repeated as long as MSR is less than δ. Missing values in the data are replaced with random numbers. This is in the hope that these random values would not form recognizable pattern and thus would get removed in the node deletion phase.

In order to find more biclusters, the elements of the submatrix representing the discovered bicluster are masked by random numbers. The masking of the discovered bicluster eliminates the related behaviour in it so that other biclusters could be discovered. However, such a masking could interfere with the identification of overlapping biclusters [YWWY03]. The algorithm does not work well when a large amount of noise is present. When the noise levels are high the MSR is also high and some important biclusters may be missed.

FLexible Overlapped biClusters (FLOC) by Yang et al. [YWWY03] extended Cheng and Church's algorithm to find k biclusters simultaneously without random replacement. Starting from a random set of biclusters they iteratively try to move a gene or a condition from one bicluster to another such that the average mean square residue of the entire scheme of biclusters is reduced. FLOC allows biclusters with limited number of missing values. The quality of FLOC's biclusters depends on the initial seed clusters which are generated randomly.

Zhang et al. [ZTOT04] in Deterministic Biclustering by Frequent pattern mining (DBF) extended FLOC by generating the initial seed cluster in a deterministic manner rather than randomly. They generated a set of good quality (with low mean square residue) biclusters using CHARM [ZH02], a frequent pattern mining algorithm.

Iterative Signature Algorithm (ISA) by Ihmels et al. [IFB$^+$02, BIB03] is another iterative algorithm that starts with a random set of genes and computes the set of conditions under which the input genes are most tightly regulated. Using these conditions it iteratively refines the set of genes and then the set of conditions until the set of genes and samples converges i.e. they do not change anymore. ISA works on the hypothesis that although the set of possible input seeds is huge, usually there is only a limited number of fixed points for a given set of thresholds. They run the algorithm for a large number of input seeds and reconstruct the modules from the recurring fixed points by fusing the solutions that were distinct but very similar, using a procedure that resembles agglomerative clustering.

ISA extracts biclusters consisting of genes, which exhibit similar expression pattern with high expression values. The biclusters may be overlapping on both genes and conditions. In presence of high values of expression, ISA misses out biclusters where genes show similar expression pattern but have low values. **Progressive Iterative Signature Algorithm by Kloster et al. [KTW05]** extended ISA to find orthogonal modules. These modules are hard to interpret as they are in different condition space.

Ben Dor et al. [BDCKY02] defined a bicluster as an **Order Preserving Sub Matrix (OPSM)** in which the expression levels of the genes move up and down together on the conditions of the bicluster. For a set T of s conditions, they define a linear order (permutation) π. A gene belongs to a bicluster defined by T if the s corresponding entries of the gene expression, ordered according to the permutation π, are strictly increasing. They proposed a heuristic algorithm in which they discover l order preserving submatrix for every possible value for s, and select the most significant of these solutions. For a given value of s the algorithm iteratively computes condition permutations starting with a permutation of size 2 (partial linear order), choosing the best l permutations in each iteration and increasing the number of conditions by one until the permutations are of length s. Finally, one best of the l solutions with sample size s are selected. The biclusters retrieved by the algorithm have linear relationships and may overlap both on genes and conditions.

Liu et al. [LW07] projected the biclustering problem as an optimization problem and presented a polynomial time solution for it. With one gene selected as a reference gene, they defined similarity score of a gene as the average Euclidean distance of the gene from the reference gene over a set of conditions. The similarity score of a condition is defined as the average contribution of the sample to the similarity of a set of genes with the reference gene. Finally they defined the similarity score of a bicluster to be the minimum similarity score of rows and columns of the bicluster. The aim is to extact a bicluster with maximum score. Their algorithm is essentially a greedy algorithm which iteratively removes a row or a column that contributes minimum to the score of the bicluster i.e a row or a column whose similarity score is the smallest (worst) among all rows and columns in the current bicluster. Several submatrices are generated in the process; they select the one with the maximum similarity score. More biclusters are generated by changing the reference genes selected randomly by the authors. Instead of generating the gene seeds randomly, a method to select well separated reference genes from the expression data has been used in this work (Chapter 5).

Bozdag et al. [BPC09] proposed an algorithm to extract biclusters named as **Correlated Pattern Biclusters (CPB)**. Pearson's correlation coefficient (PCC) calculated between any two genes of the CPB over its conditions is required to be greater than a threshold. The algorithm starts by randomly selecting a set of genes and conditions to form a bicluster. It iteratively improves the bicluster by moving genes and conditions in and out of the bicluster. They compare the PCC between each gene and a reference gene to decide which gene to move. To decide on the inclusion of a column c into the bicluster they compute the impact of c on the PCC between the genes of the bicluster. A column c

is included only if it does not decrease correlation among the rows in the bicluster.

3.4.2 Enumerative Algorithms

Enumerative algorithms extract biclusters from the expression data by listing or **enumerating** all possible biclusters and then selecting the best amongst them. Wang et al. [WWYY02] used prefix tree to enumerate all the biclusters. They extract δ pClusters from the gene expression data (p stands for pattern). A δ pCluster is defined as a bicluster in which the change of values on every pair of conditions between every pair of genes is less than a user defined threshold δ. They try to capture those sets of genes for which the change of expression values on conditions show similar patterns. The algorithm in its first step examines the data to form a set of candidate maximum dimension sets (MDS) for all pairs of genes and for all pairs of conditions. MDSs are pruned using the relationship between the gene pair MDSs and the condition pair MDSs. A prefix tree is then built using the remaining MDSs. Finally the postorder traversal of the tree gives the output biclusters.

Ayadi et al. [AEH09] proposed another enumerative algorithm called **BiMine** which used a **Binary Enumeration Tree (BET)** to enumerate all biclusters and an evaluation function to throw away the bad quality biclusters. The algorithm proceeds in three steps. The first step involves preprocessing the data during which irrelevant expression values of the data matrix that do not contribute in obtaining relevant results are removed. A gene is considered insignificant if the difference of its expression under every condition and its average under all the conditions is very small. In the next step a BET is constructed from biclusters of single genes and few relevant conditions. The single gene biclusters are then combined to form biclusters of two genes. Two gene biclusters are then combined to construct biclusters of three genes. In this way larger biclusters are built from the smaller ones. At each step the quality of the obtained biclusters is evaluated using an evaluation function based on Spearman's correlation coefficient. Low quality biclusters are discarded as there is no point in expanding a bad quality bicluster.

Ahn et al. [AYP11] proposed an algorithm to identify biclusters with functionally highly correlated gene sets called **Robust to Noise cluster (RNC)**. A RNC does not contain genes for which the expression values are constant over a sample pair. A p-RNC is a bicluster where the number of samples are p. Initially, the algorithm obtains all the initial 2-RNCs. For all the 2-RNCs, they make a 3-RNC by examining the current sample s_i such that $last < i$ where s_{last} is the last sample in the sample set of 2-RNC. They obtain 3-RNC from 2-RNCs, 4-RNC from 3-RNCs and so on. Those p-RNCs having a larger number of genes and those which have a higher probability of growing to a bigger p-RNC are selected with the help of priority queues. Finally duplicate RNCs are eliminated and the remaining are selected for output.

3.4.3 Divide and Conquer Algorithms

These algorithms typically divide the data matrices into a number of submatrices, work recursively on each of these submatrices using some heuristics and select the best biclusters amongst them.

Prelic et al. proposed a fast divide and conquer approach, namely the **Binary Inclusion MAXimal biclustering (BIMAX) algorithm [PBZ$^+$06]** that finds all inclusion maximal biclusters (that are not entirely contained in any other bicluster). They preprocess the data matrix to convert it into a binary matrix by fixing a threshold. Expression level of gene above the threshold are set to 1 and those below it are set to 0. A bicluster is then defined as a submatrix in which all the elements equal 1. They partition the expression matrix into three submatrices one of which contains only 0 cells and can be neglected. The other two submatrices contain both 0 and 1 cells. The algorithm is recursively applied to these two submatrices. The recursion ends when the reduced matrix represents a bicluster (i.e. contains only 1s). A basic problem with the process of discretization is when the noise levels in the data are high the difference between the bicluster and the background values becomes very small. As a result many small biclusters may be extracted.

3.4.4 Two Way Clustering Algorithms

These algorithms view the data matrix in two different ways. The first view considers genes as objects and the conditions as dimensions. The second view considers the conditions as objects and the genes as dimensions. Traditional one way clustering is applied to cluster the genes and conditions separately. The one dimensional clusters are then combined and improved to obtain the final biclusters.

Coupled Two Way Clustering (CTWC) by Getz et al. [GLD00] used a one dimensional clustering algorithm called Super Para-magnetic Clustering (SPC) [GLDZ00], to obtain stable (statistically significant) clusters G_i of genes and C_i of conditions. Submatrices are formed by picking one gene cluster from the set $\{G_i\}$ of gene clusters and one condition cluster from the set $\{C_j\}$ of condition clusters. It then recursively computes new biclusters from these submatrices. The process terminates when no new stable biclusters are formed. The type of biclusters obtained depends on the choice of the one way clustering algorithm. SPC uses Euclidean distance as a similarity measure. The algorithm falls in the category of divide and conquer algorithm also.

Iterative Two Way Clustering (ITWC) by Tang et al. [TZZR01] used correlation coefficient to perform two way clustering. They identify a reduced set of genes which distinguish the samples from one another. Though Madiera and Oliviera have referred to it as a biclustering algorithm in their survey we feel that the work is closer to the problem of feature selection. Like BiMine, the data is preprocessed to eliminate the genes which do not contribute towards distinguishing the conditions i.e. the genes which do not show much variation on the set of conditions are removed. After preprocessing, the gene clusters are obtained using correlation coefficient. These gene clusters are further used to obtain the condition clusters. The condition clusters are then combined pairwise to form heterogeneous groups. A reduced set of genes is obtained for each heterogeneous group. Finally using cross validation techniques one reduced gene set is selected for the next iteration. The entire process is repeated iteratively until the number of genes are reduced to some threshold value or the condition clusters reach a certain level of similarity. Finally a set of genes are selected which are used to cluster the conditions. The algorithm also falls in the category of iterative algorithms.

Double Conjugate Clustering (DCC) by Busygin et al. [BJKA02] use self organizing maps (SOM) to perform clustering in both the gene space and the condition space. It alternates between clustering genes and conditions. Every node in gene/condition space is assigned a node called the conjugate node in the condition/gene space. In every iteration the nodes of current clustering space are mapped to their conjugate nodes which are moved accordingly. Following this, clustering is done in the other space using SOM. For every gene cluster the corresponding sample cluster contains those samples which can be used to distinguish the genes of the cluster from the rest of the genes. Similarly a gene cluster corresponding to a sample cluster contains those genes which distinguish the samples of the cluster from the rest of the samples.The entire process is repeated iteratively till the number of movements are reduced to a threshold. Finally, a set of gene clusters and its conjugate set of conditions are selected as output. The algorithm also falls under the category of iterative algorithm.

Chandra et al. [CSM06] proposed a two way clustering algorithm which use the concept of entropy and correlation coefficient to cluster the genes and fuzzy C-means algorithm to cluster the samples like BiMine and ITWC. Preprocessing of data is done to eliminate the genes that do not contribute to the sample clustering i.e. the genes whose values do not vary much across the samples are eliminated. Next the samples are clustered using Fuzzy C-means algorithm. The number of clusters in the data set are determined by the algorithm itself by measuring the amount of overlap between the clusters. Gene clusters are formed using entropy and the remaining genes are selected for the next iteration resulting in mutually exclusive biclusters. This is repeated till the number of genes are reduced to a minimum limit.

3.4.5 Probabilistic Algorithms

Murali and Kasif [MK03] defined a bicluster as a set of samples and a set of genes that are conserved under the set of samples. A set of genes is said to be conserved under a set of conditions if it is present in same abundance (state) in all the conditions. A gene state is represented by a range of expression values. They use the term **xMotif** to refer to a bicluster.

For each condition seed c, several sets of samples are selected at random. These sets serve as candidates for the discriminating set D_c. A discriminating set D_c distinguishes between the genes of the bicluster (those having the same state in all the conditions of D_c) and the rest of the genes (those having different values on the conditions of D_c. Given a condition seed c and a discriminating set D_c, an xMotif contains exactly those genes that have the same state on c and all the conditions in D_c. These xMotifs are extracted for all the condition seeds. Finally the one containing the largest number of genes is selected as output. To obtain more biclusters the conditions belonging to the extracted xMotif are removed and the entire process is repeated until all the samples are assigned to some xMotif. It is clear that the biclusters are mutually exclusive and exhaustive on the set of conditions whereas they may overlap and may be non-exhaustive on genes.

3.4.6 Graph based Algorithms

Statistical Algorithmic Method for Bicustering Analysis (SAMBA) by Tanay et al. [TSS02] use graph based techniques along with probabilistic modeling of the data to identify biclusters. They represent the expression data as a bipartite graph whose nodes correspond to genes and conditions. An edge between a gene and a sample represents significant change in the expression value (up and down regulation) of the gene under that experimental condition with respect to its normal level. Edges and non-edges are assigned weights according to a probabilistic model so that the problem of extracting biclusters is reduced to finding the heavy subgraphs.

Li et al. [LMT$^+$09] proposed a **QUalitative BIClustering algorithm (QUBIC)** which converts the expression matrix into a representing matrix in which the expression level of a gene under each condition is represented as an integer value. Two genes are considered to be correlated under a subset of conditions if the corresponding integers along the two rows of the representing matrix are identical. They find all optimal submatrices from the representing matrix. For the given matrix, a weighted graph with genes represented as vertices, edges connecting every pair of genes, and the weight of each edge being the similarity level between the two corresponding genes is constructed. The algorithm identifies all biclusters in the matrix by starting with the unused edge as a seed to build an initial bicluster and then iteratively adds more genes till a threshold level is achieved.

3.4.7 Other Algorithms and approaches

Pattern Based Biclustering Algorithms search for specific patterns formed by genes of the bicluster over its conditions. Kluger et al. in [KBCG03] assume that the expression matrix has a checker board like structure. Their method is based on singular value decomposition of the expression matrix. Ben Dor et al. [BDCKY02] also extract biclusters, genes of which show patterns in their expression values. **Factor Analysis for Bicluster Acquisition (FABIA) by Hochreiter et al. [HBH$^+$10]** is based on a multiplicative model. Two vectors are similar if one is a multiple of the other and the angle between them is zero. The algorithm selects the model parameters using an expectation maximization algorithm.

Gan et al. [GLY08] proposed a geometric interpretation of the biclustering problem. They show that different types of biclusters are different spatial arrangements of hyperplanes in a high dimensional data space. Thus the biclustering process is reduced to detection of such hyperplanes or linear geometries. They used Hough transform based hyperplane detection algorithm to discover all the hyperplanes that exist in the gene expression data. Each hyperplane is searched for a pattern in the genes lying in it. If a pattern exists it will be selected for output. Tchagang and Tewfix [TT05] proposed **Robust Biclustering Algorithm (ROBA)** which uses basic linear algebra and arithmetic tools to extract the bicusters. Mitra et al. and others [DMBM07, MB06, MDBM09, NTAR11] use evolutionary approaches to bicluster gene expression data. Conjugate Column Clustering (CCC) by Madiera et al. [MO05] finds biclusters in continous columns from time series gene expression data.

Other work closely related to biclustering pertains to coclustering [DMM03, ST00] and projected clustering [APW$^+$99, AY00, YCN04]. Though researchers sometimes claim that coclustering, projective clustering and biclustering are all same. But the solutions provided for coclustering and projective clustering do not allow a gene/condition to appear more than once in the biclusters. In [DMM03] Dhillon et al. and in [ST00] Slonim and Tishby have used mutual information for coclustering (simultaneous clustering of rows and columns) word document data. They present the coclustering problem as an optimization problem in which they maximize the mutual information between the clustered random variables subject to restrictions on the number of rows and column clusters. An element e_{ij} of the input matrix represents the frequency of occurrence of i^{th} word in the j^{th} document. Dhillon et al. treat the word document matrix as a co-occurrence matrix and use it to represent the joint probability distribution of the words (represented by random variable X) and the documents (represented by random variable Y). They approximate the original matrix with a new matrix consisting of a reduced set of rows $\hat{X}$ and a reduced set of columns $\hat{Y}$, so that the new matrix contains as much information about the earlier one as possible. Thus their approach typically leads to dimensionality reduction. However, it is different from traditional dimensionality reduction in the sense that they do it simultaneously on rows as well as on columns. Though the paper beautifully exploits the information contained in the columns viz a viz rows and the vice versa, it has its limitations especially with reference to gene expression data. Firstly the entries in the gene expression data cannot be treated as a measure of co-occurrence. Secondly, to treat the input matrix as a joint probability distribution the entries must be all positive which may not be the case in gene expression data as down-regulation may be represented by negative values. Banerjee et al. in [BDG$^+$07] propose a generalized coclustering algorithm which works for negative entries in the input matrix as well. They assume that the probability distribution of the input data is either predefined or follows uniform distribution. Both Banerjee and Dhillon identify non-overlapping biclusters whereas a gene may be responsible for more than one cellular function and thus may belong to more than one bicluster. Similarly biclusters may overlap on conditions as well.

3.5 Validation of a Bicluster

A wide variety of clustering and biclustering algorithms exists in literature, yet it is difficult to assess the quality of their solutions. Different algorithms give different solutions on the same data. Most of the time the output depends upon the input parameters as well. Different measures or validity indices are used to evaluate the quality and reliability of the traditional clusters. These measures can be divided into three categories namely **internal, external and relative** [TK99, HBV01]. Internal measures like **intra cluster homogeneity** or **inter cluster separation** rely only on the input data to evaluate the quality of the clusters. External measures use **additional information** to validate the output. Each cluster can be scored based on prior biological knowledge. For example functional enrichment of the genes in a bicluster can be used to validate the biclusters. Also, quality of a bicluster can be measured by searching for common motifs in the promoter region [THC$^+$99] of genes belonging to a bicluster. Relative measures compare the different clustering schemes produced by the same algorithm with different input parameter values. They measure the effect of varying the input parameters on the output of an algorithm.

Different quality measures are applicable in different scenarios depending on the data and on the availability of the ground truth. [GVSS03]. **Rand index** and **Jacard index** are two measures that are popularly used to assess a clustering solution against the ground truth. Jacard index is defined as the ratio of the correctly identified objects to the sum of the correctly identified and incorrectly identified objects. Clearly if all the objects are correctly identified the Jacard index will have the highest value 1 and the least value could be 0. The rand index on the other hand is the ratio of the number of agreements to the number of disagreements. Unlike clusters, different biclusters have different sets of conditions and they may overlap not only on genes but also on conditions. Thus, it is not clear how to extend these measures to biclustering. Also, these measures do not give any indication about the reliability of the biclusters. To the best of our knowledge no general internal index like rand index or jacard index has been developed for biclustering solutions. Many biclustering algorithms [CC00, YWWY03] have used **mean square residue** (explained earlier) as a measure of quality of biclusters. MSR may be a good measure for distance based approaches and would require normalization of data for it to be meaningful.

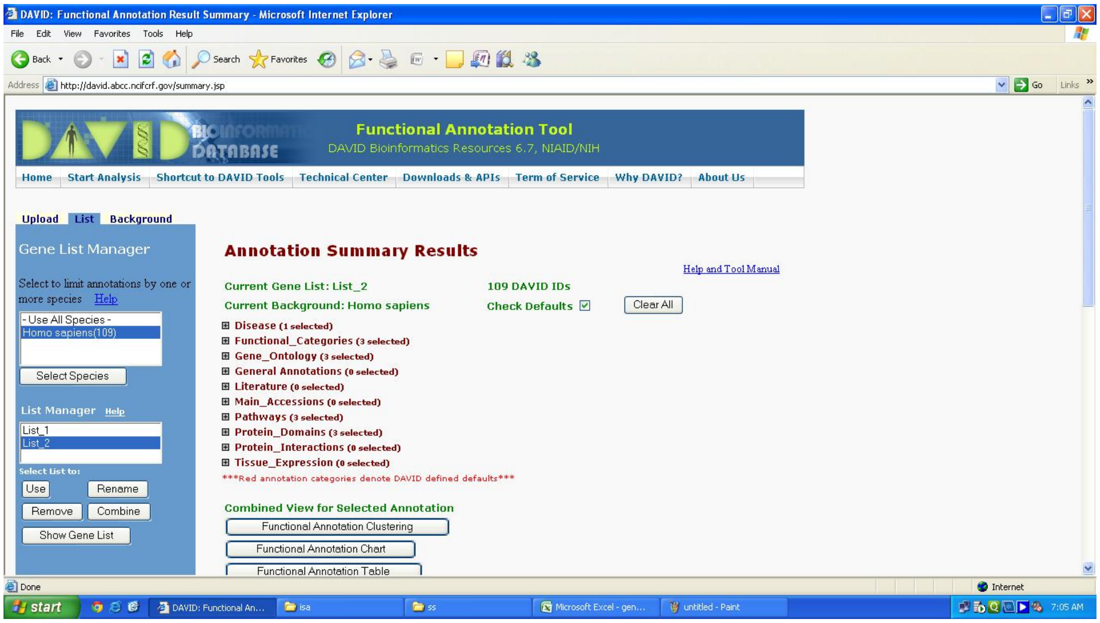
3.5.1 Biological Validation of Biclusters

Most of the biclustering algorithms use external validation methods like GO annotation term [LW07], metabolic pathways [BIB03], protein protein interaction network [PBZ$^+$06] and patterns in promoter regions [THC$^+$99] to assess the quality of biclusters. These methods are based on the hypothesis that a group of related genes are responsible for some biological activity in a cell. We validated our biclusters using functional annotation (GO terms) and common patterns (motifs) in the promoter regions of the genes of a bicluster with the help of biological tools like DAVID and RSAT as explained ahead.

3.5.2 Functional Annotation using DAVID Toolbox

DAVID (Database for Annotation, Visualization and Integrated Discovery), a free online bioinformatics resource, consisting of knowledge database and analytical tools, that help in extracting biological relevance of a set of genes [DWHL08]. The knowledge database integrates major public bioinformatics resources. DAVID's knowledge base collects and integrates diverse gene annotation categories, assigns a centralized internal DAVID identifier to each of them in a nonredundant manner. The wide range of biological annotation coverage in the DAVID knowledge base enables a user's gene ID to be mapped across the entire database thus providing a broad coverage of gene associated annotation. Also, if a significant portion ($> 20\%$) of input gene IDs fail to be mapped to an internal DAVID ID, another DAVID tool, the Gene ID conversion tool starts up to help in the mapping of such IDs.

