Biclustering Analysis for Large Scale Data

Akdes Serin

September 2011

Dissertation zur Erlangung des Grades
eines Doktors der Naturwissenschaften (Dr. rer. nat.)
am Fachbereich Mathematik und Informatik
der Freien Universität Berlin

Gutachter:
Prof. Dr. Martin Vingron
Prof. Dr. Tim Beissbarth

1. Referent: Prof. Dr. Martin Vingron
2. Referent: Prof. Dr. Tim Beissbarth

Tag der Promotion: 18 November 2011

To my parents and grandmother...

Preface

Acknowledgments Foremost, I am very grateful to my supervisor *Martin Vingron* for his scientific support, wise feedbacks and giving me the unique opportunity to pursue my PhD in his group. I always enjoyed and felt privileged to work in Vingron group with exceptional scientists in a motivating working environment. I want to thank *Alexander Bockmayr*, *Max von Kleist* and *Tim Beissbarth* for their counsel as members of my PhD committee. I also thank my collaborators *Yuhui Hu* and *Sebastiaan Meijsing* for all the scientific discussions that has deepened my biological knowledge. I further thank *International Max Planck Research School for Computational Biology and Scientic Computing* for funding my PhD. I am greatly indebted to the coordinators of the school, *Hannes Luz* and *Kirsten Kelleher* for always helping me right away whenever I needed.

I would like to give my gratitudes to *Morgane Thomas Chollier*, *Julia Lasserre* and *Kirsten Kelleher* for proofreading the thesis and giving their valuable comments. I am deeply grateful to *Sarah Behrens* for all her advices and support. I also thank *Corinna Blasse*, *Patricia Marquardt* and *Kirsten Kelleher* for helping me in writing the German abstract of the thesis. I give my special thanks to my dear office-mates *Alena Mysickova*, *Jonathan Goeke* and *Christian Roedelsperger* for forming the most productive, entertaining, delicious etc., shortly the best office ever. I would like to thank *Ruping Sun*, *Yves Clement*, *Mike Love*, *Juliane Perner*, *Rosa Karlic*, *Paz Polak*, *Yasmin Aristei*, dear *Annalisa Marsico* and all the current and past members of Vingron department that I forgot to mention. I am very happy to know all these nice variety of special people. You made the moments of my stay in Berlin very enjoyable and l learned a lot from all of you. I know that it will be hard to find such a wonderful group in the future.

Finally, my greatest gratitudes goes to my dear parents, brother, sister and grandmother. This thesis would not been possible without their endless support, understanding and love. I feel very lucky to have you, I love you.

Publications Our novel fast biclustering algorithm called DeBi (Differentially Expressed BIclusters) appeared in *Algorithms for Molecular Biology* [102] (described in chapter 5). The project titled "Medicinal Connectivity of Traditional Chinese Medicine (MecoTCM)" will be submitted very soon (described in chapter 6).

Akdes Serin Berlin, September 2011

Contents

List of Figures

List of Tables

Almost all aspects of life are engineered at the molecular level, and without understanding molecules we can only have a very sketchy understanding of life itself.

Francis Crick

Chapter 1

Introduction

We are concerned with the development and application of computational methods for the analysis of gene expression data. In this thesis, we focus on two approaches applied to gene expression data. The first approach called biclustering, detects sets of genes that have similar expression patterns under certain biological conditions. We developed an efficient novel biclustering algorithm for detecting statistically significant patterns especially in large datasets. The second approach aims to find the connections between compounds with unknown functions and drugs using a pattern matching tool. In this introduction chapter, we first give a concise biological background on gene expression (section 1.1). Then, we introduce the microarray technology and the computational challenges in analyzing microarray datasets (section 1.2). Finally, we give the motivation, structure and contribution of the thesis (section 1.3 and section 1.4).

1.1 Biological Background

Deoxyribonucleic acid (DNA) is the most important constituent of the cell. Almost all the cells in our body contain the same DNA molecules. DNA stores and passes the genetic information from one generation to the next. Thus, understanding the way DNA functions will give us a deeper insight into the mechanism of the cell.

DNA The information in DNA is stored in the sequence of four chemical *bases*: adenine (A), cytosine (C), guanine (G) and thymine (T). A *nucleotide* is composed of a base, a five-carbon sugar, and one to three phosphate groups. Nucleotides are arranged in two long strands that form a spiral called a *double helix*. In 1953, Watson and Crick discovered the double helix structure of DNA. Each type of base on one strand forms a bond with a given type of base on the other strand. Purines (A or G) form hydrogen bonds to pyrimidines (T or C), with A bonding only to T, and C bonding only to G (Fig. 1.1).

Genes are the DNA segments and their base sequence specifies the sequence of the amino acids within proteins. The definition of a gene is; a locatable region of genomic

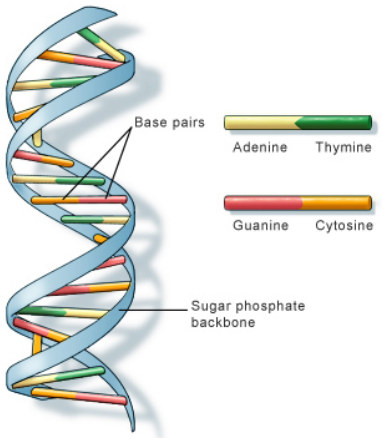


Figure 1.1: Double helix structure of the DNA molecule. Purines (A or G) form hydrogen bonds to pyrimidines (T or C), with A bonding only to T, and C bonding only to G. The figure is adapted from a public domain illustration of the U.S. National Library of Medicine `http://ghr.nlm.nih.gov/`

sequence, corresponding to a unit of inheritance, which is associated with regulatory regions, transcribed regions, and or other functional sequence regions [42].

Gene Expression and Regulation Central Dogma describes the flow of genetic information from DNA to RNA (transcription) and from RNA to protein (translation) (Fig. 1.2) [30]. In the *transcription* step, an enzyme called RNA polymerase transcribes the information encoded in the DNA segment, i.e. gene, to precursor messenger RNA (pre-mRNA). The pre-mRNA forms the mature mRNA after post-transcriptional modifications (see step 2 in Fig. 1.2). The pre-mRNA contains both *exons* and *introns*. Exons are the template sequences for the protein, whereas introns are noncoding sequences mostly have a regulatory rule. The introns are removed from pre-mRNA by splicing factors. Sometimes pre-mRNA is spliced in different ways, allowing a single to encode multiple different proteins. This process is called *alternative splicing*. In the *translation* step, the mRNA is translated into amino-acid chain, forming proteins (see step 3 in Fig. 1.2).

Gene expression is the process by which information encoded in a gene is used for the synthesis of a functional gene product. The control of the timing, location, and amount of gene expression can have a profound effect on the functions of the gene in a cell [5].

Even though almost all cells in our body contain the same genetic information, we have different cell types (e.g. neurons, muscle cells, liver cells, ...). These differences arise from different gene expression regulation in the cells. Genes are differentially expressed in different conditions: cell types, environment, developmental phases, cell cycle stages, disease stages.

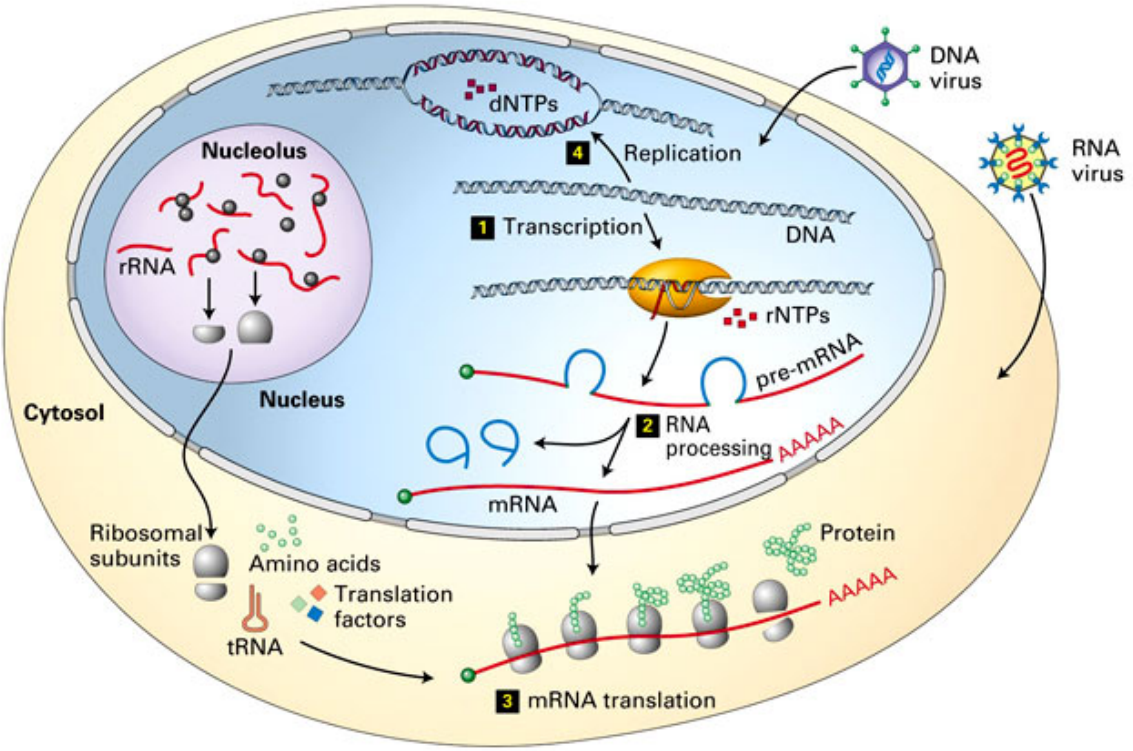


Figure 1.2: Biological processes in a eukoryatic cell. Central Dogma describes the flow of genetic information from DNA to RNA (transcription) and from RNA to protein (translation). In step 1, the information encoded in the DNA segment transcribed to precursor messenger RNA (pre-mRNA). In step 2, the pre-mRNA forms the mature mRNA after post-transcriptional modifications. In step 3, the mRNA is translated into amino-acid chain, forming proteins. Step 4 shows the replication of DNA.The figure is adapted from [75].

Gene regulation studies how genes perform their functions at the right time in the right place. There are several control mechanisms of gene regulation. The most important one takes part in transcription process where the amount of synthesized mRNA is controlled. Each gene, begins with one or more *transcription start site* (TSS). The term promoter designate the regions upstream of transcription start sites. The core promoters are bound by RNA polymerase together with *Transcription Factors* (TF) to control the transcription of a gene. The DNA segments bound by such factors, called binding sites, are usually very short ($\sim$ 15 nucleotides) are recognized by the factors in a sequence specific manner. The binding preference of a given TF is referred to as a *regulatory motif*.

1.2 Microarray Technology

In recent years, various high throughput technologies such as cDNA microarrays [99, 34, 22], short oligo-microarrays [67, 74] and sequence-based approaches (RNA-Seq) [118]

for transcriptome profiling have been developed. These latest technologies allow us to monitor gene expression of tens of thousands of genes in parallel. In this thesis, we are focusing on analyzing microarray data, therefore we will not be discussing RNA-Seq technology.

The main principle behind DNA microarrays is that complementary nucleotide sequences pair with each other by forming hydrogen bonds. DNA microarrays are prepared on a solid surface divided into small grids. Each grid contains a piece of DNA known as probes. The probes are short sections of a gene with the length ranging from 25 to 75 bases. The mRNAs extracted from the samples are reverse transcribed into complementary DNA called cDNA. The amount of cDNA sample (target) bound to each spot on the array indicates the expression level of the genes.

There are different types of microarray technologies, mainly cDNA microarrays and oligonucleotide microarrays. In cDNA microarrays, the probes are synthesized by robots and attached to the glass microscopic slides. They are double channel microarrays. Two samples are hybridized to one microarray at the same time. One sample is coming from the control, whereas the other one is coming from the sample of interest such as cancer tissue. Each sample is dyed with distinct marker, a red Cy3 dye versus a green Cy5 dye. After hybridization, the array is read by the scanner. The differential expression level of a given gene is measured by the ratio between the signal intensities of two colors. In oligonucleotide microarrays, the probes are short sequences designed to match the subsequences of the mRNA transcript. The mRNA transcript is, in average, thousands of base pairs long and the probes are 25 or 75 base pairs long depending on the platform. Each mRNA is represented by several probes. The probes are synthesized directly onto the array surface. Then the sample containing the mRNAs are attached to the array, and for each probe the amounts of bound mRNA is measured. The probe types and normalization methods specific to oligonucleotide microarrays will be discussed in Chapter 2.

DNA microarrays have been used successfully in various research areas such as gene discovery [57], disease diagnosis [94] and drug discovery [44]. The functions of the genes and the mechanisms underlying diseases can be identified using microarrays. Additionally, microarrays has an extensive application in drug discovery. Drug is a chemical substance used in the treatment, cure or prevention of disease. Microarrays assist in drug discovery by monitoring changes in gene expression in response to drug treatment.

There are several challenges in analyzing microarray datasets. Most importantly, the microarray data tend to be noisy due to nonspecific hybridization, image scanning and etc. The technical biases like different dye efficiencies, must be removed by normalization. However, choosing the best normalization method for the data is not very easy [93]. We will discuss the normalization methods for oligonucleotide microarrays in Chapter 2.

1.3 Thesis Motivation

High throughput technologies are the latest breakthroughs in experimental molecular biology. These technologies provide insight into the molecular mechanism of the cell which was impossible to study with traditional approaches. However, sophisticated statistical and computational methods are required to extract useful information from these datasets.

The most common approach for detecting functionally related gene sets from such high throughput data is clustering [9]. Traditional clustering methods like hierarchical clustering [107] and k-means [53], have several limitations. Firstly, they are based on the assumption that a cluster of genes behaves similarly in all samples. However, a cellular process may affect a subset of genes, only under certain conditions. Secondly, clustering assigns each gene or sample to a single cluster. However, some genes may not be active in any of the samples and some genes may participate in multiple processes. Biclustering overcomes these limitations by grouping genes and samples simultaneously. Recent studies showed that biclustering has a great potential in detecting marker genes that are associated with certain tissues or diseases. Several biclustering algorithms have been proposed. However, it is still a challenge to find biclusters that are significant based on biological validation measures. Nowadays, It is possible to download large gene expression datasets from publicly available repositories, such as GEO [38] and ArrayExpress [20]. There is a need for a biclustering algorithm that is capable of analyzing very large datasets in reasonable time.

The first part of the thesis focuses on biclustering algorithms. We have proposed a novel fast biclustering algorithm especially for analyzing large data sets. Our algorithm aims to find biclusters where each gene in a bicluster should be highly or lowly expressed over all the bicluster samples compared to the rest of the samples. Unlike other algorithms, it is not required to define the number of biclusters apriori.

In the second part of the thesis we are interested in revealing connections between small molecules and drugs using gene set enrichment metric. A variety of cell lines treated with a variety of small molecules are analyzed to derive induced and repress gene sets. Then, the derived gene sets are used to reveal the similarities between small molecules and drugs. The small molecules with high similarities thus hold a potential to be alternative to existing drugs and the underlying mechanisms are likely disclosed by the affected genes and pathways. We also used biclustering to discover novel drugs from compunds with unknown functions.

1.4 Thesis Organization

The organization of the dissertation is as follow:

In Chapter 2, we will give a brief overview on different high throughput technologies for measuring gene expression. Then we will explain statistical normalization techniques specific to the type of the technology. Afterwards, the detection of differentially expressed genes, gene set enrichment analysis and clustering will be overviewed.

In Chapter 3, we will extend the discussion of clustering to biclustering. We will give a brief survey of existing biclustering algorithms. Then, we will discuss evaluation and validation measures for the biclustering results. Finally, we will present different ways of visualizing biclustering results.

In Chapter 4, we will present a well known data mining approach called frequent itemset. We will explain several algorithms for identifying frequent itemsets. Frequent itemset aproach will be used later in Chapter 5 for our proposed biclustering algorithm.

In Chapter 5, we will introduce our novel fast biclustering algorithm called DeBi (Differentially Expressed BIclusters). Then, we will evaluate the performance of DeBi on a yeast dataset, on synthetic datasets and on human datasets. We will also compare our algorithm with existing biclustering algorithms based on biological validation measures.

In Chapter 6, we will introduce the gene set enrichment method for revealing the hidden connections among drugs, genes and small molecules. Then, we will use this method for elucidating molecular mechanisms of Traditional Chinese Medicine (TCM) compounds and for identifying new drug candidates from TCM against different human diseases. The method will be applied to existing data describing the effect of well-known drugs, provided in the so-called Connectivity Map (cMap). Finally, we will apply our novel biclustering algorithm on cMap microarray data combined with TCM microarray data.

. . .to call in the statistician after the experiment is done may be no more than asking him to perform a post-mortem examination: he may be able to say what the experiment died of. . .

Ronald A. Fisher

Chapter 2

Analysis of oligonucleotide microarray data

In this chapter, we first give a brief overview of different high throughput technologies for measuring gene expression (section 2.1). Then the importance of microarray experiment design is mentioned (section 2.2). In section 2.3, we explain preprocessing techniques specific to Affymetrix Gene Chips and Illumina Bead Arrays platforms. Afterwards, the identification of differentially expressed genes, gene set enrichment methods and clustering is discussed (section 2.4).

2.1 Overview of Oligonucleotide Microrarray Technologies

The leading high-density microarray technologies are Illumina Bead Arrays and Affymetrix Gene Chips [67, 74]. Both Affymetrix Gene Chips and Illumina Bead Array platforms are oligonucleotide microarrays. Below we will review the probe types and normalization methods for both platforms. Knowing the properties specific to each platform is necessary for correctly processing and analyzing the data.

Affymetrix Gene ChipsTechnology In Affymetrix oligonucleotide microarrays, probes are 25 base pairs long. Affymetrix microarrays contain 11-16 pairs of probes per gene. Multiple probes are designed for a single gene because different probes for the same gene have different binding affinities. The probe pair contains a perfect match(PM) oligonucleotide and a mismatch(MM) oligonucleotide. The PM probes are paired with MM probes to control cross-hybridization. The MM probe sequence differs from PM by a complementary base located in the 13th position. The average of the PM-MM differences for all probe pairs in a probe set is used as the expression value for the target gene [59]. Figure 2.1 summarizes the probe design in Affymetrix micorarrays.

Affymetrix oligonucleotide probe set annotation: In the latest human microarrays (HG-U133) three probe set designations are used [2].

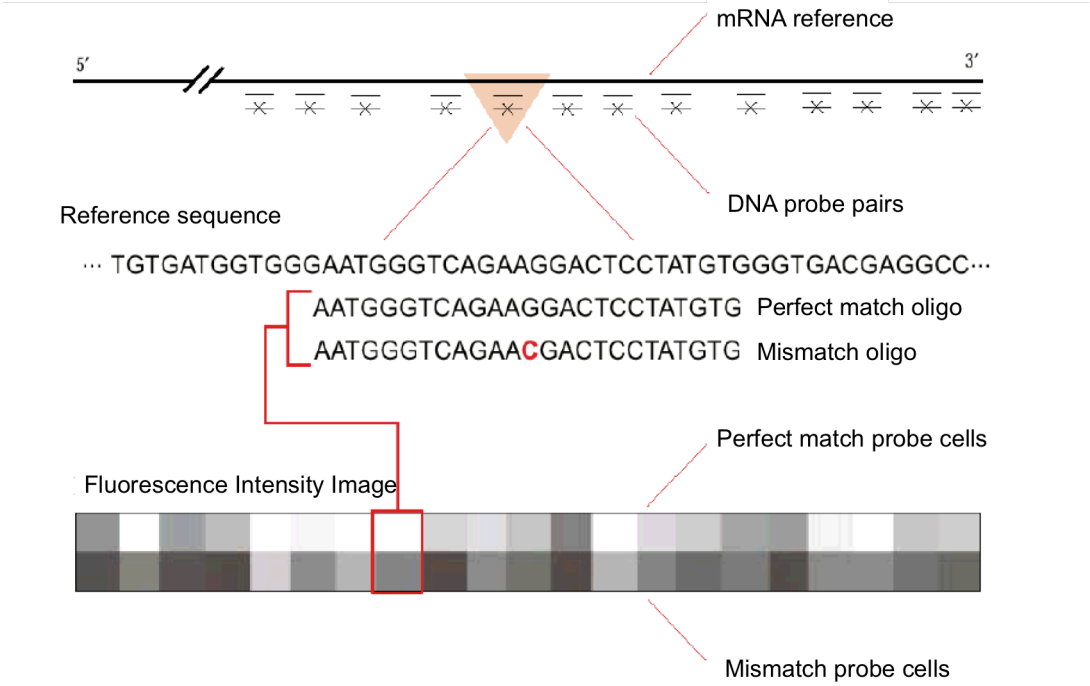


Figure 2.1: Affymetrix oligonucleotide probe design. Affymetrix microarrays contain 11-16 pairs of probes per gene and each probe is 25 base pair long. The probe pair contains a perfect match(PM) oligonucleotide and a mismatch(MM) oligonucleotide. The PM probes are paired with MM probes to control cross-hybridization. The MM probe sequence differs from PM by a complementary base located in the 13th position. The figure is adapted from [108].

1. *_at*: A probe set name is appended with the *_at* extension when all the probes hit one known transcript.

2. *s_at*: A probe set name is appended with the *s_at* extension when all the probes exactly match alternative transcripts from the same gene. However, sometimes it can also represent transcripts from homologous genes.

3. *x_at*: A probe set name is appended with the *x_at* extension when some probes are identical, or highly similar, to unrelated sequences. This may lead to cross-hybridization to the sequences other than the target transcript.

Advantages of Affymetrix Microarrays: Different probes for the same gene have different binding affinities. Affymetrix microarrays have multiple probes for a single gene and thus avoid probe binding affinity differences. Additionally, the mismatch probes helps to identify and minimize the effects of nonspecific hybridization and background signal [84].

Disadvantages of Affymetrix Microarrays: Only one sample can be measured per chip and thus leads to noise between chips. Some probes may cross-hybridize in an

unpredictable manner with sequences other than the target mRNAs, as explained in probe set designation x_at [84].

Illumina Microarrays The Illumina BeadArray technology is based on randomly arranged beads [87]. A gene-specific 50-mer oligonucleotide is assigned to each bead. There are roughly 30 copies of each gene-specific bead in an array.

Illumina oligonucleotide probe id annotation: Illumina probes are designed to hybridize with one transcript of a single gene. However, the changes in genome annotations may give rise to one probe mapping to multiple genes. In addition to probe annotations, nucleotide universal identifier (nuIDs) are developed for each probe to build a generic annotation pipeline that is independent of manufacturer or different versions of the same company's microarray [35].

Advantages of Illumina Arrays: Technical replicates on each array, over exact 30 copies of beads per probe, increases the precision of the measure.

Disadvantages of Illumina Arrays: The annotation of Illumina probes is different from Affymetrix probes. In Affymetrix, if one probe is defective, then the other probes in the probeset can still be used to measure the target gene. However, in Illumina where one gene is represented by only one type of bead, if a probe is defective, then the target gene can not be measured. Long probes have a higher chance of folding, self-hybridizing or forming a hairpin.

2.2 Experimental Design

Experimental design is crucial to ensure that questions of interest can be answered clearly and efficiently. There are several issues in microrarray experiment design. Most importantly, replicates of the biological samples are needed to draw statistical conclusions from the experiment. Technical replicates, i.e. two samples from the same extraction, can also be performed but they are not as necessary as the biological replicates [27].

2.3 Preprocessing Microarray Data

Preprocessing removes the non-biological effects from the data and thus leads to better answers for our biological questions. For each platform we have a different processing pipeline, as reviewed below.

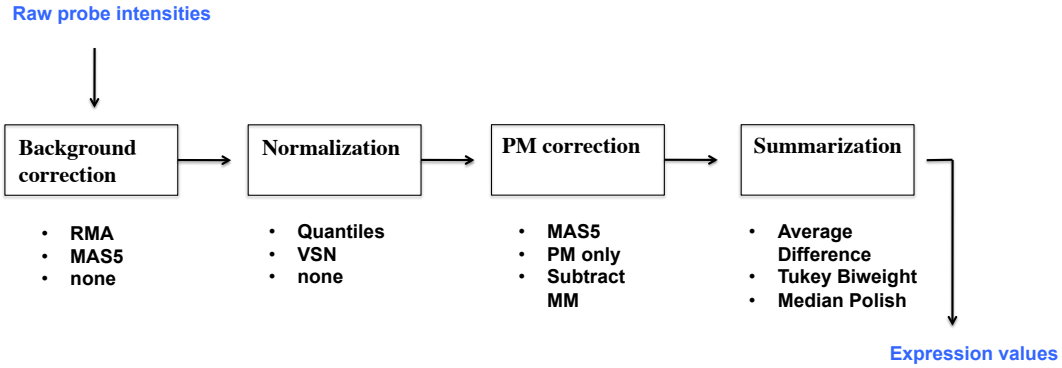


Figure 2.2: Affymetrix data preprocessing steps.The steps are; (1) background correction, (2) normalization and (3) summarization.

Affymetrix Microarrays We have thousands of intensity values for probes that are grouped into probesets. The non-biological systematic effects should be removed from the data prior to performing high-level analysis. Processing the raw data consists of three steps: Background correction, normalization and probe summarization (Fig. 2.2).

Step 1. Background Correction: The background adjustment corrects for the background noise arising from non-specific DNA-binding of the sample. There are two common models for background adjustment.

RMA convolution: The proposed model is: $S = X + Y$, where X is signal and Y is background [19, 60]. It is assumed that X is exponentially distributed $exp(\alpha)$ and Y is normally distributed $N(\mu, \sigma^2)$. Assumptions are made based on the observation of the distribution of the probes. According to the model we can estimate the background corrected probe intensities by $E(X|S)$ and the distribution parameters can be estimated in an ad-hoc approach from the data. Only the PM probes are corrected. The drawback of the method is that all the chips assumed to have the same distribution and MM probes are not used in the model.

MAS5: The MAS 5.0 background correction divides the chip into 16 zones [1]. An average background value of a cell is estimated using lowest 2% intensities of the corresponding cell. Then a weighted combination of these background estimates is subtracted from each probe intensity.

Step 2. Normalization: In order to compare the expression measures of different arrays, variations between arrays should be removed from the data. The general assumption in normalization is that each array contains equal amounts of RNA. Therefore, the simplest method for normalization is to scale the intensities of each array to have the same average intensity across all arrays. However there is a non-linear relation among probes from different arrays. Below we will review more sophisticated normalization methods.

