University of London
Imperial College of Science, Technology and Medicine
Department of Computing

# Data Integration for Regulatory Module Discovery

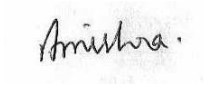
Alok Mishra

Submitted in part fulfilment of the requirements for the degree of
Doctor of Philosophy in Computing of the University of London and
the Diploma of Imperial College, September 2011

## Declaration

I hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person nor material which has been accepted for the award of any other degree or diploma of the university or other institute of higher learning, except where due acknowledgement has been made in the text.

Name: Alok Mishra

Signature:  Date: 26 September 2011

## Abstract

Genomic data relating to the functioning of individual genes and their products are rapidly being produced using many different and diverse experimental techniques. Each piece of data provides information on a specific aspect of the cell regulation process. Integration of these diverse types of data is essential in order to identify biologically relevant regulatory modules. In this thesis, we address this challenge by analyzing the nature of these datasets and propose new techniques of data integration.

Since microarray data is not available in quantities that are required for valid inference, many researchers have taken the blind integrative approach where data from diverse microarray experiments are merged. In order to understand the validity of this approach, we start this thesis with studying the heterogeneity of microarray datasets. We have used KL divergence between individual dataset distributions as well as an empirical technique proposed by us to calculate functional similarity between the datasets. Our results indicate that we should not use a blind integration of datasets and much care should be taken to ensure that we mix only similar types of data. We should also be careful about the choice of normalization method.

Next, we propose a semi-supervised spectral clustering method which integrates two diverse types of data for the task of gene regulatory module discovery. The technique uses constraints derived from DNA-binding, PPI and TF-gene interactions datasets to guide the clustering (spectral) of microarray experiments. Our results on yeast stress and cell-cycle microarray data indicate that the integration leads to more biologically significant results.

Finally, we propose a technique that integrates datasets under the principle of maximum entropy. We argue that this is the most valid approach in an unsupervised setting where we have no other evidence regarding the weights to be assigned to individual datasets. Our experiments with yeast microarray, PPI, DNA-binding and TF-gene interactions datasets show improved biological significance of results.

# Acknowledgements

First and foremost, I would like to thank my supervisor, Professor Duncan Gillies, for his support and advice over the past three years. Without his help and patience during many difficult periods, this thesis would not have been possible. Duncan is not only an excellent mentor but also a researcher in its truest meaning. I feel privileged that I had the opportunity to work with and learn from him to be humane while striving for excellence. I would also like to thank Duncan for providing me the opportunity to help three MSc students in their dissertation as well as tutorial opportunities.

I would like to take this opportunity to thank my thesis examiners, Professor Michael Stumpf and Professor David Gilbert, for agreeing to examine my work. Also, my second supervisor, Professor Daniel Rueckert, for examining my transfer report and his valuable advice.

Imperial College and Beit Trust funded my stay for three years in this expensive city (London) with *Imperial College Deputy Rector's Scholarship* and *Beit Fellowship for Scientific Research*. I am thankful from the bottom of my heart for providing me this opportunity. I would also like to express my gratitude to the Department of Computing for funding my travel costs both within the United Kingdom and internationally. Many thanks to the Zebra Housing Association, for providing me a place to stay at subsidised cost, without which I would certainly have missed the joy and privilege of living in central London so close to the university.

I would like to thank all the denizens of Room-433 of Imperial College DOC (Huxley Building) for lifelong memories. This includes non-logicians Georgia, Alexei, Joao, Peter, Simon and logicians Uri, Mohammed, Jonathan, Mark, Nick and Clemens.

A huge thanks to all my friends, most importantly, Lennon and Alexei for their company, support and help during various ups and downs. I would also like to thank Orlando for many a stimulating discussion and help with filling gaps in my education.

My thanks go to the BR forum members and multitudes of other websites that provided much needed distraction when I was not working. A big thank you, to the coffee shops in and around Imperial College for the caffeine, when I was going through the daunting task of writing up this thesis.

I am grateful to my parents for their unending love, support and encouragement. Without the values that they taught me, I would not be here. I thank all my family members for encouraging me, and bearing my mood changes all along.

Finally, and most importantly, my wife Saswati, for her endless love, support and encouragement throughout this journey. It would not have been possible without you.

*To Ma, Papaji and Saswati*

# Contents

**Abstract** **iii**

**Acknowledgements** **iv**

**1 Introduction** **1**

1.1 Biological Background . . . . . 3

1.2 Data Sources . . . . . 5

1.2.1 Microarrays . . . . . 6

1.2.2 ChIP on chip . . . . . 7

1.2.3 Transcription factor binding motifs . . . . . 7

1.2.4 Protein-protein interactions . . . . . 8

1.3 Research Goals and Our Approach . . . . . 9

1.3.1 Thesis Scope . . . . . 10

1.4 Thesis Contributions and Publications . . . . . 11

1.5 Thesis Outline . . . . . 12

**2 Regulatory Module Discovery Algorithms** **14**

2.1 Plain Clustering . . . . . 14

2.2 Causal Networks . . . . . 15

2.3 Supervised Module Algorithms . . . . . 17

**3 Data Integration for Regulatory Module Discovery 22**

3.1 Introduction . . . . . 22

3.2 Methodology . . . . . 24

3.2.1 Kullback Leibler divergence among datasets . . . . . 26

3.2.2 Cluster similarity . . . . . 28

3.2.3 Cluster similarity indices . . . . . 29

3.3 Results . . . . . 31

3.3.1 Cluster similarity among datasets . . . . . 31

3.3.2 KL divergence among datasets . . . . . 34

3.3.3 Correlation between KL divergence and cluster similarity . . . 37

3.3.4 Effect of data heterogeneity . . . . . 38

3.4 Discussion . . . . . 40

3.5 Conclusion . . . . . 42

**4 Semi-supervised Regulatory Module Discovery 43**

4.1 Spectral Clustering . . . . . 44

4.1.1 Graph notations . . . . . 44

4.1.2 Similarity matrices and graph Laplacians . . . . . 45

4.1.3 Graph clustering . . . . . 47

4.1.4 Algorithm explanation . . . . . 49

4.2 Datasets and Our Algorithm . . . . . 51

4.2.1 Microarray datasets . . . . . 51
4.2.2 DNA-binding dataset . . . . . 52
4.2.3 PPI dataset . . . . . 53
4.2.4 TF-gene interactions dataset . . . . . 55
4.2.5 Semi-supervised spectral clustering . . . . . 55
4.2.6 Toy dataset explorations . . . . . 58
4.2.7 Parameter optimization . . . . . 59
4.3 Statistical validation of results . . . . . 66
4.4 Biological Significance Analysis . . . . . 73
4.4.1 Numerical Biological Significance comparison (using mutual information) . . . . . 74
4.4.2 Qualitative Biological Significance (using Gene Ontology annotations) . . . . . 77
4.5 Related Work and Discussion . . . . . 90
4.5.1 Constrained clustering . . . . . 90
4.5.2 Semi-supervised clustering . . . . . 91
4.5.3 Co-clustering . . . . . 94
4.6 Conclusion . . . . . 94

**5 Maximum Entropy Kernel Integration for Regulatory Module Discovery 96**

5.1 Introduction . . . . . 96
5.1.1 Spectral or eigen-decomposition . . . . . 98
5.2 Kernel Methods . . . . . 99

5.2.1 Various kernel or similarity functions . . . . . 99
5.2.2 From similarities to a valid kernel . . . . . 102
5.2.3 Kernel normalization . . . . . 103
5.3 Principle of Maximum Entropy . . . . . 103
5.3.1 Entropy . . . . . 103
5.3.2 Principle of maximum entropy . . . . . 104
5.4 Maximum Entropy Kernel Integration . . . . . 105
5.4.1 Algorithm . . . . . 108
5.5 Datasets and Methodology . . . . . 108
5.5.1 Parameter optimisation results . . . . . 110
5.6 Biological Significance Analysis . . . . . 112
5.6.1 Numerical Biological Significance comparison (using mutual information) . . . . . 112
5.6.2 Qualitative Biological Significance (using Gene Ontology annotations) . . . . . 114
5.7 Related Work and Discussion . . . . . 123
5.8 Conclusion . . . . . 126

**6 Summary and Future Work** **128**

6.1 Summary . . . . . 128
6.2 Challenges and Future Directions . . . . . 129
6.3 Final Remarks . . . . . 132

**Bibliography** **133**

**Acronyms** **143**

# List of Tables


3.1 Comparison of Rand's Index and *adjusted* Rand's Index . . . . . 30

3.2 Cluster similarity among full (non scale-normalized) datasets . . . . . 32

3.3 Cluster similarity among filtered (non scale-normalized) datasets . . . 32

3.4 Cluster similarity among full (scale-normalized) datasets . . . . . 32

3.5 KL Divergence among full (non scale-normalized) datasets . . . . . 35

3.6 KL Divergence among filtered (non scale-normalized) datasets . . . . 35

3.7 KL Divergence among full (scale-normalized) datasets . . . . . 35

3.8 Pearson's Correlation betwen KL divergence and Cluster similarity . 37

4.1 Number of constraints from DNA-binding dataset at various p-value thresholds . . . . . 53

4.2 Dunn's and Davies Bouldin's index values for perturbed Stress dataset. 69

4.3 Dunn's and Davies Bouldin's index values for perturbed Cell-Cycle dataset. . . . . 69

4.4 Summary of mean and standard deviation values of individual microarray datasets. . . . . 69

4.5 Dunn's and Davies Bouldin's index values after combination of subsampled Stress and ChIP-chip datasets . . . . . 70

4.6 Dunn's and Davies Bouldin's index values after combination of sub-sampled Stress and PPI datasets . . . . . 70

4.7 Dunn's and Davies Bouldin's index values after combination of sub-sampled Cell-cycle and ChIP-chip datasets . . . . . 71

4.8 Dunn's and Davies Bouldin's index values after combination of sub-sampled Cell-cycle and PPI datasets . . . . . 71

4.9 Dunn's and Davies Bouldin's index values after combination of sub-sampled Cell-cycle and Yeastract datasets . . . . . 72

4.10 Mean and standard deviation of Dunn's and Davies Bouldin's index values for all combined datasets. . . . . 72

4.11 Comparison of Biological significance index values before and after semi-supervised integration. . . . . 76

4.12 A subset of GO term enrichment values for the stress microarray dataset 79

5.1 Number of genes considered for individual dataset pairs . . . . . 109

5.2 Stress microarray dataset: Comparison of Biological significance index values before and after maximum entropy data integration . . . . 113

5.3 Cell-cycle microarray dataset: Comparison of Biological significance index values before and after maximum entropy data integration . . . 113

# List of Figures


1.1 Eukaryote cell . . . . . 3

1.2 Central dogma of molecular biology: $DNA \longrightarrow RNA \longrightarrow Protein$ . . 3

1.3 Gene regulation . . . . . 4

1.4 A cross-section of hybridized cDNA microarray . . . . . 6

1.5 Protein-protein interactions in yeast . . . . . 8

3.1 Variation of cluster similarity with data homogeneity . . . . . 39

4.1 Constraints derived from DNA-binding dataset at various p-value cut-offs . . . . . 54

4.2 Visual representation of semi-supervised spectral clustering . . . . . 57

4.3 Visual indication of Spirals dataset clustering quality improvement with increasing number of constraints. . . . . 60

4.4 Various runs showing clustering quality improvement with increasing number of known constraints. . . . . 61

4.5 Stress dataset: Sigma optimization using Dunn's Index . . . . . 65

4.6 Stress dataset: Sigma optimization using Davies Bouldin's Index . . . 65

4.7 Cell-cycle dataset: Sigma optimization using Dunn's Index . . . . . 67

4.8 Cell-cycle dataset: Sigma optimization using Davies Bouldin's Index . 67

4.9 Sections of the image showing significant GO term enrichment in Stress only dataset. . . . . 81

4.10 Sections of the image showing significant GO term enrichment in Stress dataset combined with knowledge (constraints) from ChIP-chip dataset. . . . . 82

4.11 Sections of the image showing significant GO term enrichments in Stress dataset combined with knowledge (constraints) from PPI dataset. 83

4.12 Sections of the image showing significant GO term enrichment in Stress dataset combined with knowledge (constraints) from Yeastract dataset. . . . . 84

4.13 Sections of the image showing significant GO term enrichment in Cell-cycle only dataset. . . . . 86

4.14 Sections of the image showing significant GO term enrichment in Cell-cycle dataset combined with knowledge (constraints) from ChIP-chip dataset. . . . . 87

4.15 Sections of the image showing significant GO term enrichment in Cell-cycle dataset combined with knowledge (constraints) from PPI dataset. . . . . 88

4.16 Sections of the image showing significant GO term enrichment in Cell-cycle dataset combined with knowledge (constraints) from Yeastract dataset. . . . . 89

5.1 ChIP-chip, PPI and Yeastract datasets: Beta optimization . . . . . 111

5.2 Sections of the image showing significant enrichment in Stress dataset combined with ChIP-chip dataset. . . . . 115

5.3 Sections of the image showing significant enrichment in Stress dataset combined with PPI dataset. . . . . 116

5.4 Sections of the image showing significant enrichment in Stress dataset combined with Yeastract dataset. . . . . 117

5.5 Sections of the image showing significant enrichment in Cell-cycle dataset combined with ChIP-chip dataset. . . . . 119

5.6 Sections of the image showing significant enrichment in Cell-cycle dataset combined with PPI dataset. . . . . 120

5.7 Sections of the image showing significant enrichment in Cell-cycle dataset combined with Yeastract dataset. . . . . 121

# Chapter 1

# Introduction

> "We are drowning in information, while starving for wisdom" - *E.O. Wilson*

The most fundamental unit of life, a living cell, functions by a complex orchestration of various genes and their products (mRNA, proteins etc.). In a more abstract manner we can say that genes are interrelated in highly structured networks of information flow in order for the cells to function. A network of such gene products (proteins) known as transcription factors (TFs) which regulate the production of other transcription factors or other proteins is called a *transcriptional regulatory network* (TRN) or *gene regulatory network* (GRN). The understanding and reconstruction of this regulation process at a global level is one of the major challenges for the nascent field of bio-informatics (Schlkopf et al., 2004).

Considerable work has been done by molecular biologists over the past many years in identifying the functions of specific genes. In an ideal world it would be desirable to apply these results in order to build detailed models of regulation where the precise action of each gene is understood. However, the large number of genes and the complexity of the regulation process means that this approach has not been feasible. Research into discovering causal models based on the actions of individual genes has encountered a major difficulty in estimating a large number of parameters from a paucity of experimental data. Fortunately however, biological organisation opens up the possibility of modelling at a less detailed level. In nature, complex functions of living cells are carried out through the concerted activities of many genes and

gene products which are organized into co-regulated sets also known as *regulatory modules* (Segal et al., 2003). Understanding the organization of these sets of genes will provide insights into the cellular response mechanism under various conditions.

Recent advances in measurement technologies and computing resources have led to the wide availability of a considerable volume of genome-wide data on gene activity measured using several diverse techniques. By fusing this data using an integrative approach, we can try to unravel the regulation process at a more global level. Although an integrated model could never be as precise as one built from a small number of genes in controlled conditions, such global modelling can provide insights into higher processes where many genes are working together to achieve a task. Various techniques from statistics, machine learning (Hastie et al., 2001) and computer science have been employed by researchers for the analysis and combination of the different types of data in an attempt to identify and understand the function of regulatory modules.

There are two underlying problems resulting from the nature of the available data. Firstly, each of the different data types (microarray, DNA-binding, protein-protein interaction and sequence data) provides a partial and noisy picture of the regulatory process. They need to be integrated in order to obtain an improved and reliable picture of the whole underlying process. Secondly, the amount of data that is available from each of these techniques is severely limited. To learn good models we need lots of data (Yeung et al., 2004), yet data is only available for a few experiments of each type. To alleviate this problem many researchers have taken the path of merging all available datasets before carrying out an analysis. Thus there can be some confusion regarding the term *integrative* because it has been used to describe both of these two very different approaches to data integration: one among datasets of the same type, for example microarrays, but from different experiments, and the other among different types of data, for example microarray and DNA binding data. The goal of this thesis is to analyze the problems resulting from the existing integrative approaches and suggest better solutions for improved data integration.

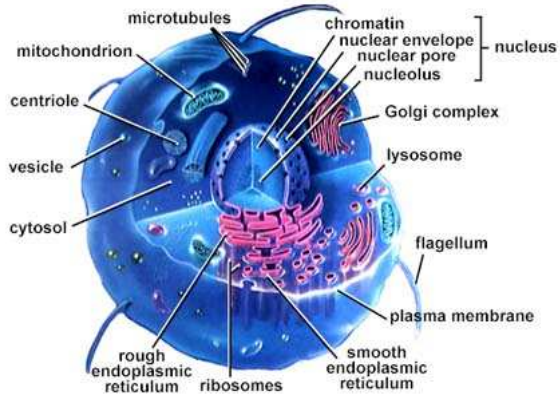



Figure 1.1: Eukaryote cell

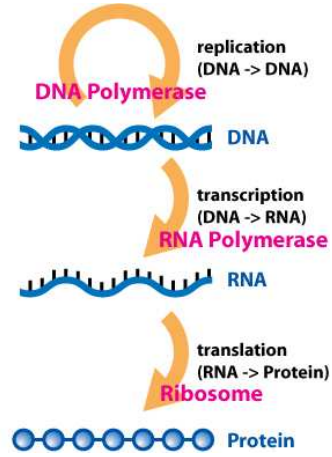



Figure 1.2: Central dogma of molecular biology: $DNA \longrightarrow RNA \longrightarrow Protein$

## 1.1 Biological Background

This section presents a brief overview of elementary molecular biology concepts that are essential in order to fully understand this thesis. For a detailed understanding please refer to any standard text on molecular biology like Alberts et al. (2002)*; Hunter (1993) has written a brief yet excellent introduction specially for people with computing background.

Cells, which are the most fundamental units of life, differ in higher organisms, e.g. multicellular animals and plants, known as *eukaryotes* from those of the less evolved *prokaryotes*, e.g. bacteria, in having a well-defined *nucleus* (see Figure-1.1†) which carries the genetic material. The nucleus is the most prominent structure inside the eukaryotic cell and the genetic information inside it is contained in the form of a deoxyribonucleic acid (DNA) molecule. DNA has a *double-helix* structure formed by two complementary strands, each made up of a sequence of *nucleotides* which are composed of adenine, thymine, cytosine or guanine.

The central dogma of molecular biology is that the sequence information on the

*a free web book is available at `http://www.web-books.com/MoBio/`

†image taken from online book at `http://www.estrellamountain.edu/faculty/farabee/biobk/biobooktoc.html`

DNA is converted into ribonucleic acid (RNA) through the process of *transcription*. This sometimes is also referred as *gene expression*. RNA, through the process of *translation* is later converted into a sequence of amino acids that results in a *protein* (see Figure-1.2[‡]). The functional unit on the DNA that codes for an individual protein is known as a *gene*. The sequence of the nucleotides in the double helix within a gene specifies the primary structure of a protein. The complete sequence of nucleotides on the DNA is also referred to as the *DNA sequence*. Higher organisms are made up of various different cell types each of which performs a specific role requiring a specific set of gene products. The fascinating fact is that each of these cells contains exactly the same set of genes (DNA) but a different set of gene products. The remarkable diversity among the cells is a result of a precisely controlled mechanism of expression and regulation of a subset of genes in each cell type.

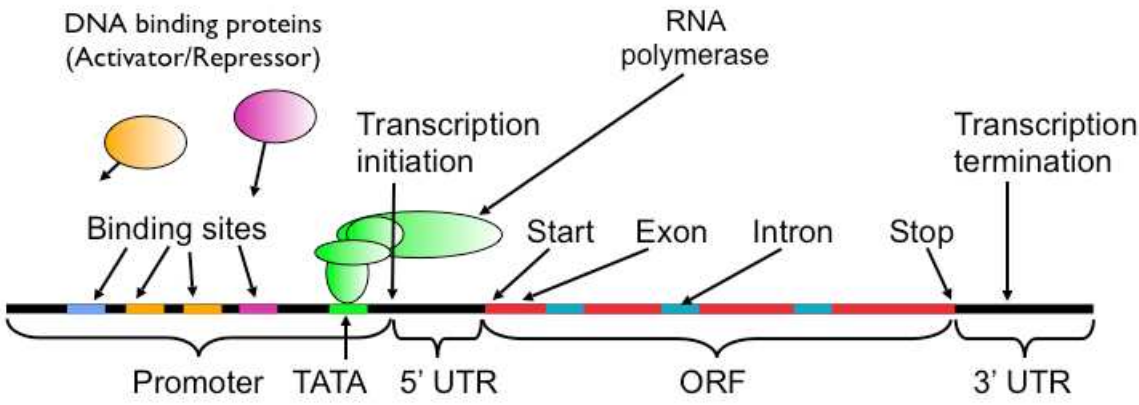



Figure 1.3: Gene regulation

The transcription process begins when TFs are activated by a trans-membrane receptor, leading them to bind to gene regulatory elements and to promote access to the DNA and facilitate the recruitment of RNA polymerase to the transcriptional start site as shown in Figure-1.3. The gene regulatory elements of the DNA, also known as *promoter regions*, are situated upstream of the gene at a distance which can vary from a few base pairs to hundreds of base pairs. The regulatory elements contain binding sites for multiple transcription factors allowing each gene to respond

[‡]image source: `http://en.wikipedia.org/wiki/File:Central\_Dogma\_of\_Molecular\_Biochemistry\_with\_Enzymes.jpg`

to multiple signalling pathways and facilitate fine-tuning of the messenger ribonucleic acids (mRNAs) that are produced. Once the transcription factors are bound on the regulatory elements, they can either promote or inhibit gene expression. In the case of a promoter, the process of transcription starts. A protein called RNA polymerase starts to copy the information contained in the gene into mRNA. These mRNA molecules, being exact replicas of the gene, contain both *exons* (which will be used in the later process) and *introns* (which will be removed). A process known as *splicing* removes the introns and the remaining mRNA, called spliced mRNA, is transported out of the nucleus into the cellular material. There it is translated into a polypeptide chain with the help of ribosomes and this chain then folds into a three-dimensional structure known as protein.

The previous paragraph gives only a partial picture. Since transcription factors themselves are proteins, the same process may regulate them. In fact, there are genes that code just for transcription factors. This process is similar to a feedback loop in which transcription factors are regulated by other transcription factors. A major goal of bioinformatics is to understand how transcription factors affect gene expression and which groups of genes are co-regulated by certain sets of transcription factors.

## 1.2 Data Sources

Recent technological advances have led to an explosion in both the quantity and types of data being generated. Various observation techniques capture different facets of the cell regulatory process. These are primarily generated by molecular biologists using experimental techniques. Some of the types currently available are:

- mRNA expression measured using microarrays.
- Whole genome transcription factor binding measured using chromatin immunoprecipitation (ChIP) on chip.
- TF binding motifs from the promoter sequences of genes.
- Protein-protein interactions using co-immunoprecipitation and other techniques.

### 1.2.1 Microarrays

One of the most important sources of data related to the transcription process is the genome-wide measurement of mRNA expression levels carried out using microarrays. These have received considerable attention in the last six years and various technologies for microarray measurement have been developed (Schulze and Downward, 2001). A microarray allows simultaneous measurement of the expression levels of a large number of genes. It consists of a grid of a large number of microscopic *spots* of DNA oligonucleotides on a silicon chip, each containing a specific DNA sequence. Depending on the specific technique of manufacturing, these are either called *cDNA microarrays* or *oligonucleotide microarrays*. Microarray technology is based on the concept of *hybridization* which means that DNA has complementary strands and given the right conditions, complementary strands will bind to each other. We want to measure mRNA content, and since mRNA is complementary to the DNA strand from which it was created, it will bind to its complement on the probe. Therefore, each of the spots contain a short section of a gene or other DNA element that we want to study as a probe to *hybridize* a complementary cDNA sample (mRNA). Before hybridization, the target is tagged with a fluorescent dye so that after hybridization, the intensity of colour detected by a fluorescence-based detector indicates the abundance of the target.

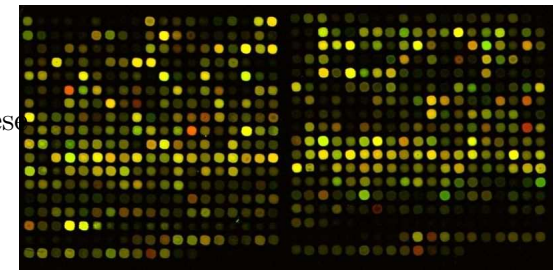


Figure 1.4: A cross-section of hybridized cDNA microarray

*Two-channel microarrays* as shown in Figure-1.4 are hybridized with samples from two conditions, e.g. diseased tissue versus healthy tissue, and are tagged with two different colours. Fluorescent dyes Cy3 (green) and Cy5 (red) are commonly used for tagging these samples. The two coloured cDNA samples are mixed and hybridized to a single microarray that is then scanned. Relative intensities of each colour are then used to identify up-regulated and down-regulated genes. On the other hand, in *single-channel microarrays*, the arrays are designed to give estimates of the absolute levels of gene expression. Therefore, comparison of two conditions requires two separate single-dye hybridizations.

Similar expression profiles identify genes that may be controlled by a shared regulatory mechanism. Paul Spellman is one of the microarray pioneers who used it to study global expression of genes at various time points in yeast cell cycle (Spellman et al., 1998). He along with some other researchers (Gasch et al., 2000) also studied the response of the yeast genes when subjected to various kinds of stress. Processing microarray data to reduce the errors introduced at various stages is known as *normalization*. Quackenbush (2006) provides a good overview of the techniques used for normalization and analyzing while Smyth et al. (2003) discuss in detail the statistical issues involved in normalization.

### 1.2.2 ChIP on chip

ChIP on chip, also referred to as *ChIP-chip* assay is a technique which allows us to study genome wide binding of transcription factors to the DNA simultaneously. It combines *ChIP* with microarray technology (chip) to determine *in vivo* all the regions of interest on the DNA's promoter regions, i.e., where each of the transcription factors bind. Harbison et al. (2004) determined the global genomic occupancy of 203 transcription factors in yeast, which are all known to bind to DNA in the yeast genome. Lee et al. (2002) produced a similar yeast dataset for a smaller number of transcription factors. Both these researchers reported results in the form of a confidence value (statistical P value) of a transcription factor attaching to the promoter region of a gene. The reason behind using statistical techniques was to average the errors in microarray technology and account for multiple cell populations. One of the prominent problems with such approaches is that in order to infer whether a transcription factor attached to the promoter sequence or not, we have to choose an arbitrary artificial threshold of the P-value.

### 1.2.3 Transcription factor binding motifs

Transcription factor binding motifs are sequence patterns observed in the intergenic regions of the genome usually located upstream of the genes (promoter region). They are thought to be responsible for allowing access of transcription factors to binding sites which eventually leads to regulation of transcription. While *ChIP-chip* provides *in vivo* evidence of TF binding, this is an indirect technique that was used before the advent of *ChIP-chip*. The primary reason for considering this as an information

source in the presence of *ChIP-chip* data is that *ChIP-chip* is very noisy and uses evidence from multiple experiments (Lee et al., 2002) to report the possibility of a TF binding to the promoter region of a gene in the form of a p-value.

Most of the approaches to identifying these were based on first clustering genes by co-expression, and then looking for common sequences in the upstream regions of the genes located in the same cluster. The upstream sequences are catalogued in the Yeast Proteome Database (YPD), as well as *Saccharomyces cerevisiae* Promoter Database (SCPD) which is dedicated to the curation of yeast genes' promoter sequences (Zhu and Zhang, 1999).

### 1.2.4 Protein-protein interactions

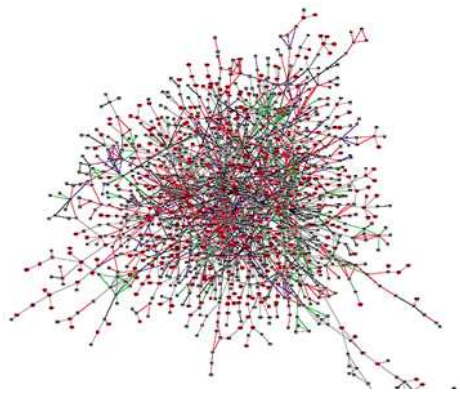


Figure 1.5: Protein-protein interactions in yeast

The interactions between proteins is important for many biological functions, e.g. signal transduction, where signals from outside a cell are transmitted to the inside by protein-protein interactions of the signaling molecules. This dataset is important for our study because proteins are gene products and proteins with similar functions and localization are more likely to interact in groups. This was shown by Schwikowski et al. (2000) where they observed that proteins of known function and cellular location tend to cluster together with 63% of the interactions between proteins with a common functional assignment and 76% occurring between proteins found in the same subcellular compartment. Therefore, genes producing interacting proteins are more likely to be co-regulated and have similar functionality. This was verified by Ge et al. (2001) who provide global evidence that genes with similar expression profiles are more likely to encode interacting proteins. Protein-protein interaction data for yeast is available as a result of advances in technologies like co-immunoprecipitation, mass-spectroscopy and yeast two-hybrid assays. There has been a tremendous growth in this type of data in the recent years. Gueldener et al. (2006) have manually com-

piled a protein-protein interaction (PPI) dataset from the literature and published large-scale experiments for yeast which is used as a reference and has been called a gold standard because of its quality and comprehensiveness (Yu et al., 2004).

## 1.3 Research Goals and Our Approach

The underlying problem that is addressed results from the nature of the available data. The problem is two fold: firstly, each of the current datasets, e.g. microarrays, DNA-binding, protein-protein interaction and sequence datasets, provide a *partial* and *noisy* picture of the whole process. Hence, we need to integrate them in order to obtain an improved and reliable picture of underlying process. Secondly, the amount of data that is available for each of these types is severely limited. To *learn* good models we need lots of data. Yet, data is only available for few of the experiments of one type. To alleviate this problem many researchers have taken the path of merging available datasets and then learning clusters from it. So, we see that there are two *distinct* types of integration happening, one among *different types* of datasets and the other among datasets of the *same type* but from *different experiments*.

Results are very hard to replicate when the datasets are different even if they result from experiments with the same conditions but done by different experimenters. As clearly stated by Orphanides and Reinberg (2002) there is no single model of regulation and each cell process has evolved its own detailed regulation model. There are certain motifs that can be seen in most of the processes but the actual details of the process are very different from one another. Furthermore, for each underlying motif, the real size of the motif is very different from process to process and geneset to geneset. So, even though many researchers have used this approach of integrating data-sets (both types), it's not very clear what the implications on the final results are.

The first part of our research work focuses on *understanding the impact of integration* of the latter type (using datasets of the same type but different experiments) on the global modelling related to the transcriptional gene regulation processes. Most of the algorithms have justified their results *qualitatively* by interpreting their results with the help of biologists. We are interested in studying the *quantitative information overlap* among various datasets and whether our current algorithms are able to leverage the integration of diverse datasets in meaningful *biologically*

*relevant* results. Specifically, we studied the correlation between the theoretical distribution difference among the datasets being merged (using Kullback-Leibler (KL) divergence among distributions) to the functional difference among them (computed using cluster similarity). We also studied how much the functional similarity (in our case, the cluster similarity) varied because of dataset integration as we slowly integrate increasingly diverse datasets.

The second part of our research deals with proposing a framework and analysis of integrating datasets of different types, e.g. microarray, *ChIP-chip* and PPI datasets. The exact amount of overlap and correlation among functional datasets is unclear (Werner-Washburne et al., 2002; Kemmeren, 2002), yet data integration has been shown to increase the accuracy of tasks like gene function prediction compared to single source of data (Ge et al., 2001; Gerstein et al., 2002). For data integration, spectral clustering has been used as our primary tool. The main reason behind this choice is that it is based on computing similarities of variables which results in *affinity* matrices. Similarity computation allows us to normalize diverse datatypes (which were previously considered unintegrable) into a common format (affinity matrices) and then integrate them. Increasingly, biological datasets are non-vectorial, e.g. sequence and PPI data (which is available as a graph). There have been a lot of recent developments in various techniques of similarity computation among these non-vectorial datasets. With this as our foundation, two innovative techniques of integrating these datasets have been proposed.

### 1.3.1 Thesis Scope

Bioinformatics has grown into a vast field. It is not possible to describe in detail all of the techniques that have been followed or that have been used by the providers of publicly available data. We have not included the details of *within array* normalization techniques of microarray data used by their experimenters and assume that all the microarray data is suitably normalized.

Even though we have used k-means clustering algorithm, we have not discussed it as it is one of the most well known and elementary techniques. Similarly, we have not discussed the Gene Ontology in great detail. For all these, suitable references have been provided in the thesis.

## 1.4 Thesis Contributions and Publications

- A comprehensive and critical review of research work done in this field has been published as
  - Mishra, A. and Gillies, D. (2008). Data integration for regulatory gene module discovery. In Daskalaki, A., editor, *Handbook of Research on Systems Biology Applications in Medicine*. IGI Global, Hershey, PA.
- Issues related to calculation of biological significance of clusters using Gene Ontology has been published as
  - Mishra, A. and Gillies, D. (to be published in 2010). Validation Issues in Regulatory Module Discovery. In: Huma Lodhi and Stephen Muggleton (eds.), Elements of Computational Systems Biology, John Wiley & Sons, Inc. ISBN: 0470180935.
- An empirical technique to calculate the functional similarity of datasets using the concept of cluster similarity has been developed. We propose this can be used as an index for dataset similarity after showing a very high correlation with underlying data distribution differences. We have also demonstrated that dataset integration should only be done by first choosing similar datasets, otherwise the signals present in the datasets could be overwhelmed by noise. Part of this work published as
  - Mishra, A. and Gillies, D. (2007). Effect of microarray data heterogeneity on regulatory gene module discovery. *BMC Systems Biology*, 1(Suppl 1):S2
- A semi-supervised spectral clustering technique to integrate two datasets where one is acting as a source of supervision on the clustering of the other has been developed. We validated the results using Gene Ontology. Part of this work published as
  - Mishra, A. and Gillies, D. (2008). Semi supervised spectral clustering for regulatory module discovery. In *Data Integration in the Life Sciences*, pages 192-203.

- A principled technique to integrate two diverse datasets where no evidence is available regarding their individual weights (importance) has been developed using the principle of maximum entropy. We validated the results after spectral clustering of the integrated matrix using Gene Ontology.
- Modular and reusable software implementations of all the techniques has been developed using python[§] and R[¶].

## 1.5 Thesis Outline

The outline of this thesis is as follows. In Chapter-2, we present a critical literature review related to the evolution of the research related to transcriptional regulatory networks and modules. Even though this field is relatively new, the amount of research done is enormous because of increasing focus and funding. This chapter tries to tie together all the past efforts into a coherent story.

In Chapter-3, we study the effect of *integrating increasingly diverse microarray datasets*. For functional similarity, we compute the cluster similarity among datasets using modified Rand's index. To estimate the theoretical difference between the underlying distributions of individual datasets, we use the KL divergence. Finally, we study the correlation between the two measures (functional and theoretical).

In Chapter-4, we start with a discussion of spectral clustering and its theoretical foundations. In order to integrate microarray datasets with other datasets e.g., PPI and DNA-binding datasets, we propose a semi-supervised spectral clustering technique. We apply this technique on two of the popular yeast microarray datasets and evaluate the results of integration using the Gene Ontology.

While the semi-supervised algorithm is heuristic in combining separate evidence of similarity, in Chapter-5, we propose a more principled approach to integration of similarity matrices. We merge similarity matrices derived from various datasets (microarray, PPI and DNA-binding) using the principle of maximum entropy (Jaynes, 1957) and analyze the results.

[§]a popular programming language with efficient Linear Algebra library
[¶]a software environment for statistical computing

Finally, Chapter-6 concludes this thesis with remarks about the drawbacks and challenges faced in our approach. It details where the field is heading and what would be the future challenges. We also discuss the scope and direction of extending the current work in future.