The Functional Annotation tool of DAVID is used for the enrichment analysis of the gene terms annotated for the input gene set. The basic principle behind the enrichment analysis is that if a biological process is active/abnormal then the co-functioning genes have a higher chance of being selected as a relevant group. To decide about the degree of enrichment, a certain background has to be setup for comparison. As per Huang et al. [DWHL08] larger backgrounds e.g. the total genes in the genome as a background tends to give more significant p values as compared to narrowed down set of genes as background. DAVID has an automatic procedure to determine the background as the global set of genes in the genome on the basis of the user's uploaded gene list. Thus normally a user does not have to setup a population background by itself. Uploading the gene lists of the bicluster is the first step of analysis. DAVID maps a number of genes in the uploaded list to the associated biological annotation i.e. **gene ontology terms** using


DAVID: Functional Annotation Result Summary - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://david.abcc.ncifcrf.gov/summary.jsp
Go Links
DAVID BIOINFORMATICS DATABASE
Functional Annotation Tool
DAVID Bioinformatics Resources 6.7, NIAID/NIH
Home Start Analysis Shortcut to DAVID Tools Technical Center Downloads & APIs Term of Service Why DAVID? About Us
Upload List Background
Gene List Manager
Help
Select Species
List Manager Help
Select List to:
Use Rename Remove Combine Show Gene List
Help and Tool Manual
Clear All
Functional Annotation Clustering
Functional Annotation Chart
Functional Annotation Table
Done
Internet
start DAVID: Functional An... isa ss Microsoft Excel - gen... untitled - Paint 7:05 AM

Select to limit annotations by one or more species

- Use All Species -
- Homo sapiens(109)

- List_1
- List_2

Annotation Summary Results

Current Gene List: List_2 — 109 DAVID IDs

Current Background: Homo sapiens — Check Defaults ☑

- ⊞ Disease (1 selected)
- ⊞ Functional_Categories (3 selected)
- ⊞ Gene_Ontology (3 selected)
- ⊞ General Annotations (0 selected)
- ⊞ Literature (0 selected)
- ⊞ Main_Accessions (0 selected)
- ⊞ Pathways (3 selected)
- ⊞ Protein_Domains (3 selected)
- ⊞ Protein_Interactions (0 selected)
- ⊞ Tissue_Expression (0 selected)

Red annotation categories denote DAVID defined defaults

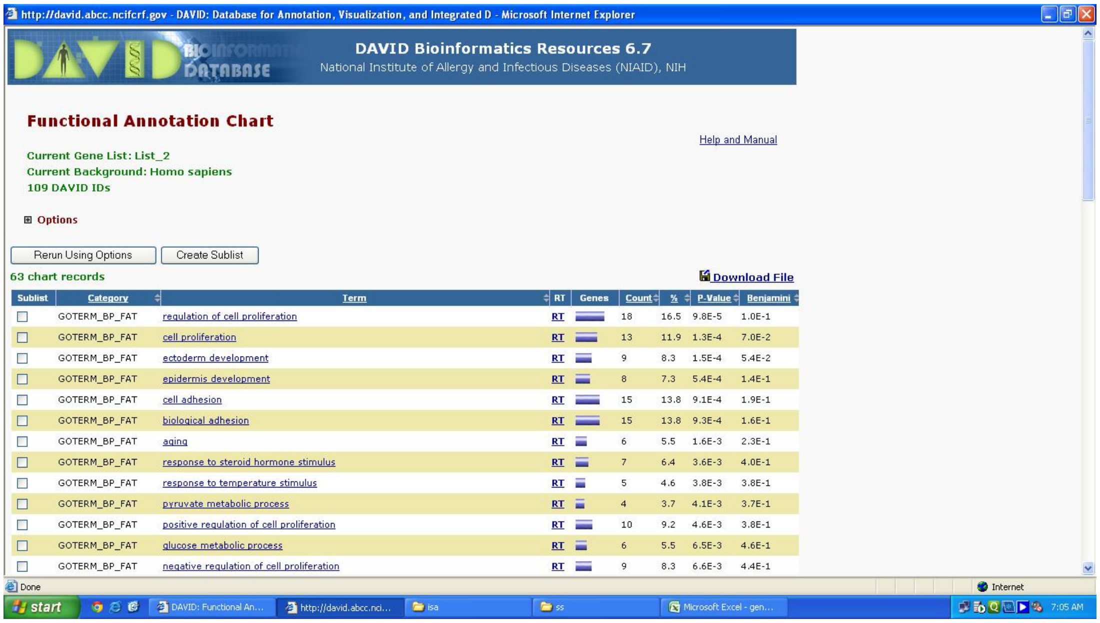
Combined View for Selected Annotation

Figure 3.5: Snapshot of Functional Annotation tool of DAVID

its functional annotation tool as shown in Figure 3.5. It then statistically examines the enrichment of gene members for each of the annotation terms by comparing the outcome to the reference background. This is done by calculating the p **values** (defined later) also called as **EASE score**. Lower is the p value, more statistically significant is the bicluster. Annotation terms below a certain threshold are reported as shown in Figure 3.6.

Gene Ontology terms

There are three Gene Ontologies (GO) that form a common language for annotation of genes of different organisms from yeast to human. They relate genes with different biological processes across different species. The three GO ontologies are (i) **Biological process** which include biological functions to which a gene or a gene's products contribute; (ii) **Cellular component** which includes complex sub-cellular structures, locations and macro-molecular complexes like RNA polymerases where the gene products are active; (iii) **Molecular function** which defines the biochemical activities like carbo-



http://david.abcc.ncifcrf.gov - DAVID: Database for Annotation, Visualization, and Integrated D - Microsoft Internet Explorer
BIOINFORMATICS DATABASE
DAVID Bioinformatics Resources 6.7
National Institute of Allergy and Infectious Diseases (NIAID), NIH
Help and Manual
⊞ Options
Rerun Using Options
Create Sublist
Download File
Done
Internet
start
DAVID: Functional An...
http://david.abcc.nci...
isa
ss
Microsoft Excel - gen...
7:05 AM

Functional Annotation Chart

Current Gene List: List_2
Current Background: Homo sapiens
109 DAVID IDs

63 chart records

Sublist	Category	Term	RT	Genes	Count	%	P-Value	Benjamini
□	GOTERM_BP_FAT	regulation of cell proliferation	RT	▬	18	16.5	9.8E-5	1.0E-1
□	GOTERM_BP_FAT	cell proliferation	RT	▬	13	11.9	1.3E-4	7.0E-2
□	GOTERM_BP_FAT	ectoderm development	RT	▬	9	8.3	1.5E-4	5.4E-2
□	GOTERM_BP_FAT	epidermis development	RT	▬	8	7.3	5.4E-4	1.4E-1
□	GOTERM_BP_FAT	cell adhesion	RT	▬	15	13.8	9.1E-4	1.9E-1
□	GOTERM_BP_FAT	biological adhesion	RT	▬	15	13.8	9.3E-4	1.6E-1
□	GOTERM_BP_FAT	aging	RT	▬	6	5.5	1.6E-3	2.3E-1
□	GOTERM_BP_FAT	response to steroid hormone stimulus	RT	▬	7	6.4	3.6E-3	4.0E-1
□	GOTERM_BP_FAT	response to temperature stimulus	RT	▬	5	4.6	3.8E-3	3.8E-1
□	GOTERM_BP_FAT	pyruvate metabolic process	RT	▬	4	3.7	4.1E-3	3.7E-1
□	GOTERM_BP_FAT	positive regulation of cell proliferation	RT	▬	10	9.2	4.6E-3	3.8E-1
□	GOTERM_BP_FAT	glucose metabolic process	RT	▬	6	5.5	6.5E-3	4.6E-1
□	GOTERM_BP_FAT	negative regulation of cell proliferation	RT	▬	9	8.3	6.6E-3	4.4E-1

Figure 3.6: Snapshot of Functional Annotation chart

hydrates binding, ATPase activity etc. of the gene products at the molecular level. A **GO term** is annotated to a group of genes responsible for a particular biological activity.

p values

The significance of a bicluster i.e. the likelihood that a bicluster is not found by chance can be measured by statistical measures like p value. p values are calculated to measure the statistical significance of functional category enrichment. The GO terms shared by the genes in the user's list are compared to the background distribution of the annotation. It is the probability of seeing x or more genes from the input list of n genes annotated to a particular GO term, given the proportion of genes in the whole genome annotated to that GO Term is F out of G. Specifically, hyper geometric distribution is used to calculate the probability of observing at least x or more genes from a functional category from an input gene list of size n given the background database consists of G genes out of which F belong to the functional category.

$$p\ value = \sum_{j=x}^{n} \frac{\binom{F}{j}\binom{G-F}{n-j}}{\binom{G}{n}} \tag{3.1}$$

This is same as calculating the chance of getting atleast x successes and can also be represented as

$$p\ value = 1 - \sum_{j=0}^{x-1} \frac{\binom{F}{j}\binom{G-F}{n-j}}{\binom{G}{n}} \tag{3.2}$$

It is clear that smaller the p value, more significant is the association of the particular GO term with the group of genes (i.e. it is less likely that the observed annotation of the particular GO term to a group of genes occurs by chance). There may be several GO terms with different p values associated with an input set of genes belonging to a bicluster. The best p value for each category may be used to compare the biclusters.

3.5.3 Motif analysis using RSA Toolbox

A set of genes showing similar behavior indicates that they are active or expressed together. As explained in Chapter 2, a gene becomes active when a **transcription factor** (protein responsible for gene regulation) binds to a **Transcription Factor Binding Site (TFBS) or motif** in the promoter region of the gene. Thus the genes responsible for one biological activity and hence belonging to a bicluster are expected to have shared elements/patterns/motifs. In order to further validate our biclusters we performed motif analysis of the genes of the biclusters using **Requence Sequence Analysis Toolbox (RSAT)**. **RSAT** consists of many modular tools for sequence retrieval and motif discovery. These tools can either be accessed separately or be connected in a pipeline. Two of these tools are **Retrieve Sequence Tool (RST)** and **Motif Discovery Tool (MDT)**.Figure 3.7 summarizes the working of RSAT. A set of genes along with the name of the organism is provided as an input to RST as shown in Figure 3.8. RST provides the sequences of the input genes as output which is then fed to MDT to extract the motifs. The ouput of MDT includes the motifs and their corresponding E **values** as shown in Figure 3.9. The E

value gives the statistical significance of the motif detected. It is the expected number of times a similarity would be observed by chance in a target database of random motifs. It is obtained by multiplying the probability of atleast n occurrences when expecting x by the number of distinct patterns. Smaller the E value more significant is the motif detected.

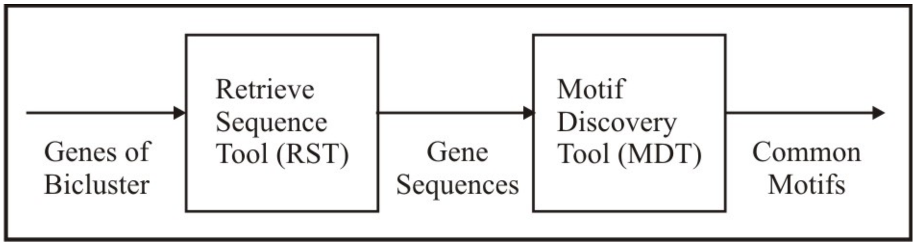
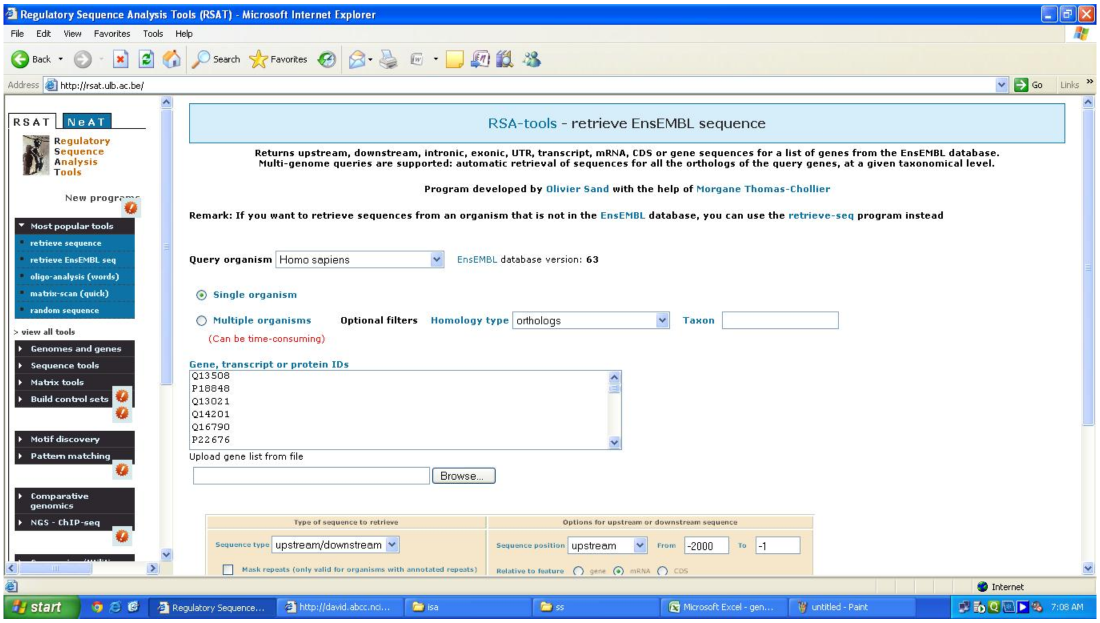


Figure 3.7: Motif analysis using RSAT


Regulatory Sequence Analysis Tools (RSAT) - Microsoft Internet Explorer

Address http://rsat.ulb.ac.be/

RSA-tools - retrieve EnsEMBL sequence

Returns upstream, downstream, intronic, exonic, UTR, transcript, mRNA, CDS or gene sequences for a list of genes from the EnsEMBL database.
Multi-genome queries are supported: automatic retrieval of sequences for all the orthologs of the query genes, at a given taxonomical level.

Program developed by Olivier Sand with the help of Morgane Thomas-Chollier

Remark: If you want to retrieve sequences from an organism that is not in the EnsEMBL database, you can use the retrieve-seq program instead

Query organism Homo sapiens EnsEMBL database version: 63

● Single organism

○ Multiple organisms Optional filters Homology type orthologs Taxon
(Can be time-consuming)

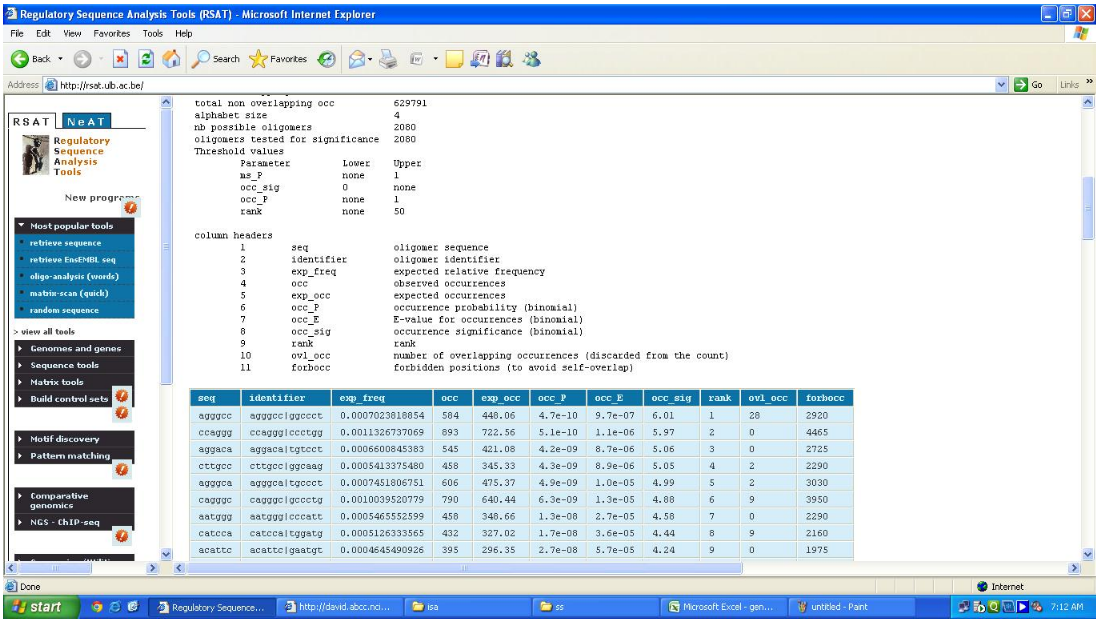
Gene, transcript or protein IDs
Q13508
P18848
Q13021
Q14201
Q16790
P22676

Upload gene list from file

Type of sequence to retrieve
Sequence type upstream/downstream
□ Mask repeats (only valid for organisms with annotated repeats)

Options for upstream or downstream sequence
Sequence position upstream From -2000 To -1
Relative to feature ○ gene ● mRNA ○ CDS

Figure 3.8: Snapshot of Retrieve Sequence Analysis Tool



```
total non overlapping occ          629791
alphabet size                      4
nb possible oligomers              2080
oligomers tested for significance  2080
Threshold values
        Parameter        Lower    Upper
        ms_P             none     1
        occ_sig          0        none
        occ_P            none     1
        rank             none     50

column headers
        1        seq              oligomer sequence
        2        identifier       oligomer identifier
        3        exp_freq         expected relative frequency
        4        occ              observed occurrences
        5        exp_occ          expected occurrences
        6        occ_P            occurrence probability (binomial)
        7        occ_E            E-value for occurrences (binomial)
        8        occ_sig          occurrence significance (binomial)
        9        rank             rank
        10       ovl_occ          number of overlapping occurrences (discarded from the count)
        11       forbocc          forbidden positions (to avoid self-overlap)
```

seq	identifier	exp_freq	occ	exp_occ	occ_P	occ_E	occ_sig	rank	ovl_occ	forbocc
agggcc	agggcc|ggccct	0.0007023818854	584	448.06	4.7e-10	9.7e-07	6.01	1	28	2920
ccaggg	ccaggg|ccctgg	0.0011326737069	893	722.56	5.1e-10	1.1e-06	5.97	2	0	4465
aggaca	aggaca|tgtcct	0.0006600845383	545	421.08	4.2e-09	8.7e-06	5.06	3	0	2725
cttgcc	cttgcc|ggcaag	0.0005413375480	458	345.33	4.3e-09	8.9e-06	5.05	4	2	2290
agggca	agggca|tgccct	0.0007451806751	606	475.37	4.9e-09	1.0e-05	4.99	5	2	3030
cagggc	cagggc|gccctg	0.0010039520779	790	640.44	6.3e-09	1.3e-05	4.88	6	9	3950
aatggg	aatggg|cccatt	0.0005465552599	458	348.66	1.3e-08	2.7e-05	4.58	7	0	2290
catcca	catcca|tggatg	0.0005126333565	432	327.02	1.7e-08	3.6e-05	4.44	8	9	2160
acattc	acattc|gaatgt	0.0004645490926	395	296.35	2.7e-08	5.7e-05	4.24	9	0	1975

Figure 3.9: Snapshot of motif discovery tool of RSAT

3.6 Datasets

We considered the real datasets used by Prelic et al. [PBZ$^+$06] and by Hochreiter et al. [HBH$^+$10]. *Saccharomyces cerevisiae* also known as brewer's yeast is a safe, easy to grow, short generation time organism. [Hun93]. As yeasts are eukaryotes and are biochemically similar to humans, they are quite popular with biologists for study purposes. Yeast datasets examines gene expression behaviour during various stress conditions. Expression profiles were normalized (subtracting the mean of each profile and dividing it by the standard deviation across the time points). Another popularly studied organism is *Arabidopsis thaliana*. It is a common weed which undergoes the same processes of growth, development, flowering etc. as most of the higher plants yet has a small genome. It produces a large number of seeds and grows to a mature plant in only about six weeks. We studied the expression dataset of *Saccharomyces cerevisiae*, *Arabidopsis thaliana* and two datasets of homosapiens. The *Human breast cancer* dataset [VDV$^+$02] aimed at predictive gene signature for the outcome of a breast cancer therapy. The *Diffuse large B-cell lymphoma* dataset [RWC$^+$02] contained the gene expression profiles of the lymphomas of patients after chemotherapy. Table 3.5 gives the details about the datasets used.

Dataset	Genes	Samples	source
Arabidopsis thaliana	619	72	www.tik.ee.ethz.ch/sop/bicat
Saccharomyces cerevisiae	2993	173	www.tik.ee.ethz.ch/sop/bicat
Diffuse large-B-cell lymphoma	661	180	www.bioinf.jku.at/software/fabia
Human breast cancer	1213	97	www.bioinf.jku.at/software/fabia

Table 3.5: Gene Expression Datasets

Chapter 4

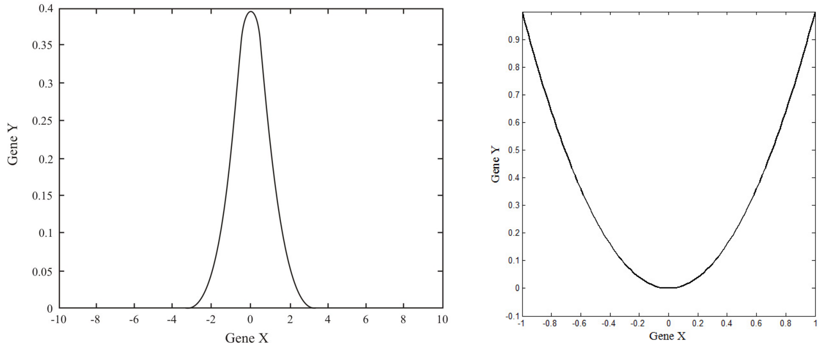
Mutual Information

Result of any biclustering algorithm depends on the choice of similarity measure. Different similarity measures on the same expression data produce different results. Most of the existing algorithms [CC00, IFB$^+$02, WWYY02, GLD00, KBCG03, KTW05, LW07, BIB03] for biclustering use Euclidean distance, Manhattan distance, correlation coefficient or variance as a measure of similarity(/dissimilarity). Though these measures have been successfully and satisfactorily used for several years they capture only the linear relationships between the objects. In particular, a vanishing correlation coefficient implies absence of only linear dependencies [HG95, PMBG07, KBG$^+$07, SKD$^+$02, SDSK03]. However, nonlinear relationships like quadratic or sinusoidal etc may exist between the genes. In such a situation, traditional measure of similarity like correlation coefficient and other distance measures would fail. We thus need similarity measures that exploit non linear dependencies to find such complex relationships between the expression values of the genes. In this chapter we present mutual information as a similarity measure to discover non linear relationships between two genes.

4.1 Mutual Information: A general measure of similarity

Many researchers [SKD+02, BK00, MCA+98, ZWD+04, SATB05, PMBG07] have used mutual information for one way clustering of genes. They have shown that the information theoretic measure is responsive to any type of dependencies, including strongly non linear structures. Kraskov et al. [KSG04], Steur et al. [SKD+02], Butte and Kohane [BK00] and Michaels et al. [MCA+98] have shown through their work that mutual information is a better and general criterion for extracting complex relationships. Kraskov et al. worked with yeast data and found that even though correlation coefficient between few gene pairs was zero, the mutual information between them was non zero thus indicating that other non linear dependencies exist between the genes. Steur et al. showed that higher correlation coefficient implies higher mutual information but two variables having very low values of correlation coefficient may still be related to each other. Butte and Kohane also worked with yeast data set. They hypothesized that gene pairs with high mutual information between them are also related biologically. They constructed networks of various genes having high mutual information between them and found that each network corresponded to some biological activity. They also found mutual information to be a better similarity measure as compared to linear correlation coefficient. Michaels et al. used both Euclidean distance and mutual information as a measure of similarity for finding association between genes belonging to mammalian central nervous system. They conjectured that genes, which share common control inputs or operate together i.e. are a part of some biological network like signaling pathway or metabolic network, are members of same gene sequence family and are regulated in a similar manner. According to them some genes may share inputs but respond differently to those inputs and only mutual information is able to identify their coordinated changes. In [PMBG07] Priness et al. also showed that the mutual information is a more generalized measure of statistical dependence and is resistant to outliers and missing data and give better quality clusters. With some procedural modifications they incorporated mutual information measure in some clustering algorithms like k-means, self organized maps, click and sIB [Koh97, SS02, Slo02b]. They found that the clusters obtained from these algorithms using mutual information were similar to each other but different from the clusters obtained when using different distance measures with these algorithms, once again endorsing the need of a different similarity measure.