Variance Stabilization Normalization (VSN): The variance of the measured intensity X_i of gene i depends on the mean intensity measure of X_i [56]. Therefore, the interpretation of fold-changes in raw data may lead to wrong conclusions. In VSN, we aim to transform the data so that the mean and variance become independent . The model is based on standard error model:

$$Y = \alpha + \mu \times e^{\eta} + \epsilon \tag{2.1}$$

where Y is the measured expression value, α is the offset, μ is the real expression value. The additive and multiplicative error terms are ϵ and η respectively. The expectation and the variance of Y are estimated as:

$$E(Y) = u = \alpha + m_{\eta}\mu \tag{2.2}$$

$$Var(Y) = v = s_{\eta}^2\mu^2 + \sigma_{\epsilon}^2 \tag{2.3}$$

The mean and variance of e^{η} are m_{η} and s_{η}^2, respectively; σ_{ϵ}^2 is the variance of ϵ. The estimate of μ from equation 2.2 is $(u - \alpha/m_{\eta})$. We can reformulate the variance defined in equation 2.3 in terms of $E(Y)$ as:

$$v(u) = \frac{s_{\eta}}{m_{\eta}}^2 u - \alpha^2 + \sigma_{\epsilon}^2 = (c_1 u + c_2)^2 + c_3 \tag{2.4}$$

The dependency between the variance v and the mean u can be seen from he equation 2.4. By using the delta method, Y is transformed to $h(Y)$ so that the mean does not depend on the variance:

$$h(Y) = \int^{y} \frac{1}{\sqrt{v(u)}}\,\mathrm{d}u \tag{2.5}$$

So, if we can estimate the intensity variance v and mean u of each probe, we can infer the functions $v(u)$ and $h(y)$, and stabilize the variance by equation 2.5.

Quantile Normalization: The goal of the quantile normalization is to make two distributions identical in statistical properties [18]. When two distributions are same then the quantile-quantile plot will have a straight diagonal line.

Let us say we have n arrays and the corresponding k^{th} quantile for all arrays is $\mathbf{q}_k = (q_{k1}, \ldots, q_{kn})$ for $k = (1, \ldots, p)$. The diagonal unit vector is

$\mathbf{d} = (\frac{1}{\sqrt{n}}, \ldots, \frac{1}{\sqrt{n}})$. In order to obtain the same distribution for all arrays, the quantiles of each array should lie along the diagonal. When we consider the projection of $\mathbf{q}$ onto $\mathbf{d}$, each array can have the same distribution by taking the mean quantile and substituting it with the value of the data item in the original dataset.

$$proj_{\mathbf{d}} q_k = \frac{q_k \cdot \mathbf{d}^T}{\mathbf{d} \cdot \mathbf{d}^T} \cdot \mathbf{d} = (\frac{1}{n} \sum_{j=1}^{n} q_{kj}, \ldots, \frac{1}{n} \sum_{j=1}^{n} q_{kj}) \tag{2.6}$$

The algorithm for the quantile normalization thus consists of three steps. First, in each array the intensity values are ordered. Second, the average of all the probes is calculated. Third, probe intensities are substituted by the average value. And finally, the probes are sorted in original order.

Step 3. Summarization: In the summarization step, normalized expression values are summarized into a single expression value per probe set.

Median Polish: It is based on the assumption that PM values follow a linear additive model with probe affinity effect, gene specific effect and an error term. The gene specific effect, expression values, are estimated by robust model fitting technique [115].

Tukey's biweight: It is done by taking a robust average of the probes. The distance of each probe intensity to the median is calculated, then the distances are used to determine how much each probe should contribute to the average [115].

To sum up, there are two well known algorithms called MAS5 and Robust Multi-Array(RMA), for preprocessing the raw intensity values [60, 1]. The two different pipelines for processing Affymetrix arrays are [18]:

1. RMA algorithm: In RMA, MM values are not subtracted from PM values, only PM values are considered. Background adjustment is done using RMA convolution. Then quantile normalization is performed. Finally, summarization is done using median polish.

2. MAS5 algorithm: Background adjustment is done using MAS5. Then, MM values are subtracted from PM values. If the subtracted value is negative then it is adjusted to give a positive score. Finally, probe summarization is based on Tukey's biweight.

Illumina The recommended pipeline for Illumina normalization includes the following steps:

Step 1. Variance Stabilizing Transformation (VST): Variance Stabilization Normalization(VSN), which is explained in section 2.3, models the mean-variance dependency of intensity values by using the non-differentially expressed genes as

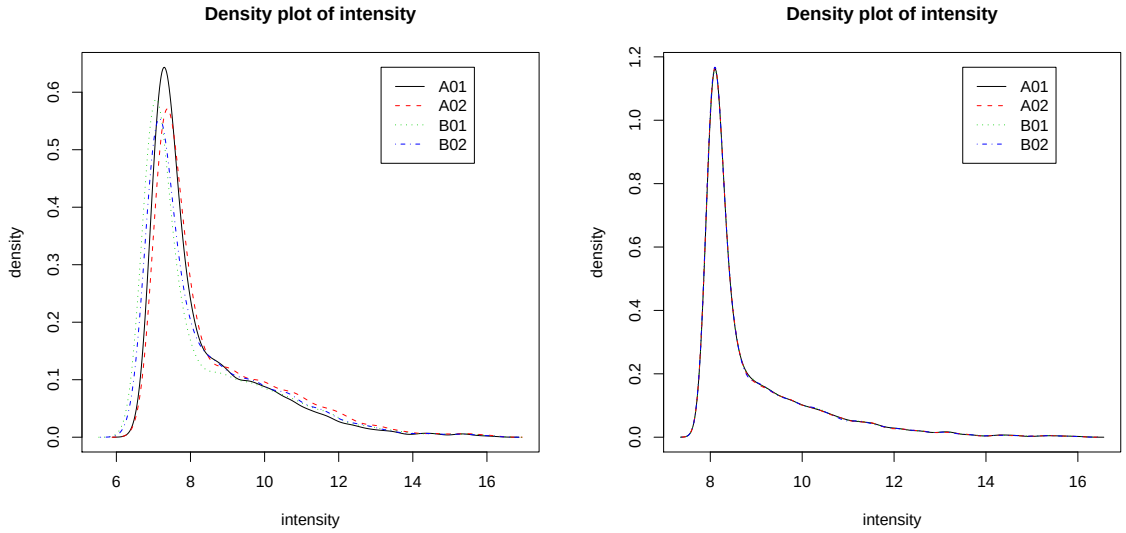


Figure 2.3: Before and after normalization density plots for four samples. In the density plots, we can observe the distrubution of the probes in different samples. Before normalization, each sample has different intensity distributions.

technical replicates. Therefore, in VSN multiple arrays are needed to estimate the data transformation parameters.

In Illumina, each probe is measured around 30 times in each array. The technical replicates of the probes help us to model the mean-variance dependency of the intensity values for each array directly. Therefore in Illumina platforms, we are able to estimate the data transformation parameters directly from single array as opposed to other microarray platforms. Variance Stabilization Transformation(VST) is a modified version of VSN, specific to Illumina platforms [73].

Step 2. Quantile Normalization: After removing the dependency between variance and mean we apply quantile normalization (section 2.3).

Figure 2.3, Figure 2.4 and Figure 2.5, illustrate the data before and after normalization using density plots, box plots and MA plots. In the density plots, we can observe the distrubution of the probes in different samples. Before normalization, each sample has different intensity distributions. The Boxplot also shows us the differences in distributions across samples. The third plot is the MA plot of all possible paired sample combinations. In a MA plot, y-axis is the difference of the measurements of two samples and the x-axis is the averages of the measurements of two samples. In the MA plot before normalization, we can observe the banana shape which indicates that low intensity values are under estimated in one sample. After normalization, this systematic effect is removed from the data.

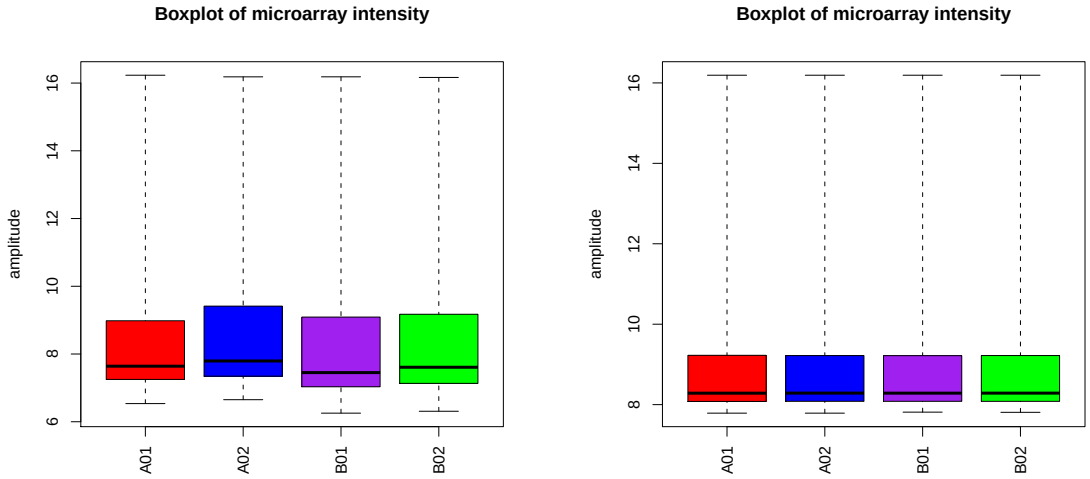


Figure 2.4: Before and after normalization box plots for four samples. The boxplot shows us the differences in distributions across samples.

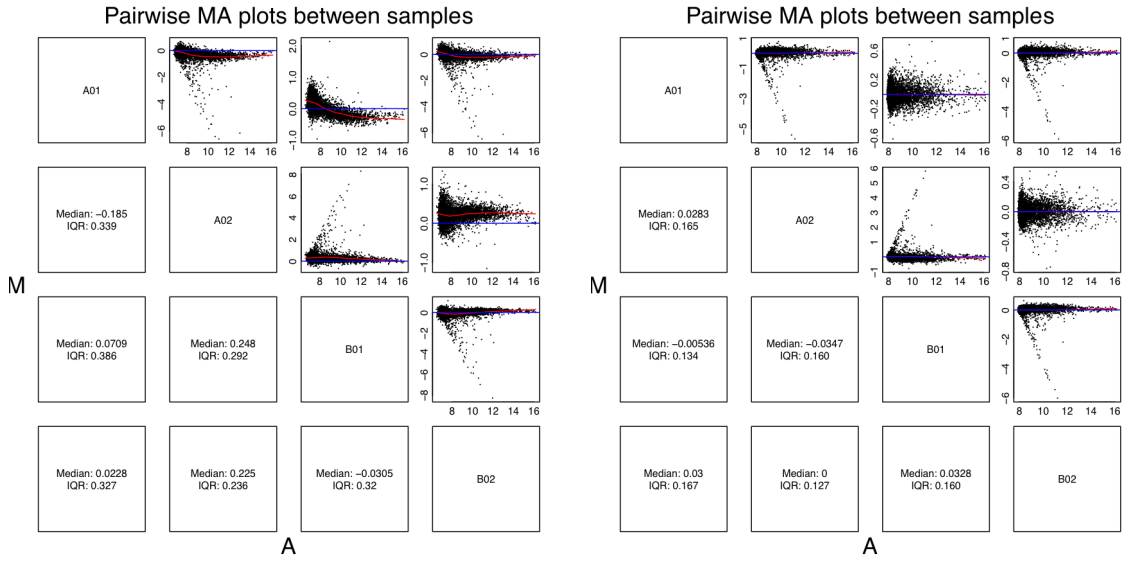


Figure 2.5: Before and after normalization MA plots for four samples. In a MA plot, y-axis is the difference of the measurements of two samples and the x-axis is the averages of the measurements of two samples. The figures show all possible paired sample combinations. In MA plot before normalization, we can observe the banana shape which indicates that low intensity values are under estimated in one sample.The MA plot after normalization illustrates that the banana shape is corrected, the systematic effects are removed.

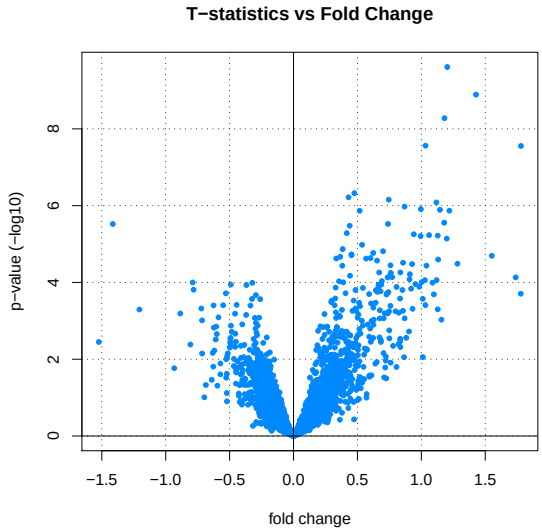


Figure 2.6: T-statistics vs Fold change. Genes with large fold change do not necessarily have high statistical significance.

2.4 High Level Analysis of Microarray Data

The preprocessing step has removed the systematic non-biological effects from the data. Now, we can apply sophisticated statistical and computational methods to extract useful information from these datasets [105]. We can identify the genes whose expression level significantly altered under different experimental conditions. Furthermore, gene set enrichment methods can be used to determine whether an a priori defined set of genes shows statistically significant under different experimental conditions [65, 111]. Additionally, classification methods are required to characterize the overall structure of the data. There are two main types of classification: unsupervised and supervised. In supervised classification, additional prior knowledge is given apart from the data. In the past, supervised methods have been successfully applied to gene expression data for disease prediction outcome [121]. On the other hand, in unsupervised classification prior knowledge is not required. The most common methods for unsupervised classification is clustering, biclustering and principal component analysis [40, 123, 78].

Identifying differentially expressed genes We are interested in identifying genes that are differentially expressed between different cell/tissue/disease types. Detecting differentially expressed genes by using the fold change doesn't account for expression variation and does not give any assessment on statistical significance. For example in Figure 2.6, we can observe that genes with large fold change does not necessarily have high statistical significance. Below, we will review a statistical method for detecting differentially expressed genes.

T-statistics: We can conduct t-statistics to compare two different treatments. In t-statistics, we test the equality of the mean of two treatments, μ_1 and μ_2 for gene g. The null hypothesis is $H_0 : \mu_1 = \mu_2$. For gene g, t-statistics is defined as:

$$T_g = \frac{\bar{X_{1g}} - \bar{X_{2g}}}{\sqrt{\hat{\sigma_g}^2(\frac{1}{n_1} + \frac{1}{n_2})}} \tag{2.7}$$

where n_1 and n_2 are the size of treatment 1 and 2 , $\bar{X}_{1g}$ and $\bar{X}_{2g}$ are mean expression values of treatment 1 and 2 respectively. The pooled variance estimate $\hat{\sigma_g}^2$ is:

$$\hat{\sigma_g}^2 = \frac{(n_1 - 1)\hat{\sigma_{1g}}^2 + (n_2 - 1)\hat{\sigma_{2g}}^2}{n_1 - 1 + n_2 - 1} \tag{2.8}$$

The pvalue p_g, is the probability under the null hypothesis that the test statistic is at least as extreme as the observed value T_g. The value T_g will have a t-distrubution with $d = n_1 + n_2 - 2$ degrees of freedom.

In the ordinary t-test, the unknown parameters μ_1, μ_2 and $\hat{\sigma_g}^2$ are fixed and they are estimated from the data. However, the score T_g becomes easily very large if measured variance is small. Therefore, for microarray it is reasonable to use a modified version of the t-test [39, 106].

In moderated t-statistics, the degrees of freedom is increased and the gene-wise residual sample variances are shrunk towards a common value. This gives more stable results especially when the number of arrays is small [106].

Gene set enrichment analysis (GSEA) In typical analyses, we compare two different conditions and produce differentially expressed gene lists. However, detecting modest changes in gene expression datasets is hard due to the large number of variables, the high variability between samples and the limited number of samples.

The goal of GSEA is to determine whether members of a gene set S tend to occur toward the top (or bottom) of the list L using Kolmogorov-Smirnov statistics [111]. The gene sets are defined based on prior biological knowledge, e.g., published information about biochemical pathways or coexpression in previous experiments.

Gene signature: The group of genes whose combined expression pattern is uniquely characteristic to a given condition constitutes the gene signature of this condition. Ideally, the gene signature can be used to select a group of patients at a specific state of a disease with accuracy that facilitates selection of treatments [85].

Using gene signatures we can identify small molecules with similar effects based on Kolmogorov-Smirnov statistic. In Chapter 6, we will use this method for elucidating molecular mechanisms of Traditional Chinese Medicine(TCM) compounds and for identifying new drug candidates from TCM against different human diseases.

Unsupervised Learning: Clustering We would like to identify the groups of genes that have similar expression patterns under different conditions. Clustering approaches are widely used in analyzing gene expression data. Clustering can be used for many purposes such as (1) predicting the function of the unknown gene based on its associated cluster [40] (2) identifying regulatory motifs in the promoter regions of the genes [23, 95] (3) obtaining seeds for gene regulatory network inference [100]. An overview of clustering methods can be found in [61].

There are four main classes of clustering algorithms: Partition clustering [53], fuzzy-clustering [32, 98], hierarchical clustering [33], spectral clustering [76] and model-based clustering [122].

In partition clustering, the goal is to partition the data points into k clusters. Generally the number of the clusters k is given apriori. There are several methods to determine the number of clusters that are most appropriate for the data [114, 37]. The most common partitioning algorithm is k-means [53] which is a variant of Expectation-Maximation (EM) algorithm [33]. The objective function of k-means algorithm is to find partitioning that minimizes the intra-cluster distances and maximize inter-cluster distances. In partition clustering each data point belongs to exactly one cluster. However, in fuzzy clustering the data points can belong to more than one cluster with different membership degrees (between 0 and 1) [41].

Hiearachical clustering is extensively used in gene expression data and is shown to give valuable results [40, 109]. The main objective of the method is to group the data points hierarchically. An agglomerative hierarchical algorithm consists of the following steps. Initially the distance matrix between observations is calculated. In first step, each object is considered as a single separate cluster. In second step, two clusters which have the smallest distance to each other is merged into a one cluster. The second step is repeated until there is only one cluster left. A natural representation of hierarchical clustering is a tree, also called a dendrogram, which shows the grouping of the samples. Dendrogram can be cut at a chosen height to produce the desired number of clusters. Figure 2.7 illustrates the hierarchical clustering of gene expression data.

Model based clustering is based on the assumption that the data is collected from finite collection of populations and the data within each population can be modeled using statistical models. The model parameters are estimated using EM algorithm [33].

Spectral clustering is more robust to noise and missing data and more useful in detecting unusual patterns [76]. K-means or model-based clustering algorithms discover compact clusters whereas spectral clustering discovers connected groups.

Drawbacks of clustering algorithms: Although encouraging results have been produced using clustering algorithms [109, 7, 40], they have several limitations. Firstly, they are based on the assumption that a cluster of genes share exact same functions and behave similarly under all conditions. However, a cellular process may affect a subset of genes, only under certain conditions. Secondly, clustering assigns each gene or

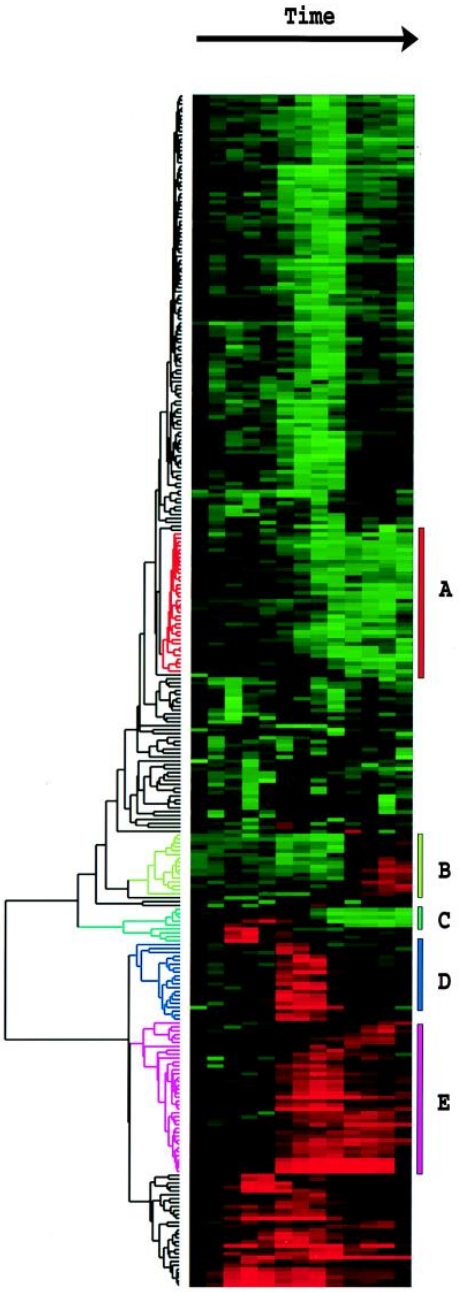


Figure 2.7: Hierarchical clustering of gene expression. A natural representation of hierarchical clustering is a tree, also called a dendrogram, which shows the grouping of the samples. Dendrogram can be cut at a chosen height to produce the desired number of clusters [40].

sample to a single cluster. However, some genes may not be active in any of the samples and some genes may participate in multiple processes.

A more refined local approach called biclustering is introduced to overcome the limitations of clustering. In chapter 3, we will explain biclustering more in detail.

We're drowning in information and starving for knowledge

John Naisbitt

Chapter 3

Biclustering

In this chapter, we extend the discussion of clustering to biclustering. In this regard, we first introduce the biclustering problem definition (section 3.1), and we provide a brief survey of existing biclustering algorithms (section 3.3). Then we discuss evaluation and validation measures for biclustering results (section 3.4). Finally, we present possible visualization methods for biclustering results (section 3.5).

Biclustering discovers subsets of genes that are co-expressed in certain samples. It is a two dimensional clustering problem where we group the genes and samples simultaneously. Figure 3.1 illustrates the difference between clustering and biclustering.

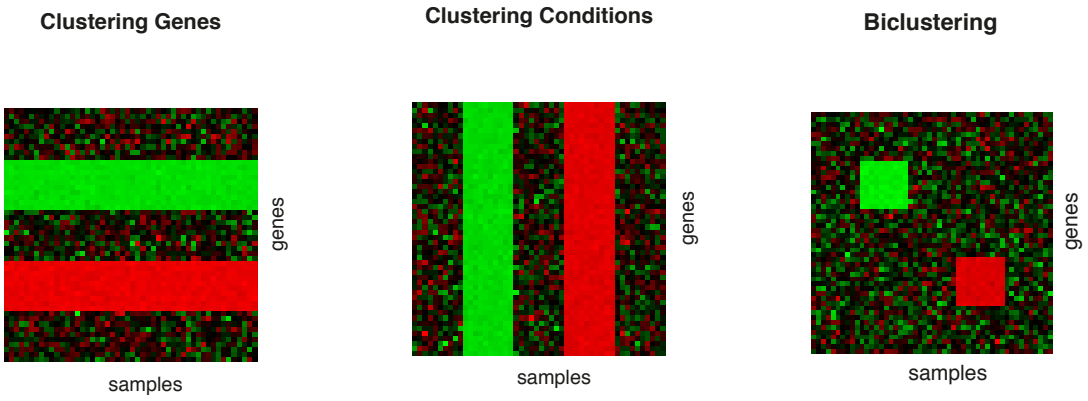


Figure 3.1: Clustering vs Biclustering. Clustering groups genes or conditions separately whereas biclustering groups subset of genes that have similar expression behavior under certain conditions. (Figures show heatmap representation of gene expression data. In heatmap, we have the genes in the rows, the conditions in the columns and the corresponding expression value is represented as color.)

Biclustering has many significant benefits.

- It can lead to a better understanding of the biological processes. Sets of genes regulated by the same transcription factor, namely module, can be detected

using biclustering. This aids in elucidating the so-called transcriptional programmes in the biological systems.

- Multi-functionality of the genes is not considered in clustering. However, genes having multiple function in different biological contexts is very common. Multi-functionality of the genes leads us to expect subset of genes to be co-expressed only under certain conditions and to be uncorrelated under the rest of the conditions.
- Biclustering has a great potential in detecting marker genes that are associated with certain tissues or diseases [91, 6]. Thus, it may lead to the discovery of new therapeutic targets.

3.1 Biclustering Problem Definition

Given an expression matrix E with genes $G = \{g_1, g_2, g_3, \ldots, g_n\}$ and samples $S = \{s_1, s_2, s_3, \ldots, s_m\}$, a bicluster is defined as $b = (G', S')$ where $G' \subset G$ is a subset of genes and $S' \subset S$ is a subset of samples. The set of biclusters are $B = \{b_1, b_2, b_3, \ldots, b_l\}$.

There are many existing biclustering algorithms and each algorithm uses a different merit function to evaluate the quality of the biclusters. Under most meaningful merit functions, the biclustering problem will be NP-complete. For example, finding a maximum size bicluster in a binary data is equivalent to finding the maximum edge biclique in a bipartite graph and maximum biclique problem is known to be NP-complete [89]. Most of the biclustering algorithms require heuristic approaches to limit the search space, so they are unlikely to find the globally optimal solution but return a good local optimum.

3.2 Classification of Biclustering Algorithms

We can classify the algorithms from several different aspects. The methods can be classified based on the *types*, *structure* of the identified biclusters and the *heuristic approaches*. Bicluster classification is summarized in Figure 3.2. More complete reviews on biclustering methods can be found in [78, 113, 48].

Types of the biclusters The algorithms can identify four different types of biclusters.

(a) *constant values:* The bicluster contains genes that have exact same constant expression values under certain conditions. In bicluster b, the value of each entry e_{ij}, $i \in G'$ and $j \in S$, is equal to constant c. The algorithms that discover biclusters with constant values are [90, 52].

(b) *constant values on rows or columns:* The bicluster contains genes (conditions) that have exact same constant expression values under subset of conditions (genes), but their expression levels differ among genes (conditions). In bicluster b with constant values on rows, the value of each entry e_{ij}, $i \in G'$ and $j \in S$, is equal either to $c + \alpha_i$ (additive model) or $c \times \alpha_i$ (multiplicative model) where α_i is the row adjustment. In bicluster b with constant values on conditions, the value of each entry e_{ij}, $i \in G'$ and $j \in S$, is equal either to $c + \beta_j$ or $c \times \beta_j$ (multiplicative model) where β_j is the column adjustment. The algorithms that discover biclusters with constant values on rows or columns are [104, 101].

(c) *coherent values:* The bicluster reveals subsets of genes with coherent values on both rows and columns. In bicluster b with coherent values, e_{ij}, $i \in G'$ and $j \in S$, is equal to constant $c + \beta_j + \alpha_i$ where β_j is the column adjustment and α_i is the row adjustment. The approaches detecting biclusters with coherent values are [25, 70, 43].

