Integrated Analysis of Genomic Data for Inferring Gene Regulatory Networks

A THESIS

SUBMITTED TO THE FACULTY OF THE GRADUATE SCHOOL

OF THE UNIVERSITY OF MINNESOTA

BY

Hossein Zare Sangederazi

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

Doctor of Philosophy

Prof. Mostafa Kaveh and Prof. Arkady Khodursky, Advisors

April, 2009

Acknowledgements

As this chapter of my life comes to close, I would like to express my gratitude to the number of people who supported me and contributed to my research, learning, thinking, and my personality. First, I would like to acknowledge the strong support of my advisor Prof. Mostafa Kaveh and my co-advisor Prof. Arkady Khodursky. My greatest appreciations and gratitude go to both. Prof. Kaveh gave me the freedom to explore my interest and curiosity and fulfill my ambition in research while he was always accessible for any advice in my research and personal life. It was a great pleasure and I was very furtunate to have him as my advisor. Prof. Khodursky was extremely generous when I approached him to seek the opportunity of joining his group. He is a passionate scientist and creative investigator, and was an encouraging, inspiring and motivating force for me during the past four years. I have always enjoyed discussing my works with him and getting involved in fruitful science discussion.

I would also like to extend my gratitude to Porf. Andrew Odlyzko, Prof. Guillermo Sapiro and Prof. Vladimir Cherkassky who kindly accepted to serve on my examining committee. Their useful comments and suggestions certainly enhanced this work.

I am also thankful to Prof. Shmuel Friedland and Dr. Amir Niknejad for their contribution to the wok presented in chapter 2 of this dissertation. I have enjoyed our collaboration on this project and in some other occasions.

Additional thanks go to administrative staff at the Department of Electrical Engineering, especially Ms. Linda Jagerson who was a great help to me in several occasions.

I would also like to thank the University of Minnesota Graduate School for awarding me the doctoral dissertation fellowship.

My warmest thanks go to my friends, roommates and officemates. They were all wonderful, and certainly their support and friendship made my stay in Minnesota a truly enjoyable and memorable stay.

I owe everything that I am today to my family. My deepest gratitude goes to my mother. Her love and support are unbounded and unconditional, and I am forever thankful to her. Also my special thanks go to my sisters and brothers, who very much supported and encouraged me to pursue my education. Finally, I would like to express my gratitude to my late father who passed away one year after I came to the states. I love him and miss him, his support, love, sacrifice and hard works are unforgettable.

In memory of my father

And to my mother.

ABSTRACT

As genomic technology and sequencing projects continue to advance, more emphasis needs to be put on data analysis, while addressing the issue of how best to extract information from diverse data sets. For example, functional annotation of new genes can no longer depends only on sequence analysis, but requires integration of additional sources of information including phylogeny, gene expression, protein interaction, metabolic and regulatory networks. Therefore, new biological discoveries will depend strongly on our ability to combine these diverse data sets. We demonstrate how information from gene expression, regulatory sequence patterns and location data can be combined to discover regulatory modules and to construct gene transcriptional regulatory networks. In the context of modeling regulatory sequences, we propose a higher order probabilistic model to efficiently discriminate between the binding sites of a transcription factor and non-specific DNA sequences. Moreover, a model-based algorithm is developed, which integrates gene expression data, modeled by mixtures of Gaussian, with the regulatory sequence patterns for clustering of functionally related genes.

For the construction of the gene regulatory network, we introduce the concept of Gene-Regulon association in contrast to Gene-Gene interaction. Unlike Gene-Gene interaction methods, where the mRNA levels of the regulators play the important role, Gene-Regulon methods rely on the activity profiles of the transcription factors. These activity profiles, in the absence of their direct measurements, are estimated concurrently via a computational model. We develop a model selection algorithm, which is capable of capturing the activity profile of a transcription factor from the transcriptional activity of its target genes. In addition, we present a data driven approach based on nonlinear kernel embedding for capturing the nonlinear correlation and geometric connectivity pattern in gene expression data. We apply these methods for integrating gene expression and interaction data to construct a network of transcriptional regulation in *Escherichia coli* (*E. coli*).

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Since the first complete genome sequence of a model organism has become available, we have been given the most powerful biological data sets ever generated. But much work still lies ahead to achieve the goal of understanding the complexity of the living systems and uncovering the meaning implicit in the DNA sequences. At the same time, advances in technology have introduced microarray techniques, which allow simultaneous measurement of activity of tens-of-thousands of genes across different conditions or time points. This burst of data coupled with new paradigm in computing through the use of computer and internet brought together researchers from different fields of life and quantitative sciences to recognize the new multidisciplinary research program with the common goal of deciphering this mountainous data. What is presented in this dissertation is an attempt to provide certain computational and mathematical tools for the integration of biological data and the construction of the gene regulatory networks. In this introductory chapter, we first provide an overview of common research areas in the field of genomics and computational biology. Next, the biological data which were used in this study are described, and we end this chapter by outlining the topics which are covered in depth in later chapters.

1.1 Genomics and computational biology

While the researchers in this field may have their own preferences to name this research area, many commonly use computational biology, genomic signal processing, bioinformatics and systems biology interchangeably. Although each field has its own definitions and its own road map, their primary goal is to increase our understanding of biological processes and systems through developing and applying computational techniques to various types of biological data. The major research areas, in the context of molecular biology, can be differentiated by their biological domain into at least four fields, which are briefly described below.

Computational genomics

Computational genomics is the analysis of the whole genome sequences with the help of computers. Its main goal is to annotate whole genomes. Such annotation involves, but may not be limited to,: i) compiling complete whole-genome sequences; ii) deciphering all genes, their coding and regulatory sequences; iii) preliminary functional classification of genes and their products based predominantly on multiple sequence alignments; iv) discovering regulatory motifs, unusual sequence features and characteristic sequence features of a genome; v) prediction of secondary structures in DNA and in regulatory and stable RNA.

Comparative genomics

Comparative genomics is the study of the genome sequence of an organism relative to genome sequences of other related and sometimes distant organisms. It deploys computational techniques to identify ortholog genes and conserved genomic regions among different species. It is the fundamental basis for computational evolutionary biology. It searches for mutation, e.g., single nucleotide polymorphism, deletion and insertion of

DNA segments among sequences of two or more different genomes to trace back their common origin and unique features of the respective evolutionary paths. Statistical and mathematical techniques are applied to build phylogenic profiles among hundreds or thousands of species, using sequence homology among their genes or proteins to identify evolutionary patterns among the species.

Functional genomics

Functional genomics is the study of gene functions and interactions among genes by applying computational techniques to biological data. Unlike computational genomics and comparative genomic, which deal with static aspects of the genome such as DNA and protein sequences, functional genomics deals with the dynamic aspects of the molecular biology of the cells. It involves analysis of gene transcription, translation, protein-protein interactions and interaction between proteins and DNA. Functional genomics uses molecular biological techniques to measure abundance of many molecular entities such as mRNA and proteins simultaneously. It utilizes advanced technologies such as DNA microarray for measuring mRNA levels of thousands of genes and mass spectrometry technology for measuring the concentration of proteins. Computational techniques such as clustering, supervised machine learning and parametric and nonparametric statistical techniques have been widely used in this area to identify differentially expressed genes and functionally related genes.

Systems biology

Systems biology is the study of biological systems at the molecular levels and how the components of biological systems interact and function. It is an exercise in integrating the known parts of a biological system to understand the system as a whole. It requires joining theory and modeling with the powerful prediction capability in order to propose

a specific testable hypothesis about the biological systems followed by genome-wide biological experiments to refine the model and theory. Therefore, the high throughput experimental techniques are the essential components of research in systems biology. An example of such research is the construction of gene regulatory networks from high throughput transcriptomic and proteomic data, which depicts the interactions among genes within the cells.

Proteomics

Proteomics is another field of research attracting bioinformaticians and computational biologists. Proteomics is the study of structure and functions of proteins on a whole genome scale. It provides direct measurements of the amount of proteins present in the cell. Proteomics is complimentary to functional genomics, but it is more complicated. This is because the genome is constant from cell to cell, but different genes are expressed in different cell result in different proteome across the cells. While the same is true for the transcriptome, the complication arises from the fact that it is not as quantitative and not as standardized as microarray-based functional genomics. The study of structure and function of the proteins and interaction between proteins is promising in bio-marker discovery, identifying the efficient diagnostic techniques and discovery of new drugs for treatment of different diseases. Prediction of the structure of proteins through computational simulation and identification of protein-protein interaction form proteomic data are active areas of research in the field, but they are outside of the scope of this dissertation.

1.2 Biological data

An important component of any scientific inquiry is the systematic collection of data for observation, explanation and validation. Most of the major discoveries in all areas

of science have been made possible only by detailed observation and data collection via invention of new measurement technologies. The most pronounced example is the stunning progress in the fields of biological and life sciences during the past decade. In the course of past years, several high throughput technologies were invented for gathering genome-wide gene expression and protein-DNA interactions in vivo. With the availability of such advanced technologies the amount of biological data available today is numerous. Nowadays whole genome sequences of thousands or even more species and massive high throughput transcriptomic and proteomic data sets are publicly available.

High throughput microarray data

A DNA microarray consists of a solid surface, usually a microscope slide, onto which DNA molecules have been attached through chemical or electrostatic interactions [1, 2]. Using DNA microarrays one can detect the presence or abundance of many thousands labeled nucleic acids in a biological sample, which will hybridize to the DNA on the array. When the labeled nucleic acids are derived from mRNA of a sample or tissue the microarray experiment measures the gene expression. There are two main technologies for making the arrays, namely robotic spotting and in-situ synthesis [3]. The spotted microarray is a technology in which pre-synthesized probes, oligonucleotides or PCR product, are attached to the array using robotic printing. In in-situ synthesized arrays oligos are built up base by base on the surface of the arrays and this is the technology which is used, for example, in Affymetrix arrays.

There are usually four main steps in doing microarray experiment to measure gene expression in a sample: sample preparation and labeling, hybridization, washing and image acquisition [1, 2].

There are a number of ways to prepare and label the samples for a microarray experiment and in all cases the first step is to extract RNA from the tissue or sample of interest. Then, the complementary DNA's are labeled using fluorescent labeling by

Cy3 (excited by a ”green laser”) and Cy5 (excited by a ”red laser”). Hybridization is the next step in which the DNA probes on the slide anneal to the complementary labeled DNA targets. Hybridization lasts for 6 to 12 hours. After hybridization, the slides are washed to remove excess hybridization solution from the arrays and to reduce the background. The final step of the laboratory process is to make an image of the hybridized array. Each spot on the array, where the target has bound to the probe, contains dye that fluoresces when excited by light of an appropriate wavelength. This can be done by placing the slide in a scanner which contains one or more lasers that are focused onto the arrays. The result is a digital image, in which each pixel represents the intensity of fluorescence induced by focusing the laser at the point on the array. This image further needs some processing before numerical mapping.

Image processing is the first step in microarray data pre-processing, which involves pixel identification, segmentation and feature and background intensity calculation. Normalization is the next step to resolve the systematic errors and bias introduced by the microarray experimental platform to ensure that the data is high quality and suitable for analysis. However, in this dissertation we pay attention to the problem of missing values in DNA microarrays, and we present a new method for imputing microarray data in the next chapter.

Protein-DNA binding data

In addition to gene expression data, another genome-wide technology is used to gather information about chromosome-wide localization of DNA binding proteins. This information accompanied by genome wide expression data facilitates functional annotation of a genome, including identification of binding sites of regulatory proteins and reverse engineering of gene regulatory networks. The process of gathering this location data can be carried out through array based platform, ChIP-chip, or sequenced based platform, ChIP-Sequencing.

ChIP-chip data

The regulation of gene expression is mediated through binding of specialized proteins to DNA in vivo. Chromatin immunoprecipitation (ChIP) experimental protocol, described in [4], is a biological tool to detect the binding location of proteins along the DNA. In brief, first the protein of interest is cross-linked with DNA in vivo using formaldehyde. Then, cells are lysed and DNA sheared using sonication, which results in double stranded DNA fragments of size 300bp-1kb. The crosslinked DNA-protein complexes can be pooled down from the solution using antibody specific to the protein of interest. The DNA is purified following crosslinks reversal. Finally, the DNA is labeled and hybridized comparatively with control DNA, usually a sample from a mock IP, on the arrays. The collected data may include fluorescent intensities from all intergenic probes as well as the coding regions. Because the detection of specific binding location is of interest the content of probes and probe sizes require careful design in order to obtain high resolution data

ChIP-Sequencing data

ChIP-Sequencing is a technique that combines chromatin immunoprecipitation (ChIP) with massively-parallel sequencing technology to identify and quantify in vivo protein-DNA interactions on a genome-wide scale. The wet lab procedure to obtain ChIP-Sequencing data is similar to that of ChIP-chip, except the purified fragments of DNA from pooled down protein-DNA complexes are sequenced instead of hybridized on the arrays. Using this technology millions of short sequence reads are produced and mapped to the whole genome. This output covers the entire genome and with high redundancy of short reads it provides very high quality data. In addition, this technology has another advantage to ChIP-chip: it does not require any probe design. In chapter 4, analysis of such data will be covered in depth.

1.3 Dissertation roadmap

As genomic technology and sequencing projects continue to advance, more emphasis needs to be put on data analysis, while addressing the issue of how best to extract information from massive data sets. For example, the prediction of the function of a gene or protein depends on many things, including gene structure, expression level, and gene neighbors in a biochemical pathway that are often co-regulated or are found in the same region along the chromosome. On the other hand, functional annotation of new genes can no longer depends only on sequence analysis, but requires integration of additional sources of information including phylogeny, gene expression, protein interaction, metabolic and regulatory networks. Therefore, new biological discoveries will depend strongly on our ability to combine these diverse data sets. Clearly, the performance and value of any integrative modeling rely on the richness of the input sources.

Data pre-processing and feature extraction techniques are the first steps to provide reliable and informative data. To this end, in chapter 2, we first discuss the problem of missing values for micro array gene expression data and propose a method to estimate these values. In chapter 3, we present a mathematical model for the extraction of proper information from regulatory sequence data. We model binding sites with higher order position weight matrices, which accounts for the position-specific dependencies between nucleotides in the sites. We then move in Chapter 4 to discuss location data, and present a methodological framework for analyzing the protein DNA binding data, including ChIP-Sequencing data. In Chapter 5, the extracted information from sequence and binding data are combined with other pieces of biological information. We propose two combining techniques to address the issue of gene function annotation and clustering of functionally similar genes from multiple data sources. The first method relies on the property of kernel matrices in combining heterogeneous data sources, and the second method is based on integrated model-based clustering of gene expression

and sequence data. In Chapter 6, we begin with an overview of existing methods, namely gene-gene interaction techniques, to model gene regulatory networks. Then we introduce a gene-regulon association framework to construct gene transcriptional regulatory network from gene expression and location data. A model driven mathematical technique, which can factor in the activity profiles of the transcription factors, is presented to identify gene-TF interactions. We also propose a data-driven approach by adapting nonlinear kernel embedding approaches that can capture nonlinear correlation and hidden geometric patterns in gene expression data. Finally Chapter 7 concludes this dissertation with the overview of what has been done and suggestions for possible future directions.

Chapter 2

Preprocessing of high-throughput biological data

The raw data of the microarray experiments are the images of the hybridization intensity generated by the scanner. Computational algorithms such as feature extraction techniques are necessary to convert these images to numerical information that quantifies gene expression. This process involves several steps like, identifying positions of the features on the arrays, identifying the pixels which are part of the features and identifying the neighboring pixels to account for background via segmentation techniques. This image processing step results in several numbers of numerical values for each feature including signal's and background's means and medians, and the standard deviation for signal and background. During this process saturated features, features with background intensity greater than signal intensity, and features with high standard deviation are flagged to be excluded for any further analysis. The microarray experiments are subject to systematic error and biases from experimental platform, labeling and etc. Therefore, the numerical microarray data generated using image processing software should be normalized before it can be analyzed to answer any scientific question. Extensive research has been done during the past several years in this area and [5]

provides a good review of normalization techniques for microarray data. In this chapter we deal with another preprocessing step before analyzing the microarray data. This is the problem of handling missing values on microarray data which is explained in the following section.

2.1 Estimation of missing values in DNA microarray data

In a course of a microarray experiment, some spots on the array may be missing due to various factors (for example, flagged features). The most common and straightforward process is to remove these features from the data set, but this has sometimes the disadvantage of removing valuable information and removing the important genes. Because it is often very costly and time consuming to repeat the experiment, molecular biologists, statisticians, and computer scientists have made attempts to recover the missing gene expressions by some ad-hoc or systematic methods. Among all methods, the Bayesian principal component analysis (BPCA) [6], the fixed rank approximation algorithm (FRAA) [7], the weighted K-nearest neighbors (KNNimpute) [8], the local least squares imputation method (LLS) [9], and the projection onto convex sets methods (POCS) [10] have received more attention due to their performance. KNNimpute and LLS are local methods, which use similarity structure of the data to impute the missing values. KNNimpute uses the weighted averages of the K-nearest uncorrupted neighbors. LLS has two versions to find similar genes whose expressions are not corrupted: the L_2-norm and the Pearson's correlation coefficients. After a group of similar genes C is identified, the missing values of the gene are obtained using least squares applied to the group C. In these two methods, the recovery of missing data is done independently, i.e. the estimation of each missing entry does not influence the estimation of the other missing entries.

BPCA is a global method consisting of three components. First, principal component regression, which is basically a low rank approximation of the data set is performed. Second, Bayesian estimation, which assumes that the residual error and the projection of each gene on principal components behave as normal independent random variables with unknown parameters, is carried out. Third, Bayesian estimation follows by iterations based on the expectation-maximization (EM) of the unknown Bayesian parameters.

FRAA is a global method which finds the optimal values of the missing entries of the gene expression matrix G, such that the obtained G minimizes the objective function $f_l(X)$. Here $f_l(X)$ is the sum of the squares of all but the first l singular values of an $n \times m$ matrix X. The minimum of $f_l(X)$ is considered on the set $\mathcal{X}$, which is the set of all possible choices of matrices $X = (x_{ij})_{i,j=1}^{n,m}$, such that $x_{ij} = g_{ij}$ if the entry g_{ij} is known. The completion matrix G is computed iteratively, by a local minimization of $f_l(X)$ on $\mathcal{X}$.

BPCA and LLS outperform FRAA algorithm, however, FRAA performs better for the low rank matrices. In the next section a new algorithm called IFRAA, a combination of FRAA and a clustering algorithm, is proposed to overcome the limitation of the FRAA for higher rank matrices.

2.1.1 Mathematical description of FRAA and IFRAA

Assume that G is the gene expression matrix with missing entries. We can estimate the effective rank of G by computing the effective rank of the submatrix $\hat{n} \times m$, corresponding to all genes with uncorrupted entries [7]. Let l be our estimate for the effective rank of the completed gene expression matrix. Denote by $\mathcal{X}$ the set of all $n \times m$ matrices whose entries coincide with the uncorrupted entries of G. Thus $\mathcal{X}$ is the set of all possible completion of the corrupted gene matrix G. FRAA completes the missing values

of G by finding the minimum to the following optimization problem:

$$\min_{X \in \mathcal{X}} \sum_{i=l+1}^{m} \sigma_i(X)^2 = \sum_{i=l+1}^{m} \sigma_i(G^*)^2. \tag{2.1}$$

Where$G^* \in \mathcal{X}$ and $\sigma_i(X)^2$'s are the singular values of matrix X.

Ideally, G^* is the completion of the gene matrix expression with missing values. In practice, FRAA uses the following iterative procedure to solve the above problem [7].

Fixed Rank Approximation Algorithm: *Let $G_p \in \mathcal{X}$ be the p^{th} approximation to a solution of optimization problem (2.1). Let $A_p := G_p^{\mathrm{T}} G_p$ and find an orthonormal set of eigenvectors for A_p, $\mathbf{v}_{p,1}, ..., \mathbf{v}_{p,m}$. Then G_{p+1} is a solution to the minimum of the following convex nonnegative quadratic function:*

$$\min_{X \in \mathcal{X}} \sum_{q=l+1}^{m} (X\mathbf{v}_{p,q})^{\mathrm{T}} (X\mathbf{v}_{p,q}).$$

The nature of the above problem is similar to inverse eigenvalue problems and in fact the algorithm to solve the above problem for FRAA is based on one of the algorithms for the inverse eigenvalue problems discussed in [11].

The flow chart of this algorithm can be given as:

Fixed Rank Approximation Algorithm (FRAA)

Input: integers $m, n, L, iter$, the locations of non-missing entries $\mathcal{S}$, initial approximation G_0 of $n \times m$ matrix G. **Output:** an approximation G_{iter} of G.

for $p = 0$ **to** $iter - 1$

- Compute $A_p := G_p^{\mathrm{T}} G_p$ and find an orthonormal set of eigenvectors for A_p, $\mathbf{v}_{p,1}, ..., \mathbf{v}_{p,m}$.
- G_{p+1} is a solution to the minimum problem (2.1) with $L = l$.

In practice IFRAA is implemented as follow. First FRAA is used to find a completion matrix G, then a cluster algorithm is applied to find a reasonable number of clusters of similar genes. Presumably each cluster is a relatively smaller matrix having an effective low rank. For each cluster of genes FRAA is separately applied to recover the missing entries in this cluster. It turns out that this modification results in a very efficient algorithm for reconstructing the missing values of the gene expression matrix.

2.1.2 Simulation results

For comparison of different imputation algorithms, six different types of data sets were used, consisting of four microarray gene expression data and two randomly generated synthetic data. Two data sets of microarray were obtained from studies for the identification of cell-cycle regulated genes in yeast (*Saccharomyces cerevisiae*) [13]. The first gene expression data set is a complete matrix of 5986 genes and 14 experiments based on the Elutriation data set in [13]. The second microarray data set is based on Cdc15 data set in [13], which contains 5611 genes and 24 experiments. Two other yeast data sets were obtained from "http://sgdlite.princeton.edu". The Evolution data set has been studied in [14] and Calcineurin data set has been studied in [15]. Two synthetic data sets are randomly generated matrices of size 2000×20 and ranks 2 and 8 respectively.

To assess the performance of missing value estimation methods, we performed the following simulations. On the first two microarray data sets and on the synthetic data 1%, 5%, 10%, 15% and 20% of the entries were randomly deleted from the complete matrix C. Then the various completions of data matrix were obtained by estimating the missing values using BPCA, IFRAA and LLS. The K-value parameter (number of similar genes) was set such that there was no increase in performance of the LLS by increasing the value of k.

A normalized root mean square error (NRMSE) was used as a metric for comparison. If C represents the complete matrix and $\hat{C}$ represents the completed matrix using

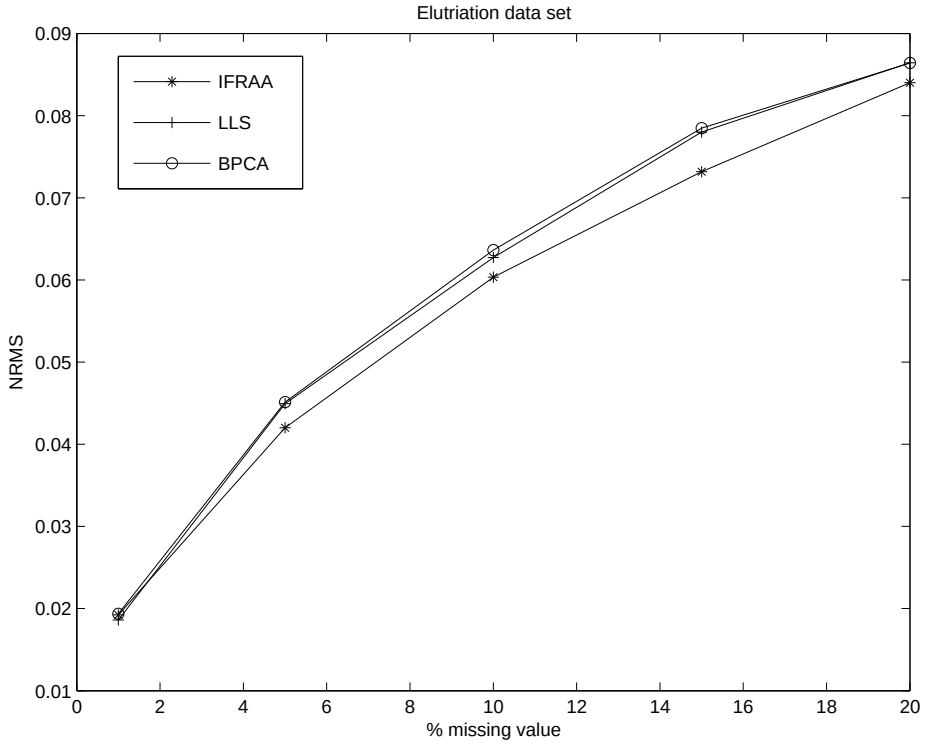

Figure 2.1: Comparison of NRMSE against the percentage of missing entries for three methods: IFRAA, BPCA and LLS. Elutriation data set in [13] with 14 samples.

an estimate to the corrupted entries in C, then the root mean square error (RMSE) is $\frac{\|D\|_{\mathcal{F}}}{\sqrt{N \times M}}$, where $D = C - \hat{C}$. We normalized the root mean square error by dividing RMSE by the average value of the entries in C.

