Gene Expression Mining Using Clustering and Association Mining Techniques

A thesis submitted in partial fulfillment of the requirements for award of the degree of Doctor of Philosophy

Swarup Roy

Registration No. 018 of 2012

School of Engineering
Department of Computer Science & Engineering

Tezpur University

December 2012

To
my father Late Sukumar Roy,
sweet daughter Srinika and my family.

Abstract

Mining DNA microarray gene expression data for discovering *in silico* biological knowledge is an emerging area of research in computational biology. Data mining is an important tool that has been applied successfully during the last two decades to analyze gene expression data. Clustering and classification have been widely used to analyze gene expression data. Association mining is relatively a promising and established technique in the area of data mining and knowledge discovery. However, a very little work has been done using association mining techniques to analyze expression data to obtain insights from data with regards to biological relevance. The purpose of this thesis is to study various association mining as well as clustering techniques and apply the techniques for gene expression data analysis. Our first contribution is a correlogram matrix based one-pass association mining technique (OPAM) for finding frequent itemsets from transaction database without candidate generation. We apply the correlogram matrix in finding strongly correlated item pairs (SCOPE) from transaction data using support based Pearson correlation coefficient. We also contribute an alternative non-parametric correlation coefficient measure for calculating strongly correlated item pairs. Comparison of SCOPE with other competitive algorithms using several synthetic and real world datasets shows the superiority of our methods. We have extended the concept for reconstruction of co-regulated gene co-expression network (GeCON). We use both synthetic and real dataset to establish that GeCON is superior in predicting gene network compare to other similar techniques and networks generated are having high biological significance. Finally,

we contribute a BiClust tree based technique (CoBi) for extracting co-regulated biclusters from gene expression data. The advantage of CoBi is twofold. First, it is one-pass in nature, and second, it can extract biclusters of high biological significance in polynomial time. We evaluated the performance of the proposed method using several benchmark gene expression datasets and the results are satisfactory.

Keywords: *Gene expression, association mining, clustering, biclustering, co-expression network, microarray*

Declaration

I, Swarup Roy, hereby declare that the thesis entitled "*Gene Expression Mining Using Clustering and Association Mining Techniques*" submitted to the Department of Computer Science and Engineering under the School of Engineering, Tezpur University, in partial fulfillment of the requirements for the award of the degree of Doctor of Philosophy in Computer Science and Engineering is based on bona fide work carried out by me. The results embodied in this thesis have not been submitted in part or in full, to any other university or institute for award of any degree or diploma.

(Swarup Roy)

DEPARTMENT OF
COMPUTER SCIENCE AND ENGINEERING

TEZPUR UNIVERSITY
NAPAAM, TEZPUR – 784 028

Certificate

This is to certify that the thesis entitled "*Gene Expression Mining Using Clustering and Association Mining Techniques*" submitted to the Tezpur University in the Department of Computer Science and Engineering under the School of Engineering in partial fulfillment of the requirements for the award of the degree of Doctor of Philosophy in Computer Science and Engineering is a record of research work carried out by Mr Swarup Roy under my personal supervision and guidance.

All helps received by him from various sources have been duly acknowledged.

No part of this thesis has been reproduced elsewhere for award of any other degree.

Signature of Research Supervisor

24/6/13

(Dhruba Kumar Bhattacharyya)
Designation: Professor
School: Engineering
Department: Computer Science and Engineering

University of Colorado at Colorado Springs

Department of Computer Science
1420 Austin Bluffs Parkway
P O Box 7150
Colorado Springs, Colorado 80933-7150
(719) 262-3325
(719) 262-3369 Fax

December 25, 2012

Certificate of the Joint Research Supervisor

This is to certify that the thesis entitled "*Gene Expression Mining Using Clustering and Association Mining Techniques*" submitted to the Tezpur University in the Department of Computer Science and Engineering under the School of Engineering in partial fulfillment of the requirements for the award of the degree of Doctor of Philosophy in Computer Science and Engineering is a record of research work carried out by Mr Swarup Roy under my personal supervision and guidance

All helps received by him from various sources have been duly acknowledged

No part of this thesis has been reproduced elsewhere for award of any other degree

Signature of Research Supervisor

(Jugal K Kalita)
Designation Professor
Affiliation
Department of Computer Science, University of Colorado,
1420 Austin Bluffs Parkway, Colorado Springs CO 80918 USA

DEPARTMENT OF
COMPUTER SCIENCE AND ENGINEERING

TEZPUR UNIVERSITY
NAPAAM, TEZPUR – 784 028

Certificate

This is to certify that the thesis entitled "*Gene Expression Mining Using Clustering and Association Mining Techniques*" submitted by Mr Swarup Roy to Tezpur University in the Department of Computer Science and Engineering under the School of Engineering in partial fulfillment of the requirements for the award of the degree of Doctor of Philosophy in Computer Science and Engineering has been examined by us on 29th September' 13 and found to be satisfactory.

The Committee recommends for award of the degree of Doctor of Philosophy.

Signature of

Principal Supervisor

External Examiner

Date: 29/9/2013

Acknowledgements

First and foremost, I would like to thank my supervisor, Prof. Dhruba Kr. Bhattacharyya of Tezpur University for giving me an opportunity to work under him on this challenging and burning topic and providing me ample guidance and support through the course of this research. I also thankful to him for inspiring my interests in data mining and gene expression data analysis. I am grateful for his guidance, encouragement and support throughout my doctoral research work at Tezpur University and also in my life. This thesis comes from his helpful discussions, supervision and meticulous attention to details.

I want to express my deep gratitude to my thesis co-supervisor, Professor Jugal Kumar Kalita of University of Colorado, USA, for his constant motivation, encouragement and support. He is a wonderful advisor and mentor. Throughout my research, he gave me countless constructive comments and insightful suggestions.

I take this opportunity to express my appreciations to all the members of my doctoral research committee including Prof. Malay Ananda Dutta, Dr Utpal Sharma and other faculty members for their constructive suggestions through out the research. I am also thankful to all my respected teachers in the Department and all my friends, especially J Binong and Pritpal Singh for their direct or indirect help, inspiration and motivation. I am grateful to all the technical and non-technical members of the Department for their support.

I want to express my greatest gratitude to my dear maa, dida, my beloved wife, my only loving sister, father and mother in-law for their endless love, constant support, encouragement and patients throughout my research without which I would not have reached this position.

Last but not least I would like to thank almighty for everything.

Contents

List of Figures

List of Tables

List of Algorithms

List of Definitions

Chapter 1

Introduction

"Computer Science is no more about computers than astronomy is about telescopes." – E. W. Dijkstra

With the rapid growth of DNA microarray technology, it is now possible to analyse expression pattern of several genes in a systematic and comprehensive manner at the genomic level[1]. Studying the expression pattern of genes in different experimental conditions, one may be able to understand the behaviour of genes and various pathways involved in biological processes. A gene expression level is a numerical value to measure how a particular gene is over-expressed or under-expressed in comparison to activity of the gene in normal conditions. Analysis of expression patterns can be helpful in discovering group of genes participate in similar biological processes or functions. Various biotechnology laboratories and pharmaceutical companies involved in *in silico* drug design, can identify the molecular targets that may interact with the drugs. Microarray analysis can assist drug companies in choosing the most appropriate candidates for participating in clinical trials of new drugs. Due to availability of diagnostic DNA microarrays, cancer research becomes very dominant in comparison to the other developing technologies since they are relatively easy to make and use.

1.1 Gene Expression Data Analysis

Gene expression data has become very essential in understanding behaviour of genes, different biological networks and cellular states. Study of such data may enable us to address various issues like, how a gene participate in a cellular process and what are the activities of different genes? In which cell and under which conditions, do the genes become active? How the activity of a gene are influenced by various diseases or drugs? Similarly, how genes contribute to diseases? One of the major goals in analyzing expression data is to determine how the expression of any particular gene may affect the expression of other genes or how one gene regulates another gene. Genes that affect one another may belong to the same gene network. A gene network is a set of related genes where expression of one gene may influence the other gene activity. Biological networks represent the biological relationships among genes or gene products (in the form of protein complexes). A group of co-regulated genes may form gene clusters that can encode proteins, which interact amongst themselves and take part in common biological processes. *In silico* reconstruction of such biological networks is essential for exploring regulatory mechanism and is useful in better understanding of the cellular environment to investigate complex interactions[2]. In an organism, co-expression of genes depend on sharing of the regulatory mechanism by them. It has been observed that genes with similar expression profiles are very likely to be regulators of one another or be regulated by some other common parent gene[3]. Another major goal of expression data analysis is to determine what genes are over expressed or under expressed as a result of certain biological conditions, such as, what genes are expressed in diseased cells that are not expressed in normal cells. Recently, it has been observed that a small set of genes are co-regulated and co-expressed under certain conditions and their behaviour being almost inactive for rest of the conditions. Discovering group of genes with similar or inverted expression profiles has been employed to identify co-expressed group of genes as well as to extract gene interactions or gene regulatory networks[4].

The advent of the microarray technology and availability of large number of expression datasets led to new challenges in extracting biologically significant knowledge from the gene expression data. As a result, mining such biological data has become an emerging area of research that requires interaction between biological research and computer science. Data mining is one of the most popular and indispensable computational tools to discover biological knowledge from large datasets.

1.2 Data Mining in Gene Expression Data Analysis

Data mining provides computational tools for effective mining of patterns or knowledge from large databases. Data mining involves various techniques from different computing paradigm such as databases and data warehouse technologies, statistics, machine learning, high-performance computing, pattern recognition, soft computing, data visualization, information retrieval, image and signal processing, and spatial or temporal data analysis. Data mining has become the first choice of researchers working towards biological knowledge extraction from gene expression data. Different data mining techniques such as classification, clustering and association rule mining are currently used on gene expression data to extract biological knowledge in the form of co-expressed genes or other relevant patterns. To improve relevance and utility of extracted knowledge, most of these applications require extending existing techniques to adapt them to biological data. During the past several decades extensive research in the field of biological data mining has made enormous contributions to our understanding of biological data. Publications of large number of articles points to the truth of this statement. However, not all of the techniques proposed address all the issues and challenges. Some of these are evolutionary, enhancements of previously developed work; others are revolutionary, introducing new concepts and methods. The present trend in research in gene expression data analysis is to mine expression data to determine genes with common functional characteristics and to discover how gene(s) regulate each other. Below

we present a few issues related to gene expression data analysis and the use of data mining techniques to address these issues.

1.3 Motivation

Based on a comprehensive literature survey we come to the following conclusions:

- While it is important to determine which genes are related, we also need to understand the mechanism of how genes relate and how they regulate one another in the form of gene networks.

- Most available gene expression data analysis methods are based on clustering algorithms which attempt to group genes on the basis of their expression correlation in different biological situations. However, the clustering approach fails to infer inter-relationship between genes, which is very significant from system biologys point of view in the reconstruction of biological networks such as gene regulatory networks. Association mining is an unsupervised data mining technique evolved with an idea to find relationships among the items from market basket data. The idea has been attempted to extend for drawing significant relationships among genes.

- Very little work has used association mining techniques to analyze expression data to get insight into data with biological relevance.

- The most costly step in association mining is the number of database passes. Minimizing the number of database passes may improve the performance of any association mining based gene expression data analysis technique.

- Data mining techniques for finding groups of biologically related genes use Euclidean distance or Pearson correlation coefficient as a measure of proximity. Euclidean distance does not score well in handling gene expression data that includes shifting and scaling patterns or profiles. The correlation coef-

ficient is not robust with respect to outliers, thus potentially yielding false positives, assigning a high similarity scores to a pairs of dissimilar patterns.

- The most common activity of gene expression analysis is the pair-wise comparison among gene expression profiles. Any traditional method needs N^2 passes over the dataset for comparison of N gene expression profiles. Computationally effective method, for comparison in limited number of passes, may improve the overall performance.

- Traditional approaches group genes based on profile similarity as co-expressed genes. All genes in a group share similar patterns. Recently, researchers have observed that co-regulated genes also share negative patterns or inverted behaviours, which existing techniques are unable to detect.

- Often, it is noted that under certain conditions, a small set of genes are co-regulated and co-expressed, their behaviour being almost independent of the rest of the conditions. As a result, biclustering techniques have been applied to gene expression data. Most interesting variants of this problem are NP-complete requiring either extensive computational effort or the use of lossy heuristics to short-circuit the calculation.

1.4 Contributions

In this thesis, we systematically study and solve problems that state-of-the-art association mining and data clustering algorithms face when applied to gene expression data. We use clustering and association mining techniques to discover relationships among co-regulated genes in the form of gene biclusters, gene clusters and gene co-expression networks. We explore expression profile based similarity measures in order to find pair-wise relationships among genes. We propose a method called OPAM to find frequent itemsets from large transaction databases using a single pass of the database without the candidate generation step. We solve the problem of strongly correlated pair finding from transactions database using a one-pass ap-

proach. We also propose an alternate support based non-parametric approach to obtain strongly correlated pairs. We apply the above mentioned concept to compare two genes expression profiles in order to capture gene co-expression networks. And, finally, we design a polynomial time co-regulated gene biclustering technique.

A brief about the developed techniques are discussed below:

OPAM: An algorithm called OPAM (One-Pass Association Mining), has been proposed, for finding all frequent itemsets from a transactions database without generating any candidate sets. OPAM uses a data structure called correlogram matrix to generate all two-element frequent itemsets, and then exploit the vertical layout of the database to generate the remaining frequent itemsets. OPAM adopts an integrated approach to solve the frequent itemset finding problem in a single pass over the database.

SCOPE: SCOPE (Strongly COrelated Pair Extraction) is a one-pass technique to find strongly correlated pairs based on Pearson correlation coefficient from a large transaction database without using any candidate generation. We also propose an extension of SCOPE to extract top k strongly correlated pairs. We use a correlogram matrix for efficiently extracting pair-wise support of all the itemsets. A support based Spearman rank order correlation finding technique is also proposed to find strongly correlated pairs.

GeCON: GeCON (Gene CO-expression Network) is proposed to extract co-regulated gene co-expression networks from microarray data using expression pattern based local proximity measure. Pair-wise supports are computed for each pair of genes based on changing tendencies over the dataset in order to calculate the local proximity between pairs of genes. Gene pairs showing similar expression profiles over a given number of conditions are used to construct the gene co-expression network. Positive and negative regulation information are also captured during network construction.

CoBi: CoBi (Co-regulated Biclustering) is an expression pattern based biclustering technique for grouping both positively and negatively regulated genes

from gene expression data. Regulation patterns and similarities in degrees are taken into account while computing similarity between two genes. Unlike traditional biclustering techniques which use greedy iterative approaches and are NP complete, CoBi uses a tree based technique inspired from OPAM, for finding a set of biologically relevant biclusters in polynomial time.

1.5 Organization of the Thesis

The thesis is organized as follows:

- *Chapter 2* gives a background of the study. It discusses in details about microarray technology and the various patterns available in expression data and how data mining is helpful in discovering such patterns.
- *Chapter 3* discusses various association mining techniques and proposes a new one-pass association mining technique, referred as OPAM for finding frequent itemsets from transactions database without the help of candidate generation step.
- *Chapter 4* presents a correlogram matrix based technique for computing all strongly correlated item pairs from transactions database. It also proposes an alternate non-parametric correlation computation techniques.
- *Chapter 5* describes a pattern based co-expression networks finding technique from gene expression data.
- *Chapter 6* presents a co-regulated biclustering technique that extracts biclusters in polynomial time.
- *Chapter 7* summarizes the work with concluding remarks.

Chapter 2

Background

2.1 Molecular Biology

The domains of biochemistry in general and molecular biology in particular, are concerned with the basic molecular principles of life. Biological objects interact with each other making possible all different forms of life. A central interest of molecular biology is the flow of information within an organism. Living organisms store all information that is necessary for growth, reproduction, and evolution in so-called genes on the DNA (sometimes RNA in simpler organisms). Deoxyribose nucleic acid (DNA) or Ribose nucleic acid (RNA) encodes the genetic instructions used in the development and functioning of all known living organism and thus act as informational molecules. The genetic information is encoded as a sequence of nucleotides: A (adenine), C (cytosine), T (thymine) G (Guanine) and U (Uracil). T appears only in DNA and U appears in RNA molecule. Watson-Crick model describes DNA molecules as double-stranded helices structure. It consists of two long polymer chains of nucleotide molecules attached with alternating sugars (deoxyribose) and phosphate backbone, which are both winded around a common axis that gives a double-helical structure to DNA. Each nucleotide molecule from one chain always bonds with a complementary nucleotide molecule from other chain and form a nucleotide pair called *base-pair* (bp). A base-pair is simply an interaction between the bases standing opposite of each other. These interactions are

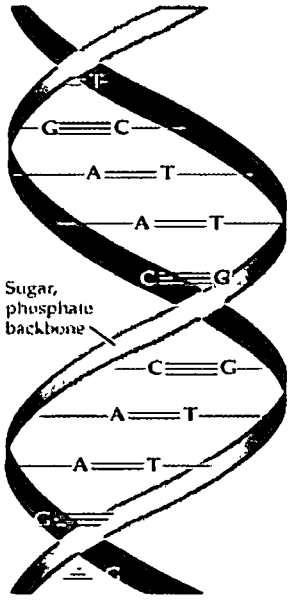


Figure 2.1: Double helix structure of DNA

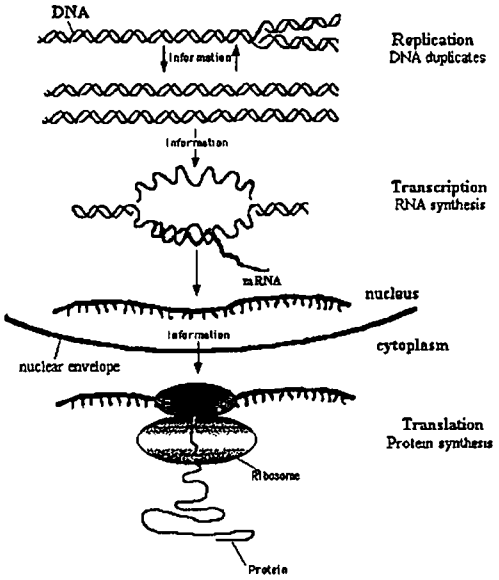


Figure 2.2: Central Dogma: flow of information from DNA to Protein

based on hydrogen bonds. Erwin Chargaff[5] suggested base pairing rule. According to the rule, A binds only to T (A-T) and C binds only to G (C-G), to form a base-pairs (see Figure 2.1, taken from[a]). A DNA molecule of 100 bp thus consists of two antiparallel sequences, each 100 nucleotides (bases) long. The human genome roughly consists of 1.3×10^9 of such base pairs[6].

In contrast to DNA molecule, RNA is a single-stranded chain consists of four nucleotide molecules A, T, G and U. In living organism, there are four different types of RNA is available. *Ribosomal RNA*s (rRNA) are structural components of multi-protein complexes called ribosomes and protein synthesis takes place at the ribosomes. *Messenger RNA*s (mRNA) acts as carrier of genetic information from the genes to the ribosomes and *transfer RNA*s (tRNA) play the role of a translator, that translates the genetic information of the mRNA into a sequence of special bio-molecules called *amino acids*. Amino acids are the basic building blocks of protein. Protein molecules are polymers, i.e. consist of thousands or millions of atoms and responsible for all major cellular activities. It act as enzyme,

[a] *http://wps.prenhall.com/wps/media/objects/3313/3393159/blb2511.html*

anti-bodies, regulatory substances, stabilisers, or carriers of other substances.

Genes are nothing but regions of the DNA and act as a repository of biological information which is necessary to build and maintain an organism's cells. It includes construction and regulation of proteins as well as other molecules that ultimately determine the growth and functioning of the living organism and transfer genetic traits to next generation. This is termed as central dogma of molecular biology. Entire DNA sequence of an organism do not play active role in cellular activities. In case of human genome, only 2-3% of the whole human DNA are functional. The functional part or the coding part of DNA is only responsible for protein synthesis. The remaining DNA does not encode for any protein. This DNA is sometimes referred to as "junk-DNA" or Non-coding DNA. Recent research reveals that junk-DNA plays critical roles in controlling how cells, organs and other tissues behave. Genes are templates for protein construction within a cell. Protein synthesis takes place within the cell through the process of transcription and translation. In *transcription* phase, a molecular complex called RNA polymerase-II creates a copy of a gene from the DNA to messenger RNA (mRNA) inside the nucleus. The mRNA travels from nucleus to the cytoplasm for protein synthesis, where it then binds with ribosome. Ribosome is a complex molecule based on ribosomal RNA (rRNA) and proteins. At the ribosome, mRNA is used as a blueprint for the production of a protein; this process is called *translation*. The mRNA moves along the protein synthesis site i.e. ribosomes, with a set of three-nucleotides called *codons*. Transfer RNA (tRNA) provides a compatible *anticodon* and is hybridised onto the mRNA. Finally, the amino acids bound to the RNA form polypeptide chain. This process continues until the translation process reaches a stop codon, which terminates the polypeptide synthesis. The entire process is called *gene expression*. A schematic drawing of the process of protein synthesis is illustrated in Figure 2.2 taken from National Health Museum[a].

[a] *http://www.accessexcellence.org*

2.2 Overview of Microarray Technology

Traditional experimentation system in molecular biology are capable of studying only a few genes in a single experiment. However, for a traditional methods, it is difficult to capture the dynamic behaviour or the activities of a gene that is going on inside a cell. DNA microarray technology provides a convenient and effective platform for monitoring activity of thousands of genes simultaneously. DNA microarray analysis is a fast and versatile approach to perform high throughput explorations of genome structure, gene expression, and gene function at both cellular and organism levels. Microarray analysis is a complex multi-step process involving various areas of expertise such as molecular biology, image analysis, computing and statistics.

There are five major steps in performing a typical microarray experiment[a]. The steps are illustrated in Figure 2.3 taken from[b].

1. **Preparation of microarray:** In the preparation process, polymerase chain reaction (PCR) technique is used to amplify the DNA of interest using a universal primer or gene specific primers to generate thousands to millions of copies of a particular DNA sequence. The purity of the DNA fragments are then checked by sequencing or using an agarose gel through the estimation of the DNA concentration. The next step is spotting the DNA solution onto special glass slides coated with chemical materials such as polyethyleneimine polymer p-aminophenyl trimethoxysilane. Precision in spotting is achieved using precisely controlled robotic pins or other equivalent technology such as inkjet printing. The last step of manufacturing glass DNA microarrays is the post-print processing step involving drying of the DNA on the slide overnight at room temperature and the use of UV cross-linking to prevent subsequent binding of the DNA, and to decrease the background signal upon hybridisation of a labelled target.

[a] http://grf.lshtm.ac.uk/microarrayoverview.htm
[b] http://www.uni-koeln.de/med-fak/biochemie/transcriptomics/07_analysis.shtml

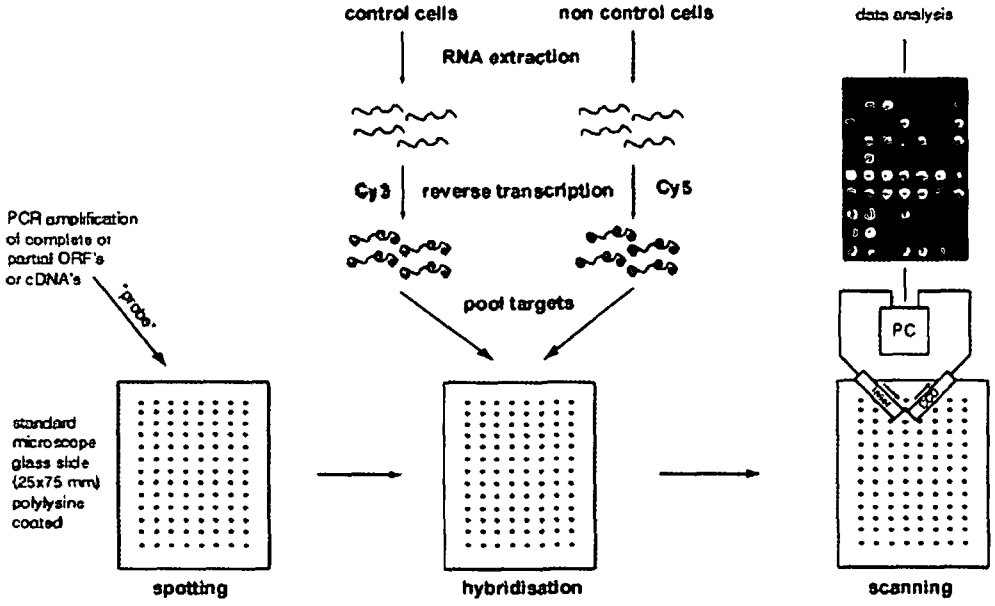


Figure 2.3: Steps in Miroarray experiments

2. **Sample preparation and labelling:** Sample preparation begins with isolation of mRNA copies that represents genes, i.e., coding genes that expressed during sample collection. It is a very vital step as because the overall success of any microarray experiment highly depends on the quality of the RNA collected. Purity in terms of homogeneity or uniformity of the mRNA is an important factor for proper hybridization process, particularly when fluorescence is used, as cellular proteins, lipids, and carbohydrates can mediate significant nonspecific binding of labeled cDNAs to matrix surfaces. The mRNA extracted from both the target and the reference samples are then converted into complementary DNA (cDNA) using a reverse-transcriptase enzyme. To initiate cDNA synthesis, this step also requires a short primer. Next, each cDNA (target and reference)is labelled with a fluorescent cyanine dye (i.e. either Cy3 or Cy5).

3. **Hybridisation:** Hybridisation is the step of combining two complementary single stranded-DNA to form a double-stranded molecule. The labelled cDNAs (target and reference) are purified to remove contaminants such as

primers, unincorporated nucleotides, cellular proteins, lipids, and carbohydrates. After purification, the labelled cDNA is hybridised against cDNA molecules spotted already on a glass slide. Each molecule in the labelled cDNA will only bind to its appropriate complementary target sequence on the static array. Before hybridisation, the microarray slides are incubated at a high temperature with solutions of saline-sodium buffer (SSC), Sodium Dodecyl Sulfate (SDS) and bovine serum albumin (BSA) to reduce the background due to nonspecific binding.

4. **Washing:** To remove any unhybridized labelled cDNA from the array and to increase stringency of the experiment by reducing cross hybridisation, the slides are washed after hybridisation. The latter is achieved by either increasing the temperature or lowering the ionic strength of the buffers

5. **Image acquisition and Data analysis:** The final step of microarray experiments involve image acquisition and data analysis of the array. The slide is dried at first and then scanned using a laser scanner to determine how much labelled cDNA (probe) is bound to each target spot. Laser excitation of the incorporated targets yields an emission with characteristic spectra, which is measured using a confocal laser microscope. Software used for microarray analysis often represents green spots as up-regulation a gene compared the to control, red sport as down-regulation a gene in the experimental sample, and yellow to represent equal abundance in both experimental and control samples. In the data analysis phase, the relative expression levels of the genes in the sample and in the controlled populations can be estimated from the fluorescence intensities and colour for each spot. Based on the amount of probe hybridized to each target spot, information is gained about the specific mRNA composition and the representative in the sample. The logarithm of the ratio of raw red/green fluorescence intensities are taken to convert them into log intensities. In the case of microarray experiments, there are many sources of systematic variation that affect measurements of gene expression

levels. The process of eliminating such variations allows appropriate comparison of data obtained from the two samples by using various normalization processes. The processed data, after normalization, can then be represented in the form of a matrix, often called gene expression matrix.

2.3 Gene Expression Data

Microarray is an indispensable technology in molecular biology that helps in assessing expression of a large number of genes under multiple conditions such as time-series, tissue samples (e.g., normal versus cancerous tissues), and experimental conditions. With the help of microarray experiments one can monitor simultaneously, the expression levels of several genes at a genome scale. To gain better understanding of a gene and its behaviour inside cell, various patterns can be derived by analysing the change in expression of the genes. An expression profile (of a gene or a sample) can be represented in vector space[7]. For example, an expression profile of a gene can be considered a vector in n dimensional space (where n is the number of conditions), and an expression profile of a sample with m genes can be considered a vector in m dimensional space (where m is the number of genes). In the example given below, the gene expression matrix X with m genes across n conditions is an $m \times n$ matrix, where the expression values for gene i in condition j is denoted as x_{ij}:

$$X = \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,n} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m,1} & x_{m,2} & \cdots & x_{m,n} \end{bmatrix}.$$

Formally, it can be defined as:

Definition 2.3.1 (Gene Expression Data) : Let $G = \{G_1, G_2, \cdots, G_m\}$ be a

set of m genes and $R = \{T_1, T_2, \cdots, T_n\}$ be the set of n conditions or time points of a microarray dataset. The gene expression dataset X can be represented as an $m \times n$ matrix, i.e., $X_{m \times n}$ where each entry $x_{i,j}$ in the matrix corresponds to the logarithm of the relative abundance of mRNA of a gene.

The expression profile of a gene i can be represented as a row vector:

$$G_i = \begin{bmatrix} x_{i,1} & x_{i,2} & x_{i,3} & \cdots & x_{i,n} \end{bmatrix}.$$

The expression profile of a sample j can be represented as a column vector:

$$G_i = \begin{bmatrix} x_{1,j} \\ x_{2,j} \\ x_{3,j} \\ \cdots \\ x_{m,j} \end{bmatrix}.$$

A subset of real gene expression data from a Homo-sapiens microarray dataset is given in Table 2.1.

2.4 Patterns in Gene Expression Data

Microarray data is essentially the logarithm of the ratio of raw red/green fluorescence intensities at a certain spot and is continuous in nature. The notion of pattern in microarray data introduced in[8] as follows:

Definition 2.4.1 (Expression Pattern) : Given a gene G_i, its expression values under a series of varying conditions or under a single condition form a range of real values. Suppose this range is $[a, b]$ and an interval $[c, d]$ is contained in $[a, b]$. Thus G_i is a vector of real numbers within the range $[a, b]$, denoted as $G_i@[a, b]$, is called an item, meaning the values of G_i are limited inclusively between a and b.

