Construction the Gene Regulatory Network Using Bayesian Network and Biclustering

by

Fadhl Mohamed Ahmed Al-Akwaa

A Thesis Submitted to the

FACULTY OF ENGINEERING at CAIRO UNIVERSITY

in Partial Fulfillment of the

Requirements for the Degree of

DOCTOR of PHILOSOPHY

In

SYSTEM AND BIOMEDICAL ENGINEERING

FACULTY OF ENGINEERING, CAIRO UNIVERSITY

GIZA, EGYPT

SEPTEMBER 2009

Declaration of Authorship

I, AUTHOR NAME, declare that this thesis titled, 'THESIS TITLE' and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed: _____

Date: _____

"It is true that nothing substitutes for the collegiality and good will that arise from mutual respect for different skills and contributions"
"data mining is easy to do badly"

Isaac S. Kohane, 2003, MIT
DANIEL T. LAROSE

To ALLAH, YARB Forgive Me

Contents

List of Figures

List of Tables

GLOSSARY

- **DNA Deoxyribonucleic acid** Nucleic acid that contains the genetic instructions used in the development and functioning of allknown living organisms.
- **CDNA or Complementary DNA** DNA that is synthesized in the laboratory from messenger RNA template.
- **RNA Ribonucleic acid** Nuclic acid very similar to DNA, but differs in a few important structural details.
- **Gene** is a long strand of DNA sequence corresponding to a unit of inheritance .
- **Transcription** is the synthesis of RNA under the direction of DNA, and then this RNA sequence information translated to amino acids then to protein.
- **Translation** is the process in which the mature mRNA is translated into polypeptide chain of amino acids (then to protein) according to certain genetic codes.
- **Genomics** is the study of an organism's entire genome. The field includes intensive efforts to determine the entire DNA sequence of organisms and fine-scale genetic mapping efforts.
- **Functional Genomics** is a field that attempts to make use of the vast wealth of data produced by genomic projects (such as genome sequencing projects) to describe gene (and protein) functions and interactions.
- **Transcriptome** is the set of all messenger RNA (mRNA) molecules or transcripts produced in one or a population of cells.
- **Proteomics** is the large-scale study of proteins, particularly their structures and functions.
- **Transcription factor** is a protein that binds to specific parts of DNA using DNA binding domains and is part of the system that controls the transfer (or transcription)of genetic information from DNA to RNA.

- **Proteins** are large organic compounds made of amino acids arranged in a linear chain and joined together by peptide bonds.
- **Genetic code** is the set of rules by which information encoded in genetic material (DNA or RNA sequences)is translated into proteins.
- **Promoter** is a regulatory region of DNA generally located upstream (towards the 5′ region of the sense strand) of a gene that allows and controls transcription of the gene.
- **Gene expression** is the process by which inheritable information from a gene, such as the DNA sequence, is made into a functional gene product, such as protein or RNA.
- **Ribosomes** complexes of RNA and protein that are found in all cells. **Coding region** of a gene is the portion of DNA or RNA that is transcribed into RNA.
- **GenBank** sequence database is an open access, annotated collection of all publicly available nucleotide sequences and their protein translations.
- **Phenotype** is any observable characteristic of an organism, such as its morphology, development, biochemical or physiological properties, or behavior.
- **Eukaryotes organisms** whose cells are organized into complex structures enclosed within membranes and contain nucleus inside the cell.
- **Prokaryotes organisms** that usually lack a cell nucleus or any other membrane-bound organelles.
- **Gene Ontology (GO)** provides a controlled vocabulary to describe gene and gene product function in any organism.
- **Gene regulatory network** is a collection of DNA segments in a cell which interact with each other (indirectly through their RNA and protein expression products) and with other substances in the cell, thereby governing the rates at which genes in the network are transcribed into mRNA.

ACKNOWLEDGMENT

First, of all I would like to thank ALLAH who support me and supplied me with power and faithful to comprehensive with this hot topic.
Second, this work could not comes to light without Dr Yasser Kahdah guidance. I feel very priviliged to have worked with Dr Yasser, a brilliant scientist; I learned much from our many insightful discussions. Now I know why the famous person Said "If I would not be Egyptian I hope to be an Egyptian" I hope to be like Dr Yasser some day.
My gratitude and deep appreciation to Dr Nahed, for her generous advice and motivation to take important course on bioinformatics. I pray to ALLAH to put these in her Hasanat.
Also the support of my family specially my mother, wife, sisters, brothers and my honey kids. In addition, I could not forgot the scientific support of kindly professors at the System and Biomedical Engineering Department; Cairo University especially Prof Abdullah and Prof Emad Rasmy.
Finally, I thank lab at Stanford for making microarray data available, and the lab members for the courteous help they gave me. Also, I think Dr Kevin Yip, Yale university and Prof G.Stolovitzky, IBM Computational Biology Center for their respectful discussion and generous time.
I present this work to the University of Science and Technology in Yemen. I wish that I contribute some thing valuable to this community, if there is some thing wrong this is from my self and devil and if there are some thing wright it is from ALLAH.

ABSTRACT

Understanding gene interactions in complex living systems can be seen as the ultimate goal of the system biology revolution. Hence, to fully understand disease ontology and to reduce the cost of drug development we need to construct the gene regulatory network (GRN). During the last decade, many GRN inference algorithms that are base on genome-wide data have been developed to unravel the complexity of gene regulation. Data dimensionality and variability are important problems in GRN modelling.

We propose an integrated algorithm (SSBBN) for denoising using Spectral Subtraction(SS), reducing the dimension using Biclustering(B) and Bayesian Network (BN) learning to overcome these problems. Firstly, the microarray dataset is denoised using our spectral subtraction novel method to decrease the false positive rate. Secondly, we divide the whole set of genes into a number of overlapped biclusters using our proposed BicAT-Plus toolbox. Thirdly, these biclusters are learned using Greedy Hill Climbing search algorithm to produce small subnetworks. Finally, these subnetworks were integrated to produce the whole gene regulatory network. The proposed method was applied to time series gene expression data of *Saccharomyces Cerevisiae*. The generated network was validated via available interaction databases and the result revealed the performance of our proposed method. Also, The generated network from our proposed method outperformed the network generated from previous methods. The approach could potentially be applied to other networks in yeast as well as higher organisms.

BicAT-Plus can be downloaded from `http://home.k-space.org/BicAT-plus.zip`

Chapter 1

Introduction

1.1 Thesis Overview

One of the hot topics in the area of bioinformatics is functional genomics. This topic focuses on the interactions and functions of each gene and its products (mRNA, protein) through the whole genome. In order to identify the functions of certain gene, we should be able to capture the associated gene expressions. Gene expression describes how the genetic information converted into a functional gene product through the transcription and translation processes. Functional genomics uses microarray technology to measure the genes expressions levels under certain conditions and environmental limitations. Microarray has become a central tool in biological research. Consequently, the analysis of gene expression data is a necessary and important tool for studying regulatory and other functional relationships among genes. The identification of gene regulatory networks (GRN) is of major importance in order to understand the working mechanisms of the cell in patho-physiological conditions.

Recently, and exactly this year a quartet of studies by researchers at the California Institute of Technology (Caltech) highlight a special feature on gene regulatory networks to understand how development of an animal occurs [1]. Also researchers at Institute for Cancer Genetics Columbia University study complex diseases and design novel therapies using reverse engineering approach [2].

In this dissertation we address the challenge of reconstructing gene regulation network from gene expression data.

Within the last few years, a number of sophisticated approaches for the reverse engineering of cellular networks from gene expression data have been emerged. This may include Boolean networks [3], Bayesian networks [4], association networks [5], linear

models [6], and differential equations [7].

The great challenges in GRN modeling are dimensionality reduction and denoising of microarray data. Efforts are being done to overcome these problems. Dimensionality reduction was tried many times through clustering algorithms. Clustering algorithms [8–10] were used to reduce data dimensionality. This is based on the assumption that genes which show similar expression patterns, are co-regulated or part of the same regulatory pathway. But unfortunately, this is not always true. By learning genes within each cluster, we get one subnetwork for each cluster. Integrating cluster subnetworks, we generate the whole GRN.

The problem using clustering is that clustering does not guarantee that genes within a cluster share the same biological function. This is because clustering algorithms are based on the similarity matrix, which is calculated using all data experiments. Recent understanding of the cellular process suggested that some genes should have similar expression under certain experiments and they differ under the other experiments [11]. A bicluster technique to group similar genes under appropriate experiments is required. During the last year, more than ten biclustering algorithms have been proposed, but the question is: which algorithm is better? And do some algorithms have advantages over others? Generally, comparing different biclustering algorithms is not straightforward as they differ in strategies, approaches, time complexity, number of parameters and prediction ability. They are strongly influenced by user-selected parameter values. For these reasons, the quality of biclustering results is also often considered more important than the required computation time. Although there are some analytical comparative studies to evaluate the traditional clustering algorithms [12–14], for biclustering, no such extensive comparison exist even after initial trails have been taken [15]. At the end, biological merit is the main criterion for evaluation and comparison between the various biclustering methods.

To our best knowledge, biclustering algorithms compassion toolbox has not been available in the literature. We have developed a comparative tool BicAT-Plus [16] that includes the biological comparative methodology and to be used as an extension to the BicAT program [17]. BicAT [17] is a common biclustering analysis toolbox in which most important bi/clustering algorithms like k-means, HCL [18], Bimax [15] , OPSM [19], X-motif [20],CC [11], and ISA [21, 22] were implemented. In this work one of our goals is to study the impact of using biclustering algorithms in GRN construction.

Bonneau et al [23] developed GRN algorithm (The Inferelator) based on an integrated biclustering method (cMonkey) [24]. cMonkey groups genes and conditions into biclusters on the basis of three components: the expression component, the sequence

component, and the network component, which they are not available for all the organisms. Because all the biclustering algorithms which are either implemented in BicAT or in our modified version BicAT-Plus, did not require any prior information, we excluded cMonkey from further analysis.

The problem of data denoising has also been studied. Many algorithms are found in the literature for data denoising. Among the most powerful techniques that can be used to separate signal components are those based on blind source separation such as principal component analysis (PCA) and independent component analysis (ICA) [25]. These techniques decompose the signal sources using either the second order statistics (as in PCA) or higher order statistics such as the kurtosis (as in ICA) to account for the non-Gaussian nature of the sources. According to the assumptions of both techniques, the number of independent signal components must be less than or equal to the number of signals to be analyzed. Otherwise, the separation of components yields incorrect results or even may not converge at all as in ICA. Unfortunately, this condition is not satisfied in microarray datasets. Given the general assumption of uncorrelated noise, the number of components of random noise alone is equal to the number of signals. The total number of components has to add the number of components. As a result, the use of PCA and ICA-based techniques may not be successful in practice unless the noise signals are sufficiently weak. This may account for the limited use of such techniques in low SNR applications [26]. Therefore, a technique that suppresses random noise or removes some of its components would be rather useful for making the use of PCA and ICA more robust for false positive reduction.

In this thesis, we provide an integrated new technique for denoising and dimensionality reduction of microarray data. Our proposed algorithm provides a deeper understanding of the biological systems complexity. We applied our algorithm to two well-known datasets of yeast microarray gene expression (Gasch et al [27]; Spellman et al [28]), which can downloaded from Stanford Microarray Database(http://smd.stanford.edu/) Figure 1.1 shows our proposed GRN modeling block digram.

The first step start with the question biologist required to answer for example which genes are involved in controlling the cell cycle [28] and which genes are showed a similar drastic response to the environmental changes [27]. The above questions are important biologically and clinically, and with out assistance of the bioinformatics expert who have an appropraite tools that help to answer these questions. The next step, is to prepare an appropriate experiments relative to biological question. Next is the extraction of the gene expression matrix from the microarray experiments using image processing techniques. Removing of non-informative genes and conditions, normalization and denoising data using our novel spectral subtraction (SS) method are described in the preprocessing step(Chapter 3). We chose a filtering procedure that rejects many unreliable and uninformative data points. The next step is the partitioning of all the whole

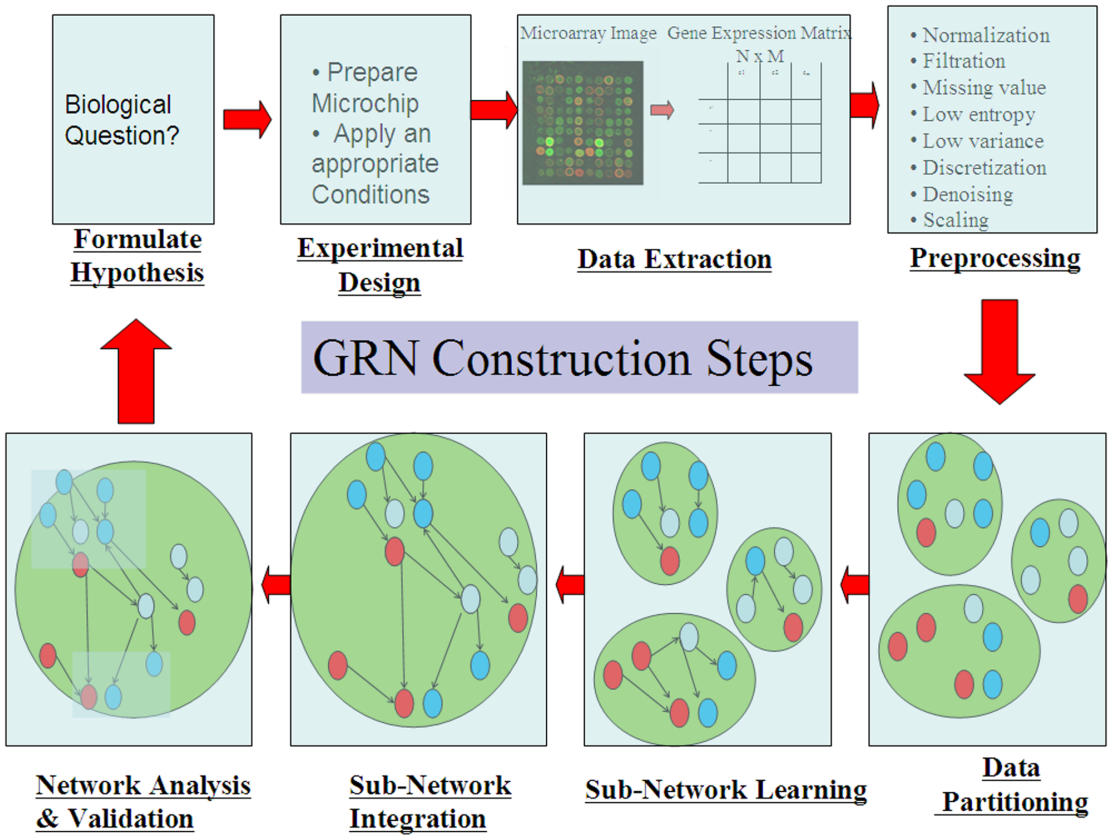


FIGURE 1.1: Basic Steps involved in GRN construction. Formulate Hypothesis:start with biological question?; Experimental Design: prepare relative experiments; Data Extraction:; Data Preprocessing:to remove non informative genes and reduce signal variability;Data Partitioning:to overcome data dimensionality problem; Subnetwork Learning:to construct the submodule network; Subnetwork Integration:to construct the whole network; Network Analysis and Validation: to test the produced network via literature.

genes in to small overlapped biclusters using our modified biclustering toolbox(BicAT-Plus) [16](Chapter 3). After that, we learn each of these biclusters using Bayesian network structure learning algorithm (Greedy Hill Climbing) to produce overlapped subnetworks and integrate them to produce the whole network(Chapter 5). Assessment the performance of the resultant network using existing interactome databases is illustrated in the vaildation step(Chapter 3). Last step, the generated validated network will open new hypotheses which need to be verified by further experiments.
It could not move without mention important issues have to be considered:

- First, the above steps are not just forward steps but it is forward-backward.If the modeler is not satisfied with results he could change the parameters to get more closer results, depends on his first hypothetical assumption.

- Second the noise in these steps is accumulated, means that the poor microarray data, alter preprocessing steps which further alter the cluster results that will change the learning network and the following validation process.
- Finally, a successful GRN construction cannot be performed by a single group or laboratory, it required resources from many disciplines and of varied backgrounds.

1.2 Thesis Objective

Due to the increasing the application of the gene regulatory network to fully understand disease ontology and to reduce the cost of drug development.
We are attempt in this thesis, to present an integrated technique that can be applied to the gene expression matrices to construct a more reliable gene regulatory network. This technique uses a new method for denoising based on spectral subtraction (SS). Moreover, we extended the available biclustering algorithm known as BicAT and provide a new one called BicAT-Plus. BicAT-Plus could be used efficiently for the sake of dimensionality reduction.
The confidence in the results obtained from our proposed algorithm, will be validated and analyzed via available previous work and litterer.

1.3 Thesis Organization

In Chapter 2 introductory to biology and description of different data source like the gene expression data which are used in this thesis are to be introduced. Chapter 3 covers gene expression analysis starting from normilazation, discretization, denoising, clustering and biclustering. This chapter also include our biclustering comparison toolbox AGO and BicAT-Plus which were published in [29] and [16]. A brief literature review about different reverse engineering approaches which were developed during last ten years were introduced in Chapter 4. It was illustrated why last GRN current research used Bayesian Network learning algorithms. Assessment the performance of the our generated network and network analyzing and vaildation via previous methods and literature are presented in Chapter 5.
Finally Chapter 6 presents the our conclusion and possible improvements in the research as well as identifying future related research areas.

Chapter 2

Biological Background

In this chapter we will start with brief introductory to bioinformatics(Section 2.1), its impact on health life and current hot research area. Second, we summarized the basic elements of biology(Section 2.2)like transcription and translation. Third, we give a short description of our model organism "*Saccharomyces Cerevisiae*"(Section 2.5) and why we select yeast(Section 2.5.1) in this study?. Finally, we describe the important biological data sources(Section 2.4) which were used in this study. We give more details about the microarrays technology(Section 2.3.2) or the gene expression matrix as it is the main data source in this study.