In IFRAA the parameter L, which is the number of significant singular values plus 1, was chosen by comparison of ratio of two consequent singular values. We observed that this parameter appeared to be equal to 2 or 3 depending on data set and may differ for each small block of data (cluster). The initial guess for the missing entries in each gene was chosen to be the row average of its corresponding row.

Figure 2.1 depicts the comparison of BPCA, IFRAA and LLS for Elutriation data set in [13]. We break the whole gene expression matrix by clustering the data into groups of genes, which form matrices with effective low ranks, and FRAA was applied

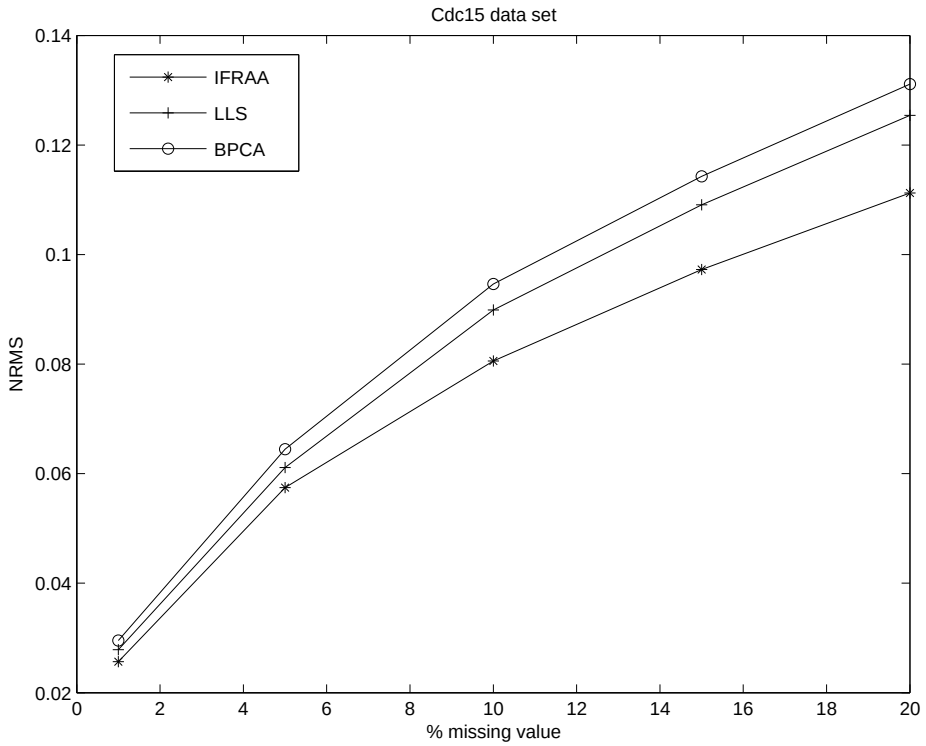

Figure 2.2: Comparison of NRMSE against the percentage of missing entries for three methods: IFRAA, BPCA and LLS. Cdc15 data set in [13] with 24 samples.

on each group. The graph is the average over 25 runs, and as can be seen for this data set IFRAA performed the best, BPCA and LLS have very close performance with significant gap with IFRAA.

Figure 2.2 depicts the comparison of BPCA, and LLS for Cdc15 data set in [13] which contains 5611 genes and 24 experiments. In this case IFRAA again performed the best and LLSimpute performed slightly better than BPCA.

The performance of the BCPA, IFRAA and LLS algorithms depends on the unknown distribution of the positions of missing entries. To study this issue we applied all methods on the original data sets containing missing values. Since NRMS error could not be calculated for these actual missing values, we transferred the missing value positions from the original data to corresponding positions in the complete data

Table 2.1: Comparison of NRMSE for four missing value estimation methods: IFRAA, LLS, BPCA and FRAA for actual missing values distribution for three gene expression data sets with different percentage of missing values.

Data sets	IFRAA	LLS	BPCA	FRAA
Cdc15 data set %0.81 missing	0.0175	0.0200	0.0216	0.0335
Evolution data set %9.16	0.0703	0.0969	0.1247	0.1107
Calcineurin data set %3.68	0.0421	0.0445	0.0453	0.0753

derived from the original data set before applying the algorithm. By doing this the distribution of missing value positions in complete data set is almost unchanged from the actual distribution. The result is illustrated in Table 2.1 for four data sets including the original data set of Cdc15 which contains $\%0.7$ missing values ($\%0.81$ missing in complete data), Evolution data set [14] which contains $\%8.457$ missing values ($\%9.1$ missing in complete data) and Calcineurin data set [15] which contains $\%3.2$ missing values ($\%3.68$ missing in complete data). This result again confirms the superiority of the IFRAA for the actual microarray data missing value estimation.

The random matrices of size 2000×20 and of ranks $k = 2, 8$ appearing in Figures 2.3 and 2.4 were generated as follows. One generates $2k$ random column vectors $\mathbf{x}_1, \ldots, \mathbf{x}_k \in \mathbb{R}^{2000}, \mathbf{y}_1, \ldots, \mathbf{y}_k \in \mathbb{R}^{20}$, where the entries of these vectors are chosen according to an uniform distribution. Then $C = \sum_{i=1}^{k} \mathbf{x}_i \mathbf{y}_i^{\mathrm{T}}$.

Figure 2.3 represents the comparisons of BPCA, IFRAA and LLS for 2000×20 random matrix of rank 2. The performance of the three algorithms is excellent for 1% of missing data. The performance of LLS constantly deteriorates with the increase percentage of missing data. The performance of BPCA also deteriorates with the increase percentage of missing data, but less than LLS. IFRAA performed outstandingly.

Figure 2.4 represents the comparisons of BPCA, IFRAA and LLS for 2000×20 random matrix of rank 8. The performance of LLS is the same as in Figure 2.3. BPCA and IFRAA performed extremely well. IFRAA slightly outperformed BPCA in particular in the case with 20% of missing data.

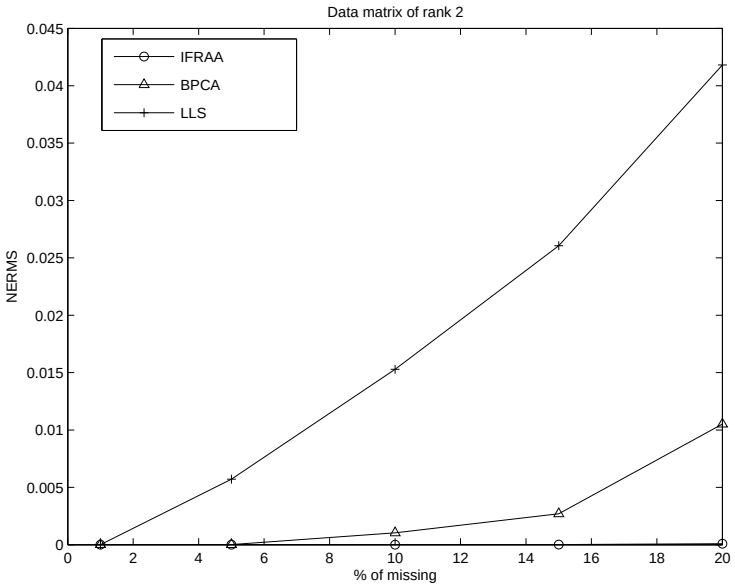

Figure 2.3: Comparison of NRMSE against the percentage of missing entries for three methods: IFRAA, BPCA and LLS. Data set was a 2000×20 randomly generated matrix of rank 2.

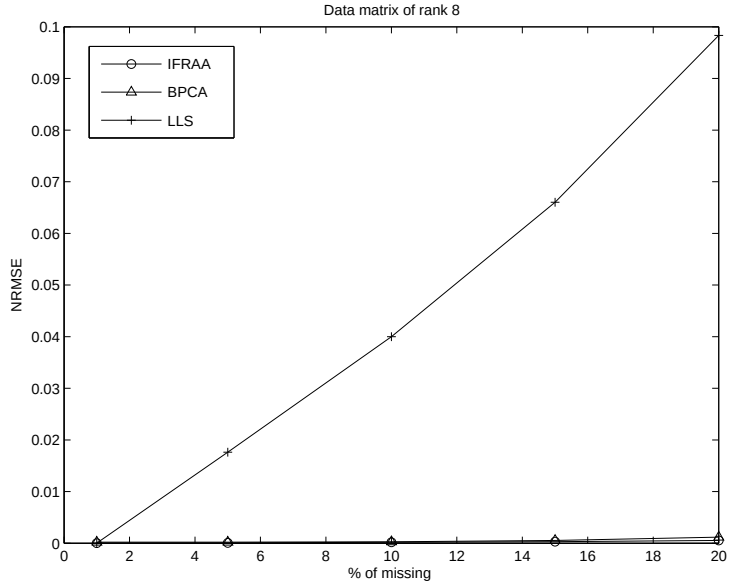

Figure 2.4: Comparison of NRMSE against the percentage of missing entries for three methods: IFRAA, BPCA and LLS. Data set was a 2000×20 randomly generated matrix of rank 8.

Chapter 3

Analysis of regulatory sequence data

Control of transcription and replication depends on recognition of specialized DNA sequences by regulatory proteins. These specialized sequences, generally referred to as ”binding sites”, are relatively short segments of DNA embedded within larger regulatory regions. Though in many cases it is essentially impossible to discriminate between the binding site and non-specific DNA by available analytical means [16], the successful execution of regulatory programs in the cell depends on timely, specific and sensitive interactions between regulators and their cognate targets. Therefore, the analysis of regulatory networks depends on how the genetic information stored in DNA sequences is retrieved and modeled. While the mechanism(s) of efficient binding site targeting in vivo are only partially understood, it is essential to identify the regulatory elements via computational means.

Over the years, considerable research effort has been applied to develop pattern discovery and pattern matching algorithms to systematically identify binding sites for a given transcriptional regulator or transcription factor (TF) across a genome.

In many situations, there is a collection of genes known to be regulated by a certain transcription factor and the problem of interest is how to discover the binding sites

pattern from the regulatory sequences of these genes. There are two group of computational approaches for binding sites pattern discovery. The first group are those which try to find overrepresented words in promoter regions of co-expressed genes [17, 18]. The second class includes algorithms which allow variability in sites to identify matrix representation of the binding sites pattern. These approaches are based on greedy algorithm [19, 20], iterative algorithms based on expectation maximization [21] and Gibbs's sampling method [22, 23, 24, 25]. These algorithms start with an arbitrary alignment matrix and iteratively remove and add new sites to the matrix in order to increase its information content. Both algorithms are proven to converge, however they may not provide the global optima of the objective function.

In pattern matching algorithm the assumption is that there is given a collection of known sites and the goal is to identify new binding sites, and therefore new targets for a given transcription factor. A brief review of matching methods is provided in the next section, and a new approach to the representation of binding sites that results in a significant improvement in differentiation between true and false-positive sites is proposed.

3.1 Pattern matching

Position weight matrix

When the genome sequence of an organism is available and there are some known binding sites for a given TF, one can computationally predict additional sites by scanning the DNA sequences for short segments sharing common features with the known sites. The simplest and most widely used method to do so relies on a position-specific scoring matrix (PSSM), or a position weight matrix (PWM), surveyed in [26]. In PSSM, DNA binding sites are modeled in such a way that nucleotides at each position of the site contribute independently to the binding. The PSSM matrices are constructed from

the alignment of known binding sites which have been identified experimentally. The PSSMs are $4 \times l$ matrices (l is the length of the sites) with rows indexed by nucleotide $i \in \{A, T, C, G\}$ and columns representing positions $j \in \{1, ..., l\}$. The entries of the matrix are the frequencies of the occurrences of each nucleotide at each position. However, there are some modifications for regularizing small sample sizes by incorporating pseudocount parameters in building the matrices. The elements of position weight matrix are log-odd values which are calculated as $w_{i,j} = \log(\frac{f_{i,j}}{p_i})$, where $f_{i,j}$'s are entries of PSSM matrix and p_i is the probability of observing the symbol i in a genome or a background model.

The weight matrix can be used to score any segment of DNA sequence of the same length as a known binding site in the alignment. The score for the particular sequence $Y = y_1, ..., y_l$ by PWM ,W, is given by $S(Y|W) = \sum_{j=1}^{l} w_{y_j,j}$, which is related to the probability that the sequence Y is a binding site given the matrix W. Typically, a cutoff value or a score threshold is applied to predict the DNA segment as a new site.

There are two main concerns with the above method. First, experimental evidence [27] suggests that the assumption of independent contribution of each position to the overall binding affinity is often not valid. Also, Stormo and his group [28, 29] analyzed binding affinity measurements using microarray technology and showed interdependence among the positions in DNA targets. They demonstrated that the additivity assumption in DNA-protein interactions does not fit the data perfectly, but in many cases provides a good approximation. Second, due to the choice of the score threshold, this method suffers from a high false positive rate and also misses some true sites. To partly overcome the problem of specificity, prior biological knowledge has been incorporated to filter out some false positive sites from a set of identified sites [24, 30]. However, even with such improvements, the performance of these methods has remained limited. On the other hand, the evolution of transcription factor DNA binding sites has been studied in [31] and shown that several positions of binding sites regulating different

genes contain non-consensus nucleotides which are conserved in distant genomes. This result indicates the necessity of constructing a comprehensive model which includes as much information as possible from both consensus and non consensus positions to represent the binding sites.

The dependency assumption between nucleotides has been investigated through biological experiments and a linear modeling of binding probabilities [27] and through modification of the PWM [30, 32, 33, 34]. It has been shown that accounting for the dependency structure of binding sites increases the specificity and prediction power of the pattern-matching algorithms, and results in a more accurate prediction of protein-DNA binding affinity. Barash et al. [32] have suggested that it should be possible to partition all known sites into few groups and define a PWM for each group in a way similar to the one described above, assuming that interaction effects of nucleotides in binding are negligible for the sites within each group. The final score for a DNA segment then would come from the mixture of these PWMs. A modification of this method has been presented in [34], where the number of PWMs is equal to the number of known sites. While relatively little has been done with predicting regulatory replication sites [35], the same issues are expected to restrict the utility of PWM in extracting replication signals from DNA sequences.

In the next section, an new algorithm is presented for constructing higher order position weight matrices which accounts for the position-specific dependencies between nucleotides in the sites. Higher order matrices can be constructed for dinucleotides or trinucleotides over the significant dependent position pairs or triads.

3.2 Higher order position dependent weight matrices

In this section the modeling of binding site sequences using PWMS of first, second and third order from a set of known binding sites is considered. Hereafter, by second

and third order we mean that PWM matrices are defined for dinucleotides and trinucleotides. In forming these matrices we consider not only the dependency between adjacent nucleotides but also the dependency of non-adjacent nucleotides. This assumption distinguishes our modeling from Markov chains of different order[36] and simple dinucleotide PWMs, which only take into account dependency of adjacent nucleotides. An optimized version of the markov chain [37] uses the ordered markov chain in which pairs of significant position are adjacent, however this algorithm suffers from high complexity to find an optimum order for the chain.

Choosing Candidate pairs and triads to construct higher order PWMs

A second-order PWM is a $16 \times L_2$ matrix, $\mathcal{M}_2$, where L_2 is the number of pairs of dependent positions among the total number of $\binom{L}{2}$ pairs. Similarly, a third-order PWM is $64 \times L_3$ matrix, $\mathcal{M}_3$, where L_3 is the number of triads of nucleotides of dependent positions having significant dependency chosen from the total number of $\binom{L}{3}$ triads. Pearson's χ^2, Chi-square, test statistic is used to find which pairs or triads are significantly dependent. Therefore, the Null hypothesis for our test is that nucleotides at positions i and j or for triads nucleotides at positions i, j and k are independent. Let $f_i(x)$ be the observed count of nucleotide x at position i for a given training set of N sequences and $G_{i,j}(x_1, x_2)$ be the joint observed count of occurrence of nucleotide x_1 at position i and nucleotide x_2 at position j. Then the expected count of nucleotides x_1 and x_2 occurring jointly at positions i and j is $E_{i,j}(x_1, x_2) = f_i(x_1)f_j(x_2)/N$. Let $X = \{A, T, C, G\}$ then, the χ^2 value for positions i and j is defined as

$$\chi^2(i,j) = \sum_{x_1 \in X} \sum_{x_2 \in X} \frac{(G_{i,j}(x_1, x_2) - E_{i,j}(x_1, x_2))^2}{E_{i,j}(x_1, x_2)}.$$

The χ^2 value for triad (i, j, k) can be defined in a similar way using joint observed count and expected count of trinucleotides occurring at triad (i, j, k).

We compute the p-values for χ^2 values to choose significant pair and triad candidates to form PWMs. Low p-values corresponding to large χ^2 values indicate some sort of dependency between nucleotide positions forming respective pairs and triads. Usually the p-value of 0.05 is used to select significant dependent-position pairs and triads.

Having chosen the candidate pairs or triads, the entries of matrices $\mathcal{M}_2$ and $\mathcal{M}3$ will be log-odd values of the observed frequency of dinucleotides or trinucleotides in dependent positions calculated from training set and that of background model. Then, depending on the information content of each matrix one can select the matrix with higher information content to compute the score for a given sequence Y. One can also build a combined model using weighted average of the normalized scores calculated from each matrices,

$$S(Y|\mathcal{M}) = \textstyle\sum_{i=1}^{3} \omega_i \hat{S}(Y|\mathcal{M}_i)$$

where $\mathcal{M}$ is a combined model, $\hat{S}(Y|\mathcal{M}_i)$ is the normalized score of sequence Y from position matrix of order i, and $\omega_i \geq 0$ with $\sum_{i=1}^{3} \omega_i = 1$, are the coefficients weights, which can be estimated from training data as follows. Let X be the set of m known binding sites for the transcription factor F and $\omega = [\omega_1, \omega_2, \omega_3]$ be the vector of coefficients weights (we only included matrices up to order 3) such that, $\sum_{i=1}^{3} \omega_i = 1$. Then one can choose ω^* to be

$$\omega^* = \arg\max_{\omega} \sum_{y \in X} \sum_{i=1}^{3} \omega_i \hat{S}(y|\mathcal{M}_i^y).$$

Here $\mathcal{M}_i^y$ is the position weight matrix of order i constructed from all known binding sites in X excluding y.

Our analysis has revealed that when one of the models performs substantially better on the training set, that model can be used for prediction of new sites in the genome and it would have prediction power comparable to that of the combined model. In

the following due to space limitation we only compare the performance of individual matrices.

3.3 Results and discussion

Data set

To assess the performance of our algorithm and to check the richness of the higher order models in capturing the dependency structure of binding sites we used the JASPAR [38] data set of eukaryotic transcription factor binding sites matrices, E. coli transcription factors' binding sites, yeast replication sites and human splicing sites. The JASPAR data set along with TRANSFAC data set; *http://www.gene-regulation.com/pub/databases.html*, are the main sources of known eukaryotic transcription factor binding sites. We chose JASPAR because it mostly contains experimentally documented sites. At the time of this study there were 106 transcription factors with the number of cognate sites ranging between 6 and 116, with an average of 30 sites per regulator. The width of the sites for these TFs ranged between 5bp and 22bp (there was only one site with the width of 4bp). Since the information content for higher order models of the sequence elements with very small number of known sites is low, we applied our model to 77 TF's with the number of known sites greater than 15.

3.3.1 Comparative analysis of performance of higher order PWMs

To compare the proposed algorithm with previously published works, we used human splicing sites that have been used in [39] to learn the maximum entropy model. Since we could not access the original training signals and decoys used in [39], we have learned our model on 2400 randomly chosen human 5' splicing sites. These sites are short segments of DNA, 9bp long, that consist of the last three bases of the exon and

six first bases of the succeeding intron. For the comparison, we used 20 sequences available in [39] and their corresponding rank for different methods, and the odds ratio defined as the frequency of occurrence of the sequence as a splice site divided by its occurrence as a decoy. Since the global rankings of these sequences varied widely from one model to another, we used only relative ranking of these sequences. Thus, the measure of performance is how the relative ranking of the sequences obtained from each model correlates with their odds ratio ranking, and the reasonable conclusion is that the model for which the correlation is higher can be assumed to be a better and more accurate model to represent the data. The sequences and their relative ranking for different methods and the odds ratio ranking are listed in Table 3.1. The Spearman's and Kendall's correlation coefficients between odds ratio ranking and relative ranking derived from each model are listed in the last two rows of Table 3.1. The correlation coefficients show significant correlation between relative ranking of the sequences derived from the third order PWM and the odds ratio ranking. These results indicate that higher order position weight matrices with their simple structure perform better than some highly complex probabilistic models, which may over-fit the data. This is consistent with previous observations that in some scenarios, due to the over-fitting problem, PWMs outperform higher order Markov models [33].

3.3.2 The learning procedures for JASPAR data set

We compared the performance of the models for each TF in JASPAR data set separately in two different ways using the following procedures. In the first procedure we used the "leave-one-out cross-validation method" to compare the performance of the first, second and third-order PWMs. For each transcription factor F, we chose one binding site out of m known sites and learned the models with the remaining sites, then we scored the "left-out" site for each model. By repeating this procedure m times we generated 3 vectors (one for each model) of length m containing the scores for all m

Table 3.1: Comparison of different methods based on sequences of human splicing sites. Entries for each method are the relative ranks of sequences. MDD is maximal dependence decomposition [40]; me2x5 and me2so are the maximum entropy models with different sets of constraints. me2x5 was the best maximum entropy model among many models of different orders and constraints [39].

Sequences	3^{rd} order PWM	me2x5	MDD	me2s0	PWM	odds ratio	odds ratio rank
ACGGTACGT	1	5	19	11	19	331	1
AAGGTACGT	3	18	18	7	12	233	2
ACGGTAAGT	6	1	2	4	10	184	3
AACGTAAGT	8	19	10	19	11	96	4
ATGGTAAGT	5	9	6	18	9	95	5
TCGGTAAGT	7	2	3	9	18	77	6
CAGGTACGG	2	14	20	16	15	68	7
GAGGTAAGT	16	8	16	5	4	38	8
CCGGTAAGT	10	12	1	6	8	22	9
CAGGTGAGT	20	20	14	3	6	21	10
CCGGTGAGT	14	13	4	15	14	18	11
CAGGTAAGA	15	16	12	10	5	14	12
CAGGTAAGG	17	7	13	12	3	13	13
AAGGTAAGT	18	10	5	2	2	12	14
ACGGTGAGT	12	3	9	14	17	11	15
GACGTAAGT	9	11	17	20	16	10	16
TCGGTGAGT	13	6	15	17	20	9	17
CAGGTAAGT	19	15	11	1	1	8	18
GCGGTAAGT	4	4	7	8	13	3	19
CGGGTAAGT	11	17	8	13	7	2	20
Kendall's corr.	0.3684	0.0211	-0.0737	-0.0211	-0.2		
Spearman's corr.	0.5128	0.0256	-0.0887	0.009	-0.194		

sites recognized by a transcription factor F. In learning the second and third order models we chose the p-value of 0.05 for χ^2 statistic test of independency between pairs and triads positions.

Since for higher order models we did not use all possible combinations of the positions, for each model we normalized the probability score of each site by the average probability scores of random segments of the same width. These random segments were taken from random sequences of length 1000kbp, which was generated by a 3rd order Markov model of intergenic sequences. We tested the difference between the scores of each model using one-tailed t-test, (e.g. $S(x|\mathcal{M}_2) - S(x|\mathcal{M}_1)$). The performance of one model was considered significantly better than that of the other, if the t-test indicated so at a significance level of 0.05. Based on this analysis, for 29 TFs the performance of the second-order model was better than that of the first-order, for 42 TFs the performance of the third-order model was better than that of the first-order, and for 55 TFs the performance of the third-order was better than that of the second-order model. In the second procedure, we ranked the scores of the known sites among the scores for random segments. This provided us with the measure of falsely discovered sites, which had higher scores than the known sites. For each transcription factor's binding site we calculated the rank of its score for each model. Let $r(Y|\mathcal{M}_i)$ be the rank of the site Y when the model i is used. For each TF, we computed the representative rank by averaging the ranks of all known sites, $R_{(\mathcal{M}_i)} = \frac{1}{N_{F_j}} \sum_{k=1}^{N_{F_j}} r(Y_k|\mathcal{M}_i)$ for transcription factor F_j having N_{F_j} known sites. A model is considered to be better for a TF if its corresponding average rank is smaller than that of other models. We assumed the rank difference is significant if for one model the average rank is more than 3 fold smaller than that of other models. In 32 cases out of 77 TFs the second-order model outperformed the first-order one, the third-order model performed better than the first-order in 45 cases, and the third-order model was better than the second-order for 60

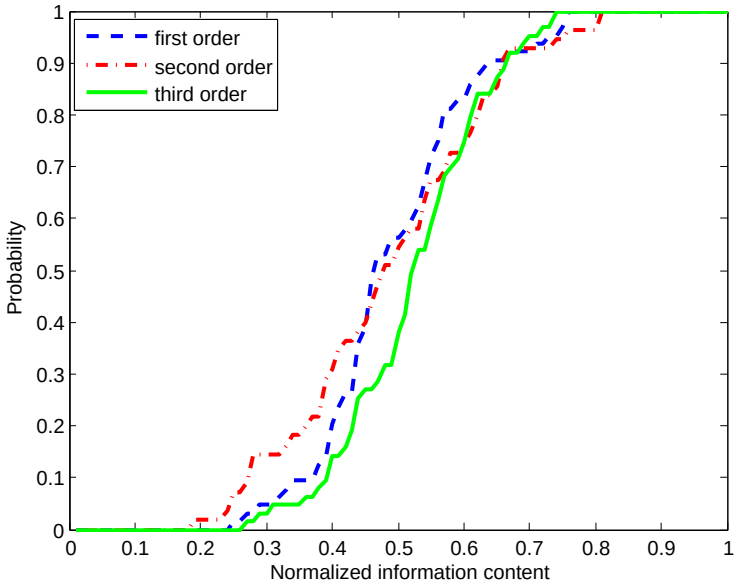

Figure 3.1: Comparison of matrices of different order for Jaspar data set: Cumulative distribution function of information content of the PWM of the first, second and third-order for selected TFs in the Jaspar data set.