A set containing one single item is called a *pattern*. A set of several items,

ORF	C1	C2	C3	C4
GALNT5	-3.474	-3.837	-4.644	-5.059
APOE	-2	-1.943	-1.786	-1.737
IDH3B	1.449	1.299	0.993	0.832

Table 2.1: Sample gene expression data from Homo sapiens

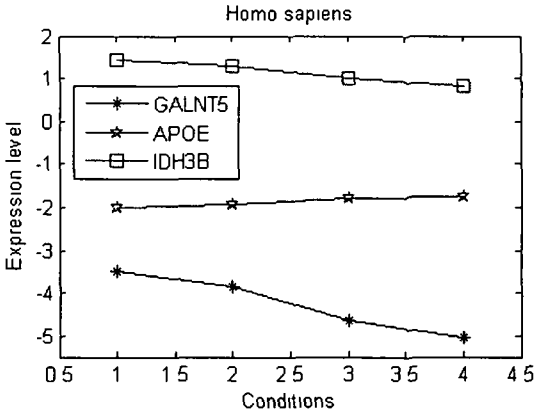


Figure 2.4: Profile plot of Homo sapiens expression data

which come from different genes is also called a pattern. So, a pattern looks like:

$$\{G_{i1}@[a_{i1}, b_{i1}], \cdots, G_{ik}@[a_{ik}, b_{ik}]\}$$

where $i_t \neq i_s, 1 \leq t, s \leq k$, if $k > 1$.

Example patterns from a Homo sapiens microarray data (Table 2.1) and its corresponding profile plots are shown in the Figure 2.4.

It has been observed that from a biological point of view, patterns play an important role in discovering functions of genes, disease targets or gene interactions. A number of different patterns have been identified in biologically significant gene groups.

2.4.1 Shifting and Scaling patterns

In shifting patterns[8] the gene profiles show similar trends, but distance-wise, they may be away from each other (see Figure 2.5).

In terms of expression values, gene patterns follow an additive distance between them. Formally, shifting pattern can be defined as follows.

Definition 2.4.2 (Shifting Pattern) : Given two gene expression profile $G_i = \{E_{i1}, E_{i2}, \cdots, E_{ik}\}$ and $G_j = \{E_{j1}, E_{j2}, \cdots, E_{jk}\}$ with k expression values, a profile

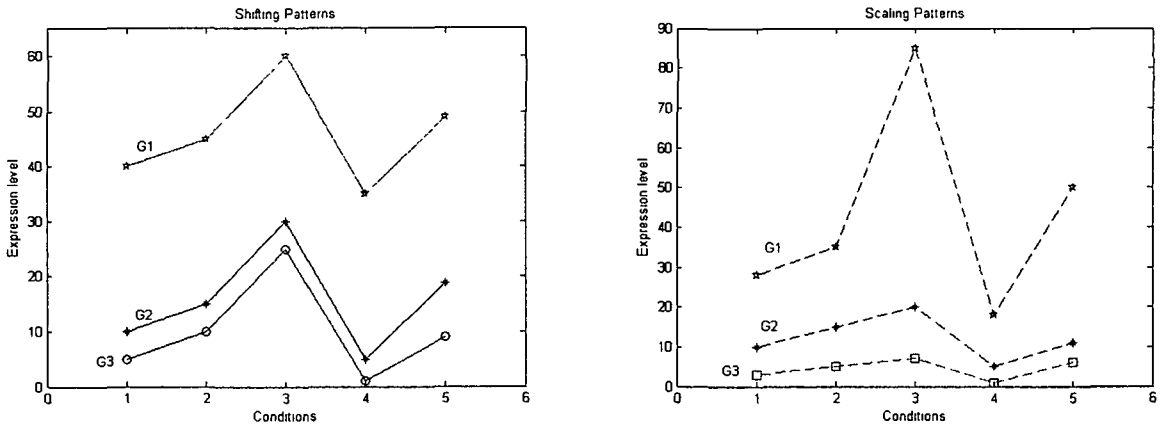


Figure 2.5: Expression profile plot shows Shifting and Scaling patterns

is called as shifted pattern, if expression value of E_{ik} can be related with E_{jk} with constant additive factor π_k under k^{th} condition. This can be written as follows.

$$E_{ik} = E_{jk} + \pi_k, \text{ for } i = 1 \text{ to } k \tag{2.1}$$

Similarly, scaling patterns in gene expression follow roughly a multiplicative distance between the patterns. Scaling pattern can be defined as:

Definition 2.4.3 (Scaling Pattern) : Given two gene expression profile $G_i = \{E_{i1}, E_{i2}, \cdots, E_{ik}\}$ and $G_j = \{E_{j1}, E_{j2}, \cdots, E_{jk}\}$ with k expression values, a profile is called as scaling pattern, if expression value of E_{ik} can be related with E_{jk} with constant multiplicative factor π_k under k^{th} condition. This can be written as follows.

$$E_{ik} = E_{jk} \times \pi_k, \text{for } i = 1 \text{ to } k \tag{2.2}$$

As shown in Figure 2.5, values of G_2 are roughly three times larger than those of G_3, and values of G_1 are roughly three times larger than those of G_2. In nature, it may happen that due to different environmental stimuli or conditions, the pattern G_3 responds to these conditions similarly, although G_1 is more responsive or more sensitive to the stimuli than the other two.

2.4.2 Coherent patterns

A group of genes showing similar pattern tendency across different conditions is called coherent. Such a group shows some kind of a co-expression in the expression profile. Co-expressed genes are likely to be involved in the same cellular processes. In practice, co-expressed genes may belong to the same or similar functional categories indicating co-regulated families[4]. Coherent gene expression patterns may characterize important cellular processes and may provide a foundation for understanding the regulation mechanism in the cells[9]. The patterns shown in Figure 2.5 are the examples of coherent patterns.

2.4.3 Co-regulated patterns

Often, coherent patterns are divided into two categories namely, positively regulated patterns and negatively regulated or inverted patterns. Sometimes, a group of genes that are positively or negatively regulated also called co-regulated genes. In Figure 2.4 genes GLANT5 and IDH3B show similar pattern or positively regulated patterns. On the other hand IDH3B or GLANT5 showing inverted or negative patterns with APOE. Biologically all three genes are very significant.

Thus, gene expression data analysis involves pattern finding. Data mining is the study of techniques that extract patterns from large amount of data. As a result, data mining provides the major tools for gene expression data analysis. Below we present a brief discussion of data mining techniques.

2.5 Data Mining

Data mining is a computational technique to analyze large volumes of data for finding relationship within the data that helps in predicting new fact such as how components of the data are related to one another. Fayyad, Piatetsky-Shapiro and Smyth in 1996[10] defined data mining as: *"The non-trivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data"*. American statistician David Hand in 1998[11] also defined data mining as: *"A new*

discipline lying at the interface of statistics, database technology, pattern recognition, and machine learning, and concerned with secondary analysis of large data bases in order to find previously unsuspected relationships, which are of interest of value to their owners". Data mining is an intermediate step in the KDD (Knowledge Discovery in Databases) process[12] that consists of applying data analysis and discovery algorithms that produce a particular enumeration of patterns (or models) in the data. In general, the knowledge discovery process consists of an iteration sequence of the following steps:

- Data cleaning: handles noisy, erroneous, missing or irrelevant data.
- Data integration: where multiple, heterogeneous data source may be integrated into one.
- Data selection: where data relevant to the analysis task are retrieved from databases.
- Data transformation: where data are transformed or consolidated into forms appropriate for mining by performing summary or aggregation operation.
- Pattern evaluation: identifies the truly interesting patterns representing knowledge based on some measures of interestingness.
- Knowledge presentation: where visualization and knowledge representation techniques are used to present the mined knowledge to the user.

2.5.1 Application of data mining

Today data mining offers value across a broad spectrum of industries.

1. *Marketing*: In marketing, the primary application[13] of data mining is to analyze customer databases to identify potential customer groups and forecast their behaviour. Another popular application is market-basket data analysis systems, which extracts interesting patterns such as, "If customer bought X, he/she is also likely to buy Y and Z".

2. *Telecommunications*: Another application of data mining is the telecommunication alarm-sequence analyzer system (TASA). It was built in collaboration with a telecommunications equipment manufacturer and three telephone networks[14]. Speciality of the system is that, it uses a novel framework for locating frequently occurring alarm episodes from the alarm stream and presenting them as rules. Large sets of discovered rules can be explored with flexible information-retrieval tools supporting interactivity and iteration. In this way, TASA offers pruning, grouping, and ordering tools to refine the results of a basic brute-force search for rules.

3. *Medical Application*: Medical applications are another fruitful area. Data mining can be used to predict the effectiveness of surgical procedures, medical tests or medications[15]. Recently, it has also been used in Medical imaging applications[16] to detect or predict diseases like cancer, which are sometime impossible for the human specialist to detect.

4. *Bioinformatics*. Data mining is used in the fields of biology and bioinformatics[17]. Currently, data mining is extensively used in the analysis of gene expression data.

5. *Pharmaceutical*: Pharmaceutical firms are mining large databases of chemical compounds and genetic material to discover substances that might be candidates for development as agents for the treatments of diseases[18].

6. *Network Security*: Data mining is successfully used to predict the usage patterns in network[19] to detect intrusion in the networks.

2.5.2 Data mining tasks

In general, data mining tasks can be classified into two categories[12].

Descriptive mining: It is the process of discovering the essential characteristics or general properties of the data in the database. Clustering, association and sequence mining are some of the descriptive mining techniques.

Predictive mining: This is the process of inferring patterns from data to make predictions. Classification, regression and deviation detection are predictive mining techniques.

There are several widely used data mining techniques. Traditionally, these techniques are used independently. These techniques include: classification, clustering, association rule mining, prediction and time-series analysis.

Classification: Classification[12] is a supervised technique that partitions a given dataset into disjoint classes using a class attribute. A classifier model is built based on training data and later the model is used for predicting class of an unknown sample The goal of classification is to analyze the training set and to develop an accurate description or model for each class using the attributes presented in the data. Many classifications models have been developed including neural networks, genetic models, and decision trees.

Clustering: Clustering[12] is an unsupervised technique to group data into clusters with high intra-cluster similarity and low inter-cluster similarity. A similarity or distance measure is important criteria in deciding the quality of the cluster. To a large extent, quality depends on the appropriateness of the similarity measure for the data set or the domain of application. For example, clustering can be used to divide a population into distinct groups, such that each group can be treated with a different strategy. A number of clustering techniques are available. Partitioning methods, hierarchical methods, density-based methods, grid-based methods, and model based methods are some of the well known clustering techniques. The basic difference between classification and clustering is that classification assumes prior knowledge on class labels, while clustering does not assume any knowledge of classes.

Association Rules: Association rule mining[12] is a data mining technique use to find interesting associations among a large set of data items. Association rule mining started with an initial idea to apply on market-basket analysis. In market-basket analysis, purchasing behaviour of customers are analyzed to

find association between different items that customers place in their "shopping baskets". The discovery of such association rules can help retailers in developing new marketing and placement strategies as well as logistics plan for inventory management that ultimately leads to business promotion. Association rules identify items that are frequently purchased together by customers. They make attempts to associate a product A with another product B so as to infer "whenever A is bought, B is also bought", with high confidence (i.e., the number of times B occurs when A occurs).

Prediction: Prediction techniques[12] are based on some continuous valued attributes. The previous history of the attributes is used to build the model. This technique is commonly used for predicting product sales.

Time-Series analysis: Time-series analysis[12] analyzes large sets of time series data to find regularities and interesting characteristics, including similar sequences or sub sequences, and sequential patterns, periodicities, trends and deviations. For example, one may predict trends in the stock values for a company based on its stock history, business situation, competitor performance and current market.

2.6 Discussion

The two classical data mining methods, i.e., data clustering and data classification, have been widely used to analyze gene expression data. These methods are valuable exploratory tools in data mining, and used successfully throughout the last two decades to explore biological knowledge from gene expression data. While classification helps in identifying genes responsible for diseases based on prior facts, clustering groups genes based on certain similarity measures into clusters that share common expression patterns. Unlike classification, clustering is effective in finding biologically significant groups of genes without any prior knowledge. These groups of genes involved in common functions and biological activities. However,

they are limited only to placing genes into disjoint groups that share certain characteristics. It has been observed that gene groups shares overlapping structures. Moreover, sometimes genes share similar expression patterns under a subset of given conditions. Biclustering[20] is an extension of classical clustering, that has been successfully used in finding groups of genes having similar expressions under a subset of conditions.

Association rule mining[21] is a relatively new and promising technique in the area of data mining and knowledge discovery. Association rule mining is a process that identifies links between sets of correlated objects in large datasets. Frequent itemset mining (we referred it simply as association mining in remaining of the thesis)is a sub-process of association rule mining technique, used to find relationship between the objects or items. Originally, the technique has been applied in market basket database and later extended to other application domains[22,23,24] including neuroinformatics[25]. However, not much work have been done so far to apply frequent mining or association mining in gene expression data analysis for finding gene regulatory networks or biclusters, with both positively and negatively regulated genes. Extension of classical association mining techniques for gene expression data analysis may suffer due to costly candidate generation phase and multi-pass nature of the techniques. Reducing the number of database passes and by removing the candidate generation phase may computationally improve gene expression data analysis based on association mining many folds.

In the next chapter, we present a new association mining technique called OPAM that needs only one pass over the database to generate all the frequent itemsets without any candidate generation.

Chapter 3

Association Mining Technique without Candidate Generation

This chapter presents an efficient One Pass Association Mining technique called **OPAM**, which finds all frequent itemsets without generating any candidate set. OPAM is an integration of two techniques: a correlogram matrix based technique to generate all frequent 1- and 2-itemsets in a single scan over the database and a technique that uses a vertical layout concept to generate the rest of the frequent itemsets. We experiment with several synthetic and real datasets and compare the performance of OPAM with competitors viz., Apriori and FP-growth and obtained satisfactory results.

3.1 Introduction

Association rules are of the form "80% of the customers who buy bread also buy butter". Association rules have numerous applications in real world, such as decision support, understanding customer behaviour, tele-communication alarm diagnosis and prediction. A formal definition of the association rule-mining problem is given by Agrawal[21] is as follows.

Definition 3.1.1 (Association Rule) : An association rule is an implication in the form of $X \Rightarrow Y$, where $X, Y \subset I$ are sets of items called itemsets, and

$X \cap Y = \phi$. X is called the antecedent while Y is called the consequent. The rule simply means that X implies Y.

Two basic measures called *support* and *confidence* and two corresponding thresholds, *minimum support* and *minimum confidence*, are used to measure the goodness of an association rule.

Definition 3.1.2 (Support) : The support of an association rule is defined as the percentage or fraction of records that contain $X \cup Y$ to the total number of records in the database. The count for each item is increased by one every time the item is encountered in a different transaction T in database D during the scanning process. Support is calculated as follows:

$$Support(X, Y) = \frac{|X \cup Y|}{|D|}. \tag{3.1}$$

Before the mining process, users can specify the minimum support as a threshold, meaning that they are interested only in association rules that are generated from itemsets whose support exceeds that threshold.

Definition 3.1.3 (Confidence) : Confidence of an association rule is defined as the percentage or fraction of the number of transactions that contain $X \cup Y$ to the total number of records that contain X. Confidence is calculated by the following equation:

$$Confidence(X, Y) = \frac{Support(X, Y)}{Support(X)}. \tag{3.2}$$

If the above percentage exceeds the threshold of minimum confidence, an interesting association rule $X \Rightarrow Y$ is generated. Confidence is a measure of strength of the association rule. The goal of association rule mining is to discover association rules that satisfy the predefined minimum support and confidence for a given database. A rule that satisfies both a minimum support threshold and a minimum confidence threshold is called a *Strong Rule*. The association rule-mining problem is usually decomposed into two sub-problems. One is to find itemsets whose occurrence frequency exceeds a predefined threshold in the database; such itemsets are called frequent or large itemsets. The second sub-problem is to generate

association rules from large itemsets with the constraint of minimal confidence. Since the second sub-problem is quite straightforward, most research focuses on the first sub-problem. The first sub-problem can be further divided into two sub-problems: generating candidate sets and generating frequent itemsets. Diagrammatic representation of the association mining technique is shown in Figure 3.1. The databases of interest are large and users are concerns only about items that are frequently purchased together (i.e., appear together in a database transaction). Usually thresholds of support and confidence are predefined by users to drop those rules that are not interesting or useful.

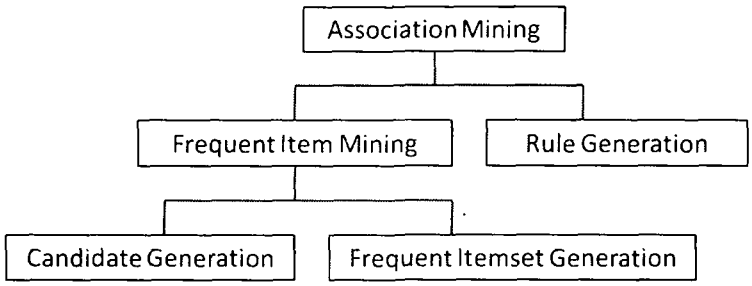


Figure 3.1: Various steps in association mining technique

3.2 Related Work

Association mining came into existence as market basket analysis on boolean datasets. In association mining, the size of databases are semi-large so that they can usually be accommodated in main memory. They are static in nature and sometimes referred as sequential association mining. Several efficient and improved sequential association mining techniques have been proposed throughout the last two decades[26]. Next, we discuss in brief some contributions in the area of the sequential association mining.

3.2.1 AIS

The AIS (Agrawal, Imielinski, Swami'93)[21] algorithm is the first algorithm proposed for mining association rules. During the first pass over the database, the

support count of each individual item is accumulated. Those items whose support counts are less than the support threshold are eliminated from the list of frequent items. From these frequent items, candidate 2-itemsets are generated by extending frequent items that occur with other items in the same transaction. To avoid generating the same itemsets repeatedly the items are ordered. Candidate itemsets are generated by joining a large item in the previous pass with another item in the transaction, which appears later than the last item in the frequent itemsets. To make this algorithm more efficient, an estimation method is introduced to prune itemset candidates that have no hope of becoming large. Consequently the unnecessary effort of counting such itemsets can be avoided. Since all candidate itemsets and frequent itemsets are assumed to be stored in main memory, memory management is necessary for AIS when memory is not enough. The main drawback of the AIS algorithm is that it generates too many candidate itemsets that finally turn out to be small, requiring wasted effort In addition, this algorithm requires too many passes over the whole database. In this algorithm only one item consequent association rules are generated, which means that the consequents of the rules contain only one item. For example we only generate rules like $X \cap Y \Rightarrow Z$ but not those rules as $X \Rightarrow Y \cap Z$.

3.2.2 Apriori

Among the popular algorithms to find large itemsets, the Apriori algorithm[27] stands at the top because of its simplicity and effectiveness. The basic property that characterizes a large itemset is that all subsets of a large itemset are large. The Apriori algorithm exploits this fact. The algorithm makes many passes over the data. Each pass starts with the seed set of large itemsets which are used to generate new potentially large itemsets called candidate itemsets. The support of each candidate itemset is found during a pass over the data and the actual large itemsets are determined. These large itemsets become the seed for the next pass. This process continues till no additional large itemsets are found. The algorithm uses a function, called $apriorigen(L_{k-1})$, which takes the set of all large $k-1$

support count of each individual item is accumulated. Those items whose support counts are less than the support threshold are eliminated from the list of frequent items. From these frequent items, candidate 2-itemsets are generated by extending frequent items that occur with other items in the same transaction. To avoid generating the same itemsets repeatedly the items are ordered. Candidate itemsets are generated by joining a large item in the previous pass with another item in the transaction, which appears later than the last item in the frequent itemsets. To make this algorithm more efficient, an estimation method is introduced to prune itemset candidates that have no hope of becoming large. Consequently the unnecessary effort of counting such itemsets can be avoided. Since all candidate itemsets and frequent itemsets are assumed to be stored in main memory, memory management is necessary for AIS when memory is not enough. The main drawback of the AIS algorithm is that it generates too many candidate itemsets that finally turn out to be small, requiring wasted effort In addition, this algorithm requires too many passes over the whole database. In this algorithm only one item consequent association rules are generated, which means that the consequents of the rules contain only one item. For example we only generate rules like $X \cap Y \Rightarrow Z$ but not those rules as $X \Rightarrow Y \cap Z$.

3.2.2 Apriori

Among the popular algorithms to find large itemsets, the Apriori algorithm[27] stands at the top because of its simplicity and effectiveness. The basic property that characterizes a large itemset is that all subsets of a large itemset are large. The Apriori algorithm exploits this fact. The algorithm makes many passes over the data. Each pass starts with the seed set of large itemsets which are used to generate new potentially large itemsets called candidate itemsets. The support of each candidate itemset is found during a pass over the data and the actual large itemsets are determined. These large itemsets become the seed for the next pass. This process continues till no additional large itemsets are found. The algorithm uses a function, called $apriorigen(L_{k-1})$, which takes the set of all large $k-1$

itemsets (L_{k-1}) as input and produces the candidates for large k-itemsets (L_k).

Despite its simplicity, the Apriori algorithm suffers from shortcomings. It is not scalable with the size of the database because it scans the database in each iteration to generate large itemsets. It produces a large number of candidate itemsets, out of which only a few are actually frequent itemsets. As a result, the ratio between the number of candidate large itemsets and the number of actual frequent itemsets becomes very high. The same technique is independently proposed by Mannila et al.[28]. Both works integrated later in[29].

3.2.3 SETM

The SETM algorithm[30] was motivated by the desire to use SQL to calculate large itemsets. In this algorithm each member of the set of large itemsets, L_k , is in the form $< TID, Itemset >$ where TID is the unique identifier of a transaction. Similarly, each member of the set of candidate itemsets, C_k, is in the form $< TID, Itemset >$. Similar to the AIS algorithm, the SETM algorithm makes multiple passes over the database. In the first pass, it counts the support of individual items and determines which of these are large or frequent in the database. Then, it generates the candidate itemsets by extending large itemsets from the previous pass. In addition, SETM remembers the TIDs of the generating transactions with the candidate itemsets. The relational merge-join operation can be used to generate candidate itemsets. The SETM algorithm saves a copy of the candidate itemsets together with TID of the generating transaction in a sequential manner. Afterwards, the candidate itemsets by sorted on itemsets, and small itemsets are deleted using an aggregation function. If the database is in sorted on the basis of TID, large itemsets contained in a transaction in the next pass are obtained by sorting L_k on TID. This way, several passes are made on the database. When no more large itemsets are found, the algorithm terminates. The main disadvantage of this algorithm is the number of candidate sets C_k. Since for each candidate itemset there is a associated TID, it requires more space to store a large number of TIDs. Furthermore, when the support of a candidate itemset is counted at the end of the

pass, C_k is not in ordered. Therefore, again sorting is needed on itemsets.

3.2.4 SEAR

SEAR (Sequential Efficient Association Rules) algorithm[31] is identical to Apriori, except that SEAR stores candidates in a prefix tree instead of a hash tree. In a prefix tree (also called a trie), each edge is labeled by items. Common prefixes are represented by tree branches, and unique suffixes are stored at the leaves. SEAR uses a pass-bundling optimization, where it generates candidates for multiple passes if the candidates fit in memory.

3.2.5 DHP

Shortly after the Apriori algorithm was published, Park et al. proposed another optimization algorithm, called DHP (Direct Hashing and Pruning) to reduce the number of candidate itemsets[32]. During the k^{th} iteration, when supports of all candidate k-itemsets are counted by scanning the database, DHP looks ahead and gathers information about candidate itemsets of size $k+1$ in such a way that all $(k+1)$-subsets of each transaction are stored in a hash table. Each bucket in the hash table consists of a counter to represent how many itemsets have been hashed to that bucket so far. When a candidate itemset of size $k+1$ is generated, the hash function is applied on that itemset. If the counter of the corresponding bucket in the hash table is below the minimal support threshold, the generated itemset is not added to the set of candidate itemsets. During the support counting phase of iteration k, every transaction is trimmed in the following way. If a transaction contains a frequent itemset of size $k+1$, any item contained in that $k+1$ itemset will appear in at least k of the candidate k-itemsets in C_k. As a result, an item in transaction T can be trimmed if it does not appear in at least k of the candidate k-itemsets in C_k. These techniques result in a significant decrease in the number of candidate itemsets that need to be counted, especially in the second iteration. Nevertheless, creating the hash tables and writing the adapted database to disk,

at every iteration, causes significant overhead.

3.2.6 Partitioning approach

The partition approach[33] divides the database into small partitions such that each partition can be handled in the main memory. Let the partitions of the database be $D_1, D_2, \cdots, D_p$. In the first scan, it finds local large itemsets in each partition D_i $(1 \leq i \leq p)$. A local large itemset, L_i, can be found by using an algorithm such as Apriori. Since each partition can fit in the main memory, there is no additional disk I/O for a partition after the partition is loaded into the main memory. In the second scan, it uses the property that a large itemset in the whole database must be locally large in at least one partition of the database. The union of the local large itemsets found in each partition is used as candidates and are counted through the whole database to find all the large itemsets.

3.2.7 Sampling

Sampling[34] reduces the number of database scans to one in the best case and two in the worst. A sample which can fit in main memory is first drawn from the database. The set of large itemsets in the sample is then found from this sample using Apriori. Let the set of large itemsets in the sample be PL, which is used as a set of probable large itemsets and used to generate candidates which are to be verified against the whole database. The candidates are generated by applying the negative border function, $\bar{BD}$, to PL. Thus the candidates are $\bar{BD}(PL) \cup PL$. The negative border of a set of itemsets PL is the minimal set of itemsets which are not in PL, but all their subsets are. After the candidates are generated, the whole database is scanned once to determine the counts of the candidates. If all large itemsets are in PL, i.e., no itemsets in $\bar{BD}(PL)$ turn out to be large, all large itemsets are found and the algorithm terminates. Otherwise, there are misses in $\bar{BD}(PL)$; some new candidate itemsets must be counted to ensure that all large itemsets are found, and thus one more scan is needed. In this case, $L \cap PL \neq \phi$,

and the candidate itemsets in the first scan may not contain all candidate itemsets of Apriori.

3.2.8 DIC

DIC (Dynamic Itemset Counting)[35] initially identifies certain 'stops' in the database. It is assumed that we read the records sequentially as we do in other algorithms, but pause to carry out certain computations at the 'stop' points. It defines four different structures: Dashed Box, Dashed Circle, Solid Box, Solid Circle. Each of these structures maintains a list of itemsets. Itemsets in the 'dashed' category of structures have a counter and the stop number with them. The counter is to track of the support value of the corresponding itemsets. The stop number is to keep track whether an itemset has completed one full pass over a database. The itemsets in the 'solid' category structures are not subjected to any counting. The itemset in the solid box is the confirmed set of frequent sets. The itemsets in the solid circle are the confirmed set of infrequent sets. The algorithm counts the support count of the itemsets in the dashed structure as it moves along from one stop point to another. During the execution of the algorithm, at any stop point, the following events take place,

- Certain itemsets in the dashed circle move into the dashed box. These are the itemsets whose support-counts reach minimum threshold value during this iteration.

- Certain itemsets enter afresh into the system and get into the dashed circle. These are essentially the supersets of the itemsets that move from the dashed circle to the dashed box.

- The itemsets that have completed one full pass, move from the dashed structure to the solid structure. That is, if the itemset is in a dashed circle while completing a full pass, it moves to the solid circle. If it is in the dashed box, it moves into the solid box after completing a full pass.

Though this method drastically reduces the number of scans of the database, its performance is heavily dependent on the distribution of the data.

3.2.9 FP-growth

FP-growth[36] finds frequent itemsets without candidate generation. The algorithm is based on a special data structure called FP-tree, which is a prefix tree of the transactions of the database such that each path represents a set of transactions that share the same prefix. The algorithm works as follows. The algorithm first scans the database once to find the one-element frequent itemsets in the database. Infrequent items are removed from the database and items in the transactions are rearranged in the descending order of the frequencies of items. Then, all transactions containing the least frequent item are selected and the item is removed from the transactions, resulting in a reduced (projected) database. This projected database is processed to find frequent itemsets. Obviously, the removed item is prefix of all frequent itemsets. The item is removed from the database and the process is repeated with the next least frequent item. It is to be noted that FP-tree contains all the information about the transactions and the frequent itemsets. So, to find any information about the transactions and frequent itemsets, one needs to just search the tree. FP-Growth is one of the fastest frequent itemsets finding algorithms. It is robust enough to find the complete set of frequent itemsets. Although the algorithm has many advantages, it suffers from two significant disadvantages.

1. The time taken to construct the FP-tree is quite large, particularly when the dimensionality is large.

2. With the decrease in minimum support threshold value, its performance degrades and at certain instance of time it becomes almost similar to Apriori.

A summary of the algorithms discussed above is given in Table 3.1.

Table 3.1: Characteristics of different Sequential AM Techniques

Algorithm	Database Layout	Data Structure	No. of DB Scan	Candidate generation
AIS	Horizontal	None	K	Yes
Apriori	Horizontal	Hash Tree	K	Yes
SETM	Horizontal	None	K	Yes
SEAR	Horizontal	Prefix Tree	K	Yes
DHP	Horizontal	Hash Tree	K	Yes
Partitioning	Vertical	None	2	Yes
Sampling	Horizontal	None	2	Yes
DIC	Horizontal	Tries	$\leq$ K	Yes
FP Growth	Horizontal	FP-Tree	2	No

(K: size of the longest frequent itemset, DB: database.)

3.3 Motivation

All these algorithms are based on candidate generation and suffer from the following two major problems.

1. They need to scan the database multiple times, which is costly, particularly when the database is very large.

2. They generate huge candidate sets in comparison to the actual frequent itemsets.

So, it is desirable to develop an algorithm, which not only obviates scanning the database repeatedly but also does not generate candidate sets. We present an one-pass association mining technique that addresses this problem by introducing an integrated approach to find the frequent itemsets.