(d) *coherent evolutions:* The bicluster reveals subsets of genes with the same tendency of expression change under certain conditions. The algorithms that discover biclusters with coherent evolutions are [16, 103, 83].

Structure of the biclusters We can also classify the algorithms according to the *structure* of the identified biclusters. Mainly, the following bicluster structures can be obtained.

(a) *single bicluster:* (example algorithms are [16, 83])

(b) *exclusive rows and columns:* every row and column belongs exclusively to one of the biclusters (example algorithms are [52, 101]).

(c) *exclusive rows biclusters:* every row belongs exclusively to one of the biclusters wheras columns can belong to several biclusters (example algorithm is [104]).

(d) *exclusive columns biclusters:* every column belongs exclusively to one of the biclusters whereas rows can belong to several biclusters (example algorithm is [104]).

(e) *nonoverlapping biclusters with tree structure:* (example algorithm is [70])

(f) *nonoverlapping nonexclusive biclusters:* (example algorithm is [117])

(g) *overlapping biclusters with hierarchical structure*

(h) *arbitrarily positioned overlapping biclusters:* (example algorithms are [25, 70, 43])

(g) *checker board structure: nonoverlapping and nonexclusive biclusters:* (example algorithm is [66])

Different heuristic approaches The biclustering methods can also be classified based on their heuristic approaches.

(a) *iterative row and column clustering:* standard clustering is applied on rows and columns iteratively and then clusters are combined to form biclusters (example algorithm is [117]).

(b) *divide and conquer:* method recursively breaks down the problem into two or more sub-problems, and then the subresults are combined to find the optimal solution (example algorithm is [52]).

(c) *greedy iterative search:* method solves the optimal local solutions to find the global solution (example algorithms are [25, 16, 83]).

(d) *exhaustive bicluster enumeration:* method searches for all the possible biclusters (example algorithm is [103]).

(e) *distribution parameter identification:* the data structure is assumed to follow a statistical model and the parameters are estimated by minimizing a certain criterion through an iterative approach (example algorithm is [70]).

3.3 Overview of Existing Biclustering Algorithms

This part provides an overview of the most popular biclustering methods [25, 17, 103, 16]. The methods with different heuristic approaches are chosen for the review. In chapter 5 we will compare these methods with our proposed biclustering algorithm.

The Cheng and Church Algorithm (CC)

Model: The value of each entry e_{ij} in the gene expression matrix can be described using an additive model such as:

$$e_{ij} = \mu + \alpha_i + \beta_j \tag{3.1}$$

where μ is the background value, α_i is the row adjustment and β_j is the column adjustment.

Goal: The algorithm tries to find biclusters with a minumum *mean squared residue score*, H.

Before defining the mean residue score, let us first introduce some notations.

$$e(G', j) = \frac{1}{|G'|} \sum_{i \in G'} e(i, j) \text{ is the average of column j in the bicluster } (G', S') \tag{3.2}$$

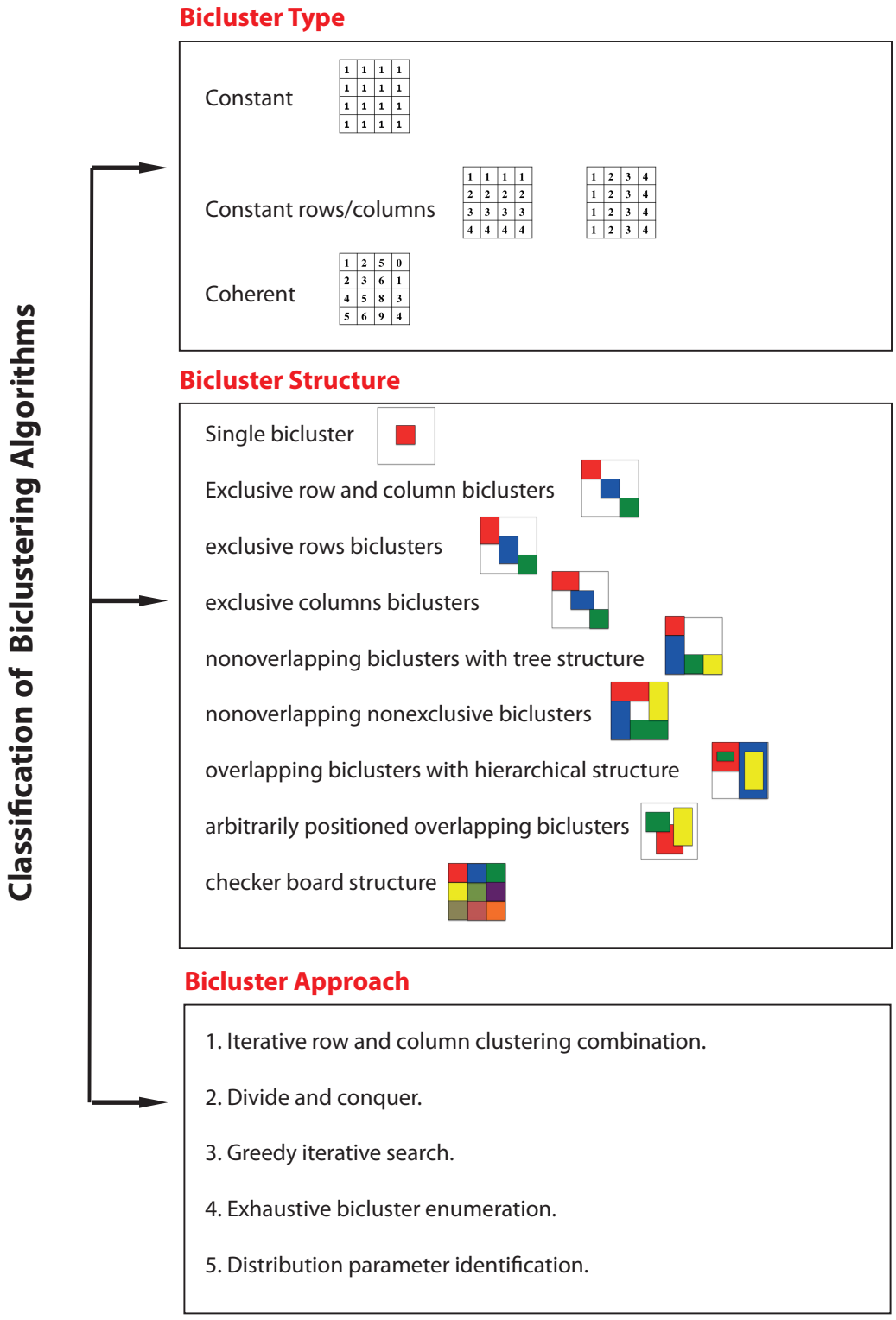


Figure 3.2: Classification of biclustering methods. The methods can be classified based on the *types*, *structure* of the identified biclusters and the *heuristic approaches*.

$$e(i, S') = \frac{1}{|S'|} \sum_{j \in S'} e(i, j) \text{ is the average of row i in the bicluster } (G', S') \quad (3.3)$$

$$e(G', S') = \frac{1}{|G'||S'|} \sum_{i \in G' j \in S'} e(i, j) \text{ is the average of the bicluster } (G', S') \quad (3.4)$$

The mean residue score H in bicluster $b = (G', S')$ is defined by:

$$H(G', S') = \frac{1}{|G'||S'|} \sum_{i \in G' j \in S'} r(i, j)^2 \quad (3.5)$$

where $r(i, j) = e(i, S') + e(G', j) - e(G', S')$ is called the residue. The residue can be derived from the additive model (equation 3.1), where $\mu = e(G', S')$, $\alpha_i = e(i, S') - e(G', S')$ and $\beta_j = e(G', j) - e(G', S')$.

A bicluster $b(G', S')$ is called a δ-bicluster if the mean residue score H is smaller than δ.

Bicluster type: The algorithm discovers biclusters with coherent values.

Bicluster discovery: The algorithm discovers one bicluster at a time.

Drawbacks: Since the CC algorithm discovers one bicluster at a time, repeated application of the method on a modified matrix is needed for discovering multiple biclusters, leading unfortunately to highly overlapping gene sets. Additionally, it may produce large biclusters containing genes that are not expressed, which shows that the mean residue score is not an appropriate measure. And, there is no evaluation of the statistical significance.

Iterative Signature Algorithm(ISA)

Goal: The ISA algorithm identifies biclusters which consist of a set of co-regulated genes and the set of conditions that induce the genes' co-regulation. It means that for each sample the average expression value of all the genes in bicluster should be surprisingly high/low and for each gene the average expression value of all the samples in bicluster should be surprisingly high/low.

Two normalized expression matrices are used, E_G and E_C. E_G is the row-normalized matrix, where the rows have mean 0 and standard deviation 1. E_C is the column-normalized matrix, where the columns have mean 0 and standard deviation 1. The average expression level of the genes in a bicluster $b = (G', C')$ is $< E_G >_{g \in G'}$. The

	Sample 1	Sample 2	Sample 3
Gene 1	0.8	1.5	2.6
Gene 2	0.4	0.7	3.2

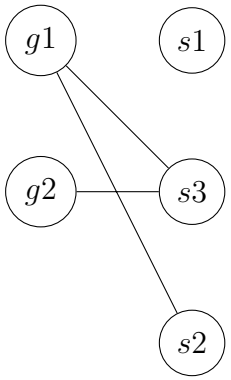


Figure 3.3: Graph representation of gene expression data (edge threshold 1). The expression data is modeled as a bipartite graph *G=(U,V,E)*. In the graph, U is the set of conditions, V is the set of genes and $(u, v) \in E$ if the expression level of v is more than the threshold 1 under condition u.

average expression level of the samples in b is $< E_C >_{c \in C'}$. A pair of subsets (G', C') is a bicluster if there exist thresholds T_C and T_G for conditions and genes, such that:

$$\exists (T_C, T_G) : \begin{cases} \forall \langle E_G \rangle_{g \in G'} > T_C \\ \\ \forall \langle E_C \rangle_{c \in C'} > T_G \end{cases} \tag{3.6}$$

Bicluster type: The algorithm discovers biclusters with coherent values.

Bicluster discovery: The algorithm discovers one bicluster at a time.

Drawbacks: There is no evaluation of the statistical significance. Additionally, two threshold parameters should be defined.

Statistical Algorithmic Method for Bicluster Analysis (SAMBA)

Model: The expression data is modeled as a bipartite graph *G=(U,V,E)* (Fig. 3.3). In the graph, U is the set of conditions, V is the set of genes and $(u, v) \in E$ if the expression level of v changes significantly in u. The edges are assigned to weights according to a statistical model, so that heavy subgraphs corresponds to biclusters with high likelihood.

Statistical Model: A bicluster is a subgraph $H = (U', V', E')$ of G. The weight of a subgraph is the sum of the gene condition pairs in it, including the edges and non-edges. The non-edges are $\bar{E}' = (U \times V) \backslash E'$. Let d_w denote the degree of the vertex $w \in U' \cup V'$. In the null model, we assume that each edge is a Bernoulli variable with a parameter $p_{u,v}$. The $p_{u,v}$ is estimated using Monte Carlo process by calculating the fraction of bipartite graphs with degree sequence identical to G that contain the edge

(u,v). The probability of the subgraph H is $p(H) = (\prod_{(u,v)\in E'} p_{u,v}) \times (\prod_{(u,v)\in \bar{E}'} 1 - p_{u,v})$. However, this score is dependent on the size of the bicluster. In order to make the score independent from the size, it is assumed that each edge occurs with a high constant probability p_c.

Then the log likelihood ratio for subgraph H is:

$$\log L(H) = \sum_{(u,v)\in E'} \log \frac{p_c}{p_{u,v}} + \sum_{(u,v)\in \bar{E}'} \log \frac{1-p_c}{1-p_{u,v}} \tag{3.7}$$

Goal: The goal of the algorithm is to find sub-graphs with high log likelihood score (equation 3.7).

Bicluster type: The algorithm discovers biclusters with coherent evolutions.

Bicluster structure: The algorithm discovers biclusters simultaneously.

Drawbacks: The algorithm is based on exhaustive enumeration of biclusters. Due to its high complexity, the number of rows the bicluster may have is restricted.

Order Preserving Sub-matrices Algorithm (OPSM)

Model: A bicluster is an ordering of the subset of samples such that the expression values of all genes in the bicluster are sorted in ascending order.

Complete model (T, π): Let $T \subset (1, 2, \ldots, m)$ be a set of conditions and $\pi = (t_1, t_2, t_3, \ldots, t_s)$ be an ordering of the conditions in T. A row supports a model (T, π), if applying the permutation π to the row results in a set of monotonically increasing values.

Partial model $(\langle t_1, t_2, \ldots, t_a\rangle, \langle t_{s-b+1}, \ldots, t_s\rangle, s)$: The first a and last b conditions are specified, but not the remaining $s - a - b$ conditions.

Goal: The goal of the algorithm is to find order preserving sub-matrices of maximum statistical significance.

Bicluster type: The algorithm discovers biclusters with coherent evolutions.

Bicluster discovery: The algorithm discovers one bicluster at a time.

Drawbacks: The model concerns only the order of values and thus makes the model quite restrictive.

Other Biclustering Algorithms In addition to ISA and OPSM, there exists other pattern recognition biclustering methods. An approach called spectral biclustering, uses the singular value decomposition technique from linear algebra [66]. The algoithm tries to identify the hidden checkerboard-like structures using a well known linear algebra technique called singular value decomposition (SVD). The method decomposes the expression matrix E into three matrices: an orthogonal matrix U, diagonal matrix Δ and the transpose of an orthogonal matrix $V^T, E = U\Delta V^T$. The columns of the matrices U and V are the eigenvectors of EE^T and E^TE, respectively. The values of Δ are the square roots of EE^T (and also E^TE). If a hidden checkboard structure exists in the data then there is at least one pair of piecewise constant eigenvectors u and v with the same eigenvalue. The drawbacks of this method are; (1) real datasets may deviate from the ideal checkerboard structure (2) the model assigns all the genes to a bicluster which may not be the case in reality and the biclusters are disjoint. Another algorithm called Binary Inclusion Maximal Algorithm (BIMAX) tries to identify all maximal biclusters where none of the biclusters are not completely contained in any other bicluster [90]. Qualitative Biclustering algorithm (QUBIC) discovers statistically significant biclusters from datasets containing tens of thousands of genes and thousands of condition [71].

There are other probabilistic and generative methods like SAMBA algorithm. An approach called plaid model is a statistical approach performing distribution parameter identification [70]. Previously, we formulated the expression value e_{ij} as an additive model (equation 3.1) or as a multiplicative model. However, both of these models do not take into account the interactions between biclusters. Plaid model represents the contributions of different biclusters on the expression value e_{ij}. Normalization of the data is very important for the performace of this algorithm. Later on, the plaid models are generalized to fully generative models called Bayesian BiClustering model (BBC) [47]. BBC avoids high percentage of overlap in the plaid models by constraining the overlap of biclusters to only one dimension. In another approach, Gibbs sampling is used to estimate the parameters of the plaid model [104]. Another biclustering algorithm based on probabilistic relational models, combine probabilistic modeling and relational logic. The algorithm can incorporate additional prior knowledge [101].

3.4 Validation and Evaluation of Biclustering Algorithms

In section 3.3, we reviewed some of the existing biclustering methods. Each of these algorithms discover different bicluster types, structures and they have different objective functions. Comparing and validating these different methods is still not very clear and remains as a challenge. There has been little work on comparison and validation of the biclustering results [90]. In clustering validation, both internal and

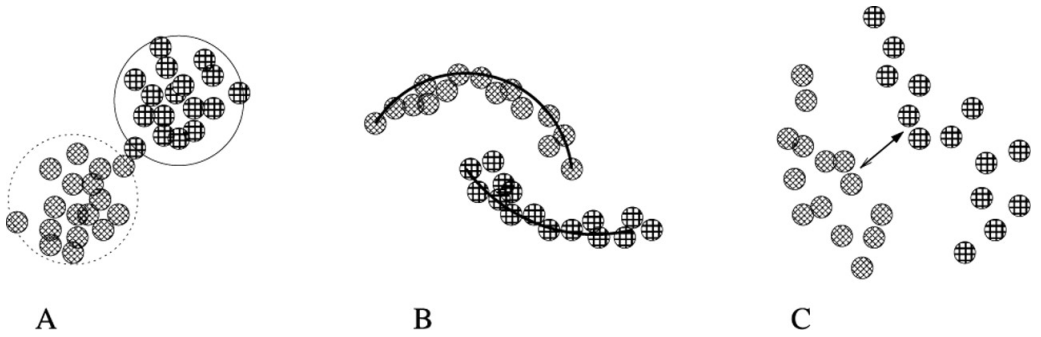


Figure 3.4: Internal measures for cluster validation. (A) compactness: assesses cluster homogeneity, with intra-cluster variance (B) connectedness: assesses how well a given partitioning groups data items together with their nearest neighbours in the data space and (C) seperation: quantifies the degree of separation between clusters. The figure is adapted from [50].

external measures are used for validation [50, 11, 21, 31]. On the other hand, in bicluster validation internal measures are not used extensively, mostly external indices are used.

Internal validation measures use the information in the data to assess the quality of the clusters. They evaluate the separation, compactness and connectivity of the data (Fig. 3.4). Compactness assesses cluster homogeneity, with intra-cluster variance. Connectedness assesses how well a given partitioning groups data items together with their nearest neighbours in the data space. Separation quantifies the degree of separation between clusters [50].

External Measures External indices use prior knowledge to evaluate biclustering results.

Biological data

In biological datasets external indices are based on the prior biological knowledge. Below we review the most common measures for assessing biological significance.

1. **Gene Ontology(GO) Term Enrichment:** Gene Ontology is a collection of controlled vocabularies describing the biology of a gene product in any organism [10]. We would like to use GO terms to evaluate the biological significance of the genes in a given bicluster. More precisely, in a given set of genes with size n, we would like to determine if there is a GO term that is more represented than what it would be by chance only. We calculate the significance of a specific GO term using hypergeometric test.

   Suppose the total number of genes is N, the total number of genes associated with the GO term of interest is m and the number of genes in the cluster

associated with the GO term of interest is k. A random variable X follows the hypergeometric distribution with parameters N, m and n:

$$P(X = k) = \frac{\binom{m}{k}\binom{N-m}{n-k}}{\binom{N}{n}} \tag{3.8}$$

2. **KEGG Pathway Enrichment:** We can also calculate the KEGG Pathway enrichment of the genes in a bicluster with the same method as in GO Term enrichment (equation 3.8). Instead of GO Terms here we use KEGG Pathway Terms [63].
3. **Transcription Factor Binding Site(TFBS) Enrichment:** The transcription factor binding site enrichment in the promoter regions of the genes in a cluster can be calculated using hypergeometric test like in GO Term or KEGG Pathway enrichment (equation 3.8).
4. **Protein Networks:** The average of the total shortest distances between connected gene pairs in the protein network of the given gene set is calculated. The distance is expected to be small for functionally related gene sets [120, 88].

Synthetic Data

The simulated data is studied to show the performance of our algorithm in recovering implanted biclusters. In order to assess the performance of different biclustering algorithms several measures can be used.

1. **Prelic score:** The measure introduced by Prelic et al. calculate the similarity between the computed biclusters and the implanted biclusters [90]. Prelic score is defined as:

   Bicluster recovery score:

   $$S_G(M, M_{opt}) = \frac{1}{|M|} \sum_{G \in M} \max_{G_{opt} \in M_{opt}} \frac{|G_{opt} \cap G|}{|G_{opt} \cup G|} \tag{3.9}$$

   Bicluster match score:

   $$S_{G_{opt}}(M_{opt}, M) = \frac{1}{|M_{opt}|} \sum_{G_{opt} \in M_{opt}} \max_{G \in M} \frac{|G_{opt} \cap G|}{|G_{opt} \cup G|} \tag{3.10}$$

   where M_{opt} is the set of true biclusters, M is the set of computed biclusters, G is the gene sets within the biclusters M and G_{opt} is the gene sets within the biclusters M_{opt}. The score $S_G(M, M_{opt})$ measures the relevance of the predicted biclusters in gene dimension. The score $S_{G_{opt}}(M_{opt}, M)$ measures how well each of the true biclusters is recovered by the biclustering algorithm. The maximum value for both scores is 1. The drawback of this measure is that different number of biclusters between M_{opt} and M is not penalized.

2. **Hochreiter score:** Hochreiter et al. calculate a consensus score using the following steps [54]. Firstly, given the two sets of biclusters M_{opt} and M, the similarities between all possible pairs of biclusters are computed. Secondly, each bicluster in M_{opt} is assigned to a bicluster in M using Munkres algorithm [82]. Munkres algorithm yields a match between M_{opt} and M where no two pairings share the same bicluster. Finally, different numbers of biclusters are penalized by dividing the sum of the similarities by the number of biclusters in the largest set.

3.5 Visualizing Biclusters

Biclustering algorithms may discover tens, hundreds or even thousands of biclusters with varying degrees of overlap. A visualization approach helps us interpret the biclustering results and gain insight into their structures.

In typical clustering, we use heatmaps or dendrograms for visualization. In a heatmap, rows represent genes, columns represent conditions and we have the genes in the rows, the conditions in the columns and the corresponding expression value is represented with a coding for the intensity. Another method named dendrogram is a tree diagram frequently used to illustrate the arrangement of the clusters produced by hierarchical clustering.

In the case of biclustering, visualizing one single bicluster is possible using heatmaps. However simultaneously visualizing multiple biclusters is very complicated since they can overlap. There are several studies about biclustering visualization. Below we briefly go over these methods.

BiVoc: BiVoc algorithm rearranges rows and conditions of the original dataset in order to represent the biclusters with minimum space. The output matrix of BiVoc, may have repeated rows and/or columns from the original matrix [46](Fig. 3.5).

Parallel Coordinate Plots: A parallel coordinate plot is another visualization technique where the conditions(genes) are visualized as vertical axes and genes(conditions) as lines joining the corresponding expression values. In parallel coordinate plots, the profile of the conditions(genes) that are included in a bicluster are shown in black, the other conditions(genes) in gray. This aids to visualize the expression difference between the conditions(genes) in a bicluster compared to the rest of the conditions(genes) (Fig. 3.6).

BiCoverlapper: BiCoverlapper visualizes large number of biclusters simultaneously which allows us to explore the overlapping structure of the biclusters. The visualization is done using force directed graph layout [97]. Figure 3.7 explains the visualization method of BiCoverlapper.

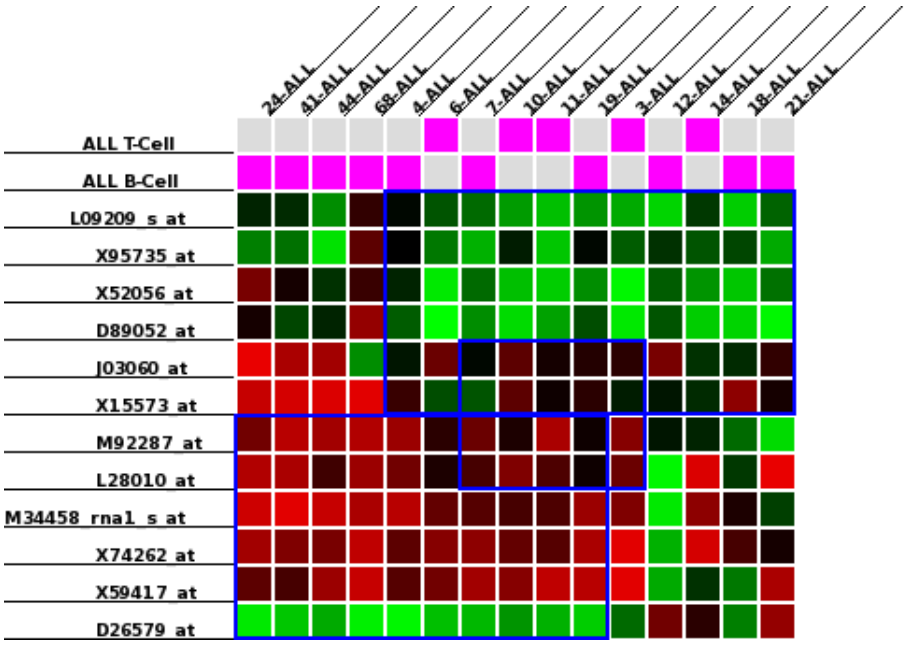


Figure 3.5: Visualizing the biclusters using BiVoc algorithm. The blue rectangles represent the biclusters. The figure is adapted from [46]

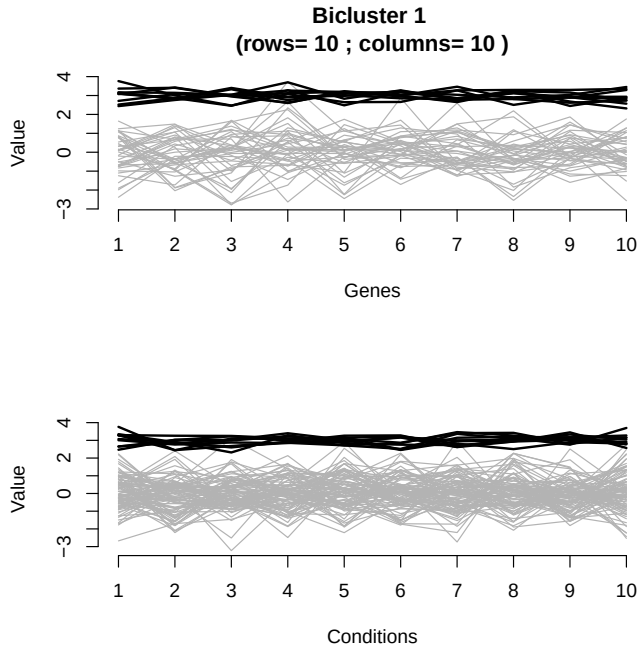


Figure 3.6: Visualizing the biclusters using parallel coordinates. The first one is the parallel coordinates plot of genes whereas the second one is the parallel coordinates plot of conditions.