TFs. When there were no second- or third-order matrices, we assumed that they performed worse than first-order model in our comparison. This happened for some TFs, since no position pairs or triads passed the significance test by χ^2 statistic. We also computed the information content of all three models (three matrices) for the selected TFs. Figure 3.1 depicts the cumulative distribution function of normalized information content for three models. It can be seen from the figure that the normalized information content of the third-order model is higher than that of the second- and first-order model for the majority of the TFs. This increase in average information content is due to dependency between adjacent and non adjacent positions which fully can not be captured using first order PWM and simple Markov models.

3.3.3 Results for E.coli LRP, HNS, and IHF transcription factors

It has been reported that DNA binding sites of three E. coli transcription factors, LRP, IHF and HNS, illustrate very low binding specificity when a position weight matrix of the first-order is used to predict new sites or to estimate the average binding energy of a collection of known sites [41]. We applied our model to construct the second and the third-order matrices by selecting significant pairs and triads. The p-value of 0.05 was chosen for χ^2 statistic test of independency between pairs and triads positions. Next, these matrices were used to score DNA segments in the windows corresponding to the width of the annotated binding sites in 500 bp upstream regions of all E. coli genes. When scanning a genome, it is common to set a threshold value on the scores to determine new sites. The choice of this threshold value may have a significant effect on the specificity and sensitivity properties of any algorithm. A low threshold value results in many false positive sites and decreases the specificity. However, a high threshold value reduces the prediction rate of the known sites and, in turn, decreases the sensitivity of the algorithm. We fixed our threshold for each model to allow the same percentage of the known sites to be predicted. This ensured similar sensitivity values for the three models. Since the number of negative samples in this problem is too high, we could not use the standard receiver-operator curves, which rely on a common definition of specificity, to compare the performance of different models. Instead, the left and right panels in Figure 3.2 depict sensitivity values versus the number of predicted sites for the first- and third-order model, respectively. This analysis indicated that, at the same sensitivity threshold, the first-order model predicted at least two orders of magnitude as many sites as the third-order model, which in turn resulted in a very poor specificity of the PWM for these transcription factors.

To show how the higher order model increases the specificity, we identified the top most significant triad positions and the corresponding trinucleotides, which contributed to the dependency test. For each dependent triad, we chose two groups of binding sites

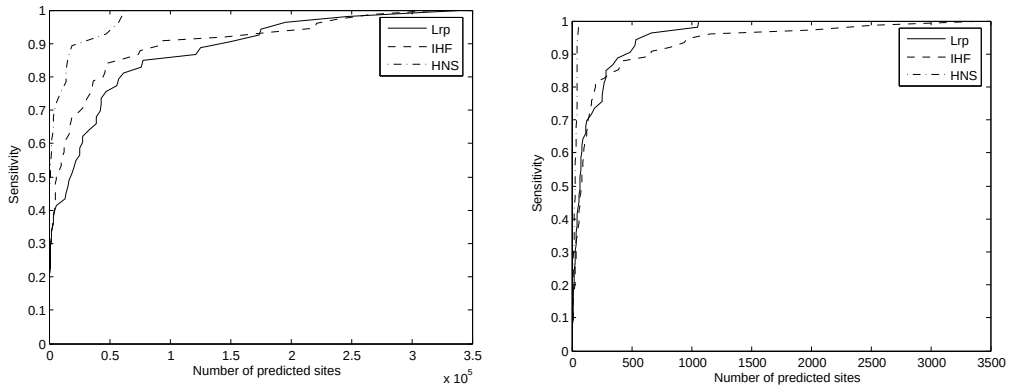

Figure 3.2: Specificity and sensitivity comparison of different order PWM's for 3 E-coli LRP, IHF and HNS transcription factors; Left: 1st order model, Right: 3rd order model. For the same sensitivity for two models, the smaller number of predicted sites by the 3rd order model results in a considerable reduction in the number of false positive sites.

each containing a selected trinucleotide. Figure 3.3 shows the sequence logo of LRP binding sites and the sequence logos of subgroups for the top 4 triads. It is clear that several positions, which have low information content based on the first-order model, have very high information content in those subgroups that can be captured with the third-order matrices.

It is important to evaluate the power of any computational model on a real data set, for example in predicting putative transcription factor binding sites. To this end we chose the transcription factor LRP and further analyzed the results of the model. (We only carried out the comparison between the third and the first-order models.)

For each gene, we selected a site in a corresponding strand with the maximal score and ranked all selected sites for both models. The median rank of the collection of the known sites was calculated for both models. For the known target genes which have more than one binding site, we ranked only the site which had the maximal score. Table 3.2 shows the ranks and median ranks of the sites for 17 known target genes. From gene expression microarray data [42], the median rank of 17 Lrp targets is 126. Thus, while the scores determined by the first-order model were clearly inconsistent

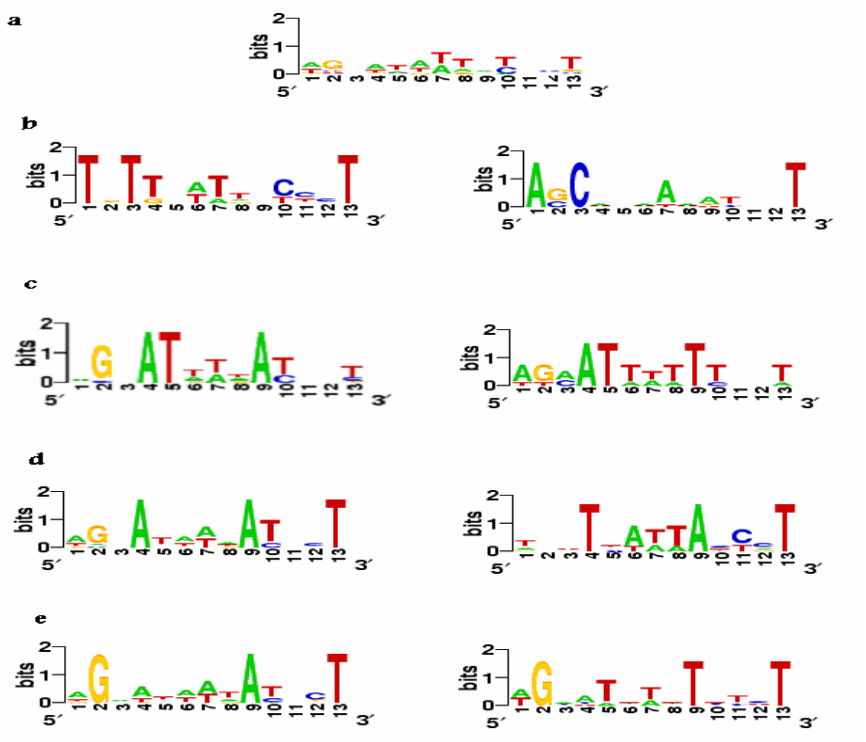

Figure 3.3: Example of significant triads for LRP: Sequence Logos of Lrp binding sites, (a) Logo from the whole set of known sites, (b) Logo of 2 groups of binding sites which showed significant dependencies at positions 1,3,13 with two different trinucleotides, (c) significant dependencies at positions 4,5,9 (d) significant dependencies at positions 4,9,13 (e) significant dependencies at positions 2,9,13.

with the observed transcriptional activity of the set, the scores from the third-order model supported the transcription data. The probability that such similarity between the median transcription and site scores occurred by chance is less than 1 in 100,000. To calculate this probability we built the null distribution of medians of transcription and site scores. The null distribution was constructed by: randomly choosing 17 genes from genome and calculating the median rank of their transcription and their site scores, then we calculated the probability that a particular set of genes having median transcription rank of less than or around 126 also have the site score median rank of smaller than 126. Moreover, we found that for sensitivity of above 80%, the lists of genes were very

significantly enriched for transcriptionally affected genes, when we examined the lists of genes with corresponding sites scored by the third-order model at different sensitivity cut-offs (Table 3.3).

Table 3.2: Rank order of the site scores for 17 known target genes of LRP.

Gene Name	First order rank of the site	Third-order rank of the site
fimA	3830	116
gtlB	224	259
gcvT	1264	695
serC	656	26
ompF	865	85
ompC	409	21
osmY	1393	57
osmC	532	1539
livJ	3687	1
oppA	4286	6
kbl	1318	7
ilvI	7	64
dadA	8	8
lrp	3257	43
stpA	3334	469
yeiL	353	152
lysU	2686	18
ilvG	3977	9
Median	1064	50

As pointed out by many investigators, one limitation of the PWM-based methodology is the assumption that the nucleotide positions in the sites contribute independently to the total activity of the sites. We were interested in seeing if our higher order model was assigning scores which better capture differential affinity of a regulator to the sites. To that end we used the genome-wide binding data for the LRP protein. The relative signal intensities for each micoarray probe were obtained as a result of the comparative two-color hybridization between the DNA sample bound by Lrp and specifically precipitated by Lrp antibodies and the DNA sample recovered from the cells lacking Lrp protein (manuscript in preparation). The normalized log ratios of the signal intensities

Table 3.3: Significance analysis of predicted sites using gene expression data.

Sensitivity	Fraction of expressed genes (%)	p-value
1	30	3.3E-18
0.96	25	2.1E-11
0.92	20	1.1E10
0.88	17	1E-7
0.84	15	3.6E-6
0.80	14	1.8E-5

from two channels were calculated and used as binding affinity scores of the sequences located upstream of the known target genes. Assuming that LRP binds in the vicinity of the known target genes in vivo, we wanted to determine whether there was any correlation between the binding affinity score and the corresponding site score for these genes. Figure 3.4 illustrates the relationship between the affinity scores and the site scores from the first-order model (left panel) and for the third-order model (right panel) for 17 known target genes. Interestingly, and consistent with our hypothesis, the scores from the first-order model did not correlate with the affinity scores, whereas the scores from the third-order model showed significant correlation with the affinities (r=0.41, with a p-value of 2.2×10^{-3}). Moreover, when we removed the two least transcriptionally responsive genes from the list, dadA and ompC, the correlation between the site scores from the 3rd-order model and affinity scores increased to 0.6, while it did not improve correlation with the 1st order affinity scores (data not shown).

Prediction of replication origins in yeast

Replication initiation sites, referred to as origins of replication, are DNA segments that must be recognized by specialized proteins prior to initiation of the DNA copying reaction at these sites or their vicinity. Due to their large size, eukaryotic chromosomes often have multiple origins of replication on each chromosome. Like many other cellular processes, replication of DNA also has two structural components: replication

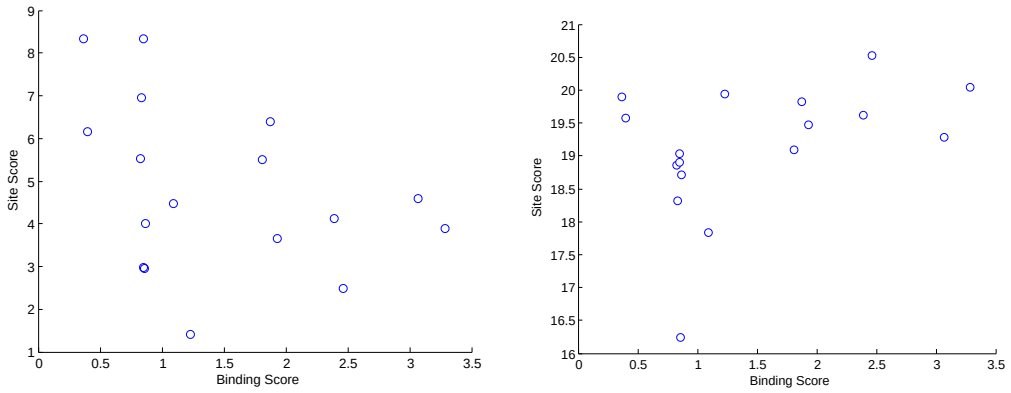

Figure 3.4: Correlation of site scores and binding affinity: Relationship between binding scores from chip-chip data and site scores from the first-order(left) and third-order model(right) for 17 known target genes. There is a positive correlation of 0.41 with p-value of 2.2×10^{-3} between the scores of third-order model and binding affinity scores for Lrp sites while no significant correlation for the first-order model exists.

initiation sites, the segments of DNA sequences that direct the initiation of DNA synthesis, and initiators, protein complexes that recognize specific DNA elements and start replication. Replication origins in the yeast *Saccharomyces cerevisiae*, also called autonomously replicating sequences, or ARS elements, have been systematically investigated. Each of the elements has a short (17bp) DNA sequence called an ARS consensus sequence or ACS that binds to replication initiation proteins [35]. Due to variation in the primary structure of the sites, computational prediction and discovery of putative replication initiation sites has been challenging. In [43], prediction of these sites using position weight matrices constructed from known ACSs had very poor performance resulting in many falsely identified origins. The authors incorporated some additional information about flanking sequences (243bp) and by doing so were able to filter out many false positive sites and improve on their predictions. To assess the performance of our model, we used the known ACS and evaluation sets provided in [43]. We trained our model using only known sites and scanned all of 16 yeast chromosomes for new replication sites. Again, the p-value of 0.05 was chosen for χ^2 statistic test of independency between pairs and triads positions. To streamline the analysis, we applied only

the third-order model to DNA segments which passed a mean threshold score for the first-order model. We were able to predict 60% of the sites in the evaluation set without the use of any additional or flanking sequences.This sensitivity value corresponds to a positive predictive value (PPV) of 0.30. The PPV for the first-order model was close to zero, when the threshold was set to achieve the same sensitivity as for the third-order model. These results demonstrate the performance and utility of the higher order PWM in extracting sequence patterns for regulatory DNA elements, including transcription and replication sites.

Chapter 4

Analysis of protein-DNA binding data

Detecting associations between proteins and DNA signals is an important part of the gene regulation studies and therefore is essential for understanding of many biological process and their anomalies. Control of transcription and replication depends on the recognition of specialized DNA sequences, binding sites, by regulatory proteins. Over the past few years, in addition to computational motif discovery algorithms which were discussed in previous chapter, high throughput technologies namely ChIP-on-chip and ChIP-Sequencing have emerged to identify transcription factor binding sites and other functional elements along the genome and they are discussed in this chapter.

4.1 Technological and computational aspects of chromatin immuno-precipitation data

ChIP-on-chip is an experimental technique which uses chromatin immunoprecipitation and microarray technology to identify the binding of proteins to DNA in vivo [4, 44]. The first step in experimental procedure involves cross linking of protein to DNA in vivo followed by lysing the cell and sonication of the genomic DNA. Then, the DNA fragments bound to the protein are isolated through chromatin immunoprecipitation

(IP) using an antibody specific to the protein of interest. The purified DNA fragments are isolated by revers cross linking process and they are amplified using PCR [1] and labeled by fluorescent dye. The sample DNAs which are not enriched by IP process are also amplified by PCR and labeled by different fluorescent dye. The enriched and unenriched samples both are hybridized on microarray chips in order to measure the abundance of the enriched sample across the whole genome. The arrays are scanned and the numerical data are extracted from the images for analysis. Similar to the analysis of the gene expression microarray, statistical techniques such as t-test and the analysis of variance can be applied to identify enriched regions. However, these analysis provide very low resolution for the specific positions of the binding sites on the genome. One possible solution is to input the sequences of the enriched regions to motif discovery algorithms to identify more precise positions of the binding sites [45].

The second solution is to use higher resolution microarray chips. Microarray chips such as affymatrix tiling arrays exist today, which are specifically designed for ChIP-chip experiments. The DNA microarray probes on these arrays are designed in order to cover the whole genome. Using these chips one can obtain a signal which covers the whole genome with high resolution and the positions of binding sites can be identified by detecting the peaks in the signal. Peak detection algorithms have been developed via mathematical modeling of the ChIP-chip data from tiling array to detect the locations of the binding sites with higher resolution [46, 47]. However, with the availability of the ChIP-Sequencing which provides data with higher resolution and redundancy, most of the research efforts are directed toward the analysis of this data.

[1] Polymerase Chain reaction (PCR) is the process of amplifying a DNA fragment via in vitro enzymatic replication using a DNA polymerase.

4.2 Entropy-based peak localization in ChIP-sequence data

ChIP-Sequencing technology combines chromatin immunoprecipitation with next generation sequencing technology [48] for the same purposes as ChIP-on-chip. Using this technology millions of short sequence reads are produced and mapped to the whole genome. The shortness (25-36bp) of sequence reads provides very high resolution in identifying the precise locations of enriched DNA fragments, which can be interpreted as binding sites. Thus, ChIP-Sequencing is a promising and alternative technique to ChIP-on-chip that allows identification of transcription factor binding sites, especially in organisms with high genome complexity. Since the technology is relatively young, there are only a few studies on ChIP-sequencing data[48, 49]. Algorithms presented in these studies identify peaks in the signal using a global threshold and depending on the choice of threshold they may have high false positive or high false negative rates. In [49] an additional ChIP-Seq data set (pool of non-immunoprecipitated DNA) has been used to adjust the threshold for a fixed false discovery rate (FDR). However, it is not apparent why the mock IP, an immunoprecipitation reaction without antibodies, would generate the relevant background distribution of sequence reads. Therefore, with the growing demand for ChIP-Sequencing it is necessary to provide a more statistically sound framework for the analysis of ChIP-Seq data.

In the next section, we present a data analysis approach that takes into account the biological fact that transcription factors bind with higher affinity to the promoter regions than to the coding regions. Based on such assumption, the data from the coding regions can be treated as background, or null distribution in the absence of the mock IP. In this framework, the regions with high affinity binding to the transcription factor will be locally identified using the regional relative entropy measure. This quantity is

related to the difference in the free energies of regulatory protein binding to the promoter region and the coding region. Therefore, it is a natural measure of the promoter binding affinity. Then, for each region, the locations of the binding sites are identified by detecting locally distinct peaks. In section this approach is evaluated on a newly generated ChIP-Seq data set for the E.coli transcription factor Lrp.

Method

Chip-Sequencing data contains millions of short (25-36bp) individual sequence reads generated from a pool of immuno-precipitated DNA. These reads come from both strands of DNA and therefore, need to be mapped to both strands of the genome which can increase redundancy in the data. The greater the number of reads that are mapped to any particular region of the genome the higher specificity of that region in binding to the transcription factor. We take into account both forward and reverse sequence reads and derive the numerical data, which represent the number of times each base pair position of the genome "participated" in binding. Let random variable B represent a binding position taking value over the whole genome. Then the probability that the protein binds to ith position in the genome can be estimated from ChIP-Seq data: $P(B = i) = \frac{n(i)}{T}$, where T is the normalization constant and $n(i)$ is the number of reads overlapping with position i.

To identify the enriched region we consider the fact that transcription factors tend to bind to the promoter regions of the genes. Because most regulatory proteins bind their cognate targets with different affinities, global peak finding algorithms may miss some, if not many, targets. We assume that the distribution of the binding in the coding region of the genes represents the background model with a very low probability of binding at those positions. To be able to compare these binding density profiles we defined them with respect to their relative distances to the start of the gene. Let $P(i)$ be the probability of binding at position i relative to the start of the gene on the promoter

side and $Q(i)$ represent the probability of binding at position i relative to the start of the gene on the coding side. Then for each gene, similar to the notion of relative entropy, we define the regional relative entropy (RRE) as follow to identify genes for which the binding distributions to their upstream and downstream regions are significantly different.

$$RRE = \sum_{i=1}^{M} P(i) \log(\frac{P(i)}{Q(i)})$$

where M is the range of the region considered in calculation of the relative entropy. Normally $400 < M < 700$ is chosen to cover the promoter regions.

To identify target genes, the null background distribution of the relative entropy is generated by considering only the coding regions and a threshold level is chosen by controlling a false discovery rate. Genes whose promoter regions have significantly high relative entropy are identified as likely targets of the transcription factor and the selected promoter region is further investigated for peak(s) to identify binding sites.

Having identified binding regions, the binding profile of the selected region is smoothed using kernel average smoother as follows .

$$\widehat{P}(i) = \frac{\sum_{j=i-\lambda}^{i+\lambda} K_h(i,j) P(j)}{\sum_{j=i-\lambda}^{i+\lambda} K_h(i,j)}$$

where $h = 2\lambda$ is kernel radius and $K_h(i,j) = g(\frac{\|i-j\|}{h})$ where g is a kernel function (here the Gaussian function). Within each region the coordinates of well-separated (50bp) local maxima of $\widehat{P}$ with a peak level above the threshold (regional threshold obtained for a significant p-value from Poisson distribution) are identified as centers of binding sites.

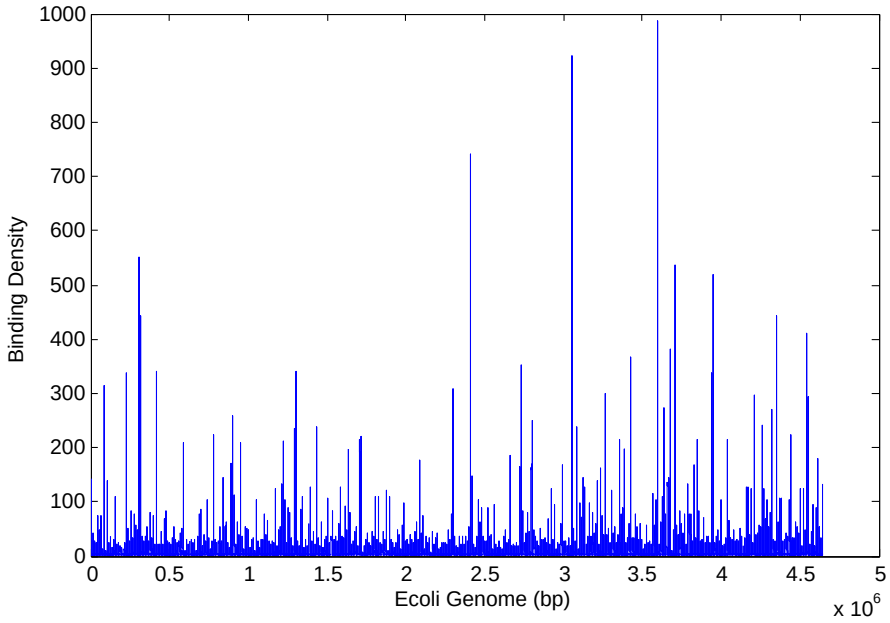

Figure 4.1: Genome scale binding density profile of transcription factor lrp.

4.3 Identifying targets of LRP from ChIP-Sequencing data

We generated Chip-Seq data for the E. coli Lrp transcription factor from sequencing a DNA pool enriched for DNA fragments bound to Lrp in vivo. The sequencing was done using Solexa platform, which generated around 3 millions short sequence reads of 25bp. Almost 1.2 million reads were mapped to the genome. Figure **??** depicts the overall profile of data points across the whole genome scale. The objective was to identify regions of the genome which were specific to the transcription factor Lrp. To identify the location of specific binding across the genome one may use a global peak finding algorithm to call significant peaks. However, due to the variable nature of the affinity of transcription factors to their cognate target genes, a global threshold may result in many false positive peaks or may miss some of the specific binding regions with low affinity.

With the prior information that transcription factors have higher affinity to the promoter regions than to the coding regions, the presented algorithm attempts to identify binding regions rather than the peaks. For each gene we considered the 600 bp upstream and downstream sequences to calculate its regional relative entropy. For our background model we only used the coding region sequences to generate the null distribution of the relative entropy. This null distribution is used to adjust the cut-off threshold to identify promoter regions with significant binding energy. We set the threshold on relative entropy for the false discovery rate (FDR) of 20%. The FDR was calculated as the ratio of the number of cases above the threshold in the background model to the number of identified promoter regions using the same threshold. At this FDR level 195 promoters were identified as enriched regions. This list included more than 90% of all known Lrp targets. Figure 4.2 depicts the binding profiles of 4 known targets with the locations of the known binding sites indicated on the horizontal axes. Although, the location of a few known sites matched with the peak locations in binding profile, there were several mismatches as well. These mismatches can be explained by the fact that the locations of the known binding sites have been identified through binding of purified protein to DNA in vitro. On the other hand, ChIP-Seq data represent the binding of the protein to DNA in vivo and therefore, the locations of the peaks likely correspond to the actual binding sites. Similar results were obtained for the rest of the known targets. This information can be used to reexamine the structure of previously identified sites and the nature of a signal that is being recognized by Lrp.