3.4 OPAM: One Pass Association Mining Technique

OPAM adopts an integrated approach to solve the frequent itemset finding problem in a single pass over the database. Initially, it attempts to generate all frequent 1- and 2-itemsets directly using a *correlogram matrix* based technique. In the next

Table 3.2: Sample market basket dataset

Transaction Id	Item Purchased
T1	I1,I2,I4,I5,I6
T2	I2,I4
T3	I2,I3,I6
T4	I1,I2,I4,I5
T5	I1,I3,I6
T6	I2,I3
T7	I1,I3
T8	I1,I2,I3,I4,I5

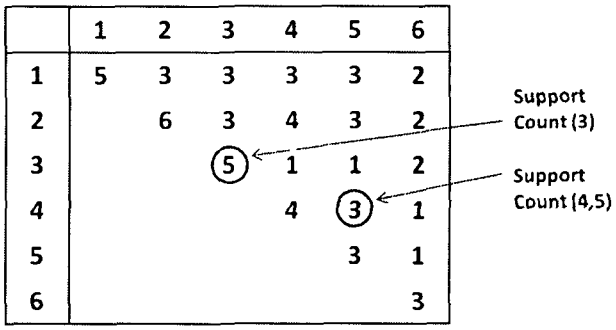


Figure 3.2: Correlogram matrix showing support counts of itemsets

phase, to find the remaining higher order frequent itemsets, it exploits a vertical layout concept for the database. Next, we provide the background of each of these techniques.

3.4.1 Correlogram matrix based technique

Correlogram matrix is a co-occurrence frequency matrix. It is a matrix of size, $N \times (N+1)/2$, for a transaction database with N items. Each cell of the matrix contains the frequency of co-occurrence of an item pair. Item pairs are specified by the row index and the column index of the cell.

For example, to specify the frequency of co-occurrence of item pair {4,5}, corresponding to the sample market basket dataset depicted in Table 3.2, the content of the cell (4,5) in the correlogram matrix (see Figure 3.2) with an index of row 4 and column 5 will indicate the co-occurrence frequency of the item pairs {4,5}. On the other hand, a cell for which, the row and column indices are the same, specifies the occurrence frequency of a single item. Thus, seen in Figure 3.2, the cell (3,3) indicates the occurrence frequency of the single itemset {3}.

3.4.2 Construction of correlogram matrix

The correlogram matrix is constructed by a single scan of the database. In order to construct the correlogram matrix, we model the situation graph theoretically. All

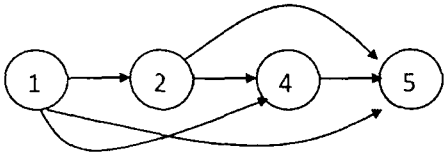


Figure 3.3: Item nodes forming directed graph

	1	2	3	4	5	6
1	1	1	0	1	1	0
2		1	0	1	1	0
3			0	0	0	0
4				1	1	0
5					1	0
6						0

Figure 3.4: Correlogram matrix showing post increment scenario

items participating in a particular transaction are considered nodes. As items appear in the transaction in a lexicographical order, we say that they form a directed graph involving all items as node of the graph. Each item is linked by a single link or edge. Thus, only a directional path exists between any two nodes. To illustrate, let us consider sample market basket dataset given in Table 3.2. Items I1, I2, I4 and I5 participate in transaction T4. Thus, they form a directed graph as shown in the Figure 3.3.

To count the co-occurrence frequency of all items participating in a particular transaction, we count links among all pairs of nodes and correspondingly increment the content of a cell with the corresponding indices. Thus, if we consider the example in Figure 3.3, we increment the contents of cells (1,2), (1,4), (1,5), (2,4) and (2,5) We also increment the count of first node of a pair. For example, when incrementing the count for the pair (1,2), we also increment the content of the cell (1,1) for storing the frequency of item I1. The scenario after incrementing the content of correlogram matrix becomes the one shown in Figure 3.4. Thus by following the procedure discussed above, one can construct the correlogram matrix by scanning the database only once. From the correlogram matrix, we can extract the frequent 1- and 2-itemsets with a given minimum support threshold in a straightforward manner.

The advantages of this technique are as follows.

Table 3.3: Vertical layout of sample market basket data

Item Purchased	Transaction Id
I1	T1,T4,T5,T7,T8
I2	T1,T2,T3,T4,T6,T8
I3	T3,T5,T6,T7,T8
I4	T1,T2,T4,T8
I5	T1,T4,T8
I6	T1,T3,T5

1. Candidate generation step is no more required to find 1- and 2- frequent itemsets.

2. Unlike other algorithms, it require only one scan over the database for finding all the frequent 1- and 2-elements itemset.

3. Since it is memory based, it is fast.

3.4.3 Mining frequent itemsets using vertical transaction layout

Transaction layout is a method that can be used to format items in a transaction database. Currently, there are three approaches: *horizontal*[21], *vertical*[37] and the *hybrid*[38]. Horizontal layout combines items in a transaction row-wise. This layout suffers from the problem of superfluous processing since there is no index on the items. In a vertical layout, each item is associated with a column of values representing the transaction in which it is present. A vertical layout creates an index on the items and reduces the effect of large data sizes since there is no need to rescan the whole database each time. The technique we propose adopts the vertical layout approach for mining the remaining higher order frequent itemsets.

To take the advantages of the vertical format, we transform our database into vertical form for mining frequent itemsets. The corresponding vertical layout of the sample market basket data given in Table 3.2 is presented in Table 3.3.

Once we create the vertical layout of the original transaction database, the next

Table 3.4: 2-element frequent item sets

Item Set	Transaction List
{I1,I2}	T1,T4,T8
{I1,I3}	T3,T5,T8
{I1,I4}	T1,T4,T8
{I1,I5}	T1,T4,T8
{I2,I3}	T3,T6,T8
{I2,I4}	T1,T2,T4,T8
{I2,I5}	T1,T4,T8
{I4,I5}	T1,T4,T8

Table 3.5: 3-element frequent item sets

Item Set	Transaction List
{I1,I2,I4}	T1,T4,T8
{I1,I2,I5}	T1,T4,T8
{I1,I4,I5}	T1,T4,T8
{I2,I4,I5}	T1,T4,T8

Table 3.6: Largest frequent item sets

Item Set	Transaction List
{I1,I2,I4,I5}	T1,T4,T8

phase is straightforward. We intersect two item records from the vertical table. If the resultant record contains number of transaction IDs greater than or equal to a given minimum support threshold, the item pairs in the intersection form the frequent itemset. Support counting is performed simply by counting the number of transaction IDs that are common in both item records in the intersection. We term such an intersection as a successful intersection.

To avoid unnecessary computation of intersection we use the same union and prune step as used in Apriori algorithm. We intersect two records if both the target itemsets pass through the union and pruning steps successfully. As the possible number of 2-element itemsets are huge, to avoid working with them directly, we use correlogram matrix as introduced in previous section to find all frequent 1- and 2-element itemsets directly. After generating all the 1- and 2-element frequent sets using the correlogram matrix, we simply update the vertical table by eliminating all the records corresponding to non-frequent itemset of size one. Next, we perform intersection among the pair of records from 1-element frequent set, which are actually in frequent 2-element itemset. The intersection of transaction records then

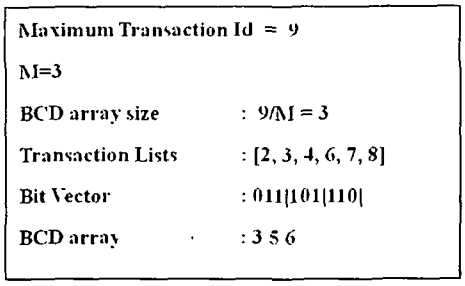


Figure 3.5: Illustration of BCD scheme

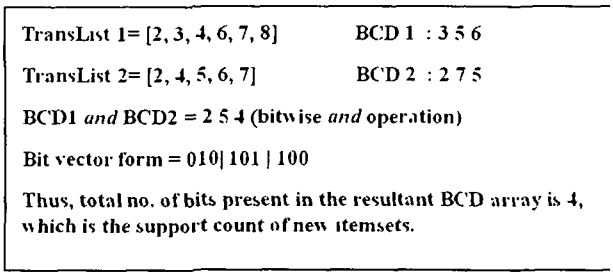


Figure 3.6: Illustration of intersection and support counting method

continue until no successful intersection is possible. For illustration, intermediate results during iterations to obtain all the frequent itemsets from market basket in Table 3.2 with minimum support threshold 3 are shown in tables 3.4, 3.5, 3.6.

3.4.4 Proposed algorithm and its implementation issues

This section presents the algorithm for the proposed integrated approach (see Algorithm 1). It also discusses some of the issues related to efficient implementation of the algorithm. The algorithm accepts the market-basket database D and minimum support σ, as input and it generates all the frequent itemsets as output. Steps 1 to 4 of the algorithm are dedicated to the first phase of the approach, i.e., finding of 1- and 2-element frequent itemsets using correlogram matrix of the original database. After step 4, we get an alternative representation of the database as discussed above in the second phase of the approach. Generally, compared to the number of transactions the numbers of items or dimensions are relatively much lower. Thus such a vertical database can be easily stored in main memory. However, if the number of transactions are very large, it becomes very difficult to store such transaction list in main memory. To handle such cases, we use a compact representation of the transaction list.

At line 9, the union operation returns the new itemset if union is possible, otherwise it returns null. Following downward closure property, pruning operation returns false if all the subsets of the new itemset generated by union operation, are frequent. In step 12, the intersection between two item record sets are carried out.

```
   input  : D (Original Dataset), σ (Minimum Support)
   output: L (List of frequent itemsets)
 1 Generate Correlogram Matrix M from D;
 2 Construct vertical database V from D;
 3 Traverse the M to generate one and two element frequent itemsets ;
 4 Write all the one and two frequent itemsets to L;
 5 Update V with frequent two element itemsets ;
 6 while successful intersection possible do
 7 |   for i ← 1 to |V| do
 8 |   |   for j ← i + 1 to |V| do
 9 |   |   |   NewItemSet =Union (V [i].ItemSet, V [j].ItemSet) ;
10 |   |   |   if NewItemSet ≠ Null then
11 |   |   |   |   if Pruning (NewItemSet)=False then
12 |   |   |   |   |   NewTransList =Intersection (V [i].TransList, V
   |   |   |   |   |   [j].TransList) ;
13 |   |   |   |   |   if Count (NewTransList)≥ σ then
14 |   |   |   |   |   |   Write the NewItemSet and NewTransList into L;
15 |   |   |   |   |   |   Update V;
16 |   |   |   |   |   end
17 |   |   |   |   end
18 |   |   |   end
19 |   |   end
20 |   end
21 end
```

Algorithm 1: OPAM:The Algorithm

To perform intersection, we apply simple bit wise *and* operation. It is very fast compare to normal intersection, which is performed by comparing elements from both participating records. In line 13, *Count* returns support count of the new itemset. All the itemsets satisfy minimum support criteria are stored in the list L. The vertical database V is updated by eliminating all the no-frequent itemsets. The process of intersection continues with new records until no successful intersection is possible.

For compact representation of transaction list, we adopt the concept used in creating binary coded decimal (BCD) representation for integers. OPAM initially stores the transaction list associated with a itemset record in a bit vector. It is then converted into BCD of M bit size. Thus, if the maximum transaction ID is T, it requires T/M sized array of a data type that can accommodate M bit data.

For example, let us consider a transaction that has maximum transaction ID i.e. $T = 9$ and a BCD scheme of 3 bits (i.e., M). For a transaction list $\{2, 3, 4, 6, 7, 8\}$, the compact BCD representation is shown in Figure 3.5. After converting the item records into BCD form, one can easily count the support as shown in Figure 3.6.

Intersection between two BCD array is performed through bitwise *and* operation. It is very fast and effective. The interesting fact is that as the iteration moves to a higher level, the number of the transaction IDs per records goes down. The number of records also gradually decreases. This is because of the fact that the higher order frequent itemsets (itemsets of size 3 or more) are normally fewer compared to possible lower order itemsets. Thus, the performance gradually increases and consumption of memory space decreases.

3.5 Analysis of Our Algorithm

Here, we present proof of correctness and completeness of OPAM and then we analyse our algorithm in terms of computational complexity.

3.5.1 Completeness and correctness

Lemma 3.5.1. *Correlogram matrix based technique generates all the 2-element itemsets which are frequent w.r.t. minimum support (minsup), a user defined threshold.*

Proof. Correlogram matrix based technique computes support counts of all the 2-element itemsets by using exhaustive search in the transaction database. Next, it extracts only those 2-element itemsets which satisfy the *minsup* condition. Hence, the proof. □

Lemma 3.5.2. *OPAM is complete, i.e., OPAM extracts all the frequent itemsets w.r.t. minsup.*

Proof. It can be proved in two steps. First, the 2-element frequent itemsets generated is complete, which is evident from Lemma 3.5.1. Second, OPAM generates

all those itemsets of size > 2 based on the output of first step and their support counts, satisfy the *minsup* condition. Similarly, it is true for any itemsets of size $\geq k$. Thus OPAM generates all the frequent itemsets which satisfy the *minsup* condition and hence the proof.

□

Lemma 3.5.3. *OPAM is correct, i.e., frequent itemset generated by OPAM satisfy min-sup criterion.*

Proof. This lemma can be proved by contradiction. Let us assume that an itemset I_{k+1} is frequent, generated by intersecting itemset X_k, with transaction record T_x, where cardinality of T_x is above or equal to *minsup*, i.e., $|T_x| \geq minsup$ and itemset Y_k, with $|T_y| < minsup$. Since $|T_y| < minsup$, resulting intersection between X_k and Y_k never satisfy minimum support criterion, i.e., $|T_x \cap T_y| \ngeq minsup$, which contradicts the assumption, hence the proof. □

3.5.2 Complexity analysis

Below we present analysis of OPAM in terms of space and time complexity.

3.5.2.1 Space complexity

OPAM requires space for correlogram matrix and transaction records of all the itemsets in each iteration. Thus, space complexity for the two data structures can be calculated as follows.

a) Space for correlogram matrix: For a transaction database with N items, the fixed space requirement for correlogram matrix is:

$$\begin{aligned} SPACE_{CM} &= O(N * (N+1)/2) \\ &\approx O(N^2/2). \end{aligned}$$

b) Space for frequent itemset: Assume that k is the number of frequent itemsets

in each iteration. T is the number of transactions in the database. If we consider, M as the bit size for BCD scheme, the space required in each level is:

$$SPACE_{FI} = O(k * (T/M))$$

The value of k is normally very high in case of two element frequent itemsets and it decreases with the increase in iteration level.

It is worth mentioning that requirement of both the data structures is not simultaneous. Correlogram matrix is needed at the early stage of the algorithm. Once vertical layout is constructed based on two element frequent itemsets, correlogram matrix can be deleted from the memory.

3.5.2.2 Time complexity

We compute the time complexity based on three different computational costs.

a) Construction of correlogram matrix: Assume that the database contains T transactions and a maximum of N items in each transaction. For storing and updating support count of item pairs in the correlogram matrix with respect to each transaction, it requires $O(T * N^2)$ time. The time requirements for accessing the correlogram matrix is $(N \times (N+1)/2) \approx N^2$. The cost for construction as well as to find the 2-element frequent itemsets from the correlogram matrix is $O(T * N^2)$ $+O(N^2)$, which become $O(T * N^2)$.

b) Construction of vertical layout: To represent each itemset with transaction list using BCD scheme, it requires to process each transaction ID from the list. Thus for N items and T transactions, the complexity is $C_{Ver} = O(T * N)$.

c) Cost of intersection: Assume that there are k frequent itemsets in each iteration and ξ is the maximum level of iteration. For each iteration it considers all pair of items for intersection. Since we are using bitwise AND operation for

intersection, it is very fast as compared to other operations and thus ignored. The cost incurred for intersection is $C_{Int} = O(\xi * k^2)$

The total cost of OPAM is

$$\begin{aligned} Cost_{OPAM} &= C_{CM} + C_{Ver} + C_{Int} \\ &= O(T * N^2) + O(T * N) + O(\xi * k^2). \end{aligned}$$

Since, we are reading T transactions in a single scan over the database, thus the performance of OPAM mostly depends on number of items, N, in the database.

3.6 Performance Evaluation

To evaluate the performance of OPAM in comparison to other techniques, we use two popular techniques, Apriori and FP-growth. We implemented OPAM using Java 1.6 on Windows 7 platform running in 2.53 GHz machine. For other two algorithms, Apriori and FP-growth, we used Java based SPMF[a] tool. SPMF (Sequential Pattern Mining Framework) is an open-source data mining platform written in Java and distributed under the GPL v3 license.

3.6.1 Dataset used

We have generated synthetic datasets according to the specifications given in Table 3.7. The synthetic datasets were created with the data generator in *ARMiner* [b] software, which follows the basic spirit of well-known IBM synthetic data generator[c] for association rule mining. The size of the data (i.e., number of transactions), the number of items and the number of unique patterns (incase of synthetic dataset) in the transactions are the major parameters in data generation. We also used

[a] http://www.philippe-fournier-viger.com
[b] *http://www.cs.umb.edu/laur/ARMiner/*
[c] *www.almaden.ibm.com/cs/quest/*

two real life Mushroom[a] and Chess dataset taken from FIMI[b]. Below we present experimental results that use these synthetic and real datasets.

Table 3.7: Details of Transaction Dataset

Data Set	No.of Transactions	No. of Items	Avg. size of Transaction	No. of Pattens
T10I400D100K	100,000	400	10	20
T10I600D100K	100,000	600	10	20
T10I800D100K	100,000	800	10	20
Mushroom	8124	128	-	-
Chess	3196	75	-	-

3.6.2 Experimental results

Performance of the three algorithms is compared in terms of execution time for different minimum support values. For different synthetic datasets the performance of the three algorithms degrades gradually along with decrease in the minimum support value. Apriori always needs very high computational time for all the datasets. Compare to other two algorithms, the performance of OPAM is found effective, especially in synthetic datasets. However, in case of real datasets, FP-growth performs well compare to other algorithms, especially when data is dense. From the Figure 3.7 and 3.8, it can be observed that FP-growth always needs certain computational time even when number of frequent items are zero (when minimum support is high). This is required because of the minimum time needed to construct the tree. For the same situation, OPAM also requires minimum time for scanning the database once and constructing the correlogram matrix.

3.7 Discussion

In this chapter, we have presented an efficient frequent itemset finding technique. The technique works in two phases, a correlogram matrix technique to generate

[a] *http://www.ics.uci.edu/~mlearn/MLRRepository.html*
[b] *http://fimi.ua.ac.be*

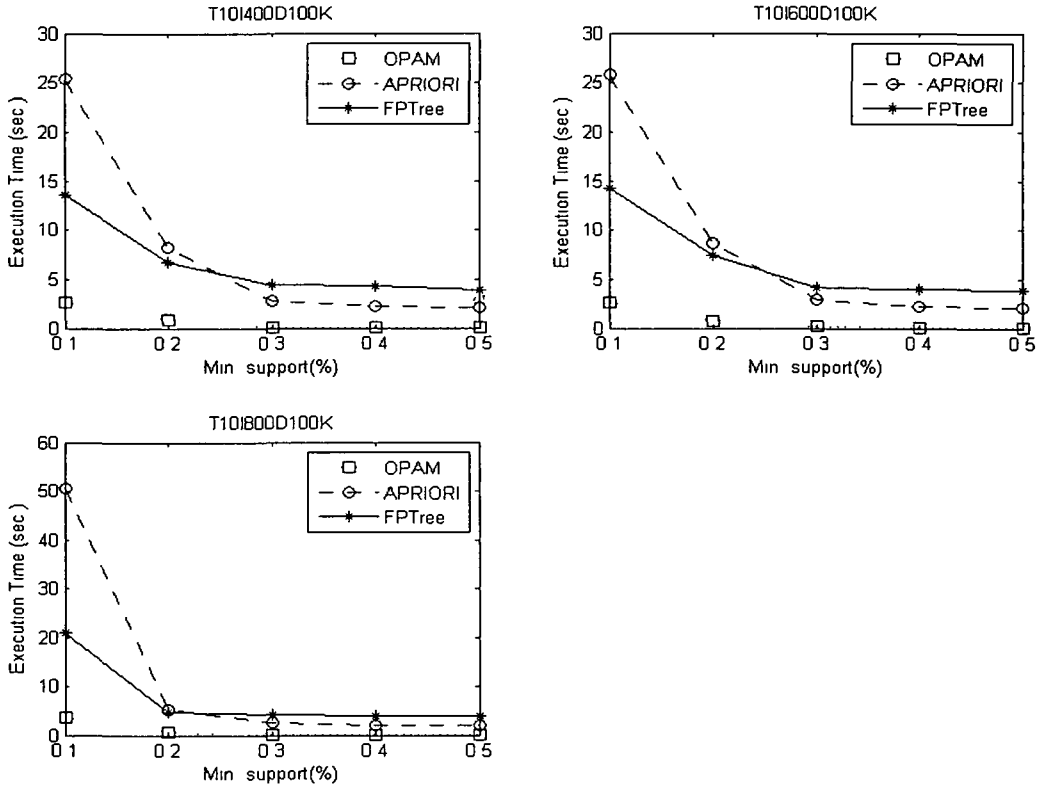


Figure 3.7: Performance comparison on synthetic dataset

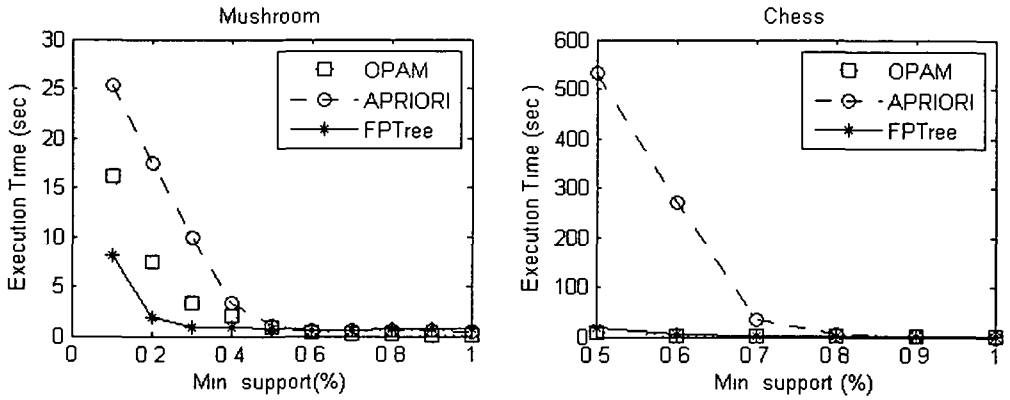


Figure 3.8: Comparison against execution time vs. minimum support on real data

those 1- and 2-element frequent itemset, and a vertical layout technique to generate higher order frequent itemsets. The technique is able to generate all frequent itemsets in one scan of the database. Another advantage of OPAM is that it also supports interactive mining of 1- and 2-element itemsets. Experiments have shown that OPAM performs well in comparison to Apriori and FP-growth algorithms.

We have explored the advantage of the correlogram matrix for extracting 1- and 2- element frequent itemsets in finding strongly correlated item pairs from a transaction database. This is discussed in the next chapter.

Chapter 4

Finding Strongly Correlated Item Pairs in Large Transaction Databases

Correlation mining is an approach that allows one to draw statistical relationships among items from transaction data. Existing techniques either generate large number of candidates or build huge trees and require multiple passes over the database. This chapter presents an effective and fast Strongly COrrelated Pairs Extraction technique called **SCOPE**, and its extension to extract k most strongly correlated pairs from large transaction databases. Many existing techniques use Pearson's correlation coefficient as a measure of correlation, which may not always perform well when data is noisy and binary. As an alternative to Pearson's correlation coefficient, we present a method of computing Spearman's rank order correlation coefficient from the transaction data. We find that the proposed technique performs satisfactorily in terms of execution time when tested with several real and synthetic datasets compared to other similar techniques.

4.1 Introduction

Starting from market basket data analysis, association mining is now applied in a wide variety of domains such as machine learning, soft-computing and computational biology. Standard association mining technique extracts all subset of items satisfying a minimum support constraint. The traditional association rule mining technique[21,39] is based on a support-confidence framework. However, the support-confidence framework can be misleading; it can identify a rule $(A \Rightarrow B)$ as interesting (strong) when in fact, the occurrence of A might not imply the occurrence of B. Thus, the support and confidence measures are insufficient in filtering out uninteresting association rules[39,40]. It has been observed that item pairs with high support value may not imply high correlation. Similarly, a highly correlated item pair may exhibit low support value. To tackle this weakness, correlation analysis can be used to provide an alternative framework for finding statistically interesting relationships [40]. It also improves the understanding of some association rules. In statistics, relationships among nominal variables can be analyzed with nominal measures of association such as Pearson's correlation coefficient and measures based on Chi Square[41]. The ϕ correlation coefficient[41] is a computational form of Pearson's correlation coefficient for binary variables. An equivalent support measure based ϕ correlation coefficient computation technique is introduced in[42,43] to find correlation of item pairs in a transaction database based on their support count. For any two items X and Y in a transaction database, the support based ϕ correlation coefficient can be calculated as:

$$\phi(X,Y) = \frac{Sup(X,Y) - Sup(X) * Sup(Y)}{\sqrt{Sup(X) * Sup(Y) * (1 - Sup(X)) * (1 - Sup(Y))}} \tag{4.1}$$

where $Sup(X)$, $Sup(Y)$ and $Sup(X,Y)$ are the individual supports and the joint support of item X, Y, respectively.

Unlike traditional association mining, the all-pair-strongly correlated query is to find statistical relationships among pair of items from a transaction database. The problem can be defined as follows.

Definition 4.1.1 (Strongly correlated pair) : Assume a market basket database D with T transactions and N items. Each transaction, T is a subset of I, where $I = \{X_1, X_2, \ldots, X_N\}$ is a set of N distinct items. Given a user-specified minimum correlation threshold θ, an all-strong-pairs correlation query (SC) finds a set of all item pairs (X_i, X_j) (for $i, j = 1 \ldots N$) with correlation, $Corr(X_i, X_j)$, above the threshold θ. Formally, it can be defined as:

$$SC(D, \theta) = \{\{X_i, X_j\} | \{X_i, X_j\} \subseteq I, X_i \neq X_j \wedge Corr(X_i, X_j) \geq \theta\}. \qquad (4.2)$$

Besides providing a statistical meaning for the traditional association mining problem, correlation mining can play a major role in addressing various issues such as how sales of a product are associated with sales of other products, which in turn may help in designing sales promotion, catalog design and store layout. Correlation mining can be helpful in efficient finding of co-citations and term co-occurrences during document analysis. Functional relationship[44] among pairs of genes based on gene expression profile and changes in functional relationship in different diseases and conditions may be indicative of disease mechanism for diseases like cancer. It has been observed that a simple pair-wise correlation analysis may be helpful in revealing new gene-gene relationship[45,46], which again in turn are useful in discovering gene regulatory pathways or gene interaction networks.

To determine the appropriate value of θ, a prior knowledge of data distribution is required. Without specific knowledge of the target data, users will have difficulty in setting the correlation threshold to obtain required results. If the correlation threshold is set too large, there may be only a small number of results or even no result. In such a case, the user may have to guess a smaller threshold and perform the mining again, which may or may not give better result. If the threshold is too small, there may be too many results for the user; too many results need an exceedingly long time to compute, and also extra effort to filter the answers.

An alternative solution to this problem could be to change the task of mining

correlated item pairs under pre-specified threshold to mine top-k strongly correlated item pairs from transaction database, where k is the desired number of item pairs that have largest correlation values. Recently the idea of top-k strongly correlated pairs has been applied in graph databases to find top-k frequent correlated subgraph[47]. The top-k correlated- pairs query problem in market-basket can be defined as follows.

Definition 4.1.2 (Top-k strongly correlated pairs) : Given a user-specified threshold k and a market basket database D of T transactions where each transaction T_i is a subset of I (set of N distinct items), a top-k correlated-pair query, $TopK(D,k)$, finds list of k top most item pairs based on correlation coefficient value. Thus, $TopK(D,k)$ can be represented as follows:

$$TopK(D,k) = \{\{X_{i1}, X_{j1}\}, \{X_{i2}, X_{j2}\}, \ldots, \{X_{ik}, X_{jk}\}\} \tag{4.3}$$

where, $Corr(X_{i1}, X_{j1}) \geq Corr(X_{i2}, X_{j2}) \geq \ldots \geq Corr(X_{ik}, X_{jk})$, for all $\{X_{ik}, X_{jk}\} \subseteq I$ and $X_{ik} \neq X_{jk}$

Thus, top-k is a sorted list of k item pairs based on any suitable correlation coefficient, $Corr$.

4.1.1 Computing support based correlated pairs: an illustration

The task of strongly correlated item pair finding generates a list of pairs from the database where $Corr$ value of a pair is greater than the user specified θ. Similarly, the task of top-k correlated-pair finding generates a sorted list of k pairs in the order of $Corr$ from the database. An illustration of both correlated-pairs query problems is given in Figures 4.1 and 4.2. In the example, we denote $Corr$ as correlation coefficient. The input to the strongly correlated query is a market basket database containing 8 transactions and 6 items. The value of θ is set to 0.05. Similarly, for

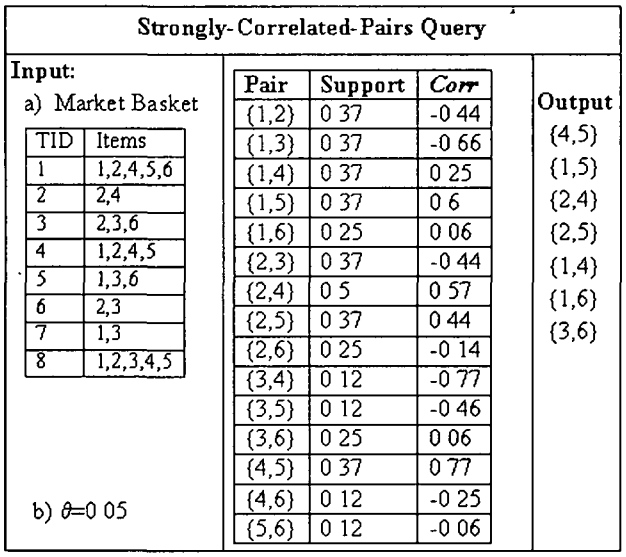


Figure 4.1: Illustration of Strongly Correlated Pairs Query Problem

the top-k problem the value of k is set to 8. Since the database has six items, there are $\binom{6}{2} = 15$ item pairs for which correlation coefficient ϕ is calculated. To compute $\phi(4,5)$ using Equation (4.1), we need the single element supports $Sup(4) = 4/8$ and $Sup(5) = 3/8$, and joint support $Sup(4,5)$=3/8, to compute correlation coefficient, $\phi(4,5)$, which is 0.77. Finally, all pairs that satisfy θ constraint are extracted, and the list of strongly correlated pairs is generated as output. Similarly, the list of k most strongly correlated pairs is generated as an output for the second problem (irrespective of any θ value).