2.1 Introduction to Bioinformatics

It is interesting to note that there is no one single definition of bioinformatics. Different organizations define it in their own way. A more simpler definition of bioinformatics is that it is the application of computer technology to the management and analysis of biological data. It is an interdisciplinary research area that is the interface between the biological and computational sciences it's ultimate goal being to uncover the wealth of biological information hidden in the mass of data and to obtain a clearer insight into the fundamental biology of organisms [30].Simply Bioinformatics is the marriage between biology and information technology. Bioinformatics concerns the development of new tools for the analysis of genomic and molecular biological data including sequence analysis ,genetic algorithms, phylogenetic inference, genome database organization and mining, optical computation and holographic memory, pattern recognition and image analysis, biologically inspired computational models [31].

2.1.1 Historical Development

The beginning of bioinformatics can be traced back to Margaret Dayhoff in 1968 and her collection of protein sequences known as the Atlas of Protein Sequence and Structure [32]. One of the early significant experiments in bioinformatics was the application of a sequence similarity searching program to the identification of the origins of a viral gene [33]. In this study, scientists used one of the first sequence similarity searching computer programs (called FASTP), to determine that the contents of v-sis, a cancer-causing viral sequence, were most similar to the well-characterized cellular PDGF gene. This surprising result provided important mechanistic insights for biologists working on how this viral sequence causes cancer [34]. From this first initial application of computers to biology, the field of bioinformatics has exploded. The growth of bioinformatics is parallel to the development of DNA sequencing technology. In the same way that the development of the microscope in the late 1600's revolutionized biological sciences by allowing Anton Van Leeuwenhoek to look at cells for the first time, DNA sequencing technology has revolutionized the field of bioinformatics. The rapid growth of bioinformatics can be illustrated by the growth of DNA sequences contained in the public repository of nucleotide sequences called GenBank.

2.1.2 The Need for Bioinformatics

The word bioinformatics has become a very popular "buzz" word in science. Many scientists find bioinformatics exciting because it holds the potential to dive into a whole new world of uncharted territory. Bioinformatics is a new science and a new way of thinking that could potentially lead to many relevant biological discoveries. Although technology enables bioinformatics, bioinformatics is still very much about biology. Biological questions drive all bioinformatics experiments. Important biological questions can be addressed by bioinformatics and include understanding the genotype-phenotype connection for human disease, understanding structure to function relationships for proteins, and understanding biological networks. Bioinformaticians often find that the reagents necessary to answer these interesting biological questions do not exist. Thus, a large part of a bioinformatician's job is building tools and technologies as part of the process of asking the question. For many, bioinformatics is very popular because scientists can apply both their biology and computer skills to developing reagents for bioinformatics research. Many scientists are finding that bioinformatics is an exciting new territory of scientific questioning with great potential to benefit human health and society.

2.1.3 Bioinformatics Impact on Health Life

The birth of bioinformatics as a result of the explosion of raw data after the completion of the Human Genome Project has added another dimension to the drug discovery process. The pharmaceutical industry has all along operated without bringing together the disciplines of biology , chemistry and information technology [35]. These fields ,though complimentary had no common interface. The pharmaceutical industry appears to have been left behind when other industries were implementing information technology to improve their operations. But due to the genome project and the resultant data explosion, it was then imperative to join these fields of science together to exploit the available data and thus expedite the drug discovery process. Traditionally , the drug discovery process takes an average of 15 years and costs about $880 million to develop each new medicine that does make it to the market. Nearly 75% of drug candidates currently being tested by pharmaceutical companies will fall short of expectations and never reach the market [36]. Added to this is the recent negative perception of the pharmaceutical industry due to the ever spiraling drug prices, recalls and recent warnings about popular prescription medications. In an attempt to improve and reduce the cost of drug discovery, the pharmaceutical industry has recently turned to bioinformatics. Some analysts predict that bioinformatics could help cut in half the cost of creating a drug and shave two to three years off its development [36].

2.1.4 Bioinformatics Research Area

Figure 2.1 shows a scheme of the main biological problems where Bioinformatics methods are being applied. These applications could be classified into six different domains:

1. Genomics
    - Extract the location and structure of the genes
    - Identification of Regulatory Elements and Non-coding RNA Genes
    - RNA Secondary Structure Prediction
2. Proteomics
    - Protein Structure Prediction.
3. Microarrays
    - Pre-processed

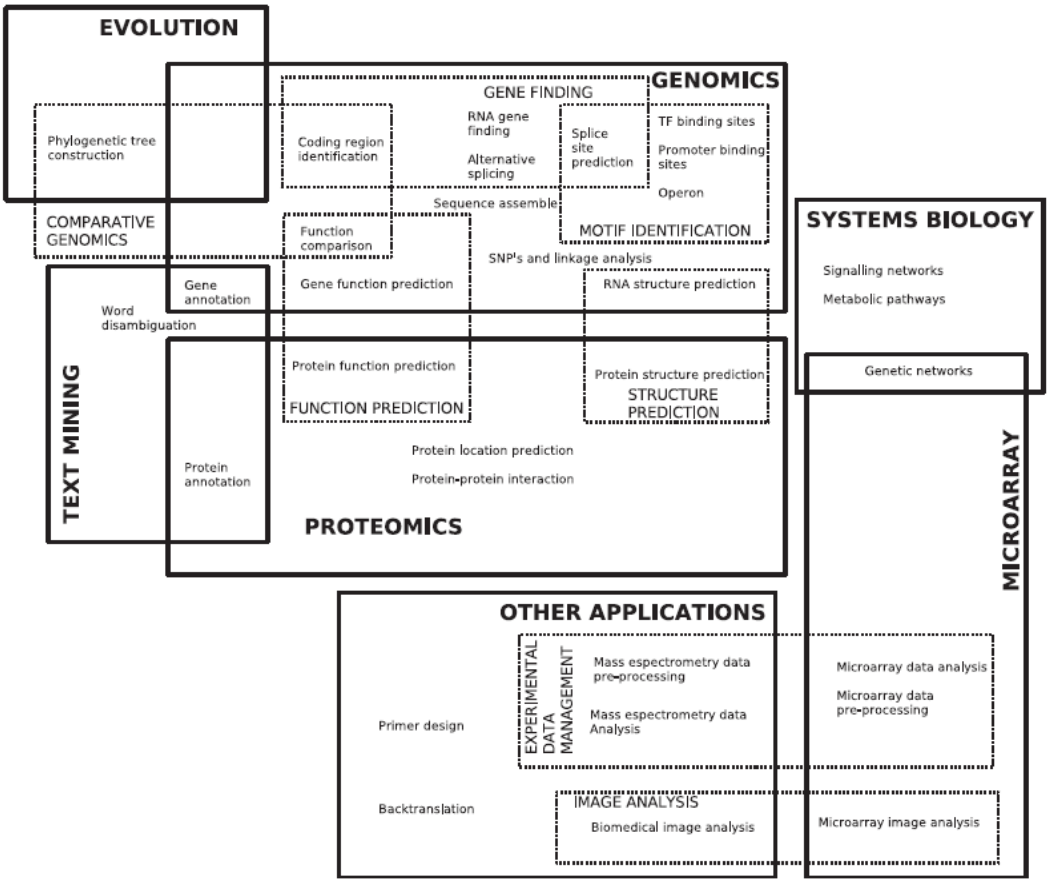


FIGURE 2.1: Bioinformatics Current Research Area (Copyright © [37]).

- Expression Pattern Identification
- Classification
- Genetic Network Induction.

4. Systems Biology

   - Modelling Biological Networks

     Genetic Networks

     Signal Transduction Networks

     Metabolic Pathways

5. Evolution

   - Phylogenetic Tree Reconstruction
   - Multiple Sequence Alignment

6. Text Mining

   - Knowledge Extraction

- Functional Annotation,
- Cellular Location Prediction
- Protein Interaction Analysis

2.2 Basic Biology

2.2.1 Prokaryotic and Eukaryotic Cell Types

All organisms consist of small cells, typically too small to be seen by a naked eye, but big enough for an optical microscope [38].There are estimated about $6x10^{13}$ cells in a human body, of about 320 different types. For instance there are several types of skin cells, muscle cells, brain cells (neurons), among many others. The world of organisms could be divided into two types: Prokaryotic and Eukaryotic cells

- Prokaryotic Cells: Prokaryotic Cells are smaller than eukaryotic cells (See Figure 2.2)(a typical size of a prokaryotic cell is about 1 micron in diameter) and have simpler structure (e.g., they do not have any inner cellular membranes that are always present in Eukaryotes, see below). Prokaryotes are single cellular organisms, but note that being a single cell does not mean that an organism is a prokaryote. Being smaller than eukaryotes does not mean that prokaryotes are any less important. For instance it is quite likely that the number of bacteria living in the mouth and digestive tract of a human are larger than the number of eukaryotic cells in the same individual and many of these bacteria are necessary for a human being to live a normal life (these numbers are rather difficult to estimate, rather a hypothesis). Prokaryotes are sometimes also known as microbes.
- Eukaryotic Cells: A Eukaryotic cell has a nucleus, which is separated from the rest of the cell by a membrane. The nucleus contains chromosomes, which are the carrier of the genetic material. There are internal membrane enclosed compartments within eukaryotic cells, called organelles, e.g., centrioles, lysosomes, golgi complexes, mitochondria among others 2.2, which are specialised for particular biological processes. The mitochondria are found in all eukaryotes and are specialised for energy production (respiration). Chloroplasts are organelles found in plant cells which produce sugar using light. Light is the ultimate source of energy for almost all life on Earth. The area of the cell outside the nucleus and the organelles is called the cytoplasm. Membranes are complex structures and they

are an effective barrier to the environment, and regulate the flow of food, energy and information in and out of the cell.

An essential feature of most (prokaryote and eukaryote) living cells is their ability to grow in an appropriate environment and to undergo cell division. The growth of a single cell and its subsequent division is called the cell cycle. However, not all cells continually grow and divide, for example neurons only undergo an initial growth phase. Prokaryotes, particularly bacteria, are extremely successful at multiplying - it is likely that natural selection has favoured single celled organisms able to grow and divide quickly. Multicellular organisms typically begin life as a single cell, usually as a result of fusion of a male and a female sex cell (gametes). The single cell has to grow, divide and differentiate into different cell types to produce tissues and in higher eukarotyes, organs. Cell division and differentiation need to be controlled. Cancerous cells grow without control and can go on to form tumours. Development of single cells into complex organisms is in itself an area of study called developmental biology.

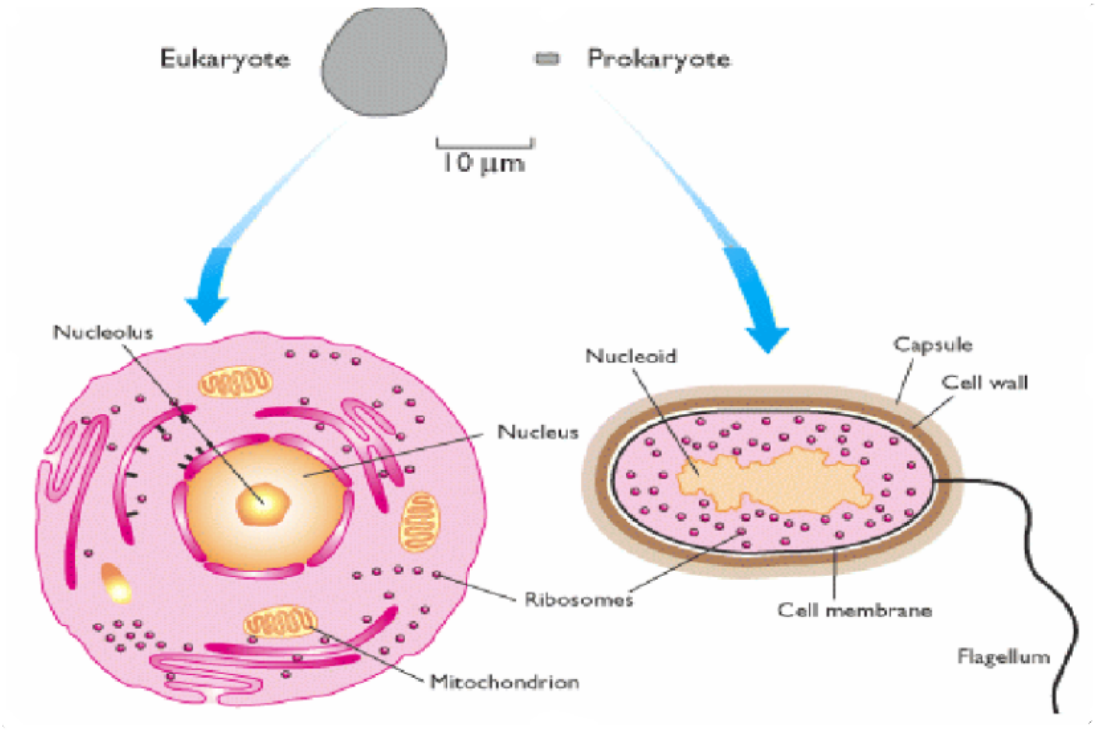


FIGURE 2.2: Model of a Eukaryotic and Prokaryotic Cell.

2.2.2 Molecules of Life

There are four basic types of molecules involved in life: (1) small molecules, (2) proteins, (3) DNA and (4) RNA. Proteins, DNA and RNA are known collectively as biological

macromolecules.

2.2.2.1 Small Molecules

These can be the building blocks of the macromolecules or they can have independent roles, such as signal transmission or being a source of energy or material for a cell. Some important examples besides water are sugars, fatty acids, amino acids and nucleotides. For instance, biological membranes are constructed from fatty acids, into which macromolecules are embedded. There are 20 different amino acid molecules, which are the building blocks for proteins (to be more precise, there are 19 amino acids and one which has a slightly different structure and therefore is called amino acid). Figure 2.3 shows three examples of amino acid moleclues, there are 17 more. They differ by R side chains which determine their properties and the order of these different amino acids within the protein determines the three dimensional structure of the protein . There is a convention that each amino-acid is denoted by a letter in Latin alphabet, for instance arginine is denoted by R, histidine by H, lysine by L and there are 20 such letters.

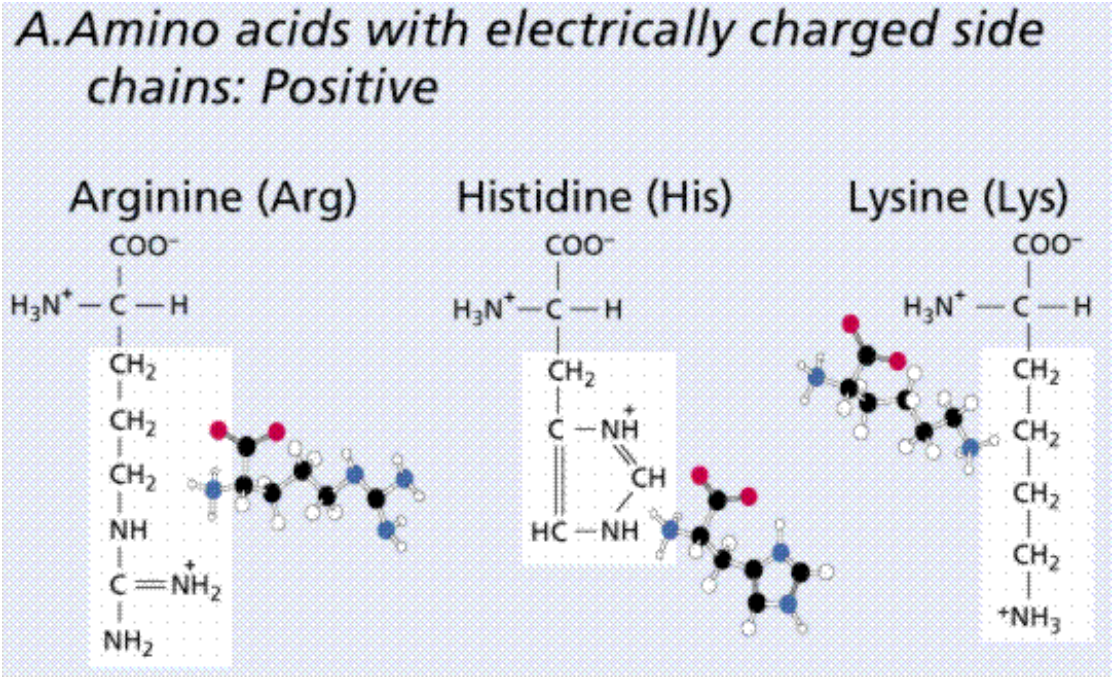


FIGURE 2.3: Amino Acid Structure.

2.2.2.2 Proteins

Proteins are the main building blocks and functional molecules of the cell, taking up almost 20% of a eukaryotic cell's weight, the largest contribution after water (70%). Among others, there are:

- Structural Proteins, which can be thought of as the organism's basic building blocks. An example is collagen, which is the major structural protein of connective tissue and bone.
- Enzymes, which perform (catalyse) a multitude of biochemical reactions, such as altering, joining together or chopping up other molecules. Together these reactions and the pathways they make up is called metabolism. For example the first step in the glycolysis pathway, which is the conversion of glucose to glucose 6-phosphate, is catalysed by the enzyme hexokinase. Usually enzymes are very specific and catalyse only a single type of reaction, however the same enzyme can play role in more than one pathway.
- Transmembrane Proteins are key in maintenance of the cellular environment, regulating cell volume, extraction and concentration of small molceules from the extracellular environment and generation of ionic gradients essential for muscle and nerve cell function. An example is the sodium/potassium pump.