4.4 Characterization of essential DNA regulatory features using protein-DNA interaction data

To characterize DNA regulatory elements of the genes regulated by transcription factor Lrp, the 50 bp sequences surrounding the highest peaks in the identified regions from

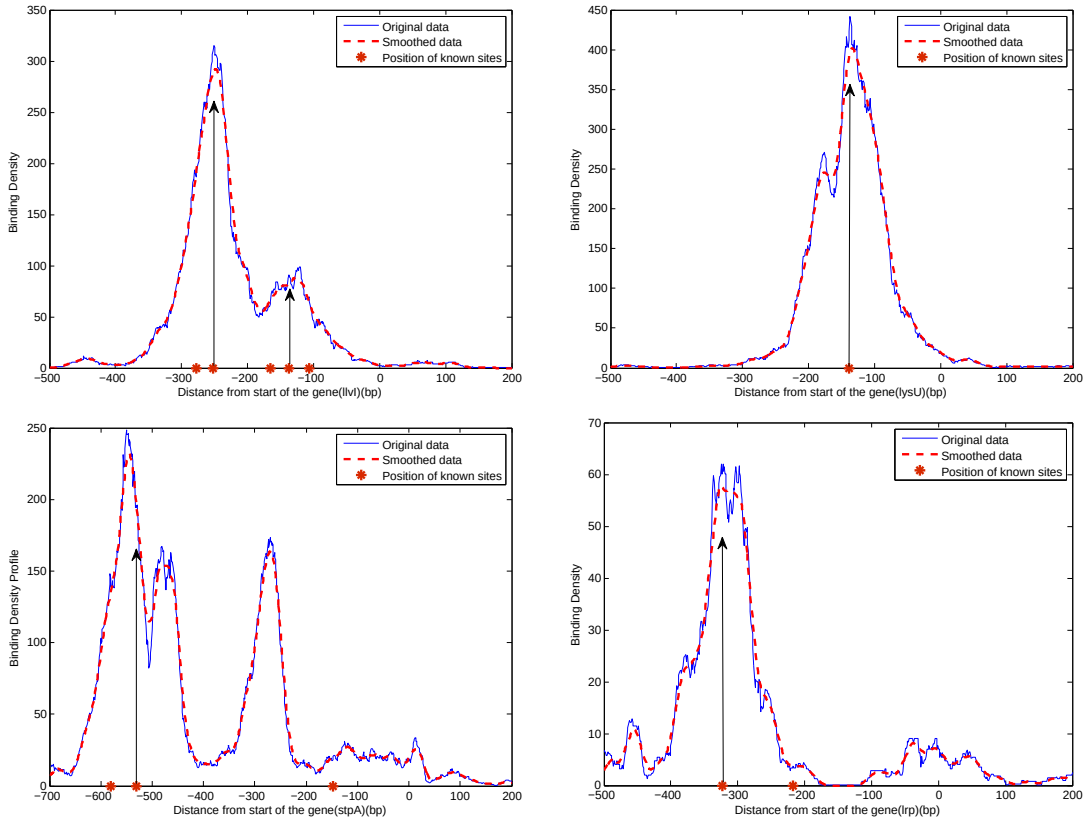

Figure 4.2: Binding density profile of 4 known lrp targets, ilvI,lysU, stpA and lrp

ChIP-Sequencing data were used to extract sequence features which can explain the binding affinity of the Lrp protein to those regions. We searched for significantly over-represented 3,4 and 5-mers among the set of 195 selected sequences. Sequence words that occurred at least two times more frequently than would be expected by chance alone were selected as significantly over represented words. The total of 45 features were selected. We defined the score for each feature as the logarithm of its odds ratio: $S(w) = \log \frac{P_w}{P_w^b}$, where P_w is the probability of occurrence of word w in our sequence list and P_w^b is the chance probability of occurrence of the word w in intergenic region of the E.coli genome. We assumed the background base pair probability of 0.3 for Nucleotide 'A' and 'T' and 0.2 for Nucleotide 'C' and 'G', which is a very good estimation for E. coli intergenic regions.

Further, the sequence score was calculated using the word score: $S(s) = \sum_{w \in \Omega} n_s(w)$ $S(w)$. Where Ω is the set of all significant words and $n_s(w)$ is equal to the number of times word w appears in sequence s. To assess the relevance of these selected sequence features in explaining binding affinity we compared the calculated sequence scores with their corresponding peak values. These scores show significant positive correlation (pvalue < 0.05) with the peak values. Figure 3 shows the correlation between these two quantities with respect to the number of selected top targets. To clarify that high correlation for low number of genes is not simply due to the effect of low sample size, we showed the correlation values for the same sample sizes when we only permuted the sequence scores. The correlation coefficients between the peak values and permuted scores were averaged over 1000 permutations and they are shown with error bars in Figure 4.3. Clearly, even when the list contains all of the selected genes the correlation between sequence scores and peak values remains positive above the random correlation. We also used the same sequences as input to a motif discovery algorithm [19] which generated a motif of length 13 bp (Figure 4.4). The motif score did not show strong correlation with peak values, but the correlation was positive around 0.25 for all sets containing different numbers of top genes. Although the positions of the binding sites using motif discovery algorithm may reflect the specificity of the protein to the regions surrounding the sites, the binding affinity can not only be explained by this short sequences representing binding sites (at least for Lrp transcription factor) and the sequence features extracted from longer DNA segments surrounding the sites are more informative.

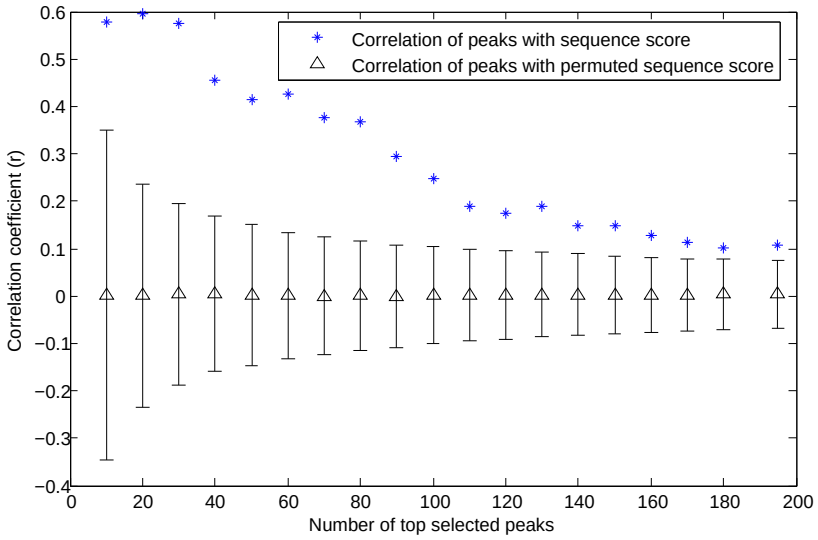

Figure 4.3: Correlation between the peak values and the scores of 50bp sequences surrounding the peaks. The correlation values were calculated for the different set sizes (number of top peaks considered).

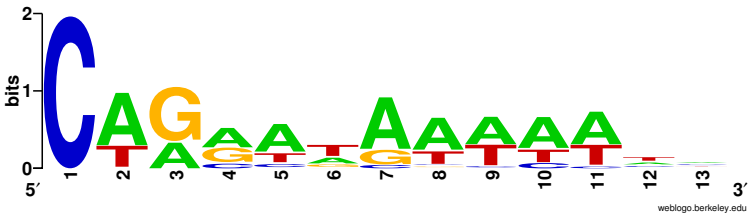

Figure 4.4: LRP Motif generated from sequences surrounding the peaks

Chapter 5

Clustering algorithms for multiple sources of data

Unsupervised learning methods, and clustering algorithms in particular, have become an integral part of the analysis of DNA microarray data. Common clustering algorithms have been extensively used not only for visualization of high-dimensional gene expression data, but also for grouping functionally related genes, which share similar expression profiles [50]. While a great deal of research has been conducted in adapting the clustering techniques for gene expression microarray data [51], a limited number of studies have been done on clustering genes from multiple sources of biological data [52, 53, 54].

Different types of biological data reveal different features of the relationships between the genes in a genome. It is quite plausible that combining high-dimensional data from multiple sources would improve capturing the critical biological information for the purpose of gene function prediction. The idea of combining information from multiple sources for gene function prediction is not new. However, the process of combining can be done in different stages of the analysis and by different methods, which may greatly affect the outcome. Moreover, given a set of computational methods and

biological tasks, the optimal procedure remains to be developed. In this chapter, we discuss the problem of establishing relevant relationships among genes using multiple sources of information. In the following, after a brief review of clustering algorithms, we explain two frameworks for statistical combining of data from microarray gene expression and sequence motif data or location data. In section 5.2 the application of Kernel functions is considered for combining heterogeneous data sources and a clustering algorithm based on simulated annealing is presented to cluster genes using Kernel matrices. In section 5.3 an integrated model-based clustering algorithm is presented which combines the mixtures of Gaussian for gene expression data and the probabilistic sequence model for motif or location data.

5.1 Overview of clustering methods

Clustering is the process of assigning N objects, $x_1, x_2, ;x_N$, in d dimensional space to one of K groups. For example, clustering algorithms are used to identify groups of samples with similar expression levels or groups of genes having similar expression profiles across several conditions. A fundamental component of any clustering algorithm is the measure of similarity, or dissimilarity, of the objects that are being clustered. Euclidean distance and pairwise correlation are common measures of dissimilarity and similarity for numerical data respectively. However, depending on an application domain, different metrics may be defined which can be much more powerful in capturing the structure of the data than the Euclidean distance or linear correlation is able to do.

Hierarchical clustering and partitioning methods are the most popular techniques used for grouping of genes and samples from gene expression data. In hierarchical clustering [55], the nested sequences of clusters are produced using a bottom-up (agglomerative) or an up-down (divisive) method. In the bottom-up method, each object

form a cluster of size one, and at each step two closest clusters are joined until all objects are put in one cluster. The common measures of closeness between two clusters are *single linkage*, which uses the minimum distance between points in two clusters, *complete linkage*, which uses the maximum distance between points in two clusters and *average linkage*, which uses the average of all distances between points in two clusters. A disadvantage of the bottom-up method is that if an incorrect join is made in the early stages, the error propagates up to the top of the tree and the top level clusters may very poorly reflect the structure of the data. Therefore, when interest is to have a few large clusters, the top-down method, which successively partitions the data is likely to produce more informative clusters. On the other hand, when the small size clusters are required the bottom-up approach is preferred because the top-down method is likely to generate poor clusters after many splits.

Unlike hierarchical clustering, partitioning methods tend to separate objects into a preset number of groups. Algorithms such as K-means [56] and self-organizing maps (SOMs) [57] seek to partition the object into K groups by minimizing some measure of within-group dissimilarity. This optimization problem is combinatorial and in most cases the optimum partition cannot be found. The K-means clustering algorithm starts with K random partitions and iteratively calculates the centers of the clusters and maps the observation to the nearest center and then update the centers. SOMs are similar to K-means, but with additional constraint that forces the clusters which are close to one another to lie in adjacent cells in a low dimensional space such as a grid.

In addition to the above algorithms, spectral clustering [58], fuzzy C-means clustering [59], which allow for the creation of the overlapping clusters and clustering based on Gaussian mixture models [60], which will be discussed in more details in section 5.3 of this chapter, are among other methods adapted for clustering of microarray data.

5.2 Clustering using kernel methods

Kernel-based statistical learning methods provide a computational framework for integrating heterogeneous data sources, and have already proven to be very effective tools in bioinformatics [52]. These methods represent data by means of a kernel function, which is an appropriate non linear mapping of the original input space to a higher dimensional feature space. This function defines similarities between pairs of genes, which can be quite complex relations in the original space, as simple inner products in a feature space.

Given data items $\mathbf{x_1}$ and $\mathbf{x_2}$ the kernel function can be defined as

$$K(\mathbf{x_1}, \mathbf{x_2}) =< \phi(\mathbf{x_1}), \phi(\mathbf{x_2}) >, \tag{5.1}$$

where ϕ is a non-linear function for embedding the data items, and $< . , . >$ denotes the inner product operation. As can be seen, what is needed to define the kernel function is the inner product, and explicit representation of the mapping function, ϕ, is not needed (see Table 5.1 for examples of popular kernel functions). Evaluating the kernel on all pairs of points will give us a symmetric, positive semidefinite matrix K known as a kernel matrix, which can be seen as a matrix of similarity measures for the genes. We define a kernel matrix independently for each set of data which represents a specific type of information. Then these kernel matrices are combined in order to improve the cluster specificity and overall prediction of genes function. An important feature of the kernel formalism is that basic algebraic operations such as addition, multiplication and exponentiation preserve the key property of positive semi-definiteness and allow simple combining of the kernel matrices. We consider weighted combination of kernels to allow different contributions of each data source for clustering and gene function prediction.

Table 5.1: Common kernel functions

Polynomial Kernel	$k(\mathbf{a}, \mathbf{b}) = (\mathbf{a}.\mathbf{b} + c)^d$
Gaussian Kernel	$k(\mathbf{a}, \mathbf{b}) = \exp(-\|\mathbf{a} - \mathbf{b}\|^2/2\sigma^2)$
Sigmoid Kernel	$k(\mathbf{a}, \mathbf{b}) = \tanh(c(\mathbf{a}.\mathbf{b}) + \theta)$

Here, given a set of kernels, $K_1, K_2, ..., K_P$, we will form the linear combination

$$K = \sum_{p=1}^{P} w_p K_p, \tag{5.2}$$

where w_p's are weight coefficients for combining these kernels. These coefficients are computed using available training data sets. Then, the final kernel, K, is used as an overall similarity measure for the genes to partition the gene set to the groups of similar and functionally related genes.

Clustering algorithm

For the simplest case when all the kernels represent similarity properties, we devise the following algorithm to cluster the genes. Let $S_1, ..., S_P$ be similarity matrices based on the individual sources of data. Let the number of clusters, N , be fixed. Suppose we want to partition the set of genes, G, into the N disjoint subsets, such as $c_1 \bigcup c_2 \bigcup ... \bigcup c_N = G$. We define the gain function as

$$f(C) = \sum_{n=1}^{N} \sum_{l=1}^{L_n} \sum_{p=1}^{P} w_p.S_p(g_{nl}), \tag{5.3}$$

where C denotes a particular partition scheme, L_n is the number of genes in the cluster n, $S_p(g_{nl})$ is the normalized sum of similarity of the lth gene in the nth cluster with other genes in the same cluster computed from the data source, G_p, and w_p is a proper weight for the p-th similarity matrix. Then the optimal partition can be found by maximizing the gain function over all possible partition schemes, $\mathcal{C}$, i.e.,

$$C^* = arg \max_{C \in \mathcal{C}} f(C), \tag{5.4}$$
$$S.t. \quad \sum_{p=1}^{P} w_p = 1.$$

The weight factors depend on the importance of each data source and can be found by training on a subset of genes with known functions. A simple method for finding these weights is based on the correlation of each individual similarity matrix for a group of gene pairs with similar functions. Any source which gives the higher number of similar pairs among the set will be assigned a higher weight.

We propose two algorithms to find the optimal partition scheme C^*. The first algorithm is as follows. Starting from any gene (the cluster seed), preferably a gene that belongs to a known functional category, the second candidate gene for this cluster will be the gene which has the maximum combined similarity with the cluster seed. The third candidate gene is the gene which has the maximum combined similarity with the seed and the second gene in the cluster. This procedure is continued while the normalized similarity of the cluster remains above a certain predefined threshold. The process is repeated by removing these genes from the data sets and starting a new cluster from a new random gene or a gene from a known functional category. The algorithm stops when the number of clusters reaches K. The remaining genes can be grouped in a separate cluster or they can be assigned to one of the obtained clusters by relaxing the threshold criterion.

For the second algorithm, the simulated annealing method [61] is employed to find the optimal partition scheme, which maximizes the gain function. At each temperature of cooling procedure, T, the state is perturbed many times and the algorithm moves to the next temperature on the basis of updating the gain function value. We have modified the procedure, as follows, to fit it to our problem and to speed up the algorithm. Let C_0 be any initial random or semi-random partition and $f(C_0)$ be the related gain

function. We choose one gene at random from the first cluster and place it into other clusters and compute the normalized combined similarity (NCS) of each cluster. We keep the selected gene in a cluster, which gives the maximum NCS and remove it from other clusters. This process is repeated by choosing the gene from another cluster and assigning it to the new cluster or keeping it to its own cluster. After doing this procedure K times, we will have a new partition C_1 according to acceptance probability of 1, if $f(C_1) \geq f(C_0)$ and $\exp(\frac{-(f(C_0)-f(C_1))}{T})$ if $f(C_1) < f(C_0)$. Then the algorithm decreases the temperature parameter T and proceeds to the next step. The procedure continues until the system reaches the steady state, i.e., there is no further change in the partition. The initial partition can be chosen in such a way that each cluster contains some genes which are known to have the same functional assignment.

Results

To assess the performance of our simple combining algorithm, we have applied the algorithm to the data available for a model prokaryotic microorganism, *Escherichia*. We used regulatory sequence information and gene expression data for over 50 physiological conditions generated in our laboratory. Considering the regulatory sequences as an independent source of information necessitates the use of an independent and universal measure of similarity between all pairs of genes. All the published approaches that have combined the analysis of regulatory sequences with expression data, have used the gene expression data as a prior source of information to extract the features of the sequences, thereby violating the assumption of the independent contribution of each source. Since pairwise similarities between relatively short strings in DNA cannot be adequately captured by means of a local or global alignment, other approaches are needed to describe similarities between regulatory sequences.

It is well known that similarly regulated genes may share binding sites for common TF's, which in turn control the expression levels of these genes under certain

conditions. We constructed a motif-count matrix using consensus motifs for 50 known transcriptional regulators. The consensus matrices for these transcription factors were obtained from RegulonDB web site [62], and we scanned all regulatory sequences for these consensus matrices to find the occurrences of the motifs in our gene sets. In the motif-count matrix, each entry corresponds to a number of occurrences of each motif in upstream region of each gene with its score higher than a threshold value. Due to incompleteness of this data set, the obtained dissimilarity matrix is not complete; nonetheless, it contains information about the similarities among a substantial fraction of genes in the genome.

To evaluate the biological relevance of the calculated clusters, we related each gene to a biological process and/or function and determined the enrichment of the functional annotation labels in a cluster. To do so, we used the gene annotation data set in which 1700 genes are assigned to one or more of 131 biological classes. We used hypergeometric distribution to compute a P-value for each pair of cluster-annotation class. Tables 5.2 and 5.3 show the results for only a few highly significant clusters obtained by using gene expression data only and for the combined method. Although the combined method improved functional significance of some clusters, and even identified new clusters, compared to the clusters obtained using expression data alone, this was not the case for all generated clusters. In this case, combining various data sources offered only a limited classification improvement, because of the incompleteness and noisiness of individual sources of information. However, with the availability of ChIP-Seq data access to a complete gene-TF interaction data for all transcription factors is not out of reach in the near future, and the application of combining techniques will be more justified and possibly will result in a more accurate classification and functional clustering of genes.

Table 5.2: Gene annotation enrichment for Clusters based on combined gene expression data and regulatory sequence data

# of genes in cluster	# of genes in class	P-value	Biological class
120	83	$9.53415E-20$	ArcA Targets
126	43	$1.52249E-11$	DNA Replication
79	206	$1.98304E-10$	CRP
91	52	$2.30326E-55$	Chemotaxis
113	114	$3.38413E-12$	FNR Targets
56	56	$9.72066E-15$	RpoS targets
125	63	$4.49837E-24$	Transposon related
91	23	$2.20766E-10$	Flhd

Table 5.3: Gene annotation enrichment for clusters based on gene expression data only

# of genes in cluster	# of genes in class	P-value	Biological class
120	83	$1.27681E-17$	ArcA Targets
147	43	$5.29658E-06$	DNA Replication
63	206	$3.39538E-24$	CRP
71	52	$1.85328E-73$	Chemotaxis
160	114	$3.04891E-12$	FNR Targets
67	56	$4.86136E-12$	RpoS targets
62	63	$1.52646E-26$	Transposon related
71	23	$1.76131E-06$	Flhd

5.3 An integrated model-based clustering method

Statistical methods based on probabilistic modeling constitute another class of technique for combining heterogenous data. Joint likelihood modeling of gene expression and promoter sequence data has been applied for gene clustering and to find transcription factor binding site motifs [63]. The idea was to find a cluster of genes which have similar expression profiles and similar promoter sequences. The underlying assumption for this approach is that a group of co-expressed genes share a common binding site motif for a transcription factor. Without loss of generality, we consider only gene expression and sequence data for later discussion. Therefore, each cluster can be modeled as a mixture of a model for gene expression data and a model for sequence data,

then genes membership is determined by maximizing the joint likelihood of the models. Although, joint modeling of expression data and sequence data has shown significant improvement over clustering based on a single data source [63], there are a few issues which have not been addressed in joint modeling which may further improve the method if taken into account. For example, to our knowledge, the models for sequence data have not considered the cooperativity of the transcription factors, meaning that there is only one model for explaining sequence data of genes in a cluster. This is in contrast with the fact that many genes are regulated by more than one transcription factor and in addition genes with similar functionality may be the targets of different transcription factors, thus they may have different binding sites for these transcription factors.

Second, the process of learning of a model for a sequence motif uses the traditional idea of position weight matrix (PWM), which assumes independent contribution by each nucleotide at each position of the site to the overall binding affinity of the site.

We propose a probabilistic clustering approach by fitting mixture of models to each cluster in which we address the above problems when modeling sequence data. We use the expectation maximization (EM) algorithm with prior knowledge of known transcription factor binding sites to learn the models, estimate model parameters, and to find gene membership, which is treated as the missing data for the EM problem in hand.

Gene expression model

Given the gene expression data matrix, $\mathbf{E}_{M\times N}$, where M and N represent number of genes and experiments respectively, we look for a model $\mathcal{M}_e$ for each cluster, which can explain the expression data of the genes in the cluster. A simple and reasonable model is represented by a multivariate Gaussian density [63];

$$P_r(\overrightarrow{e_g}|\mathcal{M}_e, \overrightarrow{\mu}, \mathbf{C}) = (2\pi|\mathbf{C}|)^{N/2} exp[\frac{-1}{2}(\overrightarrow{e_g} - \overrightarrow{\mu})^T \mathbf{C}^{-1}(\overrightarrow{e_g} - \overrightarrow{\mu})] \qquad (5.5)$$

where $\overrightarrow{e_g}$ is a vector of gene expression measurements for gene g, $\overrightarrow{\mu}$ and $\mathbf{C}$ are the

mean and covariance matrix of the cluster of expression profiles generated by the model $\mathcal{M}_e$. Further we assume that the covariance matrix, $\mathbf{C}$, is fixed and treated as a global parameter, but $\overrightarrow{\mu}$ is an adjustable parameter which has a prior multivariate Gaussian distribution with mean, $\overrightarrow{\nu}$ and covariance matrix Λ;

$$P_r(\overrightarrow{\mu}|\overrightarrow{\nu},\Lambda) = (2\pi|\Lambda|)^{N/2} exp[\frac{-1}{2}(\overrightarrow{\mu} - \overrightarrow{\nu})^T \Lambda^{-1}(\overrightarrow{\mu} - \overrightarrow{\nu})] \tag{5.6}$$

Regulatory sequences model

The major step in the process of transcription is the recognition of a unique family of short segments of DNA by sequence specific DNA binding proteins, called transcription factors (TF's). The rate at which a given gene is transcribed is determined in part by the amount and the activity of various transcription factors bound to the binding sites found adjacent to the genes. Thus, the regulatory region (usually non-coding DNA sequences) of the gene determines how the genes will be expressed. Therefore, our goal is to analyze the regulatory sequence of a group of genes which share similar functionality. We would like to construct a model, which, to some extent, is able to explain gene regulatory sequences. However, this does not actually model the whole regulatory region directly. Instead, we wish to model subsequences of length W, which in turn are transcription binding sites among subgroups of genes.

There are two ways to do sequence modeling. The first method is to use microarray gene expression data and try to find some features in upstream sequences of the same genes in such a way that these features can explain the gene expressions. These features are usually short segments of DNA in upstream region of the genes which may have multiple occurrences or co-occurrences in the promoter region of the genes.

The second method is to check each cluster for the enrichment of a TF's target genes using the list of known TF target genes and then construct the models for chosen transcription factors using the knowledge of binding sites. We will adopt the second method since it is independent of gene expression data, which is our interest.

Let $\mathcal{TF} = \{TF_1, ..., TF_Q\}$ be the set of TFs for which the gene cluster is enriched. Then the task is to learn Q motif models and update it in each iteration of the EM algorithm for mixture modeling. The process of learning the motif model is an independent subject, which was fully explained in Chapter 3.

Let $\mathcal{M}_s$ be the final sequence model for the cluster. Then we define

$$P(S|\mathcal{M}_s) = \frac{1}{Q}\sum_{q=1}^{Q} P(Y_q|\mathcal{M}_q)I(Y_q, TF_q). \tag{5.7}$$

Where $I(Y_q, TF_q)$ ia an indicator function, which is one if Y_q is a putative binding site for transcription factor TF_q and zero otherwise. Y_q is defined as

$$Y_q = \mathrm{argmax}_{\{\mathrm{Y}\in\mathrm{S}:|\mathrm{Y}|=\mathrm{W}\}}\mathrm{P}(\mathrm{Y}|\mathcal{M}_\mathrm{q}), \tag{5.8}$$

where $|Y|$ is the length of segment Y.