4.2 Related Work

We now discuss some of the state-of-the-art approaches towards finding strongly correlated item pairs and top k strongly correlated item pairs from transaction database.

4.2.1 TAPER

TAPER[42,43] is a candidate generation based technique for finding all strongly correlated item pairs. It consists of two steps: filtering and refinement. In the filtering

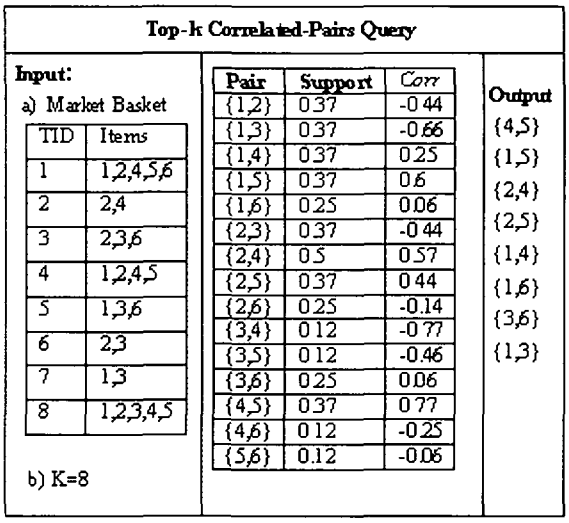


Figure 4.2: Illustration of Top-k Correlated Pairs Query Problem.

step, it applies two pruning techniques. The first technique uses an upper bound of the ϕ correlation coefficient as a coarse filter. The upper bound $upper(\phi(X,Y))$ of ϕ correlation coefficient for (X,Y) is:

$$\phi(X,Y) \leq upper(\phi(X,Y)) = \sqrt{\frac{sup(Y)}{sup(X)}}\sqrt{\frac{1-sup(X)}{1-sup(Y)}}. \tag{4.4}$$

If the upper bound of the ϕ correlation coefficient for an item pair is less than the user-specified correlation threshold θ, the item pair is pruned right away. To minimize the effort to compute upper bounds of all possible item pairs, TAPER applies the second pruning technique based on the conditional monotone property (1-D) of the upper bound of the ϕ correlation coefficient. For an item pair (X,Y), if the upper bound is less than θ, all item pairs involving item X and rest of the target items having support less than Y will also give upper bound less than θ. In other words, for item pair X,Y, if $sup(X) > sup(Y)$ and we fix item X, the upper bounds $upper(\phi(X,Y))$, is monotone decreasing with decreasing support of item Y. Based on this 1-D monotone property, straightaway one can avoid computation of upper bound for other items. In the refinement step, TAPER computes the exact correlation for each surviving pair and retrieves the pairs with correlation above θ.

It is understood that in comparison with single element item sets, usually the

two element candidate sets are huge. The upper bound based pruning technique is very effective in eliminating large numbers of item pairs during the candidate generation phase. However, when the database contains a large number of items and transactions, testing even the remaining candidate pairs is expensive.

4.2.2 Tcp

FP-tree[36] based technique, Tcp[48] is a milestone in strongly correlated item pair extraction, that overcome the bottlenecks of TAPER. Strongly correlated item pairs are generated without any candidate generation. Tcp includes two sub processes: (i) construction of the FP-tree, and (ii) computation of correlation coefficient of each item pair using the support count from the FP-tree and extraction of all strongly correlated item pairs with correlation greater than θ. The efficiency of the FP-tree algorithm can be justified as follows: (i) The FP-tree is a compressed representation of the original database, (ii) the algorithm scans the database twice only, and (iii) the support value of all item pairs is available in the FP-tree.

Although the algorithm is based on an efficient FP-tree data structure, it suffers from the following two significant disadvantages.

1. Tcp constructs the entire FP-tree with an initial support threshold of zero. The time taken to construct such an FP-tree is quite large, especially when the dimensions are large.

2. Moreover, it requires a large amount of space to store the entire FP-tree in the memory, particularly when the number of items is very large.

Below we discuss some of the top-k correlated pair finding techniques. Almost all the techniques proposed so far are minor extensions of strongly correlated pair finding techniques.

4.2.3 TOP-COP

TOP-COP[49] is an upper bound based algorithm for finding top-k strongly correlated item pairs and is an extended version of TAPER. TOP-COP exploits a 2-D

monotone property of the upper bound of ϕ correlation coefficient for pruning non-potential item pairs, i.e., pairs which do not satisfy the correlation threshold θ. The 2-D monotone property is as follows: For a pair of items X, Y, if $sup(X) > sup(Y)$ and we fix item Y, $upper(\phi(X, Y))$ is monotone increasing with decreasing support of item X. Based on the 2-D monotone property a diagonal traversal technique, combined with a refine-and filter strategy is used to efficiently mine top-k strongly correlated pairs.

Like TAPER, TOP-COP is also a candidate generation based technique. The 1-D monotone property, used in TAPER provides a one dimensional pruning window for eliminating non-potential item pairs. Moving one step further, TOP-COP exploits the 2-D monotone property, which helps further in eliminating non-potential pairs from two dimensions instead of one dimension. Compared to 1-D monotone based pruning, the 2-D pruning technique is more effective in eliminating a large number of item pairs during the candidate generation phase. Like TAPER, TOP-COP also starts with a sorted list of items based on support in non-increasing order, which needs a scan of the database once for creating such a list. Since it is a candidate generation based technique and has structural similarity with TAPER, it also suffers from the drawback of expensive testing of remaining candidates after pruning and filtering steps.

4.2.4 Tkcp

Tkcp[50], is an extension of Tcp to extract top-k strongly correlated item pairs using an FP-tree[36] based approach. Tkcp also includes two sub processes: *(a)* construction of the FP-tree, and *(b)* computation of correlation coefficient for each item pair using support count from the FP-tree and extraction of all the top-k strongly correlated item pairs based on the correlation coefficient value, ϕ. Tkcp are also suffers from the same limitations as Tcp.

4.3 Motivation

Existing correlated pair finding techniques require multiple passes over the database, which is too costly for large transaction databases. It would be more effective if both strongly correlated pairs as well as top-k strongly correlated item pairs can be extracted using a single pass over the database and without generating any large tree or candidate itemsets. Majority of correlation mining techniques use Pearson's correlation coefficient for finding strongly correlated item pairs. These are parametric techniques that work well with continuous variables. Since typical market basket data are binary in nature, a parametric approach may not always perform well for binary data. Further, the parametric correlation coefficient is sensitive to outliers in a data set.

To address the above issues, we present two fast and effective techniques, (i) SCOPE, which extracts all strongly correlated item pairs, for any large database and (ii) k-SCOPE, an extension to SCOPE, to extract top-k strongly correlated item pairs in only one pass over the database, without generating any candidate set. To overcome the problems associated with Pearson correlation, non-parametric techniques like *Spearman's* ρ [51] can be considered as better alternative in this regard. Below, we present an approach for computing *Spearman's* ρ between an item pair from market basket data.

4.4 Computing Spearman's Rank order correlation

Parametric techniques like Pearson's correlation are sensitive to the distribution of the data[52,53]. Parametric techniques may not be effective when data is noisy and binary in nature. The alternative solution is to apply non-parametric correlation. If two variables X and Y are metric (e.g., interval or ratio scale measures) and they are to be correlated, a parametric technique like Pearson's correlation coefficient is suitable. While desirable, it is not always possible to use a paramet-

ric test such as the Pearson's method. In case of non-parametric (e.g., nominal or ordinal measures) variables, correlation can be determined effectively by using a non-parametric correlation technique. Non-parametric correlation coefficients, such as Chi-square, Point biserial correlation[54], Spearman's ρ[51], Kendall's τ[55], and Goodman and Kruskal's λ[56] may perform better than parametric correlation coefficient when outliers are present. The most frequently used is the rank order based Spearman's ρ correlation. In principle, Spearman's ρ is simply a special case of Pearson's product-moment coefficient in which two sets of data X_i's and Y_i's are converted to rank x_i's and y_i's before calculating the coefficients. The raw scores are converted to ranks, and the differences D_i's between the ranks of the observations on the two variables are calculated. If there are no tied ranks, then ρ is given by:

$$\rho = 1 - \frac{6 \sum D_i^2}{N(N^2 - 1)}. \tag{4.5}$$

where, $D_i = x_i - y_i$, is the difference between the ranks of corresponding values X_i and Y_i, and N is the number of samples in each dataset (same for both sets).

If tied ranks exist, each tied score is assigned a rank equal to the average of all tied positions. For example, if a pair of scores are tied for the 2nd and 3rd rank, both scores are assigned a rank of 2.5 ((2+3)/2=2.5). In case of binary variables, a large number of tied cases are present. Binary market basket data contains score 0 and 1 only. To compute the ranks of 0 and 1, their natural ordering can be used to compute tied ranks. For computing the tied ranks, we assign 0 with greater priority than 1 and use simple frequency counts of 1 and 0. Below, we discuss the technique for calculating Spearman's ρ with tied cases between binary variables.

Assume a binary variable I with N values. The frequency of score 1 in variable I is denoted as $f(I)$. To determine the appropriate rank of tied cases, we need to add the rank positions and divide by the number of tied cases. Since the number of scores of a binary variable is only two, the tied rank of score 0 and 1 can be

calculated as:

$$Rank_0 = \sum_{i=1}^{N-f(I)} \frac{i}{N - f(I)}. \tag{4.6}$$

Similarly, the rank of score 1 can be calculated as:

$$Rank_1 = \sum_{i=f(I)}^{N} \frac{i}{f(I)}. \tag{4.7}$$

Once the ranks of 0 and 1 are calculated for the target variables (item pairs), say I_1 and I_2, whose correlation is to be calculated, the difference of their ranks is then used to compute D_i^2. In case of two binary item sets, the only possible combination of scores are (0,0), (0,1), (1,0) and (1,1). Thus just by counting the frequencies of the above patterns and using the rank of 0 and 1 for both item sets, the sum of square differences of ranks (D_i^2) can be easily calculated as:

$$\begin{aligned} \sum D_i^2 \;=\; & P_{(00)}(Rank_0(I_1) - Rank_0(I_2))^2 + P_{(01)}(Rank_0(I_1) - Rank_1(I_2))^2 + \\ & P_{(10)}(Rank_1(I_1) - Rank_0(I_2))^2 + P_{(11)}(Rank_1(I_1) - Rank_1(I_2))^2 \end{aligned} \tag{4.8}$$

where, $P_{(00)}, P_{(01)}, P_{(10)}$ and $P_{(11)}$ are the frequencies of (0,0), (0,1), (1,0) and (1,1) patterns, respectively. $Rank_0(I_1)$ and $Rank_1(I_1)$ correspond to the ranks of 0's and 1's in item I_1 and $Rank_0(I_2)$ and $Rank_1(I_2)$ are the corresponding 0's and 1's rank in item I_2, respectively.

For a long sequence of binary data occurring in a large transaction dataset, sometimes it is costly to find $P_{(00)}, P_{(01)}, P_{(10)}$ and $P_{(11)}$ for each pair of items. It would be more effective if these can be computed with minimal information. One possible way is given below, especially when frequency of score 1 in item I_1 and I_2 ($f(I_1)$ and $f(I_2)$) and joint occurrences in both item pairs ($f(I_1, I_2)$) are known.

$P_{(01)} = f(I_2) - f(I_1, I_2), P_{(10)} = f(I_1) - f(I_1, I_2), P_{(11)} = f(I_1, I_2),$
$P_{(00)} = N - (P_{(01)} + P_{(10)} + P_{(11)})$.

4.4.1 . Computing Spearman's ρ: an illustration

In this section we demonstrate how to compute Spearman's ρ with tied cases for binary market basket data. For example, let us consider the following item pairs I_1 and I_2 with $N = 6$ transactions $(T1, T2, \cdots, T6)$ with similar occurrence patterns.

Table 4.1: Sample market basket data with two items and six transactions

	T1	T2	T3	T4	T5	T6
I_1	1	1	1	0	0	0
I_2	1	1	1	0	0	0

In the above data, it is obvious that the frequency of 1 in I_1 and I_2, and joint occurrences of 1 in both I_1 and I_2 are $f(I_1) = 3, f(I_2) = 3, f(I_1, I_2) = 3$, respectively. Using Equations (4.6) and (4.7) the value of $Rank_0(I_1)$ and $Rank_1(I_1)$ for I_1 become $Rank_0(I_1) = (1+2+3)/3 = 2$ and $Rank_1(I_1) = (4+5+6)/3 = 5$ Similarly, $Rank_0(I_2)$ and $Rank_1(I_2)$ of I_2 are 2 and 5, respectively.

The next step is to calculate $P_{(00)}, P_{(01)}, P_{(10)}$ and $P_{(11)}$ using joint frequency count $f(I_1, I_2)$.

$P_{(01)} = f(I_2) - f(I_1, I_2) = 3 - 3 = 0$, $P_{(10)} = f(I_1) - f(I_1, I_2) = 3 - 3 = 0$, $P_{(11)} = f(I_1, I_2) = 3$, $P_{(00)} = N - (P_{(01)} + P_{(10)} + P_{(11)} = 6 - (0 + 0 + 3) = 3$

The summation of square rank difference D_i^2 is:

$\sum D_i^2 = P_{(00)}(Rank_0(I_1) - Rank_0(I_2))^2 + P_{(01)}(Rank_0(I_1) - Rank_1(I_2))^2 + P_{(10)}(Rank_1(I_1) - Rank_0(I_2))^2 + P_{(11)}(Rank_1(I_1) - Rank_1(I_2))^2.$

$= 3(2-2)^2 + 0(2-5)^2 + 0(5-2)^2 + 3(5-5)^2$

$= 0.$

Thus, Spearman's ρ can be calculated as: $\rho = 1 - (6 \times 0)/6(6^2 - 1) = 1 - 0 = 1$.

In a transaction database, computing $f(I_1)$, $f(I_2)$ and $f(I_1, I_2)$ for any item pair is nothing but finding their individual and joint supports (1- and 2- element item sets).

A traditional approach for counting the support for item pairs need at least two scans over the entire dataset. We present a one-pass strongly correlated item pair finding technique called SCOPE and its extension k-SCOPE. We use a correlogram matrix for counting individual and joint supports for item pair in a single pass over the database. Using support count and the method discussed above, it is straightforward to compute the correlation coefficient ϕ or ρ between an item pair.

4.5 SCOPE: Strongly COrrelated Pair Extraction Technique

SCOPE attempt to find all strongly correlated item pairs and k-SCOPE extracts k top most correlated pairs from any transaction database using a single scan over the database without generating any candidates. We use a correlogram matrix to store the support counts of all 1- and 2- element itemsets. Later, the matrix is used to calculate correlation coefficients of all item pairs.

SCOPE accepts the market-basket database D and the correlation coefficient threshold θ as input, and generates all strongly correlated item pairs as output. Step 1 of SCOPE (see Algorithm 2) is dedicated to the construction of the correlogram matrix using a single scan of the original database. In step 3, the correlation coefficient of each item pair is computed and in step 5, all item pairs whose coefficient values are greater than or equal to θ, are extracted. Finally, the algorithm returns a list of all strongly correlated item pairs.

An extended version of this algorithm for generating top-k correlated pairs, viz, k-SCOPE is presented in Algorithm 3. The algorithm accepts the market-basket database D and k as input and generates a list of top-k strongly correlated item pairs, L, as output. The first phase of the algorithm is the same as that of SCOPE.

```
input  : D (Original Dataset), θ (Correlation coefficient threshold)
output: L (List of strongly correlated item pairs)
1 Generate Correlogram Matrix M from D;
2 for each item pair (i, j) ∈ D do
3 |   Compute Corr (i, j) using support from M;
4 |   if Corr (i, j) ≥ θ then
5 |   |   L:= L ∪ (i, j) ;
6 |   end
7 end
8 Return L;
```

Algorithm 2: SCOPE: Strongly COrelated Pair Extraction

In steps 8 to 12, topmost k correlated item pairs are extracted and added to the top-k list. Top-k list L is a sorted list (descending order) of item pairs based on value of the correlation coefficient. Any pair whose correlation coefficient is lower than the k^{th} pair's correlation coefficient is straightaway pruned. Otherwise, the algorithm updates the list by eliminating the k^{th} pair and inserting the new pair in its appropriate position in the list. Finally, the algorithm returns top-k list L.

```
input  : D (Original Dataset), k
output: L (List of k strongly correlated item pairs)
1 Generate Correlogram Matrix M from D;
2 for each item pair (i, j) ∈ D do
3 |   Compute Corr (i, j) using support from M;
4 |   if |L| ≤ k then
5 |   |   L:= L ∪ (i, j) ;
6 |   end
7 |   else
8 |   |   if Corr (i, j) ≥ Corr (L[k]) then
9 |   |   |   L [k]:= L [k] ∪ (i, j) ;
10|   |   |   Sort L in descending order on Corr of each pair ;
11|   |   end
12|   end
13 end
14 Return L;
```

Algorithm 3: k-SCOPE: Top k strongly correlated Pair Extraction

4.6 Analysis of Our Algorithms

Here, we analyse SCOPE and k-SCOPE in terms of completeness, correctness and computational complexity.

4.6.1 Completeness and correctness

Lemma 4.6.1. *SCOPE is complete, i.e., SCOPE finds all strongly correlated pairs.*

Proof. Since SCOPE is based on exhaustive search and computes correlation coefficients of all pairs without pruning any item pair, SCOPE extracts all strongly correlated item pairs with coefficient greater than the threshold θ. This fact ensures that SCOPE is complete in all respects. □

Lemma 4.6.2. *SCOPE is correct, i.e., the correlation coefficient of all pairs, extracted by SCOPE, is above threshold θ.*

Proof. The correctness of SCOPE can be guaranteed by the fact that SCOPE calculates exact correlation of each pair present in the database and prunes all pairs whose correlation coefficient is lower than the user specified threshold θ. □

Lemma 4.6.3. *k-SCOPE is complete, i.e., k-SCOPE finds top-k strongly correlated pairs.*

Proof. Like SCOPE, k-SCOPE is based on exhaustive search and computes the correlation coefficient of all pairs without pruning any item pairs. Therefore, k-SCOPE extracts k top most strongly correlated item pairs based on the value ϕ. This fact ensures that k-SCOPE is complete in all respects. □

Lemma 4.6.4. *k-SCOPE is correct, i.e., correlation coefficients of the extracted pairs are the k top most correlation coefficients.*

Proof. The correctness of k-SCOPE can be guaranteed by the fact that, k-SCOPE calculates exact correlation of each pair present in the database and creates a sorted list (descending order) of item pairs based on the correlation coefficient and prunes all pairs whose correlation coefficient is lower than the k^{th} pair's correlation coefficient. □

4.6.2 Complexity analysis

Since k-SCOPE is an extension of SCOPE, we analyze only k-SCOPE in terms of space and time complexity.

4.6.2.1 Space complexity

TAPER and TOP-COP need memory for keeping the top-k list and support count of all items, and a huge number of candidate item pairs depending on the value of the θ upper bound. TOP-COP maintains a list with the pruning status of all item pairs out of N items, requiring memory space of order (N^2). Tkcp creates an entire FP-tree in the memory with initial support threshold zero (0). This tree is normally huge when the number of transactions as well as the dimensions are large. Its size also depends on the number of unique patterns of items in the database. Sometimes it is difficult to construct such a tree in the memory. However, k-SCOPE always requires a fixed memory of size, $N \times (N+1)/2$ to construct the correlogram matrix and array of size k to store top-k strongly correlated item pairs. Thus, the total space requirement is:

$$\begin{aligned} SPACE_{k-SCOPE} &= O(N * (N+1)/2) + O(k) \\ &\approx O(N^2) + O(k). \end{aligned}$$

4.6.2.2 Time complexity

The computational cost for k-SCOPE consists of two parts: *(i)* correlogram matrix construction cost (C_{CM}) and *(ii)* the cost for extraction of top-k strongly correlated item pairs (C_{EX}).

a) Construction of correlogram matrix: Cost can be calculated as describe in section 3.5.2.2 of chapter 3.

b) Extraction of top-k strongly correlated item pairs: To calculate the correlation

of each pair, k-SCOPE must traverse the correlogram matrix once. Thus, the time requirement for extracting the correlation coefficient of all item pairs is $O(N^2)$. To create the top-k list, for each item pair the algorithm compares the correlation coefficient ($Corr$) of the new pair with $(k-1)^{th}$ pair in the list. If $Corr$ of the new pair is greater than that of the k^{th} pair, the k^{th} pair is eliminated from the list and a new pair is inserted and placed in the list in descending order of $Corr$. Thus, for placing a new pair, it requires at most k number of comparison and swapping. Since, the problem is to find k top most item pairs out of $N*(N-1)/2$ item pairs, the time requirement for creating list of top k item pairs can be denoted as:

$$\begin{aligned} C_{EX} &= O(N^2) + O(k*(N*(N-1))/2) \\ &\approx O(N^2) + O(k*N^2) \\ &\approx O(k*N^2). \end{aligned}$$

Thus, in the worst case total cost incurred by k-SCOPE is:

$$\begin{aligned} COST_{k-SCOPE} &= C_{CM} + C_{EX} \\ &= O(T*N^2) + O(N^2) + O(k*(N*(N-1))/2) \\ &\approx O(T*N^2) + O(N^2) + O(k*N^2). \end{aligned}$$

The computational cost of the TOP-COP and TAPER algorithms are almost similar, except that the cost of computing the exact correlations for remaining candidates may be less in the case of TOP-COP, as it prunes more non-potential item pairs based on the 2-D monotone property. The cost of TOP-COP can be modeled as,

$$COST_{TOP-COP} = C_{Sort} + C_{Bound} + C_{Exact} + C_{k-list}$$

where $C_{Sort}, C_{Bound}, C_{Exact}$ and C_{k-list} are the costs of creating a sorted list of items in non-increasing order of support, the cost of computing upper bounds, computing the cost of exact correlation of remaining pairs, and k-top list maintenance cost,

respectively. After simplifying the above cost computation, we get

$$COST_{TOP-COP} = O(NlogN) + O(N^2) + O(N^2) + O(k^2).$$

However, this cost model, does not consider the cost of scanning the database. It requires one scan for creating the initial sorted item list and at least another whole scan (when any hash based data structure is used) of the database for computing exact correlation of existing pairs after pruning. After adding this cost, the total becomes

$$\begin{aligned} COST_{TOP-COP} &= O(T*N) + O(NlogN) + O(T*N) + O(N^2) + O(N^2) + O(k^2) \\ &\approx 2*O(T*N) + O(NlogN) + 2*O(N^2) + O(k^2). \end{aligned}$$

Similarly the cost of Tkcp algorithm can be modeled as:

$$\begin{aligned} COST_{Tkcp} &= C_{Sort} + C_{D\_Sort} + C_{FP} + C_{k-list} \\ &= (O(T*N) + O(NlogN)) + O(T*N^2)(O(T*N) + C_{FP\_Tree}) \\ &+ (O(N)*C_{Cond\_base} + O(P*k^2) \end{aligned}$$

where C_{Sort} is the cost of creating the initial sorted list of items based on support count using one pass of the database, C_{D_Sort} is the cost incurred during sorting the database in descending order of item support, and C_{FP} is the total cost of creating the FP-tree. The creation of the complete FP-tree requires one complete scan over the database and the cost of creating the pattern tree is C_{FP_Tree}. To compute the correlation of each pair and to maintain the k-top list, it requires additional cost C_{Cond_base} for creating the conditional pattern base (P) for each item. We see that the cost of scanning a database is much larger than the other computational parameters. So, the computational savings of k-SCOPE, i.e., $(O(T*N^2))$ is larger when the number of records in a transaction database is very high.

4.7 Performance Evaluation

To evaluate the performance of SCOPE and k-SCOPE and to compare them with other techniques, we test them using several synthetic as well as real-life datasets. Since, TAPER in its original form cannot generate the top-k list, we modified TAPER, so that it can generate such a top-k strongly correlated item pair list. As TAPER is dependent on the correlation threshold θ, in order to generate the same result using TAPER we set θ as the correlation coefficient of the k-th pair from the top-k list generated by k-SCOPE. The ideal θ value for TAPER for different datasets are presented in Table 4.7.1. We also provide results showing the performance of Spearman's ρ as correlation coefficient, compared to Pearson's ϕ when used with market basket data.

We implemented our techniques using Java 1.6 on Windows 7 platform running in 2.53 GHz machine. We used same environment for implementation of SCOPE, k-SCOPE, TAPER and the modified version of TAPER. For TOP-COP, we used the code as provided by the original author. Since performance of Tcp and Tkcp is highly dependent on FP-tree implementation, we use a third party FP-tree implementation from[57] for Tcp and Tkcp to avoid any implementation bias.

4.7.1 Dataset used

To generate synthetic dataset, we used *ARMiner*[a] software and generate several synthetic datasets. The details of the synthetic dataset is given in Table 4.2. We also used market basket version of three real datasets, Mushroom, Pumsb and Chess. *Mushroom dataset*[b] is taken from FIMI[c], the *Pumsb*[d] dataset from IBM, corresponding to a binarized versions of a census dataset. *Pumsb* is often used as the benchmark for evaluating the performance of association mining algorithms on dense datasets. The details of real transaction datasets are given in Table 4.3.

[a] *http://www.cs.umb.edu/laur/ARMiner/*
[b] *http://www.ics.uci.edu/mlearn/MLRepository.html*
[c] *http://fimi.ua ac.be*
[d] *http://fimi.cs.helsinki.fi/data/*

Table 4.2: Synthetic Transaction Dataset

Data Set	No.of Transactions	No. of Items	Avg. size of Transaction	No. of Pattens
T10I400D100K	100,000	400	10	20
T10I600D100K	100,000	600	10	20
T10I800D100K	100,000	800	10	20
T10I1000D100K	100,000	1000	10	20
T10P1000D100K	100,000	1000	10	1000

Table 4.3: Real Dataset

Data Set (Market Basket)	No.of Transactions	No. of Items	Source
Mushroom	8124	128	http://fimi.ua.ac.be
Pumsb	49046	2113	http://fimi.cs.helsinki.fi
Chess	3196	75	http://fimi.ua.ac be

4.7.2 Experimental results

To evaluate the performance of the proposed algorithms, we compare them with other similar techniques in terms of execution time for different values of θ and k. We find that Tcp and Tkcp, consume a lot more time compared to other two techniques, since Tcp and Tkcp generate the entire FP-tree with the initial minimum support value of 0. We also observe that Tcp and Tkcp do not work when the number of items is more than 1000. In case of T10P1000D100K dataset, both Tcp and Tkcp failed to mine, due to the large number of items and unique patterns. However, in all cases, SCOPE exhibits better performance than TAPER and Tcp. With decrease in the value of θ, the running time of TAPER also increases, since low θ value generates a large number of candidate sets. But, SCOPE and Tcp keep stable running time for the whole range of correlation thresholds in different datasets. We further confirm the fact that like Tcp, SCOPE is also robust with respect to input parameters (Figure 4.3).

From the performance graph in Figures 4.3 and 4.4, we easily observe that modified TAPER performs much better than TOP-COP, even though TOP-COP is an improved and modified version of TAPER. It is because of the use of an efficient hash

Table 4.4: Suitable θ value for different datasets

Data Set	k values				
	100	200	300	400	500
Mushroom	0.49	0.37	0.31	0.25	0.23
Pumsb	0.97	0.869	0.764	0.703	0.647
T10I400D100K	0.51	0.027	-0.006	-0.011	-0.016
T10I600D100K	0.81	0.27	0.001	-0.006	-0.009
T10I800D100K	0.63	0.290	0.001	-0.003	-0.005
T10I1000D100K	0.96	0.95	0.94	0.93	0.89
T10P1000D100K	0.95	0.92	0.87	0.83	0.80

data structure, which is lacking in the original TOP-COP implementation. This further indicates that the performance of correlation mining algorithms can be improved through efficient implementation. However, in all cases, k-SCOPE exhibits better performance than TAPER (modified), TOP-COP and Tkcp. TOP-COP exhibits an exponential performance graph (Figures 4.3 and 4.4) as the number of items increases. But k-SCOPE and Tkcp maintain stable running time in different datasets, since both algorithms are independent of θ. It further confirms the fact that SCOPE and k-SCOPE are robust with respect to input parameters θ and k.