# Chapter 2

# Regulatory Module Discovery Algorithms

In this chapter, we review the research done in data integration techniques for regulatory module discovery. Initial research in this area involved plain clustering of microarray data. This was followed by progressively sophisticated modelling as well as integration of various data types.

## 2.1 Plain Clustering

When microarray data started becoming available in the late 1990s, a prime goal was to identify sets of genes that act together functionally to perform certain cellular tasks such as metabolism or cell-cycle functions. In this early phase of data analysis, various clustering algorithms, e.g. Eisen et al. (1998), were applied in order to find such gene modules. An assumption behind this clustering approach was that co-expression implied co-regulation. In other words, if sets of genes were showing similar patterns of microarray expression, they must be co-regulated and hence belong to the same module. So, co-expression was assumed to imply co-regulation and co-regulation was assumed to imply similar function. However, both these assumptions are not always correct. The validity of the resulting clusters could be tested by identifying common promoter elements on the upstream portion of genes within the same cluster on the assumption that genes are co-regulated because they

have similar promoter elements. Another popular way to show validity was by using gene ontology to show that the majority of genes belonging to a module were similar in function. This was done by computing the enrichment of gene ontology terms in each of the clusters. Better clusters were expected to have more significant enrichment of these terms. In these early works, no external information was used to guide the process of clustering. A review of the early techniques based on ad-hoc as well as model based clustering can be found in de Jong (2002).

## 2.2 Causal Networks

Naturally, the research community wanted to model the causal relationships among various genes in much more detail, and this precipitated a second phase of modelling in which mostly Bayesian networks and their variants, such as dynamic Bayesian networks (DBNs), were applied to model the gene regulatory processes (Friedman et al., 2000; Husmeier, 2003; Murphy and Mian, 1999; Zou and Conzen, 2005). Friedman et al. (2000) were the first to utilise Bayesian networks for modelling gene expression data and they tried two types of local distribution - discrete (multinomial) and continuous (linear Gaussian) to express the relation between dependent genes. They tested the work on the microarray expression data of Spellman et al. (1998). When networks that modelled the data accurately were identified, two pairwise features were computed from them - Markov relations and order relations. The Markov relation just checks if each gene of a pair is in the Markov blanket of the other. This would imply a direct causal relationship between them indicating a biological relation. The order relation checks if X is ancestor of Y in all the networks of an equivalence class. This can be determined directly from the directed graph by checking whether there is a path from X to Y that is directed towards Y consistently. An order relation implies that the two genes have a role in some more complex regulatory process. Temporal aspects of data were incorporated into the model by adding a discrete variable as the root. They suggested that non-linear local and temporal models should be used for better accuracy. Their analysis of the results shows that the method is sensitive to the choice of local model and in the case of the multinomial distribution is also sensitive to the discretization method used. Werhli et al. (2006) carried out a comparative study of the performance of modelling gene regulatory networks using graphical Gaussian models (GGMs), relevance networks

and Bayesian networks (BNs). They used both laboratory data as well as simulated data to evaluate the different approaches. They observed that on both types of data, Bayesian networks outperformed both relevance networks and graphical Gaussian models.

The major difficulty with this fine tuned modelling approach is that for such a high dimensional problem involving many thousands of genes, the amount of experimental data available is never enough for accurate modelling. Moreover, it is very hard to deal with the cyclical feedback nature of gene networks using Bayesian networks since, without the explicit incorporation of time, they only handle acyclic relationships among the variables. The end result of such models was that the performance was not good and not many verifiable findings were made (Husmeier, 2003). In order to improve upon the results, work was done to incorporate better prior knowledge in the Bayesian network based modelling. Imoto et al. (2003) combined PPI, DNA binding, promoter element motifs as well as literature text mining. Tamada et al. (2003, 2005) also used similar diverse datasets to build Bayesian network models.

Ihmels et al. (2002) proposed an algorithm called *Signature*, which performs bi-clustering, that is to say clustering genes, and conditions together based on expression data. It is unlike the more established bi-clustering algorithms in that it does not simultaneously generate data partitions but works in steps. The input to the algorithm is a set of genes and, in the first step, experimental conditions under which these genes change their expression above a threshold are chosen. In the second stage, all genes that have changed expression significantly under these conditions are selected. They evaluate the consistency of their clustering algorithm by analysing the recurrence of the output gene sets in their resulting modules when the input is mixed with irrelevant genes. The idea is that the results of any good algorithm should not deviate too much when slight perturbations are introduced in the data. A module is considered to be reliable if it is obtained from several distinct slightly perturbed input gene sets. Since it carries out a refinement of clusters in two stages, there can be no guarantee that the results would be clustered in a globally optimal manner. A better formulation might be to use the expectation maximization (EM) algorithm in order to maximise their objective function.

## 2.3 Supervised Module Algorithms

After these initial frustrations in moving from very naive modelling (plain clustering) to highly detailed modelling (DBN), research began to tread a path somewhere in the middle. This pragmatic approach did yield very good results and is still the basis of current research. One of the most complete studies using these types of weakly supervised methods was carried out by Segal et al. (2003). Their method, called *Module Networks* algorithm, takes as input a gene expression data set and a large precompiled set of candidate regulatory genes and outputs groups of co-regulated genes (modules), their regulators, and a regulation program that specifies behaviours of the modules as a function of regulator's expression and the conditions under which regulation takes place. It uses an iterative procedure that searches for a regulation program for each module (set of genes) and is based on the EM method that is initialised with the results of another clustering algorithm. For each cluster of genes, it searches for a regulation program that provides the best prediction of the expression profiles of genes in the module as a function of the expression of a small number of genes from the regulator set. After identifying regulation programs for all clusters, the algorithm re-assigns each gene to the cluster whose program best predicts its behaviour. It iterates till convergence, refining both the regulation program and the gene partition in each iteration.

In their experiments, they compiled a set of regulators from the Saccharomyces Genome Database (SGD) (Cherry et al., 1998) and the YPD (Payne and Garrels, 1998) based on annotations that broadly suggest that certain genes have a regulatory role, as either a transcription factor or a signalling protein. They also identified more potential regulators by finding genes similar to those above but removing the global regulators from the list. Microarray data for gene expression for yeast was collected from the Stanford Microarray Database (SMD). They chose a subset that had significant gene expression change and removed from this set the cluster known to be generic environmental response genes. Finally, they added all the genes from the regulator list above. With these two datasets (expression and regulators), they use a module network learning algorithm (Segal et al., 2005) to find separate sets of regulators and the regulated modules. They obtained modules that showed significant similarity in promoter element motifs as well as annotations in the gene ontology compiled by the Gene Ontology Consortium (2001).

At about this time, more significant prior knowledge started becoming available in the form of ChIP-chip DNA binding data and other sources as described in the previous chapter. The next step of research focused on ways of integrating these datasets in order to find gene modules.

Bar-Joseph et al. (2003) describe an algorithm for discovering regulatory modules. Their algorithm is called Genetic Regulatory Modules (GRAM), and combines microarray expression data with DNA-binding data. This was one of the first papers to have combined these two sources in order to achieve better clusters. DNA-binding data provides direct physical evidence of regulation and thus offers an improvement on previous work where only indirect evidence of interaction, for example promoter sequences, were used for prior information. The GRAM algorithm begins by performing an exhaustive search over all possible combinations of transcription factors indicated by the DNA-binding dataset using certain (strict) threshold P-values. This yields sets of genes that are regulated by sets of transcription factors. This gene list is filtered by studying their expression patterns to find genes that show co-expression. These act as seeds for gene modules. The next pass revisits transcription factors and expands the seed modules by adding genes with a relaxed P-value criterion that show co-expression. GRAM allows a gene to be part of more than one module. They identified 106 modules with 655 distinct genes regulated by 68 transcription factors. Within a module, the role of each transcription factor was identified as activator or repressor by analysing the correlation between the transcription factor's expression and the expression of regulated genes. Validation was done by analysing the promoter gene sequences in same cluster using the TRANSFAC (Wingender et al., 1996) database to identify common sequences.

Tanay et al. (2004) analysed several diverse datasets in an attempt to reveal the modular organisation of the yeast regulation system. They defined modules as groups of genes with statistically significant correlated behaviour across the diverse datasets. Their algorithm is called Statistical-Algorithmic Method for Bicluster Analysis (SAMBA) which is an extensible framework that can be easily updated when new datasets become available. In their analysis, they have integrated expression, PPI and DNA-binding datasets. In SAMBA, all genomic information is modelled as weighted bi-partite graphs. Nodes on one side of graph represent genes while the other side represents properties of genes, for example proteins encoded by them. Edges between property nodes and gene nodes are assigned weights. A

module is a sub graph of this bi-partite graph and a high quality module is defined as a heavy sub graph in the weighted bi-partite graph. The key point is that all sources of data are considered as properties of genes or proteins encoded by genes and there is one unified representation of all data as a bi-partite graph. Since their algorithm is based on combinatorial principles rather than graph theoretic (spectral) methods, there are no guarantees of a globally optimum partitioning. For evaluation, they found the biological significance of resulting clusters by calculating the enrichment score of all gene ontology (GO) terms associated with the genes of a module and later annotated the modules with the highest valued terms, that is to say those terms that are shared by the highest number of genes. They also analysed 600 base pairs in the upstream promoter region of the genes in a module for common motif enrichment. For each potential motif, they calculated the enrichment score among all the genes of the module. The positive aspect of their approach is that it utilises all sources of information in one uniform representation and only requires a measure of similarity of genes across a subset of properties. It also allows overlapping modules (with common genes), which is not a feature of traditional clustering algorithms. One of the limitations of their approach is that all sources of data are assigned equal weights and it isn't possible to weigh them separately according to reliability or importance.

In a later piece of work, Tanay et al. (2005) extended the work described above by investigating the SAMBA algorithm in more detail. They analysed more diverse datasets and focused more on the biological significance of the results, explaining them much more fully. The paper mainly describes a study of fresh data in the context of an extensive compendium of existing datasets using SAMBA. They proposed that future work should be carried out on integration across species on the basis that transcription modules are highly conserved among species.

The work of Lemmens et al. (2006) is similar to other module discovery algorithms in that they propose a very simple and intuitive algorithm to find co-regulated sets of genes that have similar expression profiles, the same binding transcription factors and a commonality of promoter motifs. The principal difference from other algorithms is that where others used motif information to validate their results, they have used it in order to find the modules itself. Their algorithm, known as ReMoDiscovery works in two passes. In the first pass, known as the *seed discovery* step, tightly co-expressed genes having a minimum number of common transcription

factors and a minimum number of common conserved motifs are put together in separate modules known as *seed modules*. In the second pass, known as the *seed extension* step, the size of the modules is increased by computing the mean of the module's gene expression and ranking the remainder of the genes in the dataset in order of their decreasing correlation with the mean profile. They compared their algorithm results with SAMBA and GRAM (discussed earlier) and reported their findings. All parameters, such as the cut off for various datasets, have been chosen without much justification, and the basic idea seems very similar to the work of Bar-Joseph et al. (2003). Some of the comparison metrics used do not seem very sound, for example average functional enrichment values have been calculated for the modules without normalising to account for the size of the modules. Similarly, summary statistics like minimum and maximum number of genes in modules do not provide relevant information for comparison of algorithms .

Huang and Pan (2006) investigated a traditional clustering method known as k-medoids which is a robust version of the k-means clustering method. Unlike k-means, which uses the mean of all genes in a cluster as its centre, k-medoids uses the most central gene (median). It is found by locating the one with minimum average dissimilarity to rest of the genes. They incorporated prior knowledge into it by modifying the distance metric used while clustering. They have used microarray expression data for clustering while biological knowledge about the known similarity between pairs of genes is derived from gene ontology. Previous approaches to include biological knowledge in distance based clustering methods have used gene ontology and metabolic pathways to estimate distance or similarity measures among gene pairs and then used these along with microarray expression based distance metrics to create an average distance, which is later used to cluster expression data. Huang and Pan used a *shrinkage* approach for the distance metric to shrink it towards zero in cases where there is strong evidence that two genes are functionally related. Their algorithm has two steps in which the first step uses the shrunk distance metric to cluster genes whose functionality is known from gene ontology. The second step clusters the remaining genes. In the second step clustered genes are assigned to either one of the step one clusters or to a step two cluster, depending on their distance from the medoids. The shrinkage parameter is chosen using cross validation. They evaluated their algorithm using both simulated as well as real data. In a later piece of work, Pan (2006) used known functions of genes from existing biological research to assign different prior probabilities for a gene to belong to a cluster. He developed

an EM algorithm for this stratified mixture model.

The research described above concerns the evaluation of individual techniques to integrate data from multiple sources. Some researchers have also focused on creating generic frameworks for data integration. Troyanskaya et al. (2003) developed a *meta* framework for integration of diverse sources of data. We call it *meta* because it doesn't directly integrate the datasets but uses results from other techniques like clustering algorithms and combines them with other evidence. Their proposed framework is known as MAGIC (Multisource Association of Genes by Integration of Clusters) and is based on a Bayesian network whose conditional probability tables have been built with the advice of yeast genetic experts. Given a pair of genes, it outputs the probability that they are functionally related after weighing the evidences from various sources. Evaluation of the predictions from the system is done using gene ontology data.

Most of the techniques that we have described work well for real (numerical) data but are less effective when dealing with string data, for example gene sequences, or graph data such as protein interactions. In many cases ad-hoc techniques have been deployed. In an approach to this problem, Lanckriet et al. (2004a) have proposed a framework where such diverse data could be merged in a principled manner. It is based on kernel methods (Shawe-Taylor and Cristianini, 2004) in which algorithms work on kernel matrices that are derived from pairwise similarity among variables using kernel functions. If a valid kernel function can be defined to encode the similarity between two variables, then the methods are applicable regardless of the different types of data - strings, vectorial or graphical - being used. This framework provides a means to integrate more diverse types of data as and when they become available in the future. The original paper proposed the framework only for supervised learning but extensions to unsupervised learning are possible.

# Chapter 3

# Data Integration for Regulatory Module Discovery

## 3.1 Introduction

As discussed in the first chapter, a transcriptional regulatory module is a set of genes that is regulated by a common set of TFs. A considerable amount of work has been carried out to determine the network of inter-relationships between these regulatory modules, with the aim of understanding how they act together in order to carry out the complex biological functions of a cell. Microarrays allow us to study a large proportion of genome expression simultaneously. These expression data have been used to build models of the regulatory networks. In some of the recent work, *integrative genomics*, in which data from experiments relating to different conditions or even different organisms are merged together, has been suggested as a method to discover these regulatory modules (Tanay et al., 2005; Segal et al., 2004). The approach compensates for the fact that the models have a very large number of variables (genes) whereas the number of repeats of microarray experiments is typically quite small. Thus, for a typical single experiment, there is not enough data to model the regulatory modules reliably. Another justification for the integrative approach is that, in evolution, many genes are believed to have similar roles in different organisms and so collective analysis should help to counter experimental error in individual experiments.

The motivation behind the research presented in this chapter is that, in our view, *blind* integrative genomics can lead to misleading or incorrect results. Researchers conduct experiments with clear objectives in mind. For example, research conducted on yeast to study meiosis will profile gene expression in sporulation media and will have clear meiotic signal in the data. But, hardly any of these conditions will be in common with the experiments related to stress conditions on yeast. Integration should be used when we are *only* concerned about either the background patterns or the most dominant patterns, and are not interested in patterns that may be visible in certain individual experiments. The global regulatory module network is the sum of smaller local regulatory module networks and by taking the integrative approach we run the risk that significant information from individual networks will be masked by the pooling process.

We believe that integrative genomics can sometimes be a useful technique. Our hypothesis is that as microarrays from different experimental conditions but same experiment type, for example stress, are merged, we should be able to readily identify stress specific regulatory modules. The clusters of co-regulated genes obtained should reinforce the local (stress specific) regulatory modules while suppressing the noise. By contrast, when microarrays from various different experiment types are merged together then the local (dataset specific) regulatory modules would be masked by conflicting patterns from other diverse datasets. Only sets of the genes that are strongly expressed among the majority of the conditions for which datasets have been mixed would be observed to behave in a consistent manner while the other genes would be expressed in an unpredictable manner. This should result in regulatory modules that are not very similar to the modules obtained with the original datasets.

One possible way of determining the similarity between datasets is using a statistical measure like the Kullback-Leibler divergence between the distribution of the different datasets (Wit and McClure, 2004). We computed the difference in data distributions of various distinct and progressively mixed microarray datasets as discussed later. However, the problem with Kullback-Leibler divergence or other statistical methods is that such a theoretical measure is not guaranteed to be a good indicator of *functional* similarity. They don't say much about how the final clusters are affected when we mix datasets. Therefore we validated our results by taking the functional route and studying the eventual effects directly by calculating

the similarity among the resulting modules. We carried out experiments in which we obtained regulatory modules from various datasets and their mixtures and then measure their similarities to each other. In this way, we show that progressive mixing of data of differing types inhibits the discovery of local regulatory modules at the cost of more dominant regulatory modules.

For our experiments, we chose to use the Module Networks algorithm (Segal et al., 2003) (refer Chapter-2), which is a well established approach and has had recognised success in finding biologically relevant modules. For measuring the similarities among the regulatory gene modules resulting from this algorithm, we chose to use the *modified Rand Index* (Hubert and Arabie, 1985) which has been shown to be a very stable measure of partition similarity. To our knowledge, there hasn't been any thorough study investigating the effects of mixing diverse datasets on the resulting clusters. We also haven't seen any research on understanding the correlation between theoretical data distributions and its impact on the resulting clusters.

## 3.2 Methodology

In order to validate our hypothesis, we chose to work with two very diverse datasets compiled from yeast experiments. One of them is from experiments to study the gene expression when yeast is exposed to stress conditions. The other dataset was from the study of cell-cycle related genes in which the pattern of activity is very different from the previous one. The expression of genes when stress conditions are created is much more drastic (both repressed and induced genes) than the normal cell-cycle, where optimal conditions are created for growth and the expression levels are much smaller. We want to show how progressive dilution of these two datasets with other dissimilar datasets affects the similarity among resulting clusters. We would expect that, as we dilute the datasets, the resulting cluster similarity to the original dataset cluster decreases.

All the data used in our analysis were taken from the SMD (Sherlock and Hernandez-Boussard, 2001) which hosts c-DNA microarray data-sets from various experimenters. We decided to focus our study on yeast as the regulatory mechanisms in more complex organisms are more involved and yeast has been studied extensively in recent years. We started with analysing data by individual researchers for experiments

related to stress. In particular, we used data from Gasch et al. (2000) submitted by A.P. Gasch which we refer to as DS-STRESS1 (76 microarrays), Saldanha et al. (2004) called DS-STRESS2 (49 microarrays) and Gasch et al. (2000) submitted by P. Spellman called DS-STRESS3 (41 microarrays). We merged all 183 stress related microarray slides available (not only the above three) to create the data set that we refer to as DS-STRESS. To compare these clustering against an entirely different category, we took 93 microarray data sets for cell-cycle experiments (Spellman et al., 1998) referred to as DS-CCYCLE. A further mixing of both stress and cell-cycle data created a data set that was named DS-STRESS-CCYCLE (276 microarrays). Finally, we pooled all the available data (523 microarrays) for yeast (not only stress and cell-cycle) and named it DS-ALL. The clusters resulting from all these different data groupings were compared. In order to have a reference point to compare the similarity values, we also generated a random microarray dataset for all the genes by sampling random numbers from a Gaussian distribution with zero mean and unit standard deviation. This dataset was named DS-RANDOM.

To analyze the data we did some standard pre-processing to the datasets. We used the $log_2$ of the ratio of the mean of Channel 2 (experimental expression) to the mean of Channel 1 (control expression). We also used total intensity normalization which is based on the assumption that the average log ratio on the array should be zero. Having organised the data, we did three types of studies with the following further processing.

1. We took all the data described above without further processing.

2. We filtered the genes by choosing those where log(base2) ratio had changed by two fold at least once among all experiments. This retains only those genes that have shown significant change in their expression.

3. We scale normalized each of the above data-sets (without any filtering) across slides to account for different experimental conditions or different data-sets by using a variant of median absolute deviation (MAD) (Yang et al., 2002). Note that this is different from normalizing microarray replicates for removing noise. The data that we are using has already been normalized to account for that. MAD is a measure of statistical dispersion and is a more robust estimator of scale than the sample variance or standard deviation. MAD is unaffected by the magnitude of the distances of a small number of outliers whereas in the

standard deviation, the distances from the mean are squared, therefore, large deviations are weighted more heavily, and thus outliers can heavily influence it. For our computations, the MAD for the $i_{th}$ slide (or condition) is given by

$$MAD_i = median_j\{\mid M_{ij} - median_j(M_{ij}) \mid\} \tag{3.1}$$

where $j$ ranges across all the genes, $i$ ranges across all the slides (conditions) and $M_{ij}$ is the individual microarray value. We compute the MAD value for all the slides and then normalize this score for each slide by taking into account other slides.

$$a_i = \frac{MAD_i}{\sqrt[I]{\Pi_{i=1}^{I} MAD_i}} \tag{3.2}$$

$$M_{ij}(final) = \frac{M_{ij}}{a_i^2} \tag{3.3}$$

As suggested by the authors, we have divided each value for slide $i$ by $a_i^2$. This normalization takes into account the MADs for all slides and has been found as a reliable way to normalize variation across slides in microarray studies (Yang et al., 2002).

### 3.2.1 Kullback Leibler divergence among datasets

Before we start computing the cluster similarity among datasets, we first need to understand how different (quantitatively) the underlying data distributions are. For this, we have used the KL divergence which is a statistical measure to compare distributions.

For two distributions F and G with densities f and g respectively, the KL divergence between them is defined as

$$d_{KL}(F,G) = \int_{-\infty}^{\infty} f(x) log \frac{f(x)}{g(x)} dx \tag{3.4}$$

However, KL divergence is not a distance measure as it lacks symmetry ($(d_{KL}(F,G) \neq d_{KL}(G,F))$), i.e., the KL divergence from F to G is not the same as from G to F.

There have been some suggestions regarding how to make it symmetric. Jeffreys (1946) suggest that it should be modified to

$$d_{Jeffreys}(F,G) = \frac{d_{KL}(F,G) + d_{KL}(G,F)}{2} \tag{3.5}$$

which is the mean of both the values. Johnson and Sinanovi (2001) suggest the harmonic mean and name it as resistor-average mean

$$\frac{1}{d_{Resistor}(F,G)} = \frac{1}{d_{KL}(F,G)} + \frac{1}{d_{KL}(G,F)} \tag{3.6}$$

We have used the $d_{Jeffreys}$ to validate our results. Determining the underlying distribution of microarray data is another challenge, and more so with so few replicates. We have assumed that the gene expression data across slides (experiments or conditions) has a Gaussian distribution following Wit and McClure (2004) who have found that after a logarithmic transformation, most microarray gene expression values across slides have a Gaussian distribution. Another key benefit is that a Gaussian distribution yields an analytically simple and elegant computable form. If $r_F$ and $r_G$ replicate arrays are spotted for conditions F and G and p is the total number of genes then the two microarray datasets (F and G) could be represented as

$$f_{ij} : i = 1, \ldots p, j = 1, \ldots, r_F$$
$$g_{ij} : i = 1, \ldots p, j = 1, \ldots, r_G$$

and the empirical KL divergence under the assumption of normality (Gaussian distribution) can be calculated as (Wit and McClure, 2004)

$$\hat{d}_{KL}(F,G) = \sum_{i=1}^{p} \left[ log\frac{\hat{\sigma}_{gi}}{\hat{\sigma}_{fi}} + \frac{1}{2}\left(\frac{\hat{\sigma}^2_{fi}}{\hat{\sigma}^2_{gi}} + \frac{(\hat{\mu}_{gi} - \hat{\mu}_{fi})^2}{\hat{\sigma}^2_{gi}} - 1\right)\right]$$

where

$$\hat{\mu}_{fi} = \frac{1}{r_F}\sum_{j=1}^{r_F} f_{ij} \tag{3.7}$$

$$\hat{\sigma}_{fi} = \frac{1}{r_F - 1}\sum_{j=1}^{r_F}(f_{ij} - \hat{\mu}_{fi})^2 \tag{3.8}$$

are the sample mean and variance of the observations associated with the $i_{th}$ gene. It should be noted that this assumes that each of the genes have a univariate Gaussian distribution and they are independent of each other. In essence we are computing pairwise KL divergence between corresponding gene replicate data from two different experiments.

### 3.2.2 Cluster similarity

For clustering, we have used the software package Genomica (Segal et al., 2003) which has been provided by the authors of the Module Network algorithm. The reason we chose this algorithm was because it has been shown in literature to identify biologically meaningful clusters. Its clustering process is driven by the expression of known TFs. The algorithm works as follows: given a gene expression dataset and a precompiled set of candidate regulatory genes, it simultaneously searches for a partition of genes into modules, and for a regulation program for each module that explains the behaviour of the genes in the module. It uses an EM approach to do the search. For each module, the procedure searches for a regulation program that provides the best prediction of expression profiles of the genes in the module as a function of the expression of a subset of genes from the candidate regulator set. The approach is iterative and runs till convergence, refining both the regulation program and the gene modules in each iteration.

This algorithm, apart from microarray data, also requires a list of TFs as prior knowledge on which to base the clustering. Our TFs were taken from the Yeastract database (Teixeira et al., 2006)*.

Since we are comparing the results on different data-sets our goal is to check the closeness of these resulting clusters (on different data-sets). This closeness was

*using their web interface `http://yeastract.com/` in Sept. 2006 when it had 145 TFs

validated using *cluster similarity* as described in the next section.

### 3.2.3 Cluster similarity indices

In order to compare the clusterings obtained on the different datasets, we need a measure of similarity. We have chosen a well established measure of clustering similarity - the *adjusted Rand's Index* - which was proposed by Hubert and Arabie (1985).

The Rand's index works on the concept of pair-wise matching on each of the cluster sets that are being compared. Given a set of objects of cardinality $n$, $S = s_1, ..., s_n$, suppose we obtain two clusterings $C1$ and $C2$ such that $C1 = c1_1, ..., c1_k$ and $C2 = c2_1, ..., c2_k$ where $\bigcup_{i=1}^{k} c1_i = S = \bigcup_{j=1}^{k} c2_j$. If:

$N_{11}$ = number of pairs of objects in the same cluster in both C1 and C2

$N_{00}$ = number of pairs of objects in different clusters in both C1 and C2

$N_{01}$ = number of pairs of objects in different clusters in C1 but same cluster in C2

$N_{10}$ = number of pairs of objects in the same cluster in C1 but different clusters in C2

then *agreement(A)* is the sum of $N_{11}$ and $N_{00}$ and *disagreement(D)* is the sum of $N_{01}$ and $N_{10}$. $A$ is the sum of pairs where both clusterings agree and $D$, where both disagree. The Rand's index is simply the fraction in *agreement* to the total (*agreement* + *disagreement*), i.e.,

$$R_{C1C2} = \frac{(N_{11} + N_{00})}{(N_{11} + N_{00} + N_{01} + N_{10})}$$

and its value lies between 0 and 1. When the two partitions are identical, the Rand's index is 1. It falls to 0 when the two clusters have nothing in common. The biggest drawback of this index is that it doesn't have a good spread of values. Another problem with the Rand's index is that the *expected* value of two random partitions does not take a constant value. This is an expected statistical property of any good index to compare clusterings. The modified version of the Rand's index - also known

| Datasets | Cluster Similarity Index | |
|---|---|---|
| | Rand's | *adjusted* Rand's |
| DS-STRESS3 & DS-RANDOM | 0.945 | 0.001 |
| DS-STRESS3 & DS-CCYCLE | 0.946 | 0.100 |
| DS-STRESS3 & DS-STRESS3 (different runs) | 0.957 | 0.453 |

Table 3.1: Comparison of Rand's Index and *adjusted* Rand's Index. Spread of values by Rand's index is skewed while the *adjusted* Rand's index shows a very wide spread of values.

as *adjusted* or *modified Rand's index* corrects for this by assuming the general form

$$R_{C1C2adj} = \frac{\text{index value - expected(index value)}}{\text{maximum(index value) - expected(index value)}}$$

For a detailed derivation of the analytical form of this equation, please refer Hubert and Arabie (1985).

Its maximum value is 1 and its expected value in the case of random clusters is 0. Based on an extensive empirical comparison of several such measures, Milligan and Cooper (1985) recommended this index as the best measure of agreement even when comparing partitions having different numbers of clusters. We did a comparison of Rand's and *adjusted* Rand's index on our clustering results. Table-3.1 shows that the spread of values by Rand's index is skewed (all the values seem to be above 0.94). On the other hand, the *adjusted* Rand's index shows a very wide spread of values. Based on this justification, we chose to use it for all our cluster comparisons. Our whole methodology is summarised in Algorithm-1.

1: Pre-process the datasets using the filtering and normalization steps discussed earlier to create 3 separate datasets (normal, filtered and scale normalized).
2: Process (mix) datasets so that they contain data from progressively diverse microarray experiments.
3: Run clustering algorithm over each of these resulting datasets.
4: Calculate cluster similarity among the resulting sets of clusters.
5: Compute KL divergence among all these datasets.
6: Compute the correlation coefficient between the cluster similarities and KL divergences.

**Algorithm 1:** Summary of methodology

## 3.3 Results

### 3.3.1 Cluster similarity among datasets

The results of cluster similarity computations are in Tables-3.2, 3.3, and 3.4. The clustering algorithm groups together *functionally* similar genes. By using the modified Rand index, we measure the similarity among the resulting sets of clusters. We ran the clustering algorithm 4 times for each dataset and have reported the mean values of all these runs. This was done because the initialization of the clustering algorithm is done by another non-deterministic algorithm. We observed that the final results did not have significant standard deviation (values in brackets in Table-3.2(a)). For clarity of presentation, we have not reported standard deviation values in remaining tables.

We first compared the individual stress datasets to each other to compare how similar the various runs of the same dataset are as well as to other stress datasets. We then compared each of the stress datasets against DS-STRESS, DS-STRESS-CCYCLE, DS-ALL, DS-CCYCLE which are increasingly distant from the stress datasets as described earlier. As a reference, we also compared them against DS-RANDOM which is a randomly generated dataset and gives us a baseline against which to compare the rest of the similarity values.

The values in Table-3.2(a), 3.3(a), and 3.4(a) indicate the level of similarity among the same type of datasets. The results in all three suggest that even among datasets of the same type, e.g. stress, there is considerable variation in similarity values, for example DS-STRESS1 and DS-STRESS3 are much more similar to each other than to DS-STRESS2 which could be explained from the fact that they were done under similar conditions. We also observe that DS-STRESS1 is slightly more similar to DS-STRESS2 than DS-STRESS3 across all three classes. Another interesting observation is that DS-STRESS1 is more similar to DS-CCYCLE than DS-STRESS2 pointing to the fact that there could be large variations in results even under similar conditions.

Results in Table-3.2(b) show the expected trend. As diverse datasets are merged, the similarity of the resulting clusters to the original dataset falls progressively. We observe that all three stress datasets are most similar to the combined stress

(a) Cluster similarity among stress datasets

| | DS-STRESS1 | DS-STRESS2 | DS-STRESS3 |
|---|---|---|---|
| DS-STRESS1 | 0.5420 (0.0161) | 0.0261 (0.0024) | 0.0709 (0.0024) |
| DS-STRESS2 | - | 0.5070 (0.0162) | 0.0213 (0.0014) |
| DS-STRESS3 | - | - | 0.5227 (0.0483) |

(b) Comparison of clustering of individual stress datasets versus progressively mixed datasets

| | DS-STRESS | DS-STRESS-CCYCLE | DS-ALL | DS-CCYCLE | DS-RANDOM |
|---|---|---|---|---|---|
| DS-STRESS1 | 0.1616 | 0.1368 | 0.1186 | 0.0354 | 0.0003 |
| DS-STRESS2 | 0.0606 | 0.0555 | 0.0528 | 0.0176 | 0.0001 |
| DS-STRESS3 | 0.1105 | 0.1109 | 0.0989 | 0.0309 | 0.0001 |

(c) Comparison of cell-cycle and stress to mixed data clustering

| | DS-STRESS | DS-CCYCLE | DS-STRESS-CCYCLE | DS-ALL | DS-RANDOM |
|---|---|---|---|---|---|
| DS-CCYCLE | 0.0418 | 0.4736 | 0.0783 | 0.0638 | 0.0007 |
| DS-STRESS | 0.5288 | 0.0418 | 0.2197 | 0.1784 | 0.0003 |

Table 3.2: Cluster similarity among full (non scale-normalized) datasets

(a) Cluster similarity among stress datasets

| | DS-STRESS1 | DS-STRESS2 | DS-STRESS3 |
|---|---|---|---|
| DS-STRESS1 | 0.6600 | 0.1747 | 0.2417 |
| DS-STRESS2 | - | 0.5500 | 0.1155 |
| DS-STRESS3 | - | - | 0.5933 |

(b) Comparison of clustering of individual stress datasets versus progressively mixed datasets

| | DS-STRESS | DS-STRESS-CCYCLE | DS-ALL | DS-CCYCLE | DS-RANDOM |
|---|---|---|---|---|---|
| DS-STRESS1 | 0.3425 | 0.3378 | 0.3434 | 0.0981 | 0.0037 |
| DS-STRESS2 | 0.1060 | 0.0920 | 0.0759 | 0.0252 | 0.0022 |
| DS-STRESS3 | 0.2470 | 0.2534 | 0.2325 | 0.0925 | 0.0023 |

(c) Comparison of cell-cycle and stress to mixed data clustering

| | DS-STRESS | DS-CCYCLE | DS-STRESS-CCYCLE | DS-ALL | DS-RANDOM |
|---|---|---|---|---|---|
| DS-CCYCLE | 0.0663 | 0.4768 | 0.0812 | 0.0614 | 0.00068 |
| DS-STRESS | 0.5986 | 0.0663 | 0.3067 | 0.2244 | 0.0013 |

Table 3.3: Cluster similarity among filtered (non scale-normalized) datasets

(a) Cluster similarity among stress datasets

| | DS-STRESS1 | DS-STRESS2 | DS-STRESS3 |
|---|---|---|---|
| DS-STRESS1 | 0.4446 | 0.0125 | 0.0340 |
| DS-STRESS2 | - | 0.4740 | 0.0086 |
| DS-STRESS3 | - | - | 0.4652 |

(b) Comparison of clustering of individual stress datasets versus progressively mixed datasets

| | DS-STRESS | DS-STRESS-CCYCLE | DS-ALL | DS-CCYCLE | DS-RANDOM |
|---|---|---|---|---|---|
| DS-STRESS1 | 0.0550 | 0.0468 | 0.0532 | 0.0070 | 0.0001 |
| DS-STRESS2 | 0.0186 | 0.0165 | 0.0192 | 0.0025 | 0.0003 |
| DS-STRESS3 | 0.0419 | 0.0345 | 0.0364 | 0.0053 | 0.0004 |

(c) Comparison of cell-cycle and stress to mixed data clustering

| | DS-STRESS | DS-CCYCLE | DS-STRESS-CCYCLE | DS-ALL | DS-RANDOM |
|---|---|---|---|---|---|
| DS-CCYCLE | 0.0068 | 0.5310 | 0.0117 | 0.0093 | 0.0008 |
| DS-STRESS | 0.4751 | 0.0068 | 0.0781 | 0.0623 | 0.0003 |

Table 3.4: Cluster similarity among full (scale-normalized) datasets

dataset (DS-STRESS). As the combined stress data is mixed with cell-cycle data, the similarity value falls. Since DS-ALL is even more diluted in stress data (it combines even more non-stress data), the similarity value has fallen further. All the stress datasets' similarity to DS-CCYCLE is very low, as we expected because of very different nature of expression in these diverse experiments. The similarity values for the random data-set are near zero in all the cases. Another interesting observation is that stress dataset seem to be very dominant in the final mixture (DS-ALL) as we see that the similarity values haven't fallen a lot between DS-STRESS and DS-ALL. Across all three stress datasets, we observe that the similarity values across DS-STRESS, DS-STRESS-CCYCLE and DS-ALL are much closer than the rest.