Clustering using joint models

We now return to the problem of how to assign genes to clusters using joint gene expression and sequence model. Regarding sequence data and expression data as two independent sources of data the joint likelihood of sequence and expression model is defined as [63]

$$P_r(\mathbf{E}, \mathbf{S}|\mathcal{M}_e, \mathcal{M}_s, \theta) = P_r(\mathbf{E}|\mathcal{M}_e, \Lambda)P_r(\mathbf{S}|\mathcal{M}_s, \Phi), \tag{5.9}$$

where θ represents joint model parameters and Λ, Φ represent expression and sequence model parameters. Let Λ_j, Φ_j be the model parameters of cluster j, for $j = 1, 2, ..., K$, where K is the number of clusters. We introduce an indicator matrix D where $d_{ij} = 1$ if gene i has cluster j membership and $d_{ij} = 0$, otherwise. Then, we use the EM algorithm to estimate the model parameters and missing data matrix D. The EM algorithm iteratively maximizes the expected log likelihood of the joint model parameters over the conditional distribution of missing data D. The log likelihood of

model parameters given joint distribution of data $\mathbf{X} = \{\mathbf{E}, \mathbf{S}\}$ and missing data D is defined as

$$\begin{aligned}\log P_r(\mathbf{X}, D|\theta) &= \log \prod_{i=1}^{M} P(\mathbf{X}_i, D_i|\theta) = \log \prod_{i=1}^{M} P(\mathbf{X}_i|D_i, \theta)P(D_i|\theta) \\ &= \log \prod_{i=1}^{M}\prod_{j=1}^{K} P(\mathbf{X}_i|\theta_j)^{d_{ij}} P(d_{ij}=1|\theta)^{d_{ij}} \\ &= \sum_{i=1}^{M}\sum_{j=1}^{K} d_{ij}(\log P(\mathbf{X}_i|\theta_j) + \log P(d_{ij}=1|\theta)). \end{aligned} \tag{5.10}$$

Taking the expectation of above likelihood over D in each iteration of the EM algorithm, we have

$$E_{D|\mathbf{X},\theta} \log P_r(\mathbf{X}, D|\theta) = \sum_{i=1}^{M}\sum_{j=1}^{K} E(d_{ij}|\mathbf{X}, \theta^{(0)})(\log P(\mathbf{X}_i|\theta_j) + \log P(d_{ij}=1|\theta)) \tag{5.11}$$

Denote $d_{ij}^{(0)} = E(d_{ij}|\mathbf{X}, \theta^{(0)})$, then we have

$$\begin{aligned} d_{ij}^{(0)} &= 1.P(d_{ij}=1|\mathbf{X}_i, \theta^{(0)}) = \frac{P(\mathbf{X}_i|d_{ij}=1, \theta^{(0)})P(d_{ij}=1|\theta^{(0)})}{P(\mathbf{X}_i|\theta^{(0)})} \\ &= \frac{P(\mathbf{X}_i|\theta_j^{(0)})\lambda_j^{(0)}}{\sum_{k=1}^{K} P(\mathbf{X}_i|\theta_k^{(0)})\lambda_k^{(0)}} \end{aligned} \tag{5.12}$$

where $\lambda_k = P(d_{ik}=1|\theta^{(0)})$. Substituting $d_{ij}^{(0)}$ back into (5.11) we have

$$\begin{aligned} E_{D|\mathbf{X},\theta} \log P_r(\mathbf{X}, D|\theta) &= \sum_{i=1}^{M}\sum_{j=1}^{K} d_{ij}^{(0)}(\log P(\mathbf{X}_i|\theta_j) + \log P(d_{ij}=1|\theta)) \\ &= \sum_{i=1}^{M}\sum_{j=1}^{K} d_{ij}^{(0)}(\log P(\mathbf{X}_i|\theta_j) + \sum_{i=1}^{M}\sum_{j=1}^{K} d_{ij}^{(0)}(\log \lambda_j) \end{aligned} \tag{5.13}$$

The M step of EM maximizes (5.13) over θ and λ in order to find the next estimate of these parameters. The maximization over λ only involves the second term in (5.13)

and results in:

$$\lambda_j^{(1)} = \sum_{i=1}^{M} \frac{d_{ij}^{(0)}}{M}, \quad j = 1, \ldots, K$$

The maximization over θ can be carried out by maximizing the first term in (5.13) separately over each θ_j;

$$\theta_j^{(1)} = \arg\max_{\theta_j} \sum_{i=1}^{M} \sum_{j=1}^{K} d_{ij}^{(0)} \log P(\mathbf{X}_i|\theta_j), \quad j = 1, \ldots, K$$

Note that θ_j represents the model parameter of the joint model which include the Gaussian gene expression model parameters and PWMs parameters for the sequence model. Clearly, this algorithm by allowing d_{ij} to accept continuous value rather than zero and one, can produce overlapping clusters as well. In the next section, we applied the present mixture model to combine gene expression and Chip-chip data to identify regulatory modules.

5.4 Identification of regulatory modules

Due to the unavailability of complete sequence motifs and genome wide location data for many transcription factors in *E.coli*, we decided to apply our algorithm on *Saccharomyces cerevisiae*(Yeast) data set. We chose the well-described yeast microarray data set, the gene expression during cell cycle [13], to test the algorithm described above. This data set was downloaded from the Stanford Microarray Database (`http://smd.stanford.edu//`). For the interaction data data we downloaded the genome-wide location data performed by Harbison et al. [64] from their web site (`http://web.wi.mit.edu/young/regulatory_code`). This data set contains information regarding the binding of 204 regulators to their respective target genes in rich medium. The data matrix contains the p-value calculated from the ratios of immunoprecipitated and control DNA for each gene. These p-values were converted to the probability of the binding of each regulator to each gene using the informative prior

model introduced in [65]. These probabilities were in turn used in the sequence model part of the integrated mixture model.

To assess the differences in performance between the integrated model-based algorithm presented in this chapter and previously described algorithms for module detection, we compared our algorithm with some of the well known module detection tools such as ReMoDiscovery [66] and GRAM [67]. Module detection by ReMoDiscovery consists of two steps. In the first step, called seed discovery, stringent seed modules are identified. In this step the algorithm begins by finding gene sets that are co-expressed in microarray data, and that bind the same regulators. In the second step, called the seed extension step, genes are added to the module by relaxing the criteria for their co-expression and their p-values. The GRAM algorithm [67] detects modules by sequential analysis of the gene expression and ChIP-on-chip data. In the first step, the ChIP-chip data is used to group genes whose upstream regions are likely to bind a common set of transcription factors based on their p-values. In the second step, the microarray data is used to select a sub group of genes, whose expression profiles are similar to each other. Finally, the resulting core set is expanded with additional genes that have a small p-value for the same set of regulators in the ChIP-chip data. The difference between these two algorithms is that the ReMoDiscovery is concurrent in combining the data sets while GRAM is a sequential algorithm.

The run time of algorithm presented in this chapter is an order of magnitude faster than either of the above two algorithms. However, since the number of modules is not known a-priori, one has to run the subject algorithm several times in order to choose the best value for the number of modules. Even with this additional computation, the algorithm presented here remains faster than the above two algorithms due to the combinatorial nature of regulators selection procedure for modules in both of the latter algorithms.

To evaluate the average functional over-representation of the modules detected by

each algorithms we applied our algorithm on the cell cycle data set and compared the summary of the result with the result of GRAM and ReMoDiscovery on the same data set using their default parameters(the result for GRAM and ReMoDiscovery were adopted from [66]). We used functional categories in the MIPS database [68] and used the hypergeometric distribution to calculate a corresponding p-value for the functional enrichment of each module. Several different values for the number of modules were tested, and by comparing the average functional enrichment of the modules for all cases the number of modules equal to 70 was chosen as the best one.

Table 5.4 shows a summary of the statistics for the detected modules using the three algorithms. ReMoDiscovery tends to identify very small numbers of modules with a few number of genes in each module, while GRAM detects too many modules, but with a small number of genes in each module. This results from the fact that these two algorithms put higher weight on interaction data. If genes do not satisfy the p-value threshold for ChIP-chip data, they will not get the membership in any module, even though they might show high co-expression with many genes belonging to a module. However, for ReMoDiscovery, extending the seed module increases the number of genes in each module but still many genes remain unclassified. Unlike these algorithms, the model-based algorithm presented here tends to classify all genes to the best modules based on both data sets. Therefore, if there is no information on binding interaction for a particular gene, that gene is automatically assigned to the module which has the best co-expression with it's members. Because of this feature, the model-based algorithm detects a reasonable number of modules, which are not too small nor too big. In addition, considering the number of modules, modules detected from the integrated model-based algorithm are functionally more enriched based on their average p-values.

Table 5.4: Summary of the results of the integrated model-based algorithm, ReMoDiscovery and GRAM module detection methods.

Method	Number of modules	# of genes in each module			Average functional enrichment p-value
		Min	Mean	Max	
Mixture Model	70	5	78	648	5.00E-03
ReMoDis.(seed modules)	20	2	2.05	3	0.05
ReMoDis.(extended modules)	18	6	67.72	200	2.00E-03
GRAM	274	5	6.80	33	0.02

Chapter 6

Construction of gene transcriptional regulatory networks

One of the goals of systems biology is to elucidate functionally relevant regulatory interactions[69, 70]. Since changes in gene expression are in part determined by such interactions between regulators and their target genes, genome-wide expression data can be effectively used to impute transcriptional regulatory networks. In Chapter 5 we discussed the application of clustering algorithms in discovering modules of functionally related genes. These modules contain genes with similar transcriptional activity across time points or different environmental conditions, and could be enriched for the genes regulated by common transcription factors. However, the utility of clustering techniques is limited in distinguishing between direct and indirect interactions between regulators and their targets. Another drawback of clustering algorithms is that the assignment of a module's genes to transcription factors governing that module is ambiguous.

To overcome these limitations more complex statistical and mathematical approaches are required. These approaches can be categorized in two classes: i) networks based

on Gene-Gene interaction frameworks and ii) networks based on Gene-Regulon interaction frameworks. In the first section of this chapter we provide an overview of methods belonging to the first class, and in the remaining sections we introduce the Gene-Regulon association network model and based on that we propose two mathematical frameworks for constructing gene transcriptional regulatory networks.

6.1 Gene-Gene interaction based methods

The first class of gene regulatory network models, which we refer to as Gene-Gene interaction models, considers the interactions between genes and transcription factors in the expression domain, i.e., the control of regulation is explained by mRNA levels of the transcription factors. Algorithms belonging to this class model a gene regulatory network as a linear and time continuous system, with the transcriptional activity of genes described by a time-continuous dynamical system of first order differential equations [72, 73, 74], or by stochastic dynamical equations, a framework based on a state space model [75] or a Dynamic Bayesian Network [76, 77]. Boolean gene regulatory networks [78, 79, 80] are particular cases of dynamic networks, which assume that the time and states of the system are discrete. Genes are the network nodes, which are in one of two binary on/off states. The state of each node is defined by a boolean function of the previous states of the inputs to that node. These methods often use time series data collected over a small number of time points, compared to the large number of genes, which results in an under-determined problem [81, 82] to solve. The second category of Gene-Gene interaction methods include relevance networks [83, 84, 85], Bayesian networks [86] and graphical Gaussian models (GGM) [87, 88]. These methods impute gene networks by establishing connectivity (edges) between genes based on the dependencies in their expression profiles. The GGMs and relevance networks model conditional independencies and marginal dependencies, respectively, among the gene

pairs. The application of the GGMs is limited to the gene networks with the number of experimental measurements significantly greater than the number of genes. Similarly, the relevance network algorithm, which uses mutual information between genes and treats gene expression levels across different conditions as ensembles of single random variables, can capture the condition-specific activity of the genes only when the sample size is very large [84].

Another approach to inferring Gene-Gene interaction networks is through non-greedy decomposition of gene expression data matrices to uncover hidden, often overlapping, regulatory signals and transcriptional connectivity patterns. Since the data does not have to be a time series, one can collect data for as many different experiments as possible and combine them to increase the sample size and prevent the problem of under-determination. Principal component analysis (PCA) [89], singular value decomposition [90, 91, 92], and independent component analysis (ICA)[93] can be used to determine the low-dimensional representation of the data through decomposing the original data into a few regulatory signals which explain most of the data. However, the orthogonality assumption of PCA and the statistical independency assumption of ICA place methodological constraints on biological signals. Network component analysis (NCA) [94] is another matrix decomposition method, devised specifically for gene expression data, which takes advantage of available knowledge about the connectivity pattern of the network. The NCA method and a similar two-stage matrix decomposition model in [95] assume that the connectivity matrix is fully known, and, therefore, it does not predict any new interactions among the genes and transcription factors.

In summary, the above network reconstruction methods that rely on covariance of expression of transcription regulators and their targets ignore the fact that transcription of regulators and their targets can be controlled differently and/or independently. Such oversight would result in many erroneous predictions. To address this problem we introduce a new framework to predict interactions between regulators and genes via

associating the expressions of the target genes to the activity profiles of their regulators.

6.2 Development and application of Gene-Regulon association methods

Accurate prediction of regulatory interactions, which is essential for understanding phenotypic outcomes of genetic and environmental perturbations, depends on the quality of models capturing essential regulatory features and on their underlying assumptions. One such feature is that the transcriptional activity of co-regulated genes should sufficiently absorb in itself the activity of their common regulator. Moreover, the information about transcriptional activity of the known co-regulated genes (a core regulon) should also be sufficient for discovering new target genes, whose transcriptional activity significantly co-vary with the activity of the core regulon members. This is the principal idea behind Gene-Regulon based association methods. Unlike Gene-Gene interaction methods where the mRNA levels of the regulators play the important role, Gene-Regulon methods rely on the activity profiles of the transcription factors, which, in the absence of their direct measurements, are estimated via a computational model. Therefore, to estimate the activity profiles of the regulators the prior information about known interactions between genes and transcription factors is required which can be collected through the literature and genome wide location data. In the following sections, we present three mathematical frameworks which directly or indirectly exploit the activity profiles of transcription factors while predicting new interactions between genes and regulators.

6.2.1 Two-steps matrix decomposition and sparse regression algorithm

We consider the gene regulation process as an input-output model where the transcription rate (output product) is controlled through the activity level of the group of specialized transcription factors. We assume, given the activity of regulators, that this process is approximately linear, although the dynamic behavior of this biological process can be much more complex.

Given a matrix $E_{n\times m}$ of gene expression measurements of n genes across m different experimental conditions and the partial knowledge of the network connectivity, the goal is to fully reconstruct the hidden structure of the network and quantify the interaction coefficients, strength of edges in Figure 6.1.

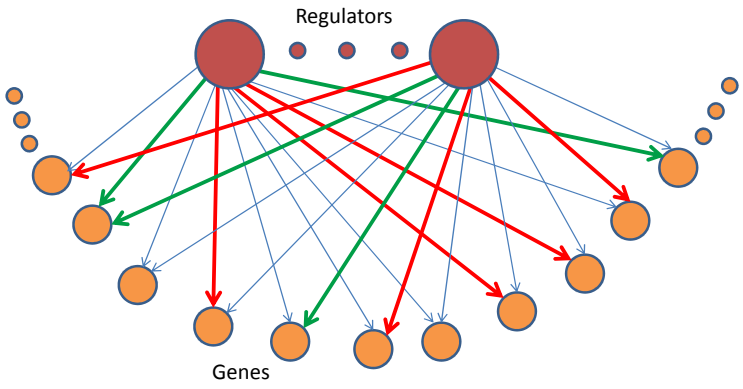

Figure 6.1: Directed bipartite graph representation of the transcriptional regulatory network. Green edges are true known interactions and red edges are true unknown interactions which we wish to predict computationally.

One way to solve this problem is to decompose data matrix E to a connectivity matrix $C_{n\times k}$ and a matrix $S_{k\times m}$ containing activity profiles of k transcription factors, *i.e.*, $E = CS$. If the connectivity matrix C is completely known then the network component analysis(NCA) method is able to decompose E to C and S uniquely up to a diagonal scaling factor [94]. However, for many biological systems there is no such complete information about connectivity pattern. But the connectivity matrix C can be partially formed using genome wide location data and the knowledge of known transcription factors and their known targets. This means that there are some known connections in the network, some connections with certain probabilities and also unknown connections due to lack of enough data. We approach this problem in two steps. In the first step, the activity profiles of the regulators can be estimated via decomposition of the gene expression data matrix, and in the second step new regulatory interactions are identified by solving a series of linear sparse regression problems.

Constrained matrix decomposition

Let $n >> m$, which is usually true for gene expression data, and $m > k$, we assume that we have enough number of measurements for each gene. Because we do not have a temporal constraint on gene expression data, one can combine as many experiments available to make m enough large. Let us split gene expression matrix E in $E1$ containing data corresponding to the completely known part of network and $E2$, the rest of data. We assume $C1$ to represent the known part of the network satisfying the NCA criteria [94], that is the rank of $C1$ is k and that the reduced matrices from $C1$, by removing each column (regulatory node) and rows having non zero value on that column (output nodes of removed regulatory node), also are full rank. This means each column of C must at least have $k-1$ zeros. This assumption can be satisfied by selecting C1 from the whole part of the known network. To quantify the connectivity elements and

regulatory signals, S, one may solve the following optimization problem:

$$\min_{C1\in\mathcal{C}(n1,k,p),\, S\in\mathcal{C}(k,m,q)} \|E1 - C1S\|_F^2, \tag{6.1}$$

where $\mathcal{C}(n1, k, p)$ is a linear subspace of $\mathbb{R}^{n1\times k}$ by setting p specific entries of matrices in $\mathbb{R}^{n1\times k}$ equal to zero, and $\mathcal{C}(k, m, q)$ is a linear subspace of $\mathbb{R}^{+k\times m}$ by setting q specific entries of matrices in $\mathbb{R}^{k\times m}$ equal to zero. In the following we assume we do not have any zero constrains on S, $q = 0$, therefore S can be any $k \times m$ matrix.

The standard approach to finding the minimum of the above problem is by applying iterative optimization algorithms corresponding to two least square problems as follows. Let

$$f(E1, C1, S) := ||E1 - C1S||_F^2. \tag{6.2}$$

Suppose one has an approximation $(C1_l, S_l)$ to the minimal point $(\hat{C}1, \hat{S})$ of above problem. Then $(C1_{l+1}, S_{l+1})$ are determined in two steps:

$$f(E1, C1_{l+1}, S_l) = \min_{C1\in\mathcal{C}(n1,k,p)} f(E1, C1, S_l), \tag{6.3}$$

$$f(E1, C1_{l+1}, S_{l+1}) = \min_{S\in\mathcal{C}(k,m)} f(E1, C1_{l+1}, S). \tag{6.4}$$

The above iterative algorithm is in the spirit of the implementation of NCA in [94]. In this section we also choose the solution to above problems to quantify elements of known connectivity matrix $C1$ and the regulatory signal matrix S. Since the number of genes is much higher than the number of regulatory signals, we assume the matrix of regulatory signals $\hat{S}$ estimated in the first step is a good approximation to S. Therefore, we can use $\hat{S}$ and the observation data corresponding to the unknown part of the network, $E2$, to discover and quantify the unknown part of the connectivity matrix,$C2$.

Sparse linear regression

Given $\mathbf{S}$ a $k \times m$ matrix, the activity profiles of the transcription factors, and $\mathbf{e}$, a vector of gene expression measurements across m conditions, we would like to find a sparse

vector, $\mathbf{c}$, which represents the interactions between the gene and the regulatory nodes. Based on the linear model assumption, we can estimate coefficients of $\mathbf{c}$ by solving the following linear regression model;

$$\mathbf{e} = \mathbf{S}^T\mathbf{c} + \epsilon, \tag{6.5}$$

where $\mathbf{e}$ represents a vector of gene expression measurements across m conditions, $\mathbf{S}$ is a $k \times m$ matrix of the k regulators' activities and $\mathbf{c}$ is a sparse vector that represents the interactions between the gene and k regulators. We assume ϵ is an m elements vector of random noise with zero mean and covariance matrix of $\sigma^2\mathbf{I}$.

When there are no constraints on $\mathbf{c}$ and with no prior information, $\mathbf{c}$ can be estimated using well known least square method, which gives the same solution as that of maximum likelihood (ML) when the noise is zero mean, white Gaussian [96]. However, when there is a prior information on $\mathbf{c}$, then the solution is given by minimum-mean square error estimator (MMSE)[96]. The fact that the expression level of the genes is controlled by a few regulatory nodes makes the resulting network and, therefore $\mathbf{c}$, sparse. That is many elements of $\mathbf{c}$ are zero. When $\mathbf{c}$ is sparse then the above solution to the linear regression problem may not be the optimum solution. Algorithms based on Bayesian maximum *a posteriori* (MAP) estimation [96] and Lasso [97] or l_1 norm regularized least square approximation (i.e. $\text{minimize} \quad \|\mathbf{e} - \mathbf{S}^{\mathrm{T}}\mathbf{c}\|_2^2 + \lambda\|\mathbf{c}\|_1$) are among some of the methods that have been proposed for sparse linear regression. For example, A genome wide network inference of whole genome of yeast has been considered in [98] using L^2-norm minimization of the coefficients and setting all elements below certain threshold equal to zero. In [99] the authors consider solving the large scale network reconstruction problem using sparse representation by applying the Lasso algorithm, which uses l_1-norm minimization.

For gene network when the sparsity or the maximum number of nonzero elements is known, instead of the l_1 norm regularized least square approximation one can find more accurate solution by solving the regressor selection problems. Let $q < k$ be the

maximum cardinality of $\mathbf{c}$ then the above regression problem can be reformulated as

$$\begin{aligned} \min_{\mathbf{c}} \quad & \|\mathbf{e} - \mathbf{S}^T\mathbf{c}\|_2^2 \\ \text{subject to} \quad & \text{card}(\mathbf{c}) \leq \text{q} \end{aligned} \tag{6.6}$$

The straight forward method to solve this problem is to check for all possible cases with q nonzero elements in $\mathbf{c}$. However, in general when k is large one has to solve the least square problem for many different combinations of sparsity patterns which is a hard combinatorial problem. A heuristic approach is to solve the l_1 norm regularized least square approximation for different values of λ and choose the one with the smallest λ which satisfies the cardinality constraint. Then, with a given sparsity pattern, one can refine the solution by solving the least square approximation. Although, the heuristic approach can find the global optimum, one might have to solve the l_1 norm regularized least square approximation many times before reaching the optimum solution.

In many cases, specially when dealing with noisy gene expression data, the coefficient values of the regulatory interactions are biologically irrelevant, and one is interested, instead, in constructing the network structure more precisely rather than quantifying the interactions. In this situation our sparse linear regression problem becomes a model selection problem, i.e., the task is to find the set of regulators which best explain the gene expression data. To this end, in the next section we propose a covariance model selection algorithm as an alternative to the above method to estimate the activity profiles of the regulators and to identify the regulatory interactions.

6.2.2 Covariance model selection method

Let us revisit Equation (6.5) while considering it only for the known part of the interaction network. Let Ω_f be a set consisting of the expression vectors of genes known to be controlled by a single regulator f. Then,

$$\mathbf{e}_g = c_f\mathbf{s}_f + \epsilon \quad \forall\, \mathbf{e}_g \in \Omega_f \tag{6.7}$$

where $\mathbf{s}_f$, the activity profile of the regulator f, is unknown and c_f is the interaction coefficient, which is assumed to be a random variable with distribution $\mathcal{N}(\upsilon, \gamma^2)$. The likelihood function for $\mathbf{s}_f$ [96] is,

$$p(\mathbf{e}_g|\mathbf{s}_f) = \frac{1}{\sqrt{2\pi}^m |\Sigma_f|^{1/2}} \exp(-\frac{1}{2}(\mathbf{e}_g - \mu_f)^T \Sigma_f^{-1}(\mathbf{e}_g - \mu_f)) \quad \forall\, \mathbf{e}_g \in \Omega_f \tag{6.8}$$

where

$$\Sigma_f = \gamma^2 \mathbf{s}_f \mathbf{s}_f^T + \sigma^2 \mathbf{I}, \quad \text{and} \quad \mu_f = \upsilon \mathbf{s}_f.$$

Among the set of all regulators, the correct regulator, $\mathbf{s}_f$, maximizes $p(\mathbf{e}_g|\mathbf{s}_f)$. On the other hand, to compute $p(\mathbf{e}_g|\mathbf{s}_f)$ the regulatory signals need not be known explicitly, and the knowledge of the sample mean (μ_f) and the covariance matrix (Σ_f) corresponding to each set Ω_f is sufficient for model selection. Therefore, one only needs to estimate these sample mean and covariance matrices from the data. The covariance matrix Σ_f, which is the covariance matrix of gene expression data in Ω_f, can be estimated using sample covariance matrix. Assume $\mathbf{E}_f$ to be the expression measurements of genes in Ω_f, then one can estimate the weighted covariance matrix corresponding to regulator f by,

$$\hat{\Sigma}_f = \text{sample cov}(E_f), \quad \hat{\Sigma}_{ij} = \frac{\sum_{k=1}^{N_f} w_k (e_{ik} - \mu_i)(e_{jk} - \mu_j)}{1 - \sum_{k=1}^{N_f} w_k^2} \tag{6.9}$$

where N_f is the number of samples in Ω_f, μ_i and μ_j are ith and jth entries of sample mean, μ_f, and w_k's are weights which sum to one.