Proteins have complex three dimensional (3D) structure (see Figure 2.4).PDbe [1] is a database of known protein structures, which is housed and developed at the EBI. The images below shows the structure of triosephosphate isomerase visualised by RasMol software package, a 3D viewer for PDBe structures.
Proteins are much too small to be seen in an optical microscope - a characteristic protein size varies from about 3 to 10 nanometers (nm), i.e., 3 to 10 times 10-9 m, and solving (i.e., discovering) their structure is a difficult and expensive exercise (approximately €50,000 - €200,000 per novel structure), which is done by a variety of methods including X-ray crystallography, nuclar-magnetic resonance spectroscopy, and advanced electron microscopy. There are roughly 15,000 protein structures deposited in public databases, though many of them are very similar to each other. Whether to consider two protein structures similar or different depends on the similarity threshold (as with cell types). Structural biologists think that currently there are about 1,500 different representative protein structures known.
Predicting protein structure from the amino-acid sequence is one of the most important problems of computational biology (another name for bioinformatics, though some try

[1] http://www.ebi.ac.uk/pdbe/

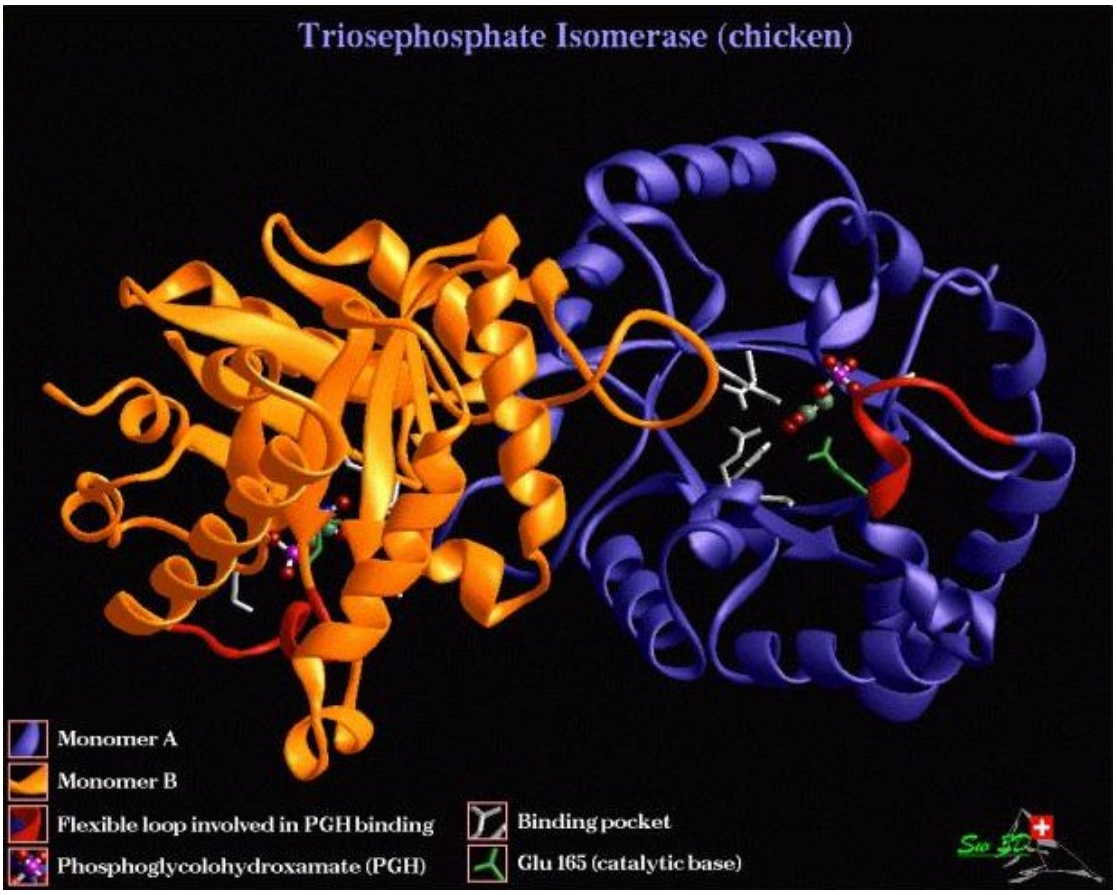


FIGURE 2.4: 3D Structure of Triosephosphate Isomerase Visualised by RasMol Software Package.

to make a distinction between these two terms) and is far from being solved. Characteristic, frequently reoccurring structural elements are called protein domains. Sometimes it is possible to identify these domains in proteins of unknown structure, if their sequence is similar to that of a known structural domain. Structural domains are often associated with a particular protein function. Protein similarity is also deemed to be the result of evolutionary relationship.

2.2.2.3 DNA

DNA is the main information carrier molecule in a cell. DNA may be single or double stranded. A single stranded DNA molecule, also called a polynucleotide, is a chain of small molecules, called nucleotides . There are four different nucleotides grouped into two types, purines: adenosine and guanine and pyrimidines: cytosine and thymine. They are usually referred to as bases (in fact bases are the only distinguishing element between different nucleotides, see Figure 2.5 and denoted by their initial letters, A,C ,G and T (not to be confused with amino acids!).

Different nucleotides can be linked together in any order to form a polynucleotide,

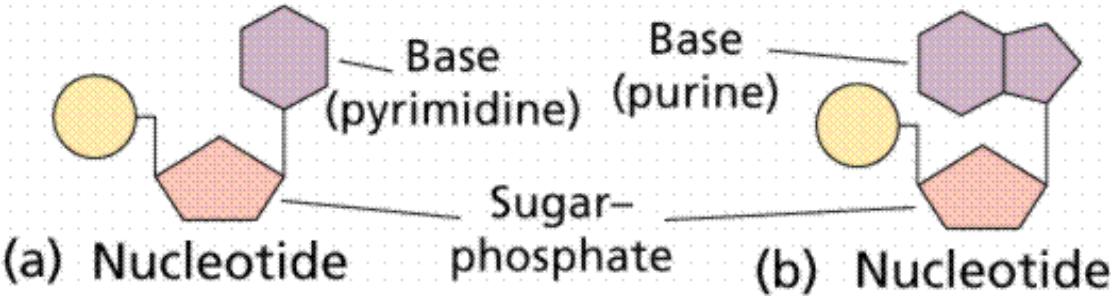


FIGURE 2.5: **DNA Basic Structure**: Nucleotide is the Basic Element of DNA, Picture Taken from On-Line Biology Book [2].

for instance, like this A-G-T-C-C-A-A-G-C-T-T. Polynucleotides can be of any length and can have any sequence. The two ends of this molecule are chemically different, i.e., the sequence has a directionality, like this $A-\rangle G-\rangle T-\rangle C-\rangle C-\rangle A-\rangle A-\rangle G-\rangle C-\rangle T-\rangle T-\rangle$. The end of the polynucleotide are marked either 5′ and 3′ (this has chemical reasons in the numbering of the -OH groups of the sugar ring); by convention DNA is usually written with 5′ left and 3′ right, with the coding strand at top. Two such strands are termed complementary , if one can be obtained from the other by mutually exchanging A with T and C with G, and changing the direction of the molecule to the opposite. For instance, $T-\langle C-\langle A-\langle G-\langle G-\langle T-\langle T-\langle C-\langle G-\langle A-\langle A-\langle$ is complementary to the polynucleotide given above. Specific pairs of nucleotides can form weak bonds between them. A binds to T, C binds to G (to be more precise, two hydrogen bonds can be formed between each A-T pair, and three hydrogen bonds between each C-G pair). Although such interactions are individually weak, when two longer complementary polynucleotide chains meet, they tend to stick together, like this:

$$\begin{array}{cccccccccccc} A-\rangle & G-\rangle & T-\rangle & C-\rangle & C-\rangle & A-\rangle & A-\rangle & G-\rangle & C-\rangle & T-\rangle & T-\rangle \\ | & | & | & | & | & | & | & | & | & | & | \\ T-\langle & C-\langle & A-\langle & G-\langle & G-\langle & T-\langle & T-\langle & C-\langle & G-\langle & A-\langle & A-\langle \end{array}$$

Vertical lines between two strands represent the forces between them (to be more accurate we could draw triple lines between each C and G and double lines between A and T). The A-T and G-C pairs are called base-pairs (bp). The length of a DNA molecule is usually measured in base-pairs or nucleotides (nt), which in this context is the same thing. Two complementary polynucleotide chains form a stable structure, which resembles a helix and is known as a the DNA double helix (Figure 2.6). About

[2] http://www.estrellamountain.edu/faculty/farabee/biobk/biobooktoc.html

10 bp in this structure takes a full turn, which is about 3.4 nm long. This structure was

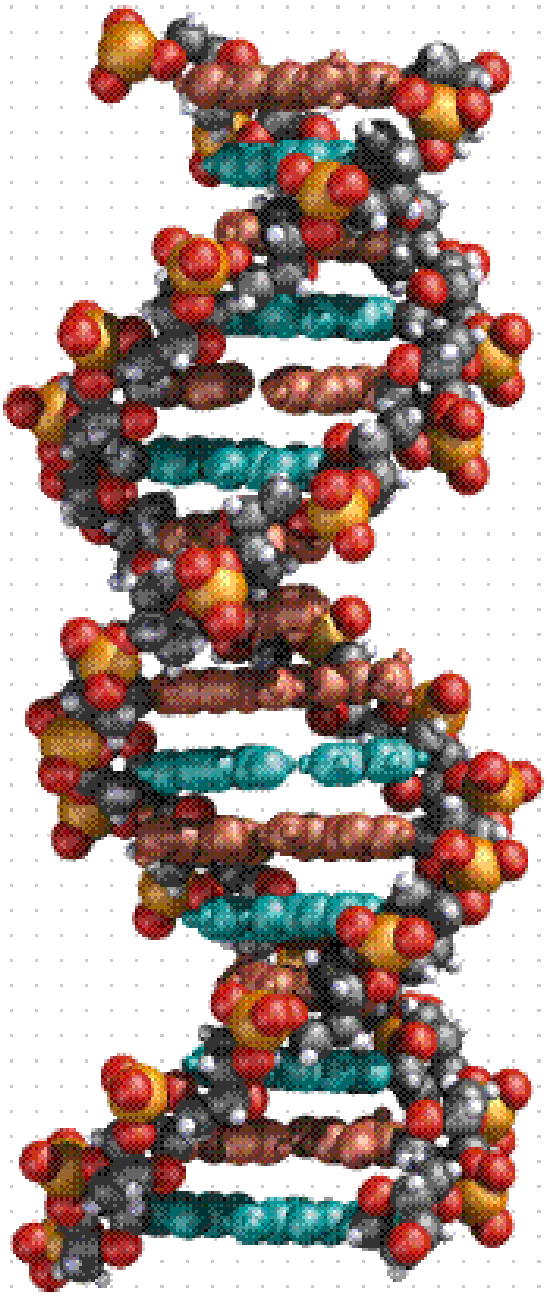


FIGURE 2.6: **DNA Double Helix Model**: DNA Helix Structure was First Figured in 1953 by Watson and Crick where Later they got the Nobel Prize for this discovery, Picture Taken from On-Line Biology Book [3].

first figured out in 1953 in Cambridge by Watson and Crick (with the help of others), Later they got the Nobel Prize for this discovery, for more see the book by Watson - The Double Helix.

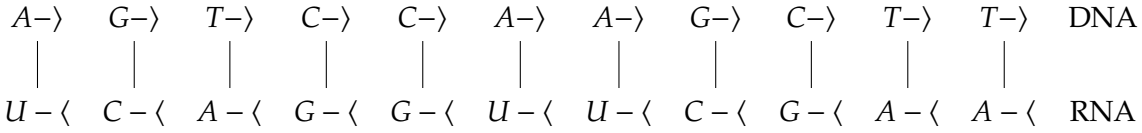
It is remarkable that two complementary DNA polypeptides form a stable double helix

[3] http://www.estrellamountain.edu/faculty/farabee/biobk/biobooktoc.html

almost regardless of the sequence of the nucleotides. This makes the DNA molecule a perfect medium for information storage. Note that as the strands are complementary, each one of them fully determining the other, therefore for the information purposes it is enough to give only one strand of the genome molecules. Thus, for many information related purposes, the molecule used on the example above, can be represented as CGATTCAACGATGC. The maximal amount of information that can be encoded in such a molecule is therefore 2 bits times the length of the sequence. Noting that the distance between nucleotide pairs in a DNA is about 0.34 nm, we can calculate that the linear information storage density in DNA is about 6x10 8 bits/cm, which is approximately 75 GB or 12.5 CD-ROMs per cm.

2.2.2.4 RNA

RNA like DNA is constructed from nucleotides. But instead of the pyrimidine thymine (T), it has an alternative uracil (U), which is not found in DNA. Because of this minor difference RNA do not form a double helix, instead usually they are single stranded, but may have complex spatial structure due to complementary links between the parts of the same strand as for instance in tRNA. RNA can bind complementary to a single strand of a DNA molecule, even though T is replaced by U, so molecules like this:



Since the discovery of DNA and RNA in the 1950s, scientists have studied the function and structure of the components that makeup these structures. The various types and functions of RNA have been investigated by numerous researchers, including Spanish physiologist Severo Ochoa (19051993), who received a Nobel prize in 1959 for his contributions to our understanding of how RNA is synthesized[4].

There are five major types of RNA that are found in the cells of eukaryotes. These include heterogeneous nuclear RNA (hnRNA), messenger RNA (mRNA), transfer RNA tRNA), ribosomal RNA (rRNA), and small nuclear RNA. Structurally, hnRNA and mRNA are both single stranded, while rRNA and tRNA form threedimensional molecular configurations. Each type of RNA has a different role in various cellular processes. In addition to these functions, RNA plays an important role in the ability of certain viruses to cause infection.

[4]http://science.jrank.org/pages/5892/RNA-Function.html

One of the primary functions of RNA is to facilitate the translation of DNA into protein. This process begins in the nucleus of the cell with a series of enzymatic reactions that transcribe DNA into heterogeneous nuclear RNA by complementary base pairing.The mRNA attaches to the ribosome to allow for the initiation of protein synthesis. Part of this process involves another type of RNA that is located in the ribosome called tRNA. tRNA is an adapter molecule, which functions as a bridge between a specific three-base sequence or codon in the mRNA strand and the amino acids that are used to construct the protein. The tRNA carries an amino acid that matches the specific codon and this process begins and stops based on specific sequences in the mRNA. Each amino acid is transferred to the growing polypeptide by chemical interactions to produce a full-length protein. Another type of RNA that is part of the ribosome and is involved in protein synthesis is rRNA. rRNA has two primary functions. First, it provides the structure and shape producing the catalytic regions of the ribosome. Second, it helps speed up, or catalyze, protein synthesis by interactions between the tRNA and the protein synthesis machinery.

2.2.2.5 Chromosomes and Genomes

In a typical cell there are one or several long double stranded DNA molecules organised as chromosomes. In eukaryotes chromosomes have a complex structure where DNA is wound around structural proteins called histones. A human has 23 pairs of chromosomes , which are large enough to be seen in an optical microscope. The total length of the DNA in one human cell, if we could stretch it out, would be more than 1m. Mitochondria contain DNA too, but the amount is minuscule in comparison to chromasomal DNA. Chromasomal and mitochondrial DNA forms the genome of the organism. All organisms have genomes and they are believed to encode almost all the hereditary information of the organism. In eukaryotes chromosoms are in the nucleus (apart from mitochondrial genomes), contained by the nuclear membrane. All cells in an organism contain identical genomes (with few rather special exceptions), as the result of DNA replication at each cell division. There is a molecular machinery in cells, which keeps both DNA strands intact and complementary (i.e., if one strand is damaged, it is repaired using the second as a template). This is important as DNA damage (caused by environmental factors like radiation) can result in breaks in one or both strands, or mispairing of the bases, which would disrupt DNA replication among other things. If damaged DNA is not repaired the result can be cell death or tumors. Changes in genomic DNA are known as mutations. The total genome size differ quite considerably in different organisms, as given in the Table 2.1.

TABLE 2.1: Genome Size and Number of Chromosomes of various Organism

Organism	Number of Chromosomes	Genome size in base pairs
Bacteria	1	400,000 - 10,000,000
Yeast	12	14,000,000
Worm	6	100,000,000
Fly	4	300,000,000
Weed	5	125,000,000
Human	23	3,000,000,000

2.2.3 Genes and Protein Synthesis

There are many discussions between biologists to find a comprehensive definition of a gene, which is not easy, if possible at all.

> A gene is a continuous stretch of a genomic DNA molecule, from which a complex molecular machinery can read information (encoded as a string of A, T, G, and C) and make a particular type of a protein or a few different proteins.

The above "definition" is not precise, and to better understand it we need to describe the molecular machinery making proteins based on the information encoded in genes. This process is called protein synthesis and has three essential stages: (1) transcription, (2) splicing, and (3) translation.

1. In transcription phase one strand of DNA molecule is copied into a complementary pre mRNA (pre stands for preliminary and m for messenger) by the protein complex RNA polymerase II(Figure 2.8) . In the process the two-stranded DNA double helix is unwound and information is read only from one strand (sometimes called the W-strand).

2. Splicing removes some stretches of the pre mRNA(Figure 2.7), called introns, the remaining sections called exons are then joined together. Note that the removal of introns is a consequence of the way how eukaryote genomes are organised. The genomic DNA that corresponds to the coding part of genes is not continuous, but consists of exons and introns. Exons are the part of the gene that code for proteins and they are interspersed with non coding introns which must be removed by splicing. The number and size of introns and exons differs considerably between genes and also between species. Only very few genes in yeast have introns, while for human threre are about 4 introns per gene on average, and the average size of

exons is 150 bp and just above 3400 bp for introns. Prokaryote genes do not have introns and the splicing step is not present. The result of splicing is mRNA. Many eukaryote genes are known to have different alternative splice variants, i.e. the same pre-mRNA producing different mRNAs, known as *alternative splicing*.