Table-3.2(c) shows the results at a more macro level using all stress data and all cell-cycle data. These results generalise and substantiate our earlier observations as the same trends are more robust here because of aggregation of data. Again we observe that similarity values across DS-STRESS, DS-STRESS-CCYCLE and DS-ALL are much closer than the rest.

When we filtered the datasets, retaining only those genes that changed their expression value by two fold at least once, the results, shown in Table-3.3 are somewhat different from the previous study. We observe that the similarity values are much higher when compared to the earlier unnormalized datasets. An explanation for this is that as we have retained only genes that are highly expressed, the total number of genes is smaller, and the uncorrelated background expression has been reduced. Results in Table-3.3(a) show similar trend as the earlier one where DS-STRESS1 and DS-STRESS3 are much more similar to each other than to DS-STRESS2. DS-STRESS1 is also slightly more similar to DS-STRESS2 than DS-STRESS3. Like the previous study, the similarity values in Table-3.3(b) indicate that DS-STRESS, DS-STRESS-CCYCLE and DS-ALL are quite similar to each other. The similarity values are in similar range and sometimes the trend is not very sharply delineated among them.

A further look at Table-3.3(c) indicates that the cell-cycle data is almost equally dissimilar from each of the mixes (DS-STRESS, DS-STRESS-CCYCLE, and DS-ALL) like the previous study. This indicates that the stress data is somehow dominating other data in the combined datasets. The combined stress data is showing the expected trend as its similarity values are falling as more diverse data is mixed.

Our third study used scale normalization (with MAD) in order to bring all the expression values across various different experimental conditions into the same range. We chose to do this after observing that the stress expression values fall in a much larger range than the cell cycle values, which might explain the disparate similarity values found when the datasets were merged together in the previous two studies. After scale normalization, we got some interesting results as shown in Table-3.4. The similarity values fell considerably, especially among different types of datasets. We believe that the reason for this is that as the range of expression values have been brought to a similar scale, the clustering algorithm is not able to identify dominant clusters as clearly as earlier.

Table-3.4(a) show similar trend as the earlier one where DS-STRESS1 and DS-STRESS3 are much more similar to each other than to DS-STRESS2. DS-STRESS1 is again slightly more similar to DS-STRESS2 than DS-STRESS3. Based on the results so far, it is possible that the DS-STRESS2 is not representative of the combined stress behaviour indicating that the data may be of lower quality.

Table-3.4(b) shows that DS-STRESS, DS-STRESS-CCYCLE and DS-ALL are again quite similar to each other as the similarity values are in similar range and sometimes the trend is not very sharply delineated among them. Table-3.4(c) shows similar trends as seen previously.

### 3.3.2 KL divergence among datasets

The results for KL divergence computations using Jeffreys' adjustment (refer eqn-3.5) among various datasets are shown in Tables-3.5, 3.6, and 3.7. Table-3.5 shows the KL divergence among the non-scale normalized datasets. Table-3.6 has the results for filtered (non-scale normalized) datasets while Table-3.7 has the results for the scale-normalized datasets. While cluster similarity was used for *functional* difference among datasets, these denote the *theoretical* difference among the datasets by computing the KL divergence between the underlying distributions. One thing to note is that these are *distances* while the values in the cluster similarity section were *similarity* values. Therefore, smaller KL-divergence indicates more similarity.

Like the computations using cluster similarity, we first compared the individual stress datasets to each other to study their similarity. The values in Table-3.5(a),

(a) KL Divergence among stress datasets

| | DS-STRESS1 | DS-STRESS2 | DS-STRESS3 |
|---|---|---|---|
| DS-STRESS1 | 0 | 4843.3 | 1329.0 |
| DS-STRESS2 | - | 0 | 4712.3 |
| DS-STRESS3 | - | - | 0 |

(b) KL Divergence among individual stress datasets versus progressively mixed datasets

| | DS-STRESS | DS-STRESS-CCYCLE | DS-ALL | DS-CCYCLE |
|---|---|---|---|---|
| DS-STRESS1 | 1066.2 | 897.29 | 1131.0 | 1885.25 |
| DS-STRESS2 | 1226.6 | 1598.2 | 1646.0 | 5116.5 |
| DS-STRESS3 | 1139.5 | 1039.2 | 1234.1 | 2747.0 |

(c) KL Divergence among cell-cycle, stress and mixed datasets

| | DS-STRESS | DS-CCYCLE | DS-STRESS-CCYCLE | DS-ALL |
|---|---|---|---|---|
| DS-CCYCLE | 2186.5 | 0 | 1190.7 | 1805.8 |
| DS-STRESS | 0 | 2186.5 | 125.5 | 194.5 |

Table 3.5: KL Divergence among full (non scale-normalized) datasets

(a) KL Divergence among stress datasets

| | DS-STRESS1 | DS-STRESS2 | DS-STRESS3 |
|---|---|---|---|
| DS-STRESS1 | 0 | 475.5 | 242.7 |
| DS-STRESS2 | - | 0 | 367.9 |
| DS-STRESS3 | - | - | 0 |

(b) KL Divergence among individual stress datasets versus progressively mixed datasets

| | DS-STRESS | DS-STRESS-CCYCLE | DS-ALL | DS-CCYCLE |
|---|---|---|---|---|
| DS-STRESS1 | 130.60 | 121.20 | 129.30 | 217.50 |
| DS-STRESS2 | 670.75 | 952.40 | 1011.20 | 638.40 |
| DS-STRESS3 | 132.10 | 226.65 | 199.20 | 380.70 |

(c) KL Divergence among cell-cycle, stress and mixed datasets

| | DS-STRESS | DS-CCYCLE | DS-STRESS-CCYCLE | DS-ALL |
|---|---|---|---|---|
| DS-CCYCLE | 340.5 | 0.0 | 198.3 | 297.5 |
| DS-STRESS | 0.0 | 340.5 | 92.0 | 147.7 |

Table 3.6: KL Divergence among filtered (non scale-normalized) datasets

(a) KL Divergence among stress datasets

| | DS-STRESS1 | DS-STRESS2 | DS-STRESS3 |
|---|---|---|---|
| DS-STRESS1 | 0 | 4960.7 | 2116.05 |
| DS-STRESS2 | - | 0 | 4390.90 |
| DS-STRESS3 | - | - | 0 |

(b) KL Divergence among individual stress datasets versus progressively mixed datasets

| | DS-STRESS | DS-STRESS-CCYCLE | DS-ALL | DS-CCYCLE |
|---|---|---|---|---|
| DS-STRESS1 | 2885.85 | 2451.55 | 3819.50 | 3179.65 |
| DS-STRESS2 | 1179.95 | 1203.85 | 1501.85 | 7768.15 |
| DS-STRESS3 | 2244.15 | 1944.65 | 2804.85 | 5644.25 |

(c) KL Divergence among cell-cycle, stress and mixed datasets

| | DS-STRESS | DS-CCYCLE | DS-STRESS-CCYCLE | DS-ALL |
|---|---|---|---|---|
| DS-CCYCLE | 6818.35 | 0.00 | 4947.10 | 7625.75 |
| DS-STRESS | 0.00 | 6818.35 | 376.70 | 466.65 |

Table 3.7: KL Divergence among full (scale-normalized) datasets

3.6(a), and 3.7(a) indicate the level of similarity among the same type of datasets. Like the results of cluster similarity, the results in all three suggest that even among datasets of the same type, e.g. stress, there is considerable variation in similarity values, for example DS-STRESS1 and DS-STRESS3 are much more similar to each other than to DS-STRESS2 which could be explained from the fact that they were done under similar conditions. We also observe that DS-STRESS1 is slightly less similar to DS-STRESS2 than DS-STRESS3 across all three classes. This is in contrast to the observations in cluster similarity where DS-STRESS1 is slightly more similar to DS-STRESS2 than DS-STRESS3. It could be explained by the fact that clustering involves a number of steps that could loose information.

We then compared each of the stress datasets against DS-STRESS, DS-STRESS-CCYCLE, DS-ALL, DS-CCYCLE which are increasingly distant from the stress datasets as described earlier. As seen in Table-3.5(b), all the stress datasets' similarity to DS-CCYCLE is very low as we expected because of very different nature of expression in these diverse experiments. Again, as diverse datasets are merged, the similarity of the resulting clusters to the original dataset falls progressively. We again observe that this trend is not very distinct among DS-STRESS, DS-STRESS-CCYCLE and DS-ALL. Like previously, we attribute it to the fact that those datasets are quite similar. Table-3.5(c) shows the results of the tests at a more macro level using all stress data and all cell-cycle data. These results generalise and substantiate our earlier observations as the trends are more robust here because of aggregation of data. This table has another very interesting result. The values for divergence between DS-STRESS and DS-STRESS-CCYCLE and DS-ALL are extremely low (125.5 and 194.5). This means that they are very close from the perspective of their data distribution. This could be the reason why other datasets' similarity to them were in close range and many times overlapping among them.

When we filtered the datasets, retaining only those genes that changed their expression value by two fold at least once, the results, shown in Table-3.6 are again (like cluster similarity) different from the unnormalized data as the similarity values are much higher when compared to the unnormalized datasets. Results in Table-3.6(a) show similar trends to the earlier one where DS-STRESS1 and DS-STRESS3 are much more similar to each other than to DS-STRESS2. DS-STRESS1 is slightly less similar to DS-STRESS2 than DS-STRESS3. Like the previous study, the similarity values in Table-3.6(b) do not show a distinct trend. But Table-3.6(c) has the

| | Non-Scale normalized | Filtered | Scale normalized |
|---|---|---|---|
| DS-STRESS1 | -0.726 | -0.834 | -0.812 |
| DS-STRESS2 | -0.731 | -0.807 | -0.433 |
| DS-STRESS3 | -0.776 | -0.932 | -0.576 |
| DS-STRESS | -0.967 | -0.952 | -0.978 |
| DS-CCYCLE | -0.925 | -0.994 | -0.660 |

Table 3.8: Pearson's Correlation betwen KL divergence and Cluster similarity. There is a very strong correlation between the results of KL divergence among the datasets and the cluster similarities.

expected trend of falling similarity with increasingly diverse data. It also indicates that DS-STRESS, DS-STRESS-CCYCLE and DS-ALL are quite similar to each other. This reinforces the idea that the stress data is somehow dominating other data in the combined datasets.

Our third study used scale normalization (MAD) in order to bring all the expression values across various different experimental conditions into the same range. Like the results of cluster similarity, the similarity values fell considerably, specially among different types of datasets (Table-3.7). Again, we believe that the reason for this is that as the range of expression values have been brought to a similar scale, the clustering algorithm is not able to identify dominant clusters as clearly as earlier. Table-3.7(a) show similar trends to the earlier one where DS-STRESS1 and DS-STRESS3 are much more similar to each other than to DS-STRESS2. DS-STRESS3 is again slightly more similar to DS-STRESS2 than DS-STRESS1. Again the trend in Table-3.7(b) is not very distinct while Table-3.7(c) shows expected trend as seen previously.

### 3.3.3 Correlation between KL divergence and cluster similarity

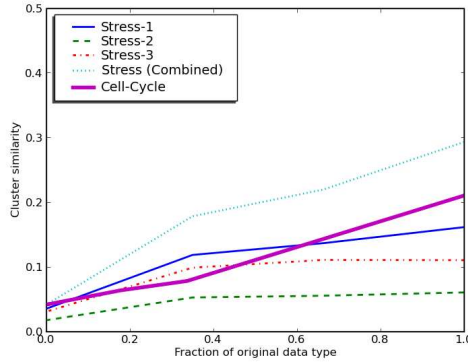
In order to correlate the findings of KL divergence computations with the cluster similarities, we computed the Pearson's Correlation among the corresponding KL divergence values and the cluster similarity values. For each of datasets, DS-STRESS1, DS-STRESS2 and DS-STRESS3, all the observations were combined into a single vector of observations. The results of correlation calculations are shown in Table-3.8. It is clear from the results, that for all the datasets, there is a very strong correlation

among the results of KL divergence among the datasets and the cluster similarities. The negative values only indicate that the correlation is negative which is expected because cluster similarity values are *similarity* while the KL divergence is a *distance*. If we convert both of them to either similarity or distance then the correlation would turn positive with the magnitude remaining the same. For unnormalised and filtered data, the correlation values are high, indicating strong correlation. The interesting pattern here is that as we move from full data to filtered data (both non scale-normalised) there is a slight improvement in the correlation values. This indicates that filtering of unwanted genes that mostly act as noise makes the datasets cleaner. On the other hand, after normalization, the values are not so strong. As we saw earlier in Table-3.4, even the similarity values had fallen after normalisation. This is the reason why the correlation is not strong here consistently.

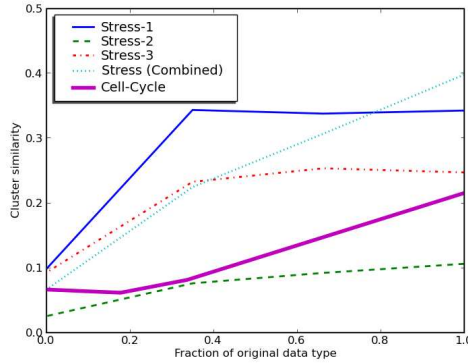
Based on the correlation results shown in Table-3.8, we consider cluster similarity to be an excellent indicator of dataset similarity. The cluster similarity values are in the range of 0 to 1 and we would like to propose cluster similarity as an index of microarray dataset similarity. It is easy to compute, unlike KL divergence. This index would be especially helpful for researchers who are anyways doing clustering as part of their analysis. This extra step would help them understand if various datasets that they are working on are compatible or not.

### 3.3.4 Effect of data heterogeneity

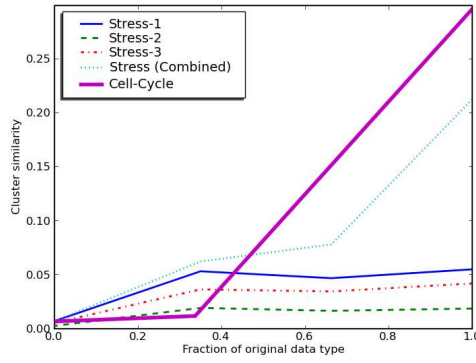
We also studied the variation of cluster similarity with the percentage of similar data as shown in Figure-3.1). For each mixed dataset, we calculated the fraction of original data type (e.g stress or cell-cycle) in the resulting mix. Similarity values were then plotted against these fractions. In all the figures, we see the general trend that as the fraction of the original datatype is increasing the cluster similarity values rises. We also observe that the combined datasets (both stress and cell-cycle) show a more consistent upward trend as compared to individual stress datasets. This is consistent with the observations we had in the earlier section and could be attributed to the dominance of stress data when mixed with other types. Figure-3.1(b) shows a gradually increasing trend though it's not as smooth as Figure-3.1(a) because of the increased dominance of the stress datasets when data is filtered. As seen from Figure-3.1(c), the general trend remains that similarity values increase with increasing




(a) Full (non scale-normalized) data




(b) Filtered (non scale-normalized) data




(c) Full (scale normalized data)

Figure 3.1: Variation of cluster similarity with data homogeneity. As the fraction of the original datatype is increasing the cluster similarity values rises.

fraction of similar data. These figures validate our hypothesis that we should be cautious towards integrating diverse datasets as they might be contributing to more noise and removing the original signals from the datasets. When we need to integrate different datasets, we should first compute their similarities and then only integrate similar ones while removing widely diverse ones.

## 3.4 Discussion

Learning the structure of genetic regulatory module networks has attracted a lot of attention in the past years. We saw a review of these techniques in Chapter-2. Recently, many researchers have focused on integrated approaches where they analyze a big compendium of microarrays gathered from various sources. Hughes et al. (2000) created a reference database or *compendium* of whole-genome microarray data for yeast from 300 diverse mutations and chemical treatments under similar growth conditions. They used this to identify the pathways perturbed by an uncharacterized mutation by computing the similarity of expression of the uncharacterized mutation to the ones in the compendium.

A similar compendium approach was followed by Tanay et al. (2005) who used data encompassing 1767 conditions from 60 different publications to find regulatory programs. The key difference from Hughes et al. (2000) is that while Hughes et al. (2000) had created the compendium from experiments under similar conditions in a single lab, Tanay et al. (2005) have used data from widely varying conditions. They followed the normalization methods suggested by individual authors to process the individual datasets. However, they did not do any combined (across all conditions) normalization to account for diversity across different datasets. This compendium was then used as a reference against which new data was compared. They used the SAMBA algorithm (refer Chapter-2) to transform all sources of information into generalized conditions (bi-partite graphs) and then analyzed them together. They also reported that stress data dominated their entire compendium because of extreme response of the organism to environmental stress.

Myers and Troyanskaya (2007) have also addressed the problem of heterogeneous data integration. In research that was carried out after the publication of our research (Mishra and Gillies, 2007), they measured context-dependent variation for a

wide variety of public genome data for yeast, including a large number of microarray, PPI and sequence datasets. Not surprisingly, one of their finding is that the quality of datasets varies dramatically and the degree to which we should trust any dataset depends on the process we are interested in. They have proposed a Bayesian approach to perform context-sensitive integration of data for protein network recovery.

We have observed in both cluster similarities and KL divergences that stress data, which has much much higher levels of change in expression, has *dominated* the final clusters when mixed with the cell-cycle data where the expression level changes are much lower. This is in line with the observations made by Tanay et al. (2005) in their large scale microarray integration study. They state that two opposite environmental stress responses dominate their entire compendium and the responses to stress are so strong and widespread that other, condition-specific regulatory programs are hard to detect without the combination of multiple studies and sensitive algorithms.

One source of error in our results might be attributed to the fact that our similarity index is based on pairwise matches of genes in each sets and even though the adjusted Rand's index is one of the most stable indices for cluster similarity, yet it's not perfect. Another drawback of our KL divergence computations between pairs of datasets is that we have assumed the covariance matrix is diagonal with no interactions among various genes. This assumption is a very naive one even though some researchers have found it to be quite useful (Wit and McClure, 2004) for practical purposes. The KL divergence between two *multivariate* Gaussian distributions $N_0(\mu_0, \Sigma_0)$ and $N_1(\mu_1, \Sigma_1)$ is given by (Kullback, 1997),

$$KL(N_0 \parallel N_1) = \frac{1}{2} \ln |\Sigma_1 \Sigma_0^{-1}| + \frac{1}{2} tr \Sigma_1^{-1}((\mu_0 - \mu_1)(\mu_0 - \mu_1)^T + \Sigma_0 - \Sigma_1) \qquad (3.9)$$

This involves estimating the covariance matrix. Because of the small number of experimental data available compared to the dimensionality of data, the resulting covariance matrix usually turns out to be singular and hence can not be inverted as required above. This forced us to use the independence assumption among genes which might have introduced some errors in KL divergence computations.

## 3.5 Conclusion

One of the original contributions of our work is that we have outlined an empirical technique to calculate functional similarity of datasets using the concept of cluster similarity. Based on its high correlation with underlying data distribution difference, we would like to propose it as an index of microarray dataset similarity. We have also showed that similarity values gradually fall with increasing fraction of dissimilar data. As argued in Orphanides and Reinberg (2002), all cellular regulatory mechanisms are very local in nature and trying to use a blind integrative approach is most likely going to prove futile in determining meaningful results. We have tried to establish this from a different point of view that as more diverse data-sets are merged then the similarity to individual data-sets (which have more local patterns) is reduced and the dominant ones overshadow the weaker signals. Therefore, before taking a blind integrative approach, much care should be taken to ensure that we mix only similar types of data. We should also be careful about the choice of normalization method. In our results we demonstrated that normalization can distort the data and affect the resulting clusters significantly.

The next chapter deals with data integration of a different type from that we have seen here. It deals with integration of different *types* of data and details a framework for that.

# Chapter 4

# Semi-supervised Regulatory Module Discovery

> "Just as the constant increase of entropy is the basic law of the universe, so it is the basic law of life to be ever more highly structured and to struggle against entropy" - *Vaclav Havel*

While our previous chapter discussed the impact of data integration of the same type (microarrays), in this chapter we focus on the integration of the other type, where datasets of different types are used in a cooperative manner. Different datasets are not being merged *literally*, but one is used to *guide* the clustering of the other. We propose a type of clustering method with *supervision* extracted from *prior biological knowledge* in order to guide the process of clustering.

When a small amount of prior knowledge is available in the form of *pairwise relationships* between genes, instead of simply using this knowledge for the external validation of the results of clustering, we can use it in order to guide the clustering process thus providing a limited form of supervision. We call these methods *semi-supervised clustering* * because unlike supervised or constrained clustering (Bradley et al., 2000), the final results of clustering is not required to enforce the constraints.

*This is different from Semi-Supervised Learning (Grira et al., 2005) which is a class of machine learning techniques that make use of both labelled and unlabelled data for training. They are called semi-supervised because the available knowledge is far from being enough for fully supervised learning even in a transductive form.

The constraints act as guidelines and are enforced only if they are complementary to the data being clustered. For semi-supervised clustering to be profitable, the two sources of information, i.e., the similarity measure (used by all clustering methods) as well as the constraints available should not completely contradict each other. In our novel formulation, named *semi-supervised spectral clustering* (SSSC), supervision (prior knowledge) is provided in the form of binary constraints and clustering is done in the spectral space (Shi and Malik, 2000; Ng et al., 2001). The prior knowledge is derived from DNA binding data from ChIP-chip experiments, PPI data and known TF-gene interactions from a curated database. These are used to guide the process of clustering microarray data.

## 4.1 Spectral Clustering

The goal of any clustering is to partition a set of points into disjoint sets where the points within a partition are as similar as possible while points within different partitions are as dissimilar as possible. In this section, we discuss how spectral clustering achieves this objective.

### 4.1.1 Graph notations

Given a set of data points, we can compute similarities between them using a suitable *similarity function*. Given these similarities between the data points, a dataset can be represented as a *graph* which is a set of *vertices* and *edges* connecting vertices *(V,E)*. The vertices *(V)* represent the data points while the edges *(E)* represent the links between the data points. Usually a certain threshold of similarity value is chosen above which the edges are linked between data points. For *weighted* graphs the edges also have the similarity values as *weights*. Once we have this undirected weighted graph, the goal of a clustering algorithm is to partition it such that edge weights between the points *within* a partition are high while the edge weights between points of *different* partitions are low. We begin with some definitions.

Let $G = (V, E)$ be an *undirected weighted* graph with vertices $V = v_1, \ldots, v_n$, edges $E = e_1, \ldots, e_n$ and each edge $e_{ij}$ between vertex $i$ and $j$ has a non-negative weight $w_{ij}$. The weights matrix $W = (w_{ij})_{i,j=1,\ldots,n}$ is also known as the *adjacency matrix*

of this graph. For non existing edges, $w_{ij} = 0$. The graph is assumed to have no self-edges, i.e., $w_{ii} = 0$. In order to understand spectral clustering we need some more definitions for this graph.

The *degree* ($d_i$) of a vertex $v_i$ is defined as

$$d_i = \sum_{j=1}^{n} w_{ij}$$

which intuitively is the row-wise sum for the respective row of the adjacency matrix. In the graph, this can also be understood in terms of sum of edge weights for that particular vertex. The *degree matrix* $D$ is defined as the *diagonal* matrix with the individual vertex degrees $d_1, \ldots, d_n$ along the diagonal, everything else being 0.

### 4.1.2 Similarity matrices and graph Laplacians

There are various ways of converting similarities between a given set of data points into a graph - both in choosing the similarity function to compute similarities among the data points as well as deciding about how to turn the similarity values into a graph.

A *$\varepsilon$-neighbourhood* graph is obtained by joining edges between points whose similarity values are larger than $\varepsilon$. Figure-4.1 shows two examples of such a graph using two different values of $\varepsilon$. A $k$-nearest neighbour (KNN) graph has edges between a point and $k$ other points that are most similar to it. This leads to a directed graph because the neighbourhood relation is not symmetric, i.e., $v_a$ might have $v_b$ as one of its k-nearest neighbours but the vice-versa might not be true. To convert this into an undirected one, we can either totally ignore the direction or take a more restrictive approach where two nodes are connected *only* if both of them are $k$-nearest neighbours of each other. The latter type is also referred to as a *mutual $k$-nearest neighbour graph*.

A *fully connected* graph is one where all pairs of points have positive similarity values and are connected. This leads to a denser graph in comparison to previous ones. While in the earlier ones, the local neighbourhood relationship was enforced with either a threshold ($\varepsilon$) or a maximum of $k$-neighbours, in a fully connected

graph we have to choose a similarity function that should do this. For all our work, we have used the Gaussian similarity function which encodes this neighbourhood relation automatically. In this function, $exp\left(-\frac{\|\mathbf{x}-\mathbf{x}^{'}\|^2}{2\sigma^2}\right)$, $\sigma$ controls the width of the neighbourhood and $\mathbf{x}$,$\mathbf{x}^{'}$ represent the two variables between which similarity is being computed. The Gaussian similarity function is an exponential function, therefore depending on $\sigma$, the similarity falls non-linearly (exponentially) with increasing distance. This property makes it desirable to use where neighbourhood relations are important. Various other similarity functions for vector data are discussed in Chapter-5.

Spectral clustering is based on the *Laplacian* matrix which has its origins in spectral graph theory (Chung, 1997). There are various types of Laplacians. All of these assume that we have an undirected graph $G$ with positive weight matrix $W$ ($w_{ij} \geq 0$) and a corresponding degree matrix $D$. An *unnormalized* Laplacian is defined as

$$L = D - W$$

The matrix $L$ has the following properties

- $L$ is always symmetric and positive semi-definite.
- $L$ has $n$ non-negative, real-valued eigenvalues $\lambda_1(=0) \leq \lambda_2 \leq \cdots \leq \lambda_n$. The number of smallest eigenvalues (=0), i.e., its multiplicity, corresponds to the number of connected components in the graph.

There are two popular variants of the *normalized* Laplacian. They are defined as

$$L_{symmetric} = D^{-1/2}LD^{-1/2} = I - D^{-1/2}WD^{-1/2} \quad (4.1)$$

$$L_{randomwalk} = D^{-1}L = I - D^{-1}W \quad (4.2)$$

where $D^{-1/2}$ is the inverse square root of matrix $D$. Since $D$ is a diagonal matrix and the square root of a diagonal matrix D is formed by taking the square root of all the entries on the diagonal

$$\mathbf{D}^{\frac{-1}{2}} = \begin{bmatrix} \frac{1}{\sqrt{d_{11}}} & 0 & \dots & 0 \\ 0 & \frac{1}{\sqrt{d_{22}}} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \frac{1}{\sqrt{d_{nn}}} \end{bmatrix}$$

Like the unnormalized Laplacian, the normalized ones are also always symmetric and positive semi-definite. They too have non-negative, real-valued eigenvalues $\lambda_1(= 0) \leq \lambda_2 \leq \cdots \leq \lambda_n$ and the multiplicity of the smallest eigenvalue is the number of connected components in the graph. Laplacians could be interpreted as *gradients* on graphs and it is related to differential geometry.

### 4.1.3 Graph clustering

As seen earlier, given a set of data points, a similarity function can be used to calculate the pairwise similarities among them, resulting in a similarity matrix. Given its graph representation, the clustering can be defined as a *graph partitioning* problem where the edges between the points of the *same* cluster have high weights while the edges between points belonging to *different* clusters have low weights. Before we discuss the algorithm in detail, we discuss the general problem of graph clustering and its relation to spectral clustering.

We know that the key objective of clustering is to find sets of points that are *maximally similar to each other within a set* and *maximally dissimilar to points in other sets*. If we have a similarity graph, as discussed earlier, the problem can be restated to find a partition of the given graph such that the edges between points within a partition have higher weights as compared to edges between points in different partitions. Graph clustering or partitioning is an old problem and has been exhaustively studied [†]. Spectral clustering can be derived as an approximation to the graph partitioning objectives (von Luxburg, 2006). Before we start, we need some definitions.

If we have two disjoint partitions A, B then

$$Cut(A, B) = \sum_{i \in A, j \in B} w_{ij}$$

[†]For a good review, see Schaeffer (2007)

So, if we have a graph G with adjacency matrix $W$, then we can construct a partition by solving the *min-cut* problem, which can be understood as choosing the partitions $A_1, \ldots, A_k$ such that we minimize the following cut.

$$Cut(A_1, \ldots, A_k) = \sum_{i=1}^{k} Cut(A_i, \bar{A}_i)$$

where $\bar{\text{A}}$ is the complement of $A$. While theoretically it can be solved, yet in practice it may yield clusters of size 1 (trivial clusters), which is not usually the goal of clustering. We want clusters that are *reasonably* big. This is specified in terms of two popular *objective functions* namely *RatioCut* and normalized cut or *NCut*.

$$RatioCut(A_1, \ldots, A_k) = \sum_{i=1}^{k} \frac{Cut(A_i, \bar{A}_i)}{|A_i|} \tag{4.3}$$

$$NCut(A_1, \ldots, A_k) = \sum_{i=1}^{k} \frac{Cut(A_i, \bar{A}_i)}{vol(A_i)} \tag{4.4}$$

While in RatioCut the normalizing factor is the $|A_i|$, which is the total number of vertices (points) in the $i_{th}$ partition, in NCut, the normalizing factor is $vol(A_i)$ which is the sum of all edge weights in the partition. The role of this normalizing factor is to make the partition *balanced* as measured by number of vertices or sum of edge weights. Even though the formulation is simple and elegant, it is an NP-hard[‡] problem. The spectral clustering algorithm is a way to solve a relaxed version of these objective functions.

Another way to understand this is that in both RatioCut and NCut, the numerator tries to achieve the objective of making different clusters as dissimilar to each other as possible, i.e., minimize the between cluster similarity. This is one half of the key requirement of any clustering algorithm's objective. The other half is that the within-cluster similarity should also be maximized. In other words $\sum_{i,j \in A} w_{ij}$ and $\sum_{i,j \in \bar{A}} w_{ij}$ should be maximized. Let's see how this is satisfied in each of the objective functions.

[‡]NP-hard (nondeterministic polynomial-time hard) is a class of problem in computational complexity theory

$$\sum_{i,j\in A} w_{ij} = \sum_{i\in A, j\in A\cup\bar{A}} w_{ij} - \sum_{i\in A, j\in\bar{A}} w_{ij} \quad (4.5)$$

$$= vol(A) - Cut(A,\bar{A}) \quad (4.6)$$

We can see that NCut satisfies this by maximizing $vol(A)$ and minimizing $Cut(A,\bar{A})$. RatioCut doesn't lead to this objective. As shown in von Luxburg (2006), relaxing NCut leads to using the *normalized* Laplacian in spectral clustering while relaxing RatioCut leads to the use of *unnormalized* Laplacian. Therefore, normalized spectral clustering satisfies both the key clustering criteria while unnormalized spectral clustering only implements the first criteria. One key point to note is that there is no guarantee on the quality of the clustering solution of the relaxed problem compared to the exact solution.

### 4.1.4 Algorithm explanation

Spectral clustering is a technique in which the eigenvectors of the Laplacian matrix (which is derived from the *similarity* matrix) corresponding to the smallest eigenvalues are used to derive a clustering of the given data points. The methods are called spectral, because they make use of the *spectrum*[§] of the graph. It has been applied to diverse domains, e.g. image segmentation (Shi and Malik, 2000; Weiss, 1999) and bioinformatics (Speer et al., 2005). Most spectral clustering algorithms can be considered to have three stages:

**Normalization** This consists of computing the similarity matrix from the raw data using a suitable similarity function. We call this the *normalization* step because different types of data (vector, graph or string) get converted to a common format (similarity matrix).

**Eigen Decomposition** This consists of computing the eigenvalues and the corresponding eigenvectors of the similarity matrix. This step could be considered as the mapping of original data to the spectral domain.

[§]The set of eigenvectors of the normalized Laplacian matrix is usually called the *spectrum* of the Laplacian (or the spectrum of the associated graph)

**Clustering** This step consists of using a traditional clustering algorithm (usually k-means) to cluster the vectors in the spectral domain.

Different spectral clustering algorithms differ in the number of eigenvectors used (single or many) as well as the type of Laplacian used (unnormalized or normalized). Verma and Meila (2003) did a systematic comparison of different popular spectral (Shi and Malik, 2000; Ng et al., 2001; Meila and Shi, 2000) and traditional clustering algorithms on artificial as well as real-world datasets. They report that spectral methods are more stable to noise than other tested algorithms. Both Shi and Malik (2000) and Meila and Shi (2000) have used the $L_{randomwalk}$ normalized Laplacian while Ng et al. (2001) have used the $L_{symmetric}$ normalized Laplacian. Apart from this, there is no major difference between the techniques of all these algorithms. Since Verma and Meila (2003) did not find significant differences among the two normalized Laplacians, we have used the algorithm by Ng et al. (2001) which is the most recent one. This algorithm is described in Algorithm-2.

**Require:** Dataset ($X$), number of clusters ($k$)

1: Calculate the symmetric similarity matrix $K_{n\times n}$ using Gaussian similarity function $K_{ij} = exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right)$ if $i \neq j$, and set $K_{ii} = 0$.
2: Calculate normalized Laplacian $L' = D^{-1/2}KD^{-1/2}$ where D is the diagonal matrix with $d_{jj} = \sum_i d_{ji}$
3: Find the eigenvectors $v^1, v^2, \ldots, v^k$ corresponding to the largest k eigenvalues of $L'$.
4: Use these k eigenvectors as columns to get $V_{n\times k}$.
5: Normalize the row to have unit norm, i.e., $U_{n\times k}$ such that $u_{ij} = v_{ij}/(\sum_k v_{ik}^2)^{\frac{1}{2}}$
6: Cluster the points representing the rows of this matrix $(u_i)_{i=1,\ldots,n}$ using k-means algorithm into k clusters, $C_1, C_2, \ldots, C_k$.
7: Output clusters $A_1, A_2, \ldots, A_k$ such that $A_i = \{x_j | u_j \in C_i\}$. This assigns the original point $x_j$ to cluster $A_i$ if $u_j$ is in cluster $C_i$.

**Algorithm 2:** Spectral clustering

In the previous description of this algorithm, we had mentioned that the eigenvectors corresponding to the smallest eigenvalues are used. However, in the step-3 of algorithm, we are proposing to take the eigenvectors corresponding to the largest k eigenvalues. This is because here we are using the Laplacian $L'$ instead of the form $L_{symmetric=}I - L'$ described earlier (refer eqn-4.1). This changes the eigenvalues from $\lambda_i$ to $1 - \lambda_i$.

To summarize the functioning of this algorithm: it changes the representation of the data points from the original space to the spectral space after the various steps (1-5) of transformation as shown in Algorithm-2. After that, any clustering algorithm can be used to cluster the data points. The reason for this transformation is that it allows better identification of non-linear clusters. Non-linear patterns are very hard to identify using traditional clustering methods but after the spatial transformation it becomes trivial to find them. For a detailed understanding of why this algorithm works, refer Ng et al. (2001).

Spectral clustering is very appealing because it yields a very standard linear algebra problem for which there are various efficient solvers (algorithm implementations) already available. It can be implemented for even large datasets if the similarity matrix is sparse. Unlike traditional clustering algorithms like k-means, there are no issues of dependency on starting point or getting stuck in local optimum. On the flip side, choosing the right similarity function and its parameters is non-trivial.