Since we considered Ω_f's as sets of genes controlled only by one regulator, if $\mathbf{s}_f$ represents the true model, then the covariance matrix Σ_f can be represented by its first rank approximation using eigenvalue decomposition. Therefore, ignoring the noise terms, the inverse of the covariance matrix can be efficiently computed through its first rank approximation, which not only speeds up the algorithm but also reduces the effects of the noise.

$$\Sigma_f^{-1} = \frac{1}{\lambda_1} \mathbf{u}_1 \mathbf{u}_1^T, \tag{6.10}$$

where λ_1 and $\mathbf{u}_1$ are respectively the principal eigenvalue and the principal eigenvector of the covariance matrix. Given data sets of known TF-gene interactions one can form the sets of Ω_f's and use equations 6.9,6.10 and 6.8, respectively, to compute $p(\mathbf{e}_g|\mathbf{s}_f)$ for all TFs and assign to each gene g the regulator f that provides the maximum value.

The knowledge of regulator-gene interactions represents a low-resolution view of molecular interactions inside a cell. It does not provide any details about how and when these interactions occur. Therefore, as a complement to this information, in some biological studies, the question might be which regulators and how they respond under different environmental or genetic perturbations. One way to tackle this problem is to study the activity levels of different transcription factors across different conditions. The principal eigenvector (eigenvector corresponding to the largest eigenvalue) of the sample covariance matrix (Σ_f) can be viewed as the activity profile of a regulator f. Notice that the activity profiles of regulators are principal eigenvectors of different covariance matrices, which necessarily need not be, and,indeed they are not, orthogonal to each other. Therefore, the estimated activity profiles are different from those estimated by algorithms such as Principal component analysis (PCA) [89] or singular value decomposition [90, 91] and independent component analysis (ICA)[93], which decompose the original data into a few regulatory signals that are orthogonal or independent.

Results and Discussions

Data sets

Microarray gene expression data for more than 100 arrays representing 46 biologically distinct conditions have been used to reconstruct the underlying large scale transcriptional regulatory network of *E. coli*. The conditions covered a spectrum of environmental and genetic perturbations and drug treatments. The environmental perturbations, in addition to those described in [100] (Data set is available at NCBI Gene Expression Omnibus (GEO) with accession number: GSE4357-GSE4380), included different

amino acid and nucleotide additions and limitations (NCBI GEO Series accession number: GSE15409). After filtering the collected expression data by a series of different criteria (removing genes with low variance expressions across conditions, with small absolute values and with low entropy profiles across conditions), expression measurements of 3658 genes were used in this study.

A second data set used in this study was published in [84] and was obtained from Many Microbe Microarray database (M^{3D}) web site (`http://m3d.bu.edu`). This set contained expression levels of *E. coli* genes across 524 arrays which resolved into 189 different experimental conditions. It should be noted here that not only do these two data sets cover very different genetic or environmental perturbations, but they also have been collected on two different microarray platforms: cDNA microarrays and Affymatrix Genechips.

The connectivity matrix obtained from RegulonDB `http://regulondb.ccg.unam.mx`[101]. This connectivity matrix contained the connectivity pattern between 1210 genes (we only considered genes having expression measurements in our data set) and 137 transcription factors having more than two known targets.

Algorithm implementation

For each transcription factor, the core regulon, consisting of the set of known targets, is identified from the known connectivity matrix described above. This information is used to learn a covariance model for each transcription factor. The covariance matrices are estimated from the expression data of genes assigned to regulons. The gene expression measurements of the group of K genes controlled by a TF are treated as K realizations of an independent random variable with the same distribution and, therefore, the weighted sample covariance matrix estimation method was applied to approximate the TF's model covariance matrix. Higher weight is given to those gene samples that

are exclusively controlled by the TF. The bootstrap procedure is incorporated to increase the estimation accuracy of covariance model for those TFs with the low number of known targets. Then the algorithm computes the Gene-Regulon association score for each gene-regulon pair using the likelihood function defined in the Method Section. Finally, the activity profiles of transcription factors are estimated using the eigenvalue decomposition procedure. The regulon-based methodology assumes that the expressions of target genes vary with the activity of their regulator, which does not have to be determined solely by its transcript levels but can be a combination of latent factors including abundance, modification status, interaction with low molecular weight effectors or other proteins.

Performance Comparison

To assess the relative performance of our algorithm, we compared our algorithm with the relevance network method developed in [84], and with GGM method presented in [88].

Since we were interested in transcriptional regulatory interactions (interactions between transcription factors and their target genes), we built a network by comparing scores for all possible pairs of transcription factor-gene targets. To make a fair comparison, in each method for each gene we ranked the regulators based on their association scores with that gene. A regulator which has the maximum association score with a gene was assigned to that gene. The second regulator was assigned to the gene if the corresponding association score was greater than the minimum of the association scores of the genes assigned to that regulator in the first round. This procedure was repeated and assignments were made, if warranted by the association score, for the lower ranking regulators as well. This procedure is different from those that use a global threshold parameter to select the edges. Assignment of regulators based on a global threshold for all TFs results in a very limited number of predictions, although with a good precision.

Due to the large scale of the data, it is reasonable to assume that there should be at least one regulator that can explain the gene expression data of each gene. Although, this association may not be discovered through the dependency between the expression level of the gene and the mRNA level of a gene coding for the transcription factor, it may be discovered using gene-regulon based association, and this is the power of Gene-Regulon based association methods.

We compared the prediction results of these three algorithms over the set of known interactions in RegulonDB database [101], which was compiled to a binary matrix of interactions between 1210 genes and 137 transcription factors. Because this data set is incomplete and there are no negative controls, the appropriate measures to compare the performance of the algorithms is recall and precision.

For the relevance network and GGM, we define recall as a fraction of 1210 genes (genes with known interactions) successfully associated with their cognate regulators. The precision is defined as a fraction of predicted interactions which are correct. However, because our algorithm uses the set of known interactions as a training data to estimate transcription factor activity profiles and covariance matrices, we split this data into a training and a test set in order to properly measure the performance of our algorithm. We only used the training part to estimate the covariance matrices in order to predict the transcription factor-gene interactions. The test data was only used to calculate the recall and precision. We repeated this process 100 times and in each run the test set included 100 genes randomly selected from 1210 genes with known interactions and the training set contains the remaining 1110 genes. The final recall and precision were calculated by averaging over 100 recalls and precisions.

The results for two data sets and three algorithms is presented in Table 6.1. Our algorithm, which takes advantage of gene-regulon associations rather than the gene-TF associations, outperforms the other two algorithms. The improvement was at least in part due to the fact that the regulatory outcome is a result of the activity of transcription

factors and not necessarily of their levels, and the knowledge of transcript levels of the regulators' genes is not sufficient to predict the interactions. Both the GGM and the relevance networks construct the relationship between the gene target expression levels and the expression levels of genes encoding transcription factors. Such models are confined to the cases when regulators are members of their own regulons. However, in many cases, if not most, transcriptional regulation of targets is not accompanied by changes in the levels of the regulators' transcripts, and even when it is, the changes don't have to be correlated with the transcription of the targets. Instead, such regulation can be captured when estimating the activity of the transcription factors, which is the basis of the method presented in the current study.

The second reason for outperforming the relevance network algorithm is the capacity of our method to capture condition-specific activity and the co-variance of the gene expression profiles across different conditions. On the other hand, the relevance network algorithm treats gene expression levels across different conditions as ensembles of single random variables to estimate the probability distribution for each gene, and therefore it can not account for condition-specific activity of the genes when the sample size is not large enough. The relevance network performed better than the GGM and it worked better for the larger data set, the Affymetrix data set. This improvement was likely due to a higher number of experimental conditions(arrays) in the second data set, which in turn resulted in a more accurate calculation of mutual information. The performance of the GGM over these two sets was very poor, which further confirms that GGMs are not suitable when dealing with large-scale gene networks.

It is worth mentioning that our algorithm is also faster than the relevance network and GGM. The expensive part of our algorithm is eigenvalue decomposition of covariance matrices, which has complexity of $O(m^3)$, where m is the number of conditions, which is much smaller than the number of genes. On the other hand, the relevance network involves estimation of marginal probability distributions of genes and joint

Table 6.1: Comparison of recall(Precision) (%),rounded to the closest integer, for the model selection algorithm, relevance network and graphical gaussian model on two large-scale microarray data sets.

	Methods		
Data sets	Covariance Model	Relevance Network	GGM
Our data set	44 (43)	8 (6)	3(3)
Data set in [84]	62 (64)	20 (16)	3(2)

probability distributions of all gene pairs to calculate pair-wise mutual information. This process is time-consuming when the number of genes is large. The expensive part of the GGM is the calculation of a partial correlation matrix, which is based on calculating the inverse or pseudo-inverse of a large matrix ($n \times n$), where n is the number of genes. The computational complexity of pseudo-inverse is $O(n^3)$, which can be very time-consuming when n is large, such as in large-scale gene networks.

Network reconstruction

We applied the proposed method to a large-scale gene expression data set to evaluate its capacity to recover known regulatory interactions and predict new ones. We ignored any TF assignments with a very small number (< 5) of known interactions. We assigned to each gene a regulator based on the maximum probability of association calculated using the proposed algorithm presented in the methods section. When we limited the number of associations for each gene to one, the algorithm was able to recover the correct association for 86% of the genes with known interaction, i.e., 1044 genes out of 1210 genes were associated with their known regulators. When two regulators were assigned to genes, an additional 542 known interactions were recovered, which included known interactions for 26 genes that were not in the set of 1044. That increased the percentage of genes with correct associations to 88%, and this implies that for 74% of genes, i.e. for 516 out of 696 genes having two or more regulators, both

Table 6.2: New targets of Lrp which were confirmed using qPCR (the fold enrichment values with '*' are from ChIP-chip)

Gene Name	Fold transcript change	Fold IP enrichment	Lrp Activity	Function
ompT	8.5	2.8	Positive	DLP12 prophage; outer membrane protease VII
eco	1.8	2.3	Negative	ecotin, serine protease inhibitor
dppA	1.6	3.4	Positive	dipeptide transporter
pntA	1.6	2.8	Positive	pyridine nucleotide transhydrogenase
artP	1.7	4	Negative	arginine periplasmic transport system
sdaC	-	1.9	Positive	predicted serine transporter
yhjE	1.5	6.9	Negative	putative transporter
csiE	-	2.1*	Negative	stationary phase inducible protein
ygdH	-	2*	Positive	unknown

predicted regulators were correct.

In addition to recovering known interactions, the algorithm discovered new, unannotated interactions. Some of the discovered interactions could be independently confirmed. For example, the algorithm predicted two targets of ArgR, *hisJ* and *artJ*, which were not among known interactions in the RegulonDB database, but have been recently reported in the literature [102]. For this particular TF, 21 out of 27 known interactions in regulonDB were recovered.

We specifically focused on the structure of the Lrp regulon. Lrp is a global transcription factor and a mediator of leucine response. It is believed that Lrp controls the expression of hundreds of genes directly or indirectly[42], although only 55 known Lrp targets were annotated in the RegulonDB at the time of this study. Overall, 85 genes were predicted to be Lrp targets. By using transcriptional data obtained on an Lrp knock-out mutant [42], we confirmed that 52 genes were differentially expressed in the knock-out strain. Using chromatin immuno-precipitation(IP), we found that Lrp binds to the upstream regions of at least 45 out of the 52 differentially expressed genes, including several new predicted targets such as *ompT*, *dppA*, *eco*, *pntA*, *pntB*, *csiE*, *sdaB sdaC*, *yhjE* and *ygdH*, most of which were also verified using a qPCR experiment

(Table 6.2). Thus, the algorithm discovered 10 *new* targets of the transcription factor Lrp that could be confirmed by a biological experiment. In addition, limited evidence in the literature suggests that *sdaBC* and *pntAB* are also likely Lrp targets [103] [104]. *pntAB* is also among predicted targets for lrp in [84]. Our results also indicated that Lrp controls expression of the *leuABC* operon. Although according to RegulonDB there is no interaction between Lrp and the *leuABC* operon, it has been previously reported that Lrp does regulate this operon [103]. Overall, this rate of discovery of true Lrp targets is higher than any previously reported discovery rate of confirmed targets for any regulator by a discovery algorithm.

Network refinement

Based on the two expression data sets, covering regulatory states for almost all genes in the genome, each gene (operon) was assigned a transcription factor(s) that was likely to control the expression of that gene. To this end, the present algorithm was applied to both data sets. For each data set, every gene was assigned three top-ranking regulators with the highest probabilities of association, calculated using equation (6.8). If a gene had the same regulator(s) predicted from both sets, the regulatory association between the gene and regulator(s) was deemed true and cataloged for the purpose of network refinement. This resulted in a list of 1719 genes associated with at least one regulator. 779 genes out of 1719 had no previously characterized regulatory interactions. The existence of consensus regulators (predicted using two completely independent data sets) for this number of genes is statistically significant. Only 167 genes would have been expected to have a consensus regulator if the assignments in one of the sets had been done at random.

We incorporated the operon information to further refine the set of regulatory interactions. The consensus regulator for an operon was chosen as a common regulator of genes in the same operon, as predicted on the basis of both data sets. This resulted in

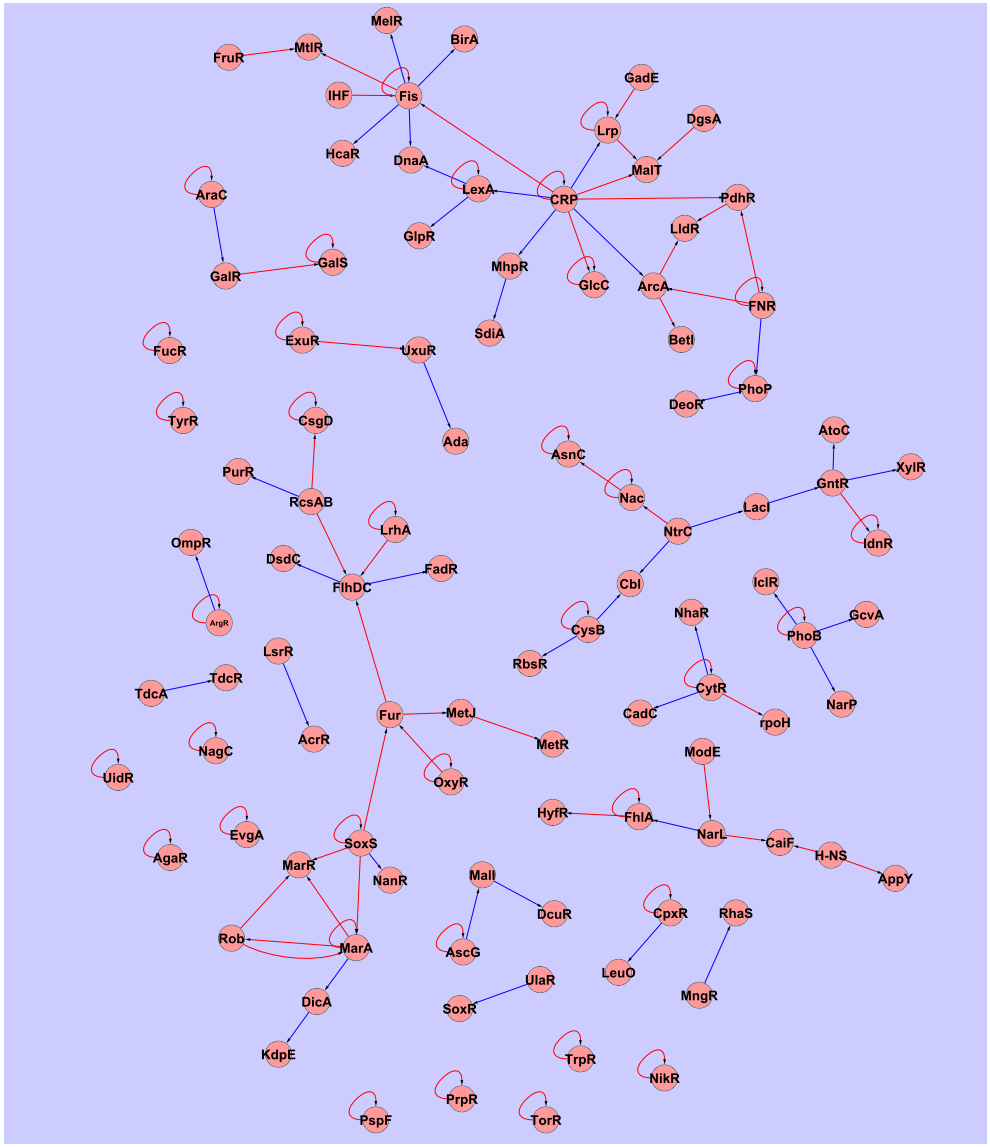

Figure 6.2: Regulatory network of transcription factors. The consensus regulatory interactions predicted using both data sets. This subnetwork comprises of 101 transcription factors (nodes) with 118 predicted interactions (edges) among them. All interactions are directed from a TF-regulator toward a TF-target. 76 (66%) predicted interactions (red edges) were previously known and include 36 known auto-regulators. The remaining 42 predicted interactions (blue edges) are new. In addition, 13 regulators identified as targets did not have any previously identified regulators.

a list of potential regulator(s) for each operon, with many already confirmed or highly plausible regulatory interactions. For example, *dinI* and *dinP* are known targets of LexA, but not among the known interactions set used in this study. Also, yafNOP, *yebB*, *yebG*, *yigN*, and *yjjB-dnaTC-yjjA* were predicted to be LexA targets. *yafNOP* is a neighboring operon of *dinB* with a weak but significant score for a LexA binding site in its promoter region. Regulatory regions of *yebG*,*yigN* and *yebB* contain high scoring LexA binding sites. The possibility that *dnaT* and *dnaC*, two genes involved in DNA replication, are under LexA control is intriguing, and warrants further experimentation.

Overall, even though the two data compendia appeared to be substantially different as far as dominant activity profiles are concerned, transcriptional profiles of as many as 1407 genes and 773 multigene operons could be explained at least in part by the activity of the same regulator(s) in both data sets. This result implies that, provided a sufficiently diverse collection of experimental conditions, the method will converge on true transcriptional regulators of any given gene in a genome, including regulators themselves.

We searched for an intra-regulatory network as a special subnetwork of the complete gene regulatory network. This network contains regulatory interactions between transcription factors. Figure 6.2 depicts such a network constructed from the list of consensus regulatory interactions predicted using both data sets. This subnetwork comprises of 101 transcription factors (nodes) with 118 predicted interactions (edges) among them. All interactions are directed from a TF-regulator toward a TF-target. 76 (66%) predicted interactions (red edges) were previously known and include 36 known auto-regulators. The remaining 42 predicted interactions (blue edges) are new. In addition, 13 regulators identified as targets did not have any previously identified regulators. It remains to be seen whether the described and potential future connectivity refinements affect global or local topological properties of the network.

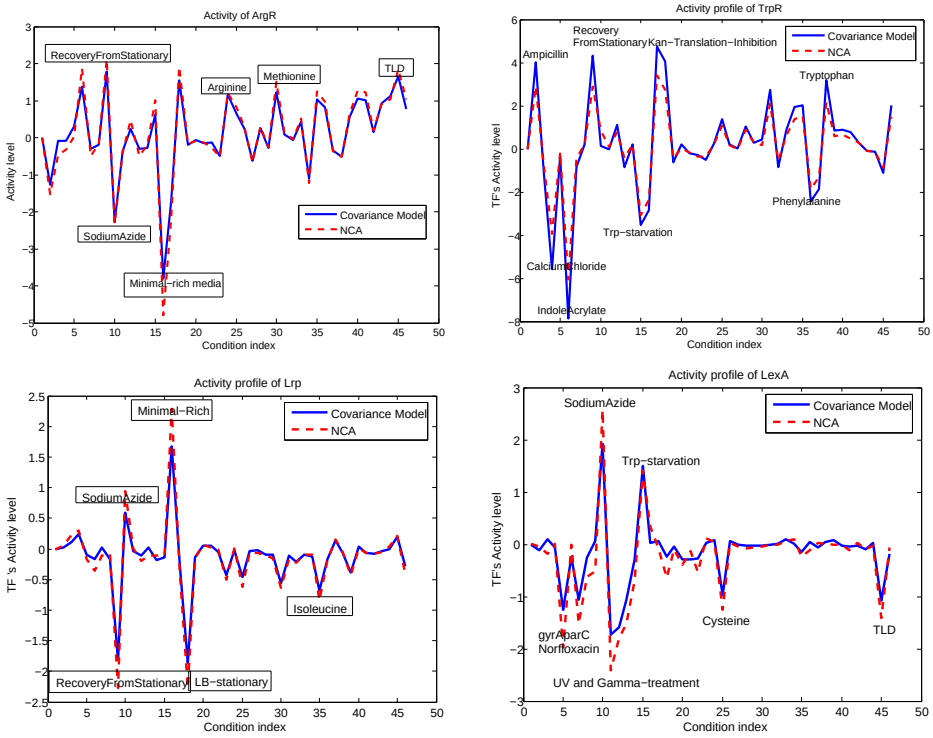

Figure 6.3: Activity profile of ArgR, TrpR, Lrp and LexA. Several conditions in our data set were expected to elicit transcriptional responses mediated by the activity of known regulators. We found that in all conditions with well-studied and understood transcriptional responses, the identity of the most active TF matched our expectations. For example, in an experiment which was conducted to measure transcriptional response to addition of the amino acid arginine, transcription factor ArgR appeared to be the most active TF. Similarly, TrpR was the most active TF in the condition when tryptophan was added to the medium, and LexA was the most active TF under conditions of UV and Gamma treatment.

Activity of regulators

The principal eigenvector of the covariance matrix computed from the set of genes controlled by each regulator was assumed to be a good and biologically sound approximation of the activity profiles of the regulators. We proceeded to evaluate this assumption by examining the estimated activity levels of transcription factors in individual conditions and by comparing the eigenvector-derived profiles with the activity profiles calculated by the Network Component Analysis, a state of the art connectivity matrix decomposition technique proposed in [94]. (Note, the activity levels of TFs are the relative activities of TFs in each condition with respect to a reference sample.)

We determined that the eigenvector-derived profiles of regulators' activity fully recapitulate NCA profiles. Figure 6.3 illustrates this point on several characteristic profiles (some conditions in which the activity level of the TF was significant are indicated).

Several conditions in our data set were expected to elicit transcriptional responses mediated by the activities of known regulators. Indeed, we found that in all conditions with well-studied and understood transcriptional responses, the identity of the most active TF matched our expectations. For example, in the experiment to measure transcriptional response to the addition of the amino acid arginine, transcription factor ArgR appeared to be the most active TF. Similarly, TrpR was the most active TF in the condition when tryptophan was added to the medium, and LexA was the most active TF under conditions of UV and Gamma treatment,6.3.

In almost all conditions, we were able to identify more than one active transcription factor. When we considered a transcription factor to be active at a significance level of 5% (the z-score corresponding to each activity level was calculated from the background distribution estimated from all activity levels), on average 13 TFs were active per condition in our data set (11 - in the Affymetrix data set). The distribution of the

number of active TFs across the conditions is shown in Figure 6.4. The number of transcription factors active in a minimal growth medium as compared to rich medium was the highest, followed by the transition from exponential to stationary phase of growth, during which the cells are known to undergo massive regulatory re-programming, followed by sodium azide treatment, which results among other things in interrupting the electron flow chain. Among the amino acid effects, addition of isoleucine appeared to stimulate the highest number of TFs, whereas addition of threonine or glutamate appeared to have no or very little effect on the regulators. The smallest number of differentially active transcription factors was observed in the comparison of chemostat cultures grown at different dilution rates ("WildTypeGrowth").