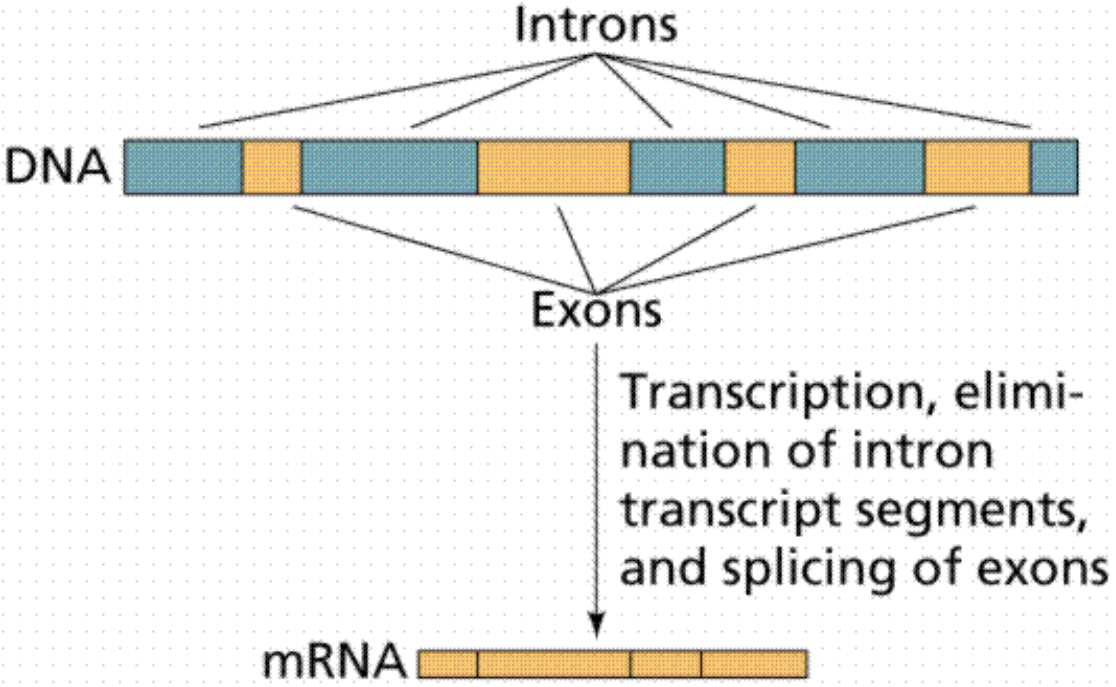


FIGURE 2.7: Splicing: Remove of Introns and Splice Exons. Picture Taken from On-Line Biology Book [5].

3. Translation is the process of making proteins by joining together amino acids in order encoded in the mRNA(Figure 2.8). The order of the amino acids is determined by 3 adjacent nucleotides (triplets) in the DNA. This is known as the triplet or genetic code . Each triplet is called a codon and codes for one amino acid. As there are 64 codons and only 20 amino acids the code is redundant, for example histidine is encoded by CAT and CAC. In cytoplasm the mRNA forms a complex with ribosomes, which are large complexes of proteins and RNA molecules. The precise interactions and functions of all protein in ribosomes are not yet fully understood. Different transfer or tRNA molecules each carries one specific amino acid to the ribosome and specifically recognises one codon on the mRNA. The amino acid carried by the tRNA is added to the nascent (growing) protein. The translation is a complex process and not all the details are understood. Luckily most of these details are not crucial for understanding of bioinformatics. What is crucial however is to realise that there is nothing magical about proteins synthesis. The end of translation is the final part of gene expression and the final product

[5] http://www.estrellamountain.edu/faculty/farabee/biobk/biobooktoc.html

is a protein, the sequence of which corresponds to the sequence encoded by the mRNA. Proteins can be post-translationally modified e.g., by adding of sugars or cleavage (chopping), and this affects their location and function.

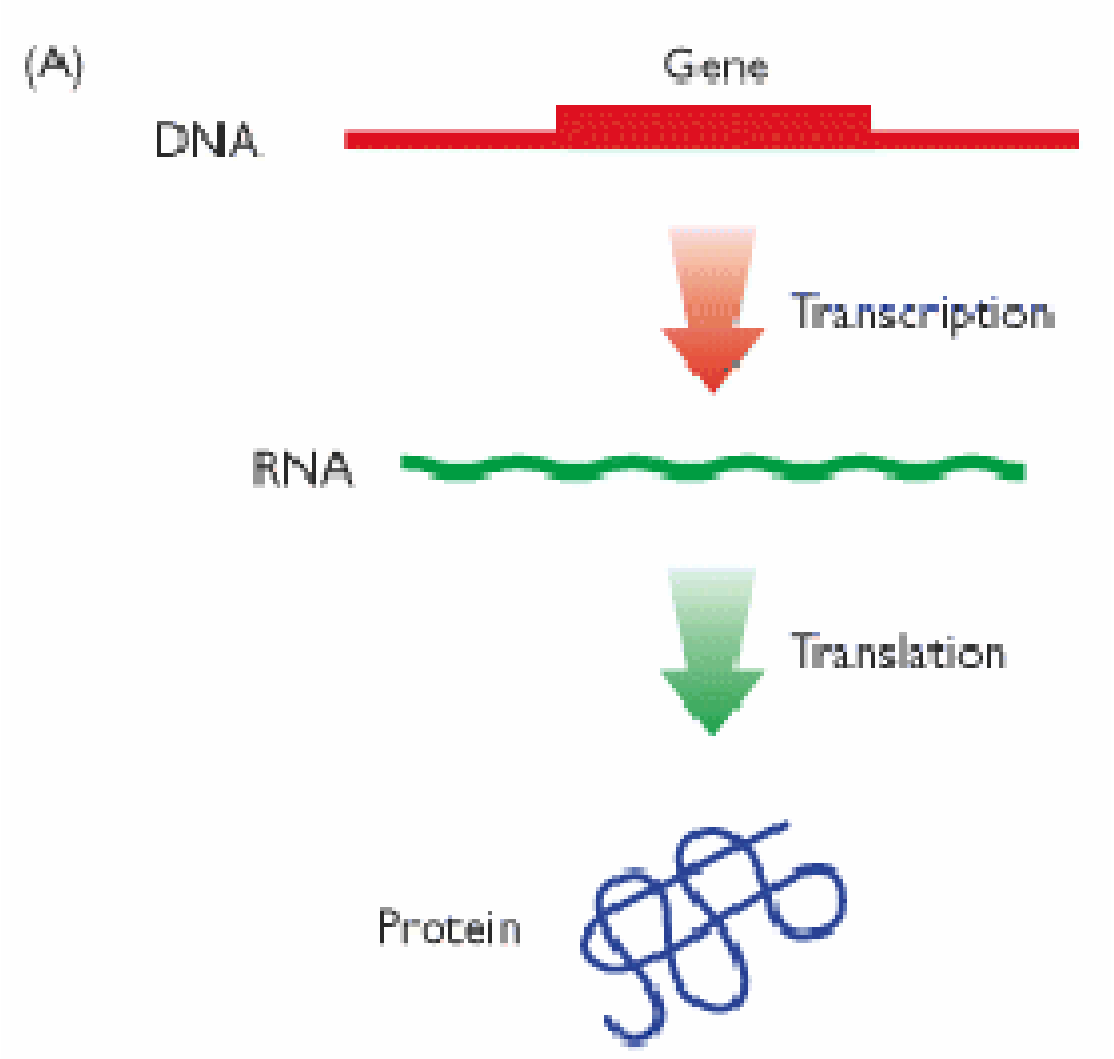


FIGURE 2.8: Transcription and Translation.

Biologists used to believe in paradigm - 'one gene - one protein'. Now this is known not to be true - due to alternative splicing and post-translational modifications one gene can produce a variety of proteins. There are also genes that do not encode proteins but encode RNA (for instance tRNA and ribosomal RNA).

Table 2.2 shows genes number and percentage of the whole genome that encodes protein per different organism. One of the surprises is the relatively small number of genes in a human genome in comparison to worm. Before most of the human genome sequencing was accomplished, it was estimated that there should be about 100,000

genes in a human. In fact some experts still think that there must be at least 40,000 - 50,000 genes in the human genome, and that 30,000 just reflects the unreliability of in silico (i.e., computational) gene prediction. Still, it seems that there is no simple correlation between the intuitive (not well-defined) complexity of an organism and the number of genes in its genome (for instance, intuitively fly is more complex organism than worm). One reason for the low number of genes in the human genome may be

TABLE 2.2: Genome Percentage that Encodes Proteins

Organism	The Number of Predicted Genes	Part of the Genome that Encodes Proteins (Exons)
E.Coli (bacteria)	5000	90%
Yeast	6000	70%
Worm	18,000	27%
Fly	14,000	20%
Weed	25,500	20%
Human	30,000	<5%

that there are more splice variants per gene in humans, though this has yet to be proved (otherwise human vanity may have to suffer).The presence of 95% of non-coding DNA in the human genome (sometimes called the junk DNA) remains a mystery. There are several hypotheses explaining this, but none is generally accepted. One controversial hypothesis (promoted by Richard Dawkins) is based on the idea of so-called selfish DNA. It states that the DNA is the basic element for natural selection, implying that DNA tries to propagate (multiply and amplify) itself, while the cells and organisms are vehicles to achieve this.

2.2.4 Gene Function

The first, and very important, step in the elucidation of the function of a novel gene is to compare the amino-acid sequence of its predicted protein product with those of other protein sequences in the public data libraries to see whether it is similar to a protein of known function that has previously been characterized in another organism [39].
Second, elucidation of gene function either by identifying homologs of known function from other species [39]. As the different systematic genome sequencing projects progress, there is a growing set of genes that have homologs in a range of organisms but in none of these organisms is their function understood.
Finally, A powerful way to elucidate the function of novel genes uncovered by systematic genome sequencing projects is to determine the physiological conditions, or

developmental stage, where those genes are expressed and relate their expression patterns to those of genes whose function is well known.

2.3 Data Acquisition Methodologies

2.3.1 DNA Sequencing Methodology

Determining the four letter sequence for a given a DNA molecule is known as the DNA sequencing. The first full genome for a bacterium was sequenced in 1995. The yeast (*Saccharomyces cerevisiae*) genome was sequenced in 1997, worm (nematode Caenorhabditis elegans) in 1999, fly (Drosophila melanogaster) in 2000, and weed (Arabidopsis thaliana) at 2001. Human genome was completed in 2003, this is known as the draft human genome. Rapid and efficient methods for DNA sequencing were first devised in the mid-1970s. Two different procedures were published at almost the same time [40]

- The chain termination method (Sanger et al., 1977), in which the sequence of a single-stranded DNA molecule is determined by enzymatic synthesis of complementary polynucleotide chains, these chains terminating at specific nucleotide positions;
- The chemical degradation method (Maxam and Gilbert, 1977), in which the sequence of a double-stranded DNA molecule is determined by treatment with chemicals that cut the molecule at specific nucleotide positions.

Both methods were equally popular to begin with but the chain termination procedure has gained ascendancy in recent years, particularly for genome sequencing. This is partly because the chemicals used in the chemical degradation method are toxic and therefore hazardous to the health of the researchers doing the sequencing experiments, but mainly because it has been easier to automate chain termination sequencing. Sequencing of the relatively small bacterial genomes has become routine and is largely done by sequencing robots and completed by human researchers, the main problem being the minimisation of costs per letter, and maximisation of the speed while maintaining quality. Sequencing of larger genomes, like a human genome, is still difficult, though most of the problems are computational. Sequencing robots are able to sequence only relatively short stretches of DNA, which afterward have to be assembled together by a computer using assembly algorithms. The main difficulty is that genomes of higher eukaryotes (like humans) have many repeated subsequences, which makes the

assembly rather tricky, this means that considerable human intervention is still needed in the final stage of sequencing projects. The worlds largest public genome sequencing project is housed at the Sanger Institute .

2.3.2 Microarray Technology

In the 1960s, in the absence of methods for direct analysis of mRNAs, cellfree translation systems provided an indirect approach to measure the abundance of a specific mRNA. The amount of a specific protein produced during the cell-free translation of an mRNA population was assumed to reflect the abundance of the cognate mRNA in that population which have many disadavatage like: First, it does not give an indication of the integrity of the mRNA being analyzed. Second, it can be difficult to quantitate low-abundance mRNAs accurately because of relatively high backgrounds [39].
During short period a variety of other approaches have been developed to determine the level of gene expression during biological process, for instance: RT-PCR, Northern Analysis, Reporter Genes ,cDNA Technologies,Serial Analysis of Gene Expression (SAGE),Hybridization Array Technology which have been described in detail elsewhere. The former, which we will be focused in this section has been growing exponentially during last years.
The Hybridization Array Technology or the DNA microarray is a high-throughput technology used in molecular biology and in Medicine and plays a central role in the field of functional genomics. With this new technology it is possible to measure the mRNA abundance (gene expression) for tens of thousands of genes in parallel in a single experiment. It makes use of the sequence resources created by the genome projects and other sequencing researches to find out what genes are expressed in a particular cell type of an organism, at a certain condition, at specific environmental restrictions. Also, it can be used to predict the gene functions by evaluating the microarray genes subsets (after clustering) with standard gene annotations such as Gene Ontology (GO) (see Section 3.3 for details). This approach has been successfully used on yeast to predict the function of over 800 uncharacterized genes [41].Moreover, microarrays can detect viruses or other pathogens in blood samples [42].Finally, gene expression profiling is another important application of microarrays. It can be used to identify genes whose expression is changed in response to pathogens or disease by comparing gene expression in infected to that in uninfected cells.
Microarrys technology help answer important questions like:which genes are expressed in all cell types, what are the functional roles of these genes, how big is the gene function universe, how many genes are needed for life, how it can be that a worm has more genes

than a fly, and the human only a bit more than a worm, and, of course, we can always revisit the question of the meaning of life.

2.3.2.1 Microarray Concepts

A microarray is typically a glass on which DNA molecules are attached at fixed coordinates (spots). There may be tens of thousands of spots on an array, each containing a huge number of identical DNA molecules(Figure 2.9). In the experiments of gene expression profiling, each of these DNA molecules in the spot should identify one gene or one exon in the genome; however, in fact, this is not always the case due to the families of similar genes in a genome. The spots printed on the microarrays by a robot, or synthesized by ink-jet printing. There are distinct methods how microarrays can be used to measure the gene expression levels. One way is to compare the gene expression levels in two different samples, e.g., the same cell type in a healthy and diseased status(Figure 2.10). The mRNA from the cells in two different conditions is extracted and labeled with two different fluorescent labels: green dye for normal cells and a red dye for tumor. Both extracts are washed over the microarray. Labeled gene products from the extracts hybridize to their complementary sequences in the spots due to the preferential binding(Figure 2.11). The dyes enable the amount of sample bound to a spot to be measured by the level of fluorescence emitted when it is excited by a laser scanner. If the RNA from the normal sample is in abundance, the spot will be green, if the RNA from the tumor sample is in abundance, it will be red. If both are equal, the spot will be yellow (red + green), while if no color is exist it will not fluoresce at all and appear black. Thus, from the fluorescence intensities and colors for each spot, the genes expression levels can be measured. See Figure 2.10.

The raw data that are produced from microarray experiments are the hybridised microarray images. To obtain information about gene expression levels, these images should be analysed, each spot on the array identified, its intensity measured and compared to the background. This is called image quantitation.
Image quantiation Figure 2.12 is done by image analysis software. To obtain the final gene expression matrix from spot quantiations, all the quantities related to some gene (either on the same array or on arrays measuring the same conditions in repeated experiments) have to be combined and the entire matrix has to be scaled to make different arrays comparable.

Recently, several gene expression microarray databases have been founded to store such huge amount of microarray data. Also these databases can be queried, compared

[6]http://www.affymetrix.com
[7]http://www.affymetrix.com
[8]http://www.ebi.ac.uk/microarray/biology_intro.html#Molecules

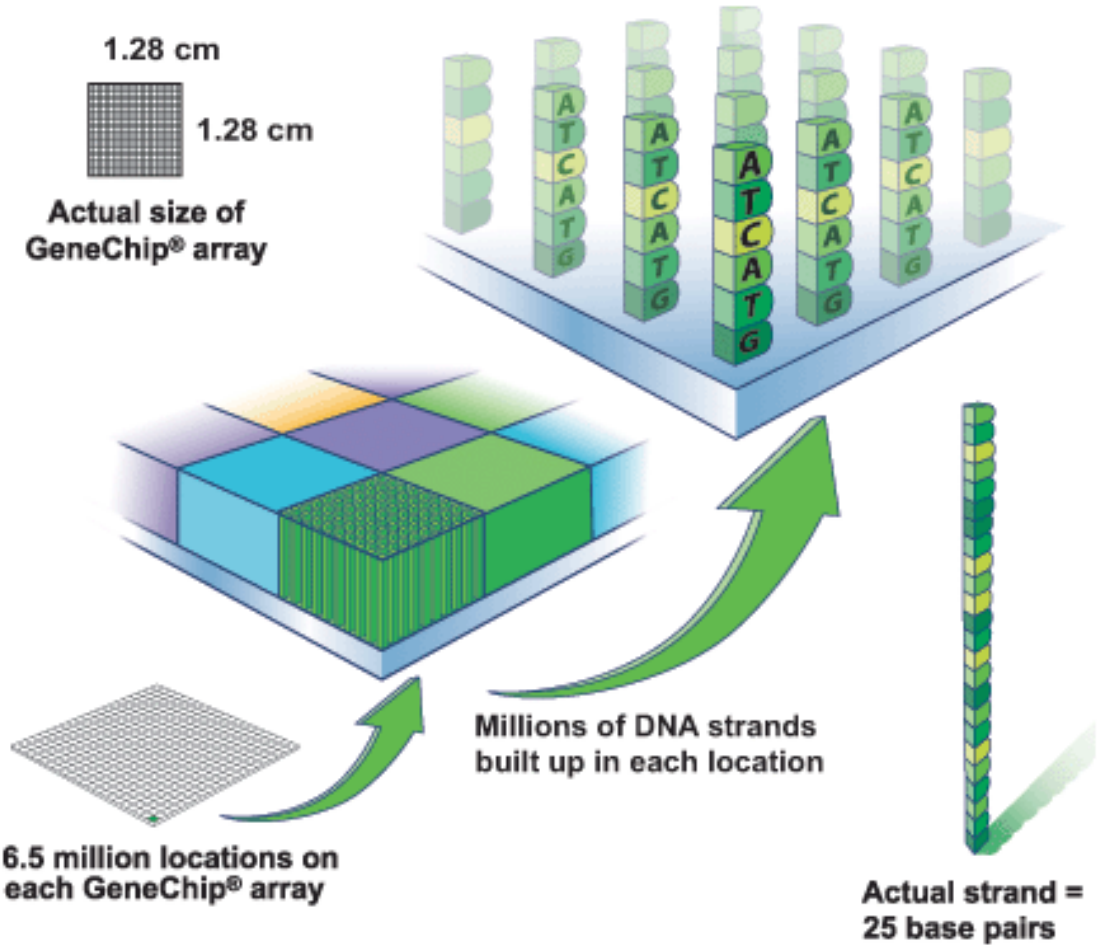


FIGURE 2.9: **Microarray Chip**: There May be Tens of Thousands of Spots on an Array, each Containing a Huge Number of Identical DNA Molecules. Picture Taken from Affymetrix [6].

and analyzed by various software tools. Gene Expression Omnibus (GEO), Array Express, GeneX and Stanford Microarray Database (SMD) are the commonly used microarrays online databases.