## 4.2 Datasets and Our Algorithm

### 4.2.1 Microarray datasets

We have used two popular microarray datasets on which the clustering is carried out, both of them based on experiments done on yeast (Saccharomyces cerevisiae). The dataset by Gasch et al. (2000) was obtained by exposing yeast to diverse environmental (stress) conditions such as temperature shocks, hydrogen peroxide, the superoxide generating drug menadione, the sulfhydryl-oxidizing agent diamide, the disulfide-reducing agent dithiothreitol, hyper and hypo-osmotic shock, amino-acid starvation and nitrogen source depletion. More than 900 genes showed drastic response to these environmental changes. We selected only those genes that displayed a change of three fold in *at least one experiment*. There were 1246 genes fulfilling this criterion. The assumption behind this selection strategy is that the majority of genes which do not show much change in their expression levels during a process are unrelated to it.

The second microarray dataset was based on cell-cycle experiments by Spellman et al. (1998) where the objective was to identify yeast genes which were involved

in cell-cycle regulation. This was achieved using DNA microarrays synchronized using three independent methods: $\alpha$ factor arrest, elutriation, and arrest of a cdc15 temperature sensitive mutant. Again, we selected only those genes that displayed a change of two fold in *at least one experiment*. There were 1732 genes fulfilling this criterion. The reason why we only filtered for two fold change is because stress leads to a much more widespread expression change across the genome. Therefore, the number of genes that show change at two fold are too high. On the other hand, expression level changes are not that severe in a normal cell-cycle study. So, the number of genes at two fold change are not that high.

In both the datasets, for data imputation we use the R *impute* package which uses *k-nearest neighbour* algorithm to impute missing values. It uses a Euclidean distance metric for finding nearest neighbours. We used the $log_2$ of the ratio of the mean of Channel 2 (experimental expression) to the mean of Channel 1 (control expression) since this is likely to create a Gaussian distribution (Wit and McClure, 2004).

For the experiments where we combined microarray with DNA-binding data, we needed to do other steps of pre-processing. We found the list of genes responsible for the TFs that were tested for binding. Then we ensured that our microarray dataset had those. The reasoning behind this is that some of them might be missing after the filtering step. In such a case, they were extracted from the original unfiltered dataset and incorporated in the final filtered one. The number of these is small compared to the total number of genes and we didn't want to loose any of them.

### 4.2.2 DNA-binding dataset

One of the datasets which we have used to guide the clustering process is the DNA-binding dataset on yeast (Harbison et al., 2004). It was created using genome-wide location analysis techniques to determine the genomic occupancy of 203 DNA-binding transcriptional factors. In this dataset, the likelihood of a particular TF binding to the promoter region of another gene is reported in terms of a confidence value (p-value). A lower p-value indicates higher confidence. In order to extract binary constraints from this dataset, we need some threshold on these reported p-values of interactions. We investigated a range of p-value cut-offs, each corresponding to a certain set of constraints. This was to study the impact of *number* and *quality* of constraints on the biological significance of clustering. Since these are

| p-value | Number of Constraints | | |
|---|---|---|---|
| | All Genes | Common Genes (Stress) | Common Genes (Cell-cycle) |
| 0.0001 | 2061 | 544 | 681 |
| 0.0005 | 3436 | 846 | 1032 |
| 0.001 | 4358 | 1053 | 1288 |
| 0.005 | 8562 | 1959 | 2442 |
| 0.01 | 12455 | 2776 | 3505 |
| 0.05 | 35917 | 7407 | 9713 |
| 0.1 | 63531 | 12579 | 17055 |

Table 4.1: Number of constraints from DNA-binding dataset at various p-value thresholds. As the p-value cutoff is increased, the number of constraints increases.

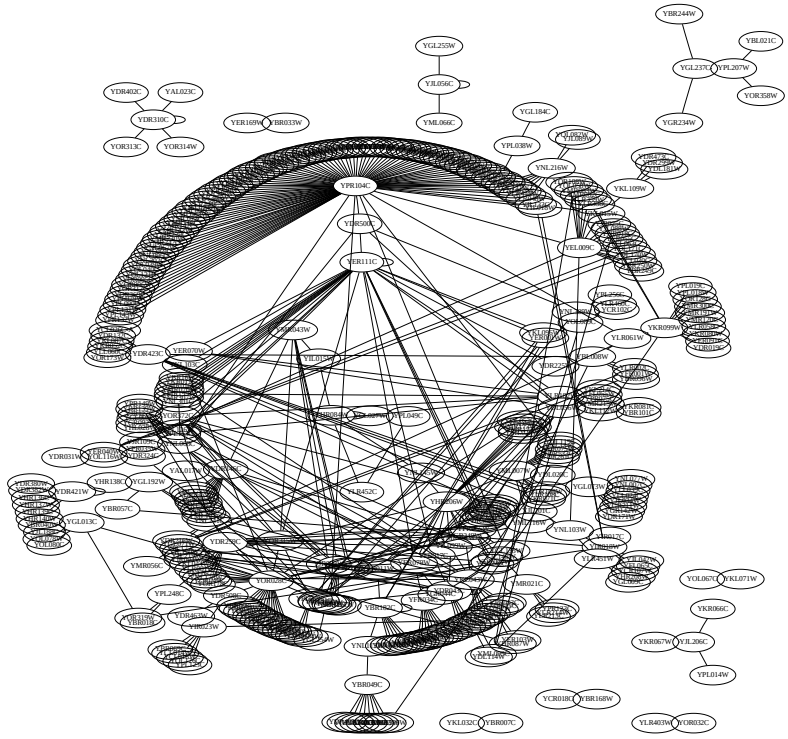
experimentally determined, we consider them as a more reliable evidence of genetic interaction. After extracting the constraints, we use them for guiding the clustering process.

As stated, we used the p-value thresholds to convert the confidence value data into binary constraints. For example, if the p-value threshold is 0.001 then all values below this are considered as *definitely bound* and hence assigned a value of 1. The rest are assigned 0 (not bound). A significant point to note is that these p-value cut-offs have a dual role. They determine the number of constraints as well as the quality of constraints. As the p-value cutoff is increased, the number of constraints increases, but a higher p-value also indicates lower confidence, hence the quality of the constraints falls. Table-4.1 shows the number of constraints corresponding to various p-value cut-offs. Figure-4.1 shows these constraints graphically where we can see that the graph density is very high at p=0.001 compared to at p=0.0001.

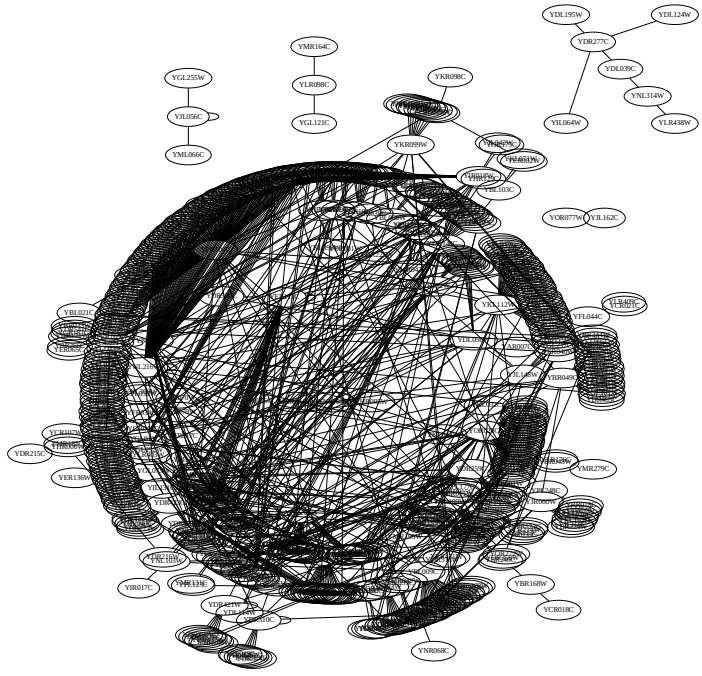
Our constraints are transformed into a $m \times n$ matrix where $m$ is the number of genes and $n$ is the number of TFs. This constraints matrix is used to modify the similarity matrix that we obtain from the microarray data as discussed in Section-4.2.5.

### 4.2.3 PPI dataset

Another source of constraints is the popular PPI dataset from MIPS Comprehensive Yeast Genome Database (CYGD) (Gueldener et al., 2006). It has been called a gold standard because of its quality and comprehensiveness (Yu et al., 2004). This dataset has information related to the proximity of proteins in yeast based on more



(a) Constraints at p=0.0001



(b) Constraints at p=0.001

Figure 4.1: Constraints derived from DNA-binding dataset at various p-value cutoffs

than 15446 protein-protein interaction records (9200 physical, 6400 genetic) which was compiled manually from the literature (3680 from single experiments) and published large-scale experiments. In addition to this, 268 manually extracted protein complexes as well as 783 complexes derived from large-scale experiments results in 87000 putative binary interactions.

### 4.2.4 TF-gene interactions dataset

We also derived constrains from an independently curated database of known TF-gene interactions known as YEASTRACT(Yeast Search for Transcriptional Regulators And Consensus Tracking). YEASTRACT¶ (Teixeira et al., 2006) is a curated repository of 34471 regulatory associations between transcription factors and target genes in Saccharomyces cerevisiae, based on more than 1000 bibliographic references. In this database, the curators consider interaction to have occurred when there is change in the expression of the target gene owing to the deletion (or mutation) of the transcription factor-encoding gene. They also consider evidence based on TF binding to the promoter region of the target gene based on band-shift, footprinting or chromatin immunoprecipitation assays. They also describe potential associations but we have not considered them as we wanted to stick to known facts.

### 4.2.5 Semi-supervised spectral clustering

We propose a semi supervised form of the spectral clustering method, which is detailed in Algorithm-3 and shown in Figure-4.2. We are clustering microarray data, hence the genes can be considered the variables and their pairwise similarity values are calculated using a Gaussian affinity function. We have chosen this similarity function because it naturally encodes the local neighbourhood property and its value falls rapidly as the pairwise dissimilarity increases. Once we have this similarity matrix, we use the constraints derived from our secondary datasets to modify it. Since our constraints already encode our belief about potential interactions, we set each value in the similarity matrix to 1 (maximum similarity) if there is a 1 in the corresponding constraints matrix. All other values are left unchanged as we have no information regarding them. The idea behind changing the values to represent

¶interactions file created on 25/12/2008

maximum similarity is to give the algorithm the maximum incentive to keep them in the same cluster. The resulting matrix is the final similarity matrix that we use for spectral clustering (Steps 3-7). We have used the algorithm suggested by Ng et al. (2001). The implementation was done in R using readily available libraries for linear algebra.

We calculate the normalized Laplacian and then find its eigenvalues. If we believe there are $k$ clusters then eigenvectors corresponding to the $k$ smallest eigenvalues are chosen. If we arrange these $k$ eigenvectors column-wise then we end up with a $n \times k$ matrix. Each row of this matrix is then normalized to have unit norm. Then we cluster the rows using the k-means clustering algorithm. For all these integrated matrices, the k-means clustering of the eigenvectors was started from fixed centres. These 50 centres, each representing a cluster, were the genes encoding the TFs that had the highest numbers of DNA-interactions in the DNA-binding dataset. The idea behind this choice is to guide the clustering process to start from meaningful positions rather than random ones. Again, in step-4 of algorithm, we are proposing to take the eigenvectors corresponding to the largest k eigenvalues. This is because here we are using the Laplacian $L'$ instead of the form $L_{symmetric=}I - L'$ described earlier (refer eqn-4.1). This changes the eigenvalues from $\lambda_i$ to $1 - \lambda_i$.

**Require:** Dataset $(X)$, number of clusters $(k)$

1: Calculate the symmetric similarity matrix $K_{n\times n}$ using Gaussian similarity function $K_{ij} = exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right)$ if $i \neq j$, and set $K_{ii} = 0$.
2: Use the constraints to modify $K, K_{final} = K \oplus C$ where C is the constraints matrix. $K \oplus C$ implies that we set $K_{i,j} = 1$ where $C_{i,j} = 1$.
3: Calculate the normalized Laplacian $L' = D^{-1/2}K_{final}D^{-1/2}$ where D is the diagonal matrix with $d_{jj} = \sum_i d_{ji}$
4: Find the eigenvectors $v^1, v^2, \ldots, v^k$ corresponding to the largest k eigenvalues of $L'$.
5: Use these k eigenvectors as columns to get $V_{n\times k}$.
6: Normalize the rows to have unit norm, i.e., $U_{n\times k}$ such that $u_{ij} = v_{ij}/(\sum_k v_{ik}^2)^{\frac{1}{2}}$
7: Cluster the points representing the rows of this matrix $(u_i)_{i=1,\ldots,n}$ using k-means algorithm into k clusters, $C_1, C_2, \ldots, C_k$.
8: Output clusters $A_1, A_2, \ldots, A_k$ such that $A_i = \{x_j | u_j \in C_i\}$. This assigns the original point $x_j$ to cluster $A_i$ if $u_j$ is in cluster $C_i$.

**Algorithm 3:** Semi-supervised spectral clustering

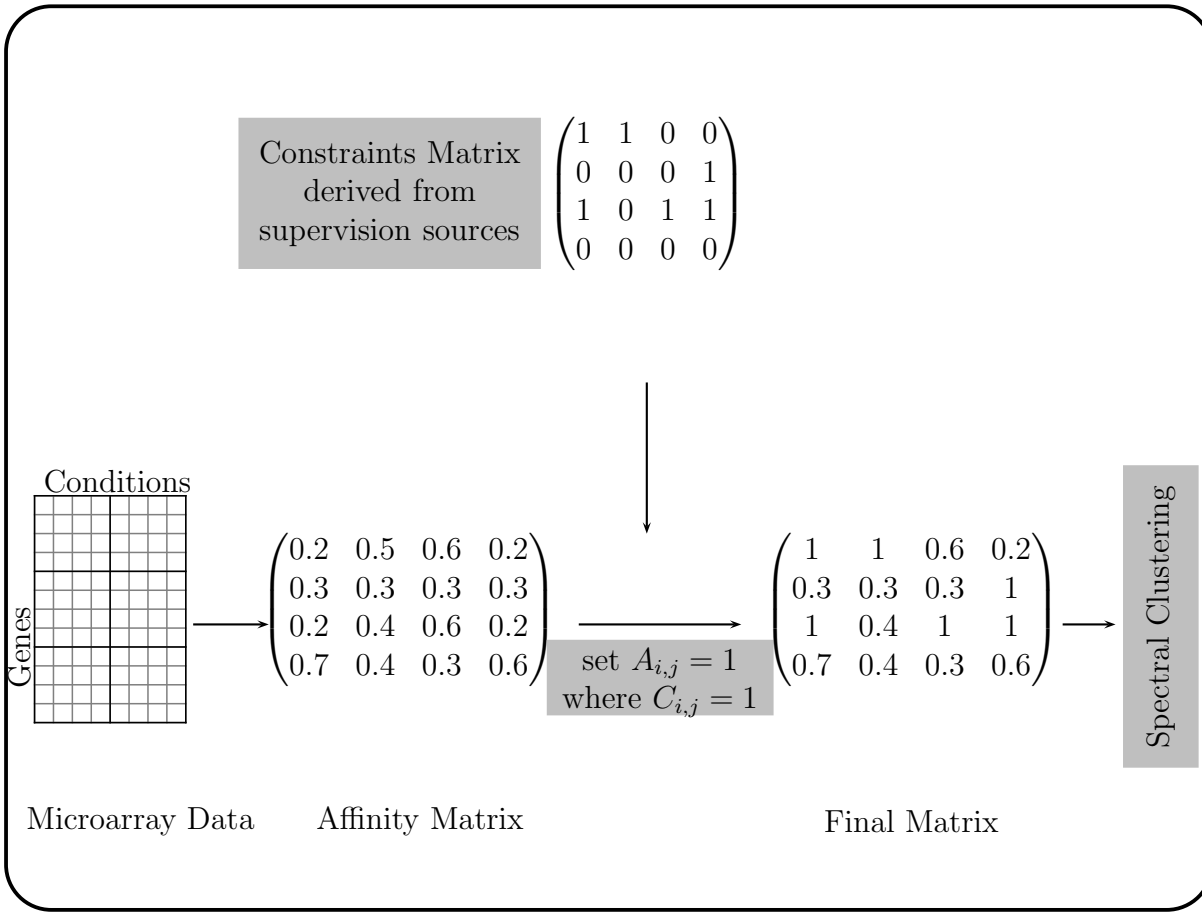



Figure 4.2: Visual representation of semi-supervised spectral clustering

### 4.2.6 Toy dataset explorations

The semi-supervised problem can be stated as - there is a real distribution of data points that have certain pairwise similarities and an ideal clustering can be derived from it. We want to recover a clustering as close to the ideal one based on observing some noisy datasets and some facts (acting as constraints in our setup) that are known to us. Before we start work on real datasets we are going to show that semi-supervised clustering indeed is able to leverage external information in the form of pairwise constraints in order to better the clustering results.

In order to do this, we take a toy dataset with non-linear patterns (points belonging to two classes are present in the form of two spirals) as seen in Figure-4.3(a). This dataset consists of two concentric clusters (300 points belonging to two classes) and was chosen because this is a specially hard problem on which most traditional clustering algorithms fail. It also shows the effectiveness of spectral clustering in finding non-linear clusters which is not possible with traditional clustering algorithms. To represent the known facts or constraints, we extract some random constraints from it. Then to represent a noisy dataset, we skew the $\sigma$ with which pairwise similarities are computed from its optimum value to some random non-optimum value. After this, we study if the addition of progressively increasing quantities of the known relationships (constraints) improves the cluster quality of the noisy dataset. We have used five-fold validation to study the effectiveness of our algorithm. So, in every run of the experiment 80% of the data acts as training data while remaining 20% is test data. The exact steps are detailed below

1. Take the spirals dataset which has two classes visually represented as red and black points as seen in Figure-4.3(a). We compute the optimal $\sigma$ with which to compute pairwise similarities by finding the least cumulative sum, across all the clusters, of sum of squared distance of all points in a cluster from its cluster centre as shown in Equation-5.29. With this optimal $\sigma$ we compute the similarity matrix. We also generate a noisy version of it by changing the $\sigma$ to a random non-optimal value with which pairwise similarities are computed. The results of clustering with optimal and non-optimal $\sigma$ values are shown in Figures-4.3(a) and 4.3(b) respectively.

2. From the original dataset, we generate all possible pairs between all the points

in each of the two classes respectively. Out of all these pairs, according to 5-fold cross validation procedure, take 4 parts as the training set and the remaining one as the testing set.

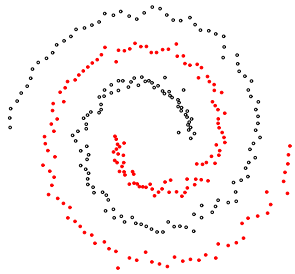
3. Draw *5* pairs of constraints from the training set randomly.
4. Apply these pairwise constraints to the non-optimal similarity matrix, i.e., set the pairwise similarity of these data points to 1. Cluster it using spectral clustering.
5. Compute the similarity between the resulting clusters and the original clustering using modified Rand's index (discussed in Chapter-3).
6. Repeat this process 10 times by randomly drawing constraints from the training set.
7. Increase the number of constraints used in Step-3 by 5 and repeat the whole process till perfect clustering is obtained all the time.
8. Repeat this process 5 times for 5-fold cross validation.

In Figures-4.3(b)-4.3(h) we observe the results of one of the runs where increasing number of constraints are being applied. We see that the quality of clustering improves as the number of known constraints increase. We also observe that with a relatively small number of constraints (50), we are able to retrieve the original clustering.

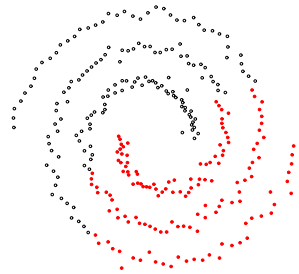
In Figures-4.4(a) to 4.4(e), we see that in all the runs, clustering quality improves quickly as the number of known constraints are applied. The cluster similarity (to the original clustering) improves quickly with reducing standard deviation as the number of constraints increase. The goal of semi-supervised clustering is to use external knowledge in the form of known pairwise relations between variables in order to improve the quality of clustering. We have shown that applying more constraints does indeed lead to better clustering.

### 4.2.7 Parameter optimization

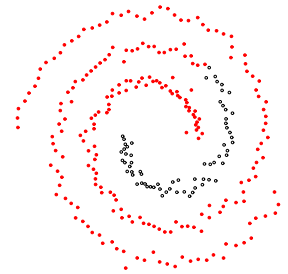
For any clustering algorithm, the most important decisions are the choice of the number of clusters and the free parameters. In our case, the similarity among gene



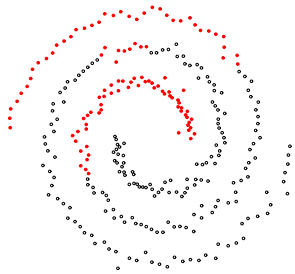
(a) Original dataset clustering



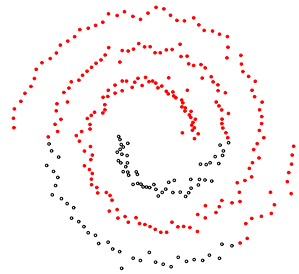
(b) Noisy dataset clustering without any constraints



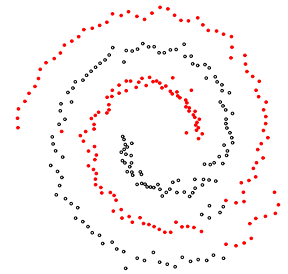
(c) Noisy dataset clustering with 5 constraints



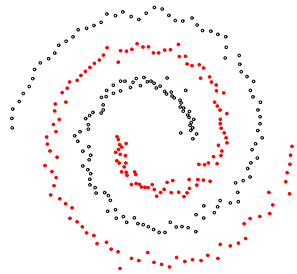
(d) Noisy dataset clustering with 10 constraints



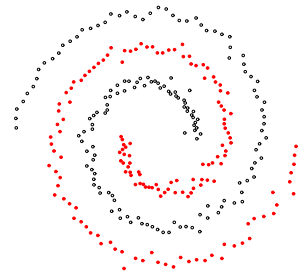
(e) Noisy dataset clustering with 15 constraints



(f) Noisy dataset clustering with 20 constraints

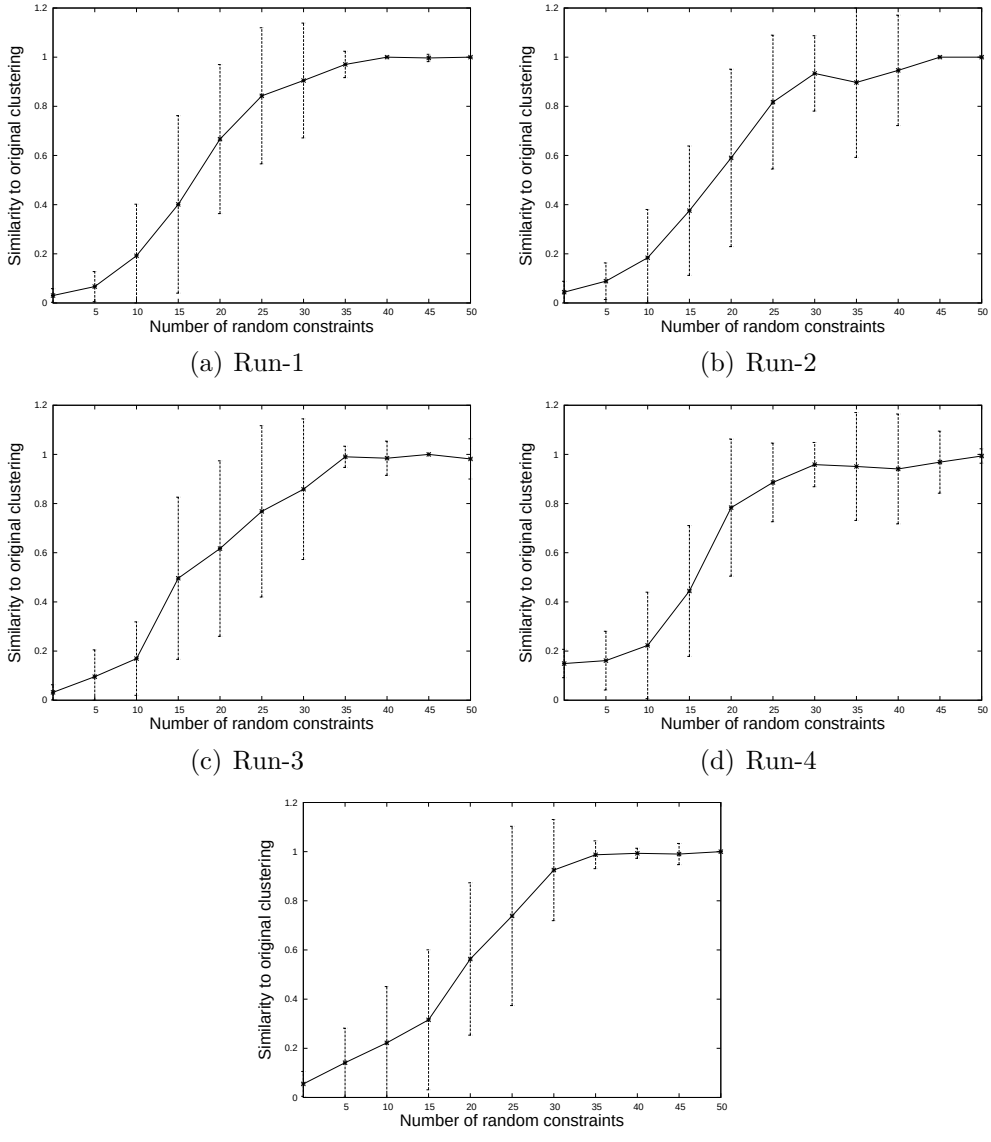


(g) Noisy dataset clustering with 25 constraints



(h) Noisy dataset clustering with 50 constraints

Figure 4.3: Visual indication of Spirals dataset clustering quality improvement with increasing number of constraints. With a relatively small number of constraints (50), we are able to retrieve the original clustering.




(a) Run-1

(b) Run-2

(c) Run-3

(d) Run-4

(e) Run-5

Figure 4.4: Various runs showing clustering quality improvement with increasing number of known constraints. The cluster similarity (to the original cluster) improves quickly with reducing standard deviation as the number of constraints increase.

pairs is calculated using a Gaussian similarity function and the only free parameter is the width of the Gaussian, $\sigma$. For any unsupervised task of an exploratory nature, the *correct* number of clusters is data dependent. We chose to use 50 clusters in our experiments, based on earlier justifications by Ihmels et al. (2002) and Segal et al. (2003) which showed that the Saccharomyces Cerevisiae genome contains approximately 50 sets of functionally related genes. Both the authors have shown statistically that this number provides a better fit to the underlying data distribution, compared to a higher or lower numbers of modules.

In order to determine the value of $\sigma$, we chose to use cluster quality as the parameter to optimise in order to get the optimal value of $\sigma$. There are two major class of algorithms to validate the cluster quality. *External* validation algorithms evaluate a clustering result based on the knowledge of the correct cluster class labels. This is external information that is not contained in the dataset, hence the name. This allows an objective evaluation and comparison of clustering algorithms based on known facts. In cases where no class labels are available, or the available labels are not reliable, we need to use *internal* validation measures. Internal validation techniques do not use external class labels, but utilise information intrinsic to the data itself. They try to measure how well a given clustering corresponds to the natural cluster structure of the data.

### Internal Validation Indices

Internal measures take a clustering and the underlying dataset as the input, and use information intrinsic to the data to assess the quality of the clustering.

*Dunn's* index can be defined as

$$\text{Dunn's index} = \min_{C_i \in C} \left( \min_{C_j \in C \setminus i} \left( \frac{dist(C_i, C_j)}{\max_{C_k \in C} diam(C_k)} \right) \right)$$

where $diam(C_k)$ is the maximum (complete) distance between two points within a cluster and $dist(C_i, C_j)$ is the minimum (single) distance between any two points in clusters $C_i$ and $C_j$. We can observe that the value of this index is high if the inter-cluster separation is high compared to the largest cluster diameter. This corresponds to a fundamental objective of good clustering, namely to maximize the inter-cluster separation and minimize the intra-cluster distances. Hence *better* clustering will

have *higher* values of this index. This index, though very easy to comprehend, can be quite unstable especially in the presence of outliers.

Another popular internal cluster quality validation index - *Davies-Bouldin's* index that aims to identify sets of clusters that are compact and well separated, is defined as

$$\text{Davies Bouldin's index} = \frac{1}{M}\sum_{i=1}^{M}\max_{\substack{j=1\ldots M\\ j\neq i}}\left(\frac{\sigma_{C_i}+\sigma_{C_j}}{\delta(C_i,C_j)}\right)$$

where M is the total number of clusters, $\sigma_{C_i}$ is the average distance of all points in the $i_{th}$ cluster from the cluster centre and $\delta(C_i, C_j)$ is the distance between the cluster centres of the $i_{th}$ cluster and $j_{th}$ cluster. The value of this index decreases if clusters $i$ and $j$ are compact and their centres are far away from each other. Hence *smaller* values of this index indicate better clustering.

The Silhouette index is another well known cluster validity index.

$$s(i) = \frac{b(i) - a(i)}{max(a(i), b(i))}$$

For each datum $i$, let $a(i)$ be the average dissimilarity of $i$ with all other data within the same cluster. We can interpret $a(i)$ as how well matched $i$ is to the cluster it is assigned (the smaller the value, the better the matching). Then find the average dissimilarity of $i$ with the data of another single cluster. Repeat this for every cluster of which $i$ is not a member. Denote the lowest average dissimilarity to $i$ of any such cluster by $b(i)$. The cluster with this average dissimilarity is said to be the *neighbouring cluster* of $i$ as it is, aside from the cluster $i$ is assigned, the cluster in which $i$ fits best.

From the above definition it is clear that

$$-1 < s(i) < 1$$

For $s(i)$ to be close to 1 we require $a(i) \ll b(i)$. As $a(i)$ is a measure of how dissimilar $i$ is to its own cluster, a small value means it is well matched. A large $b(i)$ implies that $i$ is badly matched to its neighbouring cluster. Thus an $s(i)$ close to one means that the datum is appropriately clustered. If $s(i)$ is close to negative one, then

by the same logic we see that $i$ would be more appropriate if it was clustered in its neighbouring cluster. The average $s(i)$ of a cluster is a measure of how tightly grouped all the data in the cluster are. Thus the average $s(i)$ of the entire dataset is a measure of how appropriately the data has been clustered.

We chose to use Dunn's Index and Davies Bouldin's index in our studies to find the optimum value of $\sigma$. The underlying logic of using this to choose $\sigma$ is to search for a value which results in the best quality clusters. We carried out this $\sigma$ optimization without using the supervision step, clustering only the microarray dataset because our objective is to study the impact of supervision. If we incorporate it prior to optimization then the constraints will impact the original similarity matrix.

We ran the spectral algorithm for a range of $\sigma$ values. The range of $\sigma$ values was determined as both the upper and lower extremes beyond which all the points resulted in a trivial clustering (single cluster). For each $\sigma$ value, we did 10 runs as spectral clustering depends on k-means which has random starting points. We also repeated the k-means algorithm twenty five times, each run being initialised randomly, and chose the best clustering with the minimum total *dispersion*. Dispersion was computed by taking the cumulative sum of within-cluster sum of squared distances (from each point to the centre) across all the clusters of a clustering run.

**Stress dataset**

The results for the Dunn's index based sigma optimisation for the Stress dataset is shown in Figure-4.5 which shows the mean values along with standard deviation error bars. The x-axis uses a log-scale because of the spread of the data. For Dunn's index, where higher values are better, the plot has the optimal region between 0.003 and 0.005 where the standard deviations are low. The maximum value (best clustering) within this region is at $\sigma = 0.005$. It is worthwhile to note that the best quality clustering also has a very low standard deviation.

As seen in Figure-4.6, the Davies-Bouldin's index, where lower values indicate better clustering, has its optimal region between 0.003 and 0.03. It has its minimum value (best clustering) at $\sigma = 0.01$. However, the standard deviation is unacceptably high. Considering that the next best value is $\sigma = 0.005$ and it also has low standard deviation as well as it is in agreement with the Dunn's index values we choose this

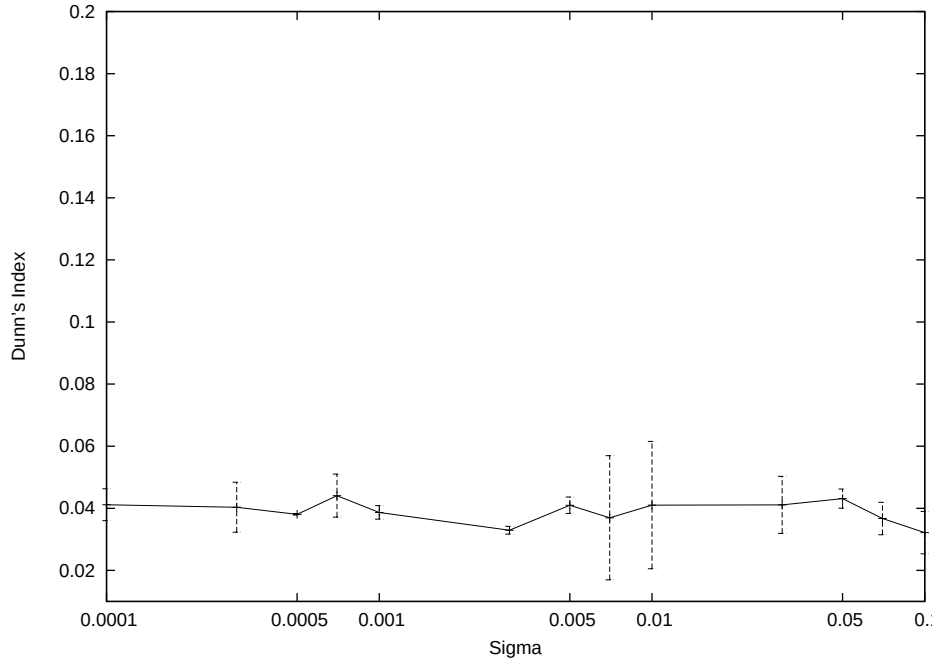



Figure 4.5: Stress dataset: Sigma optimization using Dunn's Index

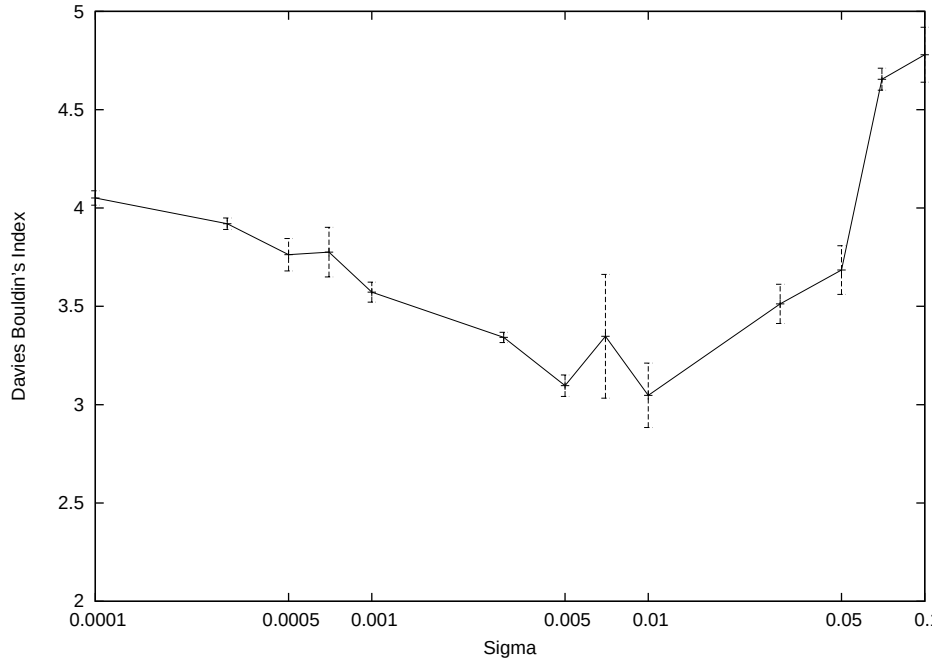



Figure 4.6: Stress dataset: Sigma optimization using Davies Bouldin's Index

as the optimal sigma value for this dataset (Stress) for further computations.