4.7.2.1 Scalability of k-SCOPE

The scalability of the k-SCOPE algorithm with respect to the number of transactions and number of items in the databases is shown in Figure 4.6. We used *ARMiner* to generate four datasets with the number of transactions ranging from 1,00,000 to 5,00,000. In each case, we kept the number of test items at 1,000 as we tested for scalability in terms of number of transactions. To test scalability in terms of number of items, we generated another five transaction datasets with numbers of items ranging from 2,000 to 10,000 keeping number of transactions equal to 1,00,000. We observe the execution time increases linearly with increase in the number of transactions and items at different k values. Figure 4.6 shows the scalability test results for k ranging from 500 and 2500. From the graph, it is clear

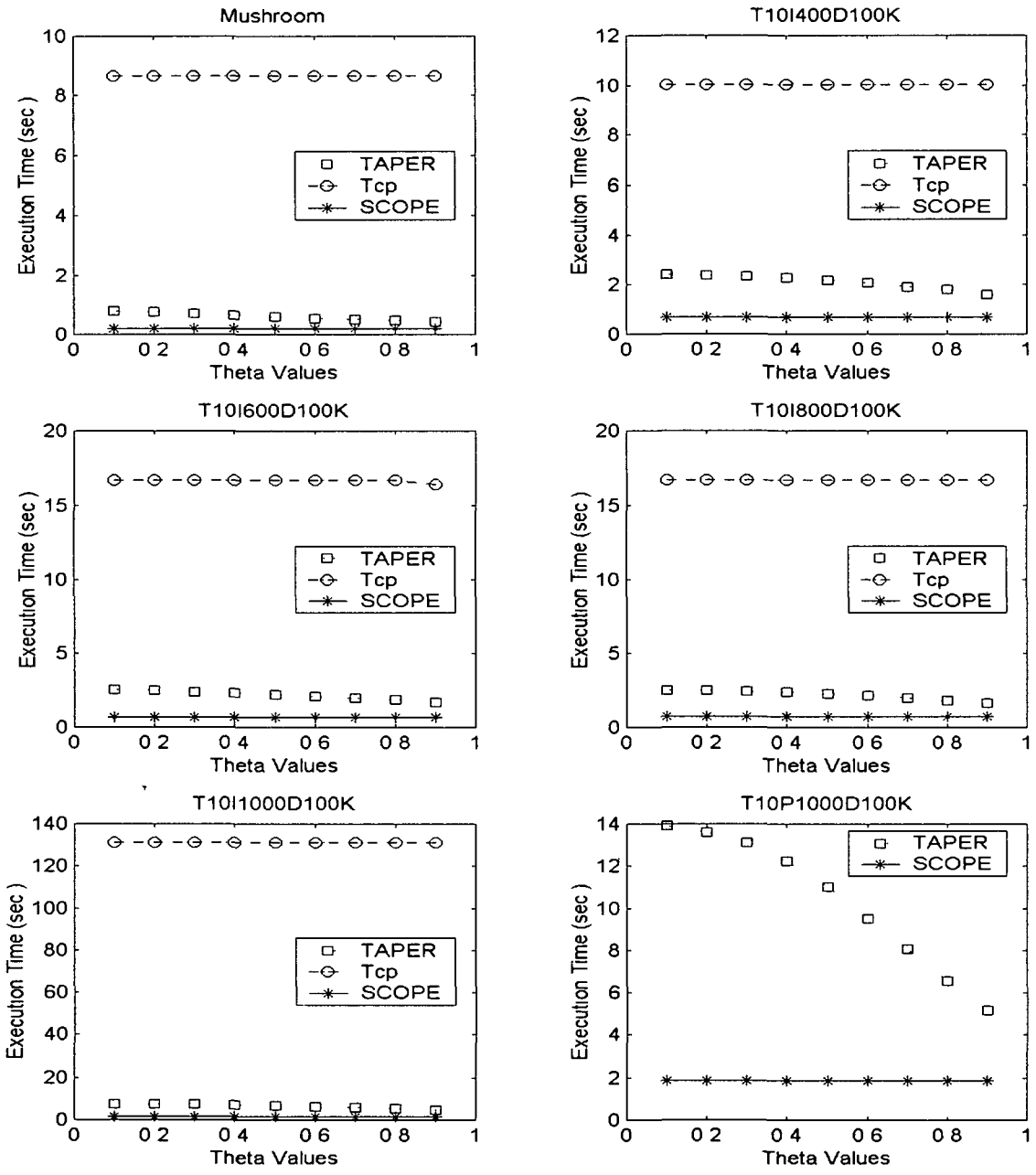


Figure 4.3: Execution time comparison between SCOPE, Tcp and TAPER

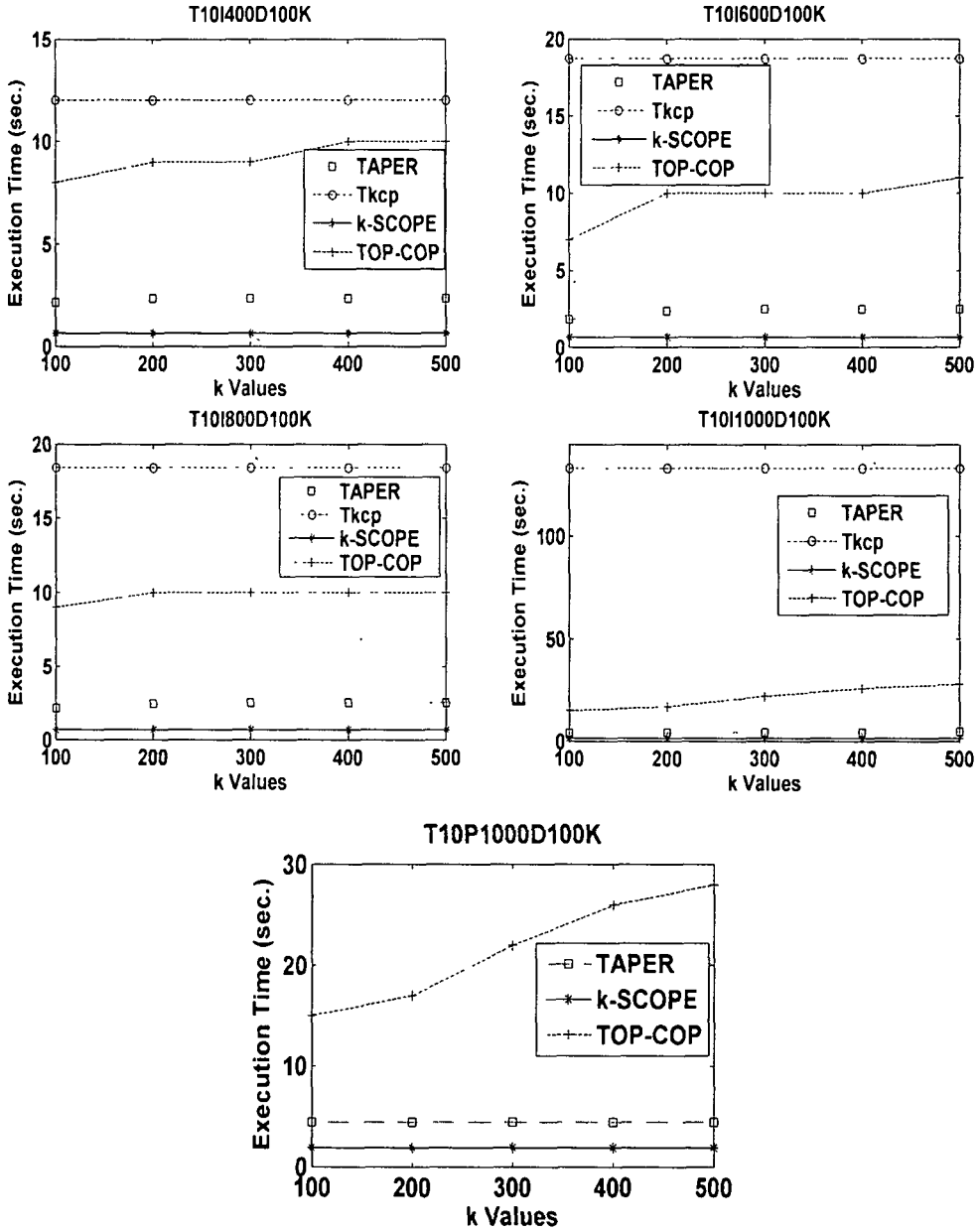


Figure 4.4: Execution time comparison of k-SCOPE with TAPER (mod), Tkcp and TOPCOP on Synthetic dataset

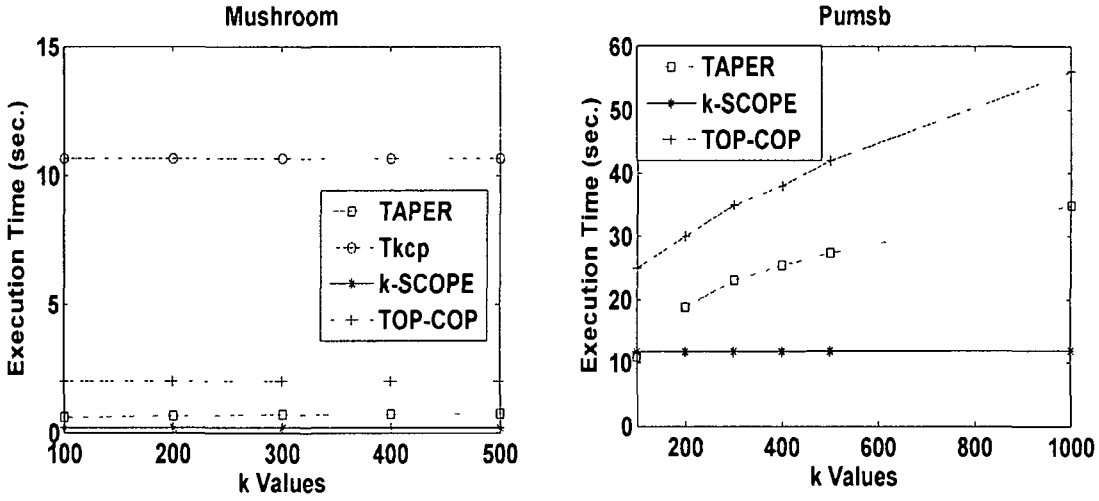


Figure 4.5: Execution time comparison of k-SCOPE on real dataset

that the performance of k-SCOPE is not sensitive to input parameter k. Thus, k-SCOPE is robust in handling large transaction databases for different values of k.

4.7.2.2 Pearson's ϕ vs. Spearman's ρ in correlated item pair findings

Now we provide a few results to establish that Spearman's ρ is superior in comparison to Pearson's ϕ over market basket dataset in terms of (i) finding number of correlated item pairs and (ii) correlation coefficient values for different k values.

For measuring the performance of ϕ and ρ as correlation coefficient for finding correlated item pairs, we used the Mushroom and Chess datasets. We measured the number of possible correlated item pairs for various θ values, generated by both Pearson's ϕ and Spearman's ρ. In the Figure 4.7 and 4.8, we easily observe that Spearman's ρ generates more correlated pairs compared to Pearson's ϕ. Similarly, when we measure k^{th} correlation coefficient value for different k values, we find that Spearman's ρ gives higher values than Pearson's ϕ. We conclude that Spearman's ρ is able to detect hidden correlated pairs undetected by Pearson's ϕ. Moreover, in some cases, ρ gives higher correlation value compared to ϕ, for a particular pair. We feel that this is due to the problems associated with Pearson's ϕ, as already discussed.

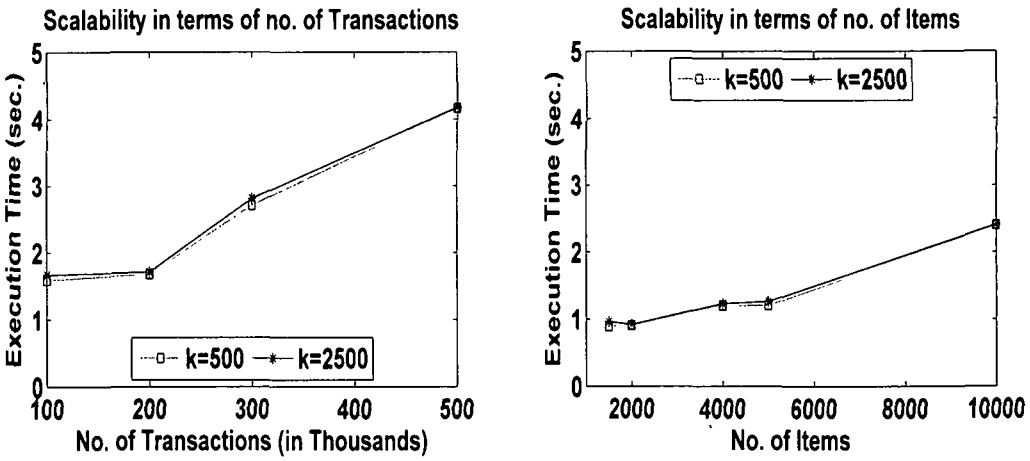


Figure 4.6: Scalability of k-SCOPE algorithm.

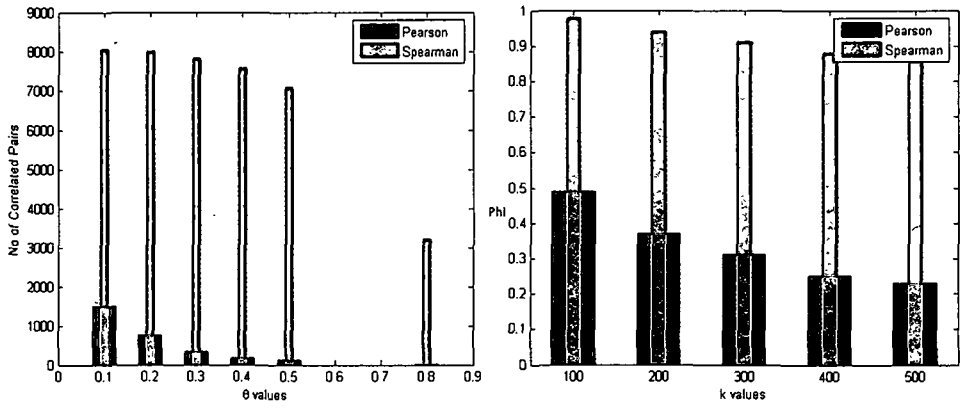


Figure 4.7: Performance comparison of Pearson's ϕ and Spearman's ρ on Mushroom dataset

4.8 Discussion

We have presented two effective techniques for finding strongly correlated item pairs and top-k strongly correlated item pairs from market basket data, in this chapter. We have also presented an alternative way of measuring correlation coefficient between item pairs from market basket data using Spearman's rank order correlation. The advantages of these techniques compared to existing similar techniques are (i)

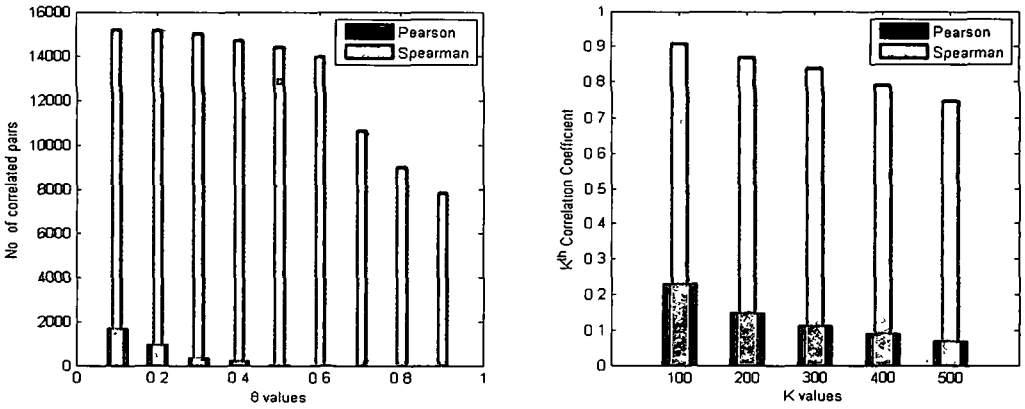


Figure 4.8: Performance comparison of Pearson's ϕ and Spearman's ρ on Chess dataset

they require single pass over the whole database, and (ii) they require no candidate generation. We evaluated both the techniques using several synthetic and real life datasets and found that the results are quite satisfactory.

In the following chapters, we present potential application of above data mining techniques in gene expression data analysis. Next chapter presents a pattern based gene co-expression network finding technique using correlogram matrix.

Chapter 5

Expression Pattern Based Reconstruction of Gene Co-expression Networks

Biological networks connect genes or gene products to one another. A network of co-regulated or co-expressed genes may form gene clusters that can encode proteins and take part in common biological processes. The most preliminary form of network is gene co-expression network which basically describes the inter-relationships between different genes. Existing techniques generally depend on proximity measures based on global similarity to draw the relationship between genes. It has been observed that expression profiles are sharing local similarity rather then global similarity. In this chapter, we propose an expression pattern based method called **GeCON** to extract **G**ene **CO**-expression **N**etworks from microarray data. Pairwise supports are computed for each pair of genes based on changing tendencies and regulation patterns of gene expression. Gene pairs showing negative or positive co-regulation under a given number of conditions are used to construct such gene co-expression network. The genes in a network with high pattern similarity form a coherent group. We construct a co-expression network with signed edges to reflect up and down regulation between a pair of genes. We apply GeCON on both real and synthetic gene expression data. Publicly available gene expression

datasets are used to generate gene co-expression networks and measure biological significance of the network modules in terms of gene ontology. We reconstruct *in silico* gene regulatory networks using DREAM3 and DREAM4 Challenge data and evaluate the predicted networks against the actual networks. We compare our results from DREAM data with three well known algorithms, viz., ARACNE, CLR and MRNET. Experimental results show that GeCON can extract biologically relevant networks as well as effectively infer of *in silico* gene regulatory networks. It outperforms other algorithms based on *in silico* regulatory network reconstruction.

5.1 Introduction

Microarray technology makes it available a large numbers of expression data over protein and metabolite activity. It allow us to study the dynamic behaviour of a gene inside cell. Reverse engineering is a promising area of research in Systems Biology, tries to recreate the cellular system for better understanding of biological mechanism. The development of suitable reverse engineering method is necessary to get insight into gene-gene relationships, which may further lead to discovery of functional gene modules. Gene-gene relationships can be described through biological pathways, which can be represented as networks and broadly classified[4] as *metabolic pathways*, *signal transduction pathways* and *gene regulatory networks*. The most preliminary network is the gene co-expression network, which describes inter-relationships among genes.

A gene co-expression network is an undirected graph, where the nodes correspond to genes or gene activities, and undirected edges between genes represent significant co-expression relationships[58,59]. In a co-expression network, two genes are connected by an undirected edge if their activities have significant association over a series of gene expression measurements. Compare to regulatory networks, a gene co-expression network does not attempt to draw direct causal relationships among the participating genes in the form of directed edges. A co-expression network may form co-regulated gene clusters that can encode proteins, which interact

among themselves and take part in common biological processes. Co-expression network analysis plays vital role in inferring relationship among biological processes[7] .

5.2 Related Works

A number of techniques have been proposed for such network construction[44,60,61,62,63,64]. Existing techniques for finding gene networks can be broadly categorized as (i) Computational approaches, and (ii) Literature based approaches. The computational approach mainly uses statistical, machine learning or soft-computing techniques[60,65] as discovery tools. On the other hand, a literature based approach gathers relevant published information on genes and their inter-relationships and constructs networks based on such documented information. The literature based approach is capable of building networks with high biological relevance but is computationally expensive. A biomedical literature search based technique is used in[66] to construct gene relation networks by mapping literature knowledge into gene expression data.

Network models such as Bayesian[67] and boolean networks[68] are used to infer interrelationships among genes. Kwon et al.[69], extract gene regulatory relationships for cell cycle-regulated genes with activation or inhibition between gene pairs. Regulatory relationships have also been deduced from correlation of co-expressions, between DNA-binding transcription regulator and its target gene, by using a probabilistic expression model[70]. Although standard statistical techniques for extracting relationships can come up with multiple models to fit the data, they often require additional data to resolve ambiguities. Soft computing tools like fuzzy sets, neuro-computing, evolutionary computing and their hybridization are alternatives for handling real life ambiguities. Mitra et al.[61] propose a bi-clustering technique to extract simple gene interaction networks. They use continuous column multi-objective evolutionary bi-clustering to extract rank correlated gene pairs. Such pairs are used to construct the gene network for generating relationship between

a transcription factor and its target's expression level. Similarly, Jung and Cho[62] also propose an evolutionary computation based approach for construction of gene (interaction) networks from gene expression time-series data. It assumes an artificial gene network and compares it with the reconstructed network from the gene expression time-series data generated by the artificial network. Next, it employs real gene expression time-series data to construct a gene network by applying the proposed approach. Mutual information[46,71] or correlation coefficient[44,63,64] based approaches have been proposed for extracting gene-gene interaction networks. It has been observed that a pair of genes with high mutual information are non-randomly associated with each other biologically or with biological significance. Butte et al.[46] compute comprehensive pair-wise mutual information for all genes in an expression data set. By picking a threshold mutual information and using only associations at or above the threshold, they construct Relevance Networks. A number of additional mutual information-based approaches have been also proposed. Some of the well known algorithms in this category are CLR[72], ARACNE[73], MRNET[74].

5.3 Motivation

Most existing computational approaches extract networks based on global similarity such as correlation or mutual information. This is computationally expensive and sometimes, may not be able to obtain biologically relevant groups of genes or networks. Pairwise correlation or mutual information may not reveal the proper relationships. Existing approaches compute similarity considering expression values in all dimensions. It is well known that gene expressions may match each other under some conditions or samples, when correlation score is penalized due to mismatch in a condition. Moreover, expressions may contain scaling and shifting patterns[75], which also may affect the correlation measure in drawing true association among genes. Mutual information based techniques are effective alternatives to correlation measures. However, most such work discretizes the expression values

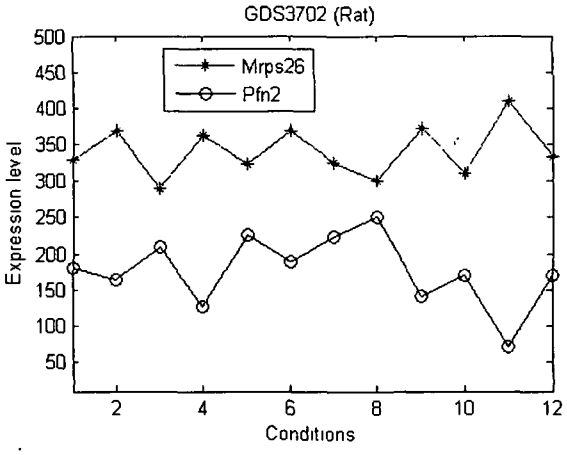


Figure 5.1: Expression profile of RAT genes showing negative or inverted regulation

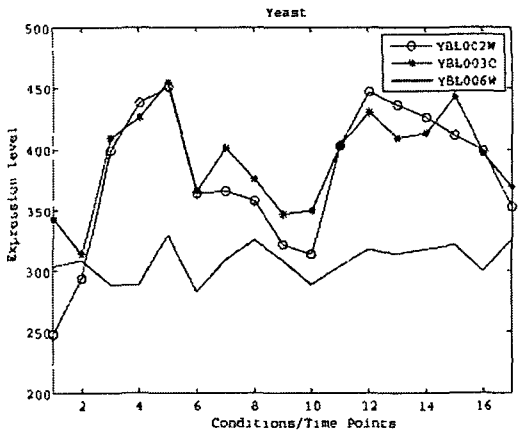


Figure 5.2: Yeast genes showing positive and negative regulation

before computing mutual information. Discretization may lead to information loss. We note that two genes may be related to each other even when their expression patterns show negative or inverted behaviour[76]. In Figure 5.1, expression patterns of Rat gene *Mrps26* and *Pfn2*, taken from NCBI dataset, GDS3702, clearly show negative behaviour. Gene ontology suggests that both are responsible for regulation of interferon-beta production. Again, we easily observe that in the Yeast datsets given in [20], genes *YBL002W* and *YBL003C* have a similar pattern and gene *YBL006W* has an inverted behaviour with respect to the other two genes. If we observe Figure 5.2 more closely, we see that expression patterns also share mixed regulation (i.e., both positive and negative). As suggested by gene ontology all three genes are involved in nucleosome organization, protein-DNA complex sub-unit organization, chromatin and chromosome organization and cellular macro-molecular subunit organization. A group of genes may share a combination of both positive and negative co-regulation under a few conditions or at a few time points. Majority of existing approaches capture genes with similar tendency as co-expression but ignore the patterns like the ones we discuss above.

In computing similarity, many well known techniques do not consider positive- or negative- regulation patterns as presenting co-expression or co-regulation with associated biological significance In our work, we capture pair-wise similarity

purely on pattern matching followed by construction of the co-expression network. We consider both positive and negative regulation as co-regulation. Unlike available measures, we use a support based approach to compute similarity between two expression patterns. We also consider the case where two genes show similar patterns only under some conditions or time points. Available co-expression network finding techniques discover only limited association between the genes. Since creating a co-expression network is a preliminary step towards gene regulatory network discovery, we use signed edges between the genes to represent positive and negative regulations, an important component in gene regulatory networks. Computing correlation or mutual information for all possible pairs is a computationally expensive task. The few approaches developed so far to discover gene co-expression networks are computationally expensive. We compute the similarity between expression patterns of two genes using a one-pass support count based approach without discretizing the expression database. Gene pairs showing high support, i.e., high pattern similarity are used to construct a gene co-expression network. We apply our approach to several real expression datasets. Since genes participating in a network form gene groups with high co-regulation, we assess our results by evaluating the gene groups against biologically significant gene ontology terms associated with a group.

5.4 Expression Pattern based Co-expression networking

Clustering based on global similarity measures, like Euclidean distance or Pearson correlation, may not always capture true gene-gene relationships[77]. On the other hand, most existing techniques give low emphasis on pattern matching based on local similarity. It has been observed that genes share local rather global functional similarity in their gene expression profiles[61]. Moreover, another observation is that most existing techniques are computationally expensive.

In this section, we develop a pattern similarity based approach to construct co-

expression networks with signed edges to represent regulatory relationships among genes. In general, comparing pair-wise gene profiles require multiple passes over the database, which often is quite expensive, especially for database with large numbers of genes (or rows). In this work, we perform pair-wise comparison using a one-pass approach, and we compute supports using single scan of the database. Pairs of gene showing similarity above a user-defined threshold θ are used to construct the adjacency matrix which is used in turn to construct and visualize the network.

To capture the pattern of an expression profile, the edge between two consecutive expression values of a gene profile is used. Thus, for an expression data with M conditions or time points, there are $(M-1)$ edges. To represent the edge we use two measures, degree of fluctuation and regulation pattern of the edge. The degree of fluctuation of an edge is the angular deviation of the edge on the 180-degree normal plane. Regulation pattern represents the up and down regulation of a pattern or edge.

5.4.1 Terminology used

Let $G = \{G_1, G_2, \cdots, G_N\}$ be the set of N genes and $R = \{O_1, O_2, \cdots, O_M\}$ be the set of M conditions or time points of a micro array dataset. The gene expression dataset D is represented as a $N \times M$ matrix $D_{N \times M}$ where each entry $d_{i,j}$ corresponds to the logarithm of the relative abundance of $mRNA$ of a gene. Following definitions and lemmas provide the theoretical basis of the proposed GeCON algorithm.

Definition 5.4.1 (Pattern Similarity) : Given degrees of fluctuation $A = \{a_1, a_2, \cdots, a_{M-1}\}$ and regulation patterns $R = \{r_1, r_2, \cdots, r_{M-1}\}$ of a gene, derived from the gene expression profile, two genes' k^{th} expression patterns are similar if the difference in the degrees of fluctuation of the two genes' k^{th} edges is less than some given threshold τ. In calculating similarity between edges of two genes, we consider two patterns: Positive similarity, Pos_sim, when the regulation patterns are the same (in case of up regulation) and Negative similarity, Neg_sim, when the patterns are inverted (in case of down regulation) for a particular edge (inverted

pattern) To calculate the degree of fluctuation based on the 180 degree plane, similarity can be defined as follows:

$$Pos\_sim(G_{ik}, G_{jk}) = \begin{cases} 1, & \text{if } G_i(r_k) = G_j(r_k) \\ & \text{and } |G_i(a_k) - G_j(a_k)| < \tau \\ 0, & \text{otherwise,} \end{cases} \tag{5.1}$$

$$Neg\_sim(G_{ik}, G_{jk}) = \begin{cases} 1, & \text{if } G_i(r_k) \neq G_j(r_k) \\ & \text{and } |180 - G_i(a_k) + G_j(a_k)| < \tau \\ 0, & \text{otherwise.} \end{cases} \tag{5.2}$$

Definition 5.4.2 (Support) : It is the ratio between the number of edges for which genes G_i and G_j are similar and the total number of edges $|E|$. We consider both positive and negative support to measure the number of edges where both genes have similar or inverted pattern tendencies, respectively. The formulas are given below.

$$Pos\_support(G_i, G_j) = \sum_{i,j=1}^{|E|} Pos\_sim(G_i, G_j)/|E| \tag{5.3}$$

$$Neg\_support(G_i, G_j) = \sum_{i,j=1}^{|E|} Neg\_sim(G_i, G_j)/|E| \tag{5.4}$$

Definition 5.4.3 (Strongly Connected) : Two genes G_i and G_j are said to be *Strongly Connected* (or have an inter-relationship) if $Pos\_support(G_i, G_j) + Neg\_support(G_i, G_j) > \theta$, where θ is a user defined threshold to indicate the minimum number of edges of two expression profiles must match.

Definition 5.4.4 (Co-expression Network) : Co-expression network is a graph $T = \{G', E\}$ containing a subset of genes that are strongly connected. If two genes $(G_i, G_j) \in G'$ are connected by an edge $E_{ij} \in E$, then G_i,G_j are strongly connected to each other.

Here, $E = \{(E_{ij}, S_k), \cdots (E_{mn}, S_k)\}$ is a set of pairs, where E_{ij} represents an edge connecting G_i and G_j, and S_k represents the sign of the edge E_{ij}. A value

of $S_k = +1$ indicates up or positive regulation and -1 indicates down or negative regulation. To calculate the value of S_k of edge E_{ij}, we use *Pos_support* and *Neg_support*. This is defined as:

$$S_k(E_{ij}) = \begin{cases} +1, & \text{if } Pos\_support(G_i, G_j) > \theta \\ -1, & \text{if } Neg\_support(G_i, G_j) > \theta. \end{cases} \tag{5.5}$$

Lemma 5.4.1. *For any two genes G_i, G_j, if $G_i \in T$, a gene co-expression network, and G_i is strongly connected to G_j, then $G_j \in T$.*

Proof. The above lemma can be proved by contradiction. Assume, G_i and G_j are two strongly connected genes and $G_i \in T$, but $G_j \notin T$. As per Definition 5.4.4, T is a subset of strongly connected genes and since G_i and G_j are strongly connected, $G_j \in T$, which contradicts and hence the proof. □

Similarly the following lemma is trivial based on the Definitions 5.4.1 through 5.4.4 and Lemma 5.4.1.