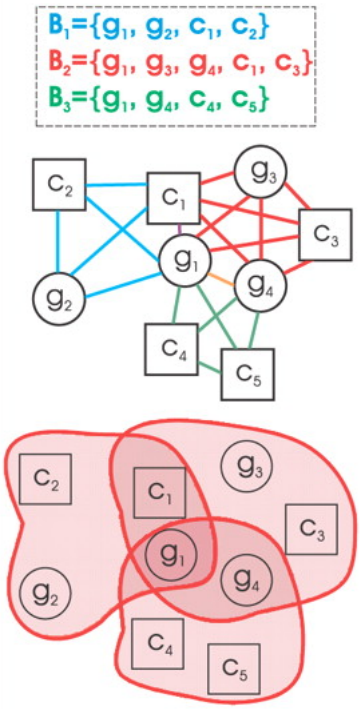


Figure 3.7: Visualizing the three biclusters using BiCoverlapper algorithm. Bicluster B_1 contains gene g1, gene g2, condition c1 and condition c2. Bicluster B_2 contains gene g1, gene g3, gene g4, condition c1 and condition c3. Bicluster B_3 contains gene g1, gene g4, condition c4 and condition c5. Condition c1 and gene g1 appear on both bicluster B_1 and bicluster B_2 so the edge between them is shorter. Intersecting areas in the graph are more opaque to highlight overlapping.The figure is adapted from [97]

Knowledge discovery in databases is the non-trivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data

Fayyad

Chapter 4

Frequent Itemset Mining

In this chapter, we present a well known data mining approach called frequent itemset (section 4.1). The running time for brute force approaches for detecting maximally frequent itemsets is exponential, therefore there is a need for more efficient implementations. In section 4.2, we explain several efficient algorithms for identifying frequent itemsets. In chapter 5, we will use an efficient algorithm called MAFIA in our proposed biclustering algorithm. MAFIA algorithm avoids an exhaustive enumeration of all the candidate gene sets by several prunning techniques (section 4.2).

Identifying maximally frequent itemsets (MFI) is a well known data mining problem. A typical example of MFI is the customer market basket behavior analysis. In market basket behavior analysis, we collect the items purchased by customers. Instead of counting how often a given item purchased (e.g., milk, bread, cereal), we count how often multiple items purchased together (e.g., both milk and cereal together). This analysis is useful for discovering customer purchasing patterns and forming marketing, sales and operation strategies.

4.1 Definitions

Let $G = \{g_1, g_2, g_3, \ldots, g_n\}$ be a set of n distinct items.

Definition 1. *An itemset I is a subset of G.*

Definition 2. *The number of items in an itemset I is called the length of an itemset. Itemsets of some length k are referred to as $k-itemsets$.*

Definition 3. *A transaction database T is a set of transactions with unique transaction identiers. For example, collections of items bought by customers constitute a transaction database T, where a customer with a set of items represents a transaction.*

Definition 4. *The fraction of transactions in T that contain an itemset I is called the support of an itemset.*

Definition 5. *A k-itemset whose support s is greater than or equal to a minimum support threshold is called a frequent itemset.*

TID	Items
1	Bread, Milk
3	Milk, Diaper, Beer
4	Bread, Diaper, Beer
5	Beer, Diaper
6	Milk, Diaper, Beer

Table 4.1: Transaction Database: Itemsets bought by each customer (In microarrays, customers are experiments and items are genes.)

$\Longrightarrow$

Itemset	Support	Length
Bread	2/6	1
Milk	3/6	1
Diaper	4/6	1
Beer	3/6	1
Bread, Milk	1/6	2
Bread,Diaper	1/6	2
Bread,Beer	0/6	2
Milk, Diaper	2/6	2
Milk,Beer	2/6	2
Diaper,Beer	4/6	2
Bread, Milk,Diaper	0/6	3
Bread, Milk,Beer	0/6	3
Bread, Diaper,Beer	1/6	3
Milk,Diaper,Beer	2/6	3
Bread,Milk,Diaper,Beer	0/6	4

Table 4.2: Candidate itemsets and their corresponding support and length. $(Milk, Diaper, Beer)$ is a frequent itemset with support and length threshold 2/6 and 3 respectively.

Frequent itemset mining was first proposed by Agrawal et al. for market basket analysis [4]. Table 4.1 illustrates the itemsets that each customers buys in a market. The simplest method for detecting maximally frequent itemsets is the brute force approach in which each itemset in the transaction database is a candidate frequent set. Then, we count the support of each itemset by scanning the database. Since running time for the brute force approach is exponential, there is a need for more efficient implementations where we can reduce the number of candidates, number of transactions and number of comparisons. Below we review two efficient algorithms for detecting maximally frequent itemsets.

4.2 Frequent Itemset Mining Algorithms

Apriori Algorithm Apriori algorithm is an efficient approach for finding maximum frequent sets [4]. It avoids an exhaustive enumeration of all candidate gene sets by monotonicity principle. The monotonicity principle states that if a set is infrequent all of its supersets must be also infrequent and if a set is frequent all of its subsets must also be frequent (Fig. 4.1). The monotonicity principle holds due to the following property of the support measure:

$$\forall X, Y : (X \subset Y) \Longrightarrow support(X) \geq support(Y) \tag{4.1}$$

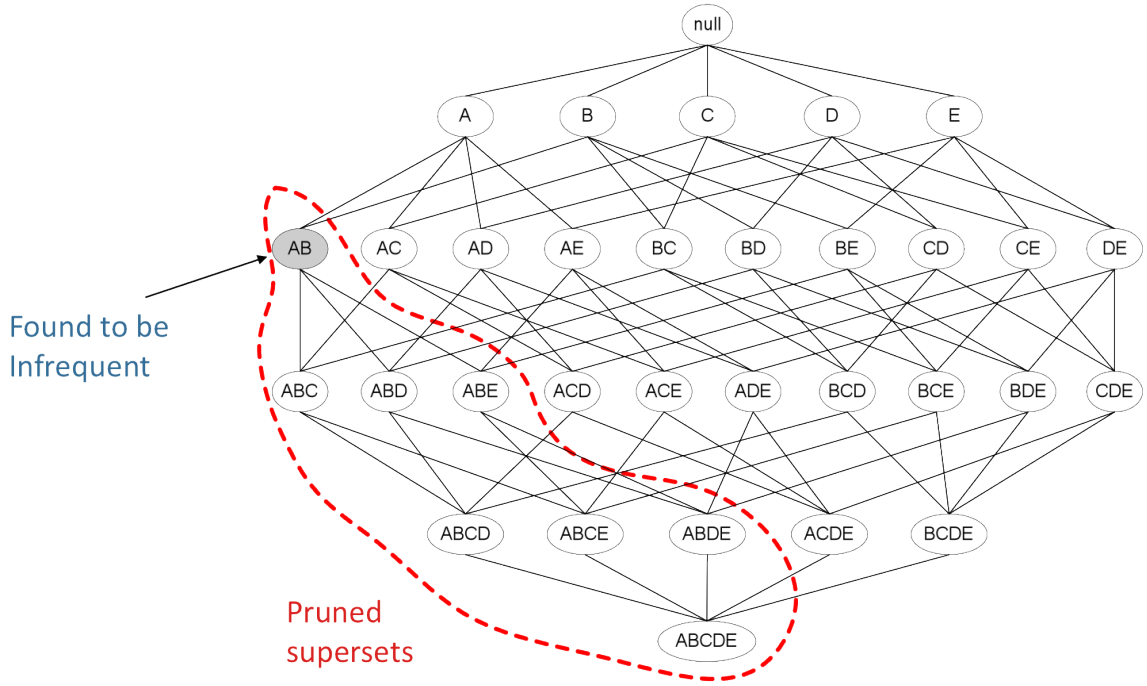


Figure 4.1: Pruning the search space using the monotonicity principle.The monotonicity principle states that if a set is infrequent, all of its supersets must be also infrequent and if a set is frequent all of its subsets must also be frequent.

where X and Y are itemsets. This property states that the support of an itemset never exceeds the support of its subsets. This is known as the anti-monotonic property of the support.

The Apriori algorithm uses a breadth-first search approach, meaning finding all k-itemsets before considering $k+1$ itemsets. It generates candidate itemsets of length k from itemsets of length $k-1$. Then, it prunes the candidates which have an infrequent subset from the candidate itemsets of length k. After that, it scans the transaction database to determine frequent itemsets among the candidates. Let us define C_k as a candidate itemset of size k, L_k as a frequent itemset of size k and T as the transaction database. Below is the pseudocode of the algorithm 1.

All 2^k subsets for each k-itemsets have to be counted in Apriori algorithm, therefore a more efficient implementation for finding maximal frequent itemsets is required. There are many efficient frequent itemset algorithms [45, 24, 72, 3, 124, 49].

Maximal Frequent Itemset Algorithm (MAFIA) In chapter 5, we will use MAFIA algorithm, it is published by [24], for our proposed biclustering algorithm. MAFIA uses a depth-first traversal of the gene set lattice with effective pruning mechanisms. It is very efficient especially when the sets in the database are very long. It avoids an exhaustive enumeration of all the candidate gene sets by the techniques described below.

Algorithm 1: Apriori Algorithm

```
Input  : Transaction database T, support threshold supp
Output: Frequent itemsets L
L_1 ← {1-itemsets that appear in more than support supp};
k ← 2;
while L_{k-1} ≠ do
    C_k ← Generate(L_{k-1});
    foreach transaction t ∈ T do
        C_t ← Subset(C_k,t);
        foreach c ∈ C_t do
            // count vector holds the support value of each
               candidate set c
            count[c] ← count[c] + 1
        end
    end
    L_k ← {c ∈ C_k| count[c] ≥ supp};
    k ← k + 1;
end
```

Pruning Techniques: In the tree, the itemset identifying the node is called **head** and the extensions of the node are called **tail**. The tail contains the elements that are alphabetically larger than the head. In a simple depth-first traversal approach, the supersets of the infrequent itemsets are pruned. In order to prune out the search space further, the following pruning techniques are used.

Frequent Head Union Tail (FHUT): The largest possible frequent itemset contained in the subtree rooted at **n** is **n**'s Head Union Tail(HUT) [15]. If a node's HUT is frequent, then there is no need to explore the subsets of HUT. Therefore, the entire subtree rooted at n can be pruned. FHUT can be computed by exploring the leftmost path in the subtree rooted at each node. Below is the pseudocode for FHUT algorithm [24].

HUTMFI Superset Pruning: If the superset of node's HUT is already identified as a Maximum Frequent Itemset (MFI) then HUT must also be frequent and the subtree rooted at n can be pruned away.

Parent Equivalence Prunning (PEP): Let us denote the transactions/samples containing the item x as $t(x)$. The node $n's$ head is x and the node $n's$ tail element is y. if $t(x) > t(y)$ then the item in the tail y, is moved to the head x because all the maximum frequent itemsets containing x also contain y. Below is the pseudocode for PEP [24].

Algorithm 2: FHUT: Frequent Head Union Tail

```
Input  : node C, MFI, Boolean isHut
foreach item i in C.tail do
    newNode ← C union i
    isHut ← whether i is the first item in the tail
    if newNode is frequent then
        FHUT(newNode, MFI, isHut)
if C is a leaf and C.head is not in MFI then
    Add C.head to MFI
if isHut and tail is frequent then
    Stop search and go back up tree
```

Algorithm 3: HUTMFI

```
input  : Current node C,MFI
HUT ← C union i
if HUT is in MFI then
    Stop searching and return
foreach item i in C.tail do
newNode ← C union i
if newNode is frequent then
    HUTMFI(newNode, MFI)
if C is a leaf and C.head is not in MFI then
    Add C.head to MFI
```

Algorithm 4: PEP: Parent Equivalence Prunning

```
input  : node C, MFI
foreach item i in C.tail do
    newNode ← C union i
    if newNode.support equal to C.support then
        Move i from C.tail to C.head
    else
        newNode is frequent
    PEP(newNode, MFI)
if C is a leaf and C.head is not in MFI then
    Add C.head to MFI
‘
```

The full algorithm for MAFIA: The pseudocode for the full MAFIA algorithm is:

Algorithm 5: The full MAFIA algorithm

```
input  : Current node C, MFI, Boolean isHUT
HUT ← C.head union C.tail
if HUT is in MFI then
    Stop searching and return
// Expand all children using C.tail and reorder children by
   increasing support
// Move children from C.tail to C.head with PEP pruning
foreach item i in C.trimmedTail do
    newNode ← C union i
    isHUT ← whether i is the first item in the tail
    MAFIA(newNode, MFI, isHUT)
if isHUT and tail is frequent then
    Stop search and go back up tree
if C is a leaf and C.head is not in MFI then
    Add C.head to MFI
```

Chapter 5

Biclustering of large-scale datasets: DeBi algorithm

In this chapter, first we introduce our novel fast biclustering algorithm called DeBi (Differentially Expressed BIclusters) (section 5.1). The pseudocode of the algorithm is presented in section 5.2. Then, we evaluate the performance of DeBi on a yeast dataset, on synthetic datasets and on human datasets. We show that DeBi compares well with existing biclustering methods such as BIMAX, SAMBA, CC, OPSM, ISA and QUBIC based on biological validation measures (section 5.3). And finally, running time analysis of the algorithms is presented in section 5.4

In this chapter we will introduce our fast biclustering algorithm called DeBi that utilizes differential gene expression analysis [102]. In DeBi, a bicluster has the following two main properties. Firstly, a bicluster is a maximumples in the bicluster and the samples not in the bicluster. Differentially expressed biclusters lead to functionally more coherent gene sets compared to standard clustering or biclustering algorithms. We developed an algorithm for binarized gene expression data.

There are several advantages of the DeBi algorithm. Firstly, the algorithm is capable of discovering biclusters on very large data sets such as the human connectivity map(cMap) [68] data with 22283 genes and 6100 samples in reasonable time. Secondly, it is not required to define the number of biclusters apriori [90, 25, 17].

We evaluated the performance of DeBi on a yeast dataset [58], on synthetic datasets [90], on the connectivity map(cMap) dataset which is a reference collection of gene expression profiles from human cells that have been treated with a variety of drugs [68], gene expression profiles of 2158 human tumor samples published by Expression Project for Oncology (expO), on diffuse large B-cell lymphoma (DLBLC) dataset [94] and on gene sets from the Molecular Signature Database (MSigDB) C2 category. We show that DeBi compares well with existing biclustering methods such as BIMAX, SAMBA, CC, OPSM, ISA and QUBIC [90, 112, 25, 16, 17].

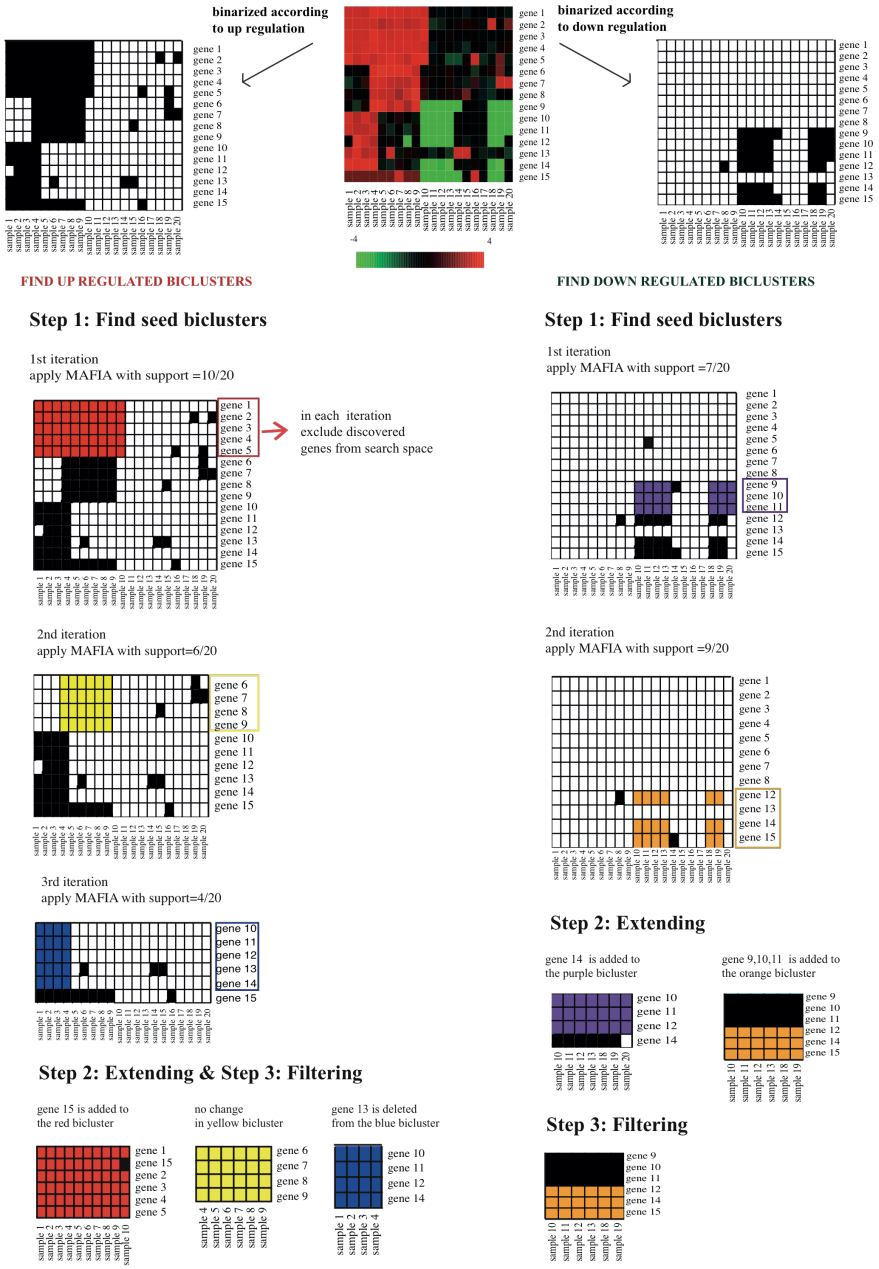


Figure 5.1: Illustration of DeBi algorithm. The algorithm is run on two different binarized datasets. One is the binarized data based on up regulation and the other is the binarized data based on down regulation. In Step 1, seed biclusters identified within each support value going from high to low. For the binarized data based on up regulation, in the 1st iteration, red gene set with support value 10/20 is detected and excluded from the search space. Similarly, in the second and third iterations yellow and blue clusters with support values, respectively 6/20 and 4/20, are found. In Step 2, seed gene sets are improved based on genes' association strength. Gene 15 is added to the red bicluster because the p-value returned by the Fisher exact test is smaller than α and gene 13 is deleted because the p-value returned by the Fisher exact test is higher than α . None of the discovered biclusters have an overlap of the gene $\times$ sample area of more than 50%.

5.1 DeBi Algorithm

Given an expression matrix E with genes $G = \{g_1, g_2, g_3, \ldots, g_n\}$ and samples $S = \{s_1, s_2, s_3, \ldots, s_m\}$ a bicluster is defined as $b = (G', S')$ where $G' \subset G$ is a subset of genes and $S' \subset S$ is a subset of samples. DeBi identifies functionally coherent biclusters $B = \{b_1, b_2, b_3, \ldots, b_l\}$ in three steps. Below we describe each step in detail. An overview of the DeBi algorithm is shown in Figure 5.1.

The DeBi algorithm is based on a well known data mining approach called Maximal Frequent Item Set [24], which has been reviewed in Chapter 4. We will refer to this as Maximal Frequent Gene Set, as given by our problem definition.

Preliminaries The input gene expression data is binarized according to either up or down regulation. Let E^u and E^d denote the up and down regulation binary matrices, respectively. Then the entries e^u_{ij} of $E^u$are defined as follows:

$$e^u_{ij} = \begin{cases} 1 & \text{if gene } i \text{ is } c \text{ fold up regulated in sample } j \\ 0 & \text{otherwise} \end{cases} \tag{5.1}$$

and the entries of e^d_{ij} of $E^d$are defined analogously with a c-fold down-regulation cut-off. The fold change cut-off c will typically be set to 2.

Step 1. Finding seed biclusters by Maximal Frequent Gene Set Algorithm The DeBi algorithm, identifies the seed gene sets by iteratively applying the maximal frequent gene set algorithm. We first define the term *support*, which we will later use in the algorithm. The *support* of the gene g_i, $i = 1, \ldots, n$, is defined as follows:

$$supp(g_i) = \frac{1}{m} \sum_{j=1}^{m} e_{ij} \tag{5.2}$$

In other words, the *support* is the proportion of samples for which the gene-vector $e_{i.}$ is 1. This is further extended to sets of genes. Let $G'_v = \{g_1, \ldots, g_k\}$ be the v^{th} gene-set. For a set of gene-vectors we define their *phenotype vector* C_v as their element-wise logical AND:

$$C_v = \wedge(e_{1.}, \ldots, e_{k.}) \tag{5.3}$$

The *support* of the gene set is then defined as the fraction of samples for which the phenotype vector is 1.

A gene set $G_v^{'}$ is $(c_1, c_2) - frequent$ iff its support $supp(G_v^{'})$ is larger than c_1 and the cardinality $|G_v^{'}|$ above c_2. When $c_1$and c_2 are not in focus, we will simply speak of a frequent gene set. A gene set is *maximally frequent* iff it is frequent and no superset of it is frequent.

In the first step of the DeBi algorithm, MAFIA is iteratively applied to the binary matrix successively reducing the support threshold. Initially, MAFIA is applied to the full binary matrix $E^u(E^d)$ with support value $(c_1)_0$ equal to support value of the gene with the highest support. In iteration k, MAFIA is applied with support value threshold of $(c_1)_k = (c_1)_{k-1} - \frac{1}{m}$. The identified maximally frequent sets are added to the set of seed gene sets B and the genes in B are deleted from the binary matrix $E^u(E^d)$. In each iteration MAFIA is applied to the modified matrix $E^{u'}(E^{d'})$. The process is repeated until a user defined *minumum support* parameter is reached.

Step 2. Extending the biclusters In the second step of DeBi, the identified seed gene sets $G^{'} = \{G_1^{'}, G_2^{'} \dots G_l^{'}\}$ are extended using a local search. For each bicluster $b_v = (G_v^{'}, S_v^{'}), v = 1, \dots, l$, we have the binary phenotype vector $C_v = \wedge(e_{1.}, \dots, e_{k.}) = (C_{v1}, \dots, C_{vm})$. The entries of C_v indicate the indices of the bicluster samples. If $C_{vj} = 1 \Rightarrow s_j \in S_v^{'}$, $j = 1, \dots, m$, i.e. that the sample s_j belongs to the bicluster b_v. The gene g_i, $i = 1, \dots, n$, is an element of gene set $G_v^{'}$ if $e_{i.}$ is associated with C_v. We evaluate the association strength between the phenotype vector of a bicluster and another gene using Fisher's exact test on a 2x2 contingency table. The cells of the contingency table count how often the four possibilities of the phenotype vector containing a 1 or a 0 and the gene-vector containing a 1 or a 0 occur. Fisher's exact test then tests for independence in the contingency table and thus among the two vectors. In this step, we can also return back to the original expression values and apply t-test between the samples in the bicluster and the samples not in the bicluster.

A gene $g_i, i = 1, \dots, n$ is added, to the gene set $G_v^{'}$ if the pvalue p_{g_i} returned by the Fisher exact test is lower than the parameter α. It gets deleted from b_v if the probability is higher than α and added to b_v if the probability is smaller than α. For this procedure the association probability p_{g_i} with the bicluster needs to be calculated for each gene. However, we reduce the computational effort using the monotonicity property of the hypergeometric distribution. We precompute cut-off values on the contingency table entries that yield a p-value just higher than α. Let $\sigma_{1,IN}$ and $\sigma_{1,OUT}$ denote the number of 1's a gene-vector has in the bicluster samples and the number of 1's a gene-vector has outside the bicluster samples, respectively. We find the minimal $\sigma_{1,IN}$ and maximal $\sigma_{1,OUT}$ at this border. Then, we apply Fisher's exact test only to those genes which have $\sigma_{1,IN} > min\sigma_{1,IN}$ and $\sigma_{1,OUT} < max\sigma_{1,OUT}$.

Step 3. Filtering the biclusters In the last step, we turn to the sometimes very complicated overlap structure among biclusters. The goal is to filter the set of biclusters such that the remaining ones are large and overlap only little. The size of

a bicluster is defined as the number of genes times the number of samples in the bicluster, $|G'_v| \times |S'_v|$. Two biclusters overlap when they share common samples and genes. The size of the overlap is the product of the number of common samples and common genes. To filter out biclusters that are largely contained in a bigger bicluster, we start with the largest bicluster and compare it to the other biclusters. Those biclusters for which the overlap to the largest one exceeds L% (typically 50%) of the size of the smaller one are deleted. This is then repeated starting with the remaining second-largest bicluster and so on.

Choosing the optimum alpha parameter To formulate an optimality criterion for α one requires an inherent measure of the quality of a set of biclusters. To this end, for a bicluster v, we define its score I_v as the negative sum of the log p-values of the included genes, where the individual p_g is the p-value from the Fisher exact test, $I_v = -\sum_{g\epsilon\ G'_v}(\log p_g)$.

However, this bicluster score I_v depends on the size (number of genes x number of conditions) of the bicluster and in order to make it comparable between biclusters one needs to correct for the size. We compute the expected bicluster score through a randomization procedure. A large number, say 500, random phenotype vectors having the same number of 1s as the bicluster has conditions is generated. For these random phenotype vectors a Fisher exact test p-value with respect to each gene in the bicluster is computed. One obtains a random I_v score by adding log p-values over the genes of the bicluster. The mean of these random bicluster scores is the desired estimator. Finally, a normalized NI_v score is definded by dividing I_v by this estimated mean and the total biclustering score CS is defined as the sum of NI_v normalized scores of all discovered biclusters $CS = \sum_{v\epsilon\ I}(NI_v)$. This score serves to distinguish between different choices of α. The program is run under $\alpha = \{10^{-2}, 10^{-3}, ..., 10^{-100}\}$ and we choose the α that maximizes CS.

5.2 DeBi Algorithm Pseudocode

Given an expression matrix E with genes $G = \{g_1, g_2, g_3, \ldots, g_n\}$ and samples $S = \{s_1, s_2, s_3, \ldots, s_m\}$ a bicluster is defined as $b = (G', S')$ where $G' \subset G$ is a subset of genes and $S' \subset S$ is a subset of samples. DeBi identifies functionally coherent biclusters $B = \{b_1, b_2, b_3, \ldots, b_l\}$ in three steps. The input gene expression data E is binarized according to either up or down regulation. Let E^u and E^d denote the up and down regulation binary matrices, respectively. The DeBi takes E^u or E^d as an input.