**Cell-cycle dataset**

The result for Dunn's index based optimisation for the Cell-cycle dataset is shown in Figure-4.7 which shows the mean values along with standard deviation error bars. The x-axis uses a log-scale because of the spread of the data. Dunn's index has its maximum value (best clustering) at $\sigma = 0.03$ and has a very low standard deviation there.

As seen in Figure-4.8, the Davies-Bouldin's index has its minimum value (best clustering) at $\sigma = 0.05$ and the optimal region between 0.01 to 0.7. However, the standard deviation there is higher in comparison to at $\sigma = 0.03$. Based on the consensus of both, we have used $\sigma = 0.03$ value for all our further analysis for this dataset (Cell-cycle).

## 4.3 Statistical validation of results

The datasets that we have, represent experiments done under particular conditions. If we base our results just on those datasets, the results that we obtain could be purely by chance because of the characteristics of those particular datasets and it would be dangerous to draw conclusions based from them. Therefore, we have perturbed the datasets and report the mean and variance of results in order to justify that the results are not random. We begin by showing that spectral clustering results are immune to minor perturbations of data and the resulting clusters are not widely different from each other. The exact procedure is outlined as follows:

1. We used the full Stress and Cell-cycle datasets. The Stress dataset had 6361 genes while Cell-cycle dataset had 6353 genes including non-annotated open reading frames (NORFs). Since not much is known about the function of NORFs, we removed all the NORFs from the datasets. That left us with 6251 genes in the Stress dataset and 6257 genes in the Cell-cycle dataset. They had 156 and 60 experiment counts respectively.

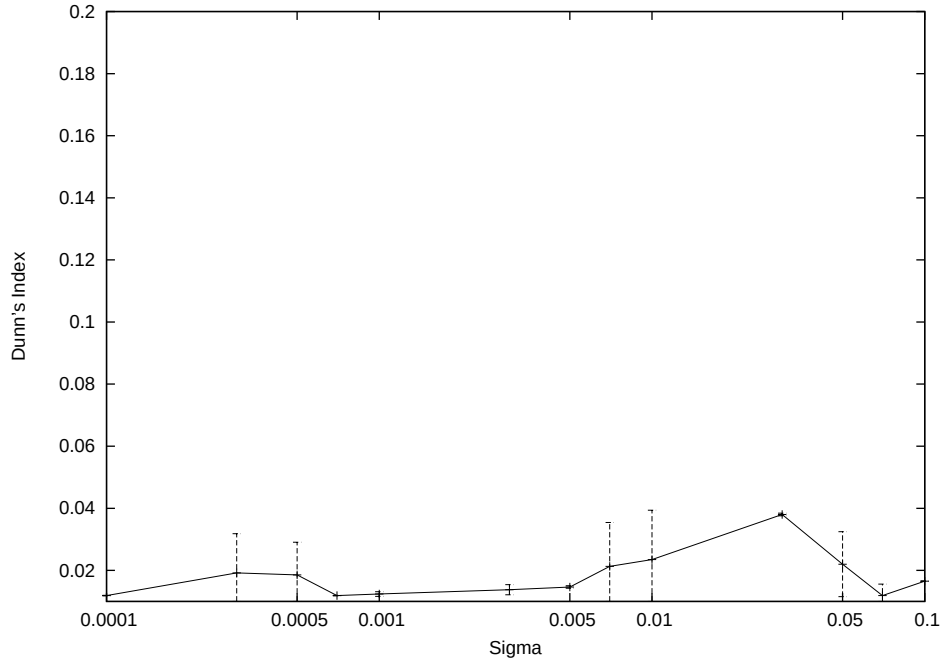



Figure 4.7: Cell-cycle dataset: Sigma optimization using Dunn's Index

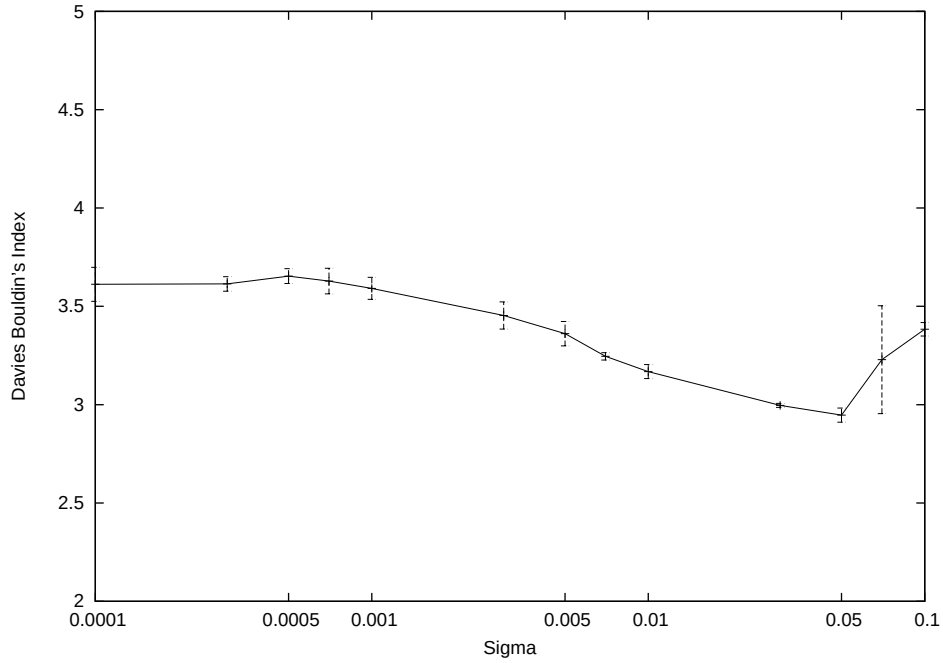



Figure 4.8: Cell-cycle dataset: Sigma optimization using Davies Bouldin's Index

2. We created ten perturbed datasets from both the Stress and Cell-cycle datasets after sub-sampling (drawing 90% of the genes randomly without replacement).

3. In order to compute the similarity matrix from the Stress and Cell-cycle datasets, we need to find the most optimum sigma. In order to do this we compute the Dunn's index and Davies Bouldin's index for each of the datasets. The same sigma is used for all the perturbed datasets because sigma should not change significantly for a dataset because of small perturbation.

4. Once we have the optimum sigma values, we compute the similarity matrices for all the original and perturbed datasets and then cluster them using Spectral clustering. Then we compute the Dunn's and Davies Bouldin's index for the resulting clusters. Finally, we report the mean and standard deviation of the indices.

As we can see in the mean and standard deviation values in Table-4.4, the Spectral clustering algorithm itself is quite stable with perturbations of data and resulting cluster qualities are not widely varying. We can see that the results of the Cell-cycle dataset have more variance in comparison to the Stress dataset. This is seen in the results of both Dunn's and Davies Bouldin's indices.

Next we did a similar analysis of our proposed semi-supervised spectral clustering algorithm to demonstrate that it too does not produce results randomly and is consistent across sub-sampled datasets. In semi-supervised clustering, we are applying constraints in order to improve the quality of clustering. In order to justify that our results are not accidental, we need to sub-sample the microarray datasets as well as the constraints and then apply the sub-sampled constraints. We created ten constraints datasets from each constraints dataset by sub-sampling (drawing 90% of the genes randomly). Then we combined pairs of perturbed micro-array and constraints datasets, clustered them using Spectral clustering and then reported the Dunn's and Davies Bouldin's clustering quality indices. We repeat this for all pairs. To reiterate, this was to demonstrate that our results are not random and that they are statistically valid.

Tables-4.5 to 4.9 show the cluster quality results of individual combinations of datasets. We have compiled the mean and standard deviation values of these individual results in Table-4.10.

| Serial No. | Dunn's index | Davies Bouldin's index |
|---|---|---|
| 1 | 0.04398243 | 3.188676 |
| 2 | 0.04306639 | 3.231177 |
| 3 | 0.04160664 | 3.136993 |
| 4 | 0.03673846 | 3.191647 |
| 5 | 0.04010983 | 3.154793 |
| 6 | 0.04787347 | 3.146264 |
| 7 | 0.02902327 | 3.212804 |
| 8 | 0.02349388 | 3.24161 |
| 9 | 0.03802108 | 3.163264 |
| 10 | 0.0563652 | 3.188100 |
| 11 | 0.03952356 | 3.168467 |

Table 4.2: Dunn's and Davies Bouldin's index values for perturbed Stress dataset. Results indicate that Spectral clustering results in consistent cluster quality.

| Serial No. | Dunn's index | Davies Bouldin's index |
|---|---|---|
| 1 | 0.03368041 | 2.989561 |
| 2 | 0.04296991 | 3.026842 |
| 3 | 0.01052886 | 3.127289 |
| 4 | 0.03681052 | 2.967362 |
| 5 | 0.04211076 | 3.005138 |
| 6 | 0.01081028 | 3.112537 |
| 7 | 0.03603748 | 3.027375 |
| 8 | 0.02896914 | 2.975535 |
| 9 | 0.03686515 | 3.038505 |
| 10 | 0.03952675 | 2.871799 |
| 11 | 0.02926203 | 3.00763 |

Table 4.3: Dunn's and Davies Bouldin's index values for perturbed Cell-Cycle dataset. Results indicate that Spectral clustering results in consistent cluster quality.

| Description | Dunn's index | | Davies Bouldin's index | |
|---|---|---|---|---|
| | Mean | St. Dev | Mean | St. Dev |
| Stress only | 0.0400 | 0.0087 | 3.1840 | 0.0341 |
| Cell cycle only | 0.0316 | 0.0113 | 3.0136 | 0.0693 |

Table 4.4: Summary of mean and standard deviation values of individual microarray datasets. This indicates that our results are not random and a small perturbation in data doesn't change the results.

| Serial No. | Dunn's index | Davies Bouldin's index |
|---|---|---|
| 1 | 0.03089432 | 3.556163 |
| 2 | 0.03321191 | 4.226998 |
| 3 | 0.02888006 | 3.957121 |
| 4 | 0.02731343 | 3.766975 |
| 5 | 0.03100476 | 3.748136 |
| 6 | 0.04172875 | 3.611934 |
| 7 | 0.02727648 | 3.70486 |
| 8 | 0.03053157 | 3.908451 |
| 9 | 0.03172383 | 3.733790 |
| 10 | 0.03420332 | 3.403322 |
| 11 | 0.0312116 | 3.477314 |

Table 4.5: Dunn's and Davies Bouldin's index values after combination of sub-sampled Stress and ChIP-chip datasets

| Serial No. | Dunn's index | Davies Bouldin's index |
|---|---|---|
| 1 | 0.04904669 | 3.563832 |
| 2 | 0.05276492 | 3.552624 |
| 3 | 0.03059342 | 3.65504 |
| 4 | 0.03120301 | 3.729151 |
| 5 | 0.03151477 | 3.712506 |
| 6 | 0.03608802 | 3.581163 |
| 7 | 0.02796911 | 3.691222 |
| 8 | 0.04236918 | 3.713929 |
| 9 | 0.03390614 | 3.513863 |
| 10 | 0.0332141 | 3.770361 |
| 11 | 0.03531316 | 3.625731 |

Table 4.6: Dunn's and Davies Bouldin's index values after combination of sub-sampled Stress and PPI datasets

| Serial No. | Dunn's index | Davies Bouldin's index |
|---|---|---|
| 1 | 0.02777897 | 3.523816 |
| 2 | 0.01287181 | 3.365275 |
| 3 | 0.03601802 | 3.40098 |
| 4 | 0.02776162 | 3.563739 |
| 5 | 0.03599602 | 3.4396 |
| 6 | 0.03671904 | 3.631468 |
| 7 | 0.03603748 | 3.429060 |
| 8 | 0.03518232 | 3.406876 |
| 9 | 0.03876811 | 3.441273 |
| 10 | 0.03532557 | 3.563856 |
| 11 | 0.03534912 | 3.297843 |

Table 4.7: Dunn's and Davies Bouldin's index values after combination of sub-sampled Cell-cycle and ChIP-chip datasets

| Serial No. | Dunn's index | Davies Bouldin's index |
|---|---|---|
| 1 | 0.01359537 | 3.289819 |
| 2 | 0.04214127 | 3.544937 |
| 3 | 0.02760288 | 3.562716 |
| 4 | 0.03359021 | 3.419286 |
| 5 | 0.03575436 | 3.438057 |
| 6 | 0.03155372 | 3.764982 |
| 7 | 0.03372035 | 3.576627 |
| 8 | 0.02835823 | 3.557268 |
| 9 | 0.03526276 | 3.68283 |
| 10 | 0.04028695 | 3.344107 |
| 11 | 0.03255372 | 3.578498 |

Table 4.8: Dunn's and Davies Bouldin's index values after combination of sub-sampled Cell-cycle and PPI datasets

| Serial No. | Dunn's index | Davies Bouldin's index |
|---|---|---|
| 1 | 0.02746167 | 4.34282 |
| 2 | 0.03222197 | 4.390683 |
| 3 | 0.02823931 | 4.232013 |
| 4 | 0.02939493 | 4.362611 |
| 5 | 0.02742154 | 4.269306 |
| 6 | 0.02760314 | 4.416886 |
| 7 | 0.02787649 | 4.089062 |
| 8 | 0.02709607 | 3.842911 |
| 9 | 0.02733636 | 4.4379 |
| 10 | 0.02829065 | 4.429637 |
| 11 | 0.02746646 | 4.165664 |

Table 4.9: Dunn's and Davies Bouldin's index values after combination of sub-sampled Cell-cycle and Yeastract datasets

| Description | Dunn's index | | Davies Bouldin's index | |
|---|---|---|---|---|
| | Mean | St. Dev | Mean | St. Dev |
| Stress-Chip | 0.0316 | 0.0040 | 3.7359 | 0.2339 |
| Stress-PPI | 0.0324 | 0.0071 | 3.6367 | 0.0431 |
| Stress-YT | 0.0367 | 0.0080 | 3.6463 | 0.0843 |
| Cell-cycle-Chip | 0.0325 | 0.0074 | 3.4603 | 0.0992 |
| Cell-cycle-PPI | 0.0322 | 0.0076 | 3.5236 | 0.1406 |
| Cell-cycle-YT | 0.0282 | 0.0015 | 4.2709 | 0.1820 |

Table 4.10: Mean and standard deviation of Dunn's and Davies Bouldin's index values for all combined datasets. This shows that the results of semi-supervised clustering are not random and small perturbations in data do not change the results significantly.

We observe that the results of semi-supervised clustering are stable with perturbations in both the datasets. For the Stress dataset the variability in results is even lower than observed for the stress only dataset according to Dunn's index, but the Davies Bouldin's index displays higher variability. For the Cell-cycle dataset, the variability according to Dunn's index has fallen in all the instances of integration as compared to the Cell-cycle only dataset. However, again Davies Bouldin's index is displaying higher variability.

Now that we are confident of clustering quality as well as the semi-supervised clustering results, we analyse the biological significance of combinations. For this, we need to observe the results of combinations of original datasets and not the perturbed versions.

## 4.4 Biological Significance Analysis

Evaluation of the results of our clustering algorithm requires careful consideration since there are no gold standards against which performance can be measured. The two prominent types of cluster validation measures are *internal* and *external* validation indices. As indicated earlier, internal indices take a dataset and the resulting clustering and use information fully intrinsic to the data itself to assess the quality of clustering while external validation indices use information independent of the dataset for validating the clustering. We already saw the use of an internal validity measure for parameter ($\sigma$) selection. As they are fully dependent on the data itself, internal indices do not give any indication of the biological significance of the resulting clusters.

There are various other methods that have been used in the past for external validation, many of which have used the information available in the Gene Ontology. They calculate the statistical significance of the gene ontology terms in the clusters. While this method gives us general ideas about which clusters might represent what functions, it doesn't allow us to functionally compare different clustering results *numerically*. Some attempts have been made to provide such a numerical index using mutual information and related concepts by Gibbons and Roth (2002) and Gat-Viks et al. (2003) using Gene Ontology annotations.

In order to compare two sets of clusters, e.g. before and after data integration, we have used the technique suggested by Gibbons and Roth (2002). We briefly discuss the ideas of these two papers and then justify our rationale behind our choice.

### 4.4.1 Numerical Biological Significance comparison (using mutual information)

Gibbons and Roth (2002) devised a figure of merit, z-score, based on mutual information between a clustering result and gene annotation data. The z-score indicates relationships between clustering and annotation, relative to a clustering method that randomly assigns genes to clusters. A higher z-score indicates a clustering result that is further from random.

The GO project is a collaborative effort to address the need for consistent descriptions of gene products in different databases. The GO collaborators are developing three structured, controlled vocabularies (ontologies) that describe gene products in terms of their associated biological processes (BP), cellular components (CC) and molecular functions (MF) in a species-independent manner. The project not only writes and maintains the ontologies themselves but more importantly also makes cross-links between the ontologies and the genes and gene products in the collaborating databases. It is organized as three separate tree structured sets (directed acyclic graph (DAG)), consisting of directed edges and vertices, such that each vertex may be descended from several others. Annotation of a gene with a descendant attribute implies that the gene holds all ancestor attributes. They have parsed annotation from SGD of S. cerevisiae genes with GO attributes in such a way that attributes are inherited through the hierarchy, producing a table of ˜6300 genes and ˜2000 attributes in which a 1 in position (i,j) indicates that the gene i is known to possess attribute j, and a 0 indicates our lack of knowledge about whether gene i possesses attribute j. In other words, absence of annotation is not the same as absence of function.

We have not used the graphical structure and inter-relationships among the terms in the ontology graph. We have used only the relationship maintained between the ontology and genes and gene products across the BP category of it as we were interested in ascertaining whether the cluster were enriched in certain biological

processes.

With this gene-attribute table, they construct a contingency table for each cluster-attribute pair, from which they compute the entropies for each cluster-attribute pair ($H_{A_iC}$), for the clustering result independent of attributes ($H_C$), and also for each of the $N_A$ attributes in the table independent of clusters ($H_{A_i}$). Using the definition of mutual information between two variables X and Y, $MI(X,Y) \equiv H(X) + H(Y)H(X,Y)$, and assuming both absolute and conditional independence of attributes, they expand the total mutual information as a sum of mutual information between clusters and each individual attribute. They compute the total mutual information between the cluster result C and all the attributes $A_i$ as:

$$MI(C, A_1A_2, ....A_{N_A}) = \sum_i MI(C, A_i) = N_A H_C + \sum_i H_{A_i} - \sum_i H_{A_i}C$$

where summation is over all attributes i.

They score a partitioning as follows:

- Compute MI for the clustered data ($MI_{real}$), using the attribute database derived from GO/SGD;
- Compute MI again, for a clustering obtained by randomly assigning genes to clusters of uniform size ($MI_{random}$), repeating until a distribution of values is obtained;
- Compute a z-score for $MI_{real}$ and the distribution of $MI_{random}$ values (with mean $m_{random}$ and standard deviation $s_{random}$) according to

$$z = \frac{MI_{real} - m_{random}}{s_{random}}$$

The z-score can then be interpreted as a standardized distance between the MI value obtained by clustering and those MI values obtained by random assignment of genes to clusters. The larger the z-score, the greater the distance, and higher scores indicate clustering results more significantly related to gene function.

Clusters to which genes were randomly assigned were chosen to be as nearly uniform in size as possible, so that some of the success of a clustering algorithm relative to

| Description | Before integration | After integration | % Gain |
|---|---|---|---|
| Stress with ChIP-chip dataset | 89.2 | 86.7 | -2.8 |
| Stress with PPI dataset | 89.2 | 95.2 | 6.72 |
| Stress with Yeastract dataset | 89.2 | 93.2 | 4.48 |
| Cell-cycle with ChIP-chip dataset | 52.2 | 44.8 | -14.17 |
| Cell-cycle with PPI dataset | 52.2 | 55.2 | 5.74 |
| Cell-cycle with Yeastract dataset | 52.2 | 64.8 | 24.13 |

Table 4.11: Comparison of Biological significance index values before and after semi-supervised integration. For both microarray datasets, combination with PPI and Yeastract datasets lead to better significance values while when combined with ChIP-chip dataset, it leads to diminished Biological significance.

random may derive from producing nonuniform cluster size distributions. Uniform cluster sizes yield the highest value of $H_C$, which allows for the highest possible $MI(C, X)$ for some variable $X$ of unknown entropy $H(X)$, because $0 \leq MI(C, X) \leq min(H_C, H(X))$.

Gat-Viks et al. (2003) devised a method that is based on projecting vectors of biological attributes of the clustered elements onto the real line, such that the ratio of between-groups and within-group variance estimators is maximized. The projected data are then scored using a non-parametric analysis of variance test, and the score's condence is evaluated.

Even though both the techniques use non-parametric techniques, we chose the former because Gat-Viks et al. (2003) have indicated that their technique is sensitive to small perturbations in the clustering solution.

**Results**

We started with the full set of Stress and Cell-cycle datasets as discussed earlier. The reason we had chosen Stress and Cell-cycle datasets is because they represent two ends of the spectrum (Tanay et al., 2005). For the ChIP-chip, PPI and Yeastract datasets, we took all the interactions that were between the common set of genes between the microarray and the interactions datasets. The biological significance index values before and after semi-supervised clustering are in Table-4.11.

The different initial biological significance values for Stress and Cell-cycle datasets

(89.2 and 52.2 respectively) are because of the nature of the data. Stress on the organism leads to extremely high levels of activity in the expression levels and many more genes are coordinating together which leads to higher significance values. Cell-cycle on the other hand is a routine activity and the expression patterns of genes do not change so extremely leading to smaller significance values.

We observe that the Stress dataset when combined with ChIP-chip constraints leads to slightly worse biological significance (-2.8%) after the combination. On the other hand, combination with PPI and Yeastract constraints lead to better significance values (6.72% and 4.48% respectively). For the Cell-cycle dataset too, combination with ChIP-chip dataset leads to reduced biological significance (-14.17%). When combined with both PPI and Yeastract constraints the biological significance has gone up. Based on the results we could say that the ChIP-chip dataset is probably the noisiest and does not bring much new information leading to reduced significance for the original datasets. One of the reasons for this is that integration could be meaningful if each of the datasets complement the other. If there are a lot of conflicting information in each of the datasets then the resulting matrix will have a lot of noise and this will be reflected in the final clusters. We know that ChIP-chip and PPI are uncurated datasets i.e. the information there is derived from experiments which always has the possibility of inducing noise. On the other hand Yeastract dataset is a manually curated dataset. That is the reason we see a high improvement when combined with it for both Stress and Cell-cycle datasets.

The goal of integration is to have more biologically relevant clusters from which hypotheses could be derived to be carried out and validated in wet labs. All our result data is available upon request.

### 4.4.2 Qualitative Biological Significance (using Gene Ontology annotations)

Now that we have results of quantitative biological significance before and after applying our algorithm, we would also like to qualitatively analyze the resulting clusters after applying our algorithm. We would like to identify the biological processes represented by the genes in each cluster. For this, we use Genomica's gene set enrichment module available online (`http://genie.weizmann.ac.il/genomica_web/`

`enrichment/gene_sets.jsp`). It allows us to compute p-value of a *HyperGeometric* distribution to find the enrichment of individual genes in a cluster as detailed below.

The result of a clustering algorithm is a set of gene clusters. In order to find out how biologically significant the cluster set is, we have again used Gene Ontology (Consortium, 2001) annotations. So, from this database, we extract annotations for each gene in a cluster. Then, we would like to know if any GO term is overrepresented in the cluster compared to that happening by chance. This can be answered by *p-value* from statistical hypothesis testing. The p-value is the probability of obtaining a result at least as extreme as the one that was actually observed, given that the null hypothesis is true. So, under our null hypothesis that the set of genes is randomly picked from the whole gene population we compute this p-value using a *HyperGeometric* distribution as the probability that $n$ randomly chosen genes will have $k$ or more annotations of a certain type and can be written as

$$P(X \geq k) = \sum_{i=k}^{n} \frac{\binom{K}{i}\binom{N-K}{n-i}}{\binom{N}{n}}$$

Here, the total number of genes is $N$ of which $K$ are known to be of the particular annotation type that we are interested in. The cluster that we test for over-representation has $n$ genes.

In all our computations, we have only used GO terms that were associated with at least three genes in any cluster. Also, we have only used those terms that had p-value less than 0.01. We excluded all clusters that were having less than 3 genes or more than 500 genes in them considering them trivial clusters. In order to correct for multiple hypothesis, we have used the False Discovery Rate with 0.05 threshold.

A sample of terms and their p-values obtained for the combined Stress and ChIP-chip datasets can be seen in Table-4.12 where we see the clusters to which a particular enriched GO term belongs and its corresponding p-value.

Genomica also outputs a visual indication of the GO annotations that are significantly enriched using a colour code which is useful for analysis. The graphical view is a matrix of the two collections - gene sets and GO annotations, where each colored entry indicates that the two collections have a statistically significant overlap, and the intensity of each colored spot represents the fraction of genes in the overlap.

| Cluster Number | Enriched GO Term | P-value |
|---|---|---|
| Cluster 1 | translation | 8.44E-093 |
| Cluster 1 | ribosome biogenesis and assembly | 6.84E-007 |
| Cluster 1 | protein-RNA complex assembly | 1.06E-007 |
| Cluster 1 | ribosome assembly | 1.62E-011 |
| Cluster 1 | ribosome | 1.83E-096 |
| Cluster 1 | small ribosomal subunit | 1.41E-042 |
| Cluster 1 | structural constituent of ribosome | 1.60E-110 |
| Cluster 1 | ribosomal small subunit biogenesis and assembly | 1.16E-008 |
| Cluster 1 | ribosomal small subunit assembly and maintenance | 4.34E-007 |
| Cluster 1 | cytosol | 2.32E-097 |
| Cluster 1 | cytosolic part | 4.11E-120 |
| Cluster 1 | telomere maintenance | 8.30E-006 |
| Cluster 1 | ribosomal large subunit biogenesis and assembly | 1.71E-005 |
| Cluster 1 | eukaryotic 48S initiation complex | 9.11E-051 |
| Cluster 1 | large ribosomal subunit | 3.36E-051 |
| Cluster 1 | cytosolic large ribosomal subunit (sensu Eukaryota) | 1.50E-060 |
| Cluster 1 | regulation of translation | 2.16E-005 |
| Cluster 1 | regulation of translational fidelity | 1.59E-007 |
| Cluster 1 | ribosomal large subunit assembly and maintenance | 7.36E-007 |
| Cluster 1 | ribosomal small subunit export from nucleus | 7.64E-005 |
| Cluster 2 | cell wall | 6.63E-006 |
| Cluster 2 | DNA bending activity | 3.43E-006 |
| Cluster 4 | pyrophosphatase activity | 1.44E-005 |

Table 4.12: A subset of GO term enrichment values for the stress microarray dataset after integration with ChIP-chip data

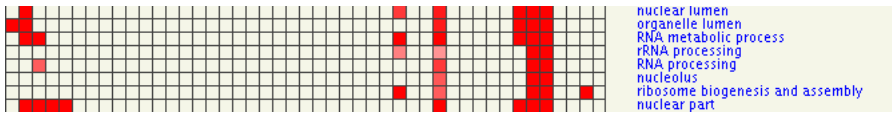
We used Genomica to output images that represent the GO term enrichment before and after the semi-supervised clustering. The images in Figures-4.9 to 4.12 are stress related while Figures-4.13 to 4.16 are cell-cycle related.

**Stress vs Stress and ChIP-chip**

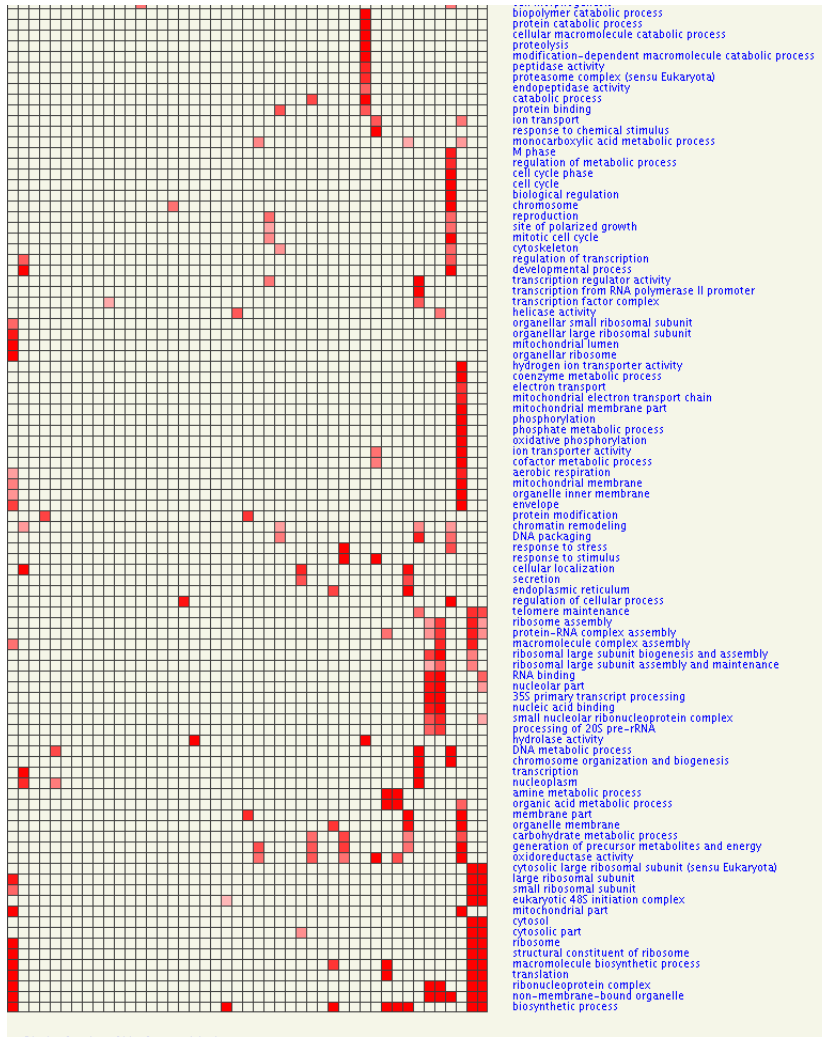
While Figure-4.9 shows the GO term enrishments in the Stress dataset alone, Figure-4.10 shows the enrichments when Stress dataset is combined with the ChIP-chip dataset. Apart from processes like RNA processing, RNA metabolic process, ribosome biogenesis, macromolecule biosynthetic process and translation which are common enrichment observed in the results both before and after the integration, the Stress dataset when combined with the ChIP-chip dataset also shows significant enrichment for processes like biosynthetic process, cellular localization, hydrolase activity, amine metabolic process, amine biosynthetic process, chromosome organization and biogenesis and regulation of cellular process. The genes associated to these biological processes should be studied further to find relationships among them.

**Stress vs Stress and PPI**

Figure-4.11 shows the GO term enrichments when Stress dataset is combined with the PPI dataset. Biological processes like RNA metabolism, RNA binding, rRNA processing, RNA processing, 35S primary transcription, nucleic acid binding, DNA metabolic process are common significant enrichments observed in the results both before and after integration. When combined with the PPI dataset, the Stress dataset also shows significant enrichments for processes like oxidoreductase activity, transcription from RNA polymerase II promoter, transcription, regulation of transcription, regulation of metabolic process, developmental process, cell cycle phase, mitotic cell cycle, biological regulation, regulation of cellular process and chromosome organization and biogenesis as compared to the Stress dataset alone.

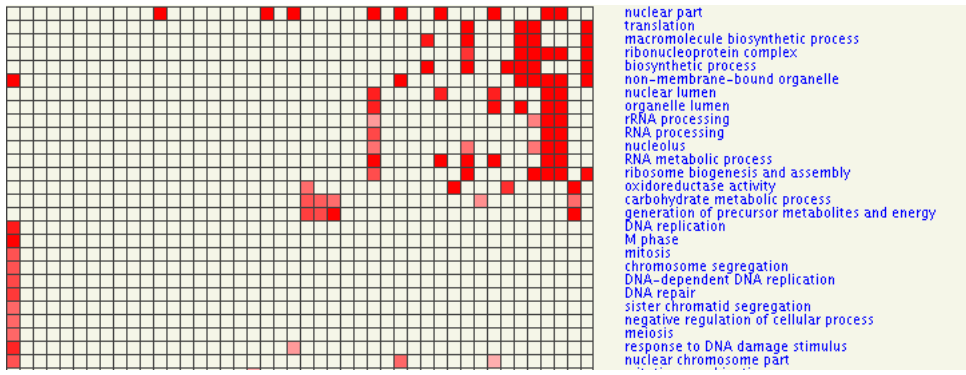



(a) Section of the image showing significant enrichment




(b) Section of the image showing significant enrichment

Figure 4.9: Sections of the image showing significant GO term enrichment in Stress only dataset.

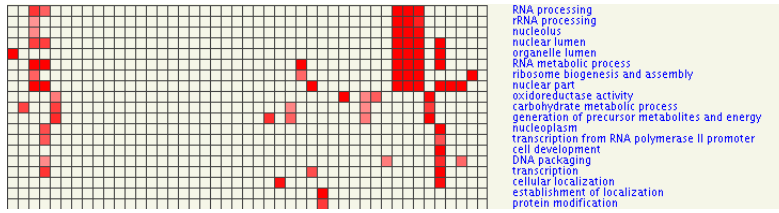



(a) Section of the image showing significant enrichment

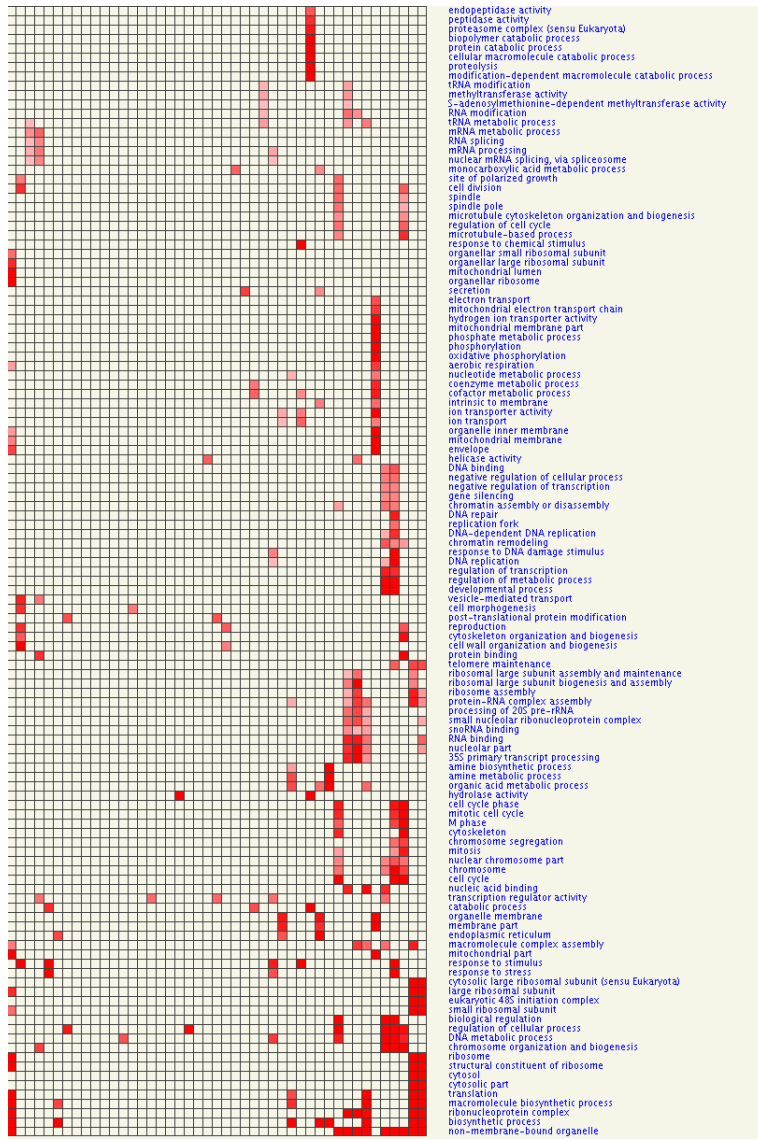



(b) Section of the image showing significant enrichment

Figure 4.10: Sections of the image showing significant GO term enrichment in Stress dataset combined with knowledge (constraints) from ChIP-chip dataset.