Lemma 5.4.2. *Let G_i and G_j be two genes, and T_1 and T_2 be two gene co-expression networks. If $G_i \in T_1$ and $G_j \in T_2$, then G_i and G_j are not connected.*

Lemma 5.4.3. *Genes belonging to the same gene co-expression network are co-regulated or similar.*

Proof. This lemma can also be proved by contradiction. Let us assume that any two genes G_i and $G_j \in T$ are not co-expressed. If G_i and G_j are in same network, they are strongly connected (as per Definitions 5.4.3 and 5.4.4), and hence G_i and G_j are strongly connected. Again, any two strongly connected genes are similar or co-expressed (as per Definitions 5.4.1 through 5.4.3), which contradicts the assumption, hence the proof. □

Similarly, the proof of the following lemma (the reverse case of lemma 5.4.3) is trivial.

Lemma 5.4.4. *Genes belonging to different gene networks are not co-expressed.*

5.4.2 Capturing expression pattern

To capture patterns of each gene expression, researchers use either angles between the edges for every pair of conditions[78] or regulation patterns in terms of up- or down- regulation[79]. Angles or regulation patterns between the edges of two conditions, alone, are ineffective in capturing the true expression pattern of a gene. We compare two gene expressions both in terms of degrees of fluctuation and regulation patterns between two adjacent conditions (edges), simultaneously. To capture both regulation patterns and degree of fluctuations of each gene, we read rows of original data with M number of expression values or conditions and convert them into another row of $(M-1)$ number of columns, each column of which contains the degree of fluctuation and the regulation pattern of two adjacent conditions. We consider regulation information 1 and -1 to represent up-regulation and down regulation respectively. Regulation value in the k_{th} edge of a gene G_i , $G_i(r_k)$, based on two consecutive conditions (say, O_{k-1} & O_k) can be calculated as:

$$G_i(r_k) = \begin{cases} 1 & \text{if } O_{k-1} < O_k \\ -1 & \text{if } O_{k-1} > O_k. \end{cases} \tag{5.6}$$

For calculating the degree of fluctuation we compute the arc tangent between two adjacent expression levels (x, y) as in[78] based on 180 degree plane. For computing arc tangent, we used two-argument $atan2$ function. $atan2(y, x)$ is the angle between the positive x-axis of a plane and the point (x, y) on it, with positive sign for counter-clockwise angles and negative sign for clockwise angles. Next, we convert the angle in 180 degree plan as follows:

$$DegreeOfFluctuation(x, y) = \begin{cases} 180 - abs(arctan2(y, x)) & \text{if } y < x \\ abs(arctan2(y, x)) & \text{otherwise.} \end{cases} \tag{5.7}$$

The fact is illustrated in Figure 5.3 taking an example of a gene expression $G = \{343, 314, 409\}$ with three conditions. After preprocessing G it become $G =$

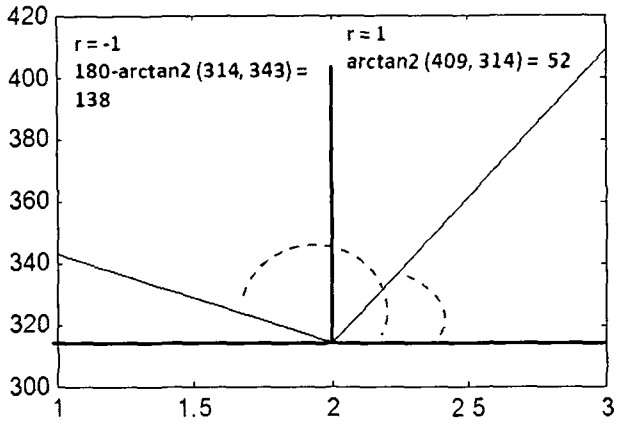


Figure 5.3: Degree of fluctuation for three expression values of a gene

$\{138, -1;\ 52, 1\}$.

5.4.3 Construction of co-expression network

This section discusses the counting of pair-wise support between genes using only *one pass* over the database to construct the co-expression network of connected genes. We use a correlogram matrix based approach[80] for computing similarity between two target genes based on the degree of fluctuation and regulation between them. Later, similarity values are used to calculate the support values needed to construct the co-expression network. We first transpose the preprocessed database (obtained using the above technique) by placing edges as rows and the genes as columns. We read each row from the database, and check whether two consecutive genes (say, G_i and G_j) satisfy the similarity criterion (in terms of degree of fluctuation and regulation information) or not, using Equations (5.1) and (5.2). If two genes are similar, the content of the correlogram matrix cell with index (i,j) is increased. This step is repeated for all pairs of genes in each row. This continues for all the rows to be processed.

From the correlogram matrix, it is very simple to extract the support count of gene pairs. Using these support counts, we compute all connected genes that satisfy the given θ constraints.

Our approach is good because (i) It constructs the network in single scan of database and hence it is faster, (ii) no discretization is needed, and (iii) our approach does not use any standard proximity measures.

Based on all strongly connected pairs, an adjacency matrix is computed as:

$$A(i,j) = \begin{cases} +1 & \text{if } G_i \text{ and } G_j \text{ are strongly connected and } S_k(E_{ij}) = +1 \\ -1 & \text{if } G_i \text{ and } G_j \text{ are strongly connected and } S_k(E_{ij}) = -1 \\ 0 & \text{otherwise} \end{cases} \quad (5.8)$$

where 0 indicates the lack of any relation between the genes. A gene co-expression network connecting various genes is constructed based on the adjacency matrix.

GeCON is given as an algorithm depicted in Algorithm 4. It takes preprocessed database D' and θ as input. Step 1 deals with construction of the correlogram matrix. In step 2 to 7, all connected genes are extracted based on θ and adjacency matrix is constructed using Equation (5.8). Finally, the algorithm returns the adjacency matrix A.

```
   input : D' (Preprocessed Database), θ (Support threshold)
   output: A (Adjacency matrix)
1  Generate correlogram matrix from D';
2  foreach gene pair (G_i, G_j) ∈ D' do
3  |  if then
4  |  |  (
5  |  end
6  |  G_i, G_j) is Strongly Connected wrt. θ;
7  |  Update adjacency matrix A with (G_i, G_j) and S_k(E_ij);
8  end
9  Return A;
```

Algorithm 4: The GeCON Algorithm

5.4.4 Complexity analysis

GeCON uses correlogram matrix for storing support for pair of genes. Thus for N genes, GeCON require fixed memory of size $SPACE_{GeCON} = O(N^2)$. In terms of computational cost, GeCON needs time for preprocessing and network construction

using correlogram matrix. For a dataset with N genes and C conditions, preprocessing step require $O(N * C^2)$ time. To construct network, it has to traverse the correlogram matrix. Thus, the time required for network construction is $O(N^2)$. The total computational cost of GeCON is:

$$\begin{aligned} Cost_{GeCON} &= O(N * C^2) + O(N^2) \\ &\approx O(N^2)(\text{compare to size of } N, \text{ normally } C \ll N, \text{ so we can ignore } C). \end{aligned}$$

5.5 Performance Evaluation

This section provides the details of experiments conducted, the data sets used and the validation of the results. We apply the GeCON on real and synthetic gene expression data consisting of publicly available seven benchmark gene expression datasets and thirteen *in silico* dataset. We used Java 1.6 running on a Windows 7, 2.53 GHz machine for implementation.

5.5.1 Dataset used

We used DREAM (Dialogue for Reverse Engineering Assessments and Methods) Challenge synthetic data on *in silico* regulatory network construction, provided by Marbach et al.[81]. Dream3 and Dream4 are the two Challenges that are available. Dream3 involves fifteen benchmark datasets, five each of various sizes (10, 50 and 100). The structures of the benchmark networks are obtained by extracting modules from real biological networks. At each size, two of the networks are extracted from the regulatory network of E. coli and Yeast. Dream4 is very similar to Dream3 containing a total of 10 networks, five of each size, 10 and 100. The *in silico* datasets generated based on Marbach et. al.[81] platform for our experiments are characterized in Table 5.1. We analyze the results from various real datasets for biological significance in terms of the GO annotation database. The details of

the datasets are presented in Table 5.2.

Table 5.1: *In silico* DREAM Challenge datasets

Challenges	Dataset	*In silico* network	Size of the network
	D1	Ecoli1	10
	D2	Ecoli2	10
	D3	Ecoli1	50
	D4	Ecoli2	50
Dream3	D5	Yeast1	10
	D6	Yeast2	10
	D7	Yeast1	50
	D8	Yeast2	50
	D9	insilico1	10
	D10	insilico2	10
Dream4	D11	insilico3	10
	D12	insilico1	100
	D13	insilico2	100

Table 5.2: Short description of the datasets

Organism	Dataset	No. of genes	No. of samples	Source
Yeast Sporulation	Yeast	474	7	http://cmgm.stanford.edu/ pbrown/sporulation
Yeast	Yeast_KY	237	18	http://faculty.washington.edu /kayee/cluster/
Yeast	Yeast cell cycle	384	18	http://faculty.washington.edu kayee/cluster
Human	GDS825	277	8	NCBI
Mouse	GDS958	308	12	NCBI
Rat	GDS3702 (Subset)	1000	12	NCBI
Rice	Thaliana	138	8	http://homes.esat.kuleuvan.be/ thijs/Work/clustering.html /kayee/cluster

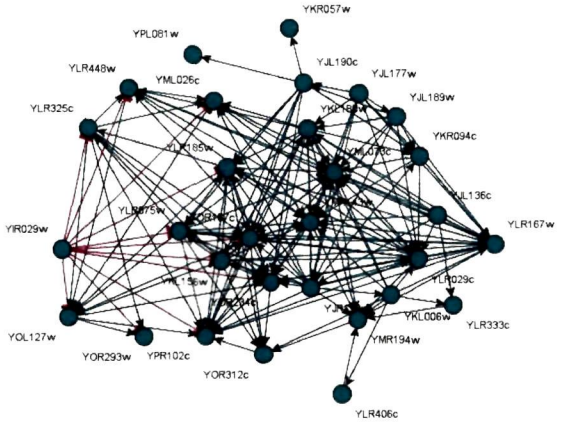
5.5.2 Experimental results

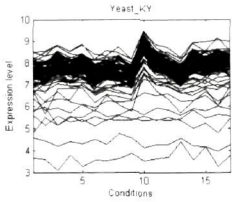
As discussed, we use the concept of support to draw links or inter-relationships among genes. A gene pair satisfying support criterion with respect to a user defined threshold θ is considered connected. We display only those genes that are

linked to others with support higher than the threshold. We use the *in silico* regulatory network construction platform provided by Marbach et al.[81] for simulation of our results. In the network, nodes represent genes and lines between nodes represent hypothesized associations among genes. A blue colored arrow head edge shows positive regulation, whereas a red colored blunt head edge indicates negative regulation between a pair of genes. We present some of the networks in Figure 5.4 and 5.5. The genes participating in a co-expression network form a group of coherent or co-expressed genes responsible for common biological activities. We consider such a group a module and analyse the biological significance of the modules in terms of the Gene Ontology in the next section. Figure 5.4 and 5 5 also show the profile plot of selected modules and the corresponding heat map of the modules. The cluster profile plot shows the normalized gene expression values of the genes within that cluster with respect to the conditions or time points for each co-expressed group. From the profile, it is evident that GeCON is able to detect both positively and negatively co-expressed gene groups as well as identify Scaling and Shifting patterns in the expression.

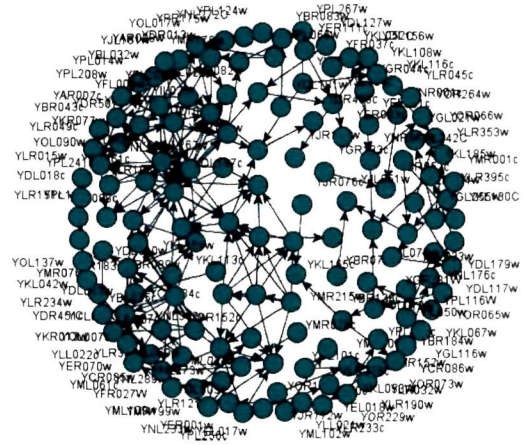
5.5.3 Biological significance

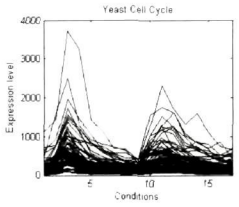
Biological significance of the results can be assessed by functional annotation of the genes participating in a module or cluster. We determine the biological relevance of the modules comprising of all the genes participating in a common co-expression network, in terms of p[4] and Q[82] values against statistically significant GO terms validated using the GO annotation database. In this annotation database, genes are assigned to three structured, controlled vocabularies (ontologies) that describe gene products in terms of associated biological processes, components and molecular functions in a species-independent manner. Statistical significance is evaluated for the genes in each group by computing p-values, which signify how well they match GO categories. A smaller p-value (close to zero) indicates better match which in turn indicates more close and compact cluster structure. For evaluating functional enrichment of a module in terms of p values we use FuncAssociate[83].

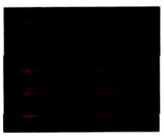




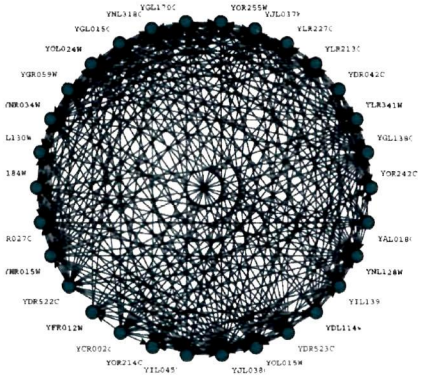
(a) Yeast_KY

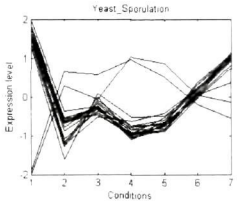






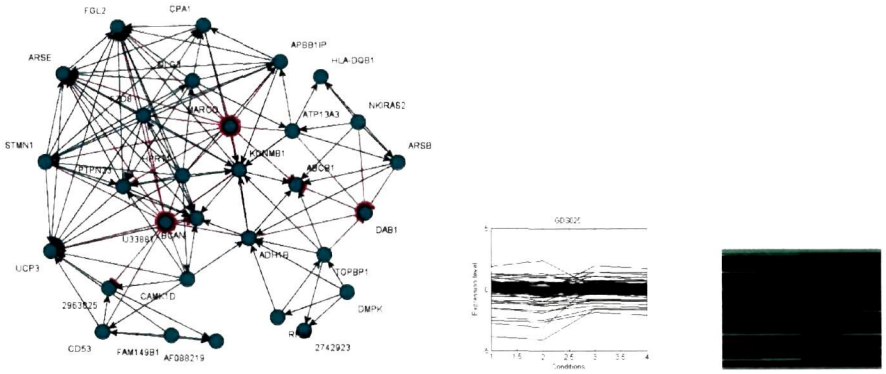
(b) Yeast Cell Cycle



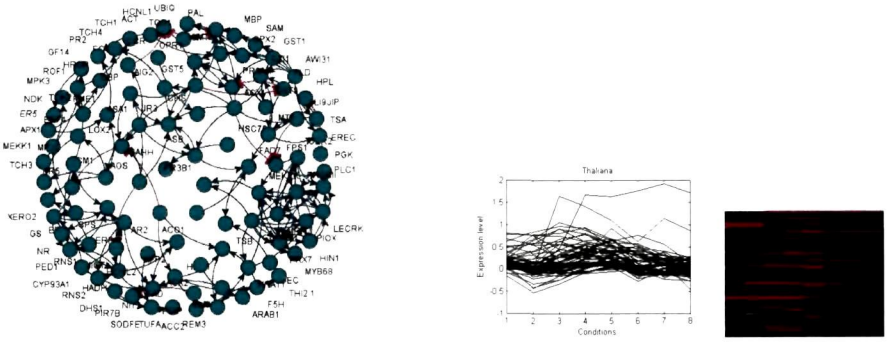


(c) Sporulation

Figure 5.4: Network, Module profile plot and heat map for each selected modules from three Yeast datasets



(a) GDS825



(b) Thaliana

Figure 5.5: Network, Module profile plot and heat map for each selected modules from Human and Thaliana datasets

The Q-value is the minimal False Discovery Rate (FDR) at which this gene appears significant. The GO categories and Q-values from a FDR corrected hypergeometric test for enrichment are obtained using GeneMANIA [84]. Q-values are estimated using the Benjamini Hochberg procedure[82]. We report here p and Q-values of selected modules from several datasets. Along with Q-values, GeneMania also provides Co-expression, Physical and Genetic interaction scores for the networks. The co-expression percentage indicates the level of similarity in expressions across conditions. On the other hand, the physical interaction percentage shows the level of protein-protein interaction within a module. In Table 5.3, we present results from GeneMANIA for selected modules. A module obtained from the Yeast Sporulation network is mainly responsible for cytosolic ribosome formation with Q score **1.11e-47** and it also exhibits good co-expression. On the other hand,

modules responsible for sporulation activities show very high expression but the protein-protein interaction is nil. A very high physical interaction can be observed in the module responsible for DNA replication. Kayee's Yeast dataset shows a very high Q value of **2.16E-130**. However, the same module shows very poor physical interaction. We also observe 100% co-expression from GDS3702 where modules are responsible for oxidoreductase activities, aging regulation and lipid catabolic process.

Table 5.3: Q-value, Co-expression and Physical interaction score for different modules from different datasets

Dataset	Module	GO-Annotation	Q Value	Co-expression(%)	Physical Interaction(%)
Sporulation	1	cytosolic ribosome	**1.11E-47**	74.71	7.24
	2	nucleolus	2.32E-30	72.46	8.96
	3	sporulation	9.87E-20	**96.96**	0
	4	DNA replication preinitiation complex	2.92E-09	3.07	**95.08**
Yeast_KY	1	cytosolic ribosome	**2.16E-130**	69.1	3.56
	2	structural constituent of ribosome	2.64E-126	69.1	3.56
	3	DNA-dependent DNA replication	2.38E-27	65.05	8.08
GDS3702	1	mitochondrial inner membrane	8.29E-07	68.5	5.41
	2	oxidoreductase activity	3.29E-02	**100**	
		aging	1.55E-01	**100**	
		regulation of			
	3	lipid catabolic process	1.40E-03	**100**	
	4	iron-sulfur cluster binding	5.51E-03	43.75	9.72
GDS958	1	vacuolar proton-transporting V-type ATPase complex	4.67E-16	27.59	32.75
	2	cell cortex	5.01E-03	27.59	32.75
Thaliana	1	negative regulation of cellular process	2.19E-04	29.41	29.41
	2	response to wounding	1.36E-08	92.48	5.63
	3	receptor binding	2.49E-03	29.41	29.41

Table 5.4 presents p scores obtained by FuncAssociate for selected modules

submitted from different datasets. For Kayee's dataset, GeCON shows better performance in terms of high enrichment with p-value, e.g., p-value **5.20E-96**. Similarly, GDS825, GDS958 and Sporulation datasets also contain modules with good functional enrichments.

Table 5.4: p-values for different modules from different datasets

Dataset	Module	GO Annotation	p value
GDS825	1	folic acid and derivative biosynthetic process	**3.10E-15**
	2	cullin-RING ubiquitin ligase complex	5.40E-08
	3	chemoattractant activity	5.60E-07
	4	biotin binding	8.30E-07
Yeast_KY	1	cytosolic ribosome	**5.20E-96**
	2	DNA replication	9.64E-20
GDS3702	1	response to neutrient	1.47E-05
	2	hydrolase activity	1.60E-05
	3	protein complex	8.00E-04
GDS958	1	intracellular part	**9.83E-19**
	2	intracellular membrane-bounded organelle	2.57E-05
Sporulation	1	cytoplasmic translation	**2.22E-22**
	2	anatomical structure formation	1.25E-17
	3	ribonucleoprotein complex biogenesis	1.07E-10
	4	cell cycle phase	2.36E-06
	5	cellular component assembly	4.66E-06

5.5.4 Performance comparison

We compare our predictions using DREAM Challenge dataset with three well known gene regulatory network reconstruction algorithms, ARACNE[73], CLR[72] and MRNET[74]. R implementation of all the three algorithms are available in[85]. Prediction effectiveness is compared against the actual networks generated from *in slico* DREAM Challenge data, using three different metrics for evaluating accuracy: AUPvR (Area under Precision vs Recall curve), AUROC (Area under Receiver-Operator Characteristics curve) and F_β score. The ROC is also known as a relative operating characteristic curve, because it is a comparison of two oper-

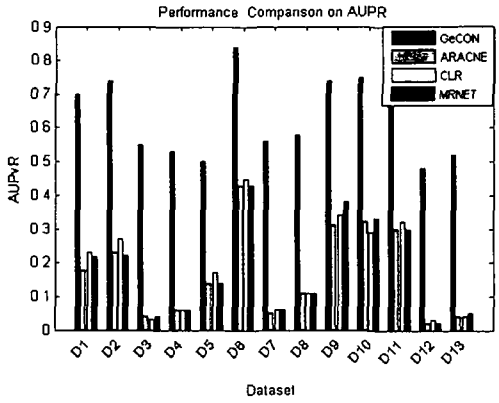
ating characteristics (True Positive Rate and False Positive Rate) as the criterion changes[86]. ROC curves may not be the appropriate measure when a dataset contains large skews in the class distribution, which is commonly the case in transcriptional network inference. As an alternative, precision vs. recall (PvR) curves are considered for measuring prediction accuracy[87]. ROC curves are commonly used to evaluate prediction results. However, PvR curve may be more sensitive when there is a much larger negative set than positive set. Computing the area under the curve (AUC) of a ROC or PvR is a way to reduce ROC or PvR performance to a single value, representing expected performance. A compact representation of the PvR diagram is the maximum and/or the average F score[88], which is a harmonic average of precision and recall. The general formula for non-negative real β is:

$$F_\beta = (1+\beta^2)\frac{precision \times recall}{(\beta^2.precision) + recall} \tag{5.9}$$

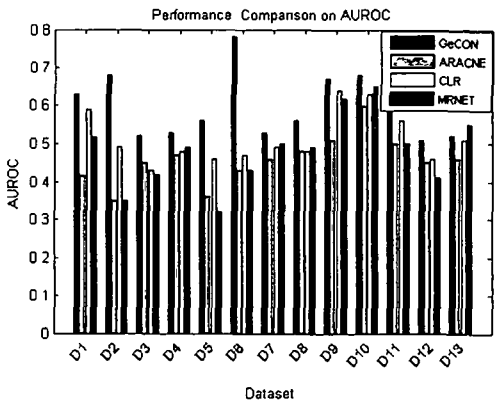
Two other commonly used F measures are the F_2 measure, which weights recall higher than precision, and the $F_{0.5}$ measure, which puts more emphasis on precision than recall. The F-measure measures the effectiveness of retrieval assuming recall is β times more important than precision. In our experiments we preferred $F_{0.5}$ score. Prediction effectiveness of GeCON is compared with other algorithms and the results are shown in Figure 5.6.

From the figures it is evident that GeCON outperforms all other algorithms in terms of network prediction on three different scores. In case of dataset $D6$, GeCON achieved a very high AU(PvR) score of .84, AUROC of .78 and F_β score of .86. Other algorithms exhibit consistent and almost similar trends in all experiments. To justify our claim on one-pass nature of GeCON, which is fast in general, we perform execution time comparison of GeCON with ARACNE. Due to unavailability of executable codes of all other target algorithms on a Java platform, we used only the Java version of the original ARACNE code [a] for comparison with GeCON. The result given in Figure 5.7 clearly shows that GeCON is much faster than ARACNE.

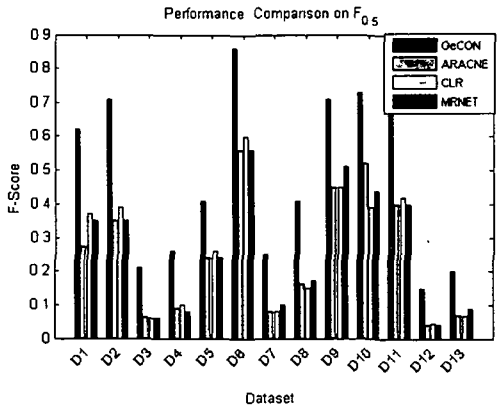
[a] http://wiki.c2b2.columbia.edu/califanolab/index.php/Software/ARACNE



(a) AU(PvR) curve of different algorithms



(b) AUROC curve showing prediction performance



(c) Comparative $F_{0.5}$ scores

Figure 5.6: Performance comparison of four algorithms on *in silico* dataset

5.6 Discussion

In this chapter, we have presented an effective gene co-expression network finding algorithm called GeCON for discovering biologically related gene pairs that may form a network of co-expressed genes. We have established that the genes participating in a network have similar functional behaviour. The GeCON algorithm exploits a fast correlogram matrix based technique for capturing the support of each gene pair in order to compute the relationship between gene pairs. Gene pairs with strong relationship are used to construct the association network. When constructing the networks, GeCON exploits the regulation relationship among the genes. We have shown that GeCON performs well in determining the gene-gene re-

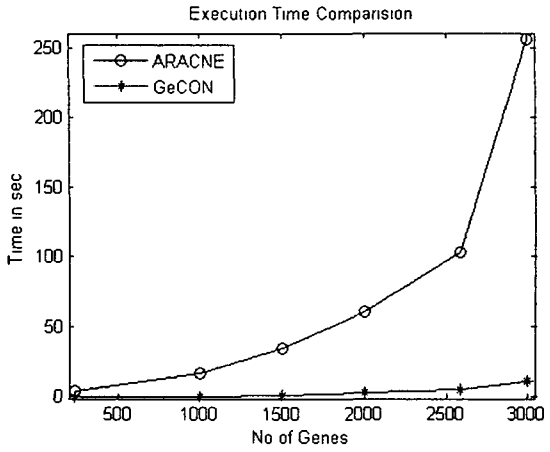


Figure 5.7: Execution time comparison of GeCON with ARACNE on different sized networks

lationship network in both *in silico* and real datasets. Our literature survey reveals that most existing techniques do not emphasize on computational efficiency. We report results to show that GeCON is effective to predict *in silco* networks based on DREAM Challenge data. We validate the claims that the simple gene-gene relation based co-expression networks are capable of detecting biologically significant set of genes. We provide results to show that co-expressed groups formed from the network have high biological significance. Moreover, we further establish that the simple expression pattern matching is helpful in finding biologically relevant genes. Gene co-expression networks can be used further to predict more complex biological networks.

Chapter 6

Pattern Based Approach for Co-Regulated Biclustering of Gene Expression Data

Co-regulation is a common phenomenon in gene expression. Finding positively and negatively co-regulated gene clusters from gene expression data is a real need. Existing techniques based on global similarity are unable to detect true up- and down-regulated gene clusters. This chapter presents an expression pattern based biclustering technique, CoBi, for grouping both positively and negatively regulated genes from microarray expression data. Regulation pattern and similarity in degree of fluctuation are accounted for while computing similarity between two genes. Unlike traditional biclustering techniques, which use greedy iterative approaches, it uses a *BiClust* tree that needs single pass over the entire dataset to find set of biologically relevant biclusters. Biclusters determined from different gene expression datasets by the technique show highly enriched functional categories.

6.1 Introduction

In the last two decades, clustering has become a popular data-analysis tool in genomic studies, particularly in the context of gene-expression microarrays[89,90,91,92].

Each microarray provides expression measurements for thousands of genes and clustering is a useful exploratory technique to analyze gene expression data since it groups similar genes together and allows biologists to identify groups of potentially meaningful genes which have related functions or are co-regulated, which in turn helps in finding the relationships among them in the form of gene regulatory networks[4]. Another common use of cluster analysis is the grouping of samples (arrays) by relatedness in expression patterns, i.e., finding groups of co-expressed genes.

A cluster is a group of objects that are similar to one another within the group but dissimilar to the objects of other groups[93,94]. Clustering normally partitions genes into disjoint groups according to the similarity of their expressions across all conditions. However, it has frequently been observed that subsets of genes are co-regulated and co-expressed under a subset of environmental conditions or time points[95]. Biclustering algorithms tackle the problem of finding a set of submatrices where each submatrix or bicluster meets a given homogeneity criterion. This special instance of clustering was originally introduced by Hartigan[96] and later applied by Cheng and Church[20] in expression data to capture the coherence of a subset of genes and a subset of conditions. Several techniques have been proposed so far to find quality biclusters from expression data. Below we present a brief discussion on some of the techniques already proposed.

6.2 Related Work

In Cheng and Church's approach, the degree of coherence is measured using the concept of mean squared residue (MSR) and the algorithm greedily inserts/removes rows and columns to arrive at a certain number of biclusters achieving some predefined residue score. The lower the score, stronger the coherence exhibited by the bicluster, and better is the quality of the bicluster. Followed by Cheng and Church, a number of biclustering techniques have been proposed[20,75,79,97,98,99,100,101,102,103,104] to determine quality biclusters.

A greedy iterative search[20,97] based approach finds a local optimal solution with

an expectation to obtain finally a globally good solution. A divide and conquer[96] approach divides the whole problem into sub-problems and solves them recursively. Finally, it combines all the solutions to solve the original problem. In exhaustive biclustering[79], the best biclusters are identified using exhaustive enumeration of all possible biclusters extant in the data, in exponential time. A detailed categorization of heuristic approaches is available in[98]. A number of techniques based on metaheuristics such as evolutionary and multi-objective evolutionary framework have been explored[99] while generating and iteratively refining an optimal set of biclusters. All of them use MSR as the merit function.

An MSR based technique is effective in finding optimized maximal biclusters. From a biological point of view, the interest resides in finding biclusters with subsets of genes showing similar behaviour and not similar values. Interesting and relevant patterns from a biological point of view, such as shifting and scaling patterns may not be detected using this measure as it considers only expression values, not the pattern or tendency of gene expression profile. It is important to discover this type of patterns because, frequently the genes can present similar behaviour although their expression levels vary in different ranges or magnitudes. Aguilar-Ruiz[75] proved that the MSR is not a good measure in discovering patterns in data when the variance of gene values is high, that is, when the genes present scaling and shifting patterns. To detect biologically relevant biclusters with scaling and shifting patterns, a scatter search based approach is proposed[100]. This method uses a fitness function based on the linear correlation among genes and an improvement method to select just positively correlated genes.