Not only did we find that more than one TF appeared to be active in any given condition, but also that many TFs were likely to be mediating transcriptional responses in multiple conditions (Figure 6.5). Given that the conditions in our study were enriched by perturbations of amino acid, nucleotide and DNA metabolism, it was not surprising that the list of most frequently active regulators included ArgR, GcvA, CysB, MetR/MetJ, DeoR, PurR, LexA. What we found surprising was that the transcription factor TrpR, the main transcriptional regulator of genes involved in tryptophan biosynthesis, appeared to be the most responsive regulator in a sense that it was not only among the top responsive TFs to different conditions, but also its activity level was higher than those of other TFs (the method allowed comparison of the activity levels of various regulators across a uniform scale; see the scale on Y axis in Figure 6.3). Tryptophan is the scarcest amino acid in the cell; it is plausible that many perturbations, including those that don't affect tryptophan metabolism directly, may result in biologically significant fluctuations in the size of the amino acid pool. Despite the absence of any apparent bias in the Affymetrix data set [84], the frequency of OxyR activity (OxyR activates hydrogen peroxide induced genes) dwarfed the frequencies of all other factors: OxyR was active in almost thrice as many conditions as the next most

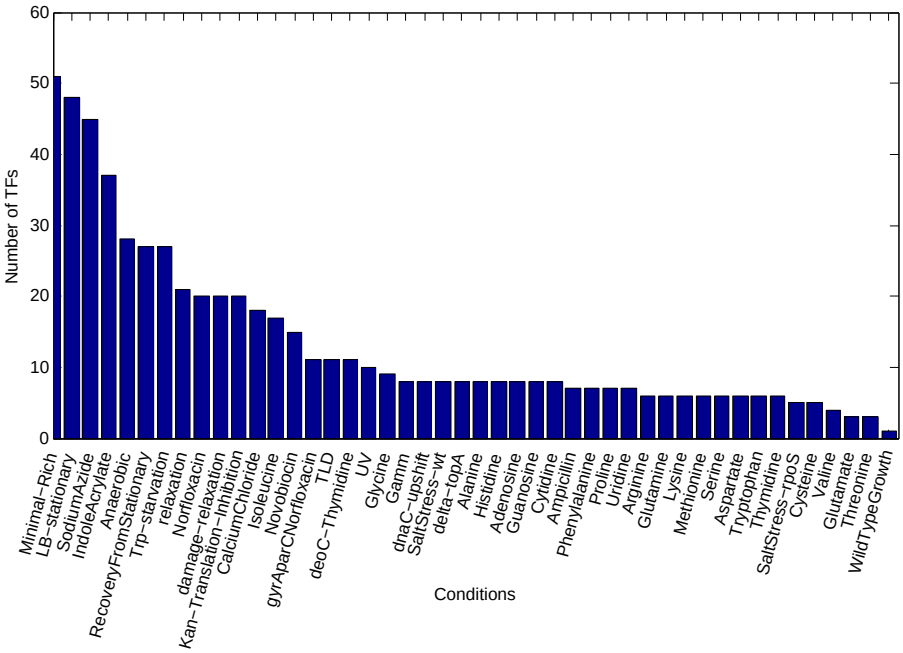

Figure 6.4: A number of active regulators varies across conditions. The number of transcription factors active in minimal growth medium as compared to rich medium was the highest, followed by the transition from exponential to stationary phase of growth, during which the cells are known to undergo massive regulatory re-programming, and then sodium azide treatment, which results, among other things, in an interruption of the electron flow chain. Among the amino acid effects, addition of isoleucine appeared to stimulate the highest number of TFs, whereas addition of threonine or glutamate appeared to have no or very little effect on the regulators. The smallest number of differentially active transcription factors was observed in the comparison of chemostat cultures grown at different dilution rates (”WildTypeGrowth”)

frequently active regulator.

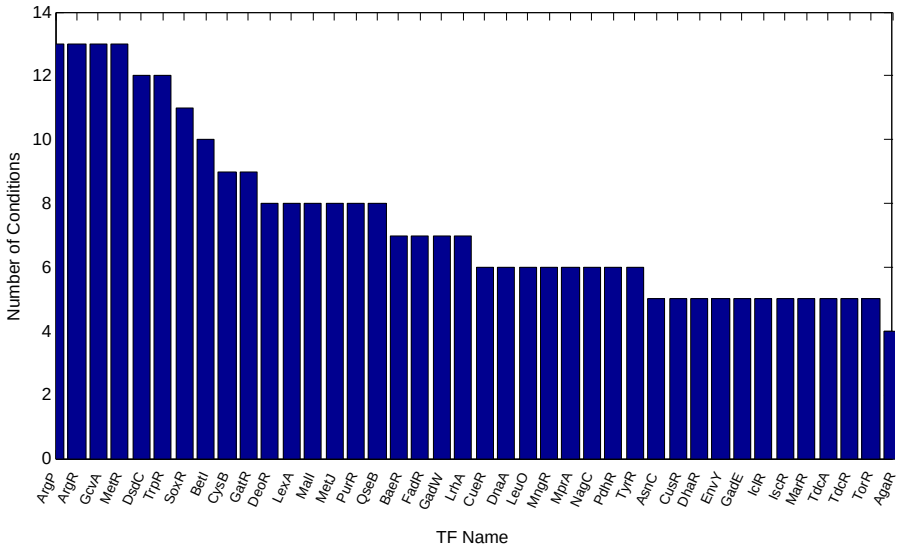

Figure 6.5: Frequency of condition-specific activity for top regulators. Many TFs were likely to be mediating transcriptional responses in multiple conditions. Given that the set of conditions in our study was enriched by those in which metabolism of various amino acids or nucleotides was directly or indirectly perturbed and by conditions causing DNA damage, it was not surprising that the list of most frequently active regulators included ArgR, GcvA, CysB,MetR/MetJ,DeoR, PurR, LexA.

The approximation of TF activities by principal eigenvectors of the covariance matrices is in agreement with the activity profile computed by NCA algorithm (Figure 6.3). However, the former is much faster due to the fact that NCA needs to find activity profiles and to quantify connectivity matrix by iteratively solving many least square problems. If K iterations are required for convergence of the algorithm, then $K(n+m)$ least square problems (each $O(r^3)$, where r is the number of transcription factor)have to be solved, where n is the number of genes and m is the number of conditions in the gene expression data set. Even considering the sparsity of the connectivity matrix the complexity is much higher than eigenvalue decomposition of L small covariance matrices, where L is the number of covariance matrices corresponding to L transcription factors.

The correlation analysis of TFs activity profiles revealed that the activity profiles of almost half of the transcription factors considered in this study are correlated with one another. Some global regulators such as CRP, IHF, FNR were correlated with more than 10 TFs. However, some local regulators, such as OmpR, PhoP, and EnvY, also showed a high degree of correlation with other TFs. Figure 6.6 shows the network view of correlation between transcription factors. The existence of the edge between two TFs indicates that the correlation between their activity profiles is above a threshold value of 0.70. Such similarities may indicate a certain degree of regulatory redundancy, i.e. different regulators controlling subsets of overlapping genes. Indeed, when we examined to what extent the correlations between the profiles are indicative of TFs regulating common genes, we observed that transcription factor pairs with high correlation regulate common genes with higher probability than TF pairs with low correlations. 55% of TF pairs with correlation above 0.70 appeared to have common targets, compared to 20% of TF pairs with correlation less than 0.70.

However several transcription factors, including LexA, GcvA, SoxR, DsdC and FadR, did not show high degree of correlation with other TFs, even though they were relatively responsive in a high number of conditions in the study.

6.2.3 Manifold learning approaches

Another drawback of gene-gene interaction algorithms is their inability to capture the geometric connectivity pattern hidden in the gene expression data. Even those which can capture nonlinear dependencies in the data, e.g. relevance networks using mutual information, still fail to capture the whole structure and hidden geometric patterns in data. On the other hand, while the structure of the small network can be learned using Bayesian methods, these methods fail to predict the structure of the large gene networks including thousands of genes.

Manifold learning [106, 107, 108] and kernel embedding [109, 110] approaches

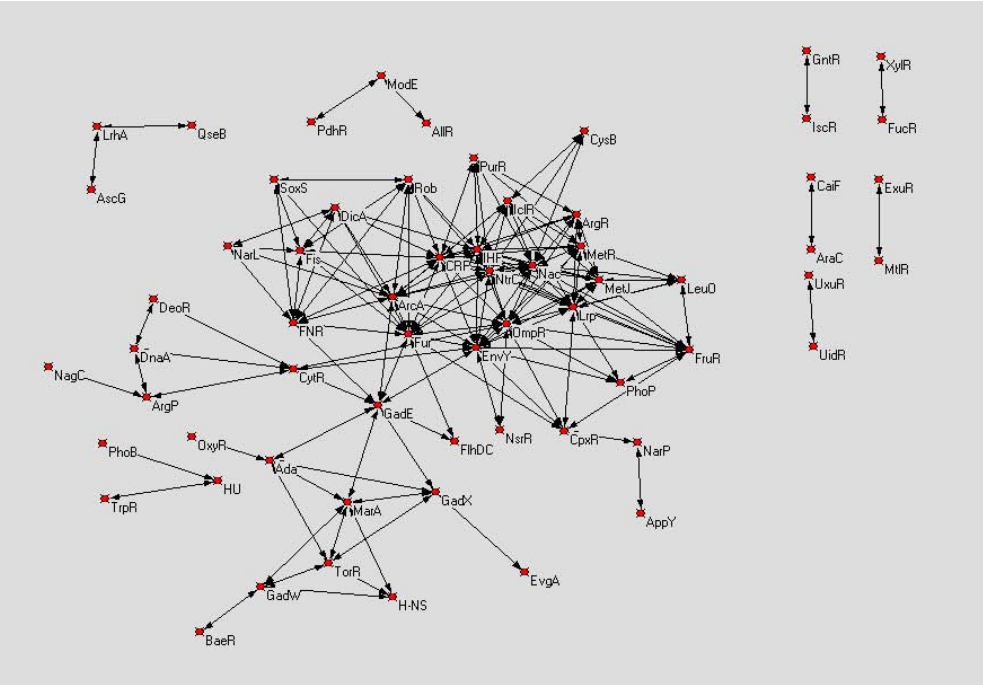

Figure 6.6: Network of transcription factors with correlated profiles. The existence of an edge between two TFs indicates that the correlation between their activity profiles is above a threshold value of 0.70. Such similarities may indicate a certain degree of regulatory redundancy, i.e. different regulators controlling subsets of overlapping genes. Indeed, when we examined to what extent the correlations between the profiles are indicative of TFs regulating common genes, we observed that transcription factor pairs with high correlation regulate common genes with higher probability than TF pairs with low correlations. 55% of TF pairs with correlation above 0.70 appeared to have common targets, compared to 20% of TF pairs with correlation less than 0.70.

provide the mathematical framework for not only nonlinear dimensionality reduction in data but also for capturing the structure and geometric distribution of the data and have already been successfully applied on high dimensional data such as images and motion pictures as well as in several problems in bioinformatics. In Chapter 5 we discussed the application of Kernel approaches as a means for integrating different sources of the data for clustering. This mathematical technique has been also proposed for classification of genes [111, 112]. Recently in [113] the functional distance has been defined via diffusion-type manifold embedding to discover relationships between protein domain functions. These distances were defined between functions in the GO annotations data set to uncover the relationship between the structure and functions.

In this section we aim to provide a framework for the application of the nonlinear manifold learning techniques to reconstruct gene transcriptional regulatory networks. These techniques not only capture nonlinear correlation in data but also are able to capture the underlying connectivity pattern in gene expression data. We define regulon-based association scores between genes and transcription factor core regulon to combine uncovered hidden connectivity patterns in gene expression data with the known connectivity pattern in order to expand the regulatory interaction network of E.coli.

Regulon-Gene association score

Given gene expression data across many diverse conditions, it is reasonable to assume that each regulon is only responsive to a subset of conditions. Therefore, the intuitive and natural approach is to identify the manifold for each regulon where all the data points belonging to the regulon lie on that manifold. Then, one can define the distances between genes and the manifold for each regulon. However, due to the limited number of training points to learn the manifold, the learning problem becomes an ill-posed problem. Since the kernel embedding matrices preserve local similarity, a get around approach is to use these matrices to define a similarity measure between the genes and

the regulons.

We define a group of genes which are regulated by the same transcription factor as a regulon and a subset of this group containing known targets of a transcription factor as a core regulon. Let R be the latent variable representing the regulon. Then, we define the similarity between genes and the regulon as follows.

$$S(g, R) = \frac{1}{|\Omega_R|} \sum_{h \in \Omega_R \, g \neq h} K(g, h), \tag{6.11}$$

where Ω_R is the core regulon for regulon R and $|\Omega_R|$ is the cardinality of Ω_R. K is a kernel embedding matrix. The choice of the kernel embedding method depends on how well it can capture the geometric connectivity pattern in the data by comparing the prediction power of different kernels. In the next section, we describe the construction of the kernel embedding matrix using locally linear embedding algorithm.

Locally linear embedding

The locally linear embedding (LLE) algorithm presented in [106] is a nonlinear dimensionality reduction algorithm, which recovers global nonlinear structure through locally linear fits. It first reconstructs each data point in the original space from its neighbors and assumes the same reconstruction coefficients are valid in the embedding space. Let W be a reconstruction weight matrix in LLE or a normalized local similarity matrix whose ith row sums to unity then a LLE kernel matrix can be defined as follows [110]. Let e be a uniform and unity vector of size N (its elements are $1/\sqrt{N}$), and set

$$M = (I - ee^T)(I - W)^T(I - W)(I - ee^T), \tag{6.12}$$

then an LLE kernel can be formed as

$$K = \lambda_{max} I - M, \tag{6.13}$$

where λ_{max} is the largest eigenvalues of M. Other forms of the kernel embedding matrices such as Isomap kernel, Laplacian kernels [110] and diffusion kernel of powers [109] can also be defined. However, our result on real gene expression data shows the formation of the LLE kernel from local similarity matrix constructed using correlation provides the best result.

LLE kernel from mutual information matrix

Mutual information proximity measure has been extensively used in the analysis of gene expression data. Unlike Pearson correlation, mutual information can capture non-linear correlation in the data. In the context of gene transcriptional regulatory networks, the class of relevance networks [83, 84, 85] has been presented in literature, which rely on computing the mutual information between genes and transcription factor's gene expression levels. However, it is more appropriate to consider the mutual information in the context of gene-regulon. Consider genes $x_1, x_2, ..., x_n$ which are transcriptionally regulated by the same transcription factor to form a regulon set Ω. Then the activity profile of the transcription factor, denoted by the latent variable Z, minimizes the conditional mutual information between the gene variables, and it is given by:

$$Z^* = \arg\min_Z I(x_1, x_2, ..., x_n|Z)$$

Given a regulon, one can estimate Z^* which is, in many cases, different from the gene expression level of the transcription factor. Therefore, the more accurate prediction of gene regulatory interactions can be achieved by assigning genes to the regulon in the following manner. Let,

$$I(g, R) = \frac{1}{|\Omega_R|} \sum_{h \in \Omega_R \, g \neq h} I(g; h).$$

Then the true regulator for a gene g maximizes $I(g, R)$ among all regulons.

Although mutual information captures the nonlinear dependency in the data, it cannot capture the geometric connectivity pattern in data. However, kernel embedding built upon the mutual information similarity matrix can clearly capture this geometric information. The LLE kernel embedding matrix can be constructed from a local similarity matrix (contains only pairwise similarity of genes in the k nearest neighborhoods of each other) derived from the pairwise mutual information matrix.

Results and discussions

To evaluate the application of kernel embedding to the construction of gene regulatory networks, we used the gene expression data and the knowledge of known interactions between genes and transcription factors described in section 6.2.2. We extracted core regulon information for each TF from regulonDB [101]. The most recent data set contains known interactions for 137 transcription factors with at least 3 interactions. The whole set covers 1446 genes with the total of 3213 interactions.

Following the algorithm described above, we first constructed the Pearson linear correlation matrix and pairwise mutual information matrix for each data sets. Then, the local similarity matrices were constructed from these matrices by only keeping the K nearest neighbors of each gene and adjusting other elements to zero, while forcing the matrix to remain symmetric. This guarantees that the obtained local similarity matrix has the proper properties. We compared the prediction accuracy of the proposed method with other potential approaches. We used the set of known interactions and compared different approaches by their ability to recover these interactions. Each method provides the association score between the genes and regulators. For example in the case of the relevance network (CLR network [84]) the scores are CLR scores between genes and TFs. For regulon based approaches the association scores are the similarity scores between genes and core regulon. We assumed that each method should assign at least

Table 6.3: Comparison of Gene-TF and 4 Gene-Regulon based methods. Recall and precision values are in % for two microarray data sets, cDNA data set and Affymatrix data set.

	(Gene-TF) Mutual Information		Gene-Regulon Correlation		Gene-Regulon MI		Gene-Regulon LLE kernel-MI		Gene-Regulon LLE kernel-Corr.	
	Recall	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall	Precision
cDNA	13.5	9.5	26	10.4	24.6	12.2	48.2	36.5	51.5	42.5
Affymatrix	26.8	17.1	38.8	15.9	42.3	21.8	66.9	65	69	67.7

one regulator to each gene with the known interactions. The assignment of the regulators to genes have been made by similar rules to that of section 6.2.2. The recall and precision measures were used to compare different approaches.

Table 6.3 shows the comparison between different approaches for two data sets. Gene-TF based approach is that of CLR version of the relevance network using mutual information. However, unlike in the original work [84] which uses a threshold, therefore giving limited number of predictions, we wanted the algorithm to provide at least one prediction for each gene with known interactions. We also present four different versions of Gene-Regulon based approaches. These approaches are different in how they compute the similarity between the genes and the regulon. The first approach uses the correlation values, the second uses pairwise mutual information values, the third and forth use constructed kernel embedding matrices from mutual information and correlation matrices respectively.

As seen from the table the second data set provides more accurate prediction simply because it covers many more experimental conditions. However, for both data sets across different approaches, the Gene-Regulon based approach using LLE kernel provides the most accurate predictions. Interestingly, the LLE kernel constructed from the correlation matrix performs better than the LLE kernel constructed from the mutual information. We argue that the LLE kernel by itself can capture the nonlinear correlation and the property of mutual information in capturing nonlinear dependency does not add additional merit in this case. On the other hand, the estimation of mutual information

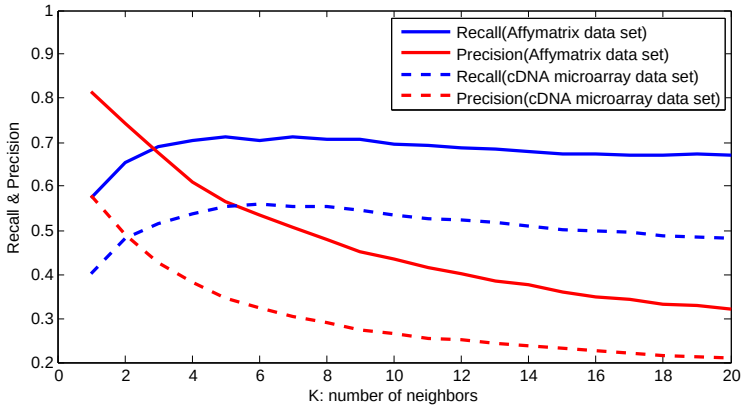

Figure 6.7: The effect of K, the number of selected neighbors, on recall and precision for two data sets.

is more sensitive to the sample size than that of correlation. This is more apparent when one compares the performance of the kernel for the data set with a smaller number of samples. Therefore, we argue that the reason for the better performance of the kernel derived from correlations is that the estimation of the correlations from data is more accurate than the estimation of the mutual information. This results in a more accurate local similarity matrix, which in turn affects the constructed kernel. The recall and precision values for the kernel approaches, in Table 6.3, were obtained using $K = 3, 4$, number of neighbors. The value of K was selected by running the algorithm for $K = 1, ..., 20$. Figure 6.7 depicts the performance comparison with respect to the values of K. The detriment effect of increasing K is more pronounced on precision than recall, which is indicative of increasing the number of predictions with little or no extra correct predictions.

Figure 6.8 provides the comparison between LLE kernel and Laplacian kernel. The Laplacian kernel [110, 108] can be formed from the local similarity matrix W, first by forming the graph Laplacian, $L = D - W$, where D is diagonal matrix called the degree matrix. The diagonal elements of D are sums of the row elements of W. Then the Laplacian kernel is defined by taking pseudo-inverse of the graph Laplacian.

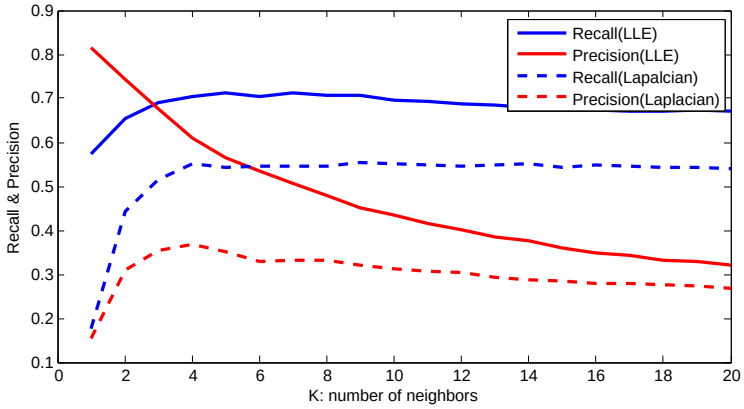

Figure 6.8: Comparison of recall and precision between LLE Kernel and Laplacian Kernel, both constructed from correlation matrix of affymatrix data set.

Though, the LLE Kernel performance is far better than the Laplacian, the Laplacian shows more robust behavior with respect to the number of neighbors, which is due to its property in capturing longer range interactions on the graph as well [113]. However, these longer range interactions derived from local similarity matrix may not be true interactions, and in turn they affect the overall performance of the Laplacian kernel.

We also carried out the following procedure to determine false discovery rate (FDR) for our predictions. To do so, we permuted the gene expression data for each gene across experimental conditions and applied the algorithm on the permuted data. The randomization process and the simulation was repeated 100 times. We only applied this procedure on the data set with the larger sample size and used the LLE kernel constructed from the correlation matrix. We calculated the recall and precision as defined before. Figure 6.9 shows the recall-precision performance for different values of K for randomized data and real data. As seen both recall and precision for randomized data remain less 7% for all values of K. To estimate the FDR we calculated the true positive (TP) value when applying the algorithm to real data and calculated the true positive when applying the algorithm to randomized data and assumed this TP to be false positive (FP). Then we approximated the false discovery rate as $FDR = \frac{FP}{TP}$. To

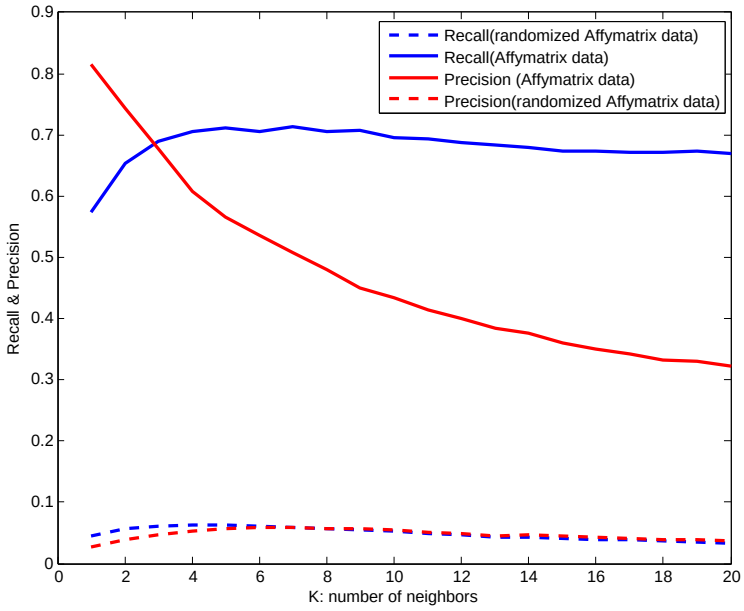

Figure 6.9: Comparison of recall and precision for real Affymatrix data and its randomized version. LLE Kernel constructed from correlation matrix in both case.

be more precise, considering that the total number of predictions in the two cases might be significantly different, we defined $FDR = \frac{Precision_r}{Precision}$, where $Precision$ is precision when using real data, and $Precision_r$ is precision value when using randomized data. This resulted in the FDR value of $\sim 9\%$ when averaging over 100 randomized data sets.

Finally using the ChIP-Sequencing data for *E.coli* transcription factor Lrp, analyzed in Chapter 4, we could verify 33 newly predicted targets for this regulator. In Figure 6.10, these new, along with many known, interactions for Lrp are color classified in a circular layered network graph for Lrp.

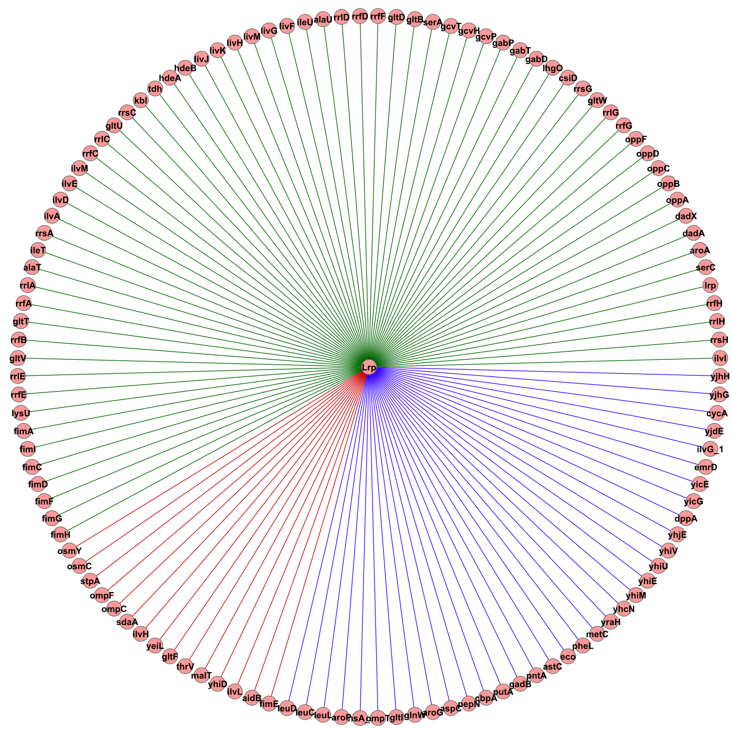

Figure 6.10: Lrp known and predicted targets: Green interactions are known interactions which are predicted, red interactions are known interactions which could not be predicted by our algorithm and blue interactions are the new predicted interactions which were verified using ChIP-Sequencing data.