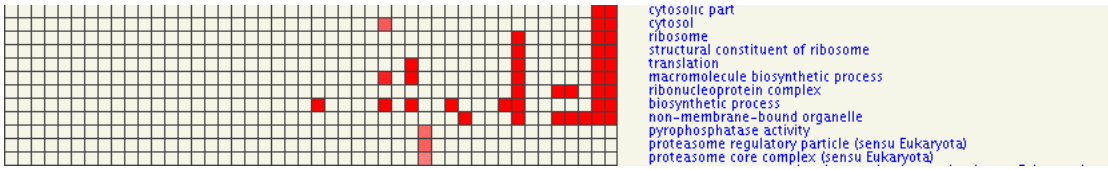



(a) Section of the image showing significant enrichments




(b) Section of the image showing significant enrichments

Figure 4.11: Sections of the image showing significant GO term enrichments in Stress dataset combined with knowledge (constraints) from PPI dataset.




(a) Section of the image showing significant enrichment




(b) Section of the image showing significant enrichment

Figure 4.12: Sections of the image showing significant GO term enrichment in Stress dataset combined with knowledge (constraints) from Yeastract dataset.

### Stress vs Stress and Yeastract

Figure-4.12 shows the GO term enrichments when Stress dataset is combined with the Yeastract dataset. Translation, macromolecule biosynthetic process, biosynthetic process, 35S primary transcript processing, ribosomal biogenesis and assembly, RNA binding, DNA metabolic process, nucleic acid bing, rRNA processing, RNA processing and RNA metabolic process are some of the common enriched processes observed before and after integration. However, processes like response to stress, transcription factor activity, positive regulation of transcription, positive regulation of biological process, RNA polymerase II transcription factor activity, response to chemical stimulus, regulation of transcription from RNA polymerase II promoter, DNA binding, response to stimulus, cellular localization, chromosome organization and biogenesis, mitotic cell cycle, cell cycle, cell cycle phase, telomere maintenance, amine metabolic process, organic acid metabolic process, oxidoreductase activity, regulation of metabolic process, regulation of transcription, transcription, regulation of cellular process, transcription of regulator activity and biological regulation are other significant enrichment observed only in the Stress combined with Yeastract dataset as compared to the Stress dataset alone.

### Cell-Cycle vs Cell-Cycle and ChIP-chip

While Figure-4.13 shows the GO term enrishments in the Cell-cycle dataset alone, Figure-4.14 shows the enrichments when the Cell-cycle dataset is combined with the ChIP-chip dataset. DNA metabolic process, translation, macromolecule biosynthetic process and biosynthetic process are found to be common to the Cell-Cycle dataset before and after combination with the ChIP-chip dataset. After combination, we see significant enrichment for response to stress, response to stimulus and regulation of transcription from polymerase II when compared to Cell-Cycle data set alone.

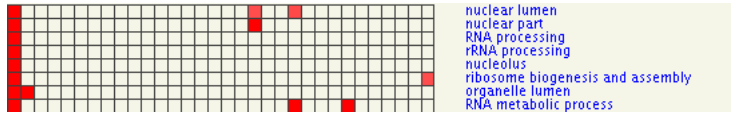
### Cell-Cycle Vs Cell-Cycle and PPI

Figure-4.15 shows the GO term enrichments when Cell-cycle dataset is combined with the PPI dataset. The DNA metabolic process, RNA processing, rRNA process-




(a) Section of the image showing significant enrichment

Figure 4.13: Sections of the image showing significant GO term enrichment in Cell-cycle only dataset.

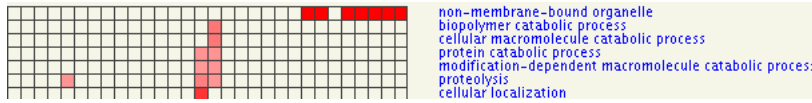



(a) Section of the image showing significant enrichment

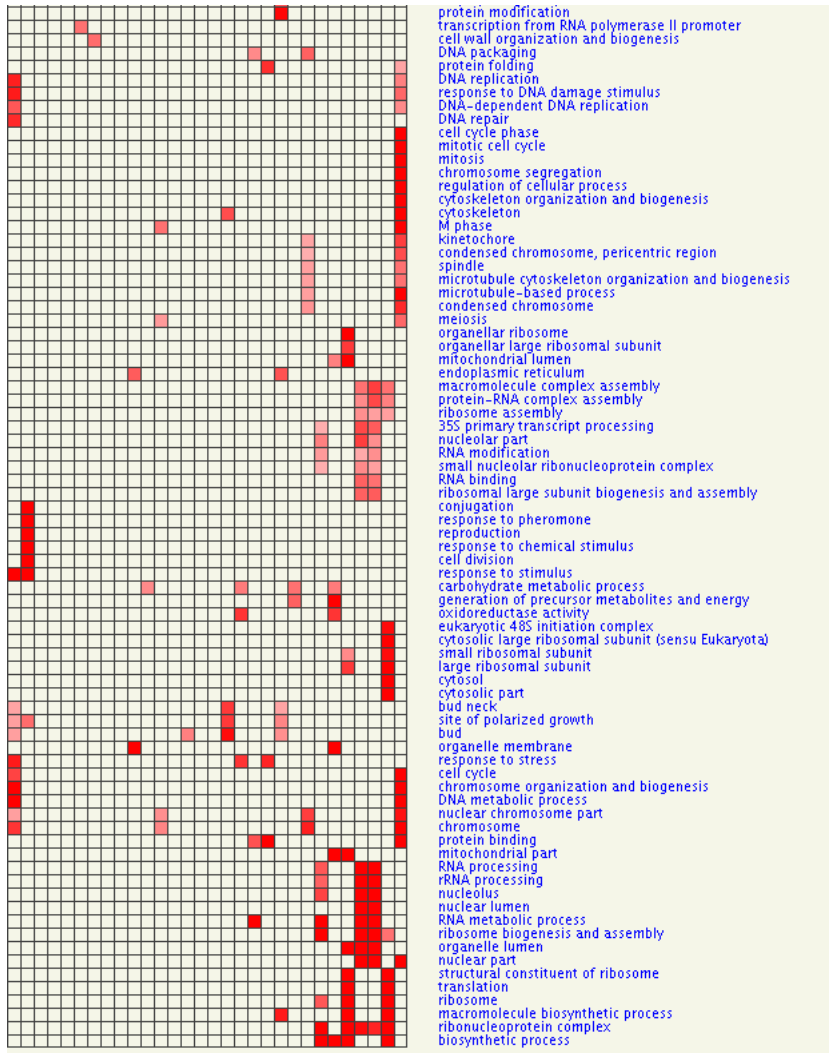



(b) Section of the image showing significant enrichment

Figure 4.14: Sections of the image showing significant GO term enrichment in Cell-cycle dataset combined with knowledge (constraints) from ChIP-chip dataset.




(a) Section of the image showing significant enrichment




(b) Section of the image showing significant enrichment

Figure 4.15: Sections of the image showing significant GO term enrichment in Cell-cycle dataset combined with knowledge (constraints) from PPI dataset.




(a) Section of the image showing significant enrichment

Figure 4.16: Sections of the image showing significant GO term enrichment in Cell-cycle dataset combined with knowledge (constraints) from Yeastract dataset.

ing, the RNA metabolic process, macromolecule biosynthetic process, biosynthetic process and translation are some of the common enriched processes observed in both the data sets. Besides these, Cell-Cycle when combined with the PPI dataset also showed significant enrichment for oxidoreuctase activity, response to stress, cell-cycle, chromosome organization and biogenesis and protein binding when compared to the Cell-cycle dataset alone. We observe that even though cell-cycle processes were not accentuated earlier, after combination with the PPI dataset, they shows up prominently.

#### Cell-Cycle vs Cell-Cycle and Yeastract

Figure-4.16 shows the GO term enrichments when Cell-cycle dataset is combined with the Yeastract dataset. The RNA binding, 35S primary transcription process, RNA processing, rRNA processing, RNA metabolic process, macromolecule biosynthetic process, translation and biosynthetic process are the common significant enrichment sites observed in the Cell-cycle dataset both before and after combination with the Yeastract dataset. The Cell-cycle dataset when combined with the Yeastract dataset also showed enrichment significantly for RNA modification, response to stimulus, mitotic cell cycle and DNA metabolic process when compared to the Cell-cycle dataset alone.

## 4.5 Related Work and Discussion

### 4.5.1 Constrained clustering

The concept of applying prior knowledge in the form of constraints to clustering algorithms is not new. Initial *supervised* clustering algorithms were modifications of traditional ones and ensured that the resulting clusters had to satisfy the applied constraints. One of the first papers in this area by Bradley et al. (2000) proposed a constrained version of the famous *k-means* (Macqueen, 1967) clustering algorithm by posing the problem in terms of minimum cost network flows. Their objective behind adding the constraints was to assign a certain minimum number of points to

each cluster. Tung et al. (2001) have done a systematic study of various constrained clustering algorithms. Basu et al. (2008) is a recent book on constrained clustering.

### 4.5.2 Semi-supervised clustering

While constrained clustering algorithms work towards satisfying known constraints, other distance based clustering algorithms were developed in which the metric that a clustering algorithm uses in order to calculate distance between a pair of data-points was modified by incorporating constraints from other sources of data. These were the first *semi-supervised* clustering algorithms. They did not enforce the constraints, but used the constraints to provide guidance in the cluster formation process. This is the crucial difference between a *supervised* and a *semi-supervised* clustering algorithm. In the former, the constraints are derived from known ground truth and have to be satisfied, whereas in the latter, the constraints are additional sources of information but are considered noisy and hence not necessarily exactly correct. This is a characteristic of the DNA-binding, PPI and TF-gene interactions data that we use for deriving our constraints, and also our justification for using the semi supervised algorithm.

Klein et al. (2002) used the concept of "must-link" and "can-not link" constraints on hierarchical agglomerative clustering (Jain and Dubes, 1988). They reported that it improved upon earlier constrained clustering algorithms and required a much smaller number of constraints for similar accuracy. According to them constraints suggest space-level generalizations beyond their instance-level assertions. In other words, if a point A is linked to another point B, then A should also probably be linked to points that are near B and *vice-versa*. They used this idea to *propagate* constraints. Basu et al. (2004) have proposed a probabilistic model for semi-supervised clustering based on Hidden Markov Random Fields (HMRFs) that provides a more principled framework for incorporating supervision. It combines the constraint based and distance based approached into a single unified model.

Our technique is likewise based on the concept of using the constraints obtained from one dataset in order to modify the similarity value that is obtained from another dataset. The key difference is that while all the previous work has used this principal to do clustering in some feature space, our technique uses the modified similarity values to cluster in spectral space (spectral clustering).

**Spectral clustering**

The field of spectral clustering was started by Donath and Hoffman (1973) who came up with the idea of constructing graph partitions using the eigenvectors of an adjacency matrix. It has generated a lot of interest in recent years (Shi and Malik, 2000; Ng et al., 2001) in clustering related research and has been applied from *object retrieval* (Jain and Zhang, 2007) to *brain surface flattening* (Angenent et al., 1999). A nice review of this subject and its relation to other related topics can be found in von Luxburg (2006) and an upcoming[‖] book by Ding and Zha (2008).

As we have discussed earlier, spectral clustering works on similarity matrices. In that respect, it is similar to *multi-dimensional scaling* or in general, the broader class of *metric multidimensional scaling* (de Leeuw, 2005) algorithms which also operate on a similarity matrix and are useful for visualization of high dimensional data by mapping it to lower dimensions. The key difference between these and spectral clustering is that while they operate in the feature space, spectral clustering works in the spectral space. Spectral clustering has close relations to the field of non-linear dimensionality reduction techniques like *manifold learning* (Saul et al., 2006; Chen et al., 2006) and semi-supervised learning (Grira et al., 2005). Nadler and Galun (2006) have discussed the fundamental limitations of spectral clustering. Dhillon et al. (2005) have shown the equivalence between kernel k-means (Shawe-Taylor and Cristianini, 2004) and spectral clustering. This result gains importance when the similarity matrix is too large for eigen-decomposition and iterative techniques need to be used.

Even before the recent interest in spectral clustering, a matrix formalism has been used to describe the functional states of transcriptional regulatory systems. Gianchandani et al. (2006) used such a model to characterise the properties of transcriptional regulatory systems (TRSs) and facilitate the computation of the transcriptional state of the genome under any given environmental conditions.

One of the first applications of spectral clustering to bioinformatics was by Kluger et al. (2003) who used it to simultaneously clusters genes and conditions (biclustering) for various cancer datasets. In a cancer context, the clusters correspond to genes that are markedly up or down regulated in patients with particular types of

[‖]to be published in Feb 2010

tumors. They present a number of variants of the approach, depending on whether the normalization over genes and conditions is done independently or in a coupled fashion. They analysed publicly available cancer expression data sets, and examined the degree to which the approach is able to identify clusters. They have also compared the performance against a number of reasonable benchmarks (e.g., direct application of SVD or normalized cuts to raw data). Speer et al. (2005) used spectral clustering to cluster Gene Ontology terms to find sets of genes that might be functionally related. They used an information theoretic measure borrowed from text mining, where it had been used to calculate semantic similarities between words, to calculate the similarity values between the terms of the Gene Ontology.

**Semi-supervised spectral clustering**

Kamvar et al. (2003) propose an algorithm for classification called *spectral classification* which modifies the similarity matrix to 1 if the known training data belong to the same class and 0 otherwise. The similarity matrix is subject to eigendecomposition and then classification is done in the spectral space. The advantage of doing this is that they can use labelled data (provides class constraints) as well as unlabelled data (similarity computation). They report better performance than the *naive Bayes* classifier in classifying newsgroups. They have also proposed a constrained spectral clustering with must-link and cannot-link constraints along with an additive normalized laplacian (Fiedler, 1975) and used it for classification. The results and the comparison with other algorithms for this is not systematic and clearly presented which makes it difficult to judge its performance.

Kulis et al. (2005) proposed a semi-supervised version of the kernel k-means algorithm. The difference between our formulation and theirs is that after modifying the similarity matrix we use spectral clustering while they have used kernel k-means algorithm. They argue that this might be better for larger datasets where eigenvalue computation might be computationally expensive whereas kernel k-means being a iterative algorithm, doesn't face this problem. This is true but the drawback with kernel k-means is that like k-means it can get stuck in local optima while spectral techniques always try to approximate the global optimum.

Most algorithms discussed till now rely critically on a good metric over their inputs. If a clustering algorithm fails to find clusters that are meaningful, then the

recourse usually is to manually tweak the metric until sufficiently good clusters are found. Xing et al. (2003) proposed an algorithm that, given examples of similar (and, if desired, dissimilar) pairs of points, *learns* a distance metric that respects these relationships. They also demonstrate that the learned metrics can be used to significantly improve clustering performance.

### 4.5.3 Co-clustering

This is a related and overlapping technique where two or more sources of data are combined. The key difference from semi-supervised or constrained clustering is that in co-clustering, one dataset is not used to guide the other. Rather, both the datasets are combined with equal or varying weights by combining their distance metrics to come up with a new one which is then used for clustering. We will see a detailed discussion and review in the next chapter.

## 4.6 Conclusion

We have proposed a technique to integrate diverse datasets where one is acting as a source of supervision on the clustering of the other. As part of this, we have investigated whether constraints are useful at all in order to retrieve the original clustering from a noisy version of the original dataset. We also investigated two methods for determining the best Gaussian kernel to obtain the affinity matrix from the data. Further, we have used a validation method which scores the resulting gene clusters by reference to a third type of data (Gene Ontology). Our results indicate that semi-supervised spectral clustering leads to improved biological significance if the datasets from which known facts are extracted is not widely varying from the datasets on which they are applied.

Since our technique is quite generic, in future, our work can be extended by using other sources of prior knowledge, for example the similarity between the promoter sequences of genes. In the next chapter, we propose a technique where instead of creating definite constraints, we extract similarities from graphs of interactions and then integrate the datasets. It is based on the principle of maximizing the entropy of the resulting matrix.

One of the shortcomings of this research is that it is known that gene regulation is a very condition specific activity and hence the expression values that we observe are a result of regulation happening at one particular time. However, the datasets that we have used are from different conditions. The microarray datasets as well as the DNA-binding, PPI and Yeastract datasets were not experimentally observed at the same time or even by the same researchers. This is also a fundamental limitation of the underlying experimental techniques, since microarrays themselves do not represent a single time point, but rather the integration of gene activity over a time period. Moreover, knowledge about gene modules is not complete and this hinders the validation process.

# Chapter 5

# Maximum Entropy Kernel Integration for Regulatory Module Discovery

> "But since the affairs of men rest still incertain, Let's reason with the worst that may befall." - *William Shakespeare (Julius Caesar)*

## 5.1 Introduction

As we saw in the previous chapters, each of the current datasets, e.g. microarrays, DNA-binding, protein-protein interaction and sequence datasets, provide a partial and noisy picture of cell regulation. Hence, integration among these is required in order to obtain an improved picture of the underlying process. Initial methods of data integration in regulatory module discovery were mostly ad-hoc approaches that used clustering with some form of prior knowledge. Later on these were enhanced to incorporate model based clustering methods as well. One of the major drawbacks of these techniques was that they worked well on vectorial data but as soon as other types of data were encountered, the principled nature of the algorithms broke down and they had to resort to ad-hoc statistical techniques for finding correlations in datasets.

In the previous chapter we proposed a similarity based method which is very pertinent for non-vectorial data as there are established techniques to compute similarity from these. We will continue using similarity based techniques in this chapter. The bigger challenge that we saw in the last chapter was that of *ad hoc* combination of datasets. Since they are reported as p-values, the DNA-binding data could be interpreted as similarity values. The similarity values in the DNA-binding dataset were converted into constraints and then combined to the microarray data using *ad hoc* p-value thresholds. In order to do the integration in a *principled* manner, we needed a framework under which various types of data could be integrated and their effects analyzed. Various earlier researchers have used the Bayesian framework for merging data, but in our opinion it is unsuitable to cope with non-vectorial data (strings, graphs) in a principled manner as it was primarily developed for vectorial data.

To summarise, the problem now is reduced to having two similarity matrices and we need some method to integrate them. A simple approach to integrating matrices is the *shrinkage* method. When there are two similarity matrices $K_1 and K_2$, a final combination $K$ could be written as,

$$K = \mu K_1 + (1 - \mu) K_2 \tag{5.1}$$

which represents a *convex combination*[*] of $K_1$ and $K_2$ with the shrinkage parameter $\mu$ ranging between 0 and 1, and controlling what fraction of each similarity matrix contributes towards the final matrix. The *shrinkage* method is named so because depending on the $\mu$, we shrink the contribution of the original evidence. For example, if we are combining microarray data with PPI data to improve the predictions based just on microarray data, then the contribution of microarray data is being shrunk from its original contribution (which is 1). Optimum $\mu$ values can be chosen after running these various weight combinations of datasets through the Spectral clustering algorithm and then optimizing for the best cluster quality using the Dunn's or Davies-Bouldin's index values. While this is reasonable from a practical viewpoint, it's not very principled. That motivates us towards our next step which is to use the principle of maximum entropy (Section-5.4) in order to merge the similarity matrices. As we will see in following sections, this allows us to merge two

[*]A convex combination is a linear combination of where all coefficients are non-negative and sum up to 1

datasets when there is no evidence available regarding their individual importance. For example, when we have two noisy data sources e.g. microarray and PPI, and no other evidence regarding their individual importance, we can combine them to get better inference using the principle of maximum entropy.

As discussed in Section-5.4, we need a more specialised version of the similarity matrix for maximum entropy integration. The extra requirement is that our similarity matrices should be positive semi-definite. There is a separate but similar branch of machine learning that operates only on such matrices and is known as *kernel methods*. We describe these in Section-5.2. Throughout this chapter, we have used similarity matrices, kernels and kernel matrices interchangeably to refer to *positive semi-definite symmetric similarity matrices*.

### 5.1.1 Spectral or eigen-decomposition

Spectral or eigen decomposition of a symmetric $n \times n$ matrix $\mathbf{A}$ is represented as

$$\mathbf{A} = \mathbf{Q}\Lambda\mathbf{Q}^{-1}$$

where $\Lambda = diag(\lambda_i)$ is the diagonal matrix, $\lambda_i$'s are the eigenvalues of $\mathbf{A}$ and $\mathbf{Q} = [q_1, \ldots, q_n]$ is the square $n \times n$ matrix whose $i_{th}$ column is the basis eigenvector $q_i$ of $\mathbf{A}$.

A symmetric matrix $\mathbf{A}$ is said to be *positive definite* if

$$\mathbf{x}^T\mathbf{A}\mathbf{x} > 0 \text{ for all non-zero x}$$

A symmetric matrix $\mathbf{A}$ is said to be *positive semi-definite* if

$$\mathbf{x}^T\mathbf{A}\mathbf{x} \geq 0 \text{ for all non-zero x}$$

All the eigenvalues of a positive definite matrix are positive whereas the eigenvalues of a positive semi-definite matrix are non-negative.

## 5.2 Kernel Methods

*Kernel methods* are algorithms that operate on a type of data representation known as a *kernel* matrix. Kernel matrices provide a general framework to represent data and satisfy certain mathematical properties. A kernel matrix is defined not in terms of individual variables but in terms of pairwise similarity among all variables. So, instead of using a mapping $\phi : \mathcal{X} \rightarrow \mathcal{F}$ to represent each object $\mathbf{x} \in \mathcal{X}$ by $\phi(\mathbf{x}) \in \mathcal{F}$, a real valued similarity function $k : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ is used and the dataset with n variables is represented by a n $\times$ n matrix of pairwise similarities $k_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$. The most significant fact regarding these methods is that once we have a kernel matrix representation of the data then the original data is not required and the methods can work on just these matrices. This is where the real beauty of these methods arise as different types of data types do not necessitate changes in the underlying algorithm. Kernel methods require that a kernel matrix is *symmetric* and *positive semi-definite*. This means that if $k$ is an $n \times n$ matrix of pairwise similarities then $k_{i,j} = k_{j,i}$ for $1 \leq i, j \leq n$, and $\mathbf{c}^\top k\mathbf{c} \geq 0$ for any $c \in \mathbb{R}^n$. This also implies that the matrix has non-negative real eigenvalues.

Each similarity value ($k_{i,j}$) in a kernel matrix is calculated using a so called kernel function ( k(x,y) ) that acts a *suitable* similarity between the variables. Hence, a real valued kernel matrix could be obtained for diverse data types (strings and graphs) as long as a similarity function can be defined over a pair. This nice property leads to complete separation of similarity function definition from the algorithms that operate on these matrices. This is specially useful in bioinformatics because of diverse types of datasets (as pointed in previous chapter) where a real valued representation of individual variables is non intuitive while a similarity score makes sense, e.g. genomic sequences. We will see different types of kernels in Section-5.2.1.

### 5.2.1 Various kernel or similarity functions

We provide a short description of various possible kernels for different data types (vectors, strings and graphs) and their properties.

### Vector Data

- The *Linear* or *Dot kernel* is the simplest one.

$$k_L(\mathbf{x}, \mathbf{x}^{'}) = \mathbf{x}^T\mathbf{x}^{'} \tag{5.2}$$

- The *Polynomial kernel* is a more general case of the linear kernel

$$k_{Poly}(\mathbf{x}, \mathbf{x}^{'}) = (\mathbf{x}^T\mathbf{x}^{'} + c)^d \tag{5.3}$$

  where d is the degree of the polynomial and c is a constant. When c is non-zero then this kernel corresponds to a feature space spanned by all products of at most 2 variables i.e., $\{1, x_1, x_2, x_1^2, x_1x_2, x_2^2\}$. When c is zero then this space is restricted to only the products of exactly 2 variables i.e., $\{x_1^2, x_1x_2, x_2^2\}$.

- The most popular and widely used kernel function used for real data is the *Gaussian* or *Radial Basis Function (RBF) kernel*

$$k_G(\mathbf{x}, \mathbf{x}^{'}) = exp\left(-\frac{\parallel \mathbf{x} - \mathbf{x}^{'} \parallel^2}{2\sigma^2}\right) \tag{5.4}$$

  the width of the Gaussian being controlled using $\sigma$. This affinity function naturally encodes the local neighbourhood property and its value falls rapidly as the pairwise dissimilarity increases.

- Another popularly used kernel is the *Sigmoid kernel*

$$k_S(\mathbf{x}, \mathbf{x}^{'}) = (k\mathbf{x}^T\mathbf{x}^{'} + \theta) \tag{5.5}$$

  where $k > 0$ and $\theta < 0$ are the *gain* and *threshold*.

### Graph data

A graph is informally defined as a set of *nodes* connected by *edges*. In bioinformatics, typical examples of a graph would be the interactions between the proteins of an organism or the interaction network representing the metabolic pathway. Other common examples of such graphs are social networks and hyperlinked internet web

pages. While a graph represents *local similarity* i.e., a node's direct interactions in its neighbourhood, we need a similarity function that represents *global similarity* i.e., a node's interaction to every other node in the graph. The simplest measure of similarity on a graph is the shortest-path distance, but it is not positive semi-definite which is our requirement. Apart from this, this is very sensitive to insertions and deletions of edges. A more robust similarity measure is required which could perhaps average over many paths. The physical process of diffusion suggests a natural way of propagating such local information and has led to the most popular type of similarity on graphs known as the *diffusion* kernel (Kondor and Lafferty, 2002).

Laplacian $L$ of an *undirected unweighted* graph is defined as,

$$L_{i,j} = \begin{cases} -1 & \text{for i} \sim \text{j}, \\ d_i & \text{for i=j}, \\ 0 & \text{otherwise} \end{cases} \tag{5.6}$$

where $i \sim j$ implies that $i$ and $j$ are connected by an edge and $d_i$ is the number of edges originating from $i_{th}$ node. The kernel function on the graph can be defined using the negative of this Laplacian ($H = -L$) as

$$K_\beta = e^{\beta H} = \lim_{m->\infty} \big(\mathbf{I} + \frac{\beta H}{m}\big)^m \tag{5.7}$$

where $\beta$ is a positive constant and $\mathbf{I}$ is an identity matrix. $K_\beta$ represents an exponential family of similarity functions with generator $H$ and bandwidth parameter $\beta$. Using power series expansion this can be expanded to

$$K_\beta = \mathbf{I} + \beta H + \frac{\beta^2 H^2}{2} + \frac{\beta^3 H^3}{3!} + \dots \tag{5.8}$$

Note that $e^{\beta H}$ yields a matrix but it is not the same as component-wise exponentiation $e^{\beta H_{ij}}$. If a matrix is diagonal then its exponential can be obtained by just exponentiating every entry on the diagonal, i.e., $e^D = diag(e^{d_{11}}, e^{d_{22}}, \dots, e^{d_{nn}})$. This is an important property that could be used for computing the exponential. If we diagonalise $H$ i.e., if $H = UDU^{-1}$ and D is diagonal, then $e^H = Ue^DU^{-1}$. Based on this, we have used the technique discussed in Moler and Loan (2003) to compute our matrix exponentials. It involves computing the normalized eigenvalues and

eigenvectors of $H$.

$$H = \sum_{i=1}^{n} v_i \lambda_i v_i^T \tag{5.9}$$

which when replaced in Equation-5.8

$$K_\beta = \sum_{i=1}^{n} v_i e^{\beta \lambda_i} v_i^T \tag{5.10}$$

This similarity function is also known as the *diffusion* function because its differential equation form resembles the diffusion equation of heat through continuous media in classical physics (Kondor and Lafferty, 2002). The function of $\beta$ is to control the extent of diffusion similar to the $\sigma$ of the Gaussian kernel. In fact, as shown in Schlkopf et al. (2004), there is a straightforward correspondence between the diffusion kernel and the Gaussian kernel. The former can be considered a discretized version of the latter. In the next section we discuss our actual technique of similarity matrix integration.

### 5.2.2 From similarities to a valid kernel

Sometimes we have a well defined measure of similarity between a pair of objects, but the resulting matrix is not a valid kernel matrix according to the strict definition of positive semi-definiteness. In such cases, two methods have been proposed in the literature that may be used to convert the similarity matrix to a valid kernel. Tsuda (1999) has proposed a principled technique called *empirical kernel map*. Roth et al. (2002) have proposed an ad-hoc technique of eigen-decomposition of the similarity matrix and then removal of negative eigenvalues. They have also showed that this preserves the cluster structure of the data. When we are not sure if the similarity matrix that we have obtained is a kernel matrix then one of these techniques could be used to make it a kernel matrix.

### 5.2.3 Kernel normalization

In order to add kernels, we need to normalize them so that they are on the same scale. Given an unnormalized kernel matrix, K, the normalized version is

$$\hat{K_{ij}} = \frac{K_{ij}}{\sqrt{K_{ii} \times K_{jj}}} \tag{5.11}$$

This can be easily computed if we define $A = (1/\sqrt{K_{11}}, \ldots, 1/\sqrt{K_{nn}})$. Then, $\hat{K} = K * (AA^T)$, where $*$ denotes element-wise product.

## 5.3 Principle of Maximum Entropy

### 5.3.1 Entropy

While the term *entropy* is popularly associated with thermodynamics, the entropy which we describe here comes from *information theory*. This branch of applied mathematics and electrical engineering which deals with quantification of information was introduced by Claude E. Shannon in his seminal paper (Shannon, 1948). While this original paper dealt with the engineering problem of the transmission of information over a noisy channel, the scope of information theory has widened a lot and touches subjects as diverse as cryptography to neurobiology. The most fundamental result of this theory is the *source coding theorem*, according to which, on average, the number of bits needed to represent the result of an uncertain event is given by its *entropy*. In other words, *entropy* is a measure of the uncertainty associated with a random variable.

If a discrete random variable $X$ takes values $x_1, \ldots, x_n$ then its entropy $H$ is

$$H(X) = E(I(X))$$

where E is the expected value and I(X) is the *information* content or *self-information* of X. Now, if $p(x_i)$ is the probability of $X$ taking value $x_i$ then the entropy can

explicitly be written as

$$H(X) = \sum_{i=1}^{n} p(x_i)I(x_i) = -\sum_{i=1}^{n} p(x_i)\log_b p(x_i),$$

where b denotes the base of the logarithm. The unit of entropy is the *bit* or *nat* for bases 2 and e respectively. If any of the probabilities vanish ($p(x_i) = 0$ for any $i$, we use the fact that $\lim_{p\to 0} p\log p = 0$ and hence the value for that particular $i$ is zero.

*Differential entropy* also known as continuous entropy tries to extend the idea of Shannon entropy which is restricted to random variables taking discrete values to continuous probability distributions, e.g. Gaussian distribution. Another widely used measure of entropy for the continuous case is the *relative entropy* of a distribution also popularly known as the KL divergence (refer Section-3.2.1). We will later maximize the differential entropy associated with a Gaussian distribution in order to merge similarity matrices (refer Section-5.4).

### 5.3.2 Principle of maximum entropy

Before we discuss our technique in detail, here we discuss the background and philosophical underpinnings of principle of maximum entropy. While in the earlier section we made a strict distinction between entropy associated to thermodynamics and information theory, at a more philosophical level, connections can be made between these two seemingly unrelated subjects. According to E.T. Jaynes in his seminal papers (Jaynes, 1957, 1982)

> Thermodynamics should be seen as an application of information theory and the thermodynamic entropy is interpreted as being an estimate of the amount of further Shannon information needed to define the detailed microscopic state of the system, that remains uncommunicated by a description solely in terms of the macroscopic variables of classical thermodynamics.

He proposed correspondence between statistical mechanics and information theory and suggested that the entropy in statistical mechanics, and the information entropy

in information theory, are essentially the same thing. Consequently, statistical mechanics should be seen just as a particular application of a general tool of logical inference and information theory.

Suppose some testable information about a probability distribution is known. If we consider the set of all probability distributions which encode this information then the *principle of maximum entropy (MaxEnt)* states that the probability distribution which maximizes the information entropy in view of the testable information is the true probability distribution. By choosing to use the distribution with the maximum entropy allowed by our information, we are choosing the most *uninformative* distribution possible. If we choose any distribution with lower entropy then that would imply that we are assuming information which we do not have. On the other hand, if we choose a distribution with a higher entropy that would violate the constraints of the information we possess. Thus the maximum entropy distribution is the only reasonable distribution. The Maximum entropy principle is used to choose the smoothest distributions out of all possible distributions.

In our context, intuitively, each similarity matrix represents a distribution and we need to merge them so that the final distribution doesn't make assumptions about the individual weights of the matrices because that information is unavailable. We allow maximum entropy for the resulting distribution implying no assumptions whatsoever. This is the only approach available to us for kernel integration in the unsupervised domain.

Information theory as shown in Section-5.3.1 defines *information* in terms of probability distributions thus providing us with a quantitative measure of uncertainty (entropy) or ignorance. This can be maximized to find the maximally unbiased probability distribution.

## 5.4 Maximum Entropy Kernel Integration

We assume the *similarity matrix* which is a symmetric positive semi-definite matrix to be the *covariance matrix* of a *Gaussian* distribution. Based on the earlier justification for the maximum entropy principle, we need to combine two similarity matrices such that the resulting one has maximum entropy.