2.3.2.2 Gene Expression Data Analysis

Capturing and storage of microarray data is not an end in itself. The amounts of data from even a single microarray experiment are so large, that preprocessing step have to be used to make any sense out of it. Data denoising, clustering, biclustering, normailzation, are typical methods currently used in gene expression data analysis and described in details in the next chapter.

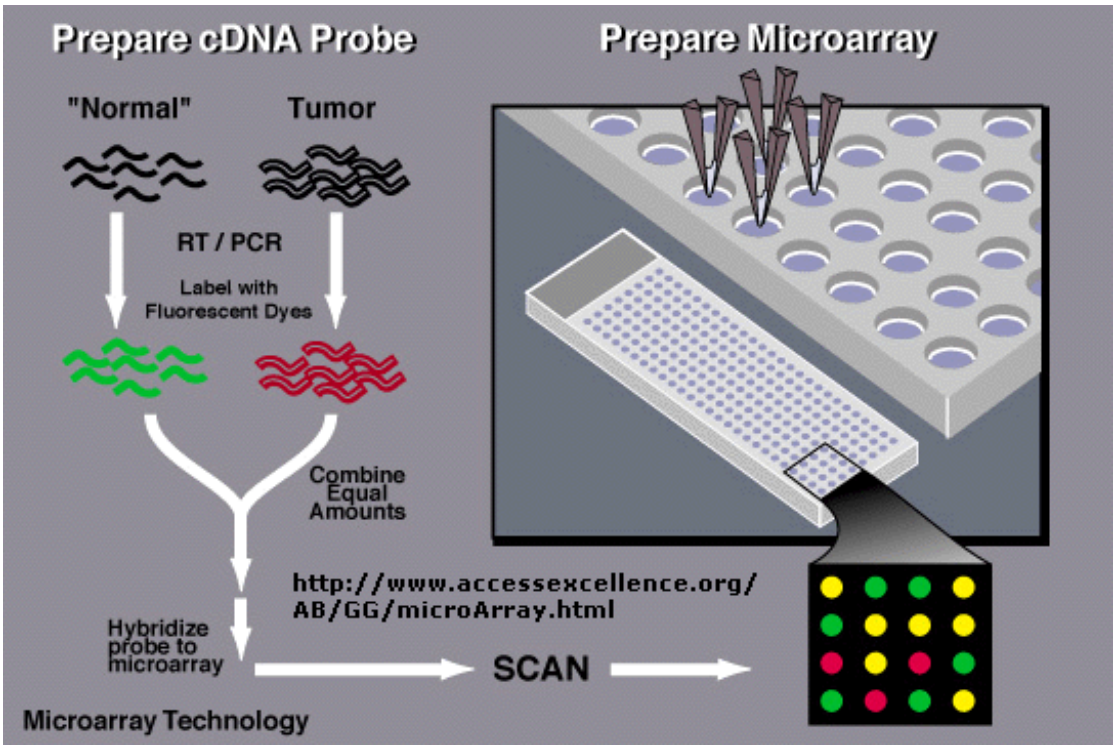


FIGURE 2.10: **Microarray Experiment**: The Total mRNA from the Normal and Tumor Cells is Extracted and Labelled with two Different Fluorescent Labels: For Example a Green Dye for Normal Cells and a Red Dye for Tumor Cells.

2.3.2.3 Microarray Data Limitations

Microarrays have some limitations, and one should note the following potential sources of problems: manufacturing reproducibility; variation in how the experiments are performed such as exposure duration, temperature gradients and flow conditions. These problems might cause negative effects on the hybridization process [43].
Many microarrays problems concerning specificity, accuracy, reproducibility and the biology have been addressed and assessed in the literature [44, 45]. If an mRNA is present in relatively high abundance it can probably be detected reliably, however, low-copy numbers lead to problems. Cross-hybridization is likely to be common and adds to the noise in the measurements. One has to keep in mind that microarrays only measure the mRNA concentration, but mRNA half-life is sequence dependent and varies widely. Some mRNAs are being stored for a long time until needed. Therefore, a high concentration of a mRNA does not necessarily mean that the corresponding gene is active; the concentration of a particular mRNA might not be correlated with the concentration of the corresponding protein, nor the concentration of the active form of the protein, because all posttranscriptional steps can be regulated individually.

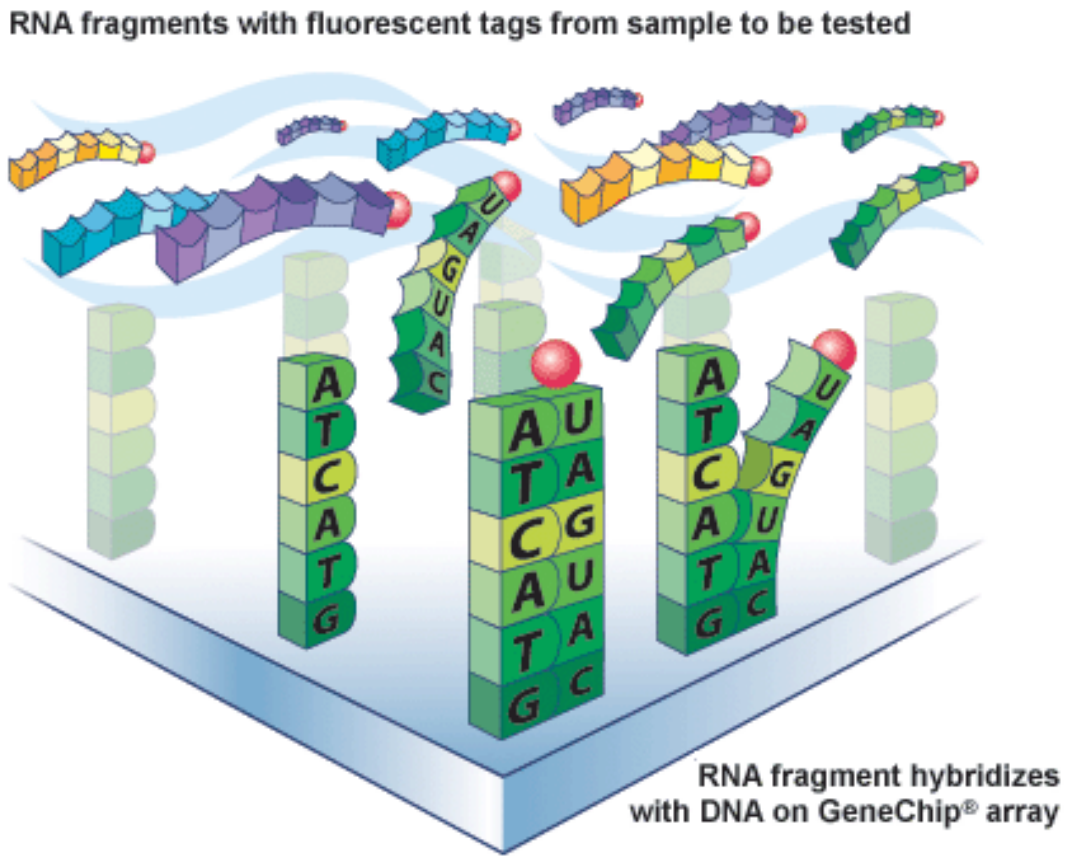


FIGURE 2.11: **Microarray Hybridization**: Labeled Gene Products from the Extracts Hybridize to their Complementary Sequences in the Spots. Picture Taken from Affymetrix [7].

2.4 Biological Database

Biological databases are libraries of life sciences information, collected from scientific experiments, published literature, high throughput experiment technology, and computational analyses. They contain information from research areas including genomics, proteomics, metabolomics, microarray gene expression, and phylogenetics [30]. Information contained in biological databases includes gene function, structure, localization (both cellular and chromosomal), clinical effects of mutations as well as similarities of biological sequences and structures [46].

- Primary Sequence Databases All the public DNA sequences are stored in the EMBL database (also known as EMBL-Bank), which is in fact a collaboration of

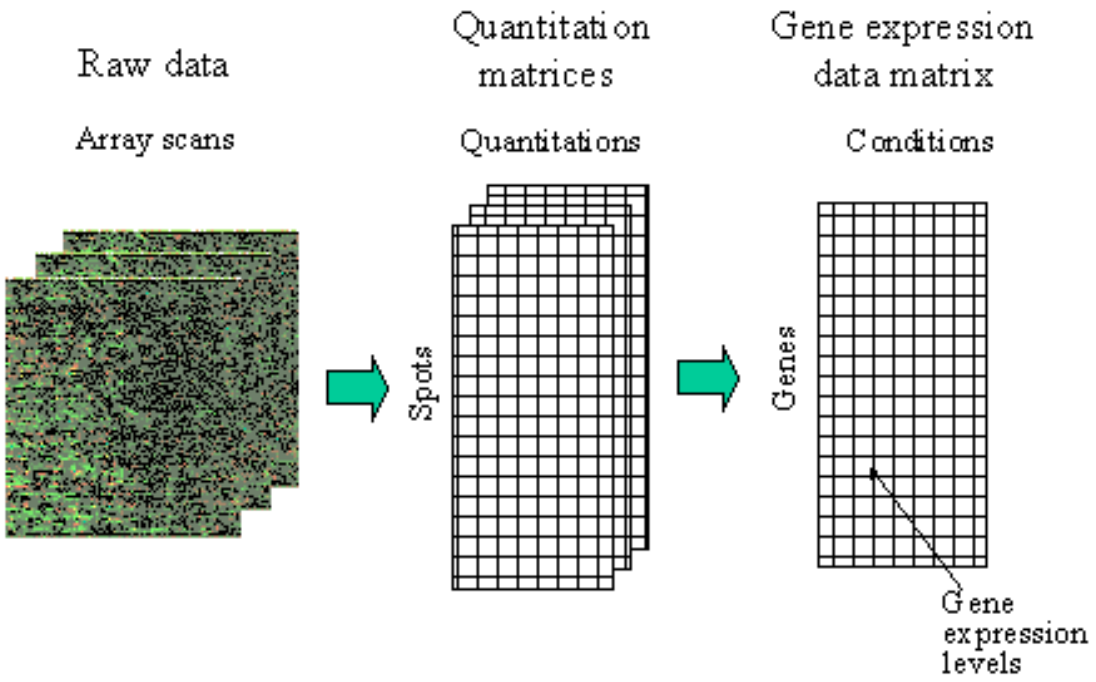


FIGURE 2.12: **Microarray Image Quantitation**: Each Spot on the Array Identified, its Intensity were Measured and Compared to the Background which is Called Quuantitation. Picture Taken from EBI [8].

three databases EMBL in Europe, GenBank in the USA and DDBJ in Japan (each database mirrors the others and they exchange data every 24 hours).
Figures 2.13(a),2.13(b),2.13(c)show how the DNA sequence is formatted.

- Protein Sequence Databases

  1. UniProt: Universal Protein Resource (UniProt Consortium: EBI, Expasy, PIR)
  2. PIR Protein Information Resource (Georgetown University Medical Center (GUMC)). Figure 2.13(d) shows PIR sequence format.
  3. Swiss-Prot: Protein Knowledgebase (Swiss Institute of Bioinformatics). Its format is very similar to EMBL format, except considerably more information about the physical and biochemical properties of the protein is provided.

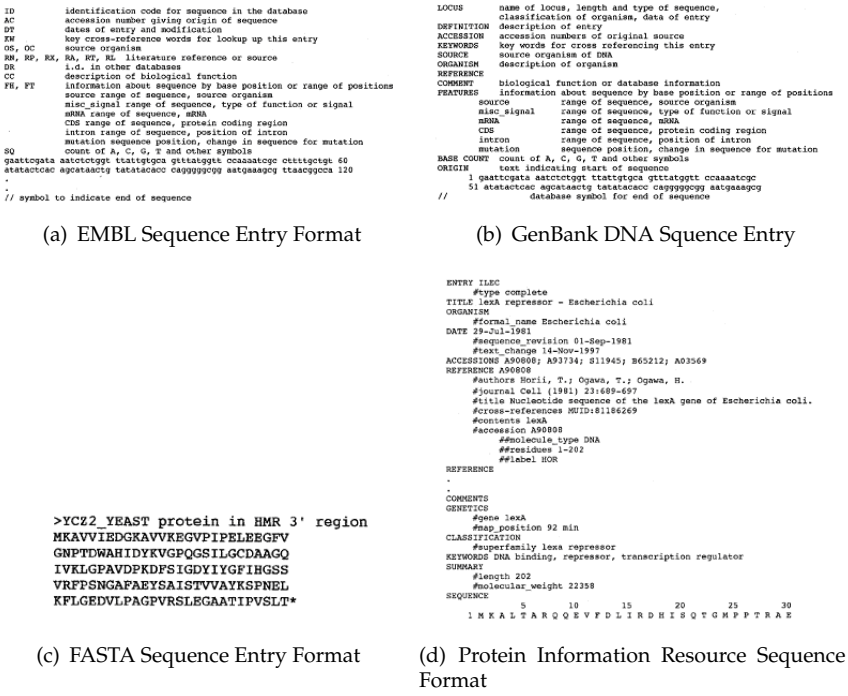
- Protein Structure Databases

  1. Protein Data Bank (PDB) (Research Collaboratory for Structural Bioinformatics (RCSB))
  2. CATH Protein Structure Classification
  3. SCOP Structural Classification of Proteins

4. SWISS-MODEL Server and Repository for Protein Structure Models
5. ModBase Database of Comparative Protein Structure Models (Sali Lab, UCSF)

- Protein Protein Interactions

1. BioGRID A General Repository for Interaction Datasets (Samuel Lunenfeld Research Institute)
2. STRING: STRING is a database of known and predicted protein-protein interactions. (EMBL)
3. DIP Database of Interacting Proteins



(a) EMBL Sequence Entry Format

(b) GenBank DNA Squence Entry

(c) FASTA Sequence Entry Format

(d) Protein Information Resource Sequence Format

FIGURE 2.13: Examples of Biological Databases Format.

2.5 Saccharornyces Cerevisiae

In April 1996, the complete genome sequence of the brewers and bakers yeast *Saccharornyces Cerevisiae* was sequenced. The project was launched by an initiative of A. Goffeau (1989) and the European Commission (EU) to sequence chromosome 111 in a pilot study. This was an important event, not just because it was the first complete eukaryotic genome sequence, but also because it was the first total sequence for an important model organism for which there is a large constituency of researchers ready and able to exploit the sequence data.

2.5.1 Why Yeast

We can summarized why the current research focus on Yeast as following

1. Yeast has already provided biologists with a valuable resource for determining the function of individual human genes involved in medical problems, such as cancer, neurological disorders, and skeletal disorders. Over the next few years, scientists in the United States and Europe will piece together for the first time a comprehensive look at how all the genes in a eukaryotic cell function as an integrated system."The yeast genome is closer to the human genome than anything completely sequenced so far," said Dr. Francis Collins, director of the National Center for Human Genome Research (NCHGR), part of the National Institutes of Health (NIH).

2. Biologists have studied yeast, known by its scientific name *Saccharomyces Cerevisiae*, for many decades because it offers valuable clues to understanding the workings of more-advanced organisms. Humans and yeast, for example, share a number of similarities in their genetic make up. For one, many regions of yeast DNA contain stretches of DNA subunits, called bases, that are very close or identical to those in human DNA. These similarities tell scientists the genes in those regions play a critical role in cell function in both species, or they would have been lost during the 1 billion years of evolution that separate yeast and humans. About one-third of yeast genes are related to those in the human. Some of these critical processes include DNA copying and repair of damaged DNA, protein synthesis and transport across membranes, and control of metabolic processes.

3. In cancer research, *S. Cerevisiae* has emerged as an important model for studying control of the eukaryotic cell cycle.Although yeast DNA shares many similarities with human DNA, finding yeast genes is easier because the yeast genome lacks the long stretches of filler DNA and repeated bases the human genome contains, which often cause scientists problems when examining a long DNA piece for the presence of genes. Yet, scientists know

4. The difficulty of experiments on human body.

5. Nobel Laureates in Physiology or Medicine for 2001 have awarded to Leland H. Hartwell, R. Timothy (Tim) Hunt and Paul M [9]. Nurse. They made seminal discoveries concerning the control of the cell cycle. They have identified key molecules that regulate the cell cycle in all eukaryotic organisms, including yeasts, plants, animals and human based on study on Yeast. Defects in cell cycle control

[9] http://nobelprize.org/nobel_prizes/medicine/laureates/2001/press.html

may lead to the type of chromosome alterations seen in cancer cells. This may in the long term open new possibilities for cancer treatment.