Often, it has been observed that genes share local rather than global similarity in their gene expression profile and only under a few conditions or time points. Thus, correlation based technique may not be effective while deciding pair wise similarity between two gene expression profiles. Other than that, various pattern-based approaches have also been proposed[101,102,105,106] for discovery of biclusters where expression levels of genes rise and fall in a subset of conditions or time points.

Recently it has been observed that[76] co-regulated genes also share negative patterns or inverted behaviours, which existing pattern based approaches are unable to detect.

6.3 Motivation

Biological processes are regulated in many ways. Examples include the control of gene expression, protein modification or interaction with protein or substrate molecules. Expression patterns with similar tendency or behaviour are normally termed positively regulated and inverted behaviour as negatively regulated. As described in Amigo[a], negative regulation or down regulation stops, prevents, or reduces the frequency, rate or extent of a biological process and positive regulation or up-regulation does the reverse. To illustrate the fact we consider examples of co-regulated clusters from real microarray Human datsets, GDS825 given in NCBI[b]. The profile plot is given in Figure 2.4. From the figure, we easily observe that genes GALNT5 and IDH3B show similar pattern or positive co-expression patterns. On the other hand IDH3B or GALNT5 showing inverted or negative pattern with APOE. As suggested by gene ontology three genes are involved in *regulation of plasma lipoprotein particle levels* and *triglyceride-rich lipoprotein particle remodeling*. More prominent inverted or negative patterns can be observed in Figure 5.1 taken from NCBI Rat dataset GDS3702. As mentioned earlier, both the genes are responsible for *regulation of interferon-beta production*. A group of genes may share a combination of both positive and negative co-regulation under a few conditions or at some time points. A majority of existing approaches try to capture genes with similar tendency.

In this work, we capture biclusters of both positively and negatively regulated genes as co-regulated genes. Moreover, as mentioned in[107], a bicluster can be considered a quality bicluster when participating genes exhibit consistent trends and similar degrees of fluctuation under consecutive conditions. We consider both

[a] *http://amigo geneontology.org/cgi-bin/amigo/term_details?term=GO:0048519*
[b] *www.ncbi nlm.nih.gov*

up- and down-regulation trends and similar degree of fluctuations under consecutive conditions for expression profiles of two genes as a measure of similarity between the genes. We use a new BiClust tree for generating biclusters in polynomial time that needs single pass over the dataset.

6.4 Biclustering of co-regulated genes

Let $G = \{G_1, G_2, \cdots G_N\}$ be a set of N genes and $R = \{T_1, T_2, \cdots, T_M\}$ be the set of M conditions or time points of a microarray dataset. Given a gene expression dataset $D_{N \times M}$, biclusters can be defined as follows.

Definition 6.4.1 (Biclusters) : A set of sub-matrices $\{(I_1, J_1), \cdots, (I_k, J_k)\}$ of the matrix $D = (N, M)$ (with $I_i \subseteq N$, $J_i \subseteq M$ $\forall i\{1, \cdots, k\}$), where each submatrix (bicluster) meets a given homogeneity criterion.

Unlike the usual clustering of genes, biclustering tries to cluster a set of genes which are similar under a subset of conditions or time points. Traditional biclustering techniques normally use global similarity measures such as Euclidean distance, Pearson correlation or MSR. These measures sometimes fail to capture the true grouping. On the other hand, most existing techniques have been found to give less emphasis to pattern matching based on local similarity. It has been observed that the genes share local rather than global functional similarity in their gene expression profiles. Moreover, they share co-regulation in terms of up- and down-regulation. While computing the similarity, well known techniques do not consider positive or negative regulation pattern as co-expression or co-regulations which having biological significance. We consider both positive- and negative- regulation as co-regulation. In this chapter, we develop a local expression pattern matching based approach to find biclusters among co-regulated genes. The following terminologies are used to describe the proposed technique.

6.4.1 Terminology used

Definition 6.4.2 (Pattern Similarity) : Given degrees of fluctuation $A = \{a_1, a_2, \cdots, a_{M-1}\}$ and regulation patterns $R = \{r_1, r_2, \cdots, r_{M-1}\}$ of a gene, derived from gene expression profile, two genes' k^{th} expression patterns are similar if the difference in degrees of fluctuation of two genes' k^{th} edge is less than some given threshold τ. In order to compute the differences in the degrees of fluctuation, we consider two cases: when the regulation patterns are the same (in case of up regulation) and when the patterns are different (in case of down regulation) under a particular edge. Mathematically it can be defined as follows:

$$sim(G_{ik}, G_{jk}) = \begin{cases} 1 & \text{if } |G_i(a_k) - G_j(a_k)| < \tau \\ & \text{when } G_i(r_k) = G_j(r_k) \\ & \text{and if } |180 - G_i(a_k) + G_j(a_k)| < \tau \\ & \text{when } G_i(r_k) \neq G_j(r_k) \\ 0 & \text{Otherwise.} \end{cases} \tag{6.1}$$

Definition 6.4.3 (Co-regulated bicluster) : Given a gene expression dataset D of N genes and C conditions, a co-regulated bicluster is a sub-matrix of n genes and c conditions where the number of genes, n satisfies a user specified $MinGene$ criterion and the number of edges, c, in the bicluster is greater than a threshold, θ, and all pairs of gene in the bicluster satisfy pattern similarity (sim) across all c edges.

$$CorBiClust(D_{N\times C}, MinGene, \theta) = \begin{array}{l} \{D_{n\times c} | \forall G_{i=1 \ n} \in D_{n\times c}, |n| > MinGene, |c| > \theta \\ \wedge \ sim(G_{ik}, G_{jk}) = 1, \forall k = 1 \cdots (c-1)\}. \end{array} \tag{6.2}$$

Table 6.1: Sample Yeast gene expression dataset

ORF	T1	T2	T3	T4	T5	T6	T7
G1	248	294	399	438	451	364	366
G2	343	314	409	426	455	366	401
G3	304	309	289	289	330	283	309

6.4.2 Preprocessing

Now, we discuss the preprocessing steps involved in capturing the degree of fluctuation and regulation pattern for each expression profile. We compare two gene expressions both in terms of degree of fluctuation[78] and pattern of regulation between two adjacent conditions (edges), simultaneously. To capture both regulation pattern and degree of fluctuation of each gene, we use the same preprocessing technique as discussed in section 5.4.2 from the previous chapter. We represent regulation information as a triplet of values [1, 0, -1] to denote up-regulation, no change and down regulation, respectively. The regulation value in the k^{th} edge of a gene G_i , $G_i(r_k)$, based on two consecutive conditions (say, O_{k-1} & O_k) is calculated as:

$$G_i(r_k) = \begin{cases} 1 & \text{if } O_{k-1} < O_k \\ 0 & \text{if } O_{k-1} = O_k \\ -1 & \text{if } O_{k-1} > O_k. \end{cases} \tag{6.3}$$

The corresponding preprocessed data from sample Yeast dataset given in Table 6.1 is shown in Table 6.2. The columns in table represent the edges between two consecutive expression values from original dataset.

Table 6.2: The transformed expression dataset after preprocessing

Gene	E1	E2	E3	E4	E5	E6
G1	49,1	53,1	47,1	45,1	142,-1	45,1
G2	138,-1	52,1	46,1	46,1	142,-1	47,1
G3	45,1	137,-1	45,0	48,1	140,-1	47,1

To find co-regulated biclusters based on pattern similarity we use a BiClust tree based technique. The main advantage of the proposed technique is that it requires

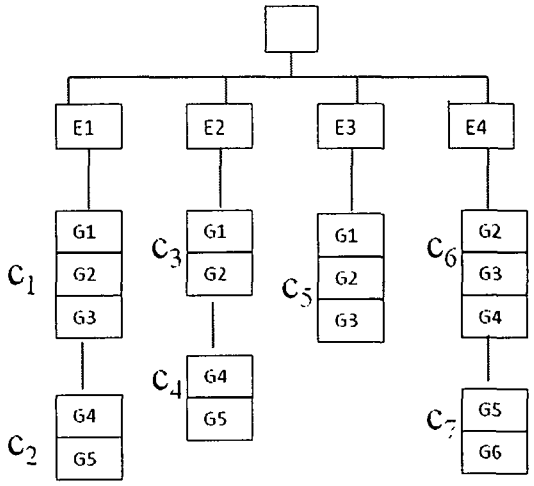


Figure 6.1: Initial BiClust tree

only single scan of the database for finding biclusters.

The following section discusses the way to find co-regulated biclusters.

6.4.3 Co-regulated biclustering using BiClust tree

BiClust tree is a m-way tree where each non-leaf node represents an edge or a set of edges and a leaf node represents gene or a group of genes that are co-regulated or co-expressed under the edge or set of edges. CoBi starts with creating an initial BiClust tree as shown in Figure 6.1.

In the figure, four edges are shown as non-leaf nodes $E1, E2, E3$ and $E4$. We use the dataset D' to construct the initial BiClust tree, BT. D' is a transformed dataset generated from original dataset D to capture degrees of fluctuation and regulation information from the expression pattern of each gene. The initial BiClust tree contains $(M - 1)$ number of edges as initial non-leaf nodes for a dataset with M number of conditions or time points. The leaf nodes are created by forming a k^{th} cluster of genes based on similarity of genes under the k^{th} edge by using Equation (6.1). For each gene, it tries to form a cluster with other genes belonging to a particular cluster. Otherwise, it creates a new cluster when there are no matching clusters. Thus, multiple clusters or leaf nodes may be formed under a particular edge. The same process is repeated for all the edges. $G1, G2$ and $G3$

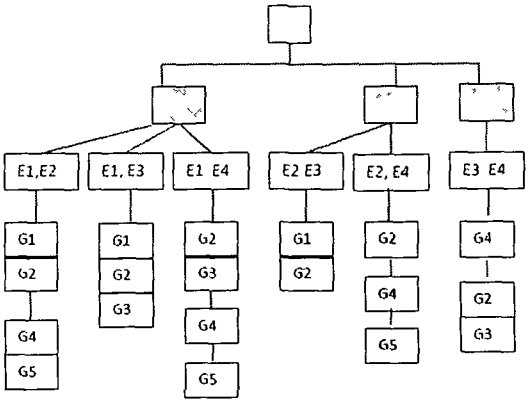


Figure 6.2: BiClust tree after expanding initial tree

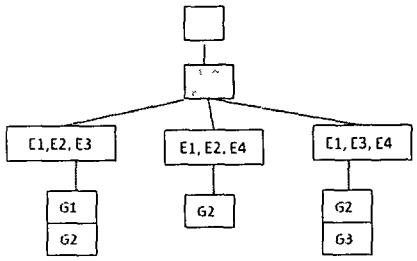


Figure 6.3: Final BiClust tree

form a cluster C_1, whereas $G4$ and $G5$ form another cluster C_2 under $E1$. While forming the k^{th}cluster, we transpose the dataset D', so that each row represents the degree of fluctuation and regulation pattern of all genes under each edge. By doing this we can compare easily all gene's expression patterns under k^{th} edge. Thus, for creating the initial BiClust tree, it requires a single pass over the dataset. No further consultation of the dataset is required in the following steps. To maintain a moderate number of gene clusters under an edge or a set of edges, it performs a pruning step. Cluster C_i is pruned if the cluster size is less than a user given threshold θ. Next, BT is expanded to get biclusters using *ExpandCluster* function. The proposed technique, CoBi is shown in Algorithm 5.

In the cluster expansion phase, iteratively tree branches are merged to get higher order biclusters. While merging two sub-trees, we apply merging in two ways, one at a non-leaf level and the other at the cluster level. Thus, from the initial BiClust tree, edges $E1$ and $E2$ are combined to form a new node $\{ E1, E2 \}$. Next, cluster leaf nodes under both nodes $E1$ and $E2$ are merged to get a new cluster node for $\{E1, E2 \}$. The cluster C_1 is compared with C_3 and C_4. A new cluster node $[G1, G2]$ is formed with all the elements that are common in both C_1 and C_3 or C_1 and C_4. In other words, it performs a intersection operation between two clusters. Since the number of genes in a dataset is normally high compared to the number of conditions, the cluster list in the subtree is expected to be large. This is more

critical especially in the initial stages of the tree. To handle the situation, we use a bit vector for storing gene IDs as a cluster. For merging we use bitwise AND operation. It is very fast compared to perform normal intersection between two clusters. In order to merge two non-leaf edges, we use the concept of union taken from[27]. The BiClust tree thus formed after the expansion of the initial BiClust tree is shown in Figure 6.2. The clusters that do not contain a minimum number of genes are pruned from the tree. During the merging of clusters under a non-leaf node, there may be a chance that a new cluster is formed such that its superset cluster is already present under the same non-leaf. Such subsets are redundant and removed. The process of sub-tree expansion continues until no further expansion is possible and all the biclusters are stored in a list with a minimum number of condition θ. After the final expansion of a sub-tree, the biclusters are extracted from the list. The same process is applied to all the sub-trees in BiClust tree. A final BiClust tree is shown in Figure 6.3 where the minimum number of genes is two. The node $\{E1, E2, E4\}$ will be pruned from the final tree as it contains a cluster with size one only. Other nodes are not shown in the final tree as they will be pruned as well. The biclusters formed are: $\{E1, E2, E3\}$ $[G1, G2]$, $\{E1, E3, E4\}$ $[G2, G3]$.

input : D' (Transformed Dataset), MinGene (Minimum number of Gene), θ (Minimum number of edge)
output: BiClust (List of Biclusters)

1 Construct initial BiClust tree, BT;
2 Prune cluster C_i from BT, if $|C_i| <$ MinGene;
3 BiClust = `ExpandCluster` (BT, MinGene,θ) ;
4 BiClust = `RemoveSubCluster` (BiClust);

Algorithm 5: CoBi: Co-regulated Biclustering

The proposed method is shown in a compact manner in Algorithm 5. At first, CoBi, constructs an initial BiClust tree using the transformed database D'. The initial BiClust tree is pruned based on user threshold *MinGene*. Next, the algorithm iteratively expands the tree to get all biclusters. The expand procedure is given in Algorithm 6. Two subtrees are merged and pruned when the number of genes in the

```
input  : BT (BiClust tree), MinGene (Minimum number of Gene), θ
         (Minimum number of edge)
output: BiClust (List of Biclusters)
1  Create a new BiClust tree BT' ;
2  foreach non-leaf node E_i = 1 → E_{n-1} of BT do
3  |  Create a subtree S of BT' ;
4  |  foreach non-leaf node E_j = E_{i+1} → E_n of BT do
5  |  |  V = Merge(E_i, E_j, MinGene) ;
6  |  |  Prune subset of V ;
7  |  |  Add V to S ;
8  |  end
9  |  Add S to BT';
10 end
11 foreach subtree S_i of BT' do
12 |  if S_i can expands further then
13 |  |  BiClust = BiClust ∪ ExpandCluster(S_i, MinGene, θ);
14 |  else
15 |  |  return GetBiClusters(S_i, θ);
16 |  end
17 end
```

Algorithm 6: ExpandCluster

merged tree is less than *MinGene*. Once the subtree reaches the end of expansion so that no further merging is possible, it then extracts biclusters from the final BiClust subtree. The same process repeated for all subtrees. At the end, *ExpandCluster* sub-function returns list of all biclusters generated. The biclusters returned may contain some redundant clusters, where genes in the clusters are same, however, conditions or time points are subset of the other. *RemoveSubCluster* function takes the list of biclusters and eliminate such clusters from the final list.

6.4.4 Complexity analysis

The complexity of the biclustering problem depends on the exact problem formulation, and particularly on the merit function used to evaluate the quality of a given bicluster. However most interesting variants of this problem are NP-complete requiring either large computational effort or the use of lossy heuristics to short circuit the calculation[98]. Our approach deterministically finds all biclusters using

a non-greedy approach in a polynomial time. The cost of our algorithm consists of two parts: initial BiClust tree construction from D' (C_{IB}) and the cost for expanding the BiClust tree and extracting biclusters (C_{EX}).

(a) Construction of initial BiClust tree: Let us assume that the preprocessed dataset D', contains N genes and M edges. So, to scan the database, the cost is $(M * N)$. For creating clusters under a edge node it requires calculation of pattern similarity among all genes under an edge. Thus, the time requirement for creating clusters is N^2. The total time complexity for construction of initial BiClust tree is $C_{IB} = O(M * N^2)$.

(b) BiClust tree expansion: Let us consider that the maximum number iterations for the algorithm is k, which is the number of edges in the final bicluster. Let ζ be the number of edges or non-leaf nodes per iteration and the number of clusters under an edge be C. Now, the cost of merging two clusters is $O(C^2)$. We observe that with the increase in k, normally C decreases. The reason behind this is that compared to the number of clusters in $(k-1)$ steps fewer clusters take part in the intersection in k^{th} step. Thus the worst case complexity for bicluster expansion is no more than $C_{EX} = O(k * \zeta * C^2)$.

Most real microarray datasets contain large numbers of genes compare to number of conditions. Scanning of the database is the costly activity. All though the complexity of the algorithm is polynomial, however compared to the cost of database scanning, it is negligible.

In the next section, we establish how co-regulated biclusters are relevant from a biological point of view.

6.5 Performance Evaluation

This section provides the details of the experiments conducted, the data sets used and the biological validation of the results. We used Java 1.6 running on a Windows 7, 2.53 GHz machine for implementation.

6.5.1 Dataset used

We applied our biclustering approach on nine benchmark gene expression datasets. Since it is difficult to present all results, we present some of the significant results from each dataset generated by the approach. The details of the datasets are given in Table 6.3.

Table 6.3: Short description of the datasets

Organism	Dataset	No. of genes	No. of samples	Source
Yeast	YeastDB	2884	17	http://arep.med.harvard.edu/biclustering/yeast.matrix
	Sporulation	474	7	http://cmgm.stanford.edu/pbrown/sporulation
	Yeast_KY	237	17	http://faculty.washington.edu/kayee/cluster/
	YeastCho (cell cycle)	384	17	http://faculty.washington.edu kayee/cluster
Rat	Rat_CNS	112	9	http://faculty.washington.edu/kayee/cluster
Human	GDS3712	325	12	NCBI
	Fibroblast Serum	517	13	http://www.sciencemag.org/feature/data/984559.hsl/
Mouse	GDS958	308	12	NCBI
Rice	Thaliana	138	8	http://homes.esat.kuleuven.be/šistawww/bioi/thijs/Work/Clustering.html

6.5.2 Experimental results

We analyze the results in terms of biological significance with the help of the GO annotation database and cluster profile plots. In Figure 6.4, we present the profile plot of some of the obtained biclusters. From the figure it is clearly evident that positive and negative co-regulations are common in biological data and it is well captured by our approach.

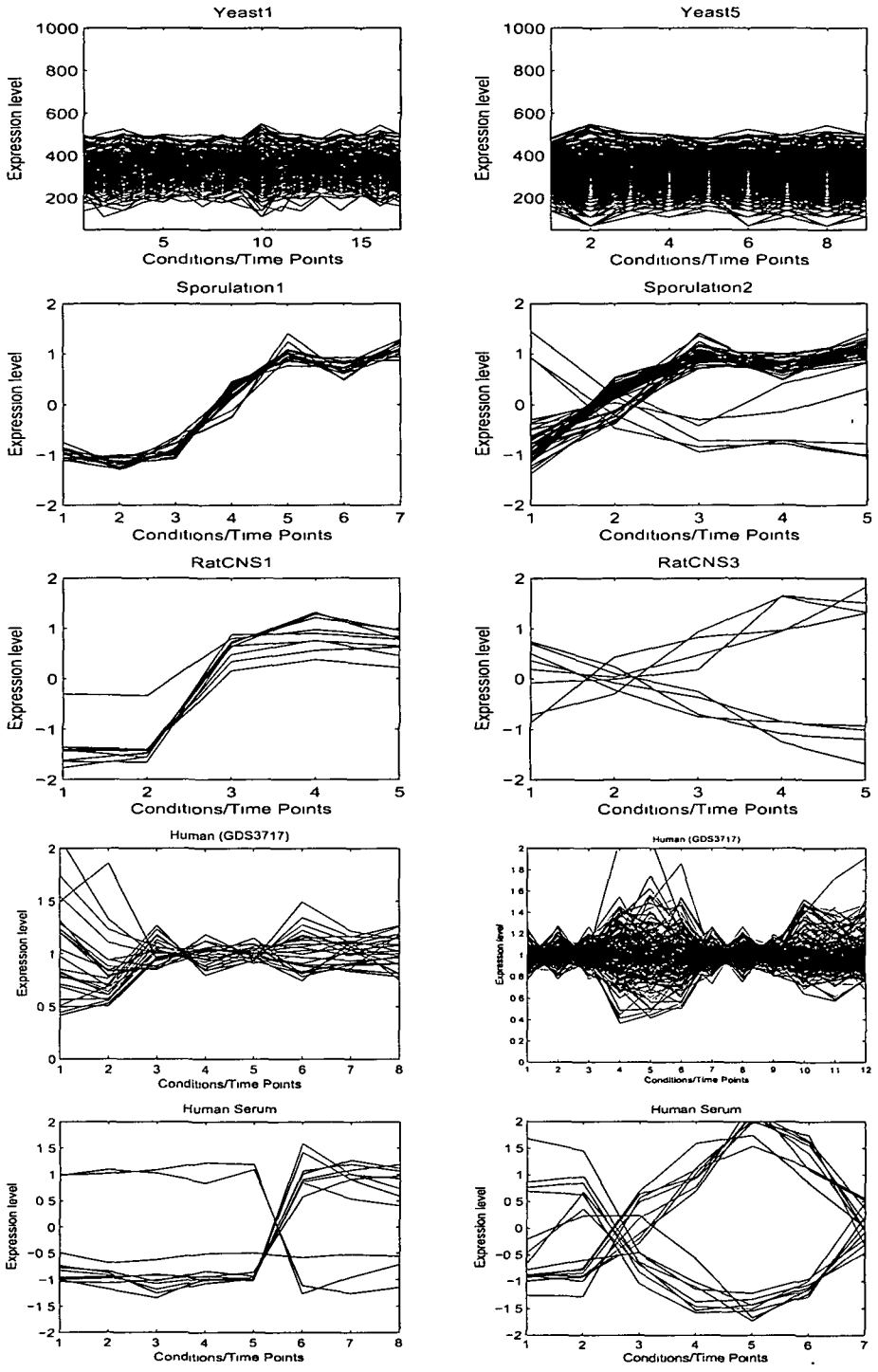


Figure 6.4: Expression profile plots of biclusters from Yeast, Yeast Sporulation, RatCNS, GDS3717 and Fibroblast Serum data

6.5.3 Biological significance

We use gene ontology (GO) and compute p-values to evaluate the results. To determine the statistical significance of the association of a particular GO term with a group of genes in a cluster, we use various online tools from the GO Project[a]. These tools use the hypergeometric distribution to calculate the p-value, which evaluates whether the clusters have significant enrichment in one or more function groups. In our experiments we use the following tools: FuncAssociate[b], Fatigo[c], GOTermFinder[d] and OntoExpress[e]. Table 6.4 shows the information on selected biclusters from the different datasets obtained by applying our biclustering technique. For each bicluster an identifier of the bicluster, the number of genes, the number of conditions, the volume and MSR score are presented. The MSR score is reported to establish a comparison of the quality of biclusters with other algorithms. We also report Q value and the associated GO terms out for some of the functionally enriched groups provided by online tool GeneMANIA[84] in Table 6.5.

To evaluate biological significance of the results produced by our technique in terms of associated biological processes, cellular components, and gene function, we applied Yeast GO term finder to some of the biclusters from sporulation data. Of 22 genes from the cluster *Spo1*, the genes {YDR523C, YLR227C, YGR059W, YDR218C, YGL170C, YLR341W, YJL038C, YLR213C} are involved in the process of sporulation, anatomical structure formation involved in morphogenesis and cell differentiation, while genes {YDR523C, YGL170C, YLR341W, YGR059W, YLR213C, YDR218C} are involved in sexual reproduction and sexual sporulation process resulting in formation of a cellular spore. On the other hand genes {YCR002c, YGR059W, YDR218C} are involved in GTP binding and guanyl ribonucleotide binding and genes {YGL170C, YCR002c, YLR227C, YGR059W, YDR218C} take part in structural molecular activity. With respect to cellular component ontology, terms associated with genes {YDR523C, YCR002c, YGR059W,

[a]http://www.geneontology.org
[b]http://llama.mshri.on.ca
[c]http://fatigo.bioinfo.cnio.es
[d]http://go.princeton.edu, http://db.yeastgenome.org/cgi-bin/GO/goTermFinder
[e]http://vortex.cs.wayne.edu

Table 6.4: Biclusters results from Yeast, Sporulation and Rat CNS data

Dataset	Bicluster Id	No. of Gene	No. of Cond.	Volume	MSR	p value	GO attribute
Yeast	Yeast1	268	17	4556	654.41	2.075e-9	Cytoplasmic translation
	Yeast2	343	15	5145	664.20	3.318e-7	Ribosome
	Yeast3	430	13	5590	608.91	8.960e-7	Structural constituent of ribosome
Sporulation	Spo1	22	7	154	0.01557	4.543e-9	Cellular development process
	Spo2	69	5	345	0.1285	4.476e-19	Anatomical structure formation for morphogenesis
Rat CNS	RatCNS1	9	5	45	0.051	6.81e-4	Male sex determination
	RatCNS3	12	4	48	0.233	4.71e-4	Insulin receptor substrate binding

YDR218C} are ascospore-type prospore, intracellular immature spore, prospore membrane, septin complex. Similarly, from *Spo2* ({YDR523C, YGR225W, YLR227C, YPL027W, YLR343W, YDR516C, YDR218C, YNL204C, YGL170C, YIL099W, YCR002c, YDR260C, YJL038C, YLR213C, YOR242C, YNL225C, YGR059W, YLR054C, YNL128W, YOL132W, YLR308W, YMR017W, YLR341W}), the most significant biological processes are sporulation and anatomical structure formation involved in morphogenesis with p-value 4.476e-19. GO terms observed in molecular function categories are glucanosyltransferase activity and 1,3-beta-glucanosyl transferase activity. In case of cellular components, genes {YDR523C, YMR017W, YCR002c, YGR059W, YLR314C, YPL027W, YLR054C, YDR218C} are involved in prospore membrane, intracellular immature spore and ascospore-type prospore formation. In case of YeastKY dataset, it is observed that majority of the genes are involved in ribosome constituent activity with Q value **1.01e-119** (Table 6.5).

To verify the biological significance of the results from RatCNS data, we submitted our resulting biclusters to Onto-Express, and obtained a hierarchy of functional annotations in terms of GO for each cluster. An example of the GO tree for a co-

Table 6.5: Q-values and GO attributes from different biclusters

Dataset	Bicluster Id	Q value	GO attribute
GDS958	Mouse1	2.18e-12	cytosolic part and ribosomal subunit formation
	Mouse2	5.57e-7	nuclear DNA-direct RNA polymerase complex
	Mouse3	1.76e-6	proteasome complex
Rat CNS	Rat1	1.82e-14	regulation of neuron apoptosis
	Rat2	3.59e-14	regulation neurological system process
	Rat3	1.14e-13	positive regulation of glucose import
	Rat4	5.27e-10	growth factor binding
YeastCho	Cho1	4.03e-10	chromosomal part
	Cho2	2.38e-10	DNA repair
	Cho2	4.23e-6	protein glycosylation
Sporulation	SP1	4.48e-19	anatomical structure formation
	SP2	8.86e-18	cellular component assembly involved in morphogenesis
	SP3	4.54e-9	cellular developmental process
YeastKY	KY1	**1.01e-119**	Structural constituents of ribosome
	KY2	**1.83e-110**	ribosome
Thaliana	Th1	4.19e-13	glutathione transferase activity
	Th2	6.69e-08	toxin catabolic process, glutathione transferase activity
	Th3	1.32e-6	glutathione transferase activity

regulated gene cluster RatCNS1 is shown in Figure 6.5. We further investigated the genes in the clusters for RatCNS3. A majority of genes in RatCNS3 are involved in the protein binding process and the rest of the genes are involved in activities like calcium ion binding, growth factor activity, and transferase activity.

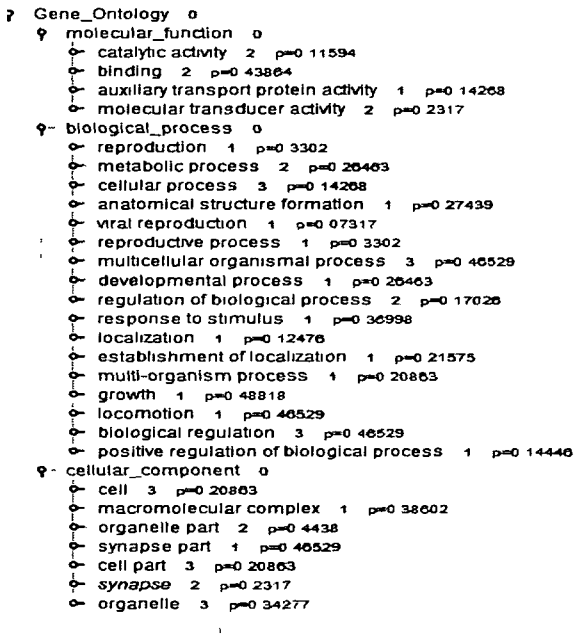


Figure 6.5: Significant GO terms on molecular function, biological process and cellular component from RatCNS1

Results show that our algorithm is capable of identifying biologically significant gene biclusters. Each group of genes in these clusters shows co-regulation (positive/negative) under a subset of conditions.

6.5.4 Performance comparison

To evaluate performance of CoBi in comparison to other algorithms, we consider three biclustering techniques Bimax[108], Cheng and Church (CC)[20] and OPSM[90] for the purpose. We use four Yeast datasets and BicAT tool[109] for analysis. We compare the performance based on functional enrichment of the biclusters. For the purpose of comparison, we set the parameter values of other algorithms as recommended in the original papers. The functional enrichment of each biclusters are measured based on Q-value associated with GO category. For each bicluster,

we calculate average of the percentage of number of genes from the biclusters with a given function against all genes in the genome with the function. Figure 6.6 shows average of functional enrichments of each biclusters obtained by different biclustering algorithms on four different datasets.