To summarize, mutual information is a more general measure of association between two random variables. When the underlying relationship is nonlinear [CT91, ZAA08, PMBG07], it outperforms the conventional measures of similarity. Zero mutual information indicates that the genes do not have any kind of dependence. This inference cannot be made using any other distance measure or correlation coefficient.



(a) Expression values of two genes Gene Y vs Gene X, when one gene Gene Y is expressed at midrange values of Gene X

(b) Expression values of two genes Gene Y vs gene X, when one gene Gene Y is a nonlinear function of the expression of Gene X

Figure 4.1: Nonlinear relationships between expression of two genes

Consider a scenario where a gene is expressed only at some midrange values of another gene. In such a case the curve between the two genes as shown in Figure 4.1(a) resembles a normal distribution curve and the correlation between the two genes is very low. However mutual information is able to capture this relationship.

Next, consider a gene X whose expression values are uniformly distributed ranging from (-5,5). Let Y be another gene whose expression value is related to gene X as $y = x^2$ as shown in Figure 4.1(b). The expression matrix of the two genes is shown in Table 4.1.

	c1	c2	c3	c4	c5	c6	c7	c8	c9	c10
g1	-5	-4	-3	-2	-1	1	2	3	4	5
g2	25	16	9	4	1	1	4	9	16	25

Table 4.1: Expression matrix of two genes with nonlinear relationship

The value of distance between X and Y turns out to be quite large and the value of Pearsons correlation coefficient turns out to be very small (almost zero) showing no relationship between X and Y. Even the Spearman's correlation coefficient between X and Y, which Ayadi et al. [AEH09] claim is a better similarity measure than the other two, turns out to be zero. Thus a vanishing correlation coefficient implies absence of linear dependencies but not complete independence. On the contrary the value of mutual information between the two genes turns out be significant. Thus mutual information is a more general criterion to investigate relationships (positive, negative correlation and non linear dependencies) between variables.

4.2 Theoretical concepts

In this section we introduce the theoretical concepts behind Mutual Information. We first introduce the concept of Entropy which is like the self information of a random variable followed by the introduction of mutual information which is a special case of relative entropy. We further explain the various advantages of mutual information followed by methods to estimate mutual information [Hay07, Bis06, CT91].

4.2.1 Entropy

Entropy is the measure of uncertainty associated with a random variable. It quantifies the expected amount of information contained in a message, usually in units such as bits. The concept was introduced by Claude E. Shannon in his paper ”A Mathematical Theory of Communication” [Sha48] hence it is also called **Shannon Entropy**. Consider a series of coin tosses with a fair coin. The probability of seeing a head and that of seeing a tail are same both being equal to $1/2$. The system has maximum entropy, since it is most difficult to predict the outcome of the next toss. The outcome of each toss gives 1 bit $(-\log \frac{1}{2})$ of information. This is not the case when the coin is not fair. One of the sides is more likely to come up as compared to the other in the next toss i.e. the element of uncertainty is less in this case. A string of coin tosses with a two-headed coin has zero entropy, since the coin will always come up heads. Mathematically, $-\sum p_i \log p_i$ best captures these facts. When the coin is fair, average amount of uncertainty (information) contained in the system is maximum i.e. $-\sum \frac{1}{2} \log \frac{1}{2} = 1$. When the two probabilities are $(p(H), p(T)) = (\frac{1}{4}, \frac{3}{4})$, the amount of information (surprise) obtained on seeing a less probable event (heads, in this case) is more $(-\log \frac{1}{4})$ than that $(-\log \frac{3}{4})$ on seeing a more probable event (tails). The average amount of uncertainty (information) contained in the system is

$$\frac{1}{4}\log 4 + \frac{3}{4}\log\frac{4}{3} = \frac{1}{4}\log 4 + \frac{3}{4}\log 4 - \frac{3}{4}\log 3 < 1$$

The inequality follows as $\frac{3}{4}\log 3 > 1$

Let X be a random vector having probability distribution $p_i = P(X = x_i)$, $i = 1 \dots N$ where N is the number of possible values X can take. Let the information gained by observing x be $h(x)$. If x is a highly probable event then we do not gain much information on its occurrence. However if x is a highly unlikely event then we gain a lot of information on its occurrence. Also, for two unrelated events x and y the information gained by their occurrence is the sum total of information from their individual occurrences i.e.

$$h(x,y) = h(x) + h(y)$$

Also if two events x and y are totally independent then their joint probability can be written as the product of their marginal probability as follows

$$p(x,y) = p(x) * p(y)$$

Motivated by this, Shannon derived the following formula for Entropy:

$$H(p(x)) = -\sum_i p_i \log(p_i)$$

4.2.2 Relative Entropy or The Kullback Leibler Divergence

Suppose an unknown distribution $p(x)$ is modeled with an assumed distribution $p^0(x)$. The difference in the amount of information given by $p^0(x)$ and that given by $p(x)$ is called the Kullback Leibler Divergence [KL51] or relative entropy and is given by

$$K(p/p^0) = -\int p(x)\log(p^0(x))dx - (-\int p(x)\log(p(x))dx) = -\int p(x)\log\frac{p^0(x)}{p(x)}$$

Using $\log(x)$ as a convex function and applying Jensons inequality for convex functions [1], we have,

$$K(p/p^0) = -\int p(x)\log\frac{p^0(x)}{p(x)} >= -\log\int p^0(x)dx = 0$$

i.e KL divergence is always non-negative. It equals zero if and only if $\{p^0\}$ and $\{p\}$ are same.

4.2.3 Mutual Information: Definition

The mutual information between two random variables X and Y is a measure of information contained in X about Y or the information contained in Y about X. It is the

[1] a function f is said to be convex if $f(\sum_{i=1}^{N} \lambda_i x_i) <= \sum_{i=1}^{N} \lambda_i f(x_i)$ where $\lambda_i \ >= 0$ and $\sum_i \lambda_i = 1$

reduction in the uncertainty of one random variable due to the knowledge of the other. If given a value of X, it is easy to predict the value of Y then X contains good amount of information about Y. Clearly with this definition, if X and Y are independent, the mutual information between them is zero and it is high if they are highly dependent or closely related to each other.

Kullback [Kul68] defined mutual information between two random variables as a measure of divergence of the observed joint distribution of X and Y from the hypothesis that X and Y are independent. If the joint probability distribution function of X and Y is given by $p_{XY}(x, y)$ and the marginal distributions of X and Y by $p_X(x)$ and $p_Y(y)$ respectively then using Kullback Leibler Divergence, mutual information (MI) between X and Y is given by

$$I(X, Y) = \int_x \int_y p_{XY}(x, y) \log \frac{p_{XY}(x, y)}{p_X(x) p_Y(y)} dx dy \tag{4.1}$$

The unit of mutual information is defined corresponding to the base of the logarithm in the above equation i.e. nats for log_e, bits for log_2, and Hartleys for log_{10}. Mutual information is non negative and symmetrical i.e. $I(X, Y) = I(Y, X)$. Also, mutual information is zero if and only if X and Y are statistically independent i.e. vanishing mutual information does imply that the two variables are independent. However, it is not a true distance between distributions as it does not satisfy the triangle inequality.

4.3 Estimating Mutual Information

As discussed in the previous section mutual information is a function of probability densities. However, one generally does not have prior knowledge about the distributions. Thus, estimating mutual information between two genes requires obtaining the estimate $\hat{p}(x, y)$ of the joint probability distribution and $\hat{p}(x),\ \hat{p}(y)$ of the marginal probability distribution of their expression values. Then, on substituting the values in the expression for mutual information in Equation 4.1 we get the following estimates for mutual inofrmation

$$\hat{I}(X,Y) = \int_x \int_y \hat{p}(x,y) \log \frac{\hat{p}(x,y)}{\hat{p}(x)\hat{p}(y)} dxdy$$

If the individual observations are independent realizations of the underlying distribution then the expression for mutual information can be approximated as follows.

$$\hat{I}\ (X,\ Y) = \frac{1}{n} \sum_{j=1}^{n} \log \frac{\hat{p}\ (x_j,\ y_j)}{\hat{p}\ (x_j)\ \hat{p}\ (y_j)} \tag{4.2}$$

where x_j, y_j are n independent realizations of the random variable X and Y respectively. Two broad classes of approaches are used to estimate the probability distribution functions namely parametric and nonparametric [Hay07, Bis06, CT91]. Parametric method involves assuming a model for the probability density function and then determining the various parameters from the data. However, if the assumption is poor the results are poor. In contrast to the parametric approach no assumption about the underlying distribution is made in the nonparametric approach. Histogram method [BK00] and Kernel density estimation [MRL95] are two methods of estimating probability density function by the nonparametric approach. We briefly explain the histogram method followed by the kernel density estimation method.

4.3.1 Histogram method

Consider a series (x_t, y_t) of n observations of two random variables X and Y. Given an origin o (which could be different for different variables resulting in different bins), define bins a_i of width h for X, as the intervals $[o + ih, o + (i+1)h],\ i = 1 \ldots N_x$,Let $f_X(i)$ denote the number of observations of X falling in the bin a_i. The probabilities $\{p(a_i)\}$ are then given by

$$p(a_i) = \frac{f_X(i)}{n}$$

Similarly define bins b_j of the random variable Y, j = 1 . . . N_y. Let $f_Y(j)$ denote the number of observations of Y falling in the bin b_j. The probabilities $\{p(b_j)\}$ are then given by

$$p(b_j) = \frac{f_Y(j)}{n}$$

Let $f_{XY}(i,j)$ denote the number of observations such that X falls in bin a_i and Y falls in bin b_j. The joint probabilities $\{p(a_i, b_j)\}$ are then given by

$$p(a_i, b_j) = \frac{f_{XY}(i,j)}{n}$$

Then the mutual information between X and Y is estimated as

$$\begin{aligned}
I(X,Y) &= \sum_{ij} p(a_i,b_j) \log \frac{p(a_i,b_j)}{p(a_i)p(b_j)} \\
&= \sum_{ij} \frac{f_{XY}(i,j)}{n} \log \frac{\frac{f_{XY}(i,j)}{n}}{\frac{f_X(i)}{n}\frac{f_Y(j)}{n}} \\
&= \sum_{ij} \frac{f_{XY}(i,j)}{n} \log \frac{n * f_{XY}(i,j)}{f_X(i) * f_Y(j)} \\
&= \sum_{ij} \frac{\log(n) * f_{XY}(i,j)}{n} + \sum_{ij} \frac{f_{XY}(i,j)}{n} \log \frac{f_{XY}(i,j)}{f_X(i) f_Y(j)} \\
&= \log(n) + \frac{1}{n} \sum_{ij} f_{XY}(i,j) \log \frac{f_{XY}(i,j)}{f_X(i) f_Y(j)}
\end{aligned}$$

4.3.2 Kernel Density Estimation method

According to Steur et al. [SKD$^+$02], if the number of data points is sufficiently large, histogram method gives fairly accurate results but if the number of datapoints is small then systematic errors may creep in because of finite size of data. The method is also sensitive to the choice of bin width. If the bin width is very large then the actual distribution of the data may be missed. On the other hand if the value of the bin width is chosen to be very small then the resulting distribution may be very spiky. Thus the bin width should be neither too large nor too small. In [Sil86] Silverman showed that the histogram method was also sensitive to the choice of origin. If we attempt to construct the histogram such that every point is the centre of the sampling interval, the histogram becomes independent from a particular choice of the bin position and origin. However, choice of rectangular shaped bins impact the probability estimate because of their discrete nature and discontinuties at the boundaries. Choosing shapes other than the rectangular bins reduces the impact of discontinuity/dissimilarities at the boundaries and hence provides a better estimate of the probability density function. Kernel density estimation (KDE) allows window shapes other than the rectangular. In fact, Silverman showed that kernel density estimation method is not only independent of the choice of origin but also has a better mean square error rate of convergence. Moon et al. [MRL95] also showed that KDE is independent of the choice of origin of the bins.

In this section, we will describe how to estimate probabilities using kernel density estimator. Let p denote the probability density function of a random variable X then

$$p\ (x) = \lim_{h \to 0} \frac{1}{2h} P\ (x - h < X < x + h)$$

where $P(x - h < X < x + h)$ is the probability that X lies in the interval $x - h$ to $x + h$. For any given h, we can estimate $P\ (x - h < X < x + h)$ by the proportion of observations $x_i' s$ falling in the interval $(x - h,\ x + h)$. Thus a natural or a naive estimator $\hat{p}\ (x)$ of the density is given by choosing a small number h and setting

$$\hat{p}\ (x) = \frac{1}{2nh}\ [\, number\ of\ x_1,\ x_2, ...x_n\ falling\ in\ (x - h,\ x + h)]$$

This is called the naive estimator. With the generalized weight function $w\ (x)$ given by

$$w\ (x) = \{ {}^{\frac{1}{2} \quad if\ |x| < 1}_{0 \quad otherwise}$$

the naive estimator can be written as

$$\hat{p}\ (x) = \frac{1}{n} \sum_{i=1}^{n} \frac{1}{h}\ w\ (\frac{x - x_i}{h})$$

On replacing the weight function by a kernel density function which satisfies the condition,

$$\int_{-\infty}^{+\infty} K\ (x)\ dx = 1$$

the kernel density estimator with kernel function K can be written as

$$\hat{p}\ (x) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{h}\ K\ (\frac{x - x_i}{h})$$

The naive estimator can be considered as a sum of boxes centered at the observations, and the kernel estimator as the sum of bumps placed at the observations where the shape of the bumps is determined by the kernel function K and the window width h also called the *smoothing parameter*, determines the width of the bumps. The Gaussian kernel function centered at the origin and unit variance is given by

$$K\ (x) = \frac{1}{\sqrt{2\pi}}\int exp\ (\frac{-x^2}{2})$$

Thus using Gaussian kernel the density estimates are given as follows:

$$\hat{p}\ (x) = \frac{1}{nh\sqrt{2\pi}}\sum_{i=1}^{n} exp\ (\frac{-(x - x_i)^2}{2h^2})$$

Similarly, the Gaussian kernel may be used to estimate the joint probability density function as

$$\hat{p}\ (x,\ y) = \frac{1}{2\pi nh^2}\sum_{i=1}^{n} exp\ (\frac{-1}{2h^2}[\ (x - x_i)^2 +\ (y - y_i)^2])$$

In [MRL95] Moon et al. defined an optimal value of h as the one that minimizes the mean integrated square error, assuming the underlying distribution is Gaussian. Following Silverman [Sil86] the optimal Gaussian bandwidth h_{opt} for marginal probability distribution is given by

$$h\ =\ (\frac{4}{3n})^{\frac{1}{5}}\sigma \approx 1.06\sigma n^{\frac{-1}{5}}$$

where σ denotes the standard deviation of the data. And, the value of h_{opt} for joint probability distribution is given by

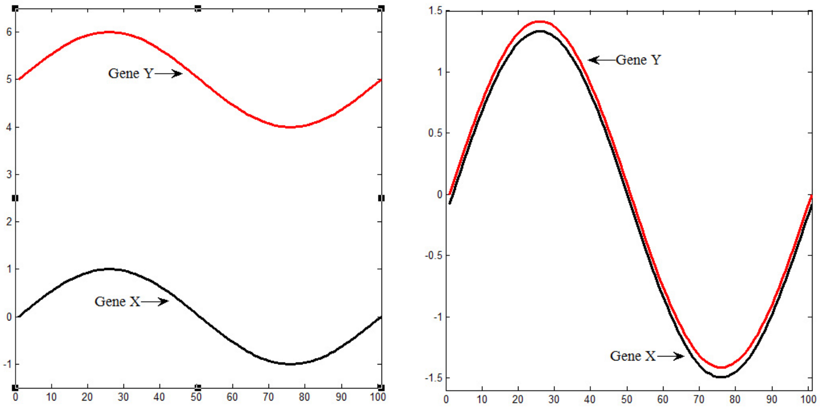
$$h = \sigma n^{\frac{-1}{6}}$$

where σ denotes the average marginal standard deviation.

4.4 Advantages of Mutual Information

Besides the fact that mutual information captures non-linear relationship, it enjoys several other benefits. As mutual information is based on the distribution of data rather than the actual values, it does not require normalization and it is robust towards noise, outliers and missing data. This is explained below:

1. Normalization of data: Mutual inoformation is not sensitive to shifting and scaling unlike other similarity measures like Euclidean distance [JTZ04, DLS99].



(a) Expression values of two genes before normalization

(b) Expression values of two genes after normalization

Figure 4.2: Normalization of data

In a microarray experiment two genes may have similar expression pattern yet their expression values may not be directly comparable [CQB04] as shown in Figure 4.2(a). There may be several reasons attributing to this gap in the expression values for example the data may be collected from different labs or quantity of starting mRNA may not be same for the samples. Differences may also occur while labeling and detecting efficiencies for the fluoroscent labels. Additional systematic errors can also alter the expression levels. Various distance measures like Euclidean distance will not find any relationship between the two genes and would put them in different clusters despite the fact that they have similar pattern of expression. To be able to use these measure to extract the common pattern, the expression levels of the genes must be normalized. After normalization, the expression of the two genes appears as shown in Figure 4.2(b) and they become comparable. As mutual information exploits the distribution in the expression level of the genes rather than their actual values, it turns out to be high both before and after normalization.

2. Robust to Outliers: Mutual information is not sensitive towards outliers whereas measures like correlation coefficient and Euclidean distance are. Let us take the scenario when two genes have a very high expression value under a condition as compared to other conditions, then the correlation between them will be high irrespective of their expression under other conditions. Consider two gene with the expression values as shown in Table 4.2.

	c1	c2	c3	c4	c5	c6	c7	c8	c9	c10	c11
g1	-5	-4	-3	-2	-1	1	2	3	4	5	100
g2	25	16	9	4	1	1	4	9	16	25	100

Table 4.2: Expression matrix of two genes with an outlier

The correlation between the two genes turns out to be quite high (≈ 1). However, if we remove condition $c11$ then the correlation turns to be zero. On the other hand mutual information between the two genes turns out to be 1.5 and 1.7 with and without condition $c11$ respectively.

3. Noise in gene expression data : A lot of noise is generally present in the gene expression data due to experimental errors like differences and impurities in the biological samples. The actual expression values change because of the presence of noise. Mutual information is robust towards noise. In Figure 4.3 we show the

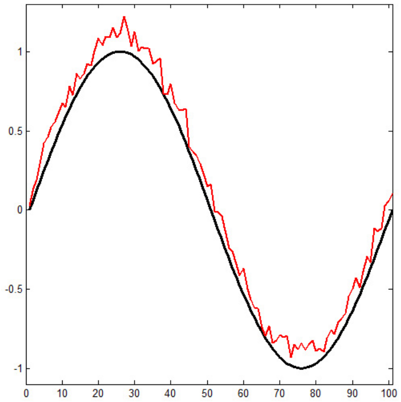


Figure 4.3: Expression values of a gene perturbed by noise

expression of two genes X and Y where Y is related to X as $Y = sin(X)$. Noise was simulated by generating random numbers from normal distribution with mean $\mu = 0.1$ and standard deviation σ varying from $.001$ to $.09$. Expression values were perturbed by adding the random numbers generated for different levels of noise. The curve between X and Y gets distorted due to noise as shown in Figure 4.3. The mutual information calculated between X and Y before and after perturbation of Y is of the same order.

4. Missing values in data: Dust particles, scratches on the microarray plate, errors in measuring the intensity often result in missing values in the gene expression data. Measures like Euclidean distance are sensitive towards missing data whereas mutual information is not.

Chapter 5

MRB: Extracting Maximum Related Biclusters

In this chapter we describe our study and results of the use of mutual information to extract biclusters from gene expression data. We simply plugged mutual information as a similarity measure in the algorithm Maximum Similarity Biclustering (MSB) by Liu et al. [LW07] to obtain what we call as Maximum Related Biclustering (MRB) algorithm [GA09]. Liu et al. have posed biclustering as an optimization problem where in they define a similarity score (based on Euclidean distance) for a bicluster and try to obtain a bicluster that maximizes the score. Starting with the entire matrix as a bicluster, a row or a column that contributes least to the score of the bicluster is removed iteratively. Finally, the bicluster with maximum score out of all the biclusters obtained is selected as output. We proposed a similarity score based on mutual information instead of Euclidean distance.

The algorithm MRB was tested both on synthetic data as well as real datasets. The main idea behind the construction of the synthetic data was to model **nonlinear** relationships between the genes of a bicluster **over a subset of conditions**. We were able to extract the implanted biclusters in the synthetic data whereas the distance based MSB could not. We also obtained biclusters from different datasets namely the expression datasets of *Arabidopsis thaliana, Diffuse large B-cell lymphoma* data set, *Human breast cancer* data, and *Saccharomyces cerevisiae*. Our biclusters were found to be significantly enriched with GO categories with very small p values, hence endorsing that mutual information is an effective similarity measure. To further biologically validate our biclusters, we searched for common patterns (motifs) from the promoter regions of the genes belonging to a bicluster. Promoter regions of the genes of most of the biclusters were found to have statistically significant (smaller E value) common motif patterns.

5.1 The Bicluster Score

Let G be the set of genes and C be the set of conditions in the expression matrix E. The number of genes in G is denoted by N_g and the number of conditions in C by N_c. We want to find biclusters which are tuples (G', C'), where G' is the subset of genes which are most closely related to a gene seed (g_*) under the subset C' of conditions and C' is the subset of conditions under which the genes in the set G' are most closely related to the gene seed.

For a gene seed g_*, let m_{ij} denote the contribution of the j^{th} condition towards the mutual information of the i^{th} gene with g_*, then from Equation 4.2 in Chapter 4 we have

$$m_{ij} = \frac{1}{N_c} \log\Big(\frac{\hat{p}(g_{*j}, g_{ij})}{\hat{p}(g_{*j}) \cdot \hat{p}(g_{ij})}\Big)$$

Let M denote the matrix with elements m_{ij} and $\hat{M}$ be the similarity matrix derived from the matrix M by setting smaller values to zero. The $(ij)^{th}$ entry $\hat{m}_{ij}$ of $\hat{M}$ is then given as:

$$\hat{m}_{ij} = \begin{cases} 0 & if\ m_{ij}<\alpha\cdot mi_{avg} \\ \frac{m_{ij}}{\alpha . mi_{avg}}-1 & otherwise \end{cases}$$

where α is a control parameter and

$$mi_{avg} = \frac{1}{N_g N_c} \sum_{i=1}^{N_g} \sum_{j=1}^{N_c} m_{ij}$$

Let $B(I, J)$ denote a bicluster consisting of a set I of rows and set J of columns. The score $s(i, J)$ of row i is then defined as follows

$$s(i, J) = \sum_{j \in J} \hat{m}_{ij} \qquad (5.1)$$

Similarly the score $s(I, j)$ of column j is defined as follows

$$s(I, j) = \sum_{i \in I} \hat{m}_{ij} \qquad (5.2)$$

and the score $s(I, J)$ of the bicluster is defined as

$$s(I, J) = min\{min_{i \in I}\ \{s(i, J)\}, min_{j \in J}\ \{s(I, j)\}\} \qquad (5.3)$$

The aim is to extract a bicluster with maximum score.