6. Yeasts have recently been used to generate electricity in microbial fuel cells, and produce ethanol for the biofuel industry.

2.5.2 Yeast Genes Features

To date of writing this thesis [10] the complete sequence reveals approximately 6607 protein-encoding genes, 27 ribosomal RNA genes, 21 pseudogene, and 299 tRNA genes. Figute 2.14 shows that one-third of yeast genes remain uncharacterized. The uncharacterized ORF is an open reading frame (ORF) is one that is likely to encode an expressed protein, as suggested by the existence of orthologs in one or more other species, but for which there are no specific experimental data demonstrating that a gene product is produced in *S.Cerevisiae*. Structural characteristics on every yeast ORF can be accessed through the MIPS WWW server[11]. This information is continuously and automatically updated as new sequence and structure information becomes available. Other sources of 3D structural information related to the yeast genome are the GeneQuiz resource[12] at EMBL (Heidelberg) and the Sach3D facility[13] provided by the Saccharomyces Genome Database (SGD) at Stanford. Functional analysis of the *Saccharornyces Cerevisiae* genes using gene ontology lastes version are shown in figure 2.15.

2.5.3 Yeast Gene Naming

2.5.3.1 Standard Name

The official Gene Name of an *S. Cerevisiae* gene is referred to as the Standard Name on an SGD locus page, and generally becomes the standard name based on its publication in a peer-reviewed paper describing characterization of that gene. Any alternative Gene Name is referred to as an Alias [47].
Gene Names in *S. Cerevisiae* are generally three letters followed by a number. Example CDC28 - a Gene Name conferred on a nuclear ORF on the basis of genetic characterization. Different copies of duplicated genes may be indicated by an extension to the end of the Gene Name. This extension can made by either adding a letter, e.g. 'A' or 'B', as in the case of the ribosomal protein genes, or by adding a hyphen and a number, e.g.

[10] http://www.yeastgenome.org/cgi-bin/search/featureSearch
[11] http:/ /www.mips.biochem.mpg.de
[12] http:/ /www.embl-heidelberg.de/-genequiz/ yeast.htm1
[13] http://genome-www.stanford.edu/Sacch3D

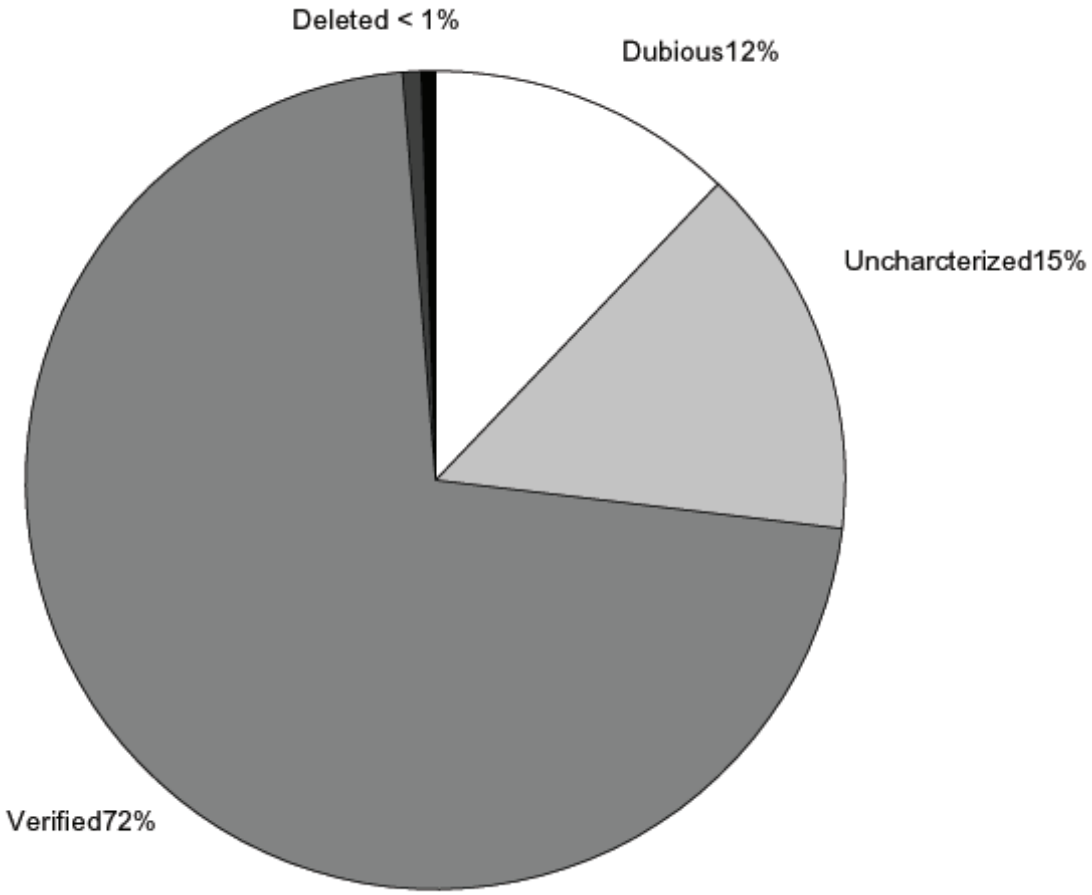


FIGURE 2.14: Chromosomal Features of the *S.Cerevisiae*.

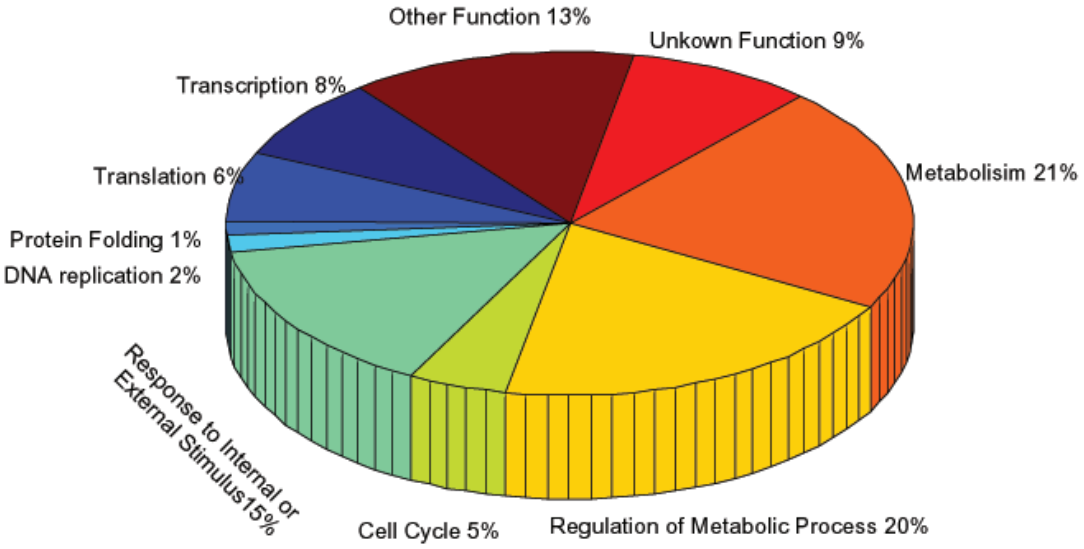


FIGURE 2.15: Classification of *Saccharornyces Cerevisiae* Genes According to the Functional Category (GO).

'-1', '-2', as in the case of the YRF1 genes encoding the Y'-helicase or the ribosomal RNA genes. Example RPL1A - a Gene Name conferred on one copy of the gene encoding a copy of large subunit protein 1
RPL1B - a Gene Name conferred on the other copy of the gene encoding a copy of large subunit protein 1

2.5.3.2 Systematic Names

The Systematic Name is the name generated by the systematic sequencing project, or conferred later according to the appropriate guidelines for systematic nomenclature for that type of new feature or gene [48].
YAL001C - first ORF to the left of the centromere on chromosome I (A is the 1st letter of the English alphabet), on the complement or Crick strand.
For mitochondrially encoded ORFs, the systematic names start with a 'Q', to designate the mitochondrial chromosome; the rest consists of a four digit number.
Example
Q0010 - an ORF encoded in the mitochrondrion
Q0032 - another ORF encoded in the mitochrondrion. For tRNAs, the systematic names begin with a lowercase 't'; the second letter corresponds to the single letter code for the appropriate amino acid, e.g. A = alanine, C = cysteine, etc.; next the sequence of the anticodon of the tRNA is given in the 5' -¿ 3' direction within parentheses, e.g. (AGC), (GUC); finally, there is an indication of which chromosome the tRNA gene resides on using the letters 'A' through 'P' to designate nuclear chromosomes (in the same way as for nuclear-encoded ORFs).
tC(GCA)B - a tRNA for cysteine, with the anticodon sequence 'GCA', located on chromosome II
tS(AGA)D1 - a tRNA for serine, with the anticodon sequence 'AGA', one of two or more tRNAs from this family (containing the AGA anticodon) located on chromosome IV.

2.5.4 Yeast Databases

2.5.4.1 Genome Database

The complete yeast genome is presented on the (World Wide Web) WWW by several groups that have either independently or collaboratively developed different approaches to organize, present, and access yeast genome data. SGD and MIPS are the important data base about the Yeast.

- The SGD at Stanford provides access to knowledge associated with yeast genes for both yeast and non-yeast researchers. It administers a S.Cerevisiue Gene Name Registry and is interconnected to other major resources as YPD, GenBank, Medline, MIPS, SWISS-PROT, and the Kyoto Encyclopedia of Genes. In addition BLAST and FASTA searches against the yeast genome sequence are provided and polymerase chain reaction (PCR) primer design is available.
- The MIPS resource[14] includes a large set of detailed annotations on yeast proteins and provides comprehensive access to several query and graphical interfaces to browse the genome

2.5.4.2 Microarray Database

There are four major microarray repositories listed as the following:

- GEO: Gene Expression Omnibus at the NCBI, provides data in a tab-delimited format.
- ArrayExpress: part of the EBI, provides data in MAGE-ML format.
- SMD: Stanford Microarray Database, provides data published at Stanford University.
- YMGV: Yeast Microarray Global Viewer, provided by the Jacq group in Paris, France.

Also, There are many web tools to retrive Yeast microarrays experiments like: SPELL[15], Webminer [16], Expressionconnection [17] in addition, for a list of micoarray database please see[18] Table 2.3 show the list of popular microarray experiments on Yeast.

In the below section we are described the two widely important dataset which are Spellman [28] and Gasch [27].

2.5.5 Spellman Cell Cycle Experiment

The cell division cycle is a complex self-regulating program, such that many genes involved in aspects of the cell cycle are also controlled by it. Such regulation might

[14]http://speedy.mips.biochem.mpg.de/mips/ yeast
[15]http://function.princeton.edu/SPELL
[16] http://genome-www.stanford.edu/webminer/
[17] http://www.yeastgenome.org/cgi-bin/expression/expressionConnection.pl
[18]http://microarrayworld.com/DatabasePage.html

TABLE 2.3: Examples of Popular Yeast Microarray Dataset from Stanford University

Expierments Type and Time Points	Author
the cell division cycle after synchronization by alpha factor arrest (ALPH; 18time points) centrifugal elutriation (ELU; 14time points) temperature-sensitive cdc15 mutant (CDC15; 15time points)	Spellman et al.,1998
Sporulation (SPO;11 time points)	Chu et al., 1998
shock by high temperature (HT, 6time points) reducing agents (D, 4time points) shock by low temperature (C; 4time points)	P.T.S., J.Cuoczo, C.Kaiser, P.O. B., and D.B., unpublished work
diauxic shift(diau;7 time points)	DeRisi, J. L et al 1997
temperature-sensitive cdc28 mutant (CDC28; 17time points)	Cho et al., 1998
heat shocks (35+5 time points) hydrogen peroxide(15+5 time points) superoxide-generating drug menadione (9 time points) sulfhydryl-oxidizing agent diamide (9 time points) disulfide-reducing agent dithiothreitol (15 time points) hyper- and hypo-osmotic shock (12 time points) amino acid starvation (5 time points) nitrogen source depletion(10 time points) progression into stationary phase(8 time points) diauxic shift(7) overexpression(3)	Gasch et al., 2000

be required for the proper functioning of mechanisms that maintain order during cell division. Alternatively, regulation of these genes could simply allow conservation of resources. Spellman et al [19][28] try to identify the genes whose RNA levels varied periodically during the cell cycle. The obtained microarray data were analyzed by deriving a numerical score based on a Fourier algorithm (testing periodicity) and by a correlation function that identified genes whose RNA levels were similar to the RNA levels of genes already known to be regulated by the cell cycle. Spellman et.al [28] found that ;800 genes are cell cycle regulated, which constitutes 10% of all protein-coding genes in the genome. Furthermore, analyzing cell cycleregulated genes for known and new promoter elements showing that several known elements (or variations thereof) contain information predictive of cell cycle regulation.

[19]http://genome-www.stanford.edu/cellcycle/

2.5.6 Gasch Environmental Changes Experiment

The complexity of the yeast cells system for detecting and responding to environmental variation is only beginning to emerge. Genes whose transcription is responsive to a variety of stresses have been implicated in a general yeast response to stress. Other gene expression responses appear to be specific to particular environmental conditions. Several regulatory systems have been implicated in modulating these responses, but the complete network of regulators of stress responses and the details of their actions, including the signals that activate them and the downstream targets they regulate, remain to be elucidated. Gasch et al [20][27] explored genomic expression patterns in the yeast *Saccharomyces Cerevisiae* responding to diverse environmental transitions. DNA microarrays were used to measure changes in transcript levels over time for almost every yeast gene, as cells responded to 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, nitrogen source depletion, and progression into stationary phase. A large set of genes (; 900) showed a similar drastic response to almost all of these environmental changes. Promoter analysis on these genes provided clues to novel regulators.

[20] http://genome-www.stanford.edu/yeast_stress

Chapter 3

Gene Expression Data Analysis

From the previous chapter we defined the gene expression matrix n X m as expression level of n genes, getting from m microarray experiments. Because of experimental error this matrix contains missed vales, low gene profile and high noise ratio. In this chapter we attempt to overcome data dimensionality problem using clustering/biclustering techniques(section 3.3,sec:biclustering) and reduce data noise using preprocessing steps(data denoising(section 3.2.6), remove non informative gene profiles(section 3.2.2,....).

3.1 Data Acquisition

The data used in this work are the two well-known datasets of yeast microarray gene expression (Gasch et al [27]; Spellman et al [28]), which can downloaded from Stanford Microarray Database [1]. The Spellman dataset consists of four synchronization experiments (alpha factor arrest, elutriation and arrest of CDC15 and CDC28 temperature-sensitive mutants) which were performed for a total of 73 microarrays during cell-cycle. The Gasch dataset contains 6152 genes and 173 conditions of diverse environmental transitions such as temperature shocks, amino acid starvation, and nitrogen source depletion.

3.2 Preprocessing

Measurements of microarrays may be biased by diverse effects such as efficiency of RNA extraction, reverse transcription, label incorporation, exposure, scanning, spot

[1] http://smd.stanford.edu/

detection, etc. This necessitates the preprocessing of microarrays prior to data analysis. The datasets used in this work have been already preprocessed for background correction and normalization. Further preprocessing steps should be applied for data refinement.

3.2.1 Update Genes List

As many chromosomal changes occurred from the date of Gasch et al [27] and Spellman et al [28] experiments, they contain genes that are not exist any more. We used the SGD Batch Download web tool [2] to remove all the merged, deleted and retired genes from any further processing.
Based on the Saccharomyces Genome database (SGD) [49] There are 6607 ORF genes of the *S.Cerevisiae* versus 6152 genes which were reported in Gasch dataset [50]. The Annotation/sequence properties of these genes are shown in Table 3.1.

TABLE 3.1: The Annotation/sequence properties of the *Saccharomyces Cerevisiae* genes: SGD database genes Vs genes which were reported in Gasch [27] dataset

Sequence Properties	Number of Genes	
	SGD	Gasch
Merged	38	20
Deleted	48	2
Verified	4807	4767
Uncharcterized	989	795
Dubious	811	568

The description of the chromosomal features which are shown in Table3.1 are described as following:

- **Merged Feature:** A chromosomal feature that was once annotated as a distinct entity, but that has now been subsumed by another feature. Typically, features become ”Merged” because of a change in chromosomal sequence or annotation (e.g. YAR004W). For record keeping, the ”Merged” feature is not removed from SGD, but is instead given the ”Merged” status as a flag [3].
- **Deleted Feature** A chromosomal feature that has been removed from the yeast genome catalog. Typically, features are ”Deleted” because they are effectively destroyed by a sequence or annotation change (e.g. YCL006C), or because the original annotation was in error or inappropriate (e.g. YCRX03C). For record

[2] http://www.yeastgenome.org/cgi-bin/batchDownload
[3] http://www.yeastgenome.org/help/glossary.html#merged

keeping, the "Deleted" feature is not removed from SGD, but is instead given "Deleted" status as a flag. Note that "Deleted" features are distinct from "Dubious" features in that "Deleted" features have been demonstrated to be incorrect and have been officially withdrawn [4].

- **Dubious ORF** A Dubious open reading frame (ORF) is one that is unlikely to encode an expressed protein. Dubious ORFs may meet some or all of the following criteria: 1) the ORF is not conserved in other Saccharomyces species; 2) there is no well-controlled, small-scale, published experimental evidence that a gene product is produced; 3) a phenotype caused by disruption of the ORF can be ascribed to mutation of an overlapping gene; and 4) the ORF does not contain an intron. Many ORFs classified as "Dubious" are small and overlap a larger ORF of the class "Verified" or "Uncharacterized"; however, overlap with another ORF does not mandate that an ORF be classified as "Dubious"[5].