Chapter 7

Conclusion

The goal of this dissertation was to provide computational frameworks for the analysis of genomic data in order to classify functionally related genes and to identify the interactions between genes and transcriptional regulators. In this chapter, as this dissertation comes to end, we summarize our contributions and provide directions for future research which can complement this study.

7.1 Contributions

In this dissertation, we have made several contributions to the existing literatures for the missing values estimation, pattern matching and identification of binding sites, analysis of Chip-sequencing data, clustering of data from multiple sources and construction of gene regulatory networks. In the data preprocessing step for the analysis of DNA microarray data, we proposed a method based on fixed rank approximation algorithm to estimate missing values on DNA arrays. This algorithm coupled with clustering of genes with similar profiles proved to be fast and accurate in imputing real gene expression data matrices containing missing values.

For the analysis of binding data, we proposed two distinct mathematical frameworks

to identify regulatory regions on the genome from sequences and ChIP-Seq data. We developed an efficient pattern matching algorithm, which accounts for the position-specific dependencies between nucleotides in the sites, to discriminate between the binding sites and non-specific DNA sequences. Also, an algorithm for the analysis of ChIP-Seq data was proposed which unlike existing methods is able to identify enriched region locally in the absence of additional set of background data. Using this algorithm applied on the new generated ChIP-Seq data we were able to identify almost all known and many new targets for transcription factor Lrp in *E.coli*

Information fusion through combining of diverse biological data sets is an important issue that we tried to address throughout this dissertation. We proposed algorithms for handling multiple sources of data in the context of gene clustering and gene regulatory network constructions. We introduced and advocated a new concept of gene-regulon association, in contrast to gene-gene association, for building the gene transcriptional regulatory networks from gene expression data. Using this new framework, we developed supervised model-driven and data-driven algorithms to identify new regulatory interactions in *E.coli*. The proposed model-driven algorithm built upon the assumption that transcriptional activity of co-regulated genes should sufficiently absorb in itself the activity of their common regulators. The nonlinear kernel embedding techniques were adopted in the data-driven algorithm to capture the nonlinear correlation and geometric connectivity patterns in gene expression data. Both algorithms showed significant increases in the precision and recall compared to existing methods while reconstructing *E.coli* known regulatory interactions.

Finally, we should note that this research area is a new application domain, and many computational techniques developed during past years in other fields can be adapted in this filed. Therefore, instead of reinventing the tools with little or no justification, it is better to have a careful strategy in selecting and applying the available

techniques which really can help in new biological discoveries. We hope the computational models presented in this dissertation, in particular the regulatory network models, can produce testable hypotheses that can be validated via biological experiments.

7.2 Directions of future works

The process of discovery in biological sciences is incremental. In some cases new discoveries may undermine the previous one. This is more pronounced when our findings are based on limited observations or data. In this dissertation we demonstrated how gene expression and location data can be combined to construct gene regulatory networks. With the availability of many other sources of data, it is important that the information from data sets such as protein-protein interaction data, textual data and gene annotation data are also considered in the proposed combining methods. We also proposed the application of functional kernels for combining multiple sources of data. These kernels require the availability of numerical data for their construction. On the other hand, the kernel embedding approaches presented in the previous chapter have the ability to form the kernel matrices from similarities between objects, and they can simply be used for categorial data as well as numerical data. This allows integration of, for example, the gene annotation data with gene expression and location data for clustering of functionally related genes. Similarly, kernel embedding matrices can be replaced for pairwise linear correlation matrix in many module detection algorithms such as those described in Chapter 5.

With similar motivation and concurrent to our study, a supervised classification algorithm using kernel functions has been proposed in [114] to complete the transcriptional regulatory network of E-coli using gene expression data. Unlike kernel functions, where only the corresponding data of each pair contribute in computing each element of kernel matrix, all objects affect the entries of kernel embedding matrices. Again, it

would be interesting to see if kernel embedding matrices can improve the classification accuracy when they are used in place of kernel functions.

Finally, an important issue which could not be easily addressed by the computational methods presented in this dissertation was the quantification of the interactions between genes and regulators. The question of how structure of the binding sites can influence regulation of transcription and why targets of a transcription factor respond differently remain yet to be answered. A careful design of new intervening experiments such as designing high throughput promoter-GFP constructs might provide a suite of data to answer these questions.

References

[1] ” http://www.microarray.org”.

[2] ”http://tigr.org/tdb/microarray/”.

[3] D. Schena, R. Shalon, RW Davis, and PO Brown. Quantitative monitoring of gene expression patterns with a complementary dna microarray. *Science*, 270(5235):467-470, 1995.

[4] B. Ren, et al. Genome-wide location and function of DNA-binding proteins. *Science*, 290(5500):2306-2309, 2000.

[5] J. Quackenbush. Microarray data normalization and transformation. *Nature Genetics*, 32:496-501, 2002

[6] S. Oba, et al. A Bayesian missing value estimation method for gene expression profile data. *Bioinformatics* 19:2088-2096, 2003.

[7] S. Friedland, A. Niknejad and L. Chihara. A Simultaneous Reconstruction of Missing Data in DNA Microarrays. *Linear Alg. Appl.* 416:8-28, 2006.

[8] O. Troyanskaya, M. Cantor, G. Sherlock, P. Brown, T. Hastie, R. Tibshirani, D. Botstein and R. Altman. Missing value estimation for DNA microarrays”, Bioinformatics 17, 520-525, (2001)

[9] H. Kim, G.H. Golub, H. Park. Missing value estimation for DNA microarray gene expression data: local least squares imputation. *Bioinformatics* 21:187-198, 2005.

[10] X. Gan, A.W.-C. Liew and H. Yan. Missing Microaaray Data Estimation Based on Projection onto Convex Sets Method. *Proc. 17th International Conference on Pattern Recognition*, 2004.

[11] S. Friedland, J. Nocedal and M. Overton. The formulation and analysis of numerical methods for inverse eigenvalue problems. *SIAM J. Numer. Anal.* 24:634-667, 1987.

[12] M.A. Shipp, et al. Diffuse large B-cell lymphoma outcome prediction by gene-expression profiling and supervised machine learning. *Nat. Med.* 8:68-74, 2002.

[13] P.T. Spellman, et al. Comprehensive identification of cell cycle-regulated genes of the yeast *Saccharomyces cerevisiae* by microarray hybridization. *Mol. Biol. Cell*, 9:3273-3297, 1998.

[14] T.L. Ferea, D. Botstein, P.O. Brown, R.F. Rosenzweig. Systematic changes in gene expression patterns following adaptive evolution in yeast. *Proc Natl Acad Sci U S A* 96(17):9721-9726 1999.

[15] H. Yoshimoto, et al. Genome-wide analysis of gene expression regulated by the calcineurin/Crz1p signaling pathway in Saccharomyces cerevisiae. *J. Biol. Chem.* 277(34):31079-31088, 2002.

[16] O.G. Berg, V. Hippel. Selection of DNA binding sites by regulatory proteins. Statistical-mechanical theory and application to operators and promoters. *Journal of Molecular Biology*, 20(193):723-750, 1987.

[17] D.J. Galas, M. Eggert, M.S. Waterman. Rigorous pattern-recognition methods for DNA sequence:analysis of promoter sequences from Escherichia coli. *Journal of Molecular Biology*, 186:117128, 1985.

[18] J.V. Helden, A.F. Rios, J. Collado-Vides. Discovering regulatory elements in non-coding sequences by analysis of spaced dyads. *Nucleic Acids Research*, 28(8):1808-1818, 2000.

[19] V.H. Jacques. Regulatory Sequence Analysis Tools. *Nucleic Acids Research.*, 31(13):3593-3596, 2003.

[20] F.P. Roth, J.D. Hughes, P.W. Estep, G.M. Church. Finding DNA regulatory motifs within unaligned noncoding sequences clustered by whole-genome mRNA quantitation. *Nat.Biotechnology*, 16:939-945, 1998.

[21] T.L. Bailey, C. Elkan. Fitting a mixture model by expectation maximization to discover motifs in biopolymers. *Proc Int. Conf. Intell. Syst. Mol. Biol.*, 2:28-36, 1994.

[22] A.F. Neuwald, J.S. Liu, C.E. Lawrence. Gibbs motif sampling: detection of bacterial outer membrane protein repeats. *Protein Science*, 4:1618-1632, 1995.

[23] X. Liu, D.L. Brutlag, J.S. Liu. BioProspector: discovering conserved DNA motifs in upstream regulatory regions of co-expressed genes. *Pac. Symp. Biocomputation*, 6:127-138, 2001.

[24] J. Shannan, et al. oPOSSUM: identification of over-represented transcription factor binding sites in co-expressed genes. *Nucleic Acids Research*, 33(10), 2005.

[25] A.M. McGuire, J.D. Hughes, G.M. Church. Conservation of DNA regulatory motifs and discovery of new motifs in microbial genomes. *Genome Research*, 10:744-757, 2000.

[26] G.D. Stormo. DNA binding sites: representation and discovery. *Bioinformatics*, 16:16-23, 2000.

[27] M.L.T. Lee, M.L. Bulyk, G.A. Whitmore, G. M. Church. A statistical model for investigating binding probabilities of DNA nucleotide sequences using microarrays. *Biometrics*, 58:981-988, 2003.

[28] T.K. Man, G.D. Stormo. Non-independence of Mnt repressor-operator interaction determined by a new quantitative multiple fluorescence relative affinity (QuMFRA) assay. *Nucleic Acids Research*, 29(12):2471-2478, 2003.

[29] P.V. Benos, M.L. Bulyk, G.D. Stormo. Additivity in protein-DNA interactions: how good an approximation is it? *Nucleic Acids Research*, 30(20):4442-4451, 2002.

[30] I.G. Nuam, G.D. Stormo, P.I. Ilya. Computational technique for improvement of the position-weight matrices for the DNA/protein binding sites. *Nucleic Acids Research*, 33(7):2290-2301, 2005.

[31] E. A. Kotelnikova, V. Makeev, M.S. Gelfand. Evolution of transcription factor DNA binding sitesf. *Gene*, 347:255-263, 2005.

[32] Y. Barash, G. Elidan, N. Friedman, T. Kaplan. Modeling dependencies in protein-DNA binding sites. *Proceedings of the Seventh Annual International Conference on Computational Molecular Biology*, 28-37, Berlin ,Germany, ACM press, NY, 2003.

[33] I. Ben-Gal. Identification of transcription factor binding sites with variable-order Bayesian networks. *Bioinformatics*, 21:2657-2666, 2005.

[34] O.D. King, F.P. Roth. A non-parametric model for transcription factor binding sites. *Nucleic Acids Research* 31(116), 2003.

[35] J.F. Theis, C.S. Newlon. The ARS309 chromosomal replicator of Saccharomyces cerevisiae depends on an exceptional ARS consensus sequence. *Proc Natl Acad Sci USA*, 94:10786-10791, 1997.

[36] R. Durbin, S. Eddy, A. Krogh, G. Mitchison. *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids.* Cambridg Universiy Press 1998.

[37] K. Ellrott, C. Yang, F.M. Sladek, T. Jiang. Identifying transcription factor binding sites through Markov chain optimization. *Bioinformatics*, 18:100-109, 2002.

[38] Albin S, Wynand A, Par E, Wyeth W, Boris L: JASPAR: an open access database for eukaryotic transcription factor binding profiles *Nucleic Acids Res 2004, 32*;1 Database Issue.

[39] G. Yeo, C. Burge. Maximum entropy modeling of short sequence motifs with applications to RNA splicing signals. *Journal of Computational Biology*, 11:377-394, 2004.

[40] C. Burge, S. Karlin. Prediction of complete gene structures in human genomic DNA. *Journal of Molecular Biology*, 268:78-94, 1997.

[41] K. Robison, A.M. McGuire, G.M. Church. A comprehensive library of DNA-binding site matrices for 55 proteins applied to the complete Escherichia coli K12 genome. *Journal of Molecular Biology*, 284:241-254, 1998.

[42] T.H. Tani, A. Khodursky, R.M. Blumenthal, P.O. Brown, R.G. Matthews. Adaptation to famine: a family of stationary-phase genes revealed by microarray analysis. *Proc Natl Acad Sci U S A*, 99:13471-13476, 2002.

[43] M.B. Adam, C. Sourav, R.C. Nicholas. Prediction of Saccharomyces cerevisiae replication origins. *Genome Biology*, 5(R22), 2004.

[44] V. Orlando. Mapping chromosomal proteins in vivo by formaldehyde-crosslinked-chromatin immunoprecipitation. *Trends in Biochemical Sciences*, 3: 99-104, 2000.

[45] X. S. Liu, D.L. Brutlag, J.S. Liu. An algorithm for finding protein-DNA binding sites with applications to chromatin- immunoprecipitation microarray experiments. *Nature Biotechnology*, 20:835-839, 2002.

[46] M. Zheng, L.O. Barrera, B. Ren, Y.N. Wu. ChIP-chip: data, model, and analysis. *Biometrics*, 63(3):787-796, 2007.

[47] T. Benoukraf, et al. CoCAS: A ChIP-on-chip analysis suite. *Bioinformatics*, 25(7):954-955, 2009.

[48] G. Robertson, et al. Genome-wide profiles of STAT1 DNA association using chromatin immunoprecipitation and massively parallel sequencing. *Nature Methods*, 4:651-657, 2007.

[49] A. Valouev, et al. Genome-wide analysis of transcription factor binding sites based on ChIP-Seq data. *Nature Method*, 5:829-834, 2008.

[50] M.B. Eisen, P.T. Spellman, P.O. Brown, D. Botstein, Cluster analysis and display of genome-wide expression patterns. *Proceeding of the Natoinal Academy of Sciences (PNAS)*, 95(25):14863-14868, 1998.

[51] D. Jiang, C. Tang, A. Zhang. Cluster analysis for gene expression data: a survey. *IEEE Trans. Know. Data Eng.* 16: 13701386 2004.

[52] G. R. G. Lanckriet, M. Deng, N. Cristianini, M. I. Jordan, and W. S. Noble. Kernel- based data fusion and its application to protein function prediction in yeast. *Pacific Symposium on Biocomputing* , 2004.

[53] E. Segal, et al. Module networks: identifying regulatory modules and their condition-specific regulators from gene expression data. *Nature Genetics*, 34:166-176, 2003.

[54] O.G. Troyanskaya, et al. A Bayesian framework for combining heterogeneous data sources for gene function prediction (in *Saccharomyces cerevisiae*) *Proc Natl Acad Sci U S A*, 100(14):8348-8353, 2003.

[55] S.C. Johnson. Hierarchical clustering schemes. *Psychometrika*, 32(3), 1967.

[56] J. A. Hartigan, M. A. Wong. A K-Means Clustering Algorithm. *Applied Statistics*, 28(1):100-108, 1979.

[57] T. Kohonen. The self-organizing maps. *Proc. IEEE*, 78:1464-1479, 1990.

[58] David Tritchler, Shafagh Fallah and Joseph Beyene, A spectral clustering method for microarray data. *Computational Statistics and Data Analysis*, 49:63-76, 2005.

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

[60] G. J. McLachlan, R. W. Bean, D. Peel. A mixture model-based approach to the clustering of microarray expression data. *Bioinformatics*, 18(3), 2002.

[61] S. Kirkpatrick, C. D. Gelatt, M. P. Vecchi. Optimization by Simulated Annealing. *Science*, 2209(4598):671-680, 1983.

[62] ”http://regulondb.ccg.unam.mx/index.html”

[63] I. Holmes, W. J. Bruno. Finding regulatory elements using joint likelihoods for sequence and expression profile data. *Proc Int Conf Intell Syst Mol Biol. (ISMB)*, 8:202-210, 2000.

[64] C.T. Harbison, et al. Transcriptional regulatory code of a eukaryotic genome. *Nature*, 431:99-104, 2004.

[65] A. Bernard, A.J. Hartemink. Informative structure priors: joint learning of dynamic regulatory networks from multiple types of data. *Pac Symp Biocomput.*, 459-470, 2005.

[66] K. Lemmens, et al. Inferring transcriptional modules from ChIP-chip, motif and microarray data. *Genome Biology*, 7(R37), 2006.

[67] Z. Bar-Joseph, et al. Computational discovery of gene modules and regulatory networks. *Nature Biotechnology*, 21:1337-1342, 2003.

[68] H.W. Mewes. et al. MIPS: analysis and annotation of proteins from whole genomes. *Nucleic Acids Research*, 41-44, 2004.

[69] N. Friedman. Inferring Cellular Networks Using Probabilistic Graphical Models. *Science*, 303:799-805, 2004.

[70] M. Levine, E.H. Davidson. Gene regulatory networks for development. *Proceeding of the Natoinal Academy of Sciences (PNAS)*, 102(14):4936-4942, 2005.

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

[72] R.H. Singer, S. Penman. Messenger RNA in HeLa cells: kinetics of formation and decay. *J. Mol. Biol.*, 78(2):321-334, 1973.

[73] H.C. Chen, H.C. Lee, T.Y. Lin, W.H. Li, B.S. Chen. Quantitative characterization of the transcriptional regulatory network in the yeast cell cycle. *Bioinformatics*, 20(12):1914-1927, 2004.

[74] R. Sasik, N. Iranfar, T. Hwa, W.F. Loomis. Extracting transcriptional events from temporal gene expression patterns during Dictyostelium development. *Bioinformatics*, 18(1):61-66, 2002.

[75] R. Yamaguchi, R. Yoshida, S. Imoto, T. Higuchi, S. Miyano. Finding module-based gene networks with state-space models - Mining high-dimensional and short time-course gene expression data. *Signal Processing Magazine, IEEE,* 24(1), 2007.

[76] B.E. Perrin. Gene networks inference using dynamic Bayesian networks. *Bioinformatics*, 19(2):38-48, 2003.

[77] S. Kim, S. Imoto, S. Miyano. Dynamic Bayesian network and nonparametric regression for nonlinear modeling of gene networks from time series gene expression data. *Biosystems*, 75(1-3):57-65, 2004.

[78] T. Akutsu, S. Miyano, S. Kuhara. Inferring qualitative relations in genetic networks and metabolic pathways. *Bioinformatics*, 16:727743, 2000.

[79] I. Shmulevich, E.R. Dougherty, S. Kim, W. Zhang. Probabilistic Boolean networks: a rule-based uncertainty model for gene regulatory networks. *Bioinformatics*, 18(2):261-274, 2002.

[80] X. Zhou, X. Wang, E. R. Dougherty. Construction of genomic networks using mutual-information clustering and reversible-jump Markov-chain-Monte-Carlo predictor design. *Signal Processing*, 83(4):745-761, 2003.

[81] N.S. Holter, A. Maritan, M. Cieplak, N.V. Fedoroff, J.R. Banavar. Dynamic modeling of gene expression data. *Proceeding of the Natoinal Academy of Sciences (PNAS)*, 98(4):16931698, 2001.

[82] M. Hoon, S. Imoto, S. Miyano. Inferring Gene Regulatory Networks from Time-Ordered Gene Expression Data Using Differential Equations. *Lecture Notes in Computer Science*, Springer Berlin/Heidelberg, 283-288, 2002.

[83] A.J. Butte, I.S. Kohane. Mutual infromation relevance networks: Functional genomic clustering using pairwise entropy measuremens. *Pacific Symposium on Biocomputing*, 418-429, 2002.

[84] J.J. Faith, et al. Large-Scale Mapping and Validation of Escherichia coli Transcriptional Regulation from a Compendium of Expression Profiles. *PLOS Biology*, 5(1):54-66, 2007.

[85] A.A. Margolin, et al. ARACNE: An algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context. *BMC Bioinformatics*, 7(1), 2006.

[86] N. Friedman, M. Linial, I. Nachman, D. Pe'er. Using Bayesian networks to analyze expression data. *J. Comput. Biol.*, 7:601620, 2000.

[87] T. Hiroyuki, H. Katsuhisa. Inference of a genetic network by a combined approach of cluster analysis and graphical Gaussian modeling. *Bioinformatics*, 18(2):287-297, 2002.

[88] J. Schafer, K. Strimmer. An empirical Bayes approach to inferring large-scale gene association networks. *Bioinformatics*, 21(6):754-764, 2005.

[89] S. Raychaudhuri, J.M. Stuart, R.B. Altman. Principal components analysis to summarize microarray experiments: application to sporulation time series. *Pac. Symp. Biocomput.*, 455-466, 2000.

[90] O. Alter, P.O. Brown, D. Botstein. Singular value decomposition for genome-wide expression data processing and modeling. *Proceeding of the Natoinal Academy of Sciences (PNAS)*, 97(18):10101-10106, 2004.

[91] N.S. Holter. Fundamental patterns underlying gene expression profiles: simplicity from complexity. *Proceeding of the Natoinal Academy of Sciences (PNAS)*, 97(15):8409-8414, 2000.

[92] O. Alter, G.H. Golub. Integrative analysis of genome-scale data by using pseudoinverse projection predicts novel correlation between DNA replication and RNA transcription. *Proceeding of the Natoinal Academy of Sciences (PNAS)*, 101(47):16577-16582, 2004.

[93] W. Liebermeister. Linear modes of gene expression determined by independent component analysis. *Bioinformatics*, 18(1):51-60, 2002.

[94] J.C. Liao. Network component analysis: reconstruction of regulatory signals in biological systems. *Proceeding of the Natoinal Academy of Sciences (PNAS)*, 100(26):15522-15527, 2003.

[95] H. Li, M. Zhan, Unraveling transcriptional regulatory programs by integrative analysis of microarray and transcription factor binding data. *Bioinformatics*, 24(17):1874-1880, 2008.

[96] S.M. Kay. *Fundamentals of statistical signal processing: Estimation Theory.* Prentice-Hall, Inc. 1993.

[97] R. Tibshirani. Regression shrinkage and selection via the Lasso. *J. Royal Statistical Soc.*, 58(1):267-288, 1996.

[98] T.G. Dewey, D.J. Galas. Dynamic models of gene expression and classification. *Functional Integrative Genomics*, 1(4):269-278, 2001.

[99] M. Gustafsson, M. Hornquist, A. Lombardi. Constructing and analyzing a large-scale gene-to-gene regulatory network Lasso-constrained inference and biological validation. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 2(3):254- 261, 2005.

[100] D. Sangurdekar, F. Srienc, A. Khodursky. Classification based framework for quantitative description of large-scale microarray data. *Genome Biology*, 7(32), 2006.

[101] H. Salgado, et al. RegulonDB (version 5.0): Escherichia coli K-12 transcriptional regulatory network, operon organization, and growth conditions. *Nucleic Acids Research*, 34:394-397, 2006.

[102] M. Caldara, D. Charlier, R. Cunin. The arginine regulon of Escherichia coli: whole-system transcriptome analysis discovers new genes and provides an integrated view of arginine regulation. *Microbiology*, 152:3343-3354, 2006.

[103] Z.Q. Shao, R.T. Lin, E.B. Newman. Sequencing and characterization of the sdaC gene and identification of the sdaCB operon in E. coli K-12. *Eur. J. Biochem.*, 222:901-907, 1994.

[104] R. DAri, R.T. Lin, E.B. Newman. The leucine responsive regulatory protein: more than a regulator? *Trends Biochem. Sci.*, 18:260-263, 1993.

[105] F. Mordelet, J.P. Vert. SIRENE: supervised inference of regulatory networks. *Bioinformatics*, 24(16):76-82, 2008.

[106] S. Roweis, L. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, 2000.

[107] J.B. Tenenbaum, V. de Silva, J.C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319-2323, 2000

[108] M. Belkin, P. Niyogi. Laplacian eigenmaps and spectral techniques for embedding and clustering. *Advances in Neural Information Processing Systems*, 14, 2002.

[109] R.R. Coifman, et al. Geometric diffusions as a tool for harmonic analysis and structure definition of data: Diffusion maps. *Proceeding of the Natoinal Academy of Sciences (PNAS)*, 102(21):7426-7431, 2005.

[110] J. Ham, D.D. Lee, S. Mika, B. Scholkopf. A kernel view of the dimensionality reduction of manifolds. *Proceedings of the twenty-first international conference on Machine learning*, 69, 2004.

[111] B. Schlkopf, K. Tsuda, J.P. Vert. *Kernel Methods in Computational Biology*, MIT Press, Cambridge MA, 2004.

[112] A. Ben-Hur, W.S. Noble. Kernel methods for predicting protein-. protein interactions. *Bioinformatic*, 21:38-46, 2005.

[113] G. Lerman, B.E. Shakhnovich. Defining functional distance using manifold embeddings of gene ontology annotations. *Proceeding of the Natoinal Academy of Sciences (PNAS)*, 104(27):11334-11339, 2006.

[114] J. Ernst, et al. A Semi-Supervised Method for Predicting Transcription Factor-Gene Interactions in Escherichia coli. *PLOS Computational Biology*, 4(3), 2008.