Now, a Gaussian distribution is represented as,

$$p(\mathbf{x}|\mu, \mathbf{\Sigma}) = \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}} \exp\big\{ -\frac{1}{2}(\mathbf{x}-\mu)^T\Sigma^{-1}(\mathbf{x}-\mu)\big\} \tag{5.12}$$

where $|\Sigma|$ is the determinant of the covariance matrix $\Sigma$, and $\mu$ is the mean of distribution. Its differential entropy is given by (Brookes, 2005),

$$\begin{aligned} H(p(\mathbf{x})) &= -\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty}\dots\int_{-\infty}^{+\infty} p(\mathbf{x})ln(p(\mathbf{x}))d\mathbf{x} &(5.13)\\ &= \frac{1}{2}(n + nln(2\pi) + ln|\Sigma|) &(5.14)\end{aligned}$$

In order to maximize $H(p(\mathbf{x}))$ we can ignore the first two terms ($n$ and $nln(2\pi)$) in equation-5.14 as they are constants. So, it becomes a problem of maximizing the $ln|\Sigma|$ term. We know that the determinant of a symmetric matrix is equal to the the product of its eigenvalues, i.e.,

$$|\Sigma| = \prod_{i=1}^{k}\lambda_i$$

where $\lambda_i$ $(i = 1 \dots k)$ are the $k$ eigenvalues of $\Sigma$. Therefore,

$$\begin{aligned} ln|\Sigma| &= ln\big(\prod_{i=1}^{k}\lambda_i\big) &(5.15)\\ &= \sum_{i=1}^{k} ln(\lambda_i) &(5.16)\end{aligned}$$

Also, since logarithmic functions are monotonically increasing, so we can restate that *in order to maximize the entropy of a Gaussian distribution, we need to maximize the* $ln|\Sigma|$[†] *which is equivalent to maximizing the sum of the eigenvalues of its covariance matrix.* Now assume that our covariance matrix is a combination of two covariance matrices and so can be rewritten as

$$\begin{aligned} K &= \sum_{i=1}^{2}\mu_i K_i &(5.17)\\ &= \mu_1 K_1 + \mu_2 K_2 &(5.18)\end{aligned}$$

[†]this is also popularly known as log det maximization in optimization theory literature (Boyd and Vandenberghe, 2004)

where $\mu_1+\mu_2 = 1$. Now, according to spectral decomposition of a symmetric matrix,

$$\begin{aligned} \Lambda &= UKU^T && (5.19)\\ &= U\big(\mu_1 K_1 + \mu_2 K_2\big)U^T && (5.20)\\ &= \mu_1 U K_1 U^T + \mu_2 U K_2 U^T && (5.21)\\ &= \mu_1 Z_1 + \mu_1 Z_2 && (5.22) \end{aligned}$$

where U is orthonormal and $\Lambda = diag[\lambda_1, \lambda_2, \ldots, \lambda_n]$ is the diagonal matrix of eigenvalues. Matrices $Z_1, Z_2$ are not diagonal matrices because U does not always diagonalises them. But, as U is the eigenvector matrix of the linear combination of $K_1$ and $K_2$, the off-diagonal elements of $Z_1$ and $Z_2$ cancel each other out (Thomaz et al., 2004; Thomaz and Gillies, 2005). Therefore,

$$\begin{aligned} \Lambda &= diag[\mu_1\lambda_1^1, \mu_1\lambda_2^1, \ldots, \mu_1\lambda_n^1] + diag[\mu_2\lambda_1^2, \mu_2\lambda_2^2, \ldots, \mu_2\lambda_n^2] && (5.23)\\ &= diag[\mu_1\lambda_1^1 + \mu_2\lambda_1^2, \mu_1\lambda_2^1 + \mu_2\lambda_2^2, \ldots, \mu_1\lambda_n^1 + \mu_2\lambda_n^2] && (5.24)\\ & && (5.25) \end{aligned}$$

where $\mu_1\lambda_1^1, \mu_1\lambda_2^1, \ldots, \mu_1\lambda_n^1$ are the variance of $K_1$ spanned by the U eigenvector matrix and $\mu_2\lambda_1^2, \mu_2\lambda_2^2, \ldots, \mu_2\lambda_n^2$ are the variance of $K_2$. In order to maximize Eqn-(5.16), we need to maximize the individual eigenvalues of the combined covariance matrix. So, effectively it implies that we need to maximize each of the $(\mu_1\lambda_i^1 + \mu_2\lambda_i^2)$ terms. As stated previously, the eigenvalues of positive semi-definite matrices are non-negative. We have used kernel functions to compute similarities, which resulted in our similarity matrices being positive semi-definite. Since this is a convex combination of two terms, and all the eigenvalues are non-negative, therefore, in order to maximize it, we just need to take the maximum out of both the terms because,

$$(\mu_1\lambda_i^1 + \mu_2\lambda_i^2) \leq \max(\lambda_i^1, \lambda_i^2) \tag{5.26}$$

when both $\mu_i, \lambda_i$ are positive. Therefore, the maximum entropy is obtained at either ($\mu_1$=0, $\mu_2 = 1$) or ($\mu_1$=1, $\mu_2 = 0$) for each eigenvalue, i.e., we do not take the combination of both the terms but only one of them which is the maximum. To summarise, our matrices are not combined using any particular values of $\mu_1, \mu_2$ but by just picking the maximum of both the variances spanned by U.

Till now we discussed the theoretical justification of the technique, next we discuss the practical aspects of its implementation.

### 5.4.1 Algorithm

From the discussion of the preceding section it is clear that in order to calculate $U$, we need a $K$ which is an unbiased (a=b) linear combination of two similarity matrices, i.e., has equal contribution from both the matrices. Since any unbiased combination gives the same set of eigenvectors we have chosen $a = b = 1$. The final algorithm is described in Algorithm-4.

**Require:** Similarity Matrices ($K_1$ and $K_2$)

1: Calculate the eigenvectors $U$ of matrix $K$ obtained by $K = K_1 + K_2$.

2: Use this $U$ to calculate the variance contribution of both $K_1$ and $K_2$. These are

$$diag[UK_1U^T] = diag[\lambda_1^1, \lambda_2^1, \ldots, \lambda_n^1] \quad (5.27)$$

$$diag[UK_2U^T] = diag[\lambda_1^2, \lambda_2^2, \ldots, \lambda_n^2] \quad (5.28)$$

3: Now form the final eigenvalue matrix $Z$ by choosing the maximum eigenvalues from each diagonal matrix (5.27 and 5.28).

$$Z = diag[max(\lambda_1^1, \lambda_1^2), max(\lambda_2^1, \lambda_2^2), \ldots, max(\lambda_n^1, \lambda_n^2)]$$

4: Finally, compute the maximum entropy matrix

$$K^{ME} = UZU^T$$

**Algorithm 4:** Maximum Entropy Similarity matrix Integration

The principal idea here is that we keep the dominant eigenvalues, while getting rid of the smaller, and hence unreliable ones, and replacing them with better ones from the other dataset.

## 5.5 Datasets and Methodology

We have used the same datasets that were used in the previous chapter as discussed in Section-4.2. They are the yeast microarray datasets (Gasch et al., 2000; Spellman

| Dataset Pair | Number of genes |
|---|---|
| Stress and ChIP-chip | 2346 |
| Stress and PPI | 4508 |
| Stress and Yeastract | 5654 |
| Cell-cycle and ChIP-chip | 2346 |
| Cell-cycle and PPI | 4508 |
| Cell-cycle and Yeastract | 5654 |

Table 5.1: Number of genes considered for individual dataset pairs

et al., 1998), DNA-binding dataset (Harbison et al., 2004), PPI dataset (from MIPS Comprehensive Yeast Genome Database (CYGD)) (Gueldener et al., 2006) and the TF-gene interactions (YEASTRACT) (Teixeira et al., 2006).

While in the previous chapter, we used a full set of genes from the microarray datasets and applied available constraints on them, in this chapter we are unable to use the full set of genes. This is because now we are combining two matrices which must be similar in size. Therefore, We first pre-process and find common genes between pairs of datasets that are being combined. In the previous chapter, after removing the NORFs, the Stress dataset had 6251 genes and Cell-cycle dataset had 6257 genes. They had 156 and 60 experiment counts respectively. The counts get changed after finding common sets of genes among dataset pairs. The final number of common genes when pairing microarray datasets with ChIP-chip, PPI and Yeastract datasets are shown in Table-5.1.

After pre-processing, for the microarray datasets, we compute the similarity matrices from both of them using parameters obtained by the optimization procedure in the previous chapter. For the PPI, Yeastract and ChIP-chip datasets that are in the form of pairwise interactions, we have used the total within-cluster sum of square distances as we did not have access to original data vectors but only the similarity (or adjacency) graph. For these datasets, we need to do the optimization of the parameter for computing the diffusion matrix. As shown in Equation-5.8 we need to find the optimum value for $\beta$. To do this, we first compute diffused matrices for a range of $\beta$ values and then we do spectral clustering on each of these diffused matrices to find the best parameter, which is the one that yields the best cluster quality. This optimization is different from the microarray dataset ones because here we don't have the original data vectors because of the nature of graphical data where only links are specified. Because of this, we can't use distance based optimization

techniques. So we have used a simpler and straightforward metric called *withinss* in order to judge the cluster quality. For a clustering run of the algorithm, we compute this by finding the cumulative sum across all the clusters of sum of squared distance of all points in a cluster from its cluster centre.

$$withinss = \sum_{i=1}^{k} \sum_{x_{ij} \in \mathbf{C}_i} (x_{ij} - \bar{x}_i)^2 \tag{5.29}$$

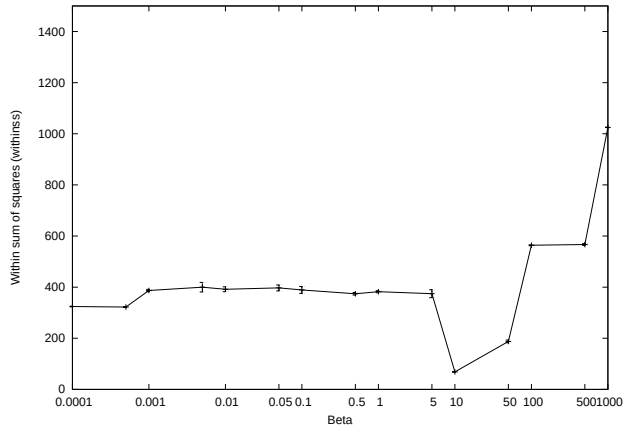
where $C_1, \ldots, C_k$ are the different clusters and $\bar{x}_i$ are the cluster centres. Since we do not have access to the data points, we compute the *withinss* of the vectors on which k-means clustering is done for spectral clustering (refer Step-6 of Algorithm-2). A better clustering will have a lower *withinss* value (hence more compact).

Once we have the similarity matrices from both datasets that are being integrated, we merge both of them using the maximum entropy technique as discussed in Algorithm-4. We then use spectral clustering on resulting similarity matrix to get our final clusters and then validate the biological significance of our results using the Gene Ontology annotations like the previous chapter.

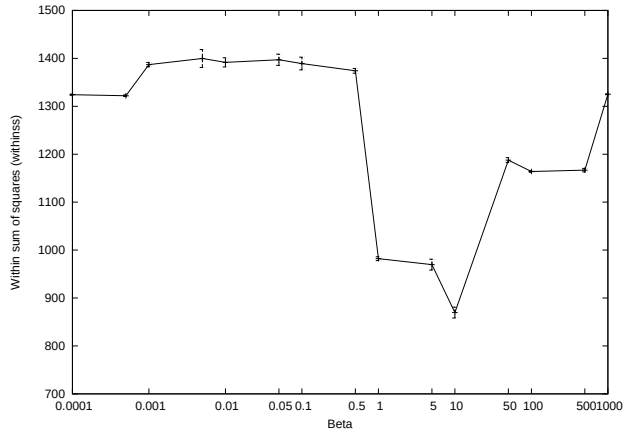
### 5.5.1 Parameter optimisation results

#### ChIP-chip, PPI and Yeastract datasets

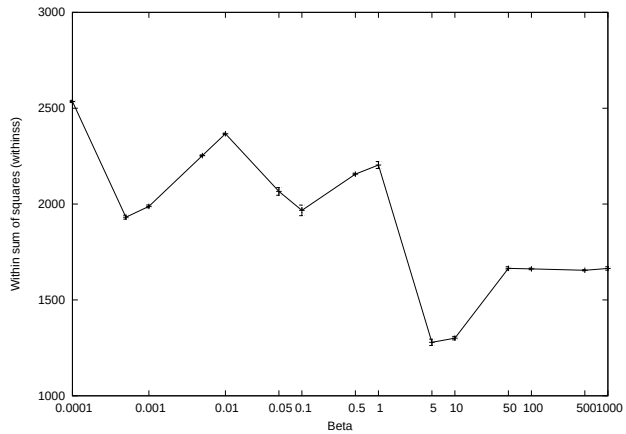
As seen in Figure-5.1(a), for the ChIP-chip dataset, the optimum value (smallest withinss value implies tightest clustering) is at $\beta = 10$ where the total withinss falls significantly. It also has one of the smallest standard deviations at that point. For the PPI dataset, as seen in Figure-5.1(b), the optimum value again is at $\beta = 10$ where it has the smallest withinss value. Even though the standard deviation is not the lowest, yet we chose it because the value is significantly lower than its neighbours. Figure-5.1(c) for the Yeastract dataset, has a optimum range between $\beta = 5$ and $\beta = 10$, where the optimum value is at $\beta = 5$ while the combination of small value and small standard deviation is at $\beta = 10$. We chose $\beta = 5$ because of the smallest withinss value.




(a) ChIP-chip $\beta$ optimization using total within-cluster sum of square distances




(b) PPI $\beta$ optimization using total within-cluster sum of square distances




(c) Yeastract $\beta$ optimization using total within-cluster sum of square distances

Figure 5.1: ChIP-chip, PPI and Yeastract datasets: Beta optimization

## 5.6 Biological Significance Analysis

### 5.6.1 Numerical Biological Significance comparison (using mutual information)

#### Results

We follow the same procedure that was followed in the previous chapter (refer Section-4.4.1) for biological validation of the resulting clusters using the technique based on mutual information by Gibbons and Roth (2002). The results of the integration of the stress microarray dataset with ChIP-chip, PPI and Yeastract are in Table-5.2. We have reported percentage gains in index values before and after integration for each pair of datasets being integrated. In order to compare our results with the semi-supervised technique, we also reran the semi-supervised experiments on the filtered (common) gene sets. For each pair of dataset, the first one is referred to as A and the second as B. For example, in the first full row of Table-5.2, Stress dataset is referred to as A, while the ChIP-chip dataset is called B.

The results in Table-5.2 clearly indicate that both our data integration techniques help improve the quality of the resulting clustering biologically. Both datasets A and B have improved biological significance after integration except in one scenario where the ChIP-chip is integrated with the Stress dataset. MaxEnt integration has resulted in improvements for the Stress datasets with every other dataset except for the ChIP-chip dataset. We also observe that the MaxEnt technique has outperformed the semi-supervised one except the ChIP-chip dataset combination where the semi-supervised technique shows a small improvement. Also, more than the Stress datasets, the improvement in the B datasets i.e. the ChIP-chip, PPI and Yeastract datasets, when combined, is better.

For the Cell-cycle dataset, as shown in Table-5.3, the results are somewhat different. Here, too both the datasets, A and B, have improved biological significance after integration. The MaxEnt technique has resulted in improved biological significance after combination in all the cases. However, the results for MaxEnt are not always better than the semi-supervised all the time. While for the Stress dataset, the semi-supervised integration with ChIP-chip dataset had performed better than the MaxEnt one, for the Cell-cycle dataset, it's the opposite. MaxEnt is better than the

| Datasets | | | Semi-supervised integration | | | Maximum entropy integration | | |
|---|---|---|---|---|---|---|---|---|
| | Dataset A | Dataset B | After integration | Gain for A | Gain for B | After Integration | % Gain for A | % Gain for B |
| Stress & ChIP-chip | 62.3 | 27.5 | 59.6 | -4.3% | 116.72% | 59.4 | -4.65% | 116% |
| Stress & PPI | 62.3 | 29.2 | 66.5 | 6.74% | 127.73% | 75.1 | 20.54% | 157.19% |
| Stress & Yeastract | 62.3 | 38.0 | 94 | 50.88% | 147.36% | 109.0 | 74.95% | 186.84% |

Table 5.2: Stress microarray dataset: Comparison of Biological significance index values before and after maximum entropy data integration. Both our data integration techniques help improve the quality of the resulting clustering biologically. Both datasets A and B have improved biological significance after integration except in one scenario where the ChIP-chip is integrated with the Stress dataset.

| Datasets | | | Semi-supervised integration | | | Maximum entropy integration | | |
|---|---|---|---|---|---|---|---|---|
| | Dataset A | Dataset B | After integration | Gain for A | Gain for B | After Integration | Gain for A | Gain for B |
| Cell-cycle & ChIP-chip | 39.1 | 27.5 | 39.7 | 1.53% | 44.36% | 48.2 | 23.27% | 75.27% |
| Cell-cycle & PPI | 39.1 | 29.2 | 40.9 | 4.60% | 40.06% | 39.3 | 0.51% | 34.58% |
| Cell-cycle & Yeastract | 39.1 | 38.0 | 59.3 | 51.66% | 35.94% | 47.6 | 21.73 % | 25.26% |

Table 5.3: Cell-cycle microarray dataset: Comparison of Biological significance index values before and after maximum entropy data integration. Both the datasets, A and B, have improved biological significance after integration. The MaxEnt technique has resulted in improved biological significance after combination in all the cases.

semi-supervised one for the ChIP-chip dataset combination. For PPI and Yeastract combinations, the semi-supervised is better.

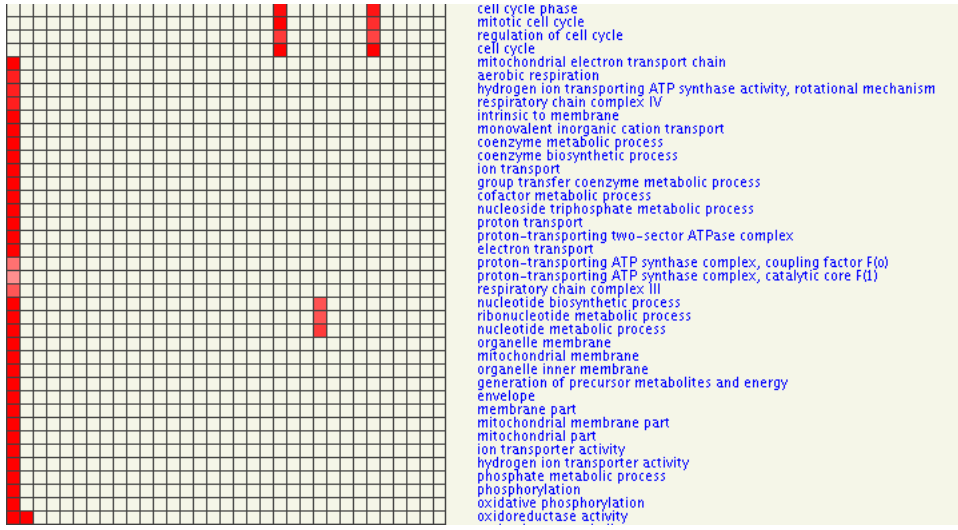
As we discussed earlier this technique merges two datasets by taking their dominant eigenvectors. The PPI and Yeastract are both curated datasets. Most of the curated datasets are taken from experiments that are conducted in non-stress environments to study the regular activities of genes. Therefore, they are more similar to the Cell-cycle dataset. Because of this similarity, the Cell-cycle dataset has not gained much from the others. In the case of Stress, they were very dissimilar and hence both gained information from each other which led to improvement in their scores.

### 5.6.2 Qualitative Biological Significance (using Gene Ontology annotations)

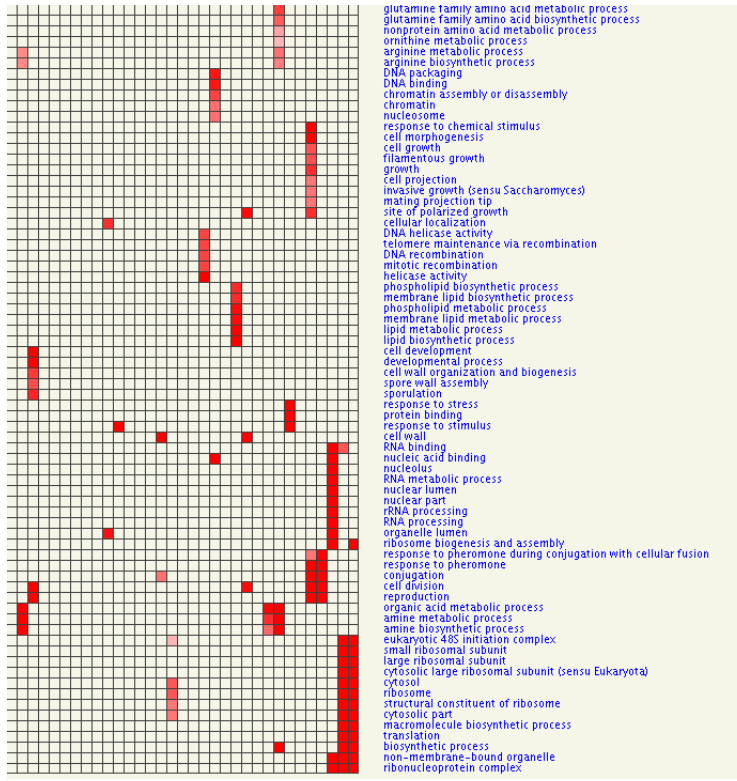
We put the final clusters of genes (before and after combination) through Genomica which is a tool to analyze the characteristics of resulting clustering using Gene Ontology and has been detailed in the previous chapter. The images in Figures-5.2 to 5.4 are stress related while Figures-5.5 to 5.7 are cell-cycle related.

#### Stress vs Stress and ChIP-chip

Figure-5.2 shows the enrichments when the Stress dataset is combined with the ChIP-chip dataset. We observe that nucleic acid binding, ribosome biogenesis and assembly, macromolecule biosynthetic process, translation and biosynthetic process are common significant enrichments observed in results, both before and after integration. Processes like mitotic cell cycle, regulation of cell cycle, nucleotide biosynthetic process, ribonucleotide metabolic process, oxidoreductase activity, response to pheromone, conjugation, cell division, reproduction, organic metabolic process, amine metabolic process, and amine biosynthetic process are significantly enriched in combined Stress and ChIP-chip dataset as compared to Stress only.

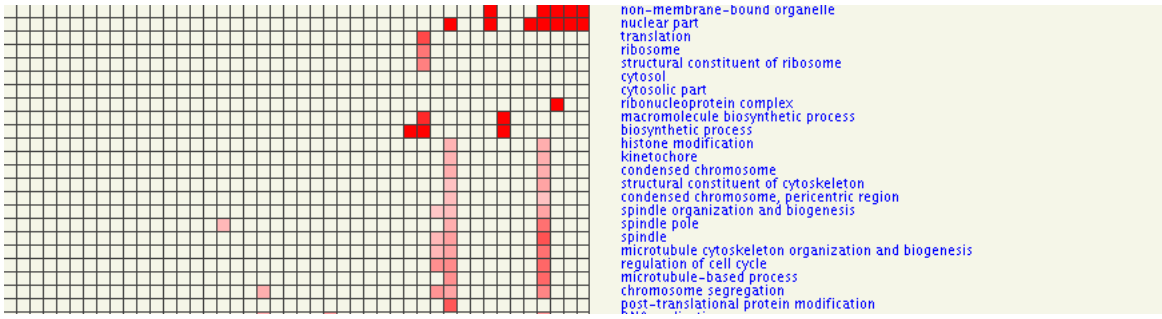



(a) Section of the image showing significant enrichment




(b) Section of the image showing significant enrichment

Figure 5.2: Sections of the image showing significant enrichment in Stress dataset combined with ChIP-chip dataset.

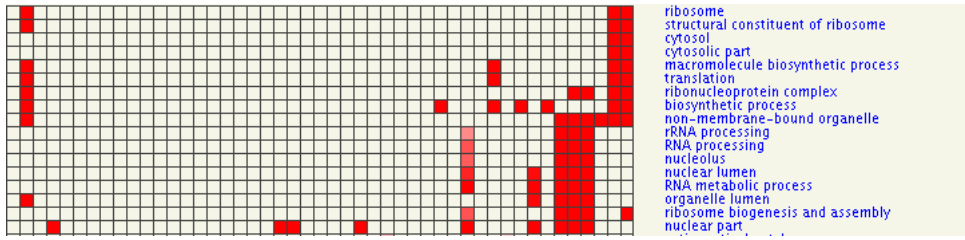



(a) Section of the image showing significant enrichment

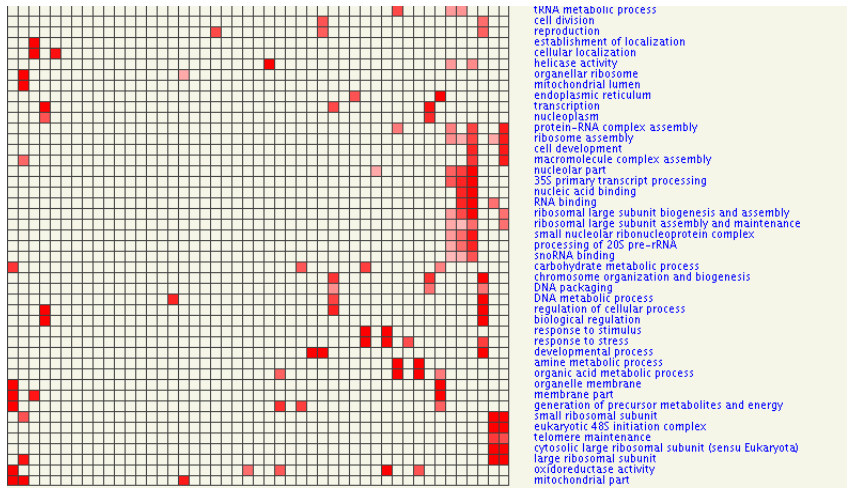



(b) Section of the image showing significant enrichment

Figure 5.3: Sections of the image showing significant enrichment in Stress dataset combined with PPI dataset.




(a) Section of the image showing significant enrichment




(b) Section of the image showing significant enrichment

Figure 5.4: Sections of the image showing significant enrichment in Stress dataset combined with Yeastract dataset.

### Stress vs Stress and PPI

Figure-5.3 shows the enrichments when the Stress dataset is combined with the PPI dataset. Protein RNA complex assembly, macromolecule complex assembly, nucleic acid binding, DNA metabolic process, rRNA processing, RNA processing, ribosome biogenesis and assembly and RNA metabolic process are the common enrichments observed before and after combination of datasets. After combination of Stress and PPI datasets, the significantly enriched processes are oxidoreductase activity, carbohydrate metabolic process, transcription from RNA polymerase II promoter, transcription regulator activity, response to stimulus, response to stress, protein binding, mitotic cell cycle, cell cycle, M phase, cell cycle phase, developmental process, cell development, chromosome organization and biogenesis, biological regulation, regulation of cellular process and transcription.

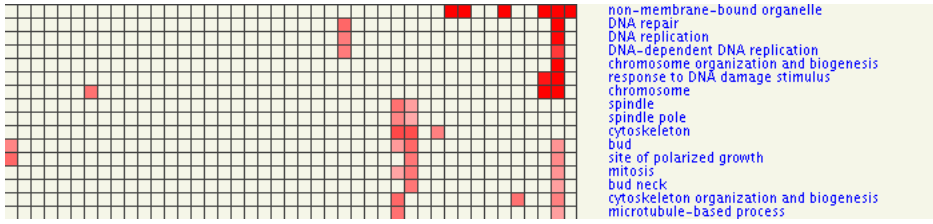
### Stress vs Stress and Yeastract

Figure-5.4 shows the enrichments when the Stress dataset is combined with the Yeastract dataset. Macromolecule biosynthetic process, translation, rRNA processing, RNA processing, RNA metabolic process, ribosome biogenesis and assembly, protein RNA complex assembly, ribosome assembly, 35S primary transcript processing, nucleic acid binding, RNA binding, processing of 20S pre-rRNA and DNA metabolic process are the common enriched processes both before and after combination. Processes like cellular localization, transcription, cell development, macromolecule complex assembly, ribosomal large subunit biogenesis and assembly, ribosomal large subunit assembly and maintenance, sno RNA binding, carbohydrate metabolic process, chromosome organization and biogenesis, DNA packaging, regulation of cellular process, biological regulation, response to stimulus, response to stress, developmental process, amine metabolic process, generation of precursor metabolites and energy, telomere maintenance, oxidoreductase activity and organic acid metabolic process are significantly enriched after the Stress and Yeastract datasets are combined.

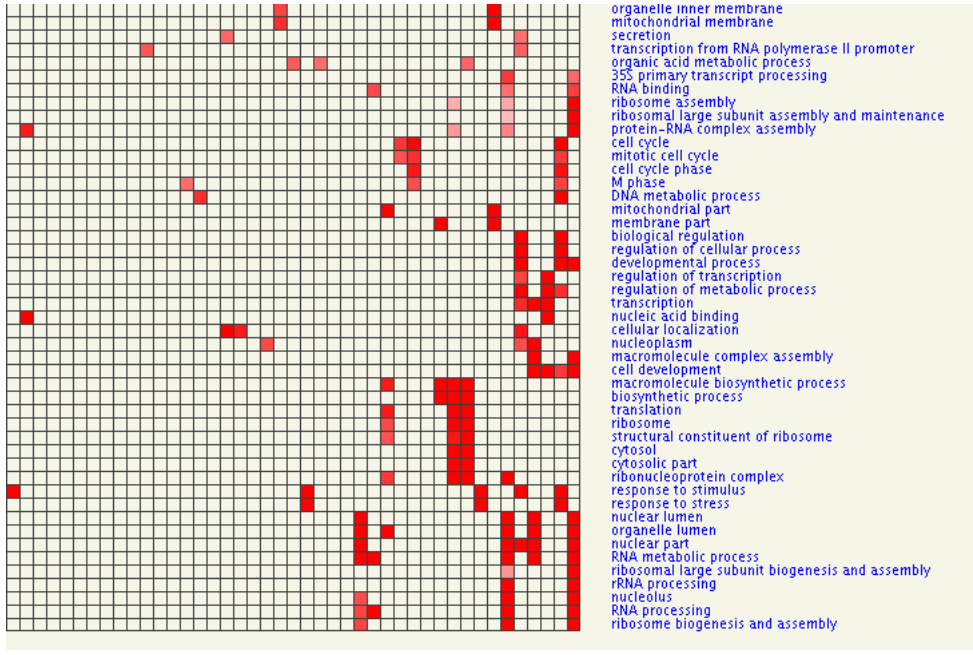



(a) Section of the image showing significant enrichment

Figure 5.5: Sections of the image showing significant enrichment in Cell-cycle dataset combined with ChIP-chip dataset.

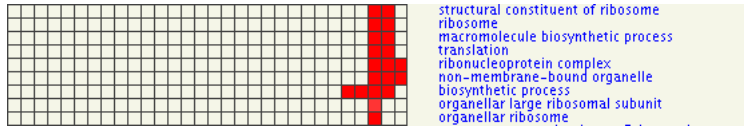



(a) Section of the image showing significant enrichment




(b) Section of the image showing significant enrichment

Figure 5.6: Sections of the image showing significant enrichment in Cell-cycle dataset combined with PPI dataset.




(a) Section of the image showing significant enrichment




(b) Section of the image showing significant enrichment

Figure 5.7: Sections of the image showing significant enrichment in Cell-cycle dataset combined with Yeastract dataset.

### Cell-Cycle vs Cell-Cycle and ChIP-chip

Figure-5.5 shows the enrichments when the Cell-cycle dataset is combined with the ChIP-chip dataset. Translation, macromolecule biosynthetic process and biosynthetic process are the only three common significant enrichments observed both before and after combination of datasets. When Cell-Cycle and ChIP-chip datasets are combined, we observe that ribosome biogenesis and assembly, cell cycle, cell cycle phase, mitotic cell cycle, regulation of cell cycle, cell wall organization and biogenesis, cell wall division, reproduction, conjugation, response to pheromone, organic acid metabolic process, amine metabolic process and response to stimulus processes have significant enrichments.

### Cell-Cycle vs Cell-Cycle and PPI

Figure-5.6 shows the enrichments when the Cell-cycle dataset is combined with the PPI dataset. DNA metabolic process, macromolecule biosynthetic process, biosynthetic process, translation, RNA metabolic process, rRNA processing, RNA processing are a few of the common enrichments in both before and after integration. However, the response to DNA damage stimulus, cell cycle, mitotic cell cycle, cell cycle phase, M phase, biological regulation, regulation of cellular process, developmental process, regulation of transcription, regulation of metabolic process, transcription, cellular localization, cell development, response to stimulus, response to stress and ribosome biogenesis assembly are processes which showed significant enrichment in the combined Cell-cycle and PPI dataset.

### Cell-Cycle vs Cell-Cycle and Yeastract

Figure-5.7 shows the enrichments when the Cell-cycle dataset is combined with the Yeastract dataset. The macromolecule biosynthetic process, translation, biosynthetic process, RNA metabolic process and DNA metabolic process are common enriched processes both before and after combination of datasets. After Cell-cycle and Yeastract datasets are combined, we observe that processes related to response to stimulus, cell cycle, ribosome biogenesis and assembly and generation of precursor metabolites and energy are significantly enriched.

The goal of integration is to have more biologically relevant clusters from which hypotheses could be derived to be carried out and validated in wet labs. All our result data is available upon request.

## 5.7 Related Work and Discussion

Combining evidence from multiple biological datasets is not a new phenomenon. Earlier efforts in this direction related to clustering were classified under co-clustering. In this technique, the datasets are combined by assigning equal or varying weights to each dataset. For distance based clustering techniques, individual distances derived from each datasets are merged in order to come up with a new one which is then used for clustering. Hanisch et al. (2002) proposed a distance metric that combines information from expression data and biological networks and uses it for clustering genes. They define a graph distance function on a metabolic network derived from MIPS (Gueldener et al., 2006) and combine it with a correlation-based distance function for microarray gene expression measurements. They assigned equal weights to both the sources and then used the resulting distance for hierarchical clustering. They show that their technique was able to find biologically meaningful clusters. Huang and Pan (2006) developed a similar algorithm in which instead of combining the two information sources with equal weights, they used a *shrinkage* approach with the genes belonging to the same functional classes assigned zero distance (maximal similarity) and the rest of the genes using the distance calculated from the microarray data. This is then used for K-medoids clustering on simulated as well as real data for gene function prediction. Brameier and Wiuf (2007) proposed co-clustering based on a combined distance metric from microarray gene expression data and Gene Ontology terms for self-organising map (SOM). Apart from distance based clustering, various researchers have combined datasets using model based clustering as well (Pan, 2006).

Kernel integration has been used in the field of supervised learning in order to combine kernels from different datasets. Holloway et al. (2006) used it for predicting the TF binding locations on DNA. They have used 18 different datasets - both sequence and non-sequence (expression, GO) and calculated kernels from them. The goal was to combine the kernels and then use the final combination for classification using support vector machines (SVMs) . They have used each kernel individually

and calculated the *F statistic* (a widely used measure for performance of a classifier). In order to combine the kernels, these F values were used as weights for each of the kernel. One of the drawbacks is that the F-statistic does not take into account the relationship of the variables but only a macro view of the performance of the whole kernel encoding. They reported that combining all datasets resulted in 73% coverage of known interactions.

A more mathematically sound approach to kernel integration was developed by Lanckriet et al. (2004b). They have formulated the optimization problem as a convex optimization problem and then used semi-definite programming (SDP) to solve it. The technique is both statistically sound and computationally efficient. They used this technique in order to classify different classes of proteins (membrane vs ribosomal) using kernels derived from protein sequence, microarray expression and protein-protein interaction data. They have reported an improvement in the classification results when the kernels are combined as compared to individual kernels. The biggest drawback of such techniques is that the weights that are assigned to each dataset do not take into account the correlation between different variables across datasets and assigns one common weight for the whole dataset. Our technique tries to solve it by picking only the highest eigenvectors.

Lewis et al. (2006) did an investigation of weighted and unweighted kernel combination in classifying GO terms associated with protein sequences using SVM. They came to the interesting conclusion that for this particular task, an unweighted combination of kernels is better than weighted ones. In order to compute the weights they too have used the SDP technique. Sun et al. (2008) developed a technique to learn an optimal diffusion kernel as a convex combination of many kernels constructed from biological networks and then used the optimal kernel for protein function prediction. They report superior performance for the combined kernel in comparison to individual kernels.

Most of the above kernel combination techniques are based on *supervised learning* and the individual kernel weights are optimized using some training data. But Kernel integration in an unsupervised setting (when training data is unavailable) is hard because we have no way to compute the individual weights.

Tsuda and Noble (2004) used a very different approach to guess a kernel matrix from incomplete data. The underlying approach is that some values in the matrix

are known and they try to fill in the rest so that the resulting matrix has maximum possible entropy. They used this to compute kernels from PPI data and metabolic network and used these for the classification task and report that the maximum entropy kernel beats the diffusion kernel in classification accuracy. Fujibuchi and Kato (2007) have used the maximum entropy devised by Tsuda and Noble (2004) and applied it on three microarray datasets (heterogeneous kidney carcinoma, noise introduced leukemia and heterogeneous oral cavity carcinoma metastasis) for the purpose of evaluating its classification performance as compared to single kernels (linear, polynomial and rbf). They report better overall performance for the maximum entropy kernel compared to others in classification performance.