- **Uncharacterized ORF** An Uncharacterized open reading frame (ORF) is one that is likely to encode an expressed protein, as suggested by the existence of orthologs in one or more other species, but for which there are no specific experimental data demonstrating that a gene product is produced in *S. Cerevisiae*. While most Uncharacterized ORFs have systematic names only (e.g., YKL100C), a few have been given genetic names (e.g., PAU8). Evidence from large-scale analyses that indicates an ORF may be biologically relevant is sometimes but not always enough to upgrade an ORF from "Uncharacterized" to ""Verified, depending on the individual case. Also see the description of "Dubious" ORFs [6].

- **Verified ORFs** ORFs for which experimental evidence exists that a gene product is produced in *S. Cerevisiae*. Generally these have obvious orthologs in one or more other Saccharomyces species. Most named genes are in this class. Evidence from large-scale analyses that indicates an ORF may be biologically relevant is sometimes but not always enough to upgrade an ORF from "Uncharacterized" to "Verified", depending on the individual case [7].

From Table 3.1, we found that some of Gasch genes become aliases, merged for other genes and some of them were deleted as in Tables[3.2, 3.3, 3.4] sequentially. These tables were produced by comparing the feature table file which was updated monthly by SGD curator [49] with Gasch ORF genes [50].
The changed in the ORF could be due to sequence changes and corrections,and ORF

[4]http://www.yeastgenome.org/help/glossary.html#deleted
[5]http://www.yeastgenome.org/help/glossary.html#dubious
[6]http://www.yeastgenome.org/help/glossary.html#uncharacterized
[7]http://www.yeastgenome.org/help/glossary.html#verified

merges[8]. For an example gene YCL006C and YCR103C were deleted i.e do not encode a protein; these ORFs were removed when a sequence update created a stop codon after residue. Another example is YAR044W which is merged open reading frame, that does not encode a discrete protein; YAR044W was originally annotated as an independent ORF, but as a result of a sequence change, it was merged with an adjacent ORF into a single reading frame, designated YAR042W as shown in Table 3.3.

TABLE 3.2: Gasch ORF Genes Which Become Aliases for Other Genes

Genes were reported in Gasch dataset which become alias for other genes	New Systematic Name
YAL035C-A [9]	YAL034C-B
YAL043C-A	YAL042C-A
YAL058C-A	YAL056C-A
YBL101W-A	YBL100W-A
YBL101W-B	YBL100W-B
YEL076W-C	YEL075W-A
YGR272C	YGR271C-A
YIL015C-A	YIL014C-A
YML010C-B	YML009C-A
YML013C-A	YML012C-A
YML032C-A	YML031C-A
YML035C-A	YML034C-A
YML048W-A	YML047W-A
YML058C-A	YML057C-A
YML095C-A	YML094C-A
YML102C-A	YML101C-A
YML117W-A	YML116W-A
YMR158W-A	YMR158W-B

After above gene merging and deletion from Gasch genes list, Gasch dataset was reduced to contain 6130 genes which can be downloaded from the below link:
http://home.k-space.org/FADL/Downloads/PhD/Data/Gasch-6130.txt

3.2.1.1 Important *S. Cerevisiae* Genes Which Are Not Included In Gasch Dataset

Gasch dataset [50] contains 6152 genes and 173 experiments. Not all *S.Cerevisiae* genes were included with Gasch genes list. Table 3.5 shows the list of important genes which are not included in Gasch experiment. These genes have high connectivity in the interaction databases. Interactome databases using Bionetbuilder [51] client server contains more than 98700 which are extracted from different online interaction data base like:BIND(107), BIOGRID(926), DIP(88), HPRD(1), INtACT(73),Interologger(2),

[8] personal communication with the SGD curator

TABLE 3.3: Gasch ORF Genes Which have Merged with Other genes

Genes as were reported in Gasch dataset which were merged with other genes	New Systematic Name
YAR044W	YAR042W
YBR075W	YBR074W
YBR100W	YBR098W
YCL012W	YCL014W
YCR062W	YCR061W
YDL038C	YDL039C
YDR474C	YDR475C
YFL006W	YFL007W
YFL043C	YFL042C
YFR024C	YFR024C-A
YGL046W	YGL045W
YJL017W	YJL016W
YJL018W	YJL019W
YJL021C	YJL020C
YKL158W	YKL157W
YKL199C	YKL198C
YML033W	YML034W
YOR088W	YOR087W
YOR240W	YOR239W
YPR090W	YPR089W

TABLE 3.4: Gasch Genes which were Deleted From the *S Cerevisiae* Genome

Genes in Gasch dataset which were deleted
YCL006C
YCR103C

KEGG(101), MINT(12), MPPI(0) and prolinks(0). The connectivity properties for each some of these genes are shown in Table 3.5. for instance,Edgecount is the number of edges connected to or from the corresponding genes(it indicates the degree of activity of this gene), Indegree is the number of inputed edges to the gene(indicates how much it was effected by other genes) and the Outdegree is the number of outputed edges from the gene(indicates gene effectiveness on other genes). Ignorance of these genes from analysis cost losing of 1309 interactions (1.3%).

3.2.2 Filtration

There are small changes of thousands of genes with important biological outcomes effected by small relative changes in the expression levels which cannot be reliably

TABLE 3.5: Important ORF *S.Cerevisiae* Genes Which Are Not Included in Gasch Dataset [50]

ORF	EdgeCount	Indegree	Outdegree
ORF:YDR363W-A	166	31	135
ORF:YBR111W-A	116	111	5
ORF:YLR337C	100	57	43
ORF:YPL249C-A	90	77	13
ORF:YCR028C-A	55	49	6
ORF:YCR020W-B	47	6	41
ORF:YHR039C-A	46	18	28
ORF:YLR438C-A	44	25	19
ORF:YBR089C-A	43	2	41
ORF:YFL017W-A	43	42	1
ORF:YKL053C-A	40	16	24
ORF:YBR058C-A	40	40	0
ORF:YER087C-B	38	12	26
ORF:YFR032C-A	34	31	3
ORF:YCR073W-A	33	29	4
ORF:YML081C-A	32	21	11
ORF:YOL077W-A	31	24	7
ORF:YDR320C-A	25	25	0
ORF:YKL138C-A	24	22	2
ORF:YFL034C-B	21	21	0
ORF:YCR087C-A	16	2	14
ORF:YIL017C	15	8	7
ORF:YBL071W-A	15	15	0
ORF:YEL020W-A	13	5	8
ORF:YKR035W-A	10	4	6

and reproducibly distinguished from noise. This may mean that even though tens of thousands of genes were measured in a microarray experiment,only obtain hundreds of genes that are reasonably convinced are involved in a particular biological system [52]. From the last section the dataset is quite large "6130 expression profiles" and a lot of the information corresponds to genes that do not show any interesting changes during the experiment. In this section we try to remove genes with expression profiles that do not show anything of interest. Below we use a number of techniques to reduce the number of expression profiles to some subset that contains the most significant genes.

3.2.2.1 Conditions Filtration

We deleted conditions that have more than 20% missed entities as it is recommended by the imputation technique which was used in section 3.2.3. The excluded experiments are 6,8 and 57, so the resultant dataset will be 6130 x 170.

3.2.2.2 Genes Filtration

In this step, We use the filtering functions in the Bioinformatics Toolbox[10] to remove genes with various types of profiles that do not provide useful information about genes affected by the metabolic change.

3.2.2.3 Remove Genes with Large Missed Values

Approximately 3% of the 6130 x 170 Gasch data were missed. Missing values occur for diverse reasons, including insufficient resolution, image corruption, or simply due to dust or scratches on the slide. Missing data may also occur systematically as a result of the robotic methods used to create them [53]. Genes which have more than third of its values were deleted, part of them are shown in Figure 3.1. The size of the resultant dataset after removing these genes is 6096x170.

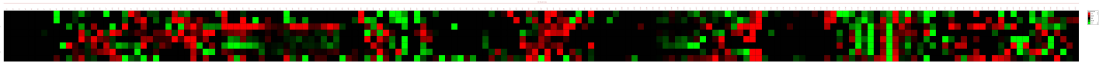


FIGURE 3.1: Part of the Removed Genes which Third of its Values Were Missed, The Dense Black Region Motivate Deletion These Genes.

3.2.2.4 Remove Genes with Small Profile Variance

From plotting the expression profiles of all the 6096 remaining profiles, we would see that some profiles are flat and not significantly different from the others. This flat data is obviously of use as it indicates that the genes associated with these profiles are not significantly affected by the multiple conditions and may still show variation due to measurement noise. If these genes are not filtered out, such measurement noise can be amplified by normalization and can appear as a true signal. The variance for each gene expression profile in data were calculated and then were deleted the expression profiles with a variance less than the 1th percentile. Figure 3.2 shows the expression profile of two genes (YFL014W and YNL020C) which have large and low variance sequentially. the gene function annotation of these genes are explained the large variance values of the YFL014W gene where it's function include the following [cell adhesion, cellular response to heat, hyperosmotic response, response to heat, response to oxidative stress, response to stress]. Comparing Gasch experiments conditions with these functions make more confidence why gene YFL014W has large variance value as also figure is described. Table 3.6 shows the properties of the part of deleted genes

[10] www.mathworks.com/

which indicate its important whereas they did not response significantly to Gasch experiments. Table 3.6 indicate the importance of design an experiments by including more conditions to target these genes in the further analysis. If the user interested in these deleted genes, he can search for suitable experiments from microarray tools like SPELL [11], Webminer [12],expressionConnection [13] and Rosetta Compendium [14]. These tools could identified which datasets are most informative for the user interested genes, then within those datasets additional genes are identified with expression profiles most similar to the query set.

Now after above filtration procedure the resultant dataset become 6035 x 170.

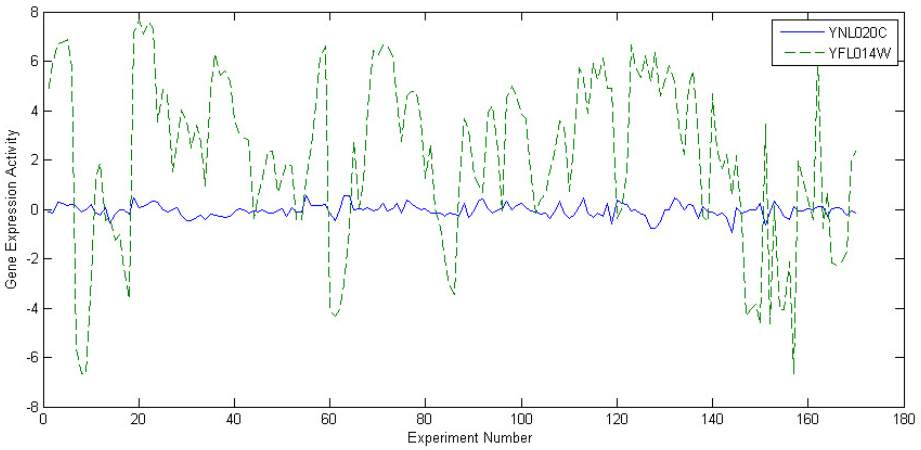


FIGURE 3.2: Gene Expression Activity of Two Genes have Large and Small Variance Values under Gasch Conditions[50].

3.2.2.5 Remove Genes with Low Absolute Values

In spite of removing genes with low variance, gene expression profiles of the remaining 6017 genes still have data where the absolute values are very low. The quality of this type of data is often bad due to large quantization errors or simply poor spot hybridization. It is commonly believed that low gene expression measurements are less reliable than high gene expression measurements. If the low expression levels are particularly noisy, this can cause artifacts in the downstream clustering process, or worse, cause the creation of incorrect clusters.

Figure 3.4 shows expression profiles of two genes (YHR139C & YHL020C) which they have large and small absolute values sequentially . Moreover Table 3.7 shows them properties which emphases the need of improving the quality of experiments to get full

[11] http://imperio.princeton.edu:3000/yeast/
[12] http://genome-www.stanford.edu/cgi-bin/webminer
[13] http://www.yeastgenome.org/cgi-bin/expression/expressionConnection
[14] http://www.rii.com

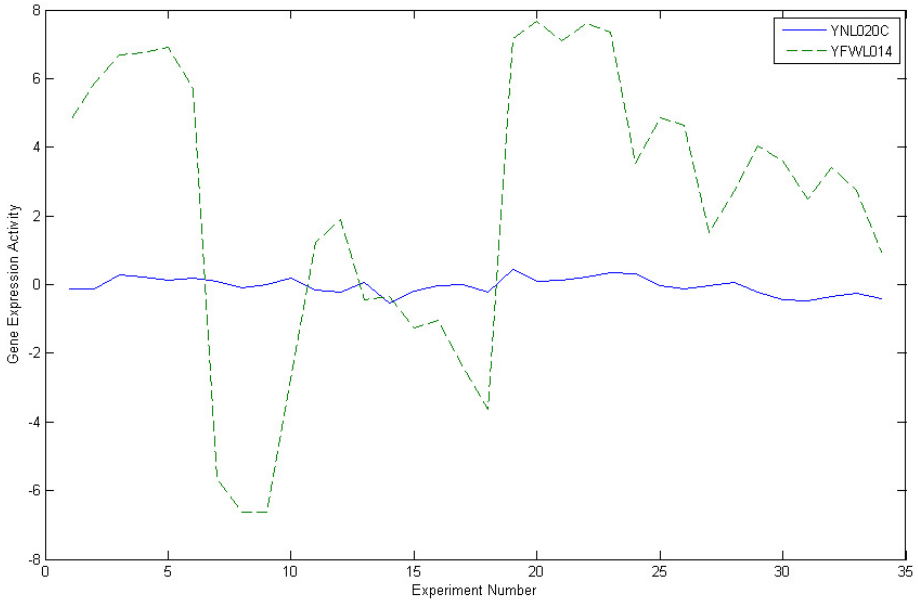


FIGURE 3.3: Gene Expression Activity for two genes in figure 3.2 under temperature stress conditions

TABLE 3.6: Properties of Part of Genes That Have Low Variance Expression Values Lower Than 1%

ORF	EdgeCount	Indegree	Outdegree
ORF:YDR477W	265	54	211
ORF:YLR373C	166	94	72
ORF:YIL084C	162	61	101
ORF:YDR207C	162	27	135
ORF:YOR304W	160	129	31
ORF:YGL025C	146	140	6
ORF:YBR095C	125	119	6
ORF:YKL139W	109	58	51
ORF:YCR033W	98	89	9
ORF:YBR289W	89	4	85
ORF:YAL032C	69	0	69
ORF:YGR186W	68	27	41

view of the story. On the other hand investigation Table 3.7 indicates that removing these genes will loss important genes in further process, but we believed that removing genes from further process will decrease the cost of the false positive when including these genes as described in Figure 3.5. Figure 3.5shows the trade off between the false positive and false negative cost from getting or neglecting genes from further process.

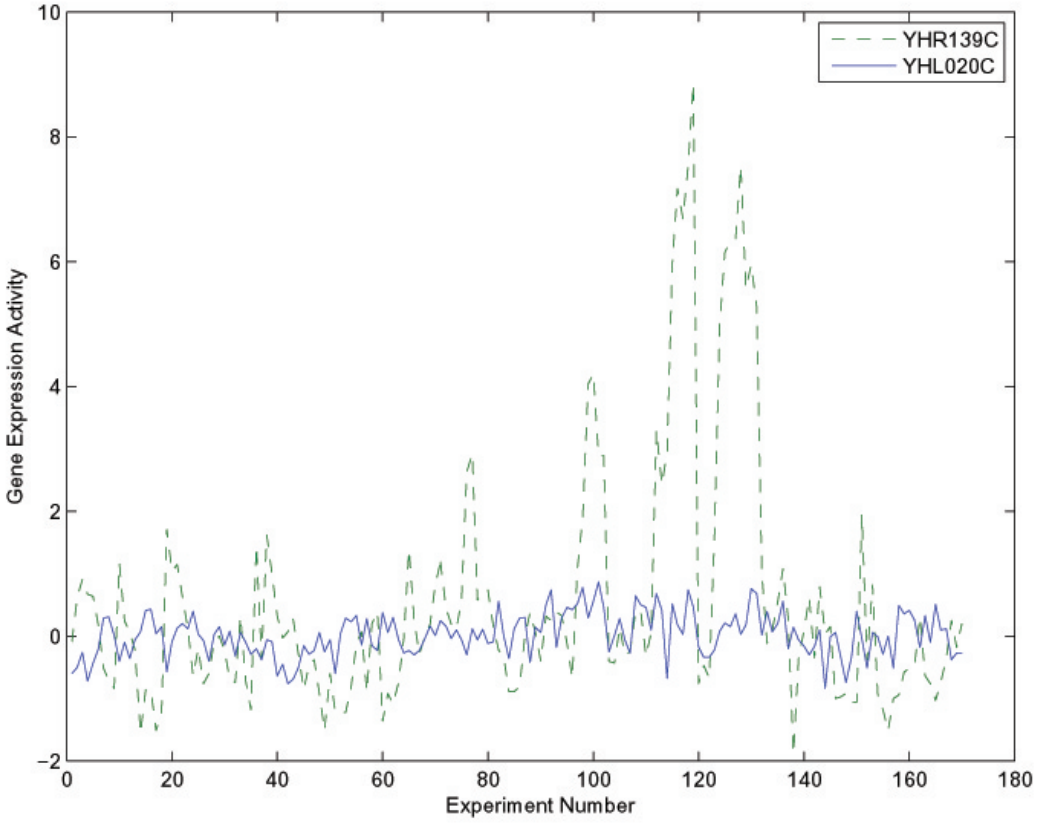


FIGURE 3.4: Gene Expression Activity of Two Genes have large and Low Absolute Values.