Algorithm 6: FindSeeds: find the seed biclusters

input : Binarized gene expression data according to up regulation: E^u, Global variables (minimum support value: $minc_1$, minumum number of genes: c_2, overlap)
output: Seed biclusters B

```
B ← ∅
E' ← E^u
// c_1 is set to support value of the gene with the highest support
c_1 ← max_i (1/m) Σ_{j=1}^{m} e_ij
while minc_1 ≤ c_1 do
    B' ← MAFIA(E', c_1, c_2)
    I ← ∅
    foreach b ∈ B' do
        // get the genes in bicluster b
        I ← I ∪ (G' ∈ b)
    // remove the discovered genes from the expression matrix E'
    E' ← E'\I
    B ← B ∪ B'
    c_1 ← c_1 − 1
```

Algorithm 7: Extend: extend the seed biclusters

input : E^u, seed biclusters: B, p-value threshold: α
output: extended seed biclusters: B, vector of bicluster scores for each bicluster in B: *score*

```
// for all seed biclusters, B ={b_1, b_2, b_3, ..., b_l}
for i ← 1 to l do
    // get the phenotype vector of bicluster i containing k genes
    C_i ← ∧(e_1., ..., e_k.)
    // for all the genes in E^u
    for j ← 1 to n do
        pval ← FisherExactTest(C_i, e_j.)
        if pval ≤ α then
            // bicluster b_i = (G'_i, S'_i)
            G'_i ← G'_i ∪ g_j
            // score is the vector of bicluster scores for each bicluster in B
            score [i] ← score[i] − log(pval)
```

Algorithm 8: NormalizedBicScore: the sum of normalized bicluster scores

input : B, number of permutations: num, vector of bicluster scores for each bicluster in B: $score$
output: sum of normalized bicluster scores in B: CS

```
CS ← 0
for i ← 1 to l do
    NS ← 0 // NS is normalized score of bicluster i
    // get the phenotype vector of bicluster i containing k genes
    C_i ← ∧(e_1., ..., e_k.)
    for j ← 1 to num do
        C'_i ← Permute(C_i)
        foreach g ∈ G'_i do
            pval ← FisherExactTest(C'_i, e_j.)
            // perm vector holds the estimated mean score of each
               bicluster in B
            perm [j] ← perm[j] − log(pval)
    NS ← score[i]/mean(perm)
    CS ← CS + NS
```

Algorithm 9: DeBi: The full DeBi algorithm

input : number of permutations: num, min support value: $minc_1$, min number of genes in the bicluster: c_2, max overlap: $overlap$
output: The final biclusters: B_3

```
// BicScore is the vector of normalized biclustering scores for
   different α values
B_1 ← FindSeeds(E^u(E^d))
maxScore ← 0
for i ← 2 to 100 do
    α ← pow(10, −i)
    (B_2, score) ← Extend(E^u(E^d), B_1, α )
    B_3 ← Filter(B_2, overlap )
    BicScore[i] ← NormalizedBicScore(B_3, num, score )
    if maxScore ≤ BicScore[i] then
        maxScore ← BicScore[i]
        maxAlpha ← α
B_2 ← Extend(B_1, maxAlpha )
B_3 ← Filter(B_2, overlap )
```

5.3 Application on Biological Data

We have evaluated our algorithm on six data sets (a) Prelic's benchmark synthetic data sets with implanted biclusters [90] (b) 300 different experimental perturbations of S. cerevisiae [58] (c) diffuse large B-cell lymphoma (DLBCL) dataset [94] (d) a reference collection of gene-expression profiles from human cells that have been treated with a variety of drugs [68] (e) gene expression profiles of 2158 human tumor samples published by Expression Project for Oncology (expO) (`http://www.intgen.org/expo.cfm`) (f) gene sets from the Molecular Signature Database (MSigDB) C2 category. The synthetic data is studied to show the performance of our algorithm in recovering implanted biclusters. Additionally, the effect of overlap between biclusters and noise on the performance of the algorithm can be studied using the synthetic data. The yeast and human gene expression datasets are studied to evaluate the biological relevance of the biclusters from several aspects.We used a fold-change of 2 for binarizing the datasets. The set of biclusters generated by all the algorithms are filtered such that the remaining ones have a maximum overlap of 0.5. (unless specified otherwise)

First, for each bicluster we calculated the statistically significantly enriched Gene Ontology (GO) terms using the hypergeometric test. We determined the proportion of GO term enriched biclusters at different levels of significance. Second, Transcription Factor Binding Sites (TFBS) enrichment is calculated by a hypergeometric test using transcription factor binding site data coming from various sources [14, 77, 51] at different levels of significance. The GO term and TFBS enrichment analyses are done using Genomica `http://genie.weizmann.ac.il`.

We have compared our algorithm with CC, OPSM, ISA and QUBIC [90, 112, 25, 16, 17]. We used QUBIC software for QUBIC, BicAT software for OPSM, ISA, BIMAX and Expander software for SAMBA with default settings for each algorithm [12, 103, 90].

Prelic's Synthetic Data We applied our algorithm to a synthetic gene expression data set. In the artificial data sets biclusters have been created on the basis of two scenarios (data available at `http://www.tik.ee.ethz.ch/sop/bimax`). In the first scenario, non-overlapping biclusters with increasing noise levels are generated. In the second scenario, biclusters with increasing overlap but without noise are produced. In both scenarios, biclusters with constant expression values and biclusters following an additive model where the expression values varying over the conditions are investigated.

In order to assess the performance of different biclustering algorithms, we used two measures from Prelic et al. [90] and Hochreiter et al. [54], respectively. In Figure 5.2 and Figure 5.3 the performance of BIMAX, ISA, SAMBA, DeBi, OPSM and QUBIC algorithms on the synthetic data is summarized based on Prelic et al. recovery score and Hochreiter et al. consensus score. The set of biclusters generated by these

algorithms are filtered such that the remaining ones have a maximum overlap of 0.25. In the Prelic et. al paper, after the filtering process the largest 10 biclusters are chosen. Since the bicluster number is not known apriori, we have considered all the filtered biclusters. We did not evaluate xMotif and CC algorithms since they have been shown to perform badly in all the scenarios, mostly below 50% of recovery accuracy [90]. The CC and xMotif algorithms produce large biclusters containing genes that are not expressed. ISA and QUBIC give high Prelic et al. recovery score and Hochreiter et al. consensus score in all scenarios. SAMBA has a lower Hochreiter et al. consensus score compared to its Prelic et al. recovery score. The reason is that, Hochreiter et al. consensus score takes into account both gene and condition dimensions and SAMBA is not very accurate in recovering the biclusters in condition dimension. In the absence of noise with an increasing overlap degree, BIMAX has a high performance based on Prelic et al. and Hochreiter et al. scores. However, BIMAX estimates a large number of biclusters upon increasing noise level (Fig. 5.4). In the absence of overlap with increasing noise levels, DeBi is able to identify 99% of implanted biclusters both in additive and constant model. High degree of overlap decreases the performance of DeBi because it considers the overlapping part of the biclusters as a seperate bicluster.

Yeast Compendium We further applied our algorithm to the compendium of gene expression profiles derived from 300 different experimental perturbations of *S. cerevisiae* [58]. We discovered 192 biclusters in the yeast data set containing 2025 genes and 192 conditions. As a binarization level we used the fold change of 1.58 as recommended in the original paper [58].

Figure 5.5 illustrates the proportion of GO term and TFBS enriched biclusters for the six selected biclustering methods (ISA, OPSM, BIMAX, QUBIC, SAMBA and DeBi) at different levels of significance. DeBi performs the second best based on biological validation measures. BIMAX discovers a higher proportion of GO term and TFBS enriched biclusters.

In the analyzed yeast data, conditions are knocked-out genes. Since biclustering discovers subsets of genes and subsets of conditions we can also examine the biological significance of the clustered conditions. Similar to the previous analysis, we measured GO term enrichment of conditions in each discovered biclusters. DeBi is the second best in discovering high percentage of GO term enriched biclusters.

In the discovered biclusters, the enriched gene functions are related to the enriched sample functions. Bicluster 83, genes are enriched in the "conjugation" GO term and conditions are enriched in "regulation of biological quality" GOterm. Moreover, there is an enrichment of the TFBS of STE12, which is known to be involved in cell cycle. Bicluster 50, consists of genes and samples that are enriched in "ribosome biogenesis and assembly" GO term. Bicluster 22, consists of genes and samples that are enriched in "lipid metabolic process" GO term, and additionally genes are enriched with TFBS of HAP1. Bicluster 9, consists of down regulated genes and

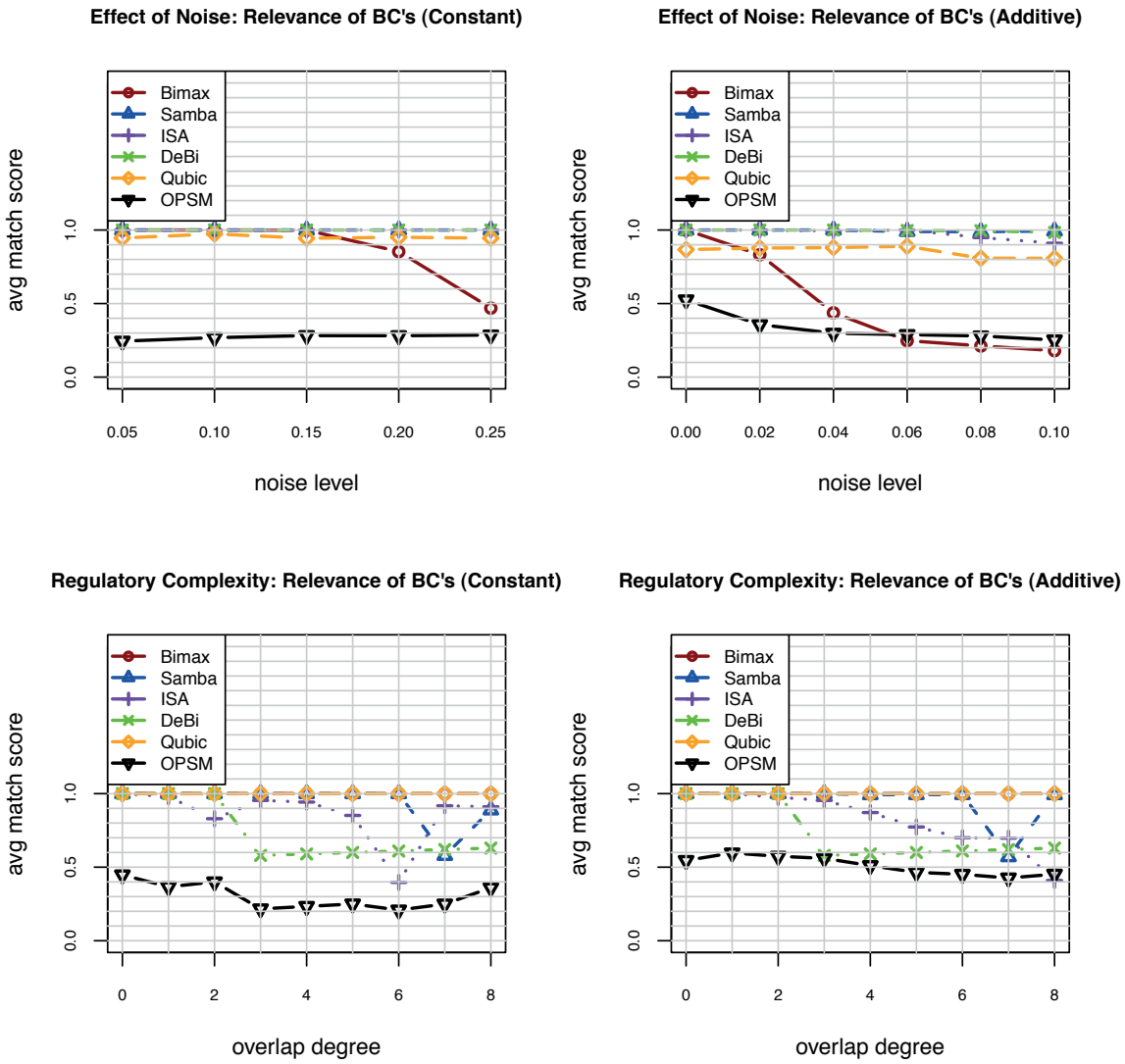


Figure 5.2: Bicluster recovery accuracy score on synthetic data. The synthetic data have been created based on two scenories (a) and (b) with increasing noise level, constant and additive model respectively. (c) and (d) with increasing degree of overlap, constant and additive model respectively.

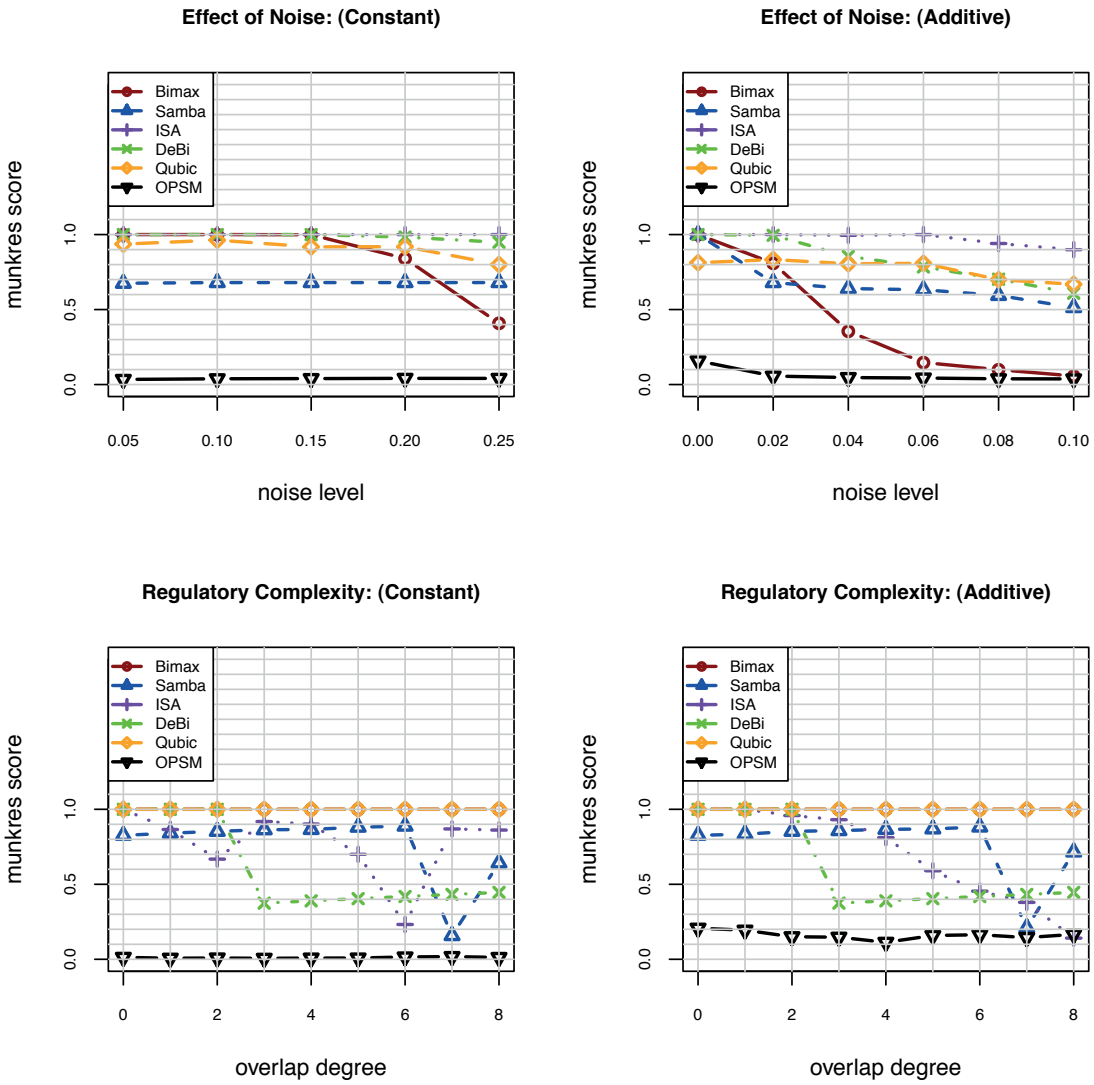


Figure 5.3: Bicluster consensus score on synthetic data. The synthetic data have been created based on two scenories (a) and (b) with increasing noise level, constant and additive model respectively. (c) and (d) with increasing degree of overlap, constant and additive model respectively.

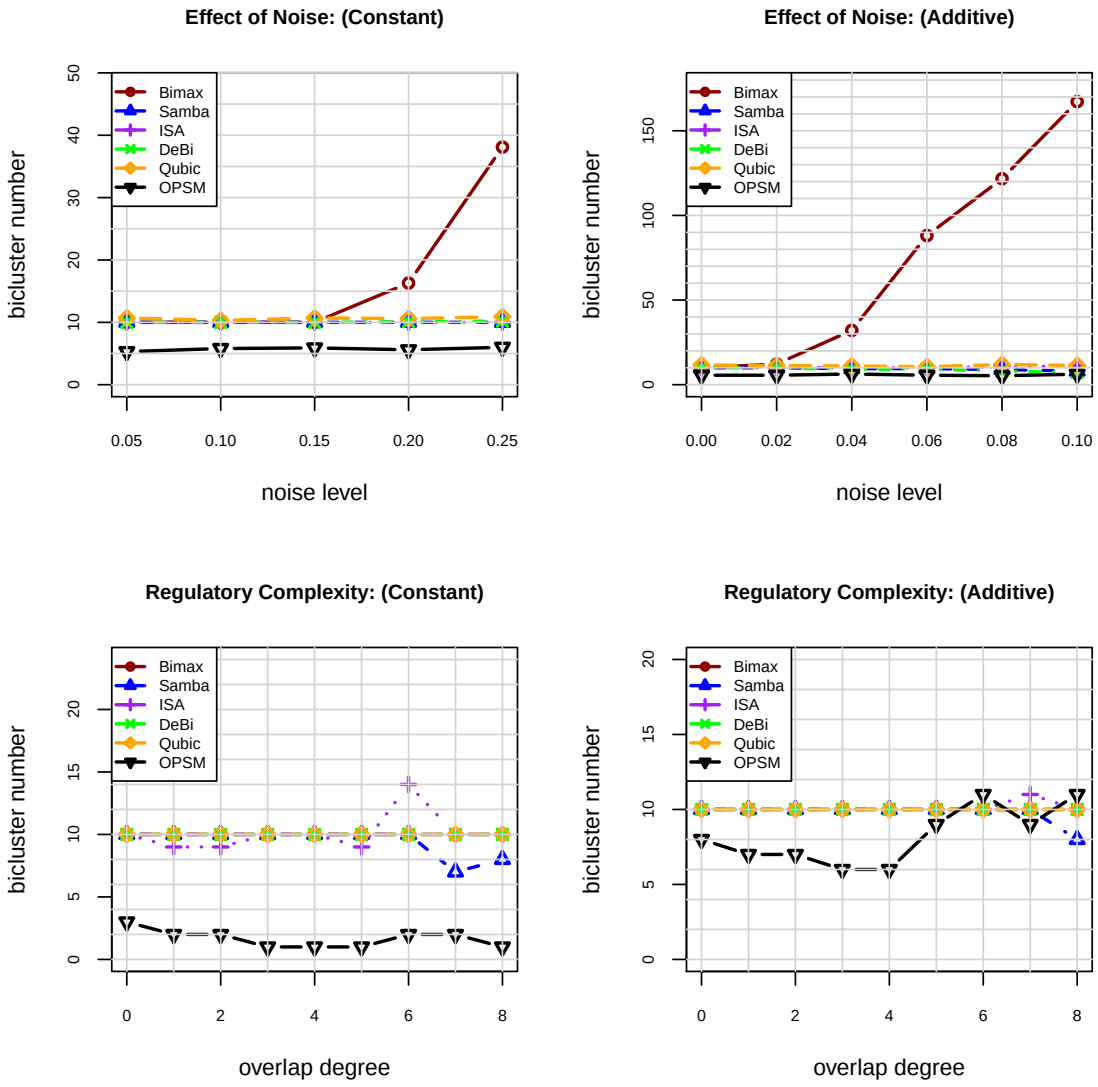


Figure 5.4: Comparision of the estimated number of biclusters with the true number of biclusters.

samples that are enriched in "cell division" GO term, and additionally genes are enriched with TFBS of STE12.

DLBCL Data We also evaluated our DeBi algorithm on "diffuse large B-cell lymphoma" (DLBCL) dataset. DLBCL dataset consists of 661 genes and 180 samples. We applied ISA, OPSM, QUBIC, SAMBA and DeBi algorithms.

Figure 5.5 illustrates the proportion of GO term and TFBS enriched biclusters for the five biclustering methods at different levels of significance. DeBi discovers the highest proportion of GO term and TFBS enriched biclusters. The up regulated bicluster 16 and down regulated bicluster 4 contains the sample classes identified by [55]. Bicluster 16 is enriched with "ribosome" and "cell cycle" GO Term and bicluster 4 is enriched with "cell cycle" and "death" GO Terms. Figure 5.6 shows the protein interaction networks of biclusters 4 and 16. The protein interaction networks are generated using STRING [62].

Human cMap Data We also evaluated our DeBi algorithm on the Connectivity Map v0.2 (cMap) [68]. CMap is a reference collection of gene expression profiles from human cells that have been treated with a variety of drugs comprised of 6100 samples and 22283 genes. Figure 5.5 summarizes the results of DeBi and QUBIC. The proportion of GO term and TFBS enriched biclusters are much more higher in DeBi compared to QUBIC.

The biclusters discovered by DeBi can be used to find drugs with a common mechanism of action and identify new therapeutics. Moreover, we can observe the effect of drugs on different cell lines. Figure 5.7 shows parallel coordinate plots of some of the identified biclusters. In parallel coordinate plots, the profile of the conditions that are included in a bicluster are shown as black, the other conditions as gray (explained in chapter 2, section 3.5). This aids to visualize the expression difference between the conditions in a bicluster compared to the rest of the conditions. Bicluster 6, contains up regulated "heat shock protein binding" genes on the one hand and "heat shock protein inhibitors" such as geldanamycin, alvespimycin, tanespimycin, monorden on the other. Heat shock proteins (Hsps) are overexpressed in a wide range of human cancers and are involved in tumor cell proliferation [28]. Additionally, genes in the bicluster are enriched with "P53 binding site", which is known to target heat shock protein binding genes. Bicluster 11, contains up regulated genes enriched with "cadmium ion binding" GO Term and calcium-binding protein inhibitors, calmidazolium. Bicluster 15, contains up regulated genes enriched with "transcription corepressor activity" GO Term. Cell lines in this bicluster are all breast cancer. Bicluster 14, contains down regulated genes enriched with "streiod hormone signalling" GO Term. Figure 5.8 and Figure 5.9 shows the protein interaction networks of biclusters 6 and 11, 15 and 14.

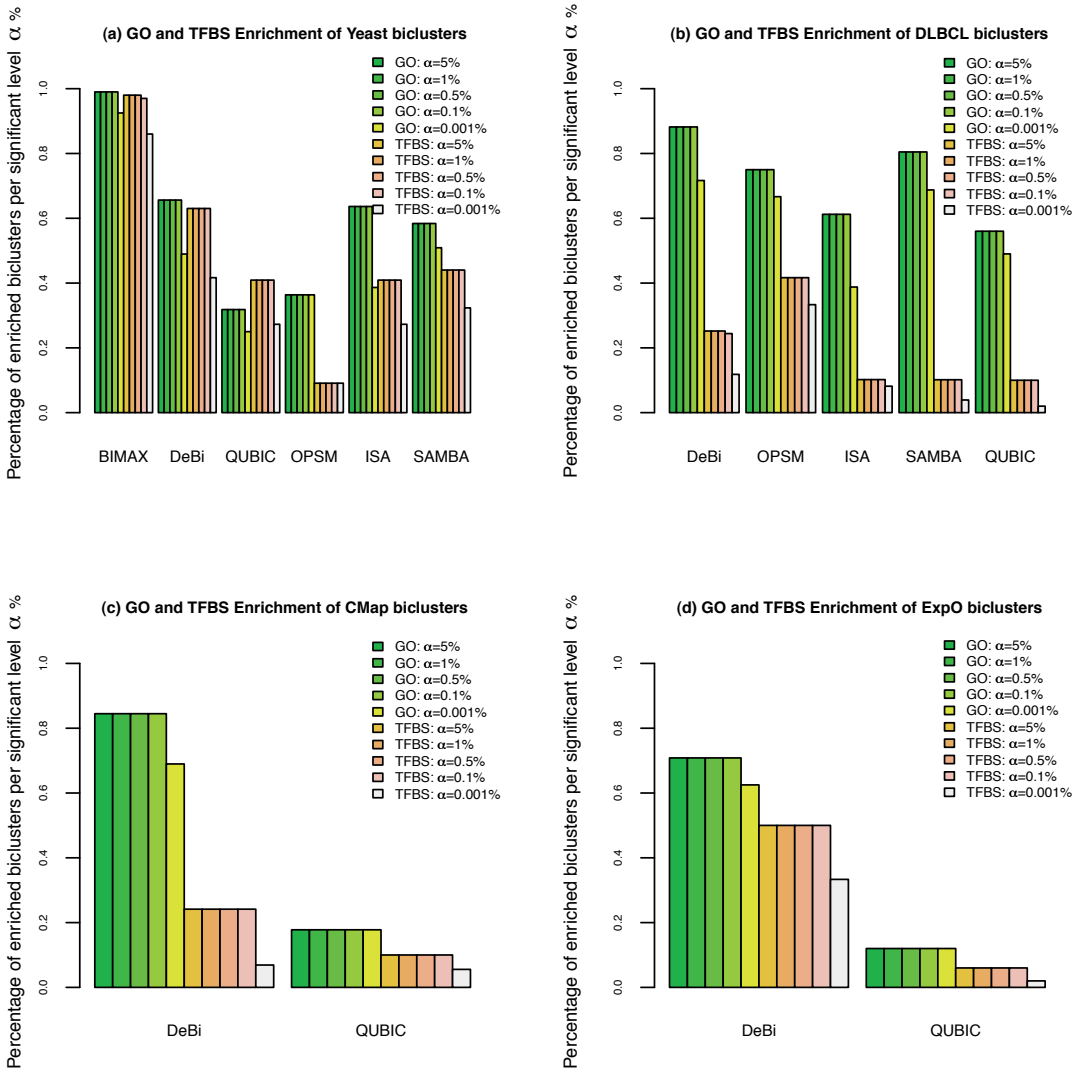
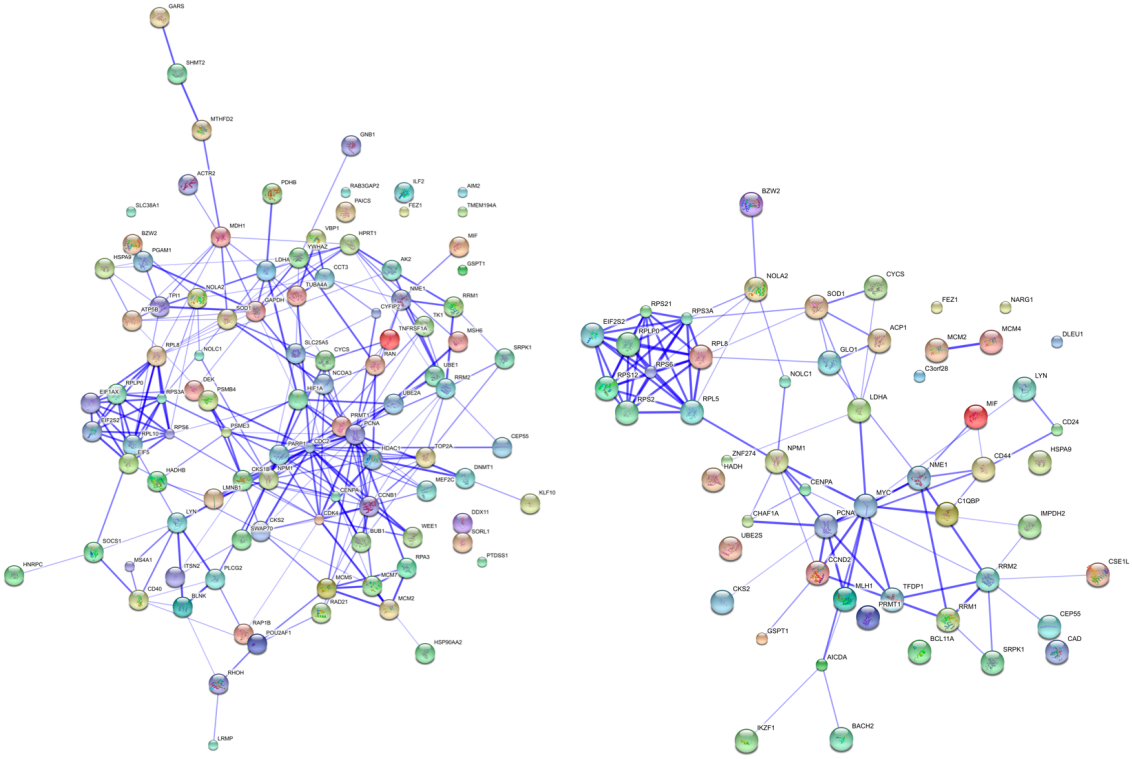


Figure 5.5: GO and TFBS enrichment of yeast, DLBCL, cMap and ExpO biclusters.