5.2 MRB: Biclustering for maximum related biclusters

In this section we describe our algorithm MRB. The broad overview of MRB is given in Algorithm 1. The algorithm starts by selecting a gene seed g_* randomly and taking the whole expression matrix as a bicluster. It then iteratively eliminates a row or a column (one with the least score) that contributes least to the similarity score of the bicluster till a bicluster having one row and one column is left. At each stage the score of the bicluster obtained is calculated. At the end, the bicluster with the maximum similarity score is selected as output. More biclusters are obtained by running the algorithm with different gene seeds.

Selecting the well separated gene seeds: We incorporate a method to intelligently select well separated gene seeds from the expression data. The number of gene seeds need not be specified by the user as is generally required by many clustering algorithms. It is determined by the algorithm from the data itself. Let g_*^i denote the gene seed in the i^{th} iteration and G_i denote the set of genes most related to g_*^i. Let S_i be the subset of gene seeds at the beginning of the $(i+1)^{th}$ iteration. Initially $S_o = \phi$. The first gene seed g_*^1 is chosen randomly for the first iteration and added to S_o to form S_1. The $(i+1)^{th}$ gene seed g_*^{i+1} is selected as the gene which is unclustered and is least related with the gene seeds chosen so far (i.e. with the genes in S_i). For this we determine the maximum relatedness of all the unclustered genes with the genes in S_i and then choose the one that is related least to the gene seeds in S_i, i.e.

$$g_*^{i+1} = argmin\{max_{g_* \in S_i} mi\ (\ g,\ g_*)\}$$

where minimum is taken over $g \in G - \cup_{k \leq i} G_k$ and $mi(x,\ y)$ denotes the pairwise mutual information between two genes x and y over all the conditions.

Procedure $MRB()$ in Algorithm 2 shows the detailed algorithm. The method of selecting well separated gene seeds is summarized in Procedure $get\_next\_gene\_seed()$. Gene score of a gene g denoted by $gscore(g)$ and the condition score of a condition c denoted by $cscore(c)$ are computed according to Equations 5.1 and 5.2 respectively.

We prove that for a given gene seed, our algorithm MRB extracts the optimal bicluster i.e. the bicluster $B(I, J)$ for which $s(I, J)$, as defined in Equation 5.3 is maximum. For this we prove the following lemma first.

Lemma 5.1 *Let $B(I_1, J_1)$ and $B(I_2, J_2)$ are two biclusters in the expression matrix such that $I_1 \subseteq I_2 \subseteq I$ and $J_1 \subseteq J_2 \subseteq J$. Then, for each row i and each column j, we have $s(i, J_1) \leq s(i, J_2)$ and $s(I_1, j) \leq s(I_2, j)$.*

Proof As all the columns in J_1 are contained in J_2, for each row i we have

$$\begin{aligned} s(i, J_2) - s(i, J_1) &= \sum_{j \in J_2} \hat{m}_{ij} - \sum_{j \in J_1} \hat{m}_{ij} \\ &= \sum_{j \in J_2 \setminus J_1} \hat{m}_{ij} \geq 0 \end{aligned}$$

The last inequality follows as $\hat{m}_{ij} \geq 0 \, \forall i$ and $\forall j$. Therefore $s(i, J_1) \leq s(i, J_2)$. Similarly, $s(I_1, j) \leq s(I_2, j)$. ■

Theorem 5.2 *For a given gene seed, the MRB algorithm outputs an optimal bicluster.*

Proof We prove the claim by contradiction. Suppose that our MRB algorithm outputs the bicluster $B(I_S, J_S)$ which is not optimal. Then there exists a bicluster $B(I^*, J^*)$ such that $s(I^*, J^*) > s(I_S, J_S)$ and $B(I_S, J_S) \neq B(I^*, J^*)$. Let the biclusters obtained by our algorithm be denoted by $B(I_l, J_l), \ l = 1 \dots N_g + N_c - 1$, in the order in which they were obtained.

Since $s(I^*, J^*) > s(I_S, J_S) = max_{1 \leq l \leq N_g + N_c - 1} s(I_l, J_l)$, therefore $B(I^*, J^*) \neq B(I_l, J_l) \, \forall \, l$. In particular, $B(I^*, J^*) \neq B(I_{N_g+N_c-1}, J_{N_g+N_c-1})$. Thus at least one row $i \in I^*$ or one column $j \in J^*$ is not in $B(I_{N_g+N_c-1}, J_{N_g+N_c-1})$ i.e. the algorithm must have removed it in some iteration.

Let k be the first iteration in which any row/column of $B(I^*, J^*)$ was removed by the algorithm, i.e. $B(I^*, J^*)$ is a sub-matrix of $B(I_l, J_l), l = 1 \dots k$ and it is not a sub-matrix of $B(I_{k+1}, J_{k+1})$. Without loss of generality, let us assume that $i \in I^*$ was removed from I_k to get I_{k+1}. The case when a column of J^* is removed can be handled analogously. As $J^* \subseteq J_k$, by Lemma 5.1, we have

$$s(i, J^*) \leq s(i, J_k).$$

Also, as $i \in I^*$, by definition of $s(I, J)$ we have

$$s(I^*, J^*) \leq s(i, J^*)$$

Further, as row i is selected to be removed from $B(I_k, J_k)$, it has the minimum similarity score amongst all the rows and columns of $B(I_k, J_k)$. i.e.

$s(I_k, J_k) \ = min\{min_{i \in I_k} \{s(i, J_k)\}, min_{j \in J_k} \{s(I_k, j)\}\} = s(i, J_k)$.

Therefore we get $s(I^*, J^*) \leq s(i, J^*) \leq s(i, J_k) = s(I_k, J_k) \leq s(I_S, J_S)$

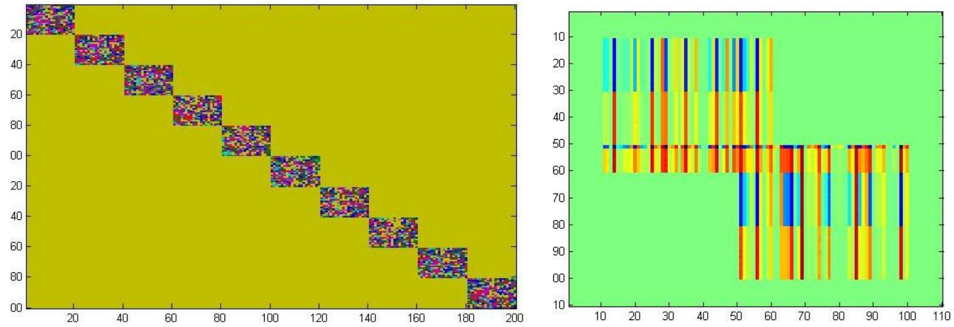
This is a contradiction to our assumption that $s(I^*, J^*) > s(I_S, J_S)$. ■

Time Complexity: In the first step of the algorithm, an $N_g \times N_c$ MI matrix is computed. The time required to compute each element of the matrix is $O(N_c)$. Thus total time required to compute the entire matrix is $O(N_g \cdot N_c^2)$. Computing the similarity matrix takes $O(N_g \cdot N_c)$ time. Selecting a row or a column with minimum score for deletion requires $O(N_g + N_c)$ time for each iteration. As the total number of iterations is $(N_g + N_c - 1)$ for each gene seed, the total time for this step is $O(k \cdot (N_g + N_c) * (N_g + N_c - 1))$ for k gene seeds. Selecting a gene seed takes $O(1)$ time for the first seed and $O(N_g)$ time for the subsequent seeds thereby incurring a total cost of $k \cdot N_g$ to select all the k seeds. Thus, the overall complexity of the algorithm is $O(k \cdot (N_g^2 + N_g \cdot N_c^2)) = O(k \cdot (N_g \cdot N_c^2))$ whenever $N_g \leq N_c^2$. Table 5.1 gives the actual runtime taken by MRB on real datasets for k set to 10. The table shows that the algorithm scales well with the size of the data set.

5.3 Experimental Results

We implemented our algorithm MRB in C++. The performance was tested both on synthetic data as well as real datasets. The main idea behind the construction of the synthetic data was to model nonlinear relationships between genes of the bicluster over a subset of conditions. The synthetic datasets were constructed for both non overlapping and overlapping biclusters as shown in Figure 5.1. The algorithm was run for different values of α and we were able to extract all the implanted biclusters for $\alpha = 0.1$ whereas MSB could not.

Synthetic data for non overlapping biclusters: We generated synthetic data (as shown in Figure 5.1(a)) containing 10 biclusters. This data was generated by implanting biclusters or sub matrices into a larger background matrix. We first generated a larger background matrix of size 200×200 corresponding to 200 genes and 200 conditions having values drawn from normal distribution (of mean $\mu = .01$ and standard deviation $\sigma = .001$). We then implanted 10 non overlapping biclusters (each exhibiting nonlinear



(a) Non overlapping biclusters

(b) Overlapping biclusters

Figure 5.1: Synthetic datasets for MRB algorithm

relationships like $(x^2,\ x^3,\ x^4\ etc.)$ of genes with some gene seed) of size 20×20 into the background matrix. While implanting, we added the elements of our bicluster with the elements of the background matrix.

Synthetic data for overlapping biclusters: We also created synthetic data with two overlapping biclusters M_1 and M_2 having nonlinear relationships between the genes (as shown in Figure 5.1(b)). We generated a larger background matrix of size 110×110 having values drawn from normal distribution (of mean $\mu = .01$ and standard deviation $\sigma = .001$). We then implanted two overlapping biclusters of size 50×50 in it. The first bicluster M_1 consisted of genes g_{11} to g_{60} and conditions c_{11} to c_{60}. The second bicluster M_2 consisted of genes g_{51} to g_{100} and conditions c_{51} to c_{100}. The genes of both the biclusters had nonlinear relationships over the conditions of the biclusters.

Effect of initial start gene: The algorithm chooses the first gene seed randomly and the subsequent gene seeds are chosen far apart from the already selected gene seeds. We studied the effect of random selection of the initial gene seed on the output biclusters. We were able to extract all the implanted biclusters from both datasets irrespective of the initial gene seed.

Effect of varying control parameter α: To study the impact of control parameter α on the output of MRB, it was varied from $.01$ to $.4$. We found that on varying α, the granularity of the extracted biclusters changes providing large bicluster containing almost all genes for α as small as $.01$ to the implanted biclusters at $\alpha = .1$ and their subsets for larger α.

Effect of noise: By perturbing the synthetic data in Figure 5.1(a) by varying levels of noise we studied how MRB behaves in presence of noise. Noise level was varied from 0.001 to 0.005. We were able to extract more than 90% of all the ten implanted biclusters from the synthetic data.

Real Datasets: We tested our algorithm on our four datasets viz. *Arabidopsis thaliana*, *Saccharomyces cerevisiae*, *Diffuse large B cell lymphoma* and *Human breast cancer* dataset. The biclusters obtained were tested for their biological significance. Table 5.2 summarizes the best p value of the obtained biclusters from various datasets. The table clearly shows that the p values of the biclusters are quite low indicating biologically significant biclusters. We also studied the promoter regions of the genes belonging to a bicluster for finding common motifs. Table 5.3 summarizes the best E value for the motif extracted from the gene sequences of the genes belonging to biclusters extracted by MRB on various datasets. Again the E values are quite low indicating biologically significant biclusters.

```
MRB: Maximum Related Biclustering
Input: Expression matrix $E$ containing a set of genes $G$, a set of conditions $C$ and a
       control parameter $\alpha$.
Output: A set of biclusters $B_i = (G_i, C_i)$.
Mark all genes as $unclustered$; Set $i$ to 1.
while there are genes to be $clustered$ do
    Set $max\_bcscore$ to 0, $G_i \ to \ G$ , $C_i \ to \ C$.
    Get next gene seed $g_*^i$.
    Compute the matrix $M$ of mutual information.
    Compute the similarity matrix $\hat{M}$.
    Compute gene scores and condition scores.
    for k = 1 to $N_g + N_c - 1$ do
        Find the gene $g_{min}$ with minimum gscore() $min\_gs$.
        Find the condition $c_{min}$ with minimum cscore() $min\_cs$.
        Compute bicluster score $bcscore$.
        if $bcscore > max\_bcscore$ then
            Update $max\_bcscore$ and save current bicluster.
        end
        if $min\_cs < min\_gs$ then
            Delete $c_{min}$ and Update gene scores.
        end
        else
            Delete $g_{min}$ and Update condition scores.
        end
    end
    Output bicluster with maximum score.
    Mark the genes of the bicluster as $clustered$; increment $i$.
end
```

Algorithm 1: Overview of MRB algorithm

```
Procedure: MRB
Input: $E,\ G,\ C,\ N_g,\ N_c,\ \alpha.$
Output: $A\ set\ of\ biclusters\ B_i\ =\ (G_i,\ C_i).$
Mark all genes as $unclustered$.
$i\ =\ 1$ ; $g_*^i$ = $random\ ()$; $S\ =\ \phi$.
while there are genes to be clustered do
    $S$ = $S \cup g_*^i$ ; $G_i\ =\ G$ ; $C_i\ =\ C$; $max\_bcscore = 0$.
    $M$ = $Compute\_mi\_matrix\ (g_*^i)$.
    $\hat{M}$ = $Compute\_similarity\_matrix\ (M,\ \alpha)$.
    $Compute\ gene\ scores\ and\ condition\ scores.$
    for k = 1 to $N_g + N_c - 1$ do
        $\hat{x} = argmin_{x \in G_i}\{gscore(x)\}$; $\hat{y} = argmin_{y \in C_i}\{cscore(y)\}$.
        $bcscore = min\{g\_score(\hat{x}),\ c\_score(\hat{y})\}$.
        if $(bcscore\ >\ max\_bcscore)$ then
            $B_i\ =\ (G_i, C_i)$ ; $max\_bcscore\ =\ bcscore$.
        end
        if $(gscore(\hat{x}) > cscore(\hat{y}))$ then
            $update\_bicluster\_n\_score\ (G_i, C_i, \hat{y}, 0)$.
        end
        else
            $update\_bicluster\_n\_score\ (G_i, C_i, \hat{x}, 1)$.
        end
    end
    $output\ (G_i,\ C_i)$; mark all genes $g \in G_i$ as $clustered$.
    $g_*^{i+1}\ =\ get\_next\_gene\_seed\ (i)$ ; $i\ =\ i + 1$.
end
```

Algorithm 2: Detailed MRB algorithm

```
Procedure: $Compute\_mi\_matrix$
Input: $g_*$.
Output: $M$.
for i = 1 to $N_g$ do
    for j = 1 to $N_c$ do
        $M[i][j] = \frac{1}{N_c} \log\left(\frac{\hat{p}(g_{*j},g_{ij})}{\hat{p}(g_{*j})\cdot\hat{p}(g_{ij})}\right)$.
    end
end
```

Algorithm 3: Compute mutual information matrix

```
Procedure: $Compute\_similarity\_matrix$
Input: $M,\ \alpha$.
Output: $\hat{M}$.
/* $avg()$ denotes the average of the matrix elements over all rows and columns*/
for i = 1 to $N_g$ do
    for j = 1 to $N_c$ do
        if $(M[i][j] < (\alpha \cdot avg(M))$ then
            $\hat{M}[i][j] \ = \ 0$.
        end
        else
            $\hat{M}[i][j] = \hat{M}[i][j]/(\alpha \cdot avg(M)) - 1$.
        end
    end
end
```

Algorithm 4: Compute similarity matrix

```
Procedure: $update\_bicluster\_n\_score$
Input: $G, C, \hat{x}, flag$.
if ($flag$ is $0$) then
    $C = C \setminus \hat{x}$.
    for each $g \in G$ do
        $score\ (g) = score(g) - \hat{M}[g][\hat{x}]$.
    end
end
else
    $G = G \setminus \hat{x}$.
    for each $c \in C$ do
        $score\ (c) = score(c) - \hat{M}[\hat{x}][c]$.
    end
end
```

Algorithm 5: Update bicluster and its score

```
Procedure: $get\_next\_gene\_seed$
Input: $i$.
Output: $next\_g_*$.
$next\_g_*$ = $argmin_{g \in G - \cup_{k \leq i} G_k}\ \{max_{g_* \in S} mi(g,\ g_*)\}$.
return $next\_g_*$.
```

Algorithm 6: Return the next gene seed

Dataset	Size	Time (s)
A. thaliana	619×72 (44,568)	70
HBC	1213×97 (1,17,661)	228
DLBCL	661×180 (1,18,980)	427
S. cerevisiae	2993×173 (5,17,789)	1620

Table 5.1: Runtime of MRB on real Datasets.

Dataset	Biological Process	Cellular Component	Molecular Function
A. thaliana	$1.9\ e^{-23}$	$2.4\ e^{-12}$	$9.7\ e^{-14}$
S. cerevisiae	$4.4\ e^{-31}$	$1.70\ e^{-43}$	$8.4\ e^{-26}$
HBC	$7.2\ e^{-20}$	$3.3\ e^{-15}$	$1.7\ e^{-7}$
DLBCL	$1.3\ e^{-9}$	$2.5\ e^{-11}$	$2.8\ e^{-09}$

Table 5.2: Best p values of MRB Biclusters.

Dataset	E value
A. thaliana	$3.7\ e^{-2}$
S. cerevisiae	$1.3\ e^{-17}$
HBC	$7.1\ e^{-18}$
DLBCL	$3.4\ e^{-16}$

Table 5.3: Best E values of the motifs from MRB Biclusters

Chapter 6

BRC: Extracting Biclusters with Related Conditions

In the previous chapter, we described a method based on gene scores, condition scores and bicluster scores that are defined using a precomputed similarity matrix. The drawback of the approach is that the values in the similarity matrix are precomputed based on the entire set of conditions. Though the scores of the genes (/conditions) are updated as a condition (/gene) is deleted, the values in the similarity matrix are not recomputed. Thus the scores computed do not completely capture the impact of a small subset of conditions. Recomputing the entire matrix after every deletion would be computationally expensive. Thus, in this chapter we present a heuristic approach to obtain biclusters where the mutual information is computed using only the local subset of conditions. Here, we assume that if there is a group of genes which exhibit some relationship in their expression values under a subset of conditions, then these conditions are also related to each other in some sense. This assumption is not unrealistic as expression of genes in the cells of same organ, say brain, are expected to be more related to each other than those in the cells of different organs. Also the conditions to which the genes in the cells of the brain tissue respond are expected to be more related to each other than the conditions to which they don’t respond.

Like MRB, the gene seeds and the number of biclusters to be extracted are determined by the algorithm from the data itself. We tested the algorithm on both synthetic and real datasets. The main idea behind the construction of the synthetic data was to model relationships between genes of the bicluster over a subset of conditions in such a way that the **subset of conditions of the bicluster are also related over the subset of related genes**. Synthetic data was also constructed to study the algorithm's ability to extract biclusters in presence of noise. The synthetic datasets were constructed for both nonoverlapping and overlapping biclusters to model complex biological processes. We were able to extract all the implanted biclusters whereas the existing biclustering algorithms were unable to identify them completely.

We also tested our algorithm on our four datasets viz. *Arabidopsis thaliana*, *Diffuse large B-cell lymphoma*, *Human breast cancer*, and *Saccharomyces cerevisiae*. We were able to extract biologically significant biclusters from all the datasets. We compared the performance of our algorithm with other biclustering algorithms namely ISA [BIB03], CC [CC00], OPSM [BDCKY02] and BIMAX [PBZ$^+$06]. The p values of the GO annotations associated with the genes of our biclusters were found to be smaller than the p values of the GO anotations associated with the biclusters of others. In other words, our biclusters were found to be biologically more significant than those of other biclustering solutions. The motifs extracted from the promoter regions of the genes of our biclusters were statistically more significant (smaller E value) as compared to the motifs extracted from the biclusters of other algorithms.

6.1 The Bicluster Score

Let G be the set of genes and C be the set of conditions in the expression matrix E. The number of genes in G is denoted by N_g and the number of conditions in C by N_c. We want to find biclusters which are tuples denoted as (G', C'), where G' is the subset of genes which are most closely related to a gene seed (g_*) under the subset C' of conditions and C' is the subset of conditions under which subset G' of genes are most closely related to the gene seed.

For a gene seed g_* we define the mutual information score of a bicluster (G', C') as the average mutual information of all the genes in G' with g_* under the conditions C' i.e.

$$MIscore(G', C') = \frac{\sum_{i \in G'} mi_{C'}(g_i, g_*)}{|G'||C'|}$$

where $|G'|$ and $|C'|$ denote the number of genes in G' and conditions in C' respectively; $mi_{C'}(g, g_*)$ denotes the mutual information between g and g_* over the condition set C' according to Equation 4.2 in Chapter 4 and is given as follows:

$$mi_{C'}(g_i, g_*) = \frac{1}{N_{C'}} \sum_{j \in C'} \log \frac{\hat{p}\left(g_{ij},\ g_{*j}\right)}{\hat{p}\left(g_{ij}\right)\hat{p}\left(g_{*j}\right)} \tag{6.1}$$

We extract biclusters with high $MIscores$.

6.2 BRC: Biclustering with related conditions

In this section we present a heuristic algorithm (BRC) to extract **biclusters with related conditions**. The broad overview of the algorithm is given in Algorithm 7.

For a given gene seed BRC proceeds in three steps. In the first step it finds the set of genes which are most related to the seed gene. For this it computes the pairwise mutual information of the gene seed with all other genes over all the conditions. Genes having mutual information greater than the gene threshold t_g are selected.

As mentioned earlier, we assume that if a group of genes exhibit some relationships in their expressions under a subset of conditions, then these conditions are also related to each other. One way then to discover relevant conditions is to find pairwise mutual information amongst all pairs of conditions and select those pairs whose mutual information is above some threshold. The problem in this approach is that if two pairs of conditions

c_1, c_2 and c_3, c_4 have high mutual information then all four will be selected whereas there may not be any relation between c_1 and c_3. Thus in the second step of our algorithm, we use a reference condition say c_* and select conditions which have high degree of relation with c_*. For this the algorithm computes the pairwise mutual information of c_* with all other conditions over the reduced set of genes. Again only those conditions are selected whose pairwise mutual information is greater than the condition threshold t_c.

In the third and the final step, the algorithm selects, from the whole expression data, those genes which are most dependent on each other under the reduced set of conditions identified in step two. For this we recompute mutual information of genes with the gene seed over the reduced set of conditions. Genes not related to the gene seed under all the conditions but related under a subset of conditions are identified in this step.