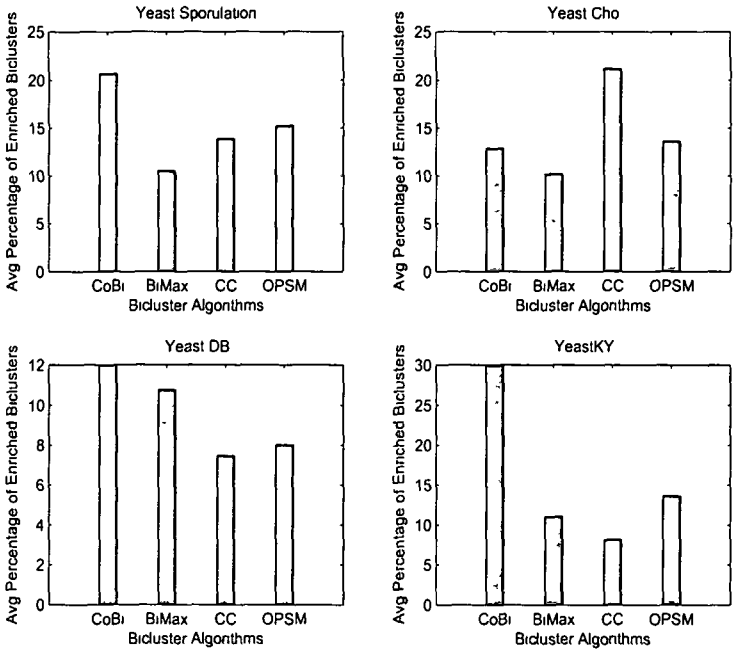


Figure 6.6: Comparison on functionally enriched biclusters from different biclustering techniques

From the graphs it is clearly evident that CoBi outperforms all three algorithms in obtaining functionally enriched biclusters. However, in case of YeastCho dataset, Cheng and Church (CC) approach performs better than other algorithms.

6.6 Discussion

In this chapter, we present a biclustering technique that is capable of detecting positively as well as negatively co-regulated genes. Unlike traditional proximity measures such as MSR, Euclidean distance or correlation, it uses a pattern based approach for finding the similarity between the genes To generate biclusters it

uses a tree based algorithm called BiClust. The results establish that co-regulated biclusters are significant from the biological point of view.

Chapter 7

Conclusions and Future work

7.1 Conclusions

In this thesis we applied association mining and clustering techniques in gene expression data analysis. We developed an one-pass association mining technique. Proposed technique uses a correlogram matrix for generating two element frequent itemsets and bitwise intersection approach for generating rest of the frequent itemsets from transaction database. We tested our technique using several synthetic and real datasets and compared the results against two well known techniques Apriori and FP-growth and found satisfactory. The advantage of correlogram matrix is used to find strongly correlated item pairs from transaction database using support based Pearson correlation coefficient. Experimental results are presented to establish that correlogram matrix based approach is effective in extracting strongly correlated item pairs compared to other similar techniques. Pearson correlation coefficient is not suitable when data are binary in nature and noisy. We proposed an alternative way of calculating correlation between item pairs using non parametric Spearman rank order correlation. Our results further reveal that Spearman rank order correlation allow to find more number of correlated pairs which are undetected by Pearson correlation approach. We extended the technique of finding pairwise relationship among item pairs using correlogram matrix, in finding co-regulated co-expression networks from gene expression data. We used a pattern based approach

for capturing both positive and negative co-regulations in gene expression data. We used several real gene expression data to test the effectiveness of our approach in extracting biologically significant co-expression networks. We compare our results with three well known gene regulatory network finding techniques ARACNE, CLR and MRNET in light of DREAM challenge datasets. Our approach outperforms all the three techniques. Finally, we contributed a BiClust tree based biclustering technique for clustering gene expression data. Pattern based similarity are calculated among the genes and biclusters are formed if genes are similar under few conditions. Various gene expression datasets are used and results are evaluated using gene ontology terms based on p and Q values associated with GO attributes. Three popular biclustering techniques BiMax, OPSM and CC (Cheng-Church) are used to compare effectiveness of our approach and found satisfactory.

7.2 Future work

The work presented in this thesis can be extended in diverse directions. Below we list some ideas for future work.

- In real world, categorical data as well as mixed data containing categorical and numerical variables are more abundant compared to binary data. The technique proposed to find frequent itemsets from market basket data can be extended to perform one-pass qualitative association mining without binarization of the data.

- Sequential association mining techniques assume that data are static in nature. However, in reality majority of the datasets are dynamic and incremental in nature. Existing techniques including our approach for finding frequent itemsets may not be suitable for finding association among incremental data. A technique that computes frequent new itemsets based on incremental data with minimal information and less computation is an important research issue. As a future work, our approach can be extended to handle dynamic data for finding frequent itemsets.

- Currently, we have presented a technique for reconstruction of co-expression network from microarray data. A gene regulatory network is a co-expression network with causality information, which is absent in our present work. Work is going on to extend our work for reconstruction of complete gene regulatory network with causality and regulation information.

- A good clustering algorithm should be capable of handling highly connected[111] and highly intersected or overlapping structures or even embedded structures prevalent in most of the gene expression data. We are working on density based approach for detecting intrinsic gene clusters from gene expression data.

- In recent study, Patrik D'haeseleer et al.[112] suggested that a single gene could be a member of multiple co-expressed groups, each reflecting a particular aspect of its function and control. A possible solution is to develop a clustering method that partitions genes into non-exclusive clusters. Traditional clustering approaches are unable to detect non-exclusive clustering. As a future work, one can use co-expression network to construct non-exclusive clusters from gene expression data.

- Though we are explicitly using data mining for gene expression data analysis, however as future work it is aimed to explore the applicability of prediction techniques in finding gene-gene relationship.

Bibliography

[1] Kurella, M., Hsiao, L., Yoshida, T., Randall, J., Chow, G., Sarang, S., Jensen, R., & Gullans, S. Dna microarray analysis of complex biologic processes, *Journal of the American Society of Nephrology*, **12**(5), 1072–1078, , 2001. 1

[2] Mitra, S. & Pedrycz, W. Guest editorial: Special issue on bioinformatics, *Pattern Recognition*, **39**(12), 2265–2266, , 2006. 2

[3] Gasch, A., Eisen, M., et al. Exploring the conditional coregulation of yeast gene expression through fuzzy k-means clustering, *Genome Biol*, **3**(11), 1–22, , 2002. 2

[4] Tavazoie, S., Hughes, J., Campbell, M., Cho, R., Church, G., et al. Systematic determination of genetic network architecture, *Nature genetics*, **22**, 281–285, , 1999. 2, 18, 74, 87, 96

[5] Chargaff, E. & Davidson, J. *The nucleic acids: chemistry and biology. Vol. 1.* Academic Press, , 1955. 9

[6] Strachan, T. & Read, A. *Human molecular genetics.* Oxford: BIOS Scientific Publishers, , 1996. 9

[7] Grant, R. *Computational Genomics: Theory and Application.* Horizon Bioscience, , 2004. 14

[8] Li, J. & Wong, L. Emerging patterns and gene expression data, *Genome Informatics Series*, , 3–13, , 2001. 15, 16

[9] Spellman, P., Sherlock, G., Zhang, M., Iyer, V., Anders, K., Eisen, M., Brown, P., Botstein, D., & Futcher, B. Comprehensive identification of cell cycle–regulated genes of the yeast saccharomyces cerevisiae by microarray hybridization, *Molecular biology of the cell*, **9**(12), 3273–3297, , 1998. 18

[10] Fayyad, U., Piatetsky-Shapiro, G., & Smyth, P. From data mining to knowledge discovery in databases, *AI magazine*, **17**(3), 37, , 1996. 18

[11] Hand, D. Data mining: statistics and more?, *The American Statistician*, **52**(2), 112–118, , 1998. 18

[12] Dunham, M. *Data mining: Introductory and advanced topics*. Pearson Education India, , 2006. 19, 20, 21, 22

[13] Berry, M. & Linoff, G. *Data mining techniques: for marketing, sales, and customer relationship management*. Wiley Computer Publishing, , 2004. 19

[14] Hatonen, K., Klemettinen, M., Mannila, H., Ronkainen, P., & Toivonen, H. Knowledge discovery from telecommunication network alarm databases, in Data Engineering, 1996. Proceedings of the Twelfth International Conference on. IEEE, 115–122. 20

[15] Antonie, M., Zaiane, O., & Coman, A. Application of data mining techniques for medical image classification, *MDM/KDD*, , 94–101, , 2001. 20

[16] Roy, S. & Bhattacharyya, D. K. *Data mining techniques and its application in medical imagery*. VDM Verlag Dr. Muller Germany, , 2010. 20

[17] Frank, E., Hall, M., Trigg, L., Holmes, G., & Witten, I. Data mining in bioinformatics using weka, *Bioinformatics*, **20**(15), 2479–2481, , 2004. 20

[18] Wilson, A., Thabane, L., & Holbrook, A. Application of data mining techniques in pharmacovigilance, *British journal of clinical pharmacology*, **57**(2), 127–134, , 2003. 20

[19] Lee, W. & Stolfo, S. *Data mining approaches for intrusion detection.* Defense Technical Information Center, , 2000. 20

[20] Cheng, Y. & Church, G. Biclustering of expression data, in ICISMB'00, Proc. of 8th Intl. Conf. on intelligent systems for molecular biology, volume 8, 93–103. 23, 77, 96, 112

[21] Agrawal, R., Imieliński, T., & Swami, A. Mining association rules between sets of items in large databases, in ACM SIGMOD Record, volume 22. ACM, 207–216. 23, 24, 26, 36, 48

[22] Fung, B. C., Wang, K., & Ester, M. Hierarchical document clustering using frequent itemsets, in Proceedings of SIAM international conference on data mining, 59–70. 23

[23] Fernando, B., Fromont, E., & Tuytelaars, T. Effective use of frequent itemset mining for image classification, in Computer Vision–ECCV 2012, 214–227. Springer, , 2012. 23

[24] Borgelt, C., Picado, D., Berger, D., Gerstein, G., & Grün, S. Cell assembly detection with frequent item set mining, *BMC Neuroscience*, **13**(Suppl 1), P126, , 2012. 23

[25] Picado-Muiño, D., Borgelt, C., Berger, D., Gerstein, G., & Grün, S. Finding neural assemblies with frequent item set mining, *Frontiers in neuroinformatics*, **7**(9), 1–15. 23

[26] Mooney, C. H. & Roddick, J. F. Sequential pattern mining–approaches and algorithms, *ACM Computing Surveys (CSUR)*, **45**(2), 19:1–19:39, , 2013. 26

[27] Agrawal, R., Srikant, R., et al. Fast algorithms for mining association rules, in Proc. 20th Int. Conf. Very Large Data Bases, VLDB, volume 1215, 487–499. 27, 104

[28] Mannila, H., Toivonen, H., & Verkamo, A. Efficient algorithms for discovering association rules, , , 1994. 28

[29] Agrawal, R., Mannila, H., Srikant, R., Toivonen, H., Verkamo, A., et al. Fast discovery of association rules, *Advances in knowledge discovery and data mining*, **12**, 307–328, , 1996. 28

[30] Houtsma, M. & Swami, A. Set-oriented mining for association rules in relational databases, in Data Engineering, 1995. Proceedings of the Eleventh International Conference on. IEEE, 25–33. 28

[31] Mueller, A. Fast sequential and parallel algorithms for association rule mining: A comparison, *Technical report, Faculty of the Graduate School of The University of Maryland*, **CS-TR-3515**, , 1998. 29

[32] Park, J., Chen, M., & Yu, P. *An effective hash-based algorithm for mining association rules*, volume 24. ACM, , 1995. 29

[33] Savasere, A., Omiecinski, E., & Navathe, S. An efficient algorithm for mining association rules in large databases, in VLDB'95. Proceedings of 20th International Conference on, 432–444. 30

[34] Toivonen, H. et al. Sampling large databases for association rules, in VLDB'96. Proceedings of 21th International Conference on. IEEE, 134–145. 30

[35] Brin, S., Motwani, R., Ullman, J., & Tsur, S. Dynamic itemset counting and implication rules for market basket data, in ACM SIGMOD Record, volume 26. ACM, 255–264. 31

[36] Han, J., Pei, J., & Yin, Y. Mining frequent patterns without candidate generation, in ACM SIGMOD Record, volume 29. ACM, 1–12. 32, 53, 54

[37] Shenoy, P., Haritsa, J., Sudarshan, S., Bhalotia, G., Bawa, M., & Shah, D. Turbo-charging vertical mining of large databases, in ACM SIGMOD Record, volume 29. ACM, 22–33. 36

[38] El-Hajj, M. & Zaïane, O. Inverted matrix: Efficient discovery of frequent items in large datasets in the context of interactive mining, in ACM

SIGKDD'03. Proceedings of the 9th International Conference on. ACM, 109–118. 36

[39] Han, J. & Kamber, M. *Data mining: concepts and techniques.* Morgan Kaufmann, , 2006. 48

[40] Brin, S., Motwani, R., & Silverstein, C. Beyond market baskets: generalizing association rules to correlations, in ACM SIGMOD Record, volume 26. ACM, 265–276. 48

[41] Reynolds, H. & Reynolds, H. *The analysis of cross-classifications.* Free Press New York, , 1977. 48

[42] Xiong, H., Shekhar, S., Tan, P., & Kumar, V. Exploiting a support-based upper bound of pearson's correlation coefficient for efficiently identifying strongly correlated pairs, in ACM SIGKDD'04. Proceedings of the 10th International Conference on. ACM, 334–343. 48, 51

[43] Xiong, H., Shekhar, S., Tan, P., & Kumar, V. Taper: A two-step approach for all-strong-pairs correlation query in large databases, *Knowledge and Data Engineering, IEEE Transactions on*, **18**(4), 493–508, , 2006. 48, 51

[44] Kuo, W., Mendez, E., Chen, C., Whipple, M., Farell, G., Agoff, N., & Park, P. Functional relationships between gene pairs in oral squamous cell carcinoma, in AMIA Annual Symposium Proceedings, volume 2003. American Medical Informatics Association, 371. 49, 75, 76

[45] Slonim, D. From patterns to pathways: gene expression data analysis comes of age, *Nature genetics*, **32**, 502–508, , 2002. 49

[46] Butte, A. & Kohane, I. Mutual information relevance networks: functional genomic clustering using pairwise entropy measurements, in Pac Symp Biocomput, volume 5, 418–429. 49, 76

[47] Shang, L. & Jian, Y. Mining top-k frequent correlated subgraph pairs in graph databases, in Intelligent Informatics, 1–8. Springer, , 2013. 50

[47] Shang, L. & Jian, Y. Mining top-k frequent correlated subgraph pairs in graph databases, in Intelligent Informatics (ASC-182), 1–8. Springer, , 2013. 50

[48] He, Z., Deng, S., & Xu, X. An fp-tree based approach for mining all strongly correlated item pairs, in Computational Intelligence and Security, LNCS, volume 3801. Springer Berlin Heidelberg, 735–740. 53

[49] Xiong, H., Zhou, W., Brodie, M., & Ma, S. Top-k ϕ correlation computation, *INFORMS Journal on Computing*, **20**(4), 539–552, , 2008. 53

[50] He, Z., Xu, X., & Deng, S. Mining top-k strongly correlated item pairs without minimum correlation threshold, *International Journal of Knowledge-based and Intelligent Engineering Systems*, **10**(2), 105–112, , 2006. 54

[51] Lehmann, E. & D'Abrera, H. *Nonparametrics: statistical methods based on ranks*. Springer New York, , 2006. 55, 56

[52] Katz, M. *Multivariable analysis: a practical guide for clinicians*. Cambridge university press, , 2006. 55

[53] Corder, G. & Foreman, D. *Nonparametric statistics for non-statisticians: a step-by-step approach*. Wiley, , 2009. 55

[54] Weinberg, S. & Goldberg, K. *Statistics for the behavioral sciences*. Cambridge University Press, , 1990. 56

[55] Litchfield Jr, J. & Wilcoxon, F. Rank correlation method, *Analytical Chemistry*, **27**(2), 299–300, , 1955. 56

[56] Goodman, L. & Kruskal, W. Measures of association for cross classifications, *J. of the American Statistical Association*, **49**(268), 732–764, , 1954. 56

[57] Borgelt, C. An implementation of the fp-growth algorithm, in Proceedings of the 1st international workshop on open source data mining: frequent pattern mining implementations. ACM, 1–5. 65

[58] Das, S., Caragea, D., Welch, S., & Hsu, W. *Handbook of research on computational methodologies in gene regulatory networks.* Medical Information Science Reference, , 2010. 74

[59] Lee, H., Hsu, A., Sajdak, J., Qin, J., & Pavlidis, P. Coexpression analysis of human genes across many microarray data sets, *Genome research*, **14**(6), 1085–1094, , 2004. 74

[60] Kommadath, A., te Pas, M., & Smits, M. Gene coexpression network analysis identifies genes and biological processes shared among anterior pituitary and brain areas that affect estrous behavior in dairy cows, *Journal of dairy science*, **96**(4), 2583–2595, , 2013. 75

[61] Mitra, S., Das, R., & Hayashi, Y. Genetic networks and soft computing, *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, **8**(1), 94–107, , 2011. 75

[62] Mitra, S., Das, R., Banka, H., & Mukhopadhyay, S. Gene interaction–an evolutionary biclustering approach, *Information Fusion*, **10**(3), 242–249, , 2009. 75, 78

[63] Jung, S. & Cho, K. Identification of gene interaction networks based on evolutionary computation, *Artificial Intelligence and Simulation*, , 428–439, , 2005. 75, 76

[64] Tong, A. et al. Global mapping of the yeast genetic interaction network, *Science*, **303**(5659), 808–813, , 2004. 75, 76

[65] Özgür, A., Vu, T., Erkan, G., & Radev, D. Identifying gene-disease associations using centrality on a literature mined gene-interaction network, *Bioinformatics*, **24**(13), i277–i285, , 2008. 75, 76

[66] Nagrecha, S., Lingras, P. J., & Chawla, N. V. Comparison of gene coexpression networks and bayesian networks, in Intelligent Information and Database Systems, 507–516. LNAI7802-Springer, , 2013. 75

[68] Davidich, M. & Bornholdt, S. Boolean network model predicts cell cycle sequence of fission yeast, *PLoS One*, **3**(2), e1672, , 2008. 75

[69] Kwon, A. T., Hoos, H. H., & Ng, R. Inference of transcriptional regulation relationships from gene expression data, *Bioinformatics*, **19**(8), 905–912, , 2003. 75

[70] Segal, E. et al. Rich probabilistic models for gene expression, *Bioinformatics*, **17**(suppl 1), S243–S252, , 2001. 75

[71] Eisen, M., Spellman, P., Brown, P., & Botstein, D. Cluster analysis and display of genome-wide expression patterns, *Proc. National Academy of Sciences*, **95**(25), 14863–14868, , 1998. 76

[72] Faith, J., Hayete, B., Thaden, J., Mogno, I., Wierzbowski, J., Cottarel, G., Kasif, S., Collins, J., & Gardner, T. Large-scale mapping and validation of escherichia coli transcriptional regulation from a compendium of expression profiles, *PLoS biology*, **5**(1), e8, , 2007. 76, 91

[73] Margolin, A., Nemenman, I., Basso, K., Wiggins, C., Stolovitzky, G., Favera, R., & Califano, A. Aracne: an algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context, *BMC bioinformatics*, **7**(Suppl 1), S7, , 2006. 76, 91

[74] Meyer, P., Kontos, K., Lafitte, F., & Bontempi, G. Information-theoretic inference of large transcriptional regulatory networks, *EURASIP Journal on Bioinformatics and Systems Biology*, **2007**, , 2007. 76, 91

[75] Aguilar-Ruiz, J. Shifting and scaling patterns from gene expression data, *Bioinformatics*, **21**(20), 3840–3845, , 2005. 76, 96, 97

[76] Yu, H., Luscombe, N., Qian, J., & Gerstein, M. Genomic analysis of gene expression relationships in transcriptional regulatory networks, *TRENDS in Genetics*, **19**(8), 422–427, , 2003. 77, 98

[77] Priness, I., Maimon, O., & Ben-Gal, I. Evaluation of gene-expression clustering via mutual information distance measure, *BMC bioinformatics*, **8**(1), 111, , 2007. 78

[78] Zhang, Z., Teo, A., Ooi, B., & Tan, K. Mining deterministic biclusters in gene expression data, in Bioinformatics and Bioengineering, 2004. BIBE 2004. in Proc. 4th IEEE Symposium on. IEEE, 283–290. 82, 101

[79] Tanay, A., Sharan, R., & Shamir, R. Discovering statistically significant biclusters in gene expression data, *Bioinformatics*, **18**(suppl 1), S136–S144, , 2002. 82, 96, 97

[80] Roy, S. & Bhattacharyya, D K Opam an efficient one pass association mining technique without candidate generation, *J. convergence information technology*, **3**(3), 32–38, , 2008. 83

[81] Marbach, D., Schaffter, T., Mattiussi, C , & Floreano, D. Generating realistic in silico gene networks for performance assessment of reverse engineering methods, *Journal of Computational Biology*, **16**(2), 229–239, , 2009. 85, 87

[82] Benjamini, Y. & Hochberg, Y. Controlling the false discovery rate: a practical and powerful approach to multiple testing, *Journal of the Royal Statistical Society, Series B (Methodological)*, , 289–300, , 1995. 87, 89

[83] Berriz, G., King, O., Bryant, B., Sander, C., & Roth, F. Characterizing gene sets with funcassociate, *Bioinformatics*, **19**(18), 2502–2504, , 2003. 87

[84] Warde-Farley, D. et al. The genemania prediction server: biological network integration for gene prioritization and predicting gene function, *Nucleic acids research*, **38**(suppl 2), W214–W220, , 2010. 89, 109

[85] Meyer, P., Lafitte, F , & Bontempi, G minet· Ar/bioconductor package for inferring large transcriptional networks using mutual information, *BMC bioinformatics*, **9**(1), 461, , 2008. 91

[86] Swets, J. *Signal detection theory and ROC analysis in psychology and diagnostics: Collected papers.* Lawrence Erlbaum Associates, Inc, , 1996. 92

[87] Craven, J. Markov networks for detecting overlapping elements in sequence data, in Advances in Neural Information Processing Systems 17: Proc. of the 2004 Conf., volume 17. MIT Press, 193. 92

[88] Sokolova, M., Japkowicz, N., & Szpakowicz, S. Beyond accuracy, f-score and roc: a family of discriminant measures for performance evaluation, *AI 2006: Advances in Artificial Intelligence*, , 1015–1021, , 2006. 92

[89] Eisen, M. B., Spellman, P. T., Brow, P. O., & Botstein, D. Cluster analysis and display of genomewide expression patterns, in Proc. Natl. Acad. Sci. USA, vol. 95, no. 25, 14863–14868. 95

[90] Ben-Dor, A., Shamir, R., & Yakhini, Z. Clustering gene expression patterns, *J. Comput. Biol.*, **6**(3-4), 281–297, , 1999. 95, 112

[91] Brazma, A. & Vilo, J. Gene expression data analysis, *FEBS Letter*, **480**(1), 17–24, , 2000. 95

[92] Chipman, H., Hastie, T., & Tibshirani, R. *Clustering microarray data.* Chapman & Hall/CRC, Boca Raton, Fla, , 2003. 95

[93] Jain, A. K. Data clustering: 50 years beyond k-means, *Pattern Recognition Letters*, **31**(8), 651–666, , 2010. 96

[94] Roy, S. & Bhattacharyya, D. K. An approach to find embedded clusters using density based techniques, *Distributed Computing and Internet Technology, LNCS*, **3816**, 523–535, , 2005. 96

[95] Mitra, S. & Banka, H. Multi-objective evolutionary biclustering of gene expression data, *Pattern Recognition*, **39**(12), 2464–2477, , 2006. 96

[96] Hartigan, J. A. Direct clustering of a data matrix, *J. Am. Stat. Assoc*, **67**, 123–129, , 1972. 96, 97

[97] Yang, J., Wang, H., Wang, W., & Yu, P. Enhanced biclustering on expression data, in Bioinformatics and Bioengineering, 2003. in Proc. 3rd. IEEE Symposium on, 321–327. 96

[98] Madeira, S. & Oliveira, A. Biclustering algorithms for biological data analysis: a survey, *Computational Biology and Bioinformatics, IEEE/ACM Transactions on*, **1**(1), 24–45, , 2004. 96, 97, 105

[99] Banka, H. & Mitra, S. Evolutionary biclustering of gene expressions, *Ubiquity*, **7**(42), 1–12, , 2006. 96, 97

[100] Nepomuceno, J., Troncoso, A., Aguilar-Ruiz, J., et al. Biclustering of gene expression data by correlation-based scatter search, *BioData mining*, **4**(3), , 2011. 96, 97

[101] Pei, J., Zhang, X., Cho, M., Wang, H., & Yu, P. Maple: A fast algorithm for maximal pattern-based clustering, in Data Mining, 2003. ICDM'03. Proc. of 3rd IEEE International Conference on. IEEE, 259–266. 96, 97

[102] Wang, H., Chu, F., Fan, W., Yu, P., & Pei, J. A fast algorithm for subspace clustering by pattern similarity, in Scientific and Statistical Database Management, 2004. Proc. of 16th Intl Conf on. IEEE, 51–60. 96, 97

[103] Roy, S., Bhattacharyya, D. K., & Kalita, J. K. Deterministic approach for biclustering of co-regulated genes from gene expression data, in KES12, Proc. of 16th Int. Conf. on, FAIA, volume 243, 490–499. 96

[104] Eren, K., Deveci, M., Küçüktunç, O., & Çatalyürek, Ü. V. A comparative analysis of biclustering algorithms for gene expression data, *Briefings in bioinformatics*, **14**(3), 279–292, , 2013. 96

[105] Wang, H., Wang, W., Yang, J., & Yu, P. Clustering by pattern similarity in large data sets, in Management of data. ACM SIGMOD'02. Proc. of Intl Conf on. ACM, 394–405. 97

[106] Zhao, Y., Yu, J., Wang, G., Chen, L., Wang, B., & Yu, G. Maximal subspace coregulated gene clustering, *Knowledge and Data Engineering, IEEE Transactions on*, **20**(1), 83–98, , 2008. 97

[107] Ji, L., Mock, K., & Tan, K. Quick hierarchical biclustering on microarray gene expression data, in BioInformatics and BioEngineering, 2006. BIBE'06. Proc. of 6th IEEE Symposium on. IEEE, 110–120. 98

[108] Prelić, A., Bleuler, S., et al. A systematic comparison and evaluation of biclustering methods for gene expression data, *Bioinformatics*, **22**(9), 1122–1129, , 2006. 112

[109] Barkow, S., Bleuler, S., Prelić, A., Zimmermann, P., & Zitzler, E. Bicat: a biclustering analysis toolbox, *Bioinformatics*, **22**(10), 1282–1283, , 2006. 112

[110] Jiang, D., Pei, J., & Zhang, A. Interactive exploration of coherent patterns in time-series gene expression data, in Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 565–570. 117

[111] Dhaeseleer, P., Liang, S., & Somogyi, R. Genetic network inference: from co-expression clustering to reverse engineering, *Bioinformatics*, **16**(8), 707–726, , 2000. 117

List of Publications

1. Roy, S., Bhattacharyya, D. K. and Kalita, J. K. CoBi: Pattern Based Co-Regulated Biclustering of Gene Expression Data, *Pattern Recognition Letters*, Elsevier, 2013 .

2. Roy, S. and Bhattacharyya, D. K. Mining Strongly Correlated Item Pairs in Large Transaction Databases", *Intl. Journal of Data Mining, Modelng and Management*, **5**(1), 76-96, 2013.

3. Roy, S., Bhattacharyya, D. K. and Kalita, J. K. GeCON: Expression Pattern Based Reconstruction of Gene Co-expression Networks from Microarray Data, (Communicated).

4. Roy, S., Bhattacharyya, D. K. and Kalita, J. K. Deterministic Approach for Biclustering of Co-Regulated Genes from Gene Expression Data, in 16th Intl. Conf.on Knowledge-Based and Intelligent Information & Engineering Systems (KES 2012), Advances in Knowledge-Based and Intelligent Information and Engineering Systems, FAIA-Vol:243, IOS Press, Spain, 490–499.

5. Roy, S. and Bhattacharyya, D. K. *Data Mining Techniques and its Application in Medical Imagery*, VDM Verleg Dr Muller, Germany, 2010.

6. Roy, S. and Bhattacharyya, D. K. Reconstruction of Genetic Network in Yeast using Support based Approach, in Trendz in Infor-

mation System & Computing (TISC'10), IEEE, India, 123–128.

7. Roy, S. and Bhattacharyya, D. K. Finding Gene-Gene Network using Support based Correlation Mining Techniques, in Algorithms in Applications, Narosa, U Sharma et.al, eds., India, 134–140.

8. Roy, S. and Bhattacharyya, D. K. Extracting Support Based k most Strongly Correlated Item Pairs in Large Transaction Databases, *Intl. Journal of Comp. Sc. Issues* **7** (5), 102–111, 2010.

9. Roy, S. & Bhattacharyya, D. K. OPAM: An Efficient One Pass Association Mining Technique without Candidate Generation, *J. Convergence Information Technology* **3**(3), 32–38, 2008.

10. Roy, S. and Bhattacharyya, D. K. Efficient Mining of Top-K Strongly Correlated Item Pairs using One Pass Technique, in Advance Computing and Communication (16th ADCOM'08), IEEE, India, 416-421.

11. Roy, S. and Bhattacharyya, D. K. SCOPE: An Efficient One Pass Approach to find Strongly Correlated Item Pairs, in Information Technology (11th ICIT'08), IEEE-CS Press, 123-126.

12. Roy, S. and Bhattacharyya, D. K. Frequent Mining: A Selective Survey, in *Networks, Security and Soft Computing*, S M Hazarika et.al., eds., Narosa, India, 2007, 96–105.

13. Roy, S. and Bhattacharyya, D. K. Data Clustering Techniques: A Review, in *Networks, Data Mining and Artificial Intelligence:*

Trends and Future Directions, S M Hazarika et.al, eds., Narosa, India, 2006, 139–152.