(a) Bicluster 4-DLBCL Data **(b)** Bicluster 16-DLBCL Data

Figure 5.6: Protein interaction networks of selected DLBCL biclusters.

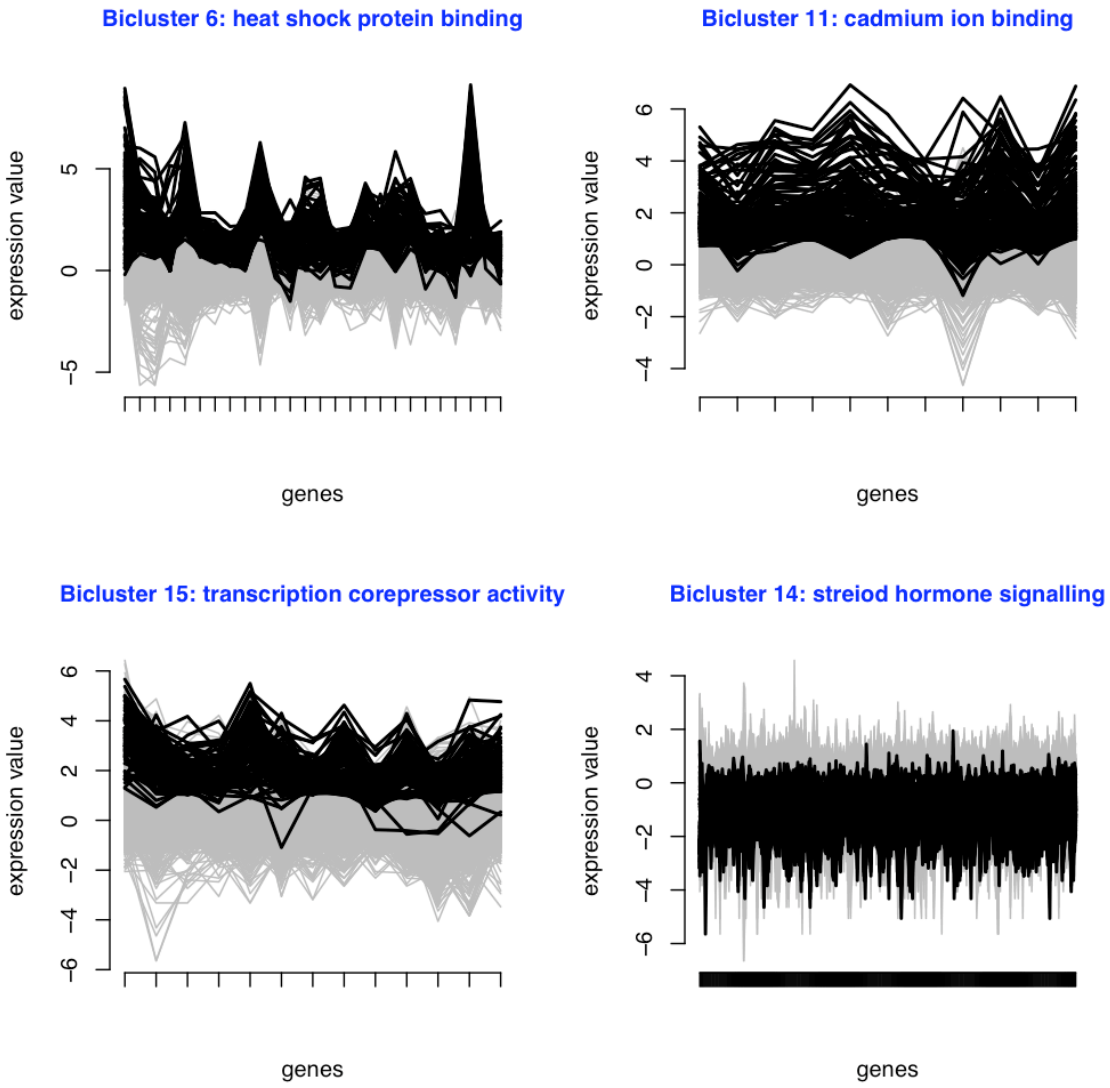
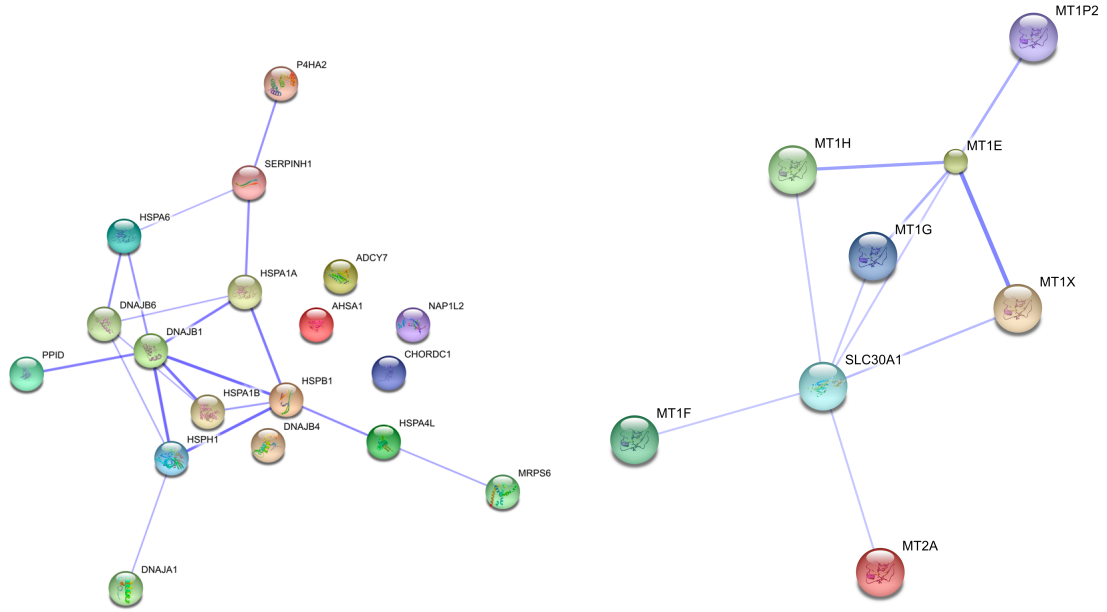
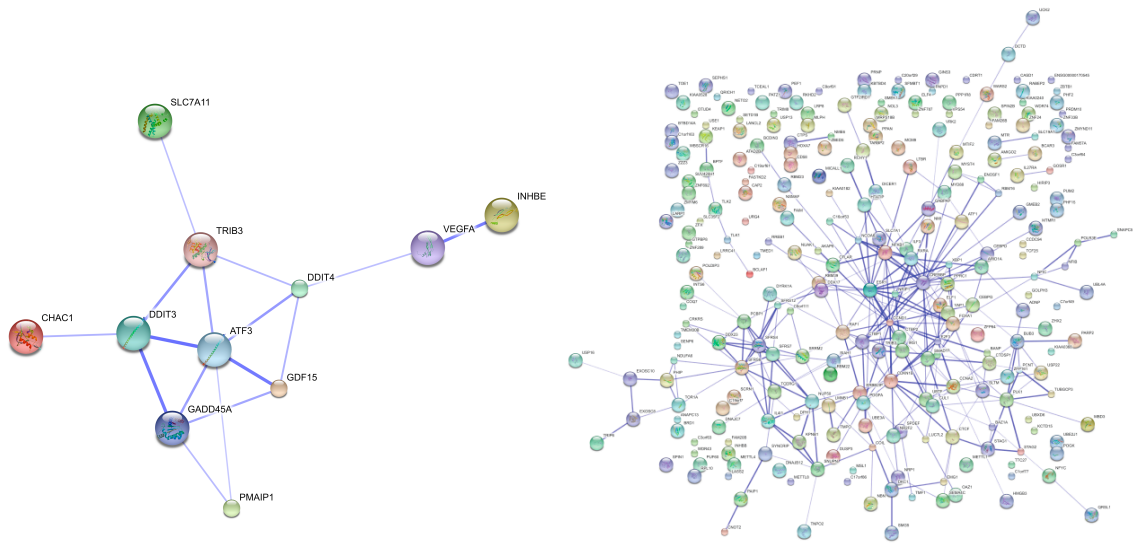


Figure 5.7: Parallel coordinate plots of some of the identified cMap biclusters using the DeBi algorithm. In parallel coordinate plots, the profile of the conditions that are included in a bicluster are shown as black, the other conditions as gray.



(a) Bicluster 6-cMap Data **(b)** Bicluster 11-cMap Data

Figure 5.8: Protein interaction networks of selected cMap biclusters



(a) Bicluster 15-cMap Data **(b)** Bicluster 14-cMap Data

Figure 5.9: Protein interaction networks of selected cMap biclusters

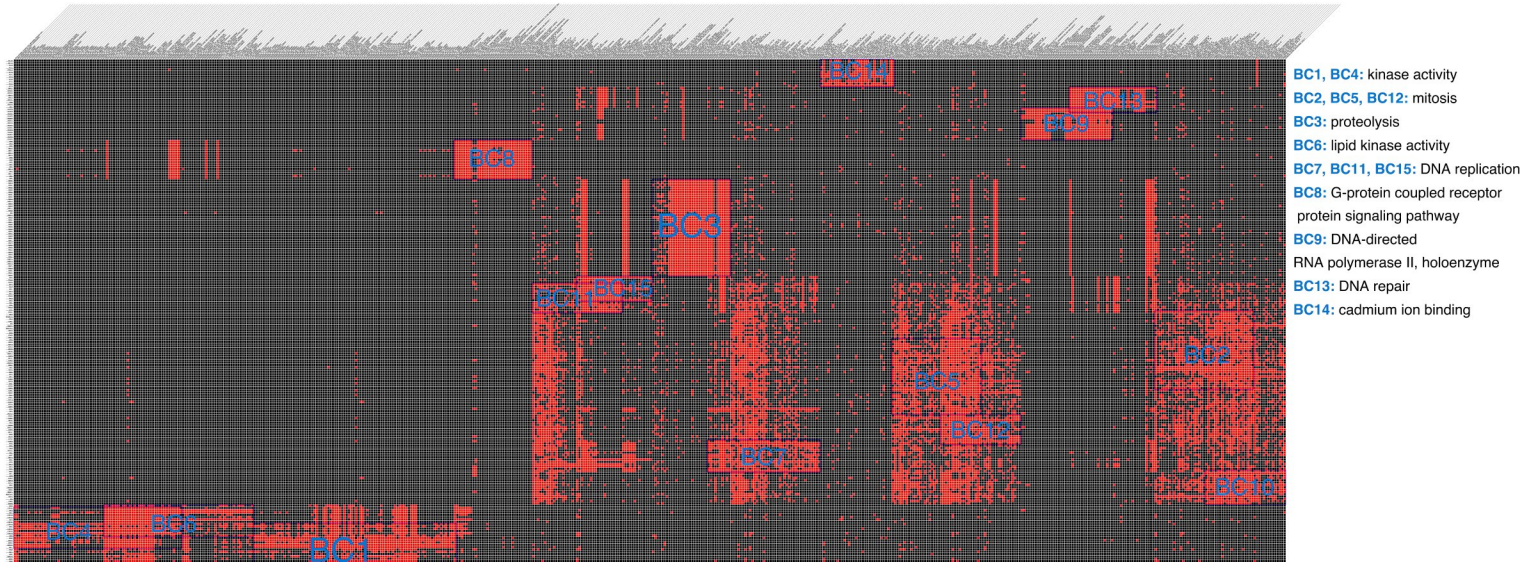


Figure 5.10: The figure illustrates all the biclusters using BiVoc algorithm [46]. BiVoc algorithm rearranges rows and conditions in order to represent the biclusters with the minimum space. The output matrix of BiVoc, may have repeated rows and/or columns from the original matrix. The function of each bicluster is specified based on GO term enrichment.

Human ExpO Data We applied our DeBi algorithm and QUBIC on Expression Project for Oncology(expO) dataset (http://www.intgen.org/expo.cfm). ExpO consists gene expression profiles of 2158 human tumor samples coming from diverse tissues with 40223 transcripts.

Figure 5.5 shows that the proportion of GO term and TFBS enriched biclusters are much more higher in DeBi compared to QUBIC. It illustrates that DeBi performs better than QUBIC in ExpO data. 70% of the DeBi biclusters are enriched with GO Terms with a p-value smaller than 0.05. Moreover biclusters contain tumor samples mostly from similar tissue types. Bicluster 13 contains thyroid tumor samples and genes enriched with "protein-hormone receptor activity". Bicluster 3 contains prostate tumor samples and genes enriched with "tissue kallikrein activity". Bicluster 22 contains mostly pancreas and colon samples and genes enriched with 'pancreatic elastase activity' GO Term.

MSigDB Data Finally, we applied our algorithm on the manually curated gene sets from the Molecular Signature Database (MSigDB) C2 category. The C2 category of MSigDB consists of 3272 gene sets in which 2392 gene sets are chemical and genetic pertubations and 880 gene sets are from various pathway databases. The gene sets naturally define a binary matrix where 1's indicate the affected gene under certain pertubation/pathway. The binary matrix contains 18205 genes and 3272 samples. This analysis aids us to identify the pathways that are affected by chemical and genetic perturbations. It has not been possible to run QUBIC on this dataset while QUBIC requires a certain amount of overlap between genes.

Figure 5.10, illustrates all the biclusters using BiVoc algorithm (explained in chapter 2, section 3.5) [46]. BiVoc algorithm rearranges rows and conditions in order to represent the biclusters with the minimum space. The output matrix of BiVoc, may have repeated rows and/or columns from the original matrix. In Figure 5.10, the function of each bicluster is specified based on GO Term enrichment. Bicluster 3, contains the down-regulated gene set from Alzheimer patients and gene set from proteasome pathway. It is known that there is a significant decrease in proteasome activity in Alzheimer patients [64]. Bicluster 3 also contains the up-regulated gene set from pancreatic cancer patients. In previous studies, high activity of ubiquitin-proteasome pathway in pancreatic cancer cell line was detected [86]. Bicluster 8 contains up-regulated gene set from liver cancer patients and gene set from G-protein activation pathway. Dysfunction of G Protein-Coupled Receptor signaling pathways are involved in certain forms of cancer.

5.4 Running Time

DeBi algorithm is capable of analyzing yeast data(size 6100 x 300) in 6 minutes, ExpO data (size 40223 x 2158) in 12 minutes, MSigDB data (size 18205 x 3272) in

11 minutes, DLBCL data (size 610 x 180) in 11 seconds, cMap data (size 22283 x 6100) in 3 hours 45 minutes. The QUBIC algorithm analyzes cMap data in 2 hours 55 minutes and ExpO data in 3 hours 54 minutes. The running time analysis was done on a 2.13 GHz Intel 2 Dual Core computer with 2GB memory.

Chapter 6

Medicinal Connectivity of Traditional Chinese Medicine (MecoTCM)

In this chapter, our goal is to elucidate the molecular mechanism of Traditional Chinese Medicine (TCM) and to identify new drug candidates from TCM against different human diseases. In section 6.1, we give an introduction to the basic properties of traditional chinese medicine. Then, the goal of the project is summarized (section 6.2). In section 6.3, we describe our two input microarray data. First microarray data is the expression profiles of human cell lines treated with TCM compounds generated by our collaborator Yuhui Hu from Max Delbruck Centrum Berlin. Second microarray data, the so-called Connectivity Map (cMap), is the expression profiles of human cell lines treated with drugs and bioactive small molecules. In section 6.4, we present the pipeline for finding the functional connections for each TCM compound in comparison to (1) 1,309 drugs and bioactive small molecules and (2) TCM compounds within this project. Then, we show some discovered connections between TCM compounds and some known drugs (section 6.5). Finally, our proposed biclustering algorithm DeBi is applied on TCM data combined with cMap data (section 6.6).

6.1 Basic Characteristics of Traditional Chinese Medicine

Traditional Chinese Medicine (TCM) is characterized with complexity and holism in both diagnostic and therapeutic principles. The combination of multiple drugs in complex formulations is thought to maximize the therapeutic efficacy by facilitating synergistic actions and ameliorating or preventing potential adverse effects while at the same time aiming at multiple targets. It makes the functional investigation of TCM difficult to perform with conventional assays. Recent advances in systems biology, particularly the high-throughput functional genomic tools, have paved the way to characterize the multiple gene and pathway targets of TCM for the interpretation of their functional mechanism. In the meantime, it allows us to unravel the novel efficacy as well as possible side-effect of large number of TCM compounds, which will contribute to formulating a more potent therapy for different human diseases [119].

6.2 Goal of the Project

This chemical genomic project wants to contribute to: (1) the elucidation of the molecular mechanism of TCM functions, (2) the identification of new drug candidates from TCM against different human disease, by using the tools of functional genomics and systems biology. Our experiments focus on the bioactive natural substances isolated from TCM materials, including both single compound and mixture of similar compounds (so-called bioactive chemical fraction). We established a research network called Medicinal Connectivity of TCM, MecoTCM, in which the systematic connections can be created among gene expression, disease status, and bioactive chemicals.

The entire approach is based on existing data describing the effect of well-known drugs and the pathways they affect, provided in the so-called Connectivity Map (cMap) (`http://www.broad.mit.edu/cmap`), a collection of over 7,000 genome-wide expression profiles representing 6,100 individual treatment instances with 1,309 bioactive small molecules.

6.3 Data

Connectivity Map Affymetrix Data The Connectivity Map (also known as cMap) is a collection of genome-wide transcriptional expression data from cultured human cells treated with bioactive small molecules. The data consists of four different cell lines HL60 (leukemia), SKMEL5 (melanoma), PC3 (prostat cancer) and MCF7 (breast cancer). High-throughput, cell-based small-molecule screens are performed at different concentrations. As with concentration, the duration of compound treatment might also differ. Every treatment instance is defined relative to a control consisting of cells grown in the same plate and treated with vehicle alone. This approach is taken to minimize the impact of batch-to-batch biological and technical variation [68].

The gene expression profiles are normalized using MAS5 method (see Chapter 2 for MAS5).

TCM Illumina Data Our collaborator Yuhui Hu, from Max Delbrueck Center Berlin, generated Connectivity Map comparable measurements of expression profiles of TCM-derived substances. This gives insight regarding similarities between TCM compounds and existing drugs in terms of affected cellular pathways. The TCM compounds with high similarities thus hold a potential to be alternative to existing drugs and the underlying mechanisms are likely disclosed by the affected genes and pathways.

Genome-wide gene expression profiles are generated, utilizing Illumina BeadChips microarray technology, from the human breast cancer cell lines (MCF7) under the

perturbation of TCM compounds. In total 138 individual treatments are performed on 31 compounds with both biological and technical replicates and the expression profiles are produced for each instance. Some of these 31 compounds are also included in Broad data. We considered them as positive controls to show the reliability of our analysis.

The gene expression profiles are normalized using variance stabilizing normalization (VSN) method (see Chapter 2 for VSN).

6.4 MecoTCM Pipeline

The MecoTCM project aims to find the functional connections for each TCM compound in comparison to (1) 1,309 drugs and bioactive small molecules from the Connectivity Map (cMap) database and (2) TCM compounds within this project. The level of similarities is represented by the respective connectivity scores (c-Scores) (defined in equation 6.3) that are calculated for each compound pair under investigation. Our analysis consists of 3 main steps: compound gene-signature creation, Connectivity Map generation and GO, Pathway and TFBS enrichment analysis (Fig. 6.2).

Compound gene-signature creation Genome-wide gene expression profiles are generated, utilizing Illumina BeadChips microarray technology, from the human breast cancer cell lines (MCF7) under the perturbation of TCM compounds. In comparison to untreated cells, differentially expressed genes are identified, serving as a gene ID of each compound for connectivity score (c-Score) calculation (c-Score defined in equation 6.3). The gene sets those expressions are significantly altered are extracted as a gene signature of the compound, serving as the query counterpart for c-Score and pathway analysis. In the query signature each gene has a sign showing whether it is up regulated or down regulated. Since the genes do not contain any unit, c-Score is not dependent on the platform.

The query signature of each compound is represented both in Illumina and Affymetrix identifiers. The Illumina ids are converted to Affymetrix ids via Refseq ids using lumi R package [36]. First, each Illumina probe sequence is BLASTed [8] against the corresponding Refseq genome. The mapping quality information is used to filter out the bad mappings. The mapped Refseq ids are then converted to Affymetrix Human Genome U133A Array ids using the data with a date stamp from the source of: 2010-March-1.

Connectivity Map generation In the cMap and TCM datasets, the genes on the arrays are rank-ordered according to their normalized expression values. In both datasets, for each treatment instance we have a rank-ordered list of $\sim$22,000 genes. In the second step of the analysis, we assess the similarity of each TCM compound query signature to rank ordered cMap dataset and TCM dataset separately. We use the

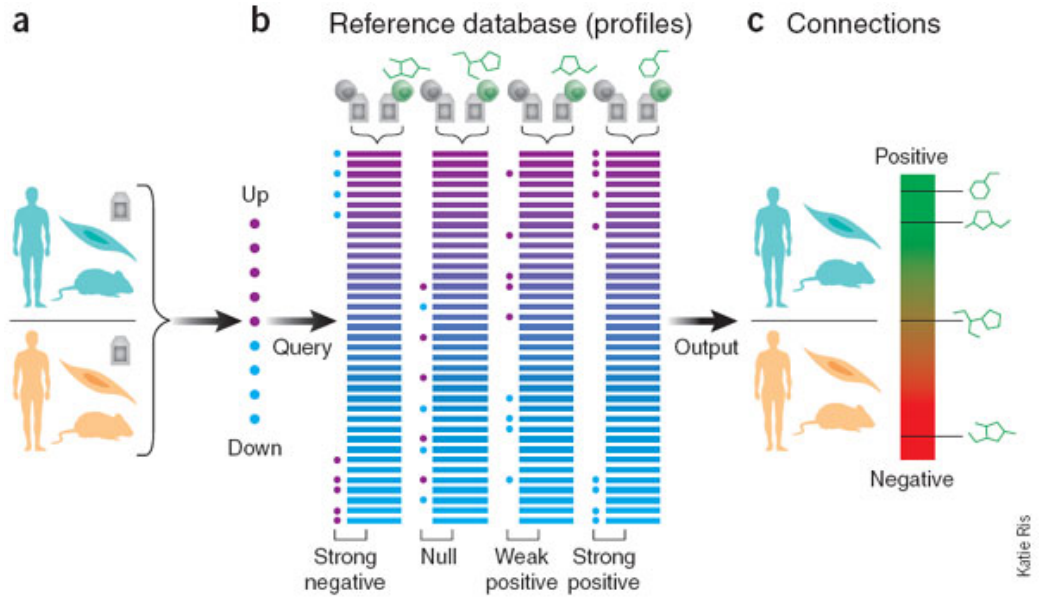


Figure 6.1: Connectivity score calculation. If up-regulated query signature genes tend to appear near the top of the list and down-regulated query genes near the bottom of the list we have positive connectivity score and if vice versa we have negative connectivity score. The c-Score of zero means up and down regulated gene enrichment score is the same. The figure is adapted from [80]

query signatures with Affymetrix ids for cMap data c-Score calculation and Illumina ids for TCM data c-Score calculation (Fig. 6.1). If up-regulated query signature genes tend to appear near the top of the list and down-regulated query genes near the bottom of the list we have positive connectivity score and if vice versa we have negative connectivity score. The c-Score ranges from +1 to -1. The c-Score of zero means up and down regulated gene enrichment score is the same. Finally, the instances are ranked according to their c-Scores. The instances that are at the top of the ranked list are strongly correlated to the query signature, and those at the bottom are strongly anticorrelated [69].