In practise, the principle of maximum entropy is useful when applied to verifiable information. A piece of information is verifiable if it can be determined whether a given distribution is consistent with it. Given this information, the maximum entropy procedure consists of seeking the probability distribution which maximizes information entropy, subject to the constraints of the information. This constrained optimization problem is typically solved using the method of Lagrange multipliers. Both these approaches to kernel approximation (Tsuda and Noble, 2004; Fujibuchi and Kato, 2007) using maximum entropy are based on maximising the entropy subject to certain constraints. However, this is very different from our approach where we do not treat it as a entropy maximisation subject to certain constraints. We try to assume a distribution with maximum entropy as the ideal distribution when no other information is available in order to combine the similarity matrices. So, when we have multiple sets of evidence, e.g. PPI and microarray data, we try to combine them through their similarity matrices such that the resulting distribution (of the combined similarity matrix) has the maximum entropy.

The ideal scenario of its use is where one of the datasets is a very noisy sample while the other one is a *compendium* or reference dataset collected over time from various sources. The compendium acts as an average of all known observations, which when combined using this technique with the sample, fills in the eigenvalues about which the sample dataset is most *confused* about. So, the sample dataset gets to keep all its prominent eigenvalues while borrowing ones from the compendium about which it is not so confident.

## 5.8 Conclusion

We have proposed a technique to integrate two diverse datasets where one is available in the form of vectorial data while the other is available in the form of a graph. The core idea is based on work done by Thomaz et al. (2004). While they had used it to combine covariance matrices where one of them is singular, we have extended its use to general kernel combination after observing the similarities between the properties of a covariance matrix and a similarity matrix obtained using a kernel function. Our results indicate that the proposed technique leads to better biological significance of resulting clusters in most circumstances.

Integrating such diverse datasets is not possible unless we resort to some kind of normalization. We have used similarity functions to compute similarity matrices for these datasets as the *normalization* step. We have used the same techniques that we used in the previous chapter in order to compute the similarity function parameters.

While in the supervised setting we have an objective function that we can use to optimize the contribution of each of the similarity functions, in an unsupervised setting there is no such facility. Hence, most previous works have used ad-hoc techniques in order to integrate different datasets. We argue that under such a setting when no further evidence is available to assign weights to individual datasets then the *principle of maximum entropy* is the most natural and valid choice. Apart from conceptual elegance, other benefits of this technique are that no time consuming optimization is required to search for contributions of each dataset and fairly simple and intuitive linear algebra computations are needed. Since our technique is quite generic, in future, our work can be extended to integrating other sources of data, for example the similarity between the promoter sequences of genes.

One of the key shortcomings for the biological validation of our results is that it is not possible to get datasets that were generated on the same strains under similar experimental conditions. Both our datasets were compiled by independent researchers. When datasets that are generated under similar conditions start becoming available we would be more confident in assessing the biological significance of the results.

There is no gold-standard for validating the clusters and there are no reference datasets on which we can compare the results with other techniques. Gene ontology while being one of the more informative sources of validation is still an indirect

validation technique. Also, there is a fundamental limitation of the underlying experimental techniques, since microarrays themselves do not represent a single time point, but rather the integration of gene activity over a time period.

# Chapter 6

# Summary and Future Work

## 6.1 Summary

The theme of this thesis is data integration. We started with outlining an empirical technique to calculate functional similarity of datasets using the concept of cluster similarity. This could be used as an index of microarray dataset similarity. We have also showed that similarity values gradually fall with increasing fraction of dissimilar data. We have established that as more diverse data-sets are merged then the similarity to individual data-sets (which have more local patterns) is reduced and the dominant ones overshadow the weaker signals. So, before taking a blind integrative approach, much care should be taken to ensure that we mix only similar types of data. We should also be careful about the choice of normalization method. In our results we demonstrated that normalization can distort the data and affect the resulting clusters significantly.

In order to integrate different types of datasets, we have proposed a technique to integrate two different types of datasets where one is acting as a source of supervision on the clustering of the other. The source of supervision is in the form of binary constraints derived from DNA-binding, PPI and TF-gene interactions data, and are applied on microarray data. We computed the biological significance of the combined datasets using Gene Ontology annotations. The results indicate that both the Stress and Cell-cycle microarray datasets when combined with PPI and Yeastract constraints lead to better biological significance index values. However,

when combined with ChIP-chip constraints it leads to slightly degraded biological significance for both. Based on the results we could say that the ChIP-chip dataset is probably the noisiest and does not bring much new information leading to reduced significance for the original datasets.

Finally, we have proposed another technique to integrate two diverse datasets where one of the datasets is non-vectorial. For this, we have used the principle of maximum entropy considering it as the most valid approach under the unsupervised clustering setting where we have no other evidence regarding the weights to be assigned to individual datasets. Again, we computed the biological significance of the combined datasets using Gene Ontology annotations. Like the semi-supervised technique, the MaxEnt integration has resulted in improvement of biological significance for both the Stress and Cell-cycle datasets when combined with other datasets (except for the ChIP-chip dataset). We also observe that the MaxEnt technique has outperformed the semi-supervised technique for the Stress dataset (except when combined with the ChIP-chip dataset) while for the Cell-cycle dataset, the results for MaxEnt are not always better than the semi-supervised integration.

The results clearly indicate that both our proposed data integration techniques help improve the biological significance of the resulting clusters.

## 6.2 Challenges and Future Directions

**Holistic Data Integration** Transcriptional regulation occurs at multiple points from transcription to actual protein synthesis. It is well known that transcription activity (mRNA concentration) alone is not a perfect indicator of protein concentration (Griffin et al., 2002) as there are many post translational factors, e.g. mRNA stability, protein degradation, post-translational modifications, that affect the process. As more protein concentration data (*proteome*) and newer types of data, e.g. nucleosome positions (Segal et al., 2006; Field et al., 2008; Kaplan et al., 2008) become available, we need to develop techniques that can integrate all these and future sources of data in order to develop more precise models of regulation. Some of the ways in which our work could be extended are detailed below.

- For semi-supervised clustering, our work can be extended by using other

sources as prior knowledge, for example the constraints derived from known genetic interactions based on metabolic interaction data or pathway data. In future, when different types of data from experiments conducted simultaneously become available, the reliability of our technique would increase.

- For maximum entropy integration, other sources of gene similarity could be derived from known genetic interactions based on metabolic interaction data or pathway data or similarity of promoter gene sequences (Vert et al., 2006) and then used with the model discussed in Chapter-5.
- We have used the maximum entropy technique in order to combine two similarity matrices and used this integration for clustering which is known as an unsupervised classification technique in machine learning literature. We would also like to explore the possibilities of this integrated matrix which has all the properties of a *kernel* to do SVM based classification and compare the results to other methods of kernel combination (Lanckriet et al., 2004b) and shrinkage based methods.
- Many researchers have used microarray data for cancer classification. Covariance matrix estimation is essential for many of these classification techniques, e.g. linear discriminant analysis (LDA) and regularized discriminant analysis (RDA) as they involve computing the inverse of the covariance matrix. If the dataset has a large number of variables but only few samples, it is known as the "Small $n$, Large $p$" or $n \ll p$ problem. Microarray datasets are a typical case of this because the number of genes (p) is very large while the number of available microarray samples (n) is very limited. In such a case, the estimated covariance matrix looses its full rank (rank deficient). This leads to many unwanted properties. If the covariance matrix is not full rank then it is not positive definite anymore, which is a requirement for many algorithms that might use this covariance matrix as a similarity matrix, e.g. kernel based classifiers (SVM). Another bigger problem is that this rank deficient covariance matrix is not invertible. There has been a lot of research in proposing better estimators of the covariance matrix, e.g. Schäfer and Strimmer (2005). As shown by Thomaz and Gillies (2005) where this formulation was first used for face detection, the maximum entropy principle is an ideal candidate for an estimator of the covariance matrix when some prior knowledge is

available. We intend to use it for cancer classification and compare the results with existing shrinkage based methods e.g., Tai and Pan (2007), who combined covariance matrices and then used the resulting covariance matrix in RDA for cancer classification.

**System Dynamics** Another growing area of research is based on more detailed modelling using reaction kinematics of gene products. This could help in understanding not only the qualitative models of regulation but also detailed quantitative ones.

**Prior Knowledge** Most of the past research in molecular biology involved working with a small number of genes. This has led to the accumulation of a huge amount of biological knowledge. Genome wide global modelling of regulation has mostly used this high quality data for validation of the results. Apart from validation, this prior biological knowledge could be used to produce better models by integrating them along with other sources of experimental data.

**Incoherently Integrated Datasets** Orphanides and Reinberg (2002) argue very explicitly that there is no single model of regulation and each cell process has evolved its own detailed regulation model. Moreover, we usually observe only a few snapshots of these processes, which makes it very hard to reconstruct the underlying mechanisms. The data that is integrated comes from various laboratories where experiments are done under different conditions and with different platforms. We must be very careful while integrating such data and care must be taken to check beforehand if the data shows similar trends. The above conditions are some of the reasons why some researchers (Dolinski and Botstein, 2005) have found the amount of overlap in the results based on different datasets to be small.

**We Don't Know Biology fully!** Another big challenge that inhibits precise modelling of the process is lack of available data about the 3D structure of chromatin (DNA). Apart from the promoter sequence, the 3-D structure of chromatin decides whether a transcription factor is allowed access to a certain position or not. Sometimes a transcription factor itself facilitates changes in the chromatin structure that allows it access to the promoter sequence. Better techniques of modelling the chromatin structure will definitely aid in a better regulation model.

**Complexities of Higher Organisms** Simple unicellular organisms have the advantage that the sample of cells used in an experiment is homogeneous. Each cell is assumed to be performing the same regulatory actions. Based on the results so far, we are far away from a fully comprehensive model of regulation in even simpler organisms like yeast. Higher organisms pose other challenges because of cell and tissue heterogeneity. Apart from this, multi-cellular organisms are a big challenge as it's very difficult to segregate the expression of one cell from its neighbouring ones. Most genomic techniques measure an average signal in a sample from a cell population. When analysing a heterogeneous tissue, this is a big concern as individual signals from different cell types are obfuscated. Moreover, the averaging effect introduces an additional source of noise as the proportions of different cells are different across samples.

**Validation of Results** Interpretation of results is very hard because it is done indirectly. Because of the huge quantity of hypothesis that could be derived from the clustering results, it is not possible to validate them experimentally. So most of the validation is done indirectly using other sources of data, e.g. gene ontology annotations. Even though gene ontology databases have contributed significantly to the creation of a common language to describe properties, we do not have annotations for all genes and gene products. Without high quality annotations, the best algorithms are rendered useless as we can never know how accurate they are. Another issue is that no standardised data sets exist on which existing and future techniques could be compared. Recent years have seen a huge increase in research on innovative techniques. Yet, there are no gold standards in validation unlike standardised data sets in other fields like Information Retrieval. We need more standard datasets and better validation metrics to more fruitfully analyse the effectiveness of various algorithms and measure their effectiveness progressively.

## 6.3 Final Remarks

Despite all the challenges, high-throughput technologies have changed the research focus from studying a handful of genes to studying interactions at the whole genome level. The explosion in data generation has made Biology quickly move towards becoming an information science. Data integration seems to be the only approach

which can help us understand the complex underlying processes responsible for functioning of organisms. An extensive amount of further research is required both in the measurement and analysis processes to improve our understanding of how genes interact. We have only begun to understand regulation quantitatively and have a long way to go before we can construct fully detailed regulatory network models.

Future research in the area of integration will continue as more data of different types become available. The focus will likely shift towards integration of data from multiple cell types, conditions and even organisms. Apart from integration techniques, future research is likely to move towards better validation of the various techniques and the creation of gold standards against which results can be assessed.

Despite all the challenges, positive results have been achieved with human tumour expression data while studying both individual cancers and, with an integrative approach (Segal et al., 2004), simultaneously studying a large cancer compendium of multiple datasets. These have shown that the future of this area of research is bright.

# Bibliography


Alberts, B., Johnson, A., Lewis, J., Raff, M., Roberts, K., and Walter, P. (2002). *Molecular Biology of the Cell*. Garland.

Angenent, S., Haker, S., Tannenbaum, A., and Kikinis, R. (1999). On the laplace-beltrami operator and brain surface flattening. *IEEE Trans. Med. Imaging*, 18(8):700–711.

Bar-Joseph, Z., Gerber, G. K., Lee, T. I., Rinaldi, N. J., Yoo, J. Y., Robert, F., Gordon, D. B., Fraenkel, E., Jaakkola, T. S., Young, R. A., and Gifford, D. K. (2003). Computational discovery of gene modules and regulatory networks. *Nature Biotechnology*, 21(11):1337–1342.

Basu, S., Bilenko, M., and Mooney, R. J. (2004). A probabilistic framework for semi-supervised clustering. In *KDD04*, pages 59–68, Seattle, WA.

Basu, S., Davidson, I., and Wagstaff, K. (2008). *Constrained Clustering: Advances in Algorithms, Theory, and Applications*. Chapman & Hall/CRC.

Boyd, S. and Vandenberghe, L. (2004). *Convex Optimization*. Cambridge University Press.

Bradley, P. S., Bennett, K. P., and Demiriz, A. (2000). Constrained k-means clustering. Technical report, MSR-TR-2000-65, Microsoft Research.

Brameier, M. and Wiuf, C. (2007). Co-clustering and visualization of gene expression data and gene ontology terms for saccharomyces cerevisiae using self-organizing maps. *J. of Biomedical Informatics*, 40(2):160–173.

Brookes, M. (2005). The matrix reference manual. online, http://www.ee.ic.ac.uk/hp/staff/dmb/matrix/intro.html.

Chen, H., Jiang, G., and Yoshihira, K. (2006). Robust nonlinear dimensionality reduction for manifold learning. *Pattern Recognition, International Conference on*, 2:447–450.

Cherry, J. M., Adler, C., Ball, C., Chervitz, S. A., Dwight, S. S., Hester, E. T., Jia, Y., Juvik, G., Roe, T., Schroeder, M., Weng, S., and Botstein, D. (1998). Sgd: Saccharomyces genome database. *Nucleic Acids Res*, 26(1):73–79.

Chung, F. R. K. (1997). *Spectral Graph Theory (CBMS Regional Conference Series in Mathematics, No. 92) (Cbms Regional Conference Series in Mathematics)*. American Mathematical Society.

Consortium, G. O. (2001). Creating the gene ontology resource: design and implementation. *Genome Res*, 11(8):1425–1433.

de Jong, H. (2002). Modeling and simulation of genetic regulatory systems: a literature review. *J Comput Biol*, 9(1):67–103.

de Leeuw, J. (2005). Modern multidimensional scaling: Theory and applications (second edition). *Journal of Statistical Software, Book Reviews*, 14(4):1–2.

Dhillon, I. S., Guan, Y., and Kulis, B. (2005). A unified view of kernel k-means, spectral clustering and graph cuts. Technical Report TR-04-25, University of Texas Dept. of Computer Science.

Ding, C. and Zha, H. (2008). *Spectral Clustering, Ordering and Ranking: Statistical Learning with Matrix Factorizations*. Springer Publishing Company, Incorporated.

Dolinski, K. and Botstein, D. (2005). Changing perspectives in yeast research nearly a decade after the genome sequence. *Genome Res*, 15(12):1611–1619.

Donath, W. E. and Hoffman, A. J. (1973). Lower bounds for the partitioning of graphs. *IBM J. Res. Dev*, 17(5):420–425.

Eisen, M. B., Spellman, P. T., Brown, P. O., and Botstein, D. (1998). Cluster analysis and display of genome-wide expression patterns. *Proc Natl Acad Sci U S A*, 95(25):14863–14868.

Fiedler, M. (1975). A property of eigenvectors of nonnegative symmetric matrices and its application to graph theory. *Czechoslovak Mathematical Journal*, 25.

Field, Y., Kaplan, N., Fondufe-Mittendorf, Y., Moore, I. K., Sharon, E., Lubling, Y., Widom, J., and Segal, E. (2008). Distinct modes of regulation by chromatin encoded through nucleosome positioning signals. *PLoS Comput Biol*, 4(11):e1000216.

Friedman, N., Linial, M., Nachman, I., and Pe'er, D. (2000). Using Bayesian networks to analyze expression data. *Journal of Computational Biology*, 7(3):601–620.

Fujibuchi, W. and Kato, T. (2007). Classification of heterogeneous microarray data by maximum entropy kernel. *BMC Bioinformatics*, 8:267+.

Gasch, A. P., Spellman, P. T., Kao, C. M., Carmel-Harel, O., Eisen, M. B., Storz, G., Botstein, D., and Brown, P. O. (2000). Genomic expression programs in the response of yeast cells to environmental changes. *Mol Biol Cell*, 11(12):4241–4257.

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

Ge, H., Liu, Z., Church, G. M., and Vidal, M. (2001). Correlation between transcriptome and interactome mapping data from saccharomyces cerevisiae. *Nat Genet*, 29(4):482–486.

Gerstein, M., Lan, N., and Jansen, R. (2002). Proteomics: Enhanced: Integrating interactomes. *Science*, 295(5553):284–287.

Gianchandani, E. P., Papin, J. A., Price, N. D., Joyce, A. R., and Palsson, B. O. (2006). Matrix formalism to describe functional states of transcriptional regulatory systems. *PLoS Comput Biol*, 2(8):e101.

Gibbons, F. D. and Roth, F. P. (2002). Judging the quality of gene expression-based clustering methods using gene annotation. *Genome Res.*, 12(10):1574–1581.

Griffin, T. J., Gygi, S. P., Ideker, T., Rist, B., Eng, J., Hood, L., and Aebersold, R. (2002). Complementary profiling of gene expression at the transcriptome and proteome levels in saccharomyces cerevisiae. *Mol Cell Proteomics*, 1(4):323–333.

Grira, N., Crucianu, M., and Boujemaa, N. (2005). Unsupervised and semi-supervised clustering:a brief survey.

Gueldener, U., Muensterkoetter, M., Oesterheld, M., Pagel, P., Ruepp, A., Mewes, H. W., and Stuempflen, V. (2006). Mpact: the mips protein interaction resource on yeast. *Nucleic Acids Res*, 34(Database issue).

Hanisch, D., Zien, A., Zimmer, R., and Lengauer, T. (2002). Co-clustering of biological networks and gene expression data. *Bioinformatics*, 18 Suppl 1.

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

Hastie, T., Tibshirani, R., and Friedman, J. (2001). *The Elements of Statistical Learning*. Springer Series in Statistics. Springer New York Inc., New York, NY, USA.

Holloway, D. T., Kon, M. A., and DeLisi, C. (2006). Machine learning methods for transcription data integration. *IBM J. Res. Dev.*, 50(6):631–643.

Huang, D. and Pan, W. (2006). Incorporating biological knowledge into distance-based clustering analysis of microarray gene expression data. *Bioinformatics*, 22(10):1259–1268.

Hubert, L. and Arabie, P. (1985). Comparing partitions. *Journal of Classification*.

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

Hunter, L. (1993). Molecular biology for computer scientists.

Husmeier, D. (2003). Sensitivity and specificity of inferring genetic regulatory interactions from microarray experiments with dynamic bayesian networks. *Bioinformatics*, 19(17):2271–2282.

Ihmels, J., Friedlander, G., Bergmann, S., Sarig, O., Ziv, Y., and Barkai, N. (2002). Revealing modular organization in the yeast transcriptional network. *Nature Genet*, 31:370–377.

Imoto, S., Higuchi, T., Goto, T., Tashiro, K., Kuhara, S., and Miyano, S. (2003). Combining microarrays and biological knowledge for estimating gene networks via Bayesian networks. In *Proc. 2nd Computational Systems Bioinformatics*, pages 104–113.

Jain, A. K. and Dubes, R. C. (1988). *Algorithms for clustering data*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA.

Jain, V. and Zhang, H. (2007). A spectral approach to shape-based retrieval of articulated 3d models. *Comput. Aided Des.*, 39(5):398–407.

Jaynes, E. T. (1957). Information theory and statistical mechanics. *Physical Review*, 106(4):620+.

Jaynes, E. T. (1982). On the rationale for maximum entropy methods. In *IEEE Volume V 70, Issue N 9*, pages 939–952.

Jeffreys, H. (1946). An invariant form for the prior probability in estimation problems. In *Proc. Royal Soc. London (A)*, volume 186, pages 453–461.

Johnson, D. H. and Sinanovi, S. (2001). Symmetrizing the kullback-leibler distance. Technical report, IEEE Transactions on Information Theory.

Kamvar, S., Klein, D., and Manning, C. (2003). Spectral learning.

Kaplan, N., Moore, I. K., Fondufe-Mittendorf, Y., Gossett, A. J., Tillo, D., Field, Y., Leproust, E. M., Hughes, T. R., Lieb, J. D., Widom, J., and Segal, E. (2008). The dna-encoded nucleosome organization of a eukaryotic genome. *Nature*.

Kemmeren, P. (2002). Protein interaction verification and functional annotation by integrated analysis of genome-scale data. *Molecular Cell*, 9(5):1133–1143.

Klein, D., Kamvar, S. D., and Manning, C. D. (2002). From instance-level constraints to space-level constraints: Making the most of prior knowledge in data clustering. In *ICML '02: Proceedings of the Nineteenth International Conference on Machine Learning*, pages 307–314, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

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

Kondor, R. I. and Lafferty, J. D. (2002). Diffusion kernels on graphs and other discrete input spaces. In *ICML*, pages 315–322.

Kulis, B., Basu, S., Dhillon, I., and Mooney, R. (2005). Semi-supervised graph clustering: a kernel approach. In *ICML '05: Proceedings of the 22nd international conference on Machine learning*, pages 457–464, New York, NY, USA. ACM Press.

Kullback, S. (1997). *Information Theory and Statistics (Dover Books on Mathematics)*. Dover Publications, New York, USA.

Lanckriet, G. R., De Bie, T., Cristianini, N., Jordan, M. I., and Noble, W. S. (2004a). A statistical framework for genomic data fusion. *Bioinformatics*, 20(16):2626–2635.

Lanckriet, G. R., Deng, M., Cristianini, N., Jordan, M. I., and Noble, W. S. (2004b). Kernel-based data fusion and its application to protein function prediction in yeast. In *Proceedings of the Pacific Symposium on Biocomputing*, pages 300–311.

Lee, T. I., Rinaldi, N. J., Robert, F., Odom, D. T., Bar-Joseph, Z., Gerber, G. K., Hannett, N. M., Harbison, C. T., Thompson, C. M., Simon, I., Zeitlinger, J., Jennings, E. G., Murray, H. L., Gordon, D. B., Ren, B., Wyrick, J. J., Tagne, J. B., Volkert, T. L., Fraenkel, E., Gifford, D. K., and Young, R. A. (2002). Transcriptional regulatory networks in saccharomyces cerevisiae. *Science*, 298(5594):799–804.

Lemmens, K., Dhollander, T., De Bie, T., Monsieurs, P., Engelen, K., Smets, B., Winderickx, J., De Moor, B., and Marchal, K. (2006). Inferring transcriptional modules from chip-chip, motif and microarray data. *Genome Biol*, 7(5).

Lewis, D. P., Jebara, T., and Noble, W. S. (2006). Support vector machine learning from heterogeneous data: an empirical analysis using protein sequence and structure. *Bioinformatics*, 22(22):2753–2760.

Macqueen, J. B. (1967). Some methods of classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, pages 281–297.

Meila, M. and Shi, J. (2000). Learning segmentation by random walks. In *NIPS*, pages 873–879.

Milligan, G. W. and Cooper, M. C. (1985). An examination of procedures for determining the number of clusters in a data set. *Psychometrika*.

Mishra, A. and Gillies, D. (2007). Effect of microarray data heterogeneity on regulatory gene module discovery. *BMC Systems Biology*, 1(Suppl 1):S2.

Mishra, A. and Gillies, D. (2008a). Data integration for regulatory gene module discovery. In Daskalaki, A., editor, *Handbook of Research on Systems Biology Applications in Medicine*. IGI Global, Hershey, PA.

Mishra, A. and Gillies, D. (2008b). Semi supervised spectral clustering for regulatory module discovery. In *Data Integration in the Life Sciences*, pages 192–203.

Mishra, A. and Gillies, D. (2008c). Validation issues in regulatory module discovery. In Lodhi, H. and Muggleton, S., editors, *Elements of Computational Systems Biology*. John Wiley & Sons, Inc., Hershey, PA.

Moler, C. and Loan, C. V. (2003). Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later. *J-SIAM-REVIEW*, 45(1):3–49.

Murphy, K. and Mian, S. (1999). Modelling gene expression data using dynamic bayesian networks.

Myers, C. L. and Troyanskaya, O. G. (2007). Context-sensitive data integration and prediction of biological networks. *Bioinformatics*, 23(17):2322–2330.

Nadler, B. and Galun, M. (2006). Fundamental limitations of spectral clustering. In *NIPS*, pages 1017–1024.

Ng, A. Y., Jordan, M. I., and Weiss, Y. (2001). On spectral clustering: Analysis and an algorithm. In *NIPS*, pages 849–856.

Orphanides, G. and Reinberg, D. (2002). A unified theory of gene expression. *Cell*, 108(4):439–451.

Pan, W. (2006). Incorporating gene functions as priors in model-based clustering of microarray gene expression data. *Bioinformatics*, 22(7):795–801.

Payne, W. E. and Garrels, J. I. (1998). Yeast protein database (ypd): a database for the complete proteome of saccharomyces cerevisiae. *Nucleic Acids Research*, 26:57–62.

Quackenbush, J. (2006). Computational approaches to analysis of dna microarray data. *Methods Inf Med*, 45 Suppl 1:91–103.

Roth, V., Laub, J., Buhmann, J. M., and Müller, K.-R. (2002). Going metric: Denoising pairwise data. In *NIPS*, pages 817–824.

Saldanha, A. J., Brauer, M. J., and Botstein, D. (2004). Nutritional homeostasis in batch and steady-state culture of yeast. *Mol. Biol. Cell*, 15(9):4089–4104.

Saul, L. K., Weinberger, K. Q., Ham, J. H., Sha, F., and Lee, D. D. (2006). Spectral methods for dimensionality reduction. *Semisupervised Learning. MIT Press: Cambridge, MA*.

Schaeffer, S. E. (2007). Graph clustering. *Computer Science Review*, 1(1):27–64.

Schäfer, J. and Strimmer, K. (2005). A shrinkage approach to large-scale covariance matrix estimation and implications for functional genomics. *Statistical Applications in Genetics and Molecular Biology*, 4(1):art32.

Schulze, A. and Downward, J. (2001). Navigating gene expression using microarrays–a technology review. *Nat Cell Biol*, 3(8).

Schwikowski, B., Uetz, P., and Fields, S. (2000). A network of protein-protein interactions in yeast. *Nat. Biotechnol.*, 18:1257–1261.

Schlkopf, B., Tsuda, K., and Vert, J.-P., editors (2004). *Kernel Methods in Computational Biology*. MIT Press, The MIT Press, Cambridge, Massachussetts.

Segal, E., Fondufe-Mittendorf, Y., Chen, L., Thåström, A., Field, Y., Moore, I. K., Wang, J.-P. Z., and Widom, J. (2006). A genomic code for nucleosome positioning. *Nature*, 442(7104):772–778.

Segal, E., Friedman, N., Koller, D., and Regev, A. (2004). A module map showing conditional activity of expression modules in cancer. *Nat Genet*, 36(10):1090–8.

Segal, E., Pe'er, D., Regev, A., Koller, D., and Friedman, N. (2005). Learning module networks. *Journal of Machine Learning Research*, 6(4):557–588.

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

Shannon, C. E. (1948). A mathematical theory of communication. *Bell System Technical Journal*, 27:379–423, 623–656.

Shawe-Taylor, J. and Cristianini, N. (2004). *Kernel Methods for Pattern Analysis*. Cambridge University Press.

Sherlock, G. and Hernandez-Boussard, T. (2001). The stanford microarray database.

Shi, J. and Malik, J. (2000). Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905.

Smyth, G. K., Yang, Y., and Speed, T. P. (2003). Statistical issues in cdna microarray data analysis. *Methods in Molecular Biology*, pages 111–136.

Speer, N., Frlich, H., Spieth, C., and Zell, A. (2005). Functional grouping of genes using spectral clustering and gene ontology. In *Proceedings of the IEEE International Joint Conference on Neural Networks*, pages 298–303. IEEE Press.

Spellman, P., Sherlock, G., Zhang, M., Iyer, V., Anders, K., Eisen, M., Brown, P., Botstein, D., and Futcher, B. (1998). Comprehensive identification of cell cycle-regulated genes of the yeast Saccharomyces cerevisiae by microarray hybridization. *Mol Biol Cell*, 9(12):3273–97.

Sun, L., Ji, S., and Ye, J. (2008). Adaptive diffusion kernel learning from biological networks for protein function prediction. *BMC Bioinformatics*, 9:162+.

Tai, F. and Pan, W. (2007). Incorporating prior knowledge of gene functional groups into regularized discriminant analysis of microarray data. *Bioinformatics*, 23(23):3170–3177.

Tamada, Y., Bannai, H., Imoto, S., Katayama, T., Kanehisa, M., and Miyano, S. (2005). Utilizing evolutionary information and gene expression data for estimating gene networks with bayesian network models. *J Bioinform Comput Biol*, 3(6):1295–313.

Tamada, Y., Kim, S., Bannai, H., Imoto, S., Tashiro, K., Kuhara, S., and Miyano, S. (2003). Estimating gene networks from gene expression data by combining bayesian network model with promoter element detection. *Bioinformatics*, 19(90002):227ii–236.

Tanay, A., Sharan, R., Kupiec, M., and Shamir, R. (2004). Revealing modularity and organization in the yeast molecular network by integrated analysis of highly heterogeneous genomewide data. *PNAS*, 101(9):2981–2986.

Tanay, A., Steinfeld, I., Kupiec, M., and Shamir, R. (2005). Integrative analysis of genome-wide experiments in the context of a large high-throughput data compendium. *Molecular Systems Biology*, 1(1):msb4100005–E1–msb4100005–E10.

Teixeira, M. C., Monteiro, P., Jain, P., Tenreiro, S., Fernandes, A. R., Mira, N. P., Alenquer, M., Freitas, A. T., Oliveira, A. L., and Sa-Correia, I. (2006). The yeastract database: a tool for the analysis of transcription regulatory associations in saccharomyces cerevisiae. *Nucl. Acids Res.*, 34(1):D446–451.

Thomaz, C., Gillies, D., and Feitosa, R. (2004). A new covariance estimate for bayesian classifiers in biometric recognition. *Circuits and Systems for Video Technology, IEEE Transactions on*, 14(2):214–223.

Thomaz, C. E. and Gillies, D. F. (2005). A maximum uncertainty lda-based approach for limited sample size problems : With application to face recognition. In *SIBGRAPI*, pages 89–96.

Troyanskaya, O. G., Dolinski, K., Owen, A. B., Altman, R. B., and Botstein, D. (2003). A bayesian framework for combining heterogeneous data sources for gene function prediction (in saccharomyces cerevisiae). *PNAS*, 100(14):8348–8353.

Tsuda, K. (1999). Support vector classifier with asymmetric kernel functions. In *European Symposium on Artificial Neural Networks*, pages 183–188.

Tsuda, K. and Noble, W. S. (2004). Learning kernels from biological networks by maximizing entropy. *Bioinformatics*, 20(1):326–333.

Tung, A. K. H., Ng, R. T., Lakshmanan, L. V. S., and Han, J. (2001). Constraint-based clustering in large databases. In *ICDT*, pages 405–419.

Verma, D. and Meila, M. (2003). A comparison of spectral clustering algorithms. Technical report, University of Washington, Tech. Rep. UW-CSE-03-05-01.

Vert, J.-P., Thurman, R., and Noble, W. S. (2006). Kernels for gene regulatory regions. In Weiss, Y., Schölkopf, B., and Platt, J., editors, *Advances in Neural Information Processing Systems 18*, pages 1401–1408. MIT Press, Cambridge, MA.

von Luxburg, U. (2006). A tutorial on spectral clustering. Technical Report 149, Max Planck Institute for Biological Cybernetics.

Weiss, Y. (1999). Segmentation using eigenvectors: A unifying view. In *ICCV '99: Proceedings of the International Conference on Computer Vision-Volume 2*, Washington, DC, USA. IEEE Computer Society.

Werhli, A. V., Grzegorczyk, M., and Husmeier, D. (2006). Comparative evaluation of reverse engineering gene regulatory networks with relevance networks, graphical gaussian models and bayesian networks. *Bioinformatics*, 22(20):2523–2531.

Werner-Washburne, M., Wylie, B., Boyack, K., Fuge, E., Galbraith, J., Weber, J., and Davidson, G. (2002). Comparative analysis of multiple genome-scale data sets. *Genome Res*, 12(10):1564–1573.

Wingender, E., Dietze, P., Karas, H., and Knüppel, R. (1996). Transfac: a database on transcription factors and their dna binding sites. *Nucleic Acids Res*, 24(1):238–241.

Wit, E. and McClure, J. (2004). *Statistics for Microarrays: Design, Analysis and Inference*. John Wiley and Sons.

Xing, E. P., Ng, A. Y., Jordan, M. I., and Russell, S. (2003). Distance metric learning, with application to clustering with side-information. In *Advances in Neural Information Processing Systems 15*, volume 15, pages 505–512.

Yang, Y. H., Dudoit, S., Luu, P., Lin, D. M., Peng, V., Ngai, J., and Speed, T. P. (2002). Normalization for cdna microarray data: a robust composite method addressing single and multiple slide systematic variation. *Nucleic Acids Res*, 30(4).

Yeung, K., Medvedovic, M., and Bumgarner, R. (2004). From co-expression to co-regulation: how many microarray experiments do we need? *Genome Biology*, 5(7):R48.

Yu, H., Luscombe, N. M., Lu, H. X., Zhu, X., Xia, Y., Han, J. D., Bertin, N., Chung, S., Vidal, M., and Gerstein, M. (2004). Annotation transfer between genomes: protein-protein interologs and protein-dna regulogs. *Genome Res*, 14(6):1107–1118.

Zhu, J. and Zhang, M. Q. (1999). Scpd: a promoter database of the yeast saccharomyces cerevisiae. *Bioinformatics*, 15(7-8):607–611.

Zou, M. and Conzen, S. D. (2005). A new dynamic bayesian network (dbn) approach for identifying gene regulatory networks from time course microarray data. *Bioinformatics*, 21(1):71–79.

# Acronyms

**mRNA** messenger ribonucleic acid

**RNA** ribonucleic acid

**TF** transcription factor

**TRN** transcriptional regulatory network

**TRS** transcriptional regulatory system

**GRN** gene regulatory network

**DNA** deoxyribonucleic acid

**PPI** protein-protein interaction

**ChIP** chromatin immunoprecipitation

**GGM** graphical Gaussian model

**KL** Kullback-Leibler

**BN** Bayesian network

**DBN** dynamic Bayesian network

**EM** expectation maximization

**SGD** Saccharomyces Genome Database

**YPD** Yeast Proteome Database

**GRAM** Genetic Regulatory Modules

**SAMBA** Statistical-Algorithmic Method for Bicluster Analysis

**SCPD** *Saccharomyces cerevisiae* Promoter Database

**SMD** Stanford Microarray Database

**MAD** median absolute deviation

**LDA** linear discriminant analysis

**RDA** regularized discriminant analysis

**SVM** support vector machine

**SDP** semi-definite programming

**SSSC** semi-supervised spectral clustering

**KNN** $k$-nearest neighbour

**HMRF** Hidden Markov Random Field

**SOM** self-organising map