Running the algorithm for more iterations did not improve the result. Since, which reference condition is best for our gene set is not known, the above process is repeated for a well separated set of reference conditions. Finally we choose the bicluster with the maximum $MIscore$. Thus we get one bicluster for a fixed gene seed. More biclusters are obtained by running the algorithm for more gene seeds. The gene seeds are chosen to be well separated as explained in the previous chapter. Biclusters which contain less than a fixed number of genes (five in our case) are discarded.

If a set of related genes is known earlier we can skip the first step of finding the gene subset and can go to step 2 directly. If no set of related genes is known, we enter into a chicken-egg problem wherein the question arises as to whether one should start grouping the genes first or the conditions first. Feature selection algorithms start with grouping the conditions first whereas most of the clustering solutions start grouping the genes first as that is the most intuitive thing to do. The two approaches have different objectives and each involves its own tradeoff.

Procedure $BRC()$ in Algorithm 8 shows the detailed algorithm. In [GA10], we had used the bin method to estimate the probability densities. However, in this chapter we present our results with kernel density estimator. Procedure $get\_next\_gene\_seed()$ is same as that of MRB.

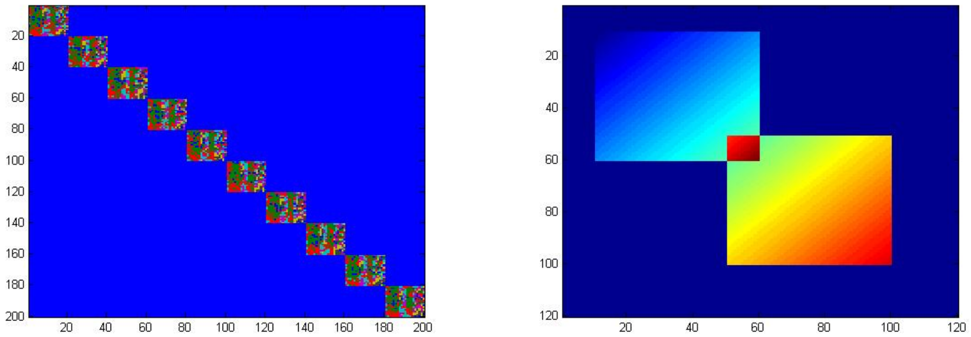
```
BRC: Biclustering with related conditions
Input: $E$, $G$, $C$, $N_g$, $N_C$, $t_g$, $t_c$
Output: A set of biclusters $B_i = (G_i, C_i)$
Mark all genes as $unclustered$
Set $i$ to 1
while there are genes to be clustered do
    get next gene seed $g^i_*$
    $G_{tmp} \leftarrow$ genes that have high MI with $g^i_*$ over $C$
    Mark all conditions as $unclustered$
    Set $j$ to 1
    while there are conditions to be clustered do
        get next condition seed $c^j_*$
        $C_j \leftarrow$ conditions having high MI with $c^j_*$ over genes $g \in G_{tmp}$
        Mark conditions $c \in C_j$ as $clustered$
        /* Compute $G_j$ based on $C_j$ */
        $G_j \leftarrow$ genes having high MI with $g^i_*$ over $C_j$
        increment $j$
    end
    /*select bicluster with maximum score */
    $\hat{i}$ = $argmax_j\{MIscore(G_j, C_j)\}$
    $G_i = G_{\hat{i}}; C_i = C_{\hat{i}}$
    Mark genes $g \in G_i$ as $clustered$
    increment $i$
end
```

Algorithm 7: Overview of BRC algorithm

Time Complexity: The time required to compute the mutual information for all the genes is $O(N_g \cdot N_c^2)$. Selecting genes with high mutual information requires $O(N_g)$ time for each gene seed. Analogously, computing columnwise mutual information requires $O(N_c \cdot N_g^2)$ time and selecting conditions with high mutual information requires $O(N_c)$ time for each condition seed. Recomputing the mutual information for all the genes under the reduced set of conditions takes no more than $O(N_g \cdot N_c^2)$ time. Selecting the final set of genes of the bicluster require $O(N_g)$ time. Thus, computing a bicluster for each condition seed takes $O(N_g \cdot N_c^2 + N_c \cdot N_g^2)$ time. Computing the MIscore of a bicluster takes $O(N_g)$ time. Hence computing the MIscore of k' biclusters (one corresponding to each condition seed) takes $O(k' \cdot N_g)$ time. Selecting the best bicluster from amongst these k' biclusters for a given gene seed takes additional $O(k')$ time. Thus, the time required for a given gene seed, is $O(N_g \cdot N_c^2 + k' \cdot (N_c \cdot N_g^2))) = O(N_g^2 \cdot N_c^2)$. Thus, the overall complexity of BRC for k biclusters is $O(k \cdot N_g^2 \cdot N_c^2)$. Time required to select k gene seeds is same as that in MRB i.e. $O(k \cdot N_g)$. Similarly, the time taken to select k' condition seeds is $O(k' \cdot N_c) = O(N_c^2)$. Table 6.1 gives the actual runtime taken by BRC on real datasets for k and k' both set to 10. The table shows that the algorithm scales well with the size of the data set.

6.3 Experimental Results

We implemented our algorithm in C++. The performance was tested both on synthetic data as well as real datasets. The synthetic datasets were constructed for both nonoverlapping and overlapping biclusters as shown in Figure 6.1. Though the synthetic data for BRC was constructed in a similar way as that in MRB, the main difference lies in modeling the biclusters themselves. Here the biclusters were constructed to capture relationship between the genes over a set of related conditions.



(a) Non overlapping biclusters

(b) Overlapping biclusters

Figure 6.1: Synthetic datasets for BRC algorithm

Synthetic data for nonoverlapping biclusters: Data was generated by implanting 10 biclusters into a larger background matrix(as shown in Figure 6.1(a)). We first generated a larger background matrix of size 200×200 having values drawn from normal distribution (of mean $\mu = .01$ and standard deviation $\sigma = .001$). We then implanted 10 nonoverlapping biclusters of size 20×20 into the background matrix. While implanting, we added the elements of our bicluster with the elements of the background matrix.

Synthetic data for overlapping biclusters: We also created synthetic data with two overlapping biclusters M_1 and M_2 (as shown in Figure 6.1(b)). We again generated a larger background matrix of size 120×120 having values drawn from normal distribution (of mean $\mu = .01$ and standard deviation $\sigma = .001$). We then implanted two overlapping biclusters of size 50×50 into the background matrix. The first bicluster M_1 consisted of genes g_{11} to g_{60} and the conditions c_{11} to c_{60}. The second bicluster M_2 consisted of genes g_{51} to g_{100} and conditions c_{51} to c_{100}.

We were able to extract all the implanted biclusters from both the datasets.

Effect of varying gene threshold t_g: We varied the gene thresholds to study their effect on the output biclusters. We found that only the granularity of the output biclusters changes as we increase or decrease the gene threshold t_g. Figure 6.2 shows the output biclusters for varying t_g for the overlapping biclusters shown in Figure 6.1(b). For the value of the condition threshold t_c set to -0.5, at a very low gene threshold t_g, the biclusters reported had almost all the genes. As we increase t_g to -0.1 we were able to find both M_1 and M_2 separately. As we increase t_g further to 1.5 we were able to find the genes in $M_1 \bigcap M_2$. Finally for a very high gene threshold the output sets were empty. Thus on varying the gene threshold, only the granularity of the resulting biclusters changes providing large biclusters for small value of the threshold and their subsets as the value increases. Similar results were observed on varying the condition threshold and keeping the gene threshold fixed. At t_g set to $-.1$ and t_c set to -0.5 the algorithm was able to

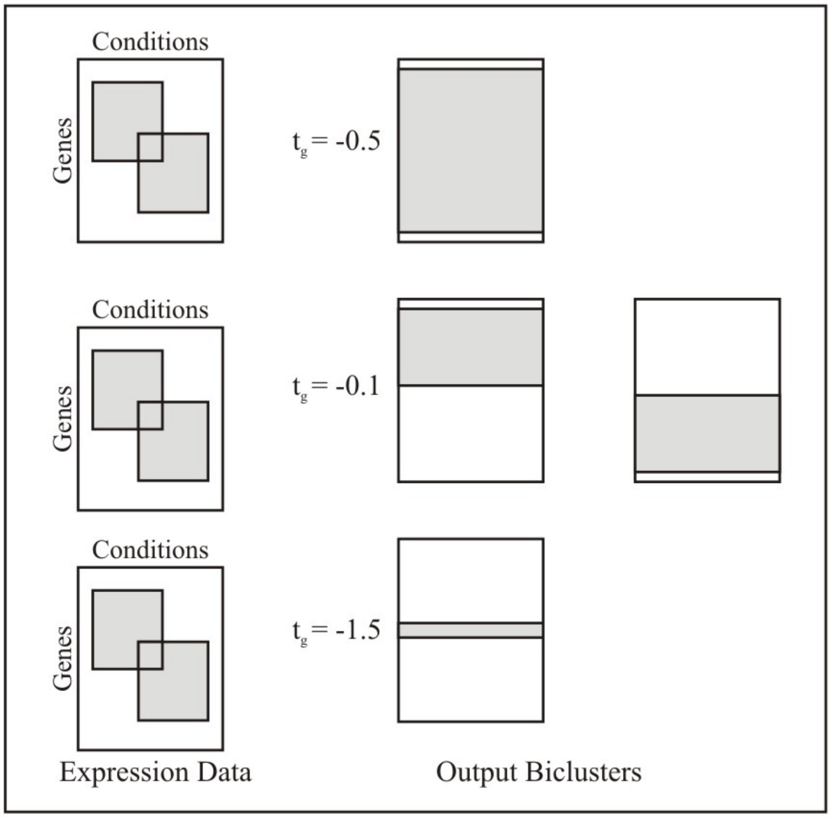


Figure 6.2: Effect of varying gene threshold t_g on output Biclusters of BRC

extract exactly M_1 and M_2. Thus, with different values of the gene threshold and the condition threshold BRC can extract the modular and overlapping structures hidden in the expression data, starting from the entire set of $M_1 \cup M_2$ followed by the components M_1 and M_2 followed by the submodule $M_1 \bigcap M_2$. Similar results were obtained for the data with non-overlapping biclusters of Figure 6.1(a).

Effect of initial start gene: To study the impact of initial gene seed, the experiments were carried out on different gene seeds. We were able to extract all the implanted biclusters from both the synthetic datasets irrespective of the initial gene seed.

Effect of noise: To study the impact of noise on the performance of BRC, synthetic data in Figure 6.1(a) was perturbed by adding noise varying from $.001$ to $.005$. We were able to extract all the ten implanted biclusters from the synthetic data.

Real datasets: We tested our algorithm on our four datasets of *Arabidopsis thaliana*, *Saccharomyces cerevisiae*, *Diffuse large B cell lymphoma* and the *Human Breast Cancer* dataset. The biclusters of *A. thaliana* and *S. cerevisiae* dataset for various algorithms were available on the BICAT toolbox [BBP$^+$06] whereas those of *DLBCL* and *HBC* were obtained by running the algorithms using BICAT toolbox. BIMAX could not be run on *DLBCL* and *HBC* as it depends heavily on the discretization of the data and the discretized data was not available. The value of the input parameters for these algorithms and BRC were chosen such that the output biclusters were of comparable sizes.

Tables 6.2, 6.3, 6.4 and 6.5 summarize the best p values of the GO terms of the biclusters obtained by different algorithms for the *A. thaliana*, *S. cerevisiae*, *DLBCL* and HBC dataset respectively. The histograms in Figures 6.3, 6.4, 6.5 and 6.6 show $-\log(p)$ values. It can be seen that the $-\log(p)$ values of the GO terms associated with the biclusters of BRC are higher in comparison to that of the biclusters of other algorithms on all organisms except Cheng and Church on *cerevisiae* dataset i.e. our algorithm outperforms all other algorithms on all the organisms except for the *S. cerevisiae* dataset where Cheng and Church performs better than ours. The results clearly provide evidence that our biclusters are biologically more significant than most of the biclusters by other algorithms.

As the genes showing dependencies in expression data are expected to have common patterns in their promoter regions as explained in Chapter 3, we studied the promoter regions of the genes belonging to a bicluster for such common patterns. Tables 6.6, 6.7, 6.8 and 6.9 summarize the best E values for the motifs extracted from the gene sequences of the genes belonging to biclusters extracted by BRC and other algorithms. The histograms in Figures 6.7, 6.8, 6.9 and 6.10 show $-\log(E)$ values for all the algorithms. Again we find that the $-\log(E)$ values corresponding to BRC are much higher than those of other algorithms on all organisms except for *S. Cerevisiae*, further endorsing that our biclusters are more biologically significant than most of the biclusters extracted by other algorithms.

For *A. thaliana*, we also checked the existence of the extracted motifs in the existing motif database of the organism using PLACE [HUIK99]. PLACE is a database of motifs found in plant cis-acting regulatory DNA elements. The PLACE database also contains a brief description of each motif. At least one of the motifs from every bicluster (except for one) belonged to the known motif database of *Arabidopsis thaliana* thereby endorsing the good quality of the biclusters. Some of the other motifs were found in other organisms but not in *Arabidopsis thaliana*. Patterns like $CGGCGACGAG$ that did not match in the *Arabidopsis thaliana* motif database but was found in other organisms like rice (Oryza sativa) and Chlamydomonas reinhardtii, cauliflower, maize, soyabean etc. can be targets for further research by biologists.

Effect of noise in real scenario: We extracted a bicluster of size 138×4 found by BRC from the expression data of *A. thaliana* and implanted it in a matrix of size 150×150 containing random numbers generated from normal distribution with mean $\mu = .01$ and variance $\sigma = .001$ as shown in Figure 6.11. To add noise, it was perturbed by adding random numbers generated from normal distribution. BRC was able to extract the bicluster from the noisy data.

```
**Procedure:** $BRC$
**Input**: $E,\ G,\ C,\ N_g,\ N_c,\ t_g,\ t_c$
**Output**: A set of Biclusters $B_i = (G_i,\ C_i)$
$i = 1$ ; $S_g = \phi$ /* $S_g$ is the set of gene seeds */
$g_*^i$ = $random\ ()$
Mark all genes $g \in G$ as $unclustered$
**while** *there are genes to be clustered* **do**
    $S_g = S_g \cup g_*^i$
    $G_{tmp}$ = $Compute\_genes\ (g_*^i, G, C, t_g, N_g, N_c)$
    $S_c = \phi$ /* $S_c$ is the set of condition seeds */
    $j$ = 1 ; $c_*^j$ = $random\ ()$
    Mark all conditions $c \in C$ as $unclustered$
    **while** *there are conditions to be clustered* **do**
        $S_c$ = $S_c \cup c_*^j$
        $C_j$ = $Compute\_conditions\ (c_*^j, G_{tmp}, C, t_c, N_{gtmp}, N_c)$
        /* $N_{gtmp}$ denotes the number of genes in $G_{tmp}$ */
        $G_j$ = $Compute\_genes\ (g_*^i, G, C_j, t_g, N_g, N_{cj})$
        /* $N_{cj}$ denotes the number of conditions in $C_j$ */
        Mark all the conditions $c \in C_j$ as $clustered$
        $c_*^{j+1}$ = $get\_next\_cond\_seed$ (j) ; $j = j + 1$
    **end**
    $\hat{i}$= $argmax_j\{MIscore\ (G_j, C_j)\}$
    $G_i = G_{\hat{i}}; C_i = C_{\hat{i}}$; $B_i$ = ($G_i, C_i$)
    mark all the genes $g \in G_i$ as $clustered$
    $g_*^{i+1}$ = $get\_next\_gene\_seed\ (i)$; $i = i + 1$
**end**
```

Algorithm 8: Detailed BRC algorithm

```
Procedure: $Compute\_genes$
Input: $g_*, G, C, t_g, N_g, N_c$
Output: $G'$
for i = 1 to $N_g$ do
    $mig\ [i]$ = $\frac{1}{N_{C'}} \sum_{j \in C'} \log \frac{\hat{p}\ (g_{ij},\ g_{*j})}{\hat{p}\ (g_{ij})\ \hat{p}\ (g_{*j})}$
end
$\mu = \Sigma_i\ mig[i]/N_g$
for i = 1 to $N_g$ do
    $\sigma^2 = \Sigma_i\ (mig[i] - \mu)^2/N_g$
end
$G' = \{g_i \in G : \frac{(mig[i]-\mu)}{\sigma} > t_g\}$
return $G'$
```

Algorithm 9: Compute the relevant gene set

```
Procedure: $Compute\_conditions$
Input: $c_*, G_0, C, t_c, N_{g0}, N_c$
Output: $C'$
for $j = 1$ to $N_c$ do
    $mic[j]$= $Compute\_mi\ (c_j, c_*, N_{g0})$
end
$\mu = \Sigma_j\ mic[j]/N_c$
for $j = 1$ to $N_c$ do
    $\sigma^2 = \Sigma_j(mic[j] - \mu)^2/N_c$
end
$C' = \{c_{j \in C} : \frac{(mic[j]-\mu)}{\sigma} > t_c\}$
return $C'$
```

Algorithm 10: Compute the relevant condition set

```
Procedure: $get\_next\_cond\_seed$
Input: $j$
Output: $next\_c_*$
$next\_c_* = argmin_{c \in C - \cup_{k \leq j} C_k}\ \{max_{c_* \in S_c} mi(c,\ c_*)\}$
return $next\_c_*$
```

Algorithm 11: Return the next condition seed

Dataset	Size	Time (s)
A. thaliana	619×72 (44,568)	480
HBC	1213×97 (1,17,661)	1200
DLBCL	661×180 (1,18,980)	643
S. cerevisiae	2993×173 (5,17,789)	9900

Table 6.1: Runtime of BRC on real Datasets.

	p values		
Method	Biological process	Cellular Component	Molecular Function
BRC	$5.7\ e^{-37}$	$3.2\ e^{-14}$	$2.0\ e^{-18}$
BIMAX	$5.1\ e^{-10}$	$2.8\ e^{-8}$	$9.1\ e^{-6}$
CC	$1.3\ e^{-25}$	$1.2\ e^{-11}$	$7.1\ e^{-11}$
ISA	$3.7\ e^{-29}$	$6.0\ e^{-14}$	$4.2\ e^{-14}$
OPSM	$9.6\ e^{-31}$	$7.7\ e^{-12}$	$5.2\ e^{-14}$

Table 6.2: Best p values of Biclusters of BRC and other algorithms on *A.thaliana* dataset

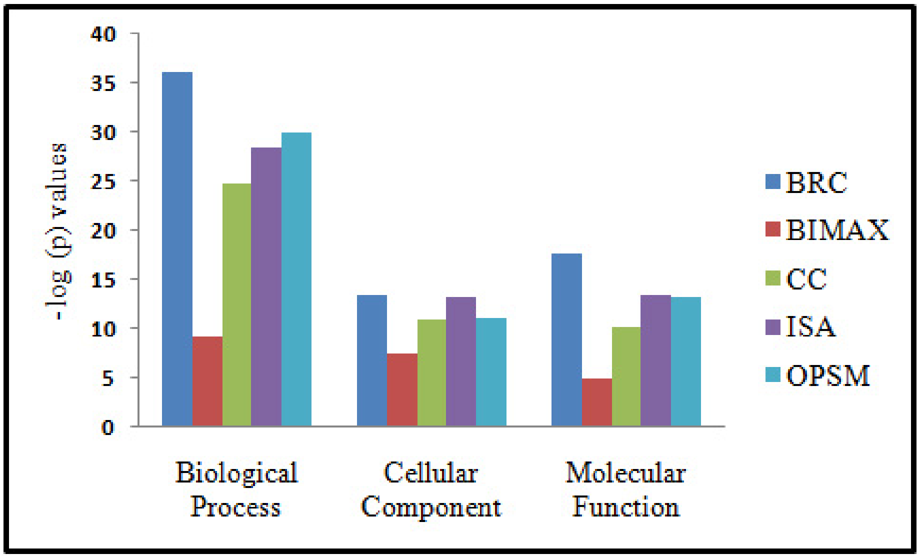


Figure 6.3: Best $-\log(p)$ values of Biclusters of BRC and other algorithms on *A. thaliana* dataset

	p values		
Method	Biological process	Cellular Component	Molecular Function
BRC	$3.1\ e^{-30}$	$1.9\ e^{-35}$	$4.9\ e^{-13}$
BIMAX	$3.5\ e^{-4}$	$9.9\ e^{-4}$	$7.6\ e^{-3}$
CC	$2.8\ e^{-26}$	$1.5\ e^{-50}$	$3.2\ e^{-36}$
ISA	$2.3\ e^{-4}$	$2.9\ e^{-3}$	$4.0\ e^{-3}$
OPSM	$1.3\ e^{-3}$	$1.4\ e^{-7}$	$8.3\ e^{-6}$

Table 6.3: Best p values of Biclusters of BRC and other algorithms on *S. cerevisiae* dataset

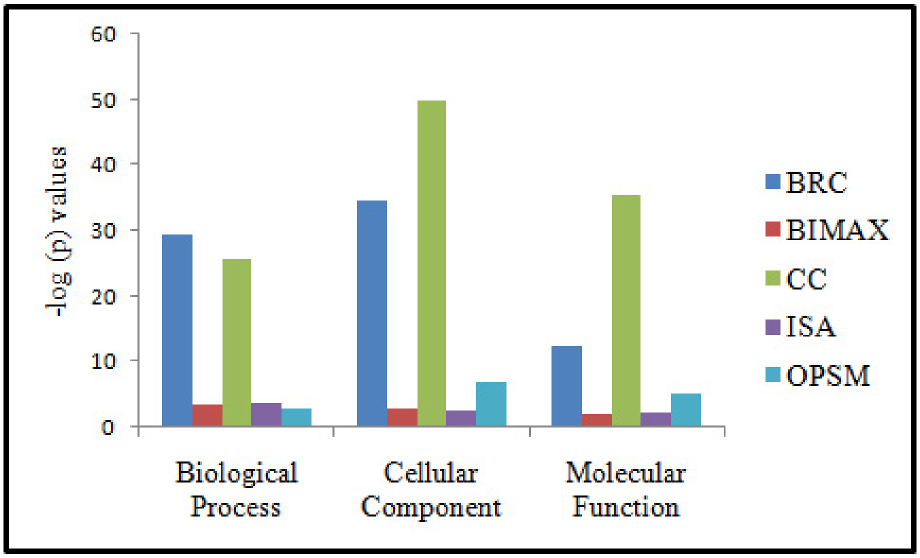


Figure 6.4: Best $-\log(p)$ values of Biclusters of BRC and other algorithms on *S. cerevisiae* dataset

	p values		
Method	Biological process	Cellular Component	Molecular Function
BRC	$1.1\ e^{-24}$	$8.5\ e^{-18}$	$2.8\ e^{-13}$
CC	$3.9\ e^{-6}$	$9.0\ e^{-5}$	$1.5\ e^{-3}$
ISA	$1.3\ e^{-11}$	$7.0\ e^{-6}$	$3.1\ e^{-5}$
OPSM	$5.7\ e^{-7}$	$5.3\ e^{-6}$	$2.3\ e^{-3}$