Connectivity Score (c-Score): In order to calculate the c-Score, first enrichment scores for both up and down regulated gene signatures, ks_{up} and ks_{down}, are calculated using Kolmogorov-Smirnov statistics (gene set enrichment method is explained in chapter 2). In our dataset D, we have a rank ordered list of genes for each treatment instance. Let n be the total number of probes in D and $t_{up(down)}$ be the number of probes in query signature. To this end, vector V representing the position $(1 \dots n)$ of the signature in D is constructed. Then, the vector V is ordered in ascending order and the following two values are computed.

$$a = \max_{j=1}^{t} \left[\frac{j}{t} - \frac{V(j)}{n} \right] \qquad b = \max_{j=1}^{t} \left[\frac{V(j)}{n} - \frac{j-1}{t} \right] \tag{6.1}$$

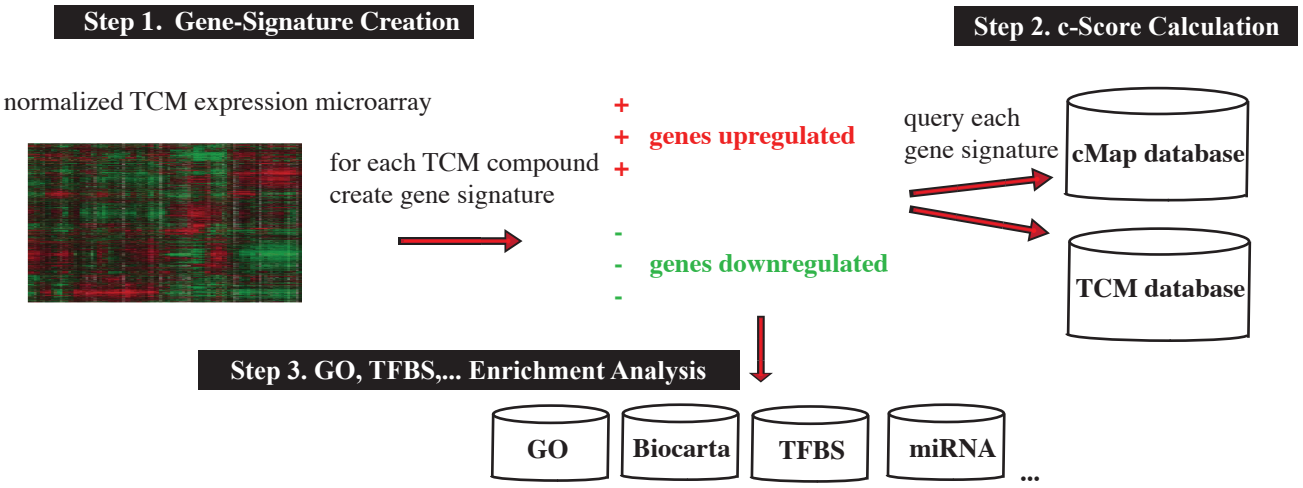


Figure 6.2: MecoTCM pipeline. Our analysis consists of 3 main steps: compound gene-signature creation, c-Scores calculation and gene set enrichment analysis. First, the gene sets those expressions are significantly altered are extracted as a gene signature of the compound. Then the query signatures is used for c-Score calculation. Finally enriched GO terms, pathways and transcription factors are discovered.

and the above a and b values are used to calculate ks_{up} and ks_{down}.

$$ks_{down(up)} = \begin{cases} a & \text{if } a > b \\ -b & \text{if } b > a \end{cases} \tag{6.2}$$

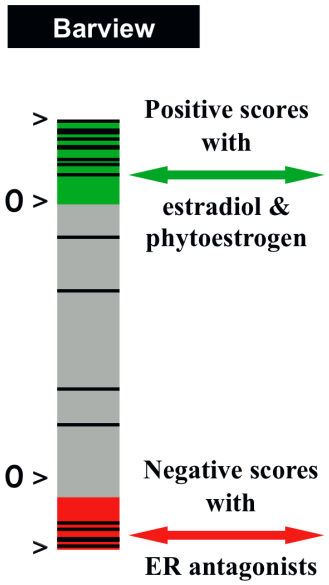
And finally the connectivity score is:

$$CS = \begin{cases} 0 & \text{if } sign(ks_{up}) \neq sign(ks_{down}) \\ ks_{up} - ks_{down} & \text{otherwise} \end{cases} \tag{6.3}$$

GO, Pathway and TFBS Enrichment analysis Further bioinformatics effort in this part aims to reveal the gene regulatory networks underlying the action of TCM efficacy. The essence of enrichment analysis (Gene Ontology, Biocarta pathway, transcription factor targets and more) bases on a cluster of genes that shift their expressions altogether along with the compound perturbation. The analyses are done using the Genomica web server `http://genomica.weizmann.ac.il/`. Genomica discovers statistically significantly enriched GO terms, pathways and transcription factors among the given gene list using a statistical test based on the hypergeometric distribution. The concurrent up- and down-regulated gene sets often fall into the same pathway and are very likely functional relevant, thus providing the information on the molecular role of TCM.

6.5 MecoTCM Results

MecoTCM project discovered many connections between TCM compounds and drugs. The identified connections reveals the common mechanism of action and phsiological processes. Below are some examples of the discovered connections.



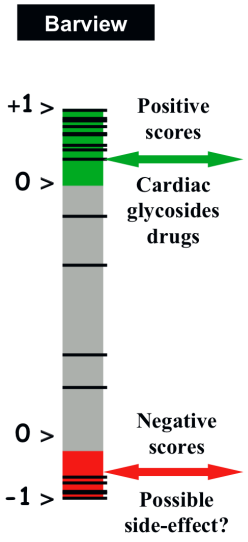
Score	Chemicals	Dose	Cell	Biological function
1	estradiol	100 nM	MCF7	estrogen
0.881	estradiol	10 nM	MCF7	estrogen
0.873	estradiol	10 nM	MCF7	estrogen
0.839	estradiol	10 nM	ssMCF7	estrogen
0.835	alpha-estradiol	10 nM	MCF7	estrogen
0.821	genistein	10 µM	MCF7	phytoestrogen
0.752	estradiol	10 nM	ssMCF7	estrogen
0.716	genistein	10 µM	MCF7	phytoestrogen
0.689	alpha-estradiol	10 nM	MCF7	estrogen
0.683	5182598	25 µM	MCF7	
0.676	genistein	10 µM	MCF7	phytoestrogen
0	alpha-estradiol	10 nM	PC3	
0	estradiol	10 nM	PC3	
0	genistein	10 µM	PC3	
0	fulvestrant	1 µM	PC3	
-0.561	fulvestrant	1 µM	MCF7	ER antagonist
-0.571	LY-294002	10 µM	MCF7	
-0.582	fulvestrant	10 nM	MCF7	ER antagonist
-0.889	fulvestrant	1 µM	MCF7	ER antagonist
-0.985	fulvestrant	1 µM	MCF7	ER antagonist
-1	fulvestrant	10 nM	MCF7	ER antagonist

Figure 6.3: Ginsenoside Re identified as a novel Phytoestrogen. The bar view is constructed from 6100 lines, each representing an individual treatment instance, ordered by their corresponding connectivity scores. The Ginsenoside Re signatures are colored in black. The green lines represent the positive scores, gray lines zero scores and the red lines negative scores. The figure shows the instance name, concentration, cell line, biological function and connectivity score for each of the selected instances.

Ginsenoside Re identified as a novel Phytoestrogen: We derived the query signature of Ginsenoside Re from our Illumina microarray data. MecoTCM yielded high positive connectivity score in instances of estrodiol in MCF7 cells and in instances of genistein, which is a phytoestrogen [79]. The MecoTCM gave high negative connectivity for fulvestrant, a known anti-estrogenic drug [116]. The GO Enrichment of up-regulated genes under the Ginsenoside Re treatment, also supports our findings. The estrogen related GO terms such as "MHC class II protein complex", "sex differentiation", "female gonad development" have significant enrihment. Both GO enrichment and connectivity score analysis suggests Ginsonide Re as a novel Phytoestrogen(Fig. 6.3).

Tanshinone IIA identified as Cardiac glycosides: MecoTCM identified strong connectivity between TCM compound named "Tanshinone IIA" and drugs like helveticoside, digoxin, perhexiline, ouabain in PC3, MCF7 and HL60 cell lines. All these highly connected drugs have cardiac glycoside function. The strong negative

connectivity scores for anti-diabetics may elucidate possible side effects. The pathway analysis revealed significant enrichment for pathway terms like "Hypoxia-Inducible Factor in the Cardiovascular System". Both pathway enrichment and connectivity score analysis suggest a cardiac glycoside effect of Tanshinone IIA (Fig. 6.4)



cScore	Drugs	Cell	Biological Function
1.00	helveticoside	PC3	cardiac glycoside
0.97	helveticoside	PC3	cardiac glycoside
0.96	digoxin	PC3	cardiac glycoside
0.92	perhexiline	MCF7	antianginal
0.89	ouabain	PC3	cardiac glycoside
0.88	prenylamine	MCF7	vasodilator
0.88	primaquine	MCF7	anti-malaria and Pneumocystis pneumonia
0.88	mefloquine	MCF7	anti-malaria
0.84	monobenzone	HL60	depigmentation
0.84	phenoxybenzamine	MCF7	α-adrenoceptors antagonists against hypertension
0.84	ouabain	MCF7	cardiac glycoside
0.84	alexidine	MCF7	disinfectant
0.83	strophanthidin	MCF7	cardenolide
0.83	lanatoside C	PC3	cardiac glycoside
0.82	anisomycin	PC3	antibiotic
0.82	digoxigenin	PC3	cardenolide
0.82	lanatoside C	PC3	cardiac glycoside
0.80	digoxigenin	MCF7	cardenolide
0.80	cicloheximide	PC3	antibiotic
0.79	tanespimycin	MCF7	antibiotic
0.79	nifuroxazide	MCF7	antibiotic
0.78	emetine	PC3	anti-protozoal
-0.91	methylergometrine	PC3	vasoconstriction
-0.92	fluorometholone	PC3	corticosteroid
-0.92	glibenclamide	PC3	antidiabetic
-0.97	dydrogesterone	MCF7	progestogen hormone
-0.97	pioglitazone	PC3	antidiabetic
-0.99	milrinone	PC3	relaxing left ventricle
-1.00	pioglitazone	PC3	antidiabetic

Figure 6.4: Tanshinone IIA identified as Cardiac glycosides. The bar view is constructed from 6100 lines, each representing an individual treatment instance, ordered by their corresponding connectivity scores. The Cardiac glycosides signatures are colored in black. The green lines represent the positive scores, gray lines zero scores and the red lines negative scores. The figure shows the instance name, concentration, cell line, biological function and connectivity score for each of the selected instances.

6.6 Biclustering Results

In addition to connectivity analysis, biclustering can be applied to elucidate the molecular mechanism of TCM functions. In this regard, we applied our DeBi algorithm on two datasets, seperate TCM Illumina data and combined TCM and cMap data. Biclustering the TCM data gives insight into the connections between TCM compounds. Furthermore, biclustering application on combined TCM-cMap data reveals the connections between TCM compounds and drugs.

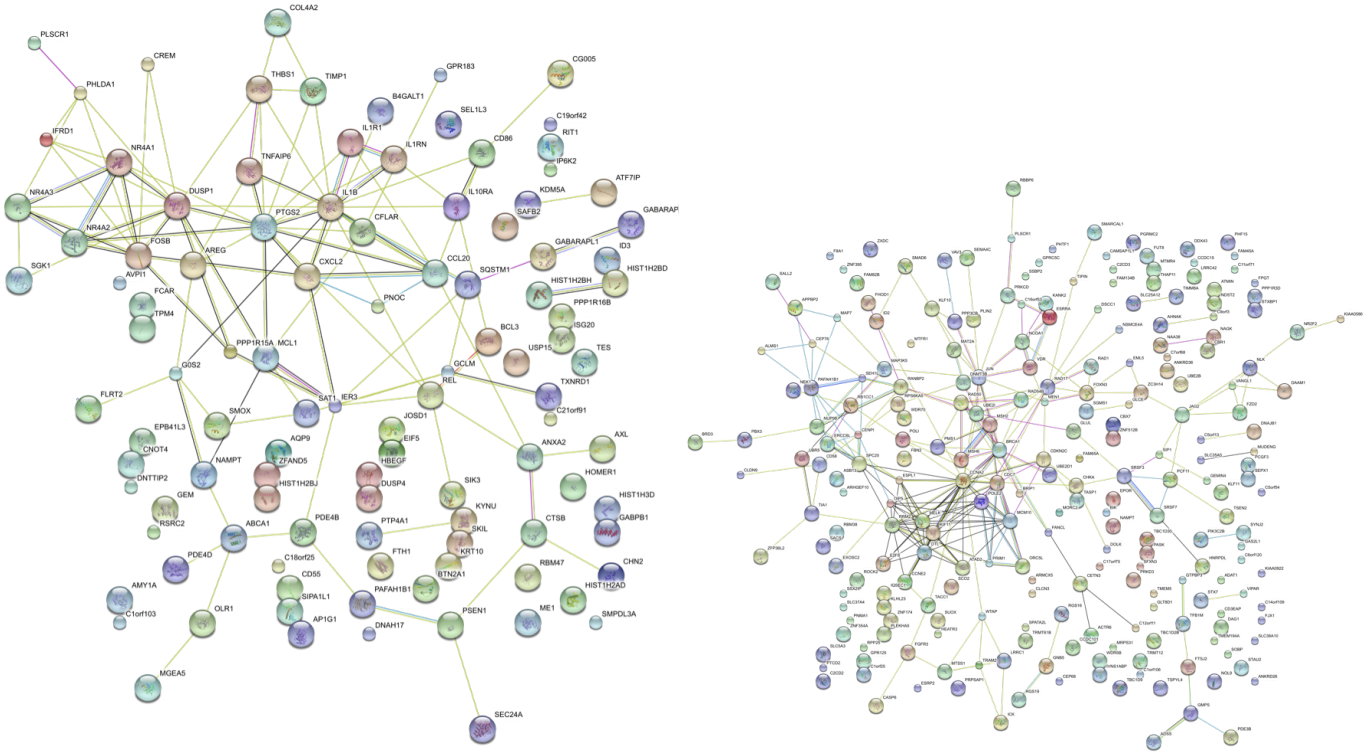
Biclustering internal TCM data: 95% of the DeBi biclusters are enriched with GO Terms with a p-value smaller than 0.05. In the discovered biclusters, the enriched gene functions are related to the TCM/drug functions. Remember that some of the compounds in TCM data are also included in Broad data. We considered them as positive controls to show the reliability of our analysis. Bicluster 22 contains genes that are enriched with "mysosin binding" GO Term. The drugs in this bicluster are staurosporine and dexamethasone and they are also associated to myosin binding activity [81, 29, 110]. Bicluster 17 contains genes that are enriched with "copper ion binding" GO Term. The compounds in this bicluster are genistein and estrogen related TCM compounds. Copper is closely related to the metabolism of the estrogen hormone [96].

Biclustering on TCM Illumina data combined with CMap data: Combining microarray data sets coming from different technologies is one of the major challenges in microarray field. There has been several studies to remove the systematical bias arising from differences of the microarray technologies. One approach focuses on comparing significantly expressed genes coming from each data set [26, 92]. Another approach focuses on integration methods for different microarray platforms [13]. Affymetrix and Illumina platforms yield highly comparable data, especially for genes predicted to be differentially expressed [13].

We applied our biclustering algorithm (see chapter 5) on TCM Illumina data combined with CMap data to identify the connections between compounds and drugs. TCM data is produced by Illumina technology whereas cMap data is produced by Affymetrix technology. Each technology has different probe annotations. The Illumina ids are converted to Affymetrix ids in the same way as we did in MecoTCM pipeline, "Compound gene-signature creation" step. If an Illumina id maps to multiple Affymetrix ids, than both Affymetrix ids are included in the combined data.

Since our algorithm is applied on binary data, we did not correct for platform effect. Our biclustering algorithm is capable of analyzing gene expression data coming from different labs or platforms.

87% of the DeBi biclusters are enriched with GO Terms with a p-value smaller than 0.05. Figure 6.5 shows the protein interaction networks of some of the selected biclusters. The protein interaction networks are generated using STRING [62]. As it is seen in the figures the protein interaction networks are highly connected. Bicluster 94 contains up regulated genes that are enriched with "steroid dehydrogenase activity" GO Term. This bicluster contains TCM compounds such as "Tanshinone IIA", "Cryptotanshinone" and drugs such as suloctidil, 15-delta prostaglandin and securinine. In connectivity analysis, the compound "Tanshinone IIA" also has a high connectivity score to suloctidil, 15-delta prostaglandin and securinine. The biclustering results support the connectivity analysis. The bicluster 97, contains down regulated genes that are enriched with "magnesium ion transporter activity" GO term. The drugs and compounds in this bicluster are Ly-294002, Staurosporine, Genistein and estrogen related TCM compound "Ginsenoside Rb1".



(a) Bicluster 94- Up Regulated **(b)** Bicluster 97- Down Regulated

Figure 6.5: Protein interaction networks of selected cMap-TCM data biclusters

Chapter 7

Summary

In this final chapter, we summarize the contributions presented in this thesis.

Biclustering of Large Scale Data

The analysis of massive high throughput data via clustering algorithms is very important for elucidating gene functions in biological systems. However, traditional clustering methods have several drawbacks. Biclustering overcomes these limitations by grouping genes and samples simultaneously. It discovers subsets of genes that are co-expressed in certain samples. Recent studies showed that biclustering has a great potential in detecting marker genes that are associated with certain tissues or diseases. Several biclustering algorithms have been proposed. However, it is still a challenge to find biclusters that are significant based on biological validation measures. Besides that, there is a need for a biclustering algorithm that is capable of analyzing very large datasets in reasonable time.

We have proposed a novel fast biclustering algorithm especially for analyzing large data sets. Our algorithm aims to find biclusters where each gene in a bicluster should be highly or lowly expressed over all the bicluster samples compared to the rest of the samples. Unlike other algorithms, it is not required to define the number of biclusters a priori. We have compared our method with other biclustering algorithms using synthetic data and biological data. It is shown that the DeBi algorithm provides biologically significant biclusters using GO term and TFBS enrichment. We have also presented the computational efficiency of our algorithm. It is a useful and powerful tool in analyzing large data sets.

In spite of efforts by many authors, comparing the performance of biclustering algorithms is still a challenge. Smaller biclusters have a higher chance to yield a coherent GO annotation, while larger biclusters would, of course, be more interesting to observe. Our α threshold influences this behavior. The optimized α threshold yields smaller values for larger numbers of samples which limits the number of genes that get accepted into a bicluster.

The binarization of the input data in order to obtain a boolean matrix is another key decision in our approach. In this we go along with many other authors and we think that it helps in applying biclustering to gene expression data coming from different

labs or platforms. The data integration promotes the discovery of subtle changes with increased sensitivity and reliability. The hope is that our method will further contribute to establishing biclustering as a general purpose tool for data analysis in functional genomics.

The DeBi algorithm algorithm is freely available at `http://www.molgen.mpg.de/~serin/debi/main.html`.

MecoTCM project

The application of high throughput functional genomic tools will greatly facilitate the elucidation of the functional mechanisms of TCM in a systematic way. Using connectivity analysis and biclustering we can discover the mechanisms of action and identify new therapeutic uses for TCM.

This chemical genomic project wants to contribute to: (1) the elucidation of the molecular mechanism of TCM functions, (2) the identification of new drug candidates from TCM against different human disease, by using the tools of functional genomics and systems biology. Our experiments focus on the bioactive natural substances isolated from TCM materials, including both single compound (purity above 98) and mixture of similar compounds (so-called bioactive chemical fraction). We established a research network called Medicinal Connectivity of TCM, MecoTCM, in which the systematic connections is created among gene expression, disease status, and bioactive chemicals.

Using our approach, we discovered biologically significant findings about TCM theory, for example on herb Ginseng. We further validated our results using Chip-Seq experiments. We plan to further upgrade this dataset by comparisons to existing tumor gene expression profiles, in an attempt to match tumor profiles and TCM treatment profiles in terms of the affected pathways.

MecoTCM project results are availble at `http://www.molgen.mpg.de/~serin/mecomap/home.html`.

Bibliography

[1] Affymetrix. Statistical algorithms reference guide,technical report. `http://media.affymetrix.com/support/technical/technotes/statistical_reference_guide.pdf`.

[2] Affymetrix. *GeneChip Expression Analysis*, 2000.

[3] R. C. Agarwal, C. C. Aggarwal, and V. V. V. Prasad. A tree projection algorithm for generation of frequent itemsets. *Journal of Parallel and Distributed Computing*, 61:350–371, 2000.

[4] R. Agrawal and R. Srikant. Fast Algorithms for Mining Association Rules in Large Databases. In *VLDB '94: Proceedings of the 20th International Conference on Very Large Data Bases*, pages 487–499, San Francisco, CA, USA, 1994. Morgan Kaufmann Publishers Inc.

[5] B. Alberts. *Molecular biology of the cell: Reference edition*. Number v. 1 in Molecular Biology of the Cell: Reference Edition. Garland Science, 2008.

[6] A. A. Alizadeh, M. B. Eisen, R. E. Davis, C. Ma, I. S. Lossos, A. Rosenwald, J. C. Boldrick, H. Sabet, T. Tran, X. Yu, J. I. Powell, L. Yang, G. E. Marti, T. Moore, J. Hudson, L. Lu, D. B. Lewis, R. Tibshirani, G. Sherlock, W. C. Chan, T. C. Greiner, D. D. Weisenburger, J. O. Armitage, R. Warnke, R. Levy, W. Wilson, M. R. Grever, J. C. Byrd, D. Botstein, P. O. Brown, and L. M. Staudt. Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling. *Nature*, 403(6769):503–511, Feb. 2000.

[7] U. Alon, N. Barkai, D. A. Notterman, K. Gish, S. Ybarra, D. Mack, and A. J. Levine. Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Proceedings of the National Academy of Sciences*, 96(12):6745–6750, 1999.

[8] S. F. Altschul, T. L. Madden, A. A. Schffer, J. Zhang, Z. Zhang, W. Miller, and D. J. Lipman. Gapped blast and psi-blast: a new generation of protein database search programs. *Nucleic Acids Research*, 25(17):3389–3402, 1997.

[9] B. Andreopoulos, A. An, X. Wang, and M. Schroeder. A roadmap of clustering algorithms: finding a match for a biomedical application. *Brief Bioinformatics*, 10(3):297–314, Dec 2008.

[10] M. Ashburner, C. A. Ball, J. A. Blake, D. Botstein, H. Butler, J. M. Cherry, A. P. Davis, K. Dolinski, S. S. Dwight, J. T. Eppig, M. A. Harris, D. P. Hill, L. Issel-Tarver, A. Kasarskis, S. Lewis, J. C. Matese, J. E. Richardson, M. Ringwald, G. M. Rubin, and G. Sherlock. Gene Ontology: tool for the unification of biology. *Nature Genetics*, 25(1):25–29, May 2000.

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

[12] S. Barkow, S. Bleuler, A. Prelic, P. Zimmermann, and E. Zitzler. Bicat: a biclustering analysis toolbox. *Bioinformatics*, 22(10):1282–1283, May 2006.

[13] M. Barnes. Experimental comparison and cross-validation of the affymetrix and illumina gene expression analysis platforms. *Nucleic Acids Res*, 33(18):5914–5923, Oct 2005.

[14] A. D. Basehoar, S. J. Zanton, and B. F. Pugh. Identification and distinct regulation of yeast tata box-containing genes. *Cell*, 116(5):699–709, Mar 2004.

[15] R. J. Bayardo, Jr. Efficiently mining long patterns from databases. In *Proceedings of the 1998 ACM SIGMOD international conference on Management of data*, SIGMOD '98, pages 85–93, New York, NY, USA, 1998. ACM.

[16] A. Ben-Dor, B. Chor, R. Karp, and Z. Yakhini. Discovering local structure in gene expression data: the order-preserving submatrix problem. *J Comput Biol*, 10(3-4):373–384, 2003.

[17] S. Bergmann, J. Ihmels, and N. Barkai. Iterative signature algorithm for the analysis of large-scale gene expression data. *Phys Rev E Stat Nonlin Soft Matter Phys*, 67(3 Pt 1):031902, Mar 2003.

[18] B. Bolstad, R. Irizarry, M. strand, and T. Speed. A comparison of normalization methods for high density oligonucleotide array data based on variance and bias. *Bioinformatics*, 19(2):185–193, 2003.

[19] B. M. Bolstad. Low-level analysis of high-density oligonucleotide array data: background, normalization and summarization. *Analysis*, 2004.

[20] A. Brazma, H. Parkinson, U. Sarkans, M. Shojatalab, J. Vilo, N. Abeygunawardena, E. Holloway, M. Kapushesky, P. Kemmeren, G. G. Lara, A. Oezcimen, P. Rocca-Serra, and S.-A. Sansone. Arrayexpressa public repository for microarray gene expression data at the ebi. *Nucleic Acids Research*, 31(1):68–71, 2003.

[21] G. Brock, V. Pihur, S. Datta, and S. Datta. clvalid , an r package for cluster validation, 2008.

[22] P. O. Brown and D. Botstein. Exploring the new world of the genome with dna microarrays. *Nature Genetics*, 21(1 Suppl):33–7, 1999.

[23] A. Brzma, I. Jonassen, J. Vilo, and E. Ukkonen. Predicting gene regulatory elements in silico on a genomic scale. *Genome Research*, 8(11):1202–1215, 1998.

[24] D. Burdick, M. Calimlim, and J. Gehrke. Mafia: a maximal frequent itemset algorithm for transactional databases. In *Proc. 17th Int Data Engineering Conf*, pages 443–452, 2001.

[25] Y. Cheng and G. M. Church. Biclustering of expression data. *Proc Int Conf Intell Syst Mol Biol*, 8:93–103, 2000.

[26] J. K. Choi, U. Yu, S. Kim, and O. J. Yoo. Combining multiple microarray studies and modeling interstudy variation. *Bioinformatics*, 19(suppl 1):i84–i90, 2003.

[27] G. A. Churchill. Fundamentals of experimental design for cDNA microarrays. *Nature Genetics*, 32 Suppl:490–495, Dec. 2002.

[28] D. R. Ciocca and S. K. Calderwood. Heat shock proteins in cancer: diagnostic, prognostic, predictive, and treatment implications. *Cell Stress Chaperones*, 10(2):86–103, 2005.

[29] B. A. Clarke, D. Drujan, M. S. Willis, L. O. Murphy, R. A. Corpina, E. Burova, S. V. Rakhilin, T. N. Stitt, C. Patterson, E. Latres, and D. J. Glass. The e3 ligase murf1 degrades myosin heavy chain protein in dexamethasone-treated skeletal muscle. *Cell Metabolism*, 6(5):376 – 385, 2007.

[30] F. H. C. Crick. On protein synthesis. *The Symposia of the Society for Experimental Biology*, 12:138–163, 1958.

[31] S. Datta and S. Datta. Comparisons and validation of statistical clustering techniques for microarray gene expression data. *Bioinformatics*, 19(4):459–466, Mar. 2003.

[32] D. Dembl and P. Kastner. Fuzzy c-means method for clustering microarray data. *Bioinformatics*, 19(8):973–980, 2003.

[33] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38, 1977.

[34] J. DeRisi, L. Penland, P. O. Brown, M. L. Bittner, P. S. Meltzer, M. Ray, Y. Chen, Y. A. Su, and J. M. Trent. Use of a cdna microarray to analyse gene expression patterns in human cancer. *Nature Genetics*, 14(4):457–460, 1996.

[35] P. Du, W. Kibbe, and S. Lin. nuid: a universal naming scheme of oligonucleotides for illumina, affymetrix, and other microarrays. *Biology Direct*, 2:1–7, 2007. 10.1186/1745-6150-2-16.

[36] P. Du, W. A. Kibbe, and S. M. Lin. lumi: a pipeline for processing illumina microarray. *Bioinformatics*, 24(13):1547–1548, 2008.