Table 6.4: Best p values of Biclusters of BRC and other algorithms on *DLBCL* dataset

	p values		
Method	Biological process	Cellular Component	Molecular Function
BRC	$9.2\ e^{-28}$	$8.00\ e^{-20}$	$2.5\ e^{-9}$
ISA	$5.9\ e^{-11}$	$7.0\ e^{-17}$	$1.1\ e^{-10}$
OPSM	$5.2\ e^{-22}$	$5.0\ e^{-8}$	$1.5\ e^{-4}$
CC	$1.0\ e^{-6}$	$1.6\ e^{-6}$	$6.9\ e^{-5}$

Table 6.5: Best p values of Biclusters of BRC and other algorithms on *HBC* dataset

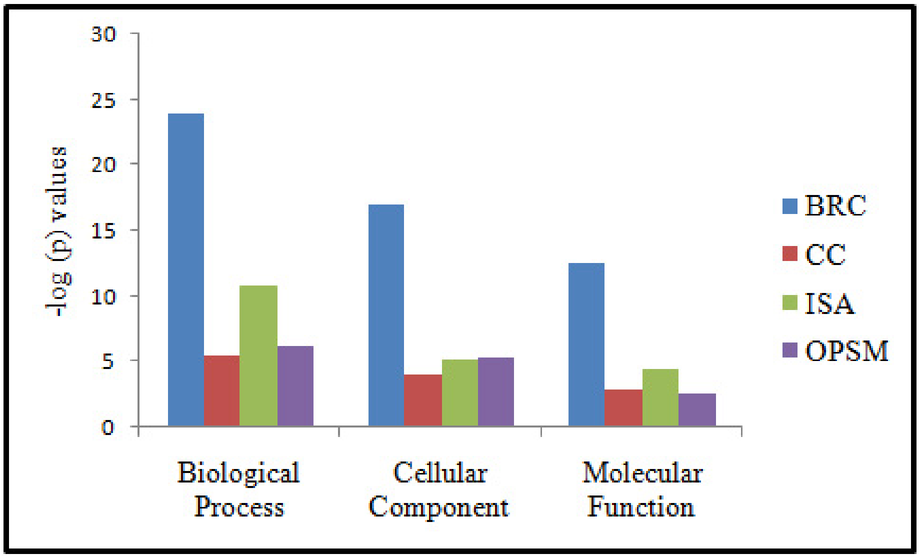


Figure 6.5: Best $-\log(p)$ values of Biclusters of BRC and other algorithms on *DLBCL* dataset

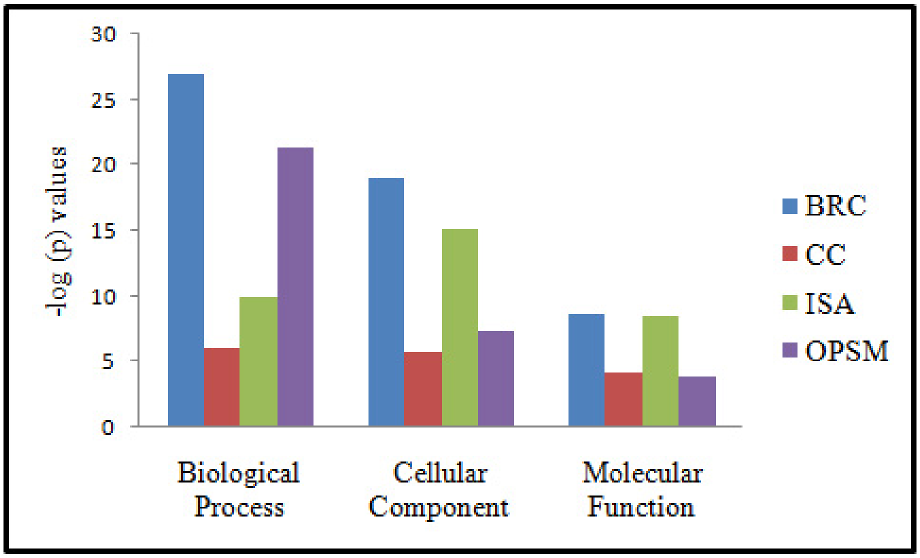


Figure 6.6: Best $-\log(p)$ values of Biclusters of BRC and other algorithms on *HBC* dataset

BRC	BIMAX	CC	ISA	OPSM
$5.9\ e^{-10}$	$6.1\ e^{-3}$	$3.8\ e^{-3}$	$1.8\ e^{-1}$	$3.1\ e^{-1}$

Table 6.6: Best E values of motifs from Biclusters of BRC and other algorithms on *A. thaliana* dataset

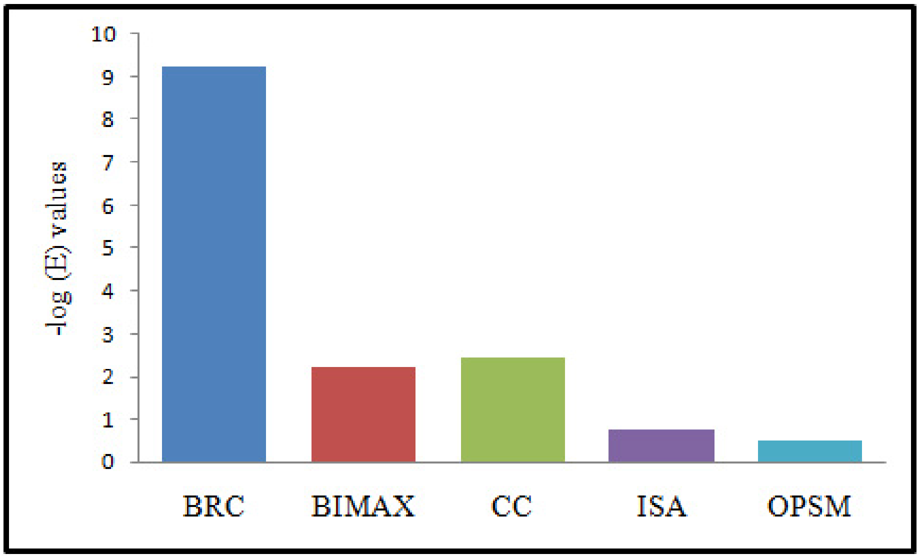


Figure 6.7: Best $-\log(E)$ values of Biclusters of BRC and other algorithms on *A.thaliana* dataset

BRC	BIMAX	CC	ISA	OPSM
$6.1\ e^{-9}$	$1.8\ e^{-3}$	$4.4\ e^{-12}$	$4.1\ e^{-2}$	$9.9\ e^{-2}$

Table 6.7: Best E values of motifs from Biclusters of BRC and other algorithms on *S. cerevisiae* dataset

BRC	CC	ISA	OPSM
$1.0\ e^{-111}$	$2.6\ e^{-10}$	$6.7\ e^{-57}$	$9.7\ e^{-12}$

Table 6.8: Best E values of motifs from Biclusters of BRC and other algorithms on *DLBCL* dataset

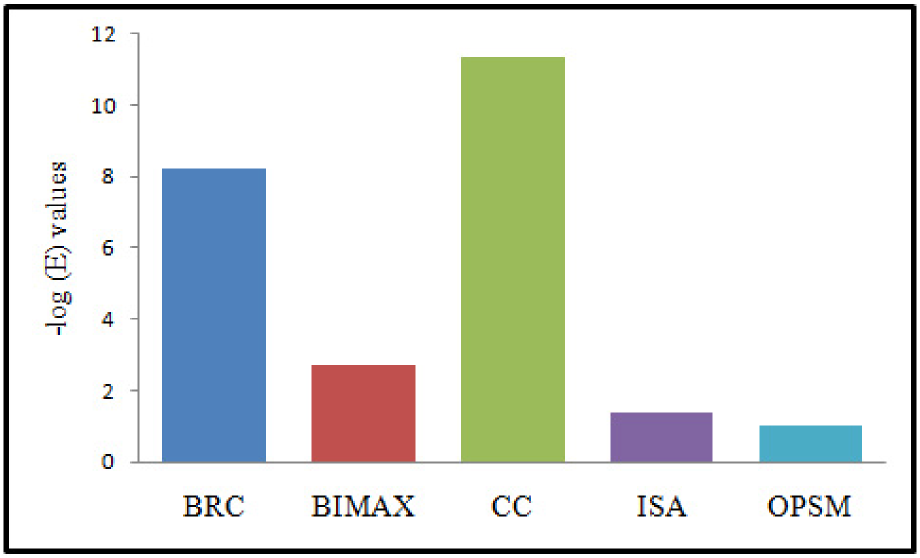


Figure 6.8: Best $-\log(E)$ values of Biclusters of BRC and other algorithms on *S. cerevisiae* dataset

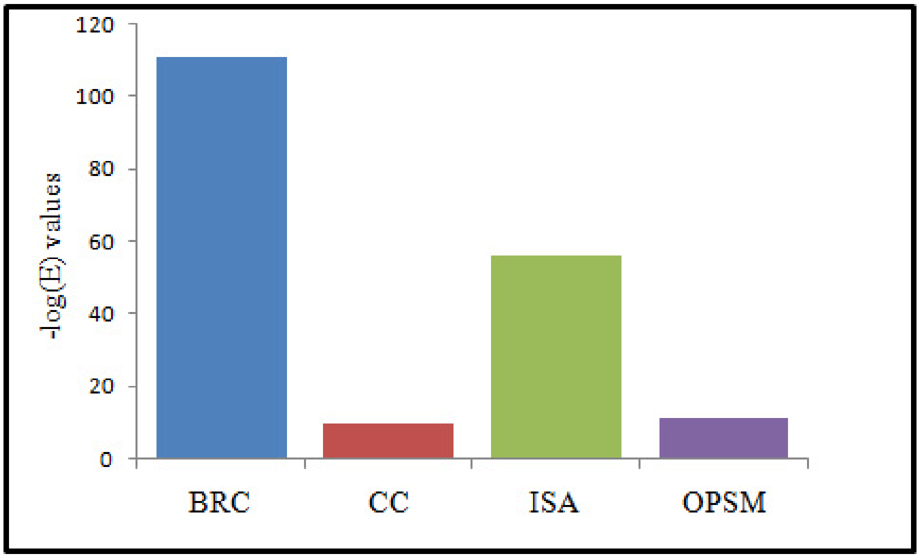


Figure 6.9: Best $-\log(E)$ values of Biclusters of BRC and other algorithms on *DLBCL* dataset

BRC	ISA	OPSM	CC
$5.0\ e^{-43}$	$2.8\ e^{-11}$	$1.2\ e^{-4}$	$2.0\ e^{-29}$

Table 6.9: Best E values of motifs from Biclusters of BRC and other algorithms on *HBC* dataset

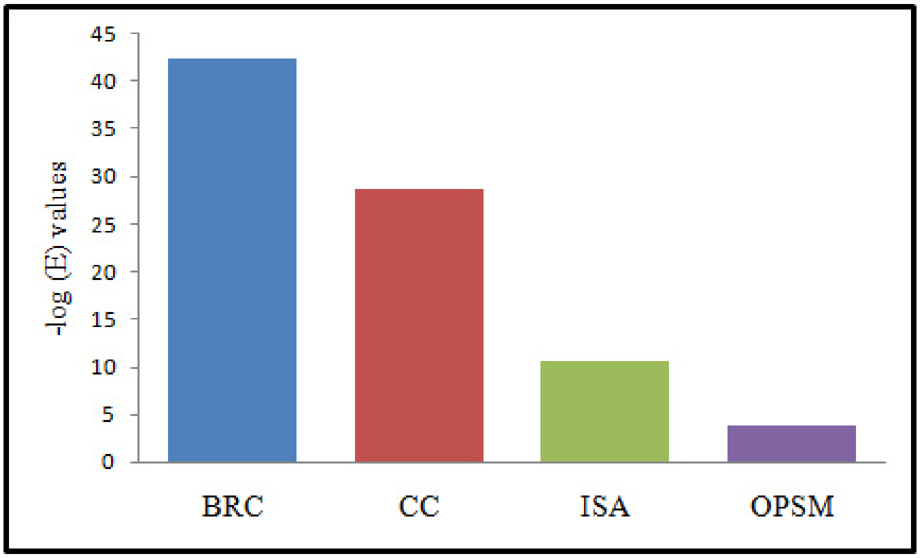


Figure 6.10: Best $-\log(E)$ values of Biclusters of BRC and other algorithms on *HBC* dataset

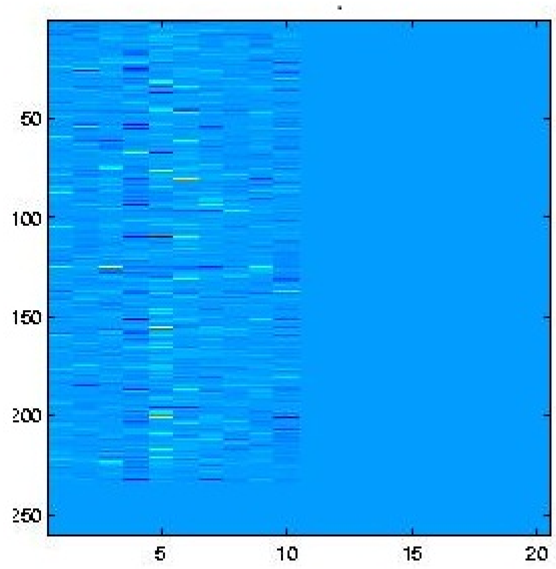


Figure 6.11: Bicluster extracted from *A.thaliana* dataset, perturbed with noise, by BRC

Chapter 7

GenBiClus: Extracting General Biclusters

Our algorithm BRC is based on the assumption that when a group of genes are related to each other under a subset of conditions, there exists some sort of relationship amongst the conditions as well. We now drop this assumption. In this chapter we describe a method [GA08] to extract biclusters from gene expression data where the conditions may or may not be interdependent. Thus the extracted biclusters are more general than the ones extracted by BRC. Instead of computing the mutual information between the conditions, we compute the contribution of each condition to the mutual information between the genes and the gene seed. This is similar to MRB. However, here we present a heuristic approach wherein we compute the mutual information between the genes over a reduced set of conditions instead of the entire set. Those conditions whose contribution is more than that of other conditions are selected. Since we no longer compute the mutual information between the conditions, the algorithm provides an improvement over BRC in terms of computation time as well. As in MRB and BRC, the gene seeds and the number of biclusters to be extracted are determined by the algorithm from the data itself. We tested the algorithm on both synthetic and real datasets. In MRB, the main idea behind the construction of the synthetic data was to model nonlinear relationships between the genes of a bicluster over its conditions. However, in BRC the conditions of the biclusters of the synthetic data were also related to each other. We tested GenBiClus on both these datasets. We were able to extract all the implanted biclusters from all the synthetic datasets of both MRB and BRC. In other words, GenBiClus is able to extract more general biclusters.

We also tested our algorithm on our four real datasets viz. *Arabidopsis thaliana*, *Diffuse large B-cell lymphoma* dataset, *Human Breast Cancer* and *Saccharomyces cerevisiae*. We were able to extract biologically significant biclusters from all the datasets. As before, we compared the performance of our algorithm with other biclustering algorithms. The p values of the GO annotations associated with the genes of our biclusters were found to be smaller than the p values of the GO anotations associated with the biclusters of others. In other words, our biclusters were found to be biologically more significant than those of other biclustering algorithms. Promoter regions of the genes of most of GenBiClus biclusters were found to have statistically more significant common motif patterns as compared to the motifs extracted from the biclusters of other algorithms.

7.1 GenBiClus: Biclustering with general conditions

In this section we describe our algorithm $GenBiClus$. The broad oveview of the algorithm is given in Algorithm 12. Like BRC, GenBiclus proceeds in three steps for a given gene seed. The first step of the two algorithms is same i.e. in the first step we find the set of genes which are most related to the gene seed. It computes the pairwise mutual information of the gene seed with all other genes over all the conditions. Genes having mutual information greater than the gene threshold t_g are selected.

In the second step, the algorithm identifies the experimental conditions under which the set of genes found in the first step show maximum relatedness. This is done by computing the contribution of each condition to the sum of pair wise mutual information between the genes in the reduced set and the gene seed. Again only those conditions whose contribution is greater than the condition threshold t_c are selected. This is the main step where GenBiClus differs from BRC. The third and the final step which is same as that in BRC is a bicluster refinement step where the algorithm selects from the entire expression data those genes which are most related to the gene seed under the subset of conditions identified in step two. For this we recompute mutual information of genes with the gene seed over the reduced set of conditions. Genes not related to the gene seed under all the conditions but related under a subset of conditions will be identified in this step. Contrast this with MRB; in MRB also we subtract the contribution of the j^{th} condition when we delete it from the bicluster but the $\hat{m}_{ij}$ entry itself contains the impact of presence of all the conditions in the data set. Recall the $\hat{m}_{ij}$ entry is computed by using kernel density estimation method for estimating $\hat{p}$ which contains the impact of other conditions as well. In GenBiClus, by recomputing the mutual information, we delete the impact of other conditions on $\hat{p}$ itself.

We tried to refine the biclusters by running the algorithm for few more iterations. However, no improvement was observed. More biclusters are obtained by taking more gene seeds. The gene seeds are chosen to be well separated as explained in the previous chapters. Biclusters which contain less than a fixed number of genes (five in our case) are discarded.

Procedure $GenBiclus()$ in Algorithm 13 shows the the detailed algorithm. Procedure $compute\_genes(),\ compute\_conditions()\ and\ get\_next\_gene\_seed()$ are same as that in BRC. However, column wise mutual information is computed using the contribution of each condition to the mutual information between the gene seed and the reduced set of genes.

Time Complexity: The time required to compute the mutual information for all the genes is same as that in MRB i.e. $O(N_g \cdot N_c^2)$ time. Selecting, genes with high mutual

```
GenBiClus: Extracting General Biclusters
Input: $E$, $G$, $C$, $N_g$, $N_C$, $t_g$, $t_c$
Output: A set of biclusters $B_i = (G_i, C_i)$
Mark all genes as $clustered$
Set $i$ to $1$
while there are genes to be $clustered$ do
    get next gene seed $g_*^i$
    $G_{tmp} \leftarrow$ genes having high MI with $g_*^i$
    $C_i \leftarrow$ conditions that contribute most to the MI between the genes $g \in G_{tmp}$
    and $g_*^i$
    $G_i \leftarrow$ genes that have high MI with $g_*$ over $C_i$
    Mark genes $g \in G_i$ as $clustered$
    output ($G_i$,$C_i$)
end
```

Algorithm 12: Overview of GenBiClus algorithm

information requires $O(N_g)$ time. Analogously, selecting conditions with high mutual information requires $O(N_c)$ time. Recomputing the mutual information and selecting the genes with high mutual information over the reduced set of conditions again, takes $O(N_g \cdot N_c^2)$ time. This is done for each gene seed. Thus, the overall time required by GenBiClus for k gene seeds is $O(k \cdot (N_g \cdot N_c^2))$. Time required to select k gene seeds is same as that in MRB i.e. $O(k{\cdot}N_g)$. Table 7.1 gives the actual runtime taken by GenBiClus on real datasets for k set to 10. The table shows that the algorithm scales well with the size of the data set.

```
Procedure: $GenBiClus$
Input: $E,\ G,\ C,\ N_g,\ N_c,\ t_g,\ t_c$
Output: A set of Biclusters $B_i\ =\ (G_i,\ C_i)$
$i = 1$; $S\ =\ \phi$
$g_*^i\ =\ random\ ()$
Mark all genes $g \in G$ as $clustered$
while there are genes to be clustered do
    $G_{tmp}$ = $Compute\_genes$ ($g_*^i$, $G$, $C$, $t_g,\ N_g,\ N_c$)
    $C_i$ = $Compute\_conditions$ ($G_{tmp}$, $C$, $t_c,\ N_{g_{tmp}},\ N_c$)
    /* $N_{g_{tmp}}$ is the number of genes in $G_{tmp}$*/
    $G_i$ = $Compute\_genes$ ($g_*^i$, $G$, $C_i$, $t_g,\ N_g,\ N_{c_i}$)
    /* $N_{c_i}$ is the number of conditions in $C_i$*/
    Output ($G_i,\ C_i$)
    mark all the genes $g \in G_i$ as $clustered$
    $g_*^{i+1}$ = $get\_next\_gene\_seed(i)$
    $i = i + 1$
end
```

Algorithm 13: Detailed GenBiClus algorithm

Dataset	Size	Time (s)
A. thaliana	619×72 (44,568)	61
HBC	1213×97 (1,17,661)	198
DLBCL	661×180 (1,18,980)	362
S. cerevisiae	2993×173 (5,17,789)	1440

Table 7.1: Runtime of GenBiClus on real Datasets.

7.2 Experimental Results

We implemented our algorithm in C++. The performance was tested on both synthetic data as well as real datasets. The synthetic data sets generated for both MRB and BRC were used. We were able to extract all the implanted biclusters from all the datasets thus endorsing our claim that GenBiClus extracts more general biclusters.

Effect of thresholds: The dataset for the overlapping biclusters shown in Figure 5.1(b) was used to study the effect of threshold on the output biclusters. As in case of BRC, we found that only the granularity of the output biclusters changes as we increase or decrease the gene threshold t_g for a fixed condition threshold as shown in Figure 7.1. For condition threshold t_c set to -0.5, at a very low gene threshold $t_g = 0.1$, the biclusters reported had almost all the genes i.e. $M_1 \cup M_2$ (genes g_{11} to g_{100}). On increasing t_g to 0.3 we obtained genes belonging to both the biclusters separately i.e. we obtained M_1 and M_2. At a still higher gene threshold, $t_g = 1$, we were able to find the genes in $M_1 \bigcap M_2$. Finally for a very high gene threshold the output sets were empty.

Thus on varying the gene threshold, only the granularity of the resulting biclusters changes providing large biclusters for small value of the threshold and their subsets as the value increases. Similar results were observed on varying the condition threshold and keeping the gene threshold fixed.

Effect of initial start gene: The effect of random selection of the initial gene seed on the output biclusters was studied by running the algorithm for different gene seeds. We were able to extract all the implanted biclusters from both the synthetic datasets irrespective of the initial gene seed.

Effect of noise: The behaviour of GenBiClus in presence of noise was studied by perturbing the synthetic data for the overlapping biclusters shown in Figure 5.1(a) by adding noise from 0.001 to 0.009. We were able to extract all the ten implanted biclusters.