[37] S. Dudoit and J. Fridlyand. A prediction-based resampling method for estimating the number of clusters in a dataset. *Genome Biology*, 3(7):research0036.1–research0036.21, 2002.

[38] R. Edgar, M. Domrachev, and A. E. Lash. Gene expression omnibus: Ncbi gene expression and hybridization array data repository. *Nucleic Acids Research*, 30(1):207–210, 2002.

[39] B. Efron, R. Tibshirani, J. D. Storey, and V. Tusher. Empirical bayes analysis of a microarray experiment. *Journal of the American Statistical Association*, 96(456):1151–1160, 2001.

[40] M. B. Eisen, P. T. Spellman, P. O. Brown, and D. Botstein. Cluster analysis and display of genome-wide expression patterns. *Proc Natl Acad Sci USA*, 95(25):14863–8, Dec 1998.

[41] A. Gasch and M. Eisen. Exploring the conditional coregulation of yeast gene expression through fuzzy k-means clustering. *Genome Biology*, 3(11), 2002.

[42] M. B. Gerstein, C. Bruce, J. S. Rozowsky, D. Zheng, J. Du, J. O. Korbel, O. Emanuelsson, Z. D. Zhang, S. Weissman, and M. Snyder. What is a gene, post-encode? history and updated definition. *Genome Research*, 17(6):669–681, 2007.

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

[44] H. Gmuender. Perspectives and challenges for dna microarrays in drug discovery and development. *BioTechniques*, 32(1):152–4, 156, 158, Jan 2002.

[45] K. Gouda and M. J. Zaki. Efficiently mining maximal frequent itemsets. In *Proceedings of the 2001 IEEE International Conference on Data Mining*, ICDM '01, pages 163–170, Washington, DC, USA, 2001. IEEE Computer Society.

[46] G. A. Grothaus, A. Mufti, and T. M. Murali. Automatic layout and visualization of biclusters. *Algorithms for molecular biology : AMB*, 1:15, Jan 2006.

[47] J. Gu and J. Liu. Bayesian biclustering of gene expression data. *BMC Genomics*, 9(Suppl 1):S4, 2008.

[48] J. Han and M. Kamber. *Data Mining: Concepts and Techniques (The Morgan Kaufmann Series in Data Management Systems)*. Morgan Kaufmann, 1st edition, Sept. 2000.

[49] J. Han, J. Pei, and Y. Yin. Mining frequent patterns without candidate generation. *SIGMOD Rec.*, 29:1–12, May 2000.

[50] J. Handl, J. Knowles, and D. B. Kell. Computational cluster validation in post-genomic data analysis. *Bioinformatics*, 21(15):3201–3212, Aug. 2005.

[51] C. T. Harbison, D. B. Gordon, T. I. Lee, N. J. Rinaldi, K. D. MacIsaac, T. W. Danford, N. M. Hannett, J.-B. Tagne, D. B. Reynolds, J. Yoo, E. G. Jennings, J. Zeitlinger, D. K. Pokholok, M. Kellis, P. A. Rolfe, K. T. Takusagawa, E. S. Lander, D. K. Gifford, E. Fraenkel, and R. A. Young. Transcriptional regulatory code of a eukaryotic genome. *Nature*, 431(7004):99–104, Sep 2004.

[52] J. A. Hartigan. Direct Clustering of a Data Matrix. *Journal of the American Statistical Association*, 67(337):123–129, 1972.

[53] J. A. Hartigan and M. A. Wong. Algorithm as 136: A k-means clustering algorithm. *Applied Statistics*, 28(1):100–108, 1979.

[54] S. Hochreiter, U. Bodenhofer, M. Heusel, A. Mayr, A. Mitterecker, A. Kasim, T. Khamiakova, S. V. Sanden, D. Lin, W. Talloen, L. Bijnens, H. W. H. Göhlmann, Z. Shkedy, and D.-A. Clevert. Fabia: factor analysis for bicluster acquisition. *Bioinformatics*, 26(12):1520–7, Jun 2010.

[55] Y. Hoshida, J.-P. Brunet, P. Tamayo, T. R. Golub, and J. P. Mesirov. Subclass mapping: Identifying common subtypes in independent disease data sets. *PLoS ONE*, 2(11):e1195, 11 2007.

[56] W. Huber, A. von Heydebreck, H. Sueltmann, A. Poustka, and M. Vingron. Parameter estimation for the calibration and variance stabilization of microarray data. *Statistical applications in genetics and molecular biology*, 2:Article3, Jan 2003.

[57] J. D. Hughes, P. W. Estep, S. Tavazoie, and G. M. Church. Computational identification of cis-regulatory elements associated with groups of functionally related genes in saccharomyces cerevisiae. *J. MOL. BIOL*, 296:1205–1214, 2000.

[58] T. R. Hughes, M. J. Marton, A. R. Jones, C. J. Roberts, R. Stoughton, C. D. Armour, H. A. Bennett, E. Coffey, H. Dai, Y. D. He, M. J. Kidd, A. M. King, M. R. Meyer, D. Slade, P. Y. Lum, S. B. Stepaniants, D. D. Shoemaker, D. Gachotte, K. Chakraburtty, J. Simon, M. Bard, and S. H. Friend. Functional discovery via a compendium of expression profiles. *Cell*, 102(1):109–126, Jul 2000.

[59] R. A. Irizarry, B. M. Bolstad, F. Collin, L. M. Cope, B. Hobbs, and T. P. Speed. Summaries of Affymetrix GeneChip probe level data. *Nucleic acids research*, 31(4):e15+, Feb. 2003.

[60] R. A. Irizarry, B. Hobbs, F. Collin, Y. D. BeazerBarclay, K. J. Antonellis, U. Scherf, and T. P. Speed. Exploration, normalization, and summaries of high density oligonucleotide array probe level data. *Biostatistics*, 4(2):249–264, 2003.

[61] A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering: a review. *ACM Comput. Surv.*, 31(3):264–323, Sept. 1999.

[62] L. J. Jensen, M. Kuhn, M. Stark, S. Chaffron, C. Creevey, J. Muller, T. Doerks, P. Julien, A. Roth, M. Simonovic, P. Bork, and C. von Mering. String 8–a global view on proteins and their functional interactions in 630 organisms. *Nucleic Acids Res*, 37(Database issue):D412–6, Jan 2009.

[63] M. Kanehisa and S. Goto. KEGG: Kyoto Encyclopedia of Genes and Genomes. *Nucleic Acids Research*, 28(1):27–30, Jan. 2000.

[64] J. N. Keller, K. B. Hanni, and W. R. Markesbery. Impaired proteasome function in alzheimer's disease. *J Neurochem*, 75(1):436–9, Jul 2000.

[65] S.-Y. Kim and D. Volsky. Page: Parametric analysis of gene set enrichment. *BMC Bioinformatics*, 6(1):144, 2005.

[66] Y. Kluger, R. Basri, J. T. Chang, and M. Gerstein. Spectral biclustering of microarray data: Coclustering genes and conditions. *Genome Research*, 13(4):703–716, 2003.

[67] K. Kuhn, S. C. Baker, E. Chudin, M.-H. Lieu, S. Oeser, H. Bennett, P. Rigault, D. Barker, T. K. McDaniel, and M. S. Chee. A novel, high-performance random array platform for quantitative gene expression profiling. *Genome Research*, 14(11):2347–2356, 2004.

[68] J. Lamb. The connectivity map: a new tool for biomedical research. *Nature reviews. Cancer*, 7(1):54–60, January 2007.

[69] J. Lamb, E. D. Crawford, D. Peck, J. W. Modell, I. C. Blat, M. J. Wrobel, J. Lerner, J.-P. Brunet, A. Subramanian, K. N. Ross, M. Reich, H. Hieronymus, G. Wei, S. A. Armstrong, S. J. Haggarty, P. A. Clemons, R. Wei, S. A. Carr, E. S. Lander, and T. R. Golub. The connectivity map: using gene-expression signatures to connect small molecules, genes, and disease. *Science*, 313(5795):1929–35, Sep 2006.

[70] L. Lazzeroni and A. Owen. Plaid models for gene expression data. *STATISTICA SINICA*, 12(1):61–86, JAN 2002.

[71] G. Li, Q. Ma, H. Tang, A. H. Paterson, and Y. Xu. QUBIC: a qualitative biclustering algorithm for analyses of gene expression data. *Nucl. Acids Res.*, 37(15):e101–, 2009.

[72] D.-I. Lin and Z. M. Kedem. Pincer-search: A new algorithm for discovering the maximum frequent set. In *In 6th Intl. Conf. Extending Database Technology*, pages 105–119, 1997.

[73] S. M. Lin, P. Du, W. Huber, and W. A. Kibbe. Model-based variance-stabilizing transformation for illumina microarray data. *Nucleic Acids Research*, 36(2):e11, 2008.

[74] D. J. Lockhart, H. Dong, M. C. Byrne, M. T. Follettie, M. V. Gallo, M. S. Chee, M. Mittmann, C. Wang, M. Kobayashi, H. Horton, and E. L. Brown. Expression monitoring by hybridization to high-density oligonucleotide arrays. *Nat Biotechnol*, 14(13):1675–1680, Dec. 1996.

[75] H. Lodish, A. Berk, P. Matsudaira, C. A. Kaiser, M. Krieger, M. P. Scott, L. Zipursky, and J. Darnell. *Molecular Cell Biology*. W. H. Freeman, fifth edition edition, Aug. 2003.

[76] U. Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17:395–416, December 2007.

[77] K. D. MacIsaac, T. Wang, D. B. Gordon, D. K. Gifford, G. D. Stormo, and E. Fraenkel. An improved map of conserved regulatory sites for saccharomyces cerevisiae. *BMC Bioinformatics*, 7:113, Jan 2006.

[78] S. C. Madeira and A. L. Oliveira. Biclustering algorithms for biological data analysis: a survey. *IEEE/ACM Trans Comput Biol Bioinform*, 1(1):24–45, 2004.

[79] P. M. MARTIN, K. B. HORWITZ, D. S. RYAN, and W. L. McGUIRE. Phytoestrogen interaction with estrogen receptors in human breast cancer cells. *Endocrinology*, 103(5):1860–1867, 1978.

[80] S. Michnick. The connectivity map. *Nat Chem Biol*, 2(12):663–664, 2006. 10.1038/nchembio1206-663.

[81] S. Muangmingsuk, P. Ingram, M. P. Gupta, R. A. Arcilla, and M. Gupta. Dexamethasone induced cardiac hypertrophy in newborn rats is accompanied by changes in myosin heavy chain phenotype and gene transcription. *Molecular and Cellular Biochemistry*, 209:165–174, 2000. 10.1023/A:1007128300430.

[82] J. Munkres. Algorithms for the Assignment and Transportation Problems. *Journal of the Society for Industrial and Applied Mathematics*, 5(1):32–38, 1957.

[83] T. M. Murali and S. Kasif. Extracting conserved gene expression motifs from gene expression data. In *Pac. Symp. Biocomput*, pages 77–88, 2003.

[84] D. Murphy. Gene expression studies using microarrays: Principles, problems, and prospects. *Advances in Physiology Education*, 26(4):256–270, 2002.

[85] J. R. Nevins and A. Potti. Mining gene expression profiles: expression signatures as cancer phenotypes. *Nat Rev Genet*, 8(8):601–609, Aug 2007.

[86] X.-G. Ni, L. Zhou, G.-Q. Wang, S.-M. Liu, X.-F. Bai, F. Liu, M. P. Peppelenbosch, and P. Zhao. The ubiquitin-proteasome pathway mediates gelsolin protein downregulation in pancreatic cancer. *Mol Med*, 14(9-10):582–9, Jan 2008.

[87] A. Oliphant, D. L. Barker, J. R. Stuelpnagel, and M. S. Chee. Beadarray technology: enabling an accurate, cost-effective approach to high-throughput genotyping. *BioTechniques*, Suppl:56–8, 60–1, Jun 2002.

[88] P. Pagel, S. Kovac, M. Oesterheld, B. Brauner, I. Dunger-Kaltenbach, G. Frishman, C. Montrone, P. Mark, V. Stmpflen, H.-W. Mewes, A. Ruepp, and D. Frishman. The mips mammalian proteinprotein interaction database. *Bioinformatics*, 21(6):832–834, 2005.

[89] R. Peeters. The maximum edge biclique problem is np-complete. *Discrete Appl. Math.*, 131:651–654, September 2003.

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

[91] S. Ramaswamy, P. Tamayo, R. Rifkin, S. Mukherjee, C.-H. Yeang, M. Angelo, C. Ladd, M. Reich, E. Latulippe, J. P. Mesirov, T. Poggio, W. Gerald, M. Loda, E. S. Lander, and T. R. Golub. Multiclass cancer diagnosis using tumor gene expression signatures. *Proceedings of the National Academy of Sciences*, 98(26):15149–15154, 2001.

[92] D. R. Rhodes, T. R. Barrette, M. A. Rubin, D. Ghosh, and A. M. Chinnaiyan. Meta-analysis of microarrays. *Cancer Research*, 62(15):4427–4433, 2002.

[93] E. Rinaldis and A. Lahm. *DNA microarrays: current applications*. Horizon Bioscience, 2007.

[94] A. Rosenwald, G. Wright, W. C. Chan, J. M. Connors, E. Campo, R. I. Fisher, R. D. Gascoyne, H. K. Muller-Hermelink, E. B. Smeland, J. M. Giltnane, E. M. Hurt, H. Zhao, L. Averett, L. Yang, W. H. Wilson, E. S. Jaffe, R. Simon, R. D. Klausner, J. Powell, P. L. Duffey, D. L. Longo, T. C. Greiner, D. D. Weisenburger, W. G. Sanger, B. J. Dave, J. C. Lynch, J. Vose, J. O. Armitage, E. Montserrat, A. López-Guillermo, T. M. Grogan, T. P. Miller, M. LeBlanc, G. Ott, S. Kvaloy, J. Delabie, H. Holte, P. Krajci, T. Stokke, L. M. Staudt, and L. M. P. Project. The use of molecular profiling to predict survival after chemotherapy for diffuse large-b-cell lymphoma. *N Engl J Med*, 346(25):1937–47, Jun 2002.

[95] F. P. Roth, J. D. Hughes, P. W. Estep, and G. M. Church. Finding DNA regulatory motifs within unaligned noncoding sequences clustered by whole-genome mRNA quantitation. *Nature biotechnology*, 16(10):939–945, Oct. 1998.

[96] E. M. Russ and J. Raymunt. Influence of estrogens on total serum copper and caeruloplasmin. *Proceedings of the Society for Experimental Biology and Medicine*, 92(3):465–466, 1956.

[97] R. Santamara, R. Thern, and L. Quintales. Bicoverlapper: A tool for bicluster visualization. *Bioinformatics*, 24(9):1212–1213, 2008.

[98] M. Sato, Y. Sato, and L. C. Jain. *Fuzzy Clustering Models and Applications*. Physica-Verlag, 1997.

[99] M. Schena, D. Shalon, R. W. Davis, and P. O. Brown. Quantitative monitoring of gene expression patterns with a complementary dna microarray. *Science*, 270(5235):467–470, 1995.

[100] E. Segal, M. Shapira, A. Regev, D. Pe'er, D. Botstein, D. Koller, and N. Friedman. Module networks: identifying regulatory modules and their condition-specific regulators from gene expression data. *Nat Genet*, 34(2):166–176, June 2003.

[101] E. Segal, B. Taskar, A. Gasch, N. Friedman, and D. Koller. Rich probabilistic models for gene expression. *Bioinformatics*, 17(suppl 1):S243–S252, 2001.

[102] A. Serin and M. Vingron. Debi: Discovering differentially expressed biclusters using a frequent itemset approach. *Algorithms for Molecular Biology*, 6(1):18, 2011.

[103] R. Shamir, A. Maron-Katz, A. Tanay, C. Linhart, I. Steinfeld, R. Sharan, Y. Shiloh, and R. Elkon. Expander–an integrative program suite for microarray data analysis. *BMC Bioinformatics*, 6:232, 2005.

[104] Q. Sheng, Y. Moreau, and B. De Moor. Biclustering microarray data by gibbs sampling. *Bioinformatics*, 19(suppl 2):ii196–ii205, 2003.

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

[106] G. K. Smyth. Linear models and empirical bayes methods for assessing differential expression in microarray experiments. *Stat Appl Genet Mol Biol*, 3:Article3, 2004.

[107] R. R. Sokal and C. D. Michener. A statistical method for evaluating systematic relationships. *University of Kansas Scientific Bulletin*, 28:1409–1438, 1958.

[108] T. Speed. *Statistical analysis of gene expression microarray data*. Interdisciplinary statistics. Chapman & Hall/CRC, 2003.

[109] P. T. Spellman, G. Sherlock, M. Q. Zhang, V. R. Iyer, K. Anders, M. B. Eisen, P. O. Brown, D. Botstein, and B. Futcher. Comprehensive identification of cell cycle-regulated genes of the yeast saccharomyces cerevisiae by microarray hybridization. *Mol. Biol. Cell*, 9(12):3273–3297, 1998.

[110] A. F. Straight, A. Cheung, J. Limouze, I. Chen, N. J. Westwood, J. R. Sellers, and T. J. Mitchison. Dissecting temporal and spatial control of cytokinesis with a myosin ii inhibitor. *Science*, 299(5613):1743–1747, 2003.

[111] A. Subramanian, P. Tamayo, V. K. Mootha, S. Mukherjee, B. L. Ebert, M. A. Gillette, A. Paulovich, S. L. Pomeroy, T. R. Golub, E. S. Lander, and J. P. Mesirov. Gene set enrichment analysis: A knowledge-based approach for interpreting genome-wide expression profiles. *Proceedings of the National Academy of Sciences of the United States of America*, 102(43):15545–15550, 2005.

[112] A. Tanay, R. Sharan, and R. Shamir. Discovering statistically significant biclusters in gene expression data. *Bioinformatics*, 18 Suppl 1:S136–S144, 2002.

[113] A. Tanay, R. Sharan, and R. Shamir. Biclustering Algorithms: A Survey. *Handbook of Computational Molecular Biology*, 2004.

[114] R. Tibshirani, G. Walther, and T. Hastie. Estimating the number of clusters in a dataset via the gap statistic. *Journal of the royal statistical socitety*, 63(2):411–423, 2001.

[115] J. W. Tukey. *Exploratory Data Analysis*. Addison-Wesley, 1977.

[116] A. E. Wakeling, M. Dukes, and J. Bowler. A potent specific pure antiestrogen with clinical potential. *Cancer Research*, 51(15):3867–3873, 1991.

[117] H. Wang, W. Wang, J. Yang, and P. S. Yu. Clustering by pattern similarity in large data sets. In *In SIGMOD*, pages 394–405, 2002.

[118] Z. Wang, M. Gerstein, and M. Snyder. Rna-seq: a revolutionary tool for transcriptomics. *Nature Reviews Genetics*, 10(1):57–63, 2009.

[119] Z. Wen, Z. Wang, S. Wang, R. Ravula, L. Yang, J. Xu, C. Wang, Z. Zuo, M. S. S. Chow, L. Shi, and Y. Huang. Discovery of molecular mechanisms of traditional chinese medicinal formula si-wu-tang using gene expression microarray and connectivity map. *PLoS ONE*, 6(3):e18278, 03 2011.

[120] I. Xenarios, L. Salwnski, X. J. Duan, P. Higney, S.-M. Kim, and D. Eisenberg. Dip, the database of interacting proteins: a research tool for studying cellular networks of protein interactions. *Nucleic Acids Research*, 30(1):303–305, 2002.

[121] E. J. Yeoh, M. E. Ross, S. A. Shurtleff, K. W. Williams, D. Patel, R. Mahfouz, F. G. Behm, S. C. Raimondi, M. V. Relling, A. Patel, and Cheng. Classification, subtype discovery, and prediction of outcome in pediatric acute lymphoblastic leukemia by gene expression profiling. *Cancer Cell*, 1(2):133–143, 2002.

[122] K. Y. Yeung, C. Fraley, A. Murua, A. E. Raftery, and W. L. Ruzzo. Model-based clustering and data transformations for gene expression data. *Bioinformatics*, 17(10):977–987, 2001.

[123] K. Y. Yeung and W. L. Ruzzo. Principal component analysis for clustering gene expression data. *Bioinformatics*, 17(9):763–774, 2001.

[124] M. J. Zaki. Scalable algorithms for association mining. *IEEE Trans. on Knowl. and Data Eng.*, 12:372–390, May 2000.

Notation and abbreviations

Chapter 1

DNA.....	Deoxyribonucleic acid
RNA.....	Ribonucleic acid
A.....	Adenine
G.....	Guanine
C.....	Cytosine
T.....	Tymine
TF.....	Transcription Factor
cDNA.....	complementary DNA
mRNA.....	messenger RNA
preRNA.....	precursor messenger RNA
cMap.....	Connectivity Map
TCM.....	Traditional Chinese Medicine

Chapter 2

PM.....	Perfect Match
MM.....	Mismatch
RMA.....	Robust Multi Array
VSN.....	Variance Stabilization Normalization
VST.....	Variance Stabilizing Transformation
$E(X)$.....	Expected value of the random variable X
$N(\mu, \sigma^2)$.....	Univariate normal distribution with mean μ and and variance σ^2
$E(X\|S)$.....	Expected value of X given Y
GSEA.....	Gene Set Enrichment Analysis

Chapter 3

CC.....	The Cheng and Church Algorithm
ISA.....	Iterative Signature Algorithm
SAMBA.....	Statistical Algorithmic Method for Bicluster Analysis
OPSM.....	Order Preserving Sub-matrices Algorithm
QUBIC.....	Qualitative Biclustering Algorithm
BIMAX.....	Binary Inclusion Maximal Algorithm
BBC.....	Bayesian BiClustering model
GO.....	Gene Ontology

TFBS.....	Transcription Factor Binding Site
KEGG.....	Kyoto Encyclopedia of Genes and Genomes
E.....	Expression matrix
G.....	Genes in expression matrix E
S.....	Samples in expression matrix E
b.....	Bicluster b
B.....	Set of biclusters

Chapter 4

MAFIA.....	Maximal Frequent Itemset Algorithm
HUT.....	Head Union Tail
FHUT.....	Frequent Head Union Tail
MFI.....	Maximum Frequent Itemset
PEP.....	Parent Equivalence Prunning

Chapter 4

DeBi.....	Discovering Differentially Expressed Biclusters
expO.....	Expression Project for Oncology
DLBLC.....	Diffuse large B-cell lymphoma
MSigDB.....	Molecular Signature Database

Chapter 5

c-Score.....	Connectivity Score

Zusammenfassung

High-Throughput-Technologien stellen einen Durchbruch in der experimentellen Molekularbiologie dar. Sie ermöglichen eine Einsicht in die molekularen Mechanismen der Zelle, die mit traditionellen Ansätzen nicht zu erforschen sind. Mithilfe von differenzierten statistischen und computergestützen Methoden können wertvolle Informationen aus diesen Datensätzen gezogen werden.

Clustering ist der am häufigsten gebrauchte Ansatz, um in solchen Hochdurchsatzdaten Gensätze mit verwandten Funktionen zu entdecken. Traditionelle Clustering-Methoden wie das hierarchische Clustering und k-means haben jedoch ihre Grenzen. Erstens basieren sie auf der Annahme, dass sich ein Gencluster in allen Proben gleich verhält. Es ist aber auch möglich, dass ein zellulärer Prozess nur eine Teilmenge der Gene beeinflusst oder dass er nur unter bestimmten Bedingungen seine Wirkung entfaltet. Zweitens wird in traditionellen Clusteriung-Methoden jedes einzelne Gen einem einzigen Cluster zugeteilt, obwohl manche Gene in bestimmten Proben nicht aktiv, andere dagegen in mehrere Prozesse involviert sind. Biclustering überwindet diese Schwierigkeiten, weil dabei Gene und Proben gleichzeitig gruppiert werden. Neue Studien haben gezeigt, dass Biclustering ein grosses Potential für die Entdeckung von Markergenen hat, die mit bestimmten Geweben oder Krankheiten assoziiert sind. Mehrere Biclustering-Algorithmen existieren, aber es ist immer noch schwierig, Bicluster zu finden, deren Signifikanz biologisch validiert ist. Zusätzlich ist es nötig, einen Biclustering-Algorithmus zu finden, der in der Lage ist, sehr grosse Datensätze innerhalb kurzer Zeit zu analysieren.

Der erste Teil dieser Doktorarbeit beschäftigt sich mit Biclustering-Algorithmen. Wir schlagen einen neuen, schnellen Biclustering-Algorithmus speziell für die Analyse von grossen Datensätzen vor. Der Algorithmus findet Bicluster, in denen jedes Gen im Vergleich zu den übrigen Proben in allen Biclusterproben hoch oder niedrig exprimiert ist. Im Gegensatz zu anderen Algorithmen muss die Anzahl der Bicluster nicht *a priori* definiert werden. Anhand synthetischer und biologischer Datensätze vergleichen wir unsere Methode mit andere Biclustering-Algorithmen. GO term und TFBS-Anreicherung zeigen, dass der DeBi-Algorithmus biologisch signifikante Bicluster identifiziert. Wir zeigen auch, dass der Algorithmus nützlich und leistungsstark in der Analyse grosser Datensätze ist. Die Methode kann auf Expressionsdatensätze aus verschiedenen Laboren und von unterschiedlichen Plattformen angewandt werden. Wir hoffen, dass unsere Methode die Entwicklung des Biclustering als Werkzeug für die Datenanalyse in der funktionellen Genomik vorantreiben und unterstützen wird.

Der zweite Teil der Doktorarbeit beschäftigt sich mit der Aufklärung von molekularen Mechanismen in der traditionellen chinesischen Medizin (TCM), sowie mit der Identifikation neuer Kandidaten aus traditionellen chinesischen Heilmitteln für die Entwicklung neuer Medikamente. Für diese Zwecke werden *gene set enrichment tools* und Biclustering eingesetzt. Weiterhin wurde einer Datenbank namens Medicinal Connectivity of TCM, MecoTCM, etabliert, welches systematische Verbindungen zwischen Genexpression, Krankheitsstatus und biochemischer Aktivitt aufbaut. Dadurch gelang es, biologisch relevante Informationen über die TCM-Theorie zu finden, zum Beispiel über die Pflanze Ginseng. Unsere Ergebnisse wurden mit ChipSeq-Experimenten validiert. Künftig soll dieser Datensatz durch die Integration von Tumorgenexpressionsprofilen verbessert werden.

Curriculum vitae

For reasons of data protection, the curriculum vitae is not included in the online version.

Ehrenwörtliche Erklärung

Hiermit erkläre ich, dass ich diese Arbeit selbstständig verfasst und keine anderen als die angegebenen Hilfsmittel und Quellen verwendet habe.

Berlin, September 2011 Akdes Serin