Real datasets: We tested our algorithm on the four datasets of *Arabidopsis thaliana*, *Saccharomyces cerevisiae*, *Diffuse large B cell lymphoma* and the *Human breast cancer*. Again, we used the biclusters available for the *Arabidopsis thaliana* and *Saccharomyces cerevisiae* for various algorithms from the BICAT toolbox. For the other two datasets viz. DLBCL and HBC, biclusters were obtained by running the algorithms available in the

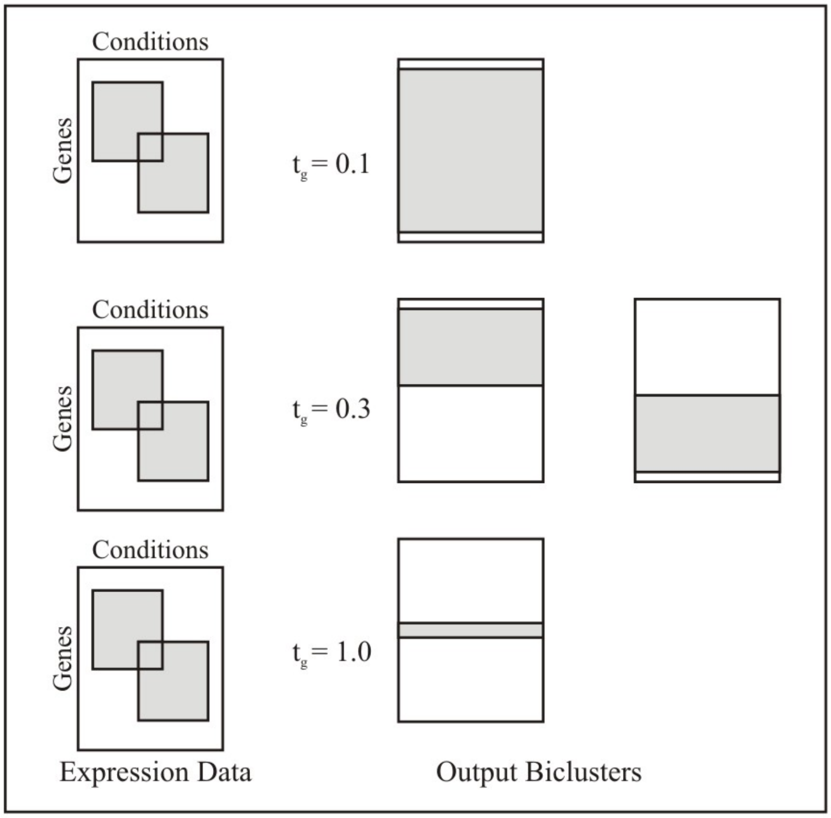


Figure 7.1: Effect of varying gene threshold t_g on output Biclusters of GenBiClus

BICAT tool box. BIMAX could not be run on DLBCL and HBC as it depends heavily on the discretization of the data and the discretized data was not available. The value of the input parameters for the algorithms were chosen such that the output biclusters were of comparable sizes.

Tables 7.2, 7.3, 7.4 and 7.5 summarize the best p values of the GO terms of the biclusters obtained by various algorithms on *A. thaliana*, *S. cerevisiae*, *DLBCL* and *HBC* respectively. The histograms in Figures 7.2, 7.3, 7.4 and 7.5 show $-\log(p)$ values. It can be seen that the $-\log(p)$ values of the GO terms associated with the biclusters of GenBiClus are higher in comparison to that of the biclusters of other algorithms on all organisms except Cheng and Church on *S. Cerevisiae* dataset i.e. our algorithm outperforms all other algorithms on all the organisms except for the *S. Cerevisiae* dataset where Cheng and Church performs better than ours. The results clearly provide evidence that our biclusters are biologically more significant than most of the biclusters by other algorithms.

As the genes showing dependencies in expression data are expected to have common patterns in their promoter regions as explained in Chapter 3, we studied the promoter regions of the genes belonging to a bicluster for such common patterns. Tables 7.2, 7.3, 7.4 and 7.5 also summarize the best (least) E values for the motifs extracted from the gene sequences of the genes belonging to biclusters extracted by GenBiClus and other algorithms. The histograms in Figures 7.6, 7.7, 7.8 and 7.9 show $-\log(E)$ values for all the algorithms. Again we find that the $-\log(E)$ values corresponding to the biclusters extracted by GenBiClus are much higher than most of the biclusters by other algorithms, further endorsing that our biclusters are biologically more significant than most of the biclusters by other algorithms.

Effect of noise in real scenario: We extracted a bicluster of size 197×13 found by GenBiClus from the expression data of *A. thaliana* and implanted it in a matrix of size 250×50 containing random numbers generated from normal distribution with mean $\mu = .01$ and variance $\sigma = .001$ as shown in Figure 7.10. To add noise, it was perturbed by adding random numbers generated from normal distribution. GenBiClus was able to extract the bicluster from the noisy data. GenBiClus was also able to extract the bicluster from the synthetic data created for BRC to study the effect of noise.

	p values		
Method	Biological process	Cellular Component	Molecular Function
GenBiClus	$2.0\ e^{-40}$	$6.1\ e^{-18}$	$9.8\ e^{-15}$
BIMAX	$5.1\ e^{-10}$	$2.8\ e^{-8}$	$9.1\ e^{-6}$
CC	$1.3\ e^{-25}$	$1.2\ e^{-11}$	$7.1\ e^{-11}$
ISA	$3.7\ e^{-29}$	$6.0\ e^{-14}$	$4.2\ e^{-14}$
OPSM	$9.6\ e^{-31}$	$7.7\ e^{-12}$	$5.2\ e^{-14}$

Table 7.2: Best p values of Biclusters of GenBiClus and other algorithms on *A. thaliana* dataset

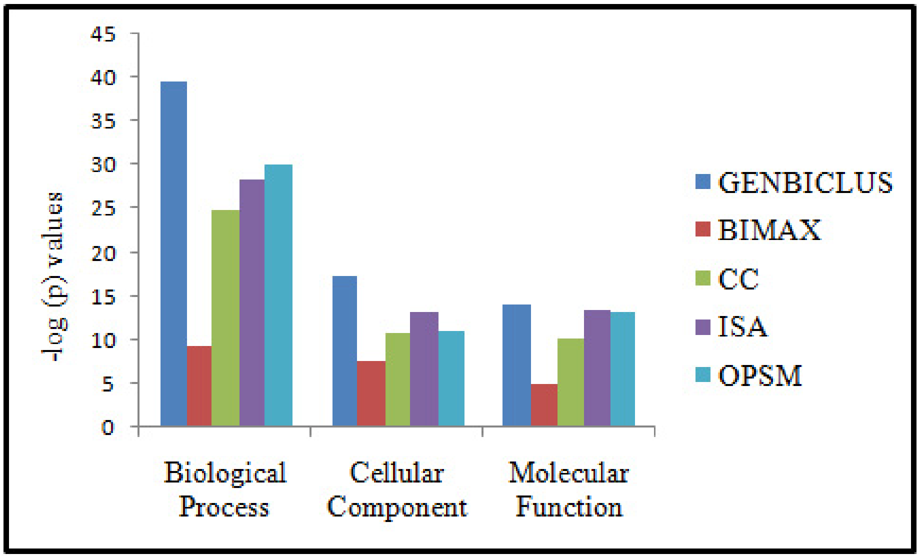


Figure 7.2: Best $-\log(p)$ values of Biclusters of GenBiClus and other algorithms on *A. thaliana* dataset

	p values		
Method	Biological process	Cellular Component	Molecular Function
GenBiClus	$1.0\ e^{-32}$	$4.3e^{-39}$	$3.4e^{-21}$
BIMAX	$3.5\ e^{-4}$	$9.9\ e^{-4}$	$7.6\ e^{-3}$
CC	$2.8\ e^{-26}$	$1.5\ e^{-50}$	$3.2\ e^{-36}$
ISA	$2.3\ e^{-4}$	$2.9\ e^{-3}$	$4.0\ e^{-3}$
OPSM	$1.3\ e^{-3}$	$1.4\ e^{-7}$	$8.3\ e^{-6}$

Table 7.3: Best p values of Biclusters of GenBiClus and other algorithms on *S. cerevisiae* dataset

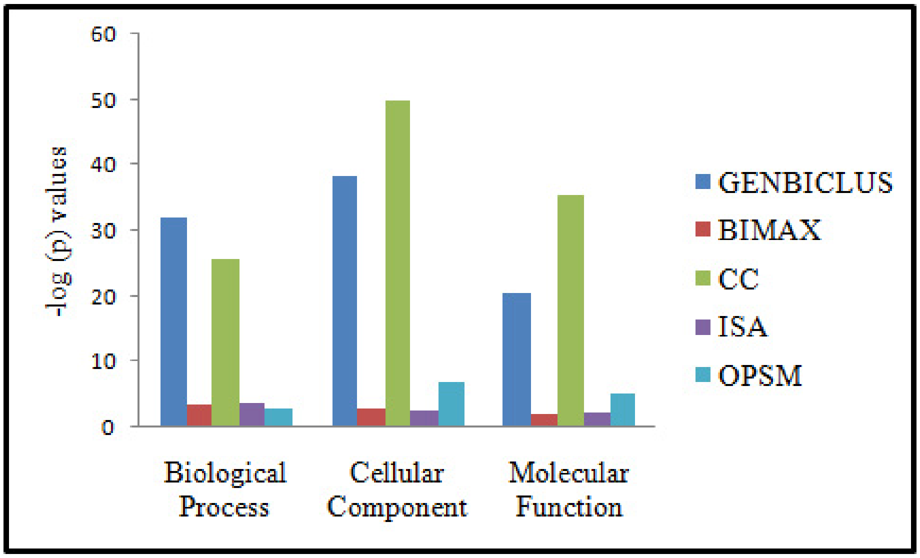


Figure 7.3: Best $-\log(p)$ values of Biclusters of GenBiClus and other algorithms on *S. cerevisiae* dataset

	p values		
Method	Biological process	Cellular Component	Molecular Function
GenBiClus	$2.3e^{-23}$	$3.1e^{-17}$	$3.2e^{-12}$
CC	$3.9e^{-6}$	$9.0e^{-5}$	$1.5e^{-3}$
ISA	$1.3e^{-11}$	$7.0e^{-6}$	$3.1e^{-5}$
OPSM	$5.7e^{-7}$	$5.3e^{-6}$	$2.3e^{-3}$

Table 7.4: Best p values of Biclusters of GenBiClus and other algorithms on *DLBCL* dataset

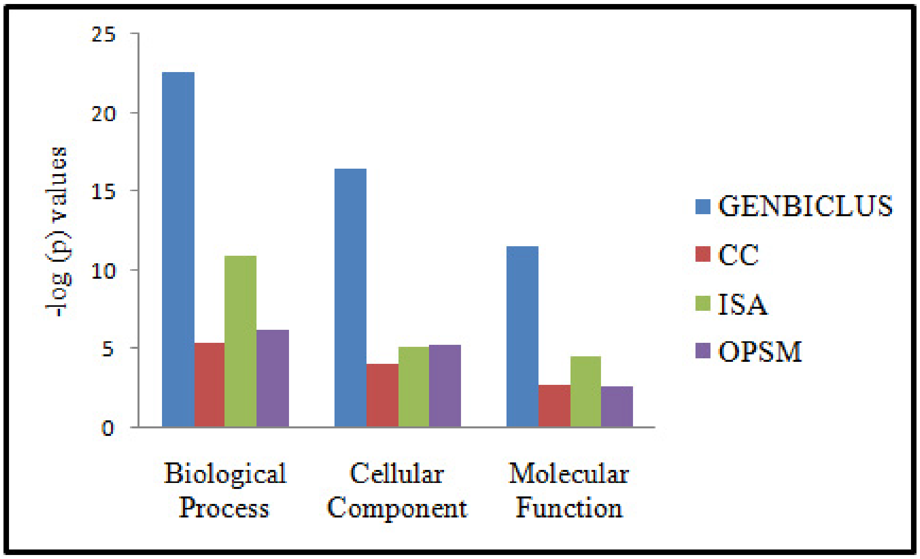


Figure 7.4: Best $-\log(p)$ values of Biclusters of GenBiClus and other algorithms on *DLBCL* dataset

	p values		
Method	Biological process	Cellular Component	Molecular Function
GenBiClus	$3.8e^{-21}$	$5.7e^{-16}$	$3.2e^{-12}$
ISA	$5.9e^{-11}$	$7.0e^{-17}$	$1.1e^{-10}$
OPSM	$5.2e^{-22}$	$5.0e^{-8}$	$1.5e^{-4}$
CC	$1.0e^{-6}$	$1.6e^{-6}$	$6.9e^{-5}$

Table 7.5: Best p values of Biclusters of GenBiClus and other algorithms on *HBC* dataset

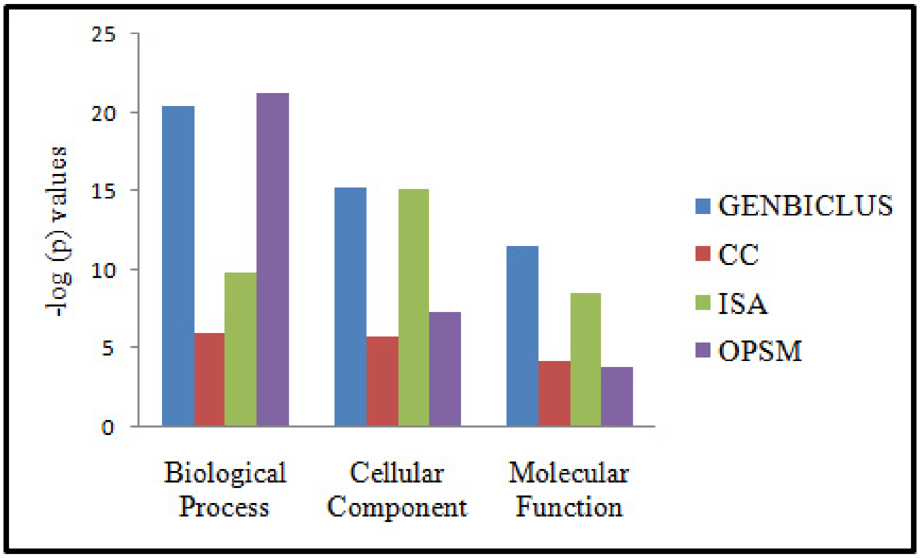


Figure 7.5: Best $-\log(p)$ values of Biclusters of GenBiClus and other algorithms on *HBC* dataset

GenBiClus	BIMAX	CC	ISA	OPSM
$7.8\ e^{-4}$	$6.1\ e^{-3}$	$3.8\ e^{-3}$	$1.8\ e^{-1}$	$3.1\ e^{-1}$

Table 7.6: Best E values of motifs from Biclusters of GenBiClus and other algorithms on *A. thaliana* dataset

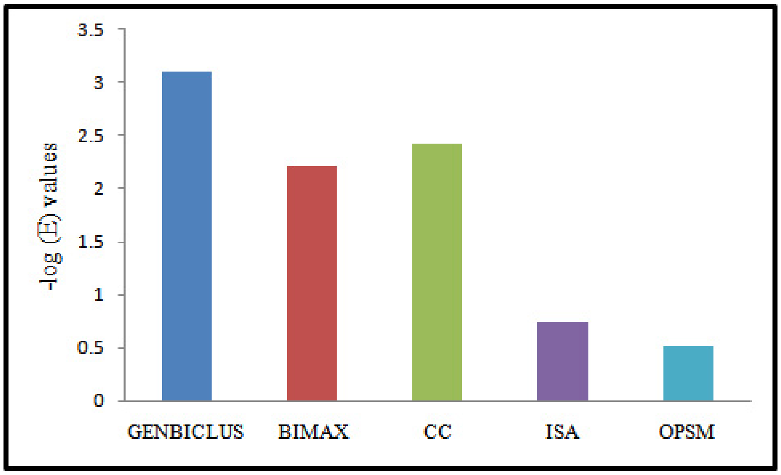


Figure 7.6: Best $-\log(E)$ values of Biclusters of GenBiClus and other algorithms on *A. thaliana* dataset

GenBiClus	BIMAX	CC	ISA	OPSM
$7.9e^{-8}$	$1.8\ e^{-3}$	$4.4\ e^{-12}$	$4.1\ e^{-2}$	$9.9\ e^{-2}$

Table 7.7: Best E values of motifs from Biclusters of GenBiClus and other algorithms on *S. cerevisiae* dataset

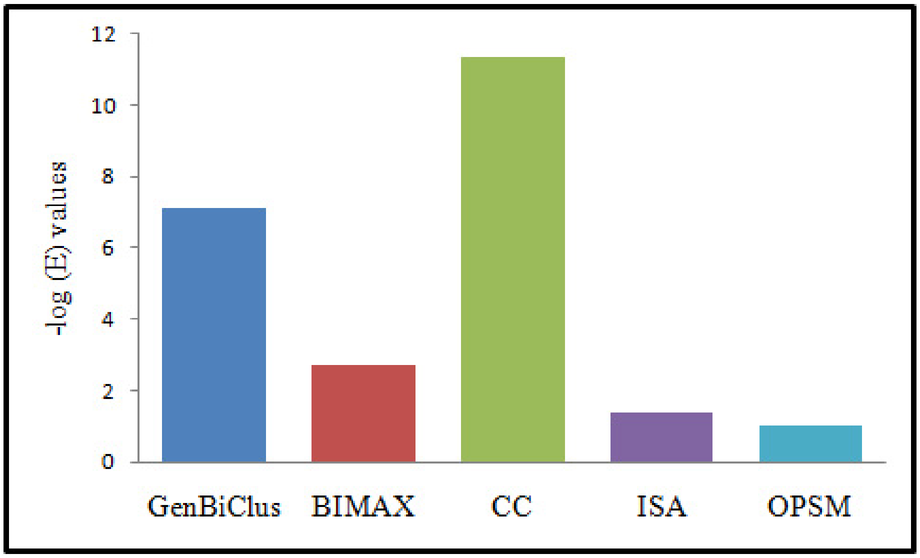


Figure 7.7: Best $-\log(E)$ values of Biclusters of GenBiClus and other algorithms on *S. cerevisiae* dataset

GenBiClus	CC	ISA	OPSM
$1.6e^{-96}$	$2.6e^{-10}$	$6.7e^{-57}$	$9.7e^{-12}$

Table 7.8: Best E values of motifs from Biclusters of GenBiClus and other algorithms on *DLBCL* dataset

GenBiClus	ISA	OPSM	CC
$4.7e^{-45}$	$2.8e^{-11}$	$1.2e^{-4}$	$2.0e^{-29}$

Table 7.9: Best E values of motifs from Biclusters of GenBiClus on *HBC* dataset

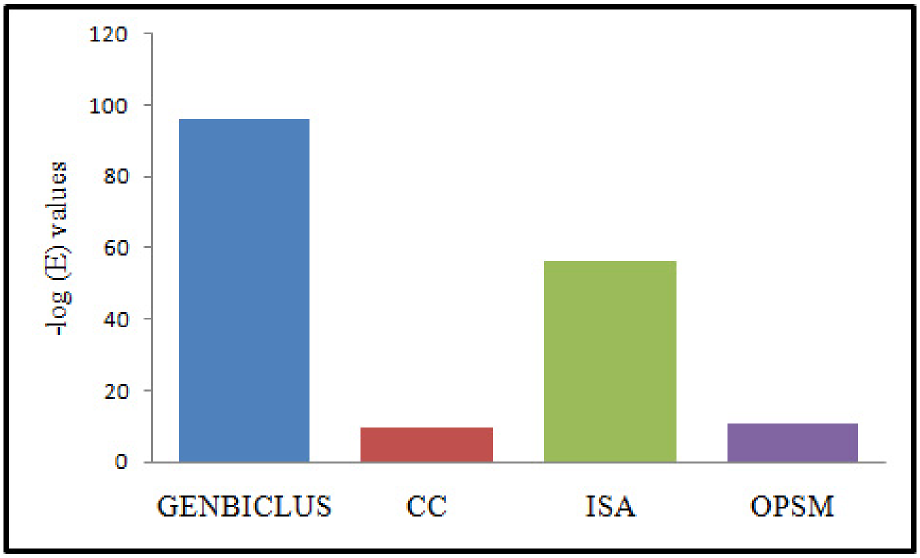


Figure 7.8: Best $-\log(E)$ values of Biclusters of GenBiClus and other algorithms on *DLBCL* dataset

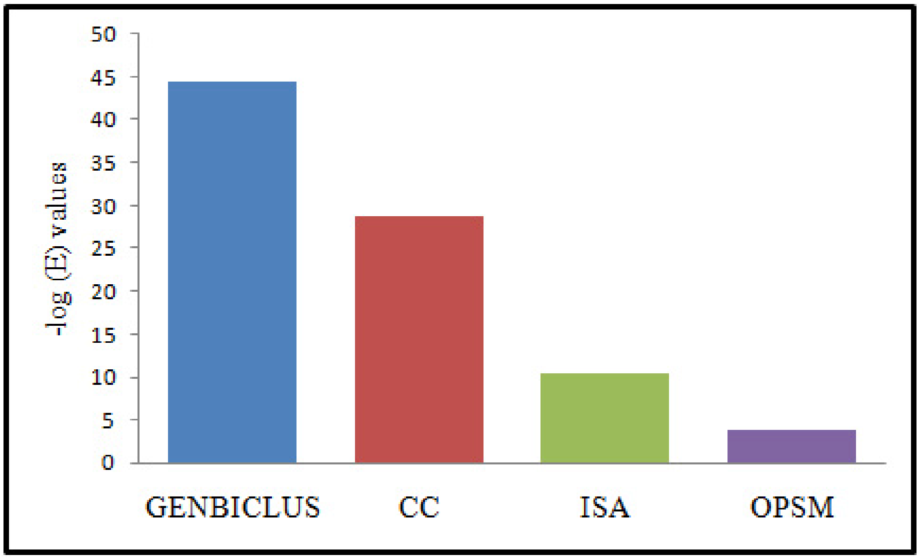


Figure 7.9: Best $-\log(E)$ values of Biclusters of GenBiClus and other algorithms on *HBC* dataset

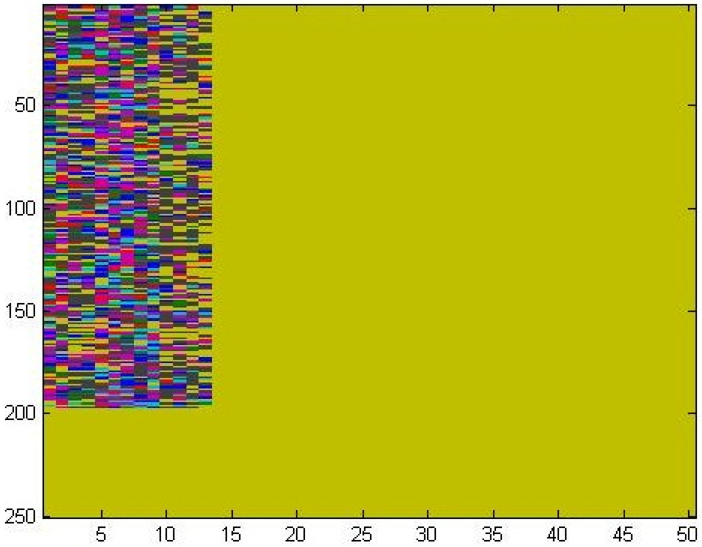


Figure 7.10: Bicluster extracted from *A. thaliana* dataset by GenBiClus perturbed with noise

Chapter 8

Concluding Remarks

Advancements in microarray technology have facilitated the generation of huge amount of gene expression data. Solutions to the *biclustering* problem is best suited for the analysis of this biological data for the following reasons: 1) genes are responsive to a small subset of conditions, 2) a gene may be responsible for more than one biological activity and 3) a condition or a group of conditions may trigger the expression of genes responsible for more than one activity.

The main contribution of this thesis is to develop a novel approach to biclustering gene expression data using **mutual information**. Unlike other similarity measures like distance measures and correlation coefficient, mutual information is a more general measure as it is able to extract both linear and nonlinear relationships. Though mutual information has been used earlier as a measure of similarity by traditional clustering algorithms, none of the biclustering algorithms have used it.

In our work we have proposed a set of biclustering algorithms which use mutual information as a measure of similarity. Through extensive experimentation on synthetic datasets and available real datasets we have demonstrated the utility of our work. The biclusters extracted by our algorithms are statistically more significant than the biclusters extracted by other algorithms. Algorithms are simple yet very effective (as exhibited by the experimental results) thereby demonstrating the strength of mutual information as a similarity measure. Time analysis of our algorithms and Tables 5.1, 6.1 and 7.1 show that the proposed algorithms scale well with the size of the data.

In MRB and GenBiClus, we have used kernel density estimator to estimate mutual information as the bin method cannot be used to compute the contribution of each condition to the mutual information. In BRC, both the bin method and the kernel density estimator can be used to estimate the mutual information. However, we have presented the results with kernel density estimator as it provides better estimates as discussed in Chapter 4.

Future Work: It will be interesting to see if machine learning tools like Discriminant Analysis, Mixture Models and Expectation Maximization which have been used successfully for the classification problem can provide better solutions to the problem. The major challenge in doing so would be to handle different sets of conditions for different biclusters and the overlapping nature of biclusters. Another direction of work would be to improve the results by generating ensembles for biclusters. The only work closely related to this is ensembles by Gullo et al. [F. 09] and Wang et al. [WLDJ11]. However, they create ensemble of biclusters which do not overlap either on genes or on conditions.

Bibliography

[AEH09] W. Ayadi, M. Elloumi, and J. K. Hao. A biclustering algorithm based on a Bicluster Enumeration Tree: Application to DNA microarray data. *BioData Mining*, 2(1), 2009.

[AMK00] T. Akutsu, S. Miyano, and S. Kuhara. Inferring qualitative relations in genetic networks and metabolic pathways. *Bioinformatics*, 16(8):727–734, 2000.

[APS06] T. K. Atwood and D. J. Parry-Smith. *Introduction to Bioinformatics*. Pearsons Eductaion, 2006.

[APW$^+$99] C.C. Aggarwal, C. Procopiuc, J. L. Wolf, P. S. Yu, and J. S. Parkoo. Fast Algorithms For Projected Clustering. In *Proceedings of ACM SIGMOD, International Conference on Management of Data, Philadelphia, Pennsylvania, USA*, pages 61–72, June 1999.

[AY00] C. C. Aggarwal and P.S. Yu. Finding Generalized Projected Clusters in High Dimensional Spaces. In *Proceedings of ACM SIGMOD, International Conference on Management of Data, Dallas, Texas, USA*, pages 70–81, May 2000.

[AYP11] J. Ahn, Y. Yoon, and S. Park. Noise-robust algorithm for identifying functionally associated biclusters from gene expression data. *Information Sciences*, 181(3):435–449, 2011.

[BBP$^+$06] S. Barkow, S. Bleuler, A. Prelic, P. Zimmermann, and E. Zitzler. BicAT: A biclustering analysis toolbox. *Bioinformatics*, 22(10):1282–1283, 2006. http://www.tik.ee.ethz.ch/sop/bicat.

[BDCKY02] A. Ben-Dor, B. Chor, R. Karp, and Z. Yakhini. Discovering Local Structure in Gene Expression Data: The Order-Preserving Submatrix Problem. In *Proceedings of the sixth annual International Conference on Computational Biology, RECOMB, Washington, DC, USA*, pages 49–57, April 2002.

[BDG$^+$07] A. Banerjee, I. S. Dhillon, J. Ghosh, S. Merugu, and D. S. Modha. A Generalized Maximum Entropy Approach to Bregman Co-clustering and Matrix Approximation. *Journal of Machine Learning and Research*, 8:1919–1986, 2007.

[BDSY99] A. Ben-Dor, R. Shamir, and Z. Yakhini. Clustering Gene Expression Patterns. *Journal of Computational Biology*, 6:281–297, 1999.

[BIB03] S. Bergmann, J. Ihmels, and N. Barkai. Iterative signature algorithm for the analysis of large-scale gene expression data. *Physical Review E*, 67(3):1–18, 2003.

[Bis06] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[BJKA02] S. Busygin, G. Jacobsen, E. Kramer, and C. Ag. Double Conjugated Clustering Applied to Leukemia Microarray Data. In *Proceedings of 2nd SIAM International Conference on Data Mining, Workshop on Clustering High Dimensional Data, Arlington , Virginia*, April 2002.

[BK00] A. J. Butte and I. S. Kohane. Mutual Information Relevance Networks: Functional Genomic Clustering using Pairwise Entropy Measurement. In *Proceedings of Pacific Symposium on Biocomputing, Hawaii*, pages 415–426, January 2000.

[BPC09] D. Bozdag, J. D. Parvin, and U. V. Catalyurek. A Biclustering Method to Discover Co-regulated Genes Using Diverse Gene Expression Datasets. In *Proceedings of the 1st International Conference on Bioinformatics and Computational Biology, BICoB, New Orleans, LA, USA*, pages 151–163, April 2009.

[CC00] Y. Cheng and G. M. Church. Biclustering of Expression Data. In *Proceedings of Eighth International Conference on Intelligent Systems for Molecular Biology, La Jolla/ San Diego, CA, USA*, pages 93–103, August 2000.

[Cla99] J.M. Claverie. Computational methods for the identification of differential and coordinated gene expression. *Human Molecular Genetics*, 8(10):1821–1832, 1999.

[CQB04] H. C. Causton, J. Quackenbush, and A. Brazma. *Microarray Gene Expression Data Analysis: A Beginner's Guide*. Blackwell Publishing, 2004.

[CSM06] B. Chandra, S. Shanker, and S. Misra. A new approach: Interrelated two way clustering of gene-expression data. *Statistical Methodology*, 3:93–102, 2006.

[CST00] A. Califano, G. Stolovitzky, and Y. Tu. Analysis of Gene Expression Microarrays for Phenotype Classification. In *Proceedings of the Eighth International Conference on Intelligent Systems for Molecular Biology, La Jolla/ San Diego, CA, USA*, pages 75–85, August 2000.

[CT91] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. John Wiley and Sons, 1991.

[DIB97] J. L. DeRisi, V. R. Iyer, and P. O. Brown. Exploring the Metabolic and Genetic Control of Gene Expression on a Genomic Scale. *Science*, 278:680–686, 1997.

[DLS99] P. D'Haeseleer, S. Liang, and R. Somogyi. Tutorial:Gene Expression Data Analysis and Modeling. In *Proceedings of Pacific Symposium on Biocomputing, Hawaii*, January 1999.

[DMBM07] R. Das, S. Mitra, H. Banka, and S. Mukhopadhyay. Evolutionary Biclustering with Correlation for Gene Interaction Networks. In *Proceedings of International Conference on Pattern Recognition and Machine Intelligence, PReMI, Kolkatta, India*, pages 416–424, December 2007.

[DMM03] I. S. Dhillon, S. Mallela, and D. S. Modha. Information-theoretic Co-clustering. In *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Washington, DC, USA*, pages 89–98, August 2003.

[Dom03] E. Domany. Cluster Analysis of Gene Expression Data. *Journal of Statistical Physics*, 110:3–6, March 2003.

[DPFS00] P. D'Haeseleer, P. Liang P, S. Fuhrman, and R. Somogyi. Genetic Network Inference: From Co-Expression Clustering to Reverse Engineering. *Bioinformatics*, 16(8):707–726, 2000.

[DWFS98] P. D'Haeseleer, X. Wen, S. Fuhrman, and R. Somogyi. Mining the Gene Expression Matrix: Inferring gene relationships from large scale gene expression data. In *Information processing in Cells and Tissues*, pages 203–212. Plenum Publishing, 1998.

[DWHL08] B. T. Sherman D. W. Huang and R. A. Lempicki. Systematic and integrative analysis of large gene lists using DAVID bioinformatics resources. *Nature Protocol*, 4(1), 2008. http://ncbi//david/gov.

[ESBB98] M.B. Eisen, P.T. Spellman, P.O. Brown, and D. Botstein. Cluster analysis and display of genome wide expression patterns. *Proceedings of the National Academy of Sciences of USA*, 95(25):14863–14868, 1998.

[F. 09] F. Gullo, C. Domeniconi and A. Tagarelli. Projective Clustering Ensembles. In *Proceedings of the Ninth IEEE International Conference on Data Mining, Miami, Florida, USA*, pages 794–799, December 2009.

[GA08] N. Gupta and S. Aggarwal. MIBiClus: Mutual Information based Biclustering Algorithm. *International Journal of Electrical and Computer Engineering*, 3:102–106, 2008.

[GA09] N. Gupta and S. Aggarwal. Modeling biclustering as an optimization problem using mutual information. In *Proceedings of the International Conference on Methods and Models in Computer Science, ICM2CS, Delhi, India*, December, 2009.

[GA10] N. Gupta and S. Aggarwal. MIB: Using Mutual Information for Biclustering gene expression data. *Pattern Recognition*, 43(8):2692–2697, 2010.

[GLD00] G. Getz, E. Levine, and E. Domany. Coupled two-way clustering analysis of gene microarray data. *Proceedings of the National Academy of Sciences of USA*, 97(22):12079–12084, 2000.

[GLDZ00] G. Getz, E. Levine, E. Domany, and M. Q. Zhang. Super-paramagnetic clustering of yeast gene expression profiles. *Physica A*, 279:457–464, 2000.

[GLY08] X. Gan, A. W. Liew, and H. Yan. Discovering biclusters in gene expression data based on high dimensional linear geometries. *BMC Bioinformatics*, 9, 2008.

[GSS91] E. J. Gardner, M. J. Simmons, and D. P. Snustad. *Principles of Genetics*. John Wiley and Sons, 1991.

[GVSS03] I. Gat-Viks, R. Sharan, and R. Shamir. Scoring clustering solutions by their biological relevance. *Bioinformatics*, 19(18):2381–2389, 2003.

[Hay07] S. Haykin. *Neural Networks-A comprehensive Foundation 2nd Ed.* Prentice Hall of India, 2007.

[HBH$^+$10] S. Hochreiter, U. Bodenhofer, M. Heusel, A. Mayr, A. Mitterecker, A. Kasim, T. Khamiakova, S. V. Sanden, D. Lin, W. Talloen, L. Bijnens, H. W. H. Gohlmann, Z. Shkedy, and D. A. Clevert. FABIA: Factor Analysis for Bicluster Acquistion. *Bioinformatics*, 26,(12):1520–1527, 2010.

[HBV01] M. Halkidi, Y. Batistakis, and M. Vazirgiannis. On Clustering Validation Techniques. *Journal of Intelligent Information Systems*, 17:2/3:107–145, 2001.

[HG95] H. Herzel and L. Grosse. Measuring Correlations in symbols sequences. *Physica A*, 216(4):518–542, 1995.

[HUIK99] K. Higo, Y. Ugawa, M. Iwamoto, and T. Korenaga. Plant cis-acting regulatory DNA elements (PLACE) database. *Nucleic Acids Research*, 27:297–300, 1999.

[Hun93] L. Hunter. Molecular Biology for Computer Scientists. In *Artificial intelligence and molecular biology*, pages 1–46. American Association for Artificial Intelligence, 1993.

[HZGD05] L. Hertzberg, O. Zuk, G. Getz, and E. Domany. Finding Motifs in Promoter Regions. *Journal of Computational Biology*, 12(3):314–330, 2005.

[IFB$^+$02] J. Ihmels, G. Friedlander, S. Bergmann, O. Sarig, Y. Ziv, and N. Barkai. Revealing modular organization in the yeast transcriptional network. *Nature Genetics*, 31(4):370–377, 2002.

[JTZ04] D. Jhiang, C. Tang, and A. Zhang. Cluster analsysis for gene expression data: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 16:1370–1386, 2004.

[Kau93] S. A. Kauffman. *The origins of Order: Self organization and Selection in Evolution*. Oxford University Press, 1993.

[KBCG03] Y. Kluger, R. Basri, J. T. Cheng, and M. Gerstein. Spectral Biclustering Of Microarray Data: Coclustering Genes And Conditions. *Genome Research*, 13(4):370–377, 2003.

[KBG$^+$07] S. Khan, S. Bandyopadhyay, A. R. Ganguly, S. Saigal, D.J. Erickson, V. Protopopescu, and G. Ostrouchov. Relative performance of mutual information estimation methods for quantifying the dependence among short and noisy data. *Physical Review E*, 76(2):026209–1–15, 2007.

[KL51] S. Kullback and R. A. Leibler. On Information and Sufficiency. *Annals of Mathematical Statistics*, 22(1):79–86, 1951.

[Koh97] T. Kohonen. *Self Organizing Maps*. Springer, 1997.

[KSAG05] A. Kraskov, H. Stogbauer, R. G. Andrzejak, and P. Grassberger. Hierarchical Clustering Based on Mutual information. *Europhysics Letters*, 70(2), 2005.

[KSG04] A. Kraskov, H. Stogbauer, and P. Grassberger. Estimating mutual information. *Physical Review E*, 69:066138–(1–16), 2004.

[KTW05] M. Kloster, C. Tang, and N.S. Wingreen. Finding regulatory modules through large-scale gene-expression data analysis. *Bioinformatics*, 21:1172–1179, 2005.

[Kul68] S. Kullback. Information theory and Statistics, 1968.

[Lan05] E. S. Lander. Finding regulatory modules through large-scale gene-expression data analysis. *The new genomics: global views of biology*, 21:536–539, 2005.

[LMT$^+$09] G. Li, Q. Ma, H. Tang, A. H. Peterson, and Y. Xiu. QUBIC: a qualitative biclustering algorithm for analyses of gene expression data. *Nucleic Acids Research*, 37(15):e101, 2009.

[LW07] X. Liu and L. Wang. Computing the Maximum Similarity Bi-clusters of gene expression data. *Bioinformatics*, 23(1):50–56, 2007.

[MB06] S. Mitra and H. Banka. Multiobjective evolutionary biclustering of gene expression data. *Pattern Recognition*, 39(4):2464–2477, 2006.

[MCA$^+$98] G. S. Michaels, D. B. Carr, M. Askenazi, S. Fuhrman, X. Wen, and R. Somogyi. Cluster Analysis and Data Visualization of Large Scale Gene Expression Data. In *Proceedings of Pacific Symposium on Biocomputing, Hawaii*, pages 42–53, January 1998.

[MDBM09] S. Mitra, R. Das, H. Banka, and S. Mukhopadhyaya. Gene interaction-An evolutionary biclustering approach. *Information Fusion*, 10:242–249, 2009.

[MDPM08] S. Mitra, S. Datta, T. Perkins, and G. Michailidis. *Introduction to Machine Learning and Bioinformatics*. Chapman and Hall book/CRC press, 2008.

[MK03] T. M. Murali and S. Kasif. Extracting conserved gene expression motifs from gene expression data. In *Proceedings of Pacific Symposium on Biocomputing, Hawaii*, pages 77–88, January 2003.

[MO04] S. C. Madeira and A. L. Oliveira. Biclustering Algorithms for Biological Data Analysis: A Survey. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 1(1):24–45, 2004.

[MO05] S. C. Madeira and A. L. Oliveira. A linear time biclustering algorithm for time series gene expression data. In *Proceedings of the 5th workshop on algorithms in bioinformatics, Mallorca, Spain*, pages 39–52, October 2005.

[MRL95] Y. I. Moon, B. Rajagopalan, and U. Lall. Estimation of Mutual Information using kernel density estimators. *Physical Review E*, 52(3):2318–2321, September 1995.

[NTAR11] J. A. Nepomuceno, A. Troncoso, and J. S. Aguilar-Ruiz. Biclustering of Gene Expression Data by Correlation-Based Scatter Search. *BioData Mining*, 4(3), 2011.

[PBZ$^+$06] 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(9):1122–1129, 2006.

[PMBG07] I. Priness, O. Maimon, and I. Ben-Gal. Evaluation of gene expression clustering via mutual information distance measure. *BMC Bioinformatics*, 8, 2007.

[RJLS10] P. H. Raven, G. B. Johnson, J. B. Losos, and S. R. Singer. *Biology*. Tata Mcgraw hill,seventh edition, 2010.

[RSA] http://rsat.ulb.ac.be/rsat.

[RWC$^+$02] A. Rosenwald, G. Wright, W. C. Chan, J. M. Connors, E. Campo, R. I. Fisher, R. D. Gascoyne, H. K. Muller-Hermelink, E. B. Smeland, and L. M. Staudt. The use of molecular profiling to predict survival after chemotherapy for diffuse large-B-cell lymphoma. *New England Journal of Medicine*, 346:1937–1947, 2002.

[SATB05] N. Slonim, G.S. Atwal, G. Tkacik, and W. Bialek. Information based clustering. *Proceedings of the National Academy of Sciences of USA*, 102:18297–18302, 2005.

[SDSK03] R. Steuer, C. O. Daub, J. Selbig, and J. Kurths. Measuring Distances Between Variables by Mutual Information. *Innovations in Classification, Data Science and Information Systems*, Proceedings of the 27th Annual Gfk Conference, Cottbus, March 2003.

[Sha48] C. E. Shannon. A mathematical theory of communication. *BellSystems Technical Journal*, 27:623–658, 1948.

[Sil86] B. W. Silverman. Density Estimation for Statistics and Data Analysis. *Monographs on Statistics and Applied Probability*, 1986.

[SKD$^+$02] R. Steuer, J. Kurths, C. O. Daub, J. Weiseand, and J. Selbig. The mutual information: Detecting and evaluating depencies between variables. *Bioinformatics*, 18(2):S231–S240, 2002.

[Slo02a] D. K. Slonim. From patterns to pathways: gene expression data analysis comes of age. *Nature Genetics*, 32:502–508, 2002.

[Slo02b] N. Slonim. *The information bottleneck: Theory and Applications*. PhD thesis, In Ph.D Thesis Computer Scence Department, Tel-Aviv University, 2002.

[SS02] R. Shamir and R. Sharan. Algorithmic Approaches to Clustering Gene Expression Data. In *Current Topics in Computational Biology*, pages 269–300. MIT Press, 2002.

[ST00] N. Slonim and N. Tishby. Document clustering using word clusters via the information bottleneck method. In *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval, Athens, Greece*, pages 208–215, July 2000.

[THC^{+}99] S. Tavazoie, J. D. Hughes, M. J. Campbell, R. J. Cho, and G. M. Church. Systematic determination of genetic network architecture. *Nature Genetics*, 22:281–285, 1999.

[TK99] S. Theodoridis and K. Koutroubas. *Pattern Recognition*. Academic Press, 1999.

[TSS02] A. Tanay, R. Sharan, and R. Shamir. Discovering statistically significant biclusters in gene expression data. *Bioinformatics*, 18(1):S136–S194, 2002.

[TT05] A. B. Tchagang and A. H. Tewfik. Robust Biclustering Algorithm (ROBA) for DNA microarray data Analysis. In *Proceedings of the 13th European Signal Processing Conference*, *Antalya Turkey*, September 2005.

[TZZR01] C. Tang, L. Zhang, A. Zhang, and M. Ramanathan. Interrelated Two-way Clustering: An Unsupervised Approach for Gene Expression Data Analysis. In *Proceeding of 2nd IEEE International Symposium on Bioinformatics and Bioengineering, Bethesda, Maryland, USA*, pages 41–48, November 2001.

[VBJ^{+}00] J. Vilo, A. Brazma, I. Jonassen, A. Robinson, and E. Ukkonen. Mining for putative regulatory elements in the yeast genome using gene expression data. In *Proceeding of International Conference of Intelligent Systems for Molecular Biology, La Jolla/San Diego, CA USA*, pages 384–394, August 2000.

[VDV^{+}02] L. J. Vanť Veer, H. Dal, M. J. Vijver, Y. D. He, A. M. Hart, M. Mao, H. L. Peterse, K. V. D. Kooy, M. J. Marton, A. T. Itteveen, G. J. Schreiber, R. M. Kerkhoven, C. Roberts, P. S. Linsley, R. Bernards, and S. H. Friend. Gene expression profiling predicts clinical outcome of breast cancer. *Nature*, 415 (6871):530–536, 2002.

[WLDJ11] P. Wang, K. B. Laskey, C. Domeniconi, and M. Jordan. Nonparametric bayesian co-clustering ensembles. In *Proceedings of the Eleventh SIAM International Conference on Data Mining, SDM, Mesa, Arizona, USA*, pages 331–342, April 2011.

[WWYY02] H. Wang, W. Wang, J. Yang, and P. S. Yu. Clustering by pattern similarity in large data sets. In *Proceedings of ACM SIGMOD International Conference on Management of Data, Madison, Wisconsin*, pages 394–405, June 2002.

[YCN04] K. Y. Yip, D. W. Cheung, and M. K. Ng. Harp: A practical projected clustering algorithm. *IEEE Transactions On Knowledge And Data Engineering*, 16:1387–1397, 2004.

[YWWY03] J. Yang, H. Wang, W. Wang, and P. Yu. Enhanced Biclustering on Expersion Data. In *Proceedings of the IEEE Symposium on Bioinformatics and Bioengineering, BIBE, Bethesda, MD, USA*, pages 321–327, 2003.

[ZAA08] J. Zola, M. Alluru, and S. Alluru. Parallel Information Theory Based Construction of Gene Regulatory Networks. In *Proceeding of 15th Interna-*

tional Conference on High Performance Computing HiPC, Bangalore, India, pages 336–349, December 2008.

[ZH02] M.J. Zaki and C. J. Hsiao. Charm: An efficient algorithm for closed itemset mining. In *Proceeding of 2nd SIAM International Conference on Data Mining, Arlington, VA, USA*, April, 2002.

[ZTOT04] Z. Zhang, A. Teo, B. C. Ooi, and K. L. Tan. Mining Deterministic Biclusters in Gene Expression Data. In *Proceedings of the fourth IEEE Symposium on Bioinformatics and Bioengineering, Taichung, Taiwan*, pages 283–292, March 2004.

[ZWD$^+$04] X. Zhou, X. Wang, E.R. Dougherty, D. Russ, and E. Suh. Gene Clustering Based on Clusterwide Mutual Information . *Journal of Computational Biology*, 11,(1):147–161, 2004.

List of Publications

- Neelima Gupta and Seema Aggarwal. MIB: Using mutual information for biclustering gene expression data. Journal of Pattern Recognition, 43(8):2692-2697, 2010.

- Neelima Gupta and Seema Aggarwal. Modeling Biclustering as an optimization problem using Mutual Information. In Proceedings of the International Conference on Methods and Models in Computer Science (ICM2CS), Delhi, India, December 2009.

- Neelima Gupta and Seema Aggarwal. MIBiClus: Mutual Information based Biclustering Algorithm. International Journal of Electrical and Computer Engineering, 3(2):102-106, 2008.

- Neelima Gupta and Seema Aggarwal. MIB: Using Mutual Information for Biclustering High Dimensional Data. In Proceedings of the European Conference on Data Mining, Amsterdam, Netherlands, pages 119-123, July 2008.

- Neelima Gupta and Seema Aggarwal. SISA: Seeded Iterative Signature Algorithm for Biclustering Gene Expression Data. In Proceedings of the European Conference on Data Mining, Amsterdam, Netherlands, pages 124-128, July 2008.