TABLE 3.7: Properties of Part of Genes That Have Low Absolute Values Lower Than $log_2(2)$

ORF	EdgeCount	Indegree	Outdegree
YLR085C	249	137	112
YPL269W	82	66	16
YOR297C	75	57	18
YDR363W	73	17	56
YLR399C	54	30	24
YKL089W	49	27	22
YGL172W	40	9	31
YAL030W	37	0	37

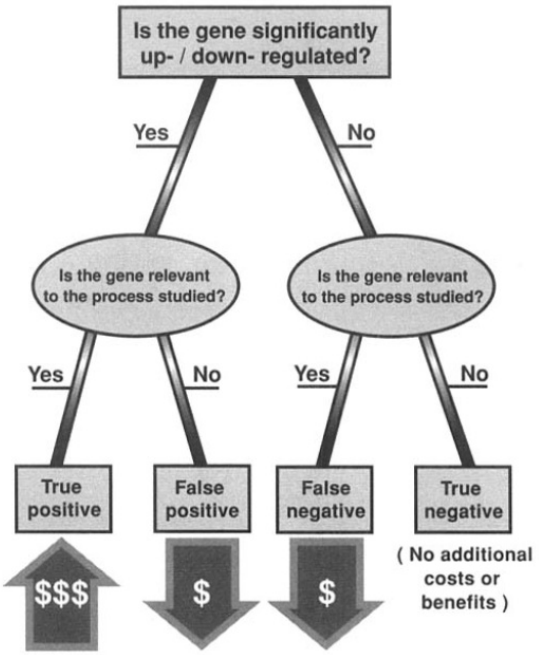


FIGURE 3.5: A Decision Analytic Procedure for Picking a Threshold For Selecting Genes for Further Process (copyrith @ [52])

At the conclusion of this filtering stage, expression values of 6017 ORFs at 170 time points remained.

3.2.2.6 Remove Genes with Low Entropy

Genes can demonstrate spiking behavior, where low expression levels are seen in all samples except one. The single high expression can dominate a pairwise analysis using correlation coefficients. Heyer et.al [54] study the effect of this spiky on increase the false positive of high correlation coefficient between unrelated genes as described in Figure 3.6.

There are two approach to remove genes with these spiky, First Heyer et al. [54] use of the jackknife correlation coefficient to counter the spiking problem. The jackknife correlation coefficient is an alternative dissimilarity measure to the standard (Pearson's) correlation coefficient. To compute this measure for two genes measured in n samples, the technique involves computing n different correlation coefficients, each time with one of the samples removed. The jackknife correlation coefficient is then the minimum of the separate correlation coefficients. Second an entropy filter can be used to remove genes that demonstrate spiking behavior,or, in other words, that are not well distributed over its range of values. Entropy is a measure of the amount of disorder in a variable which is defined by:

$$H(x) = \sum_i p(x_i) log_2(p(x_i))$$

where x is the variable whose entropy H is being calculated, log_2 is base 2 logarithm, and $p(x_i)$ is the probability a value of x was within quantile i of that feature. For example, if one were using 10 quantiles, and a gene with the expression amounts 20, 22, 60, 80 and 90 would have deciles 7 units wide, with two values in the first decile, one in the sixth decile, and one in the ninth and tenth decile, making $H = 1.92$ [52].

First, the entropies of the dynamics time series are calculated for each gene, with the above equation. Then genes are ranked according to their entropies, and the bottom 1% (entropy threshold ,6.67) are excluded from the analysis. Figure 3.7 display three genes which they have low entropy and table 3.8 shows connectivity properties of some of excluded genes.

At this stage, 5957 ORFs remained from the original 6130.

TABLE 3.8: Properties of Part of Genes Which Have Low Entropy Values Lower Than 1% (Entropy Threshold, 6.67).

ORF	EdgeCount	Indegree	Outdegree
ORF:YML032C	276	260	16
ORF:YLR320W	209	125	84
ORF:YCL016C	187	171	16
ORF:YGR252W	178	74	104
ORF:YOR039W	148	114	34
ORF:YPL008W	121	101	20

The Filtration steps and the resultant data set have described in this section are implemented in these link:

http://home.k-space.org/FADL/Downloads/PhD/Matlab_Function_code/datasetFilter.m

http://home.k-space.org/FADL/Downloads/PhD/Data/Filtered-data-ALL.xls

3.2.3 Imputation Missing Values

Many analysis methods,such as principle components analysis or singular value decomposition, require complete matrices. Of course, one solution to the missing data problem is to repeat the experiment. This strategy can be expensive, but has been used in validation of microarray analysis algorithms. Missing log_2 transformed data are often replaced by zeros or, less often, by an average expression over the row, or row average. This approach is not optimal, since these methods do not take into consideration the correlation structure of the data [53]. Olga Troyanskaya et.al [53] compare several methods (Singular Value Decomposition (SVD) based method (SVDimpute), weighted K-nearest

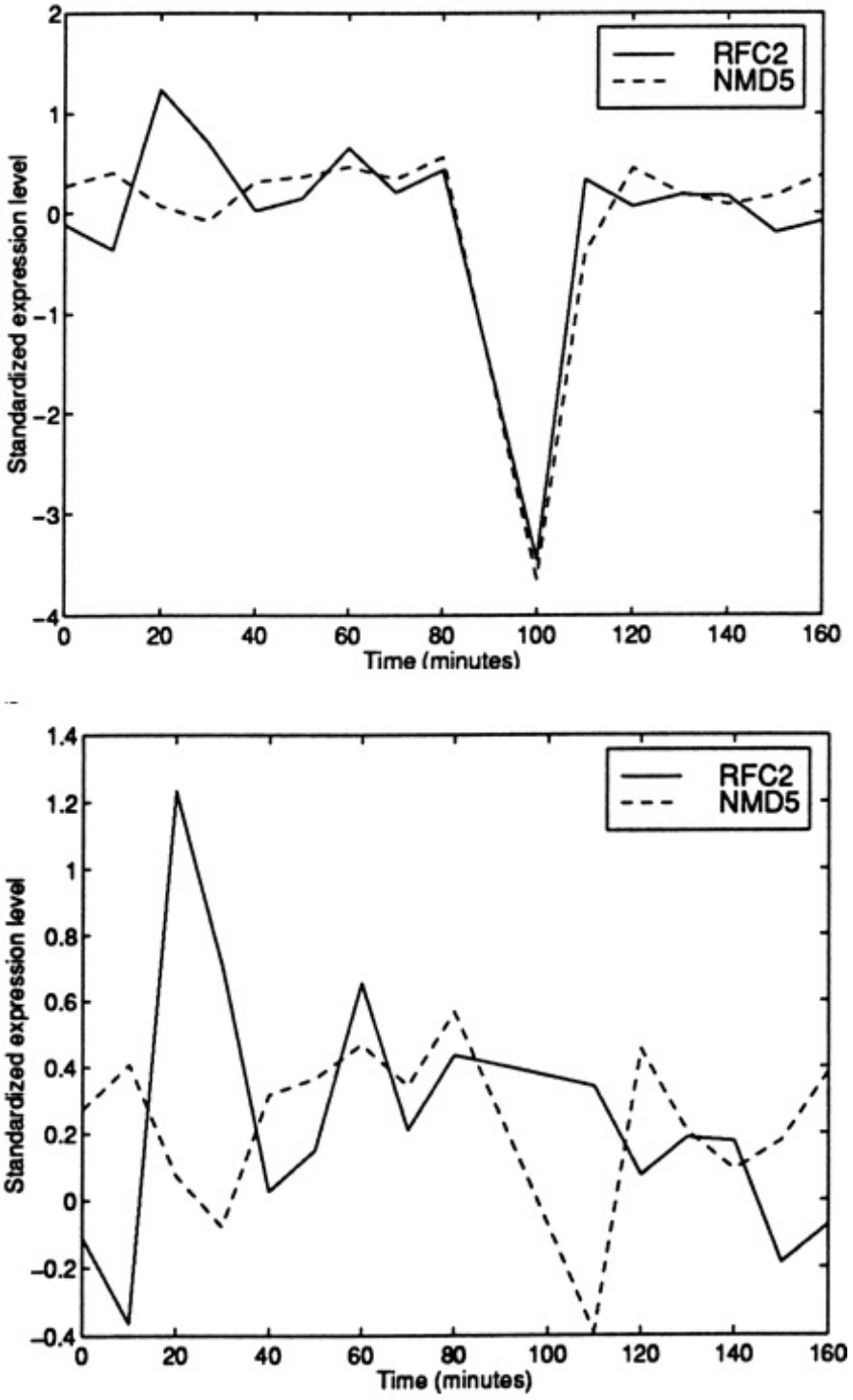


FIGURE 3.6: a) Expression data for YJR068W (RFC2) and YJR132W (NMD5). The gene pair has a correlation coefficient of 0.87. (b) Expression data for the same two genes with time 100 removed(spiky point). Using only the remaining points results in a correlation coefficient of -0.29. (Solid line) RFC2; (broken line) NMD5. (Copyright ©[54].

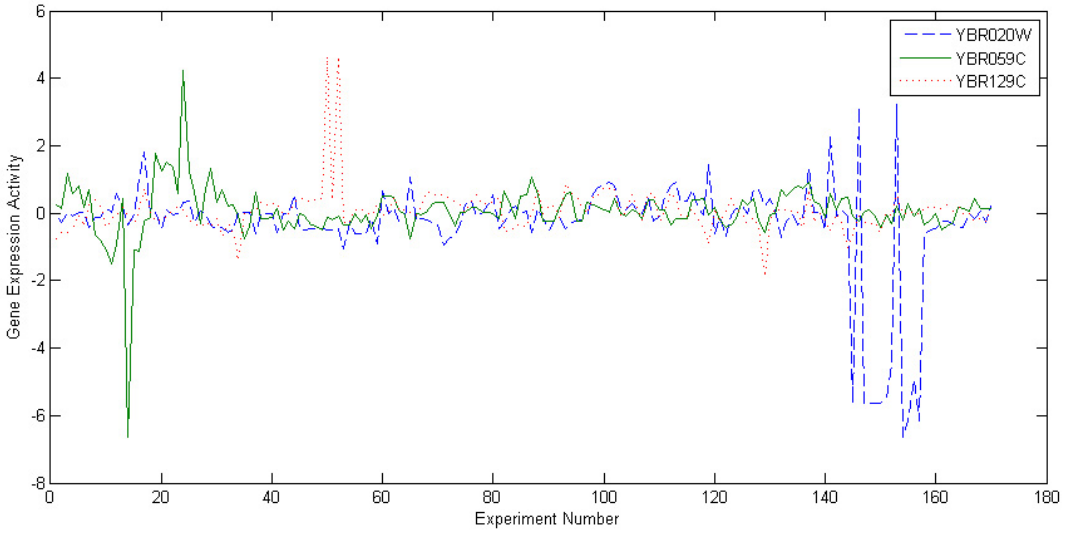


FIGURE 3.7: Example of Gene Expression Activity of genes with low entropy

neighbors (KNNimpute),and row average) for the estimation of missing values in gene microarray data. KNNimpute appeared to provide a more robust and sensitive method for missing value estimation. The KNN-based method selects genes with expression profiles similar to the gene of interest to impute missing values. If we consider gene A that has one missing value in experiment 1, this method would find K other genes, which have a value present in experiment 1, with expression most similar to A in experiments 2N (where N is the total number of experiments). A weighted average of values in experiment 1 from the K closest genes is then used as an estimate for the missing value in gene A. the percentage of the missed values after missing values imputation, was reduced to 2.7% which motivate us to use the KNNimpute algorithm implemented in

http://smi-web. stanford.edu/projects/helix/pubs/impute/.

3.2.4 Normalization

In an experimental context, normalisations are used to standardize microarray data to enable differentiation between real (biological) variations in gene expression levels and variations due to the measurement process. Since it is assumed that the cDNA microarray data is already normalized, as it is input after performing log ratio $log_2 \frac{R}{G}$, normalization was not performed on Gasch data. We rescaled the expression level of each gene, so that the relative expression of all genes have the same mean and variance.

3.2.5 Discertization

Some of biclustering and structure learning algorithms required the input data to be discredited. On other hand Florian et al [55] suggest that discretization of the continuous data leads to a large information loss. Di Camillo et.al [56] confirm that the use of discrete rather than continuous data is advantageous when few samples are available. Continuous approaches are likely to become advantageous with increasing number of samples. There are many discretization methods the simple one is to map gene expression to 0 and 1 by setting an appropriate threshold(see [56] for more details). We discretize gene expression values into three categories: -1,0 and 1 depending whether the expression rate is significantly lower than, similar to, or greater than the respective control as in [4].

3.2.6 Data Denoising

Many algorithms are found in the literature for data denoising. Among the most powerful techniques that can be used to separate signal components are those based on blind source separation such as principal component analysis (PCA) and independent component analysis (ICA) [25]. These techniques decompose the signal sources using either the second order statistics (as in PCA) or higher order statistics such as the kurtosis (as in ICA) to account for the non-Gaussian nature of the sources. According to the assumptions of both techniques, the number of independent signal components must be less than or equal to the number of signals to be analyzed. Otherwise, the separation of components yields incorrect results or even may not converge at all as in ICA. Unfortunately, this condition is not satisfied in microarray datasets. Given the general assumption of uncorrelated noise, the number of components of random noise alone is equal to the number of signals. The total number of components has to add the number of components. As a result, the use of PCA and ICA-based techniques may not be successful in practice unless the noise signals are sufficiently weak. This may account for the limited use of such techniques in low SNR applications [26]. Therefore, a technique that suppresses random noise or removes some of its components would be rather useful for making the use of PCA and ICA more robust for false positive reduction.

Yasser Kadah [57] developed a new denoising algorithm for noise suppression in event-related functional magnetic resonance imaging (fMRI) data and we applied it with the gene expression data. The proposed algorithm is an adaptive signal-preserving technique for gene expression data based on spectral subtraction. The block diagram of the proposed denoising method is shown in Figure 3.8.

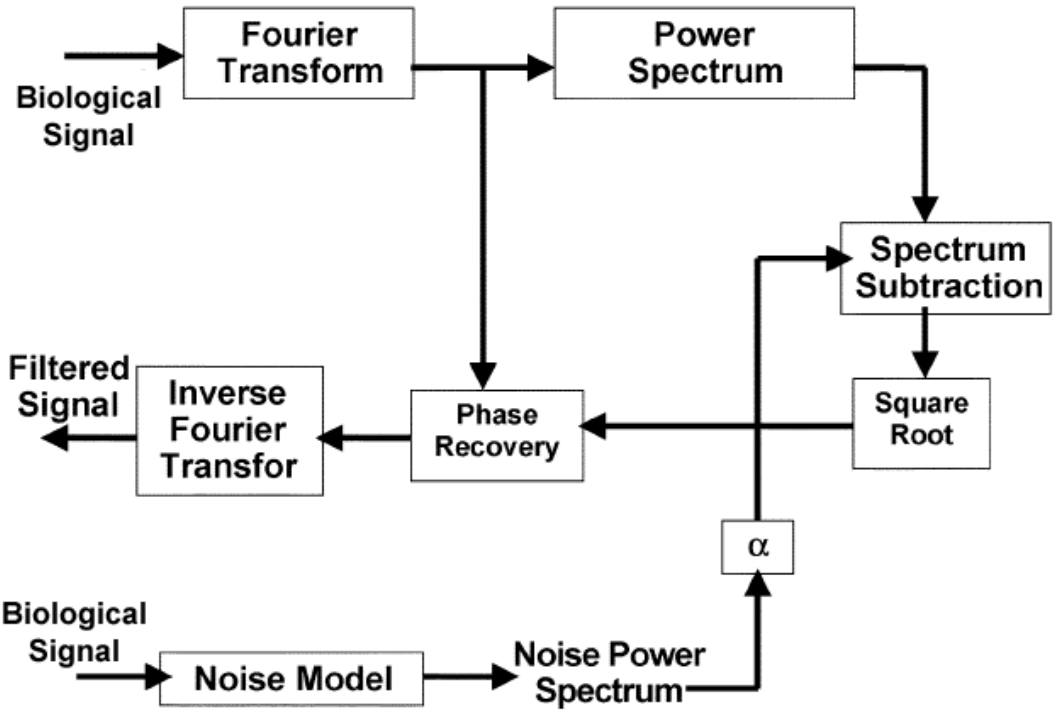


FIGURE 3.8: Spectral Subtraction Denoising Algorithm Block Diagram.

We will consider a model that is composed of the sum of one deterministic component $d(t)$ incorporating both the true gene expression signal and the experimental noise and an uncorrelated stochastic component $n(t)$. That is

$$s(t) = d(t) + n(t)$$

Since these two component are assumed to be independent, the corresponding power spectra are related by

$$P_{ss}(w) = P_{dd}(w) + P_{nn}(w)$$

where cross terms vanish because the two components are assumed uncorrelated. Hence, an estimate of the power spectrum of the deterministic component takes the form [58]

$$P_{dd}(w) = P_{ss}(w) - P_{nn}(w)$$

That is, the signal power spectrum is obtained by spectral subtraction of the noisy signal and noise power spectra. In order to compute the deterministic signal component from its power spectrum, the magnitude of the Fourier transform can be obtained as the square root of the power spectrum. The problem now becomes that of reconstructing the signal using magnitude only information about its Fourier transform. Several techniques can be used to do that. The one used here relies on an estimate obtained from the phase of the Fourier transform of the original signal $S(w)$. Hence, the Fourier transform of the processed signal $S_d(w)$can be expressed as

$$S_d(w) = \sqrt{P_{dd}(w)}.e^{jphase(S(w))}$$

The enhanced deterministic signal $s_d(t)$ is then computed as the real part of the inverse Fourier transformation of this expression.
To test the performance of SS denoising technique we applied this algorithm to DREAM3 In Silico Network Challenge4 [59]. In this challenge, the expression data(knock out and perturbation)obtained from a synthetic 10-gene network in yeast and Ecoli were provided. This allows the inference of a GRN for which the true network structure is known. Figure 3.9 shows the result of data denoising using spectral subtraction (third column) and Multi-Wavelet(fourth column) with the original signal(first column). It is clear from Figure 3.9 that the spectral subtraction denosing method outperforms the Multi-Wavelet method. The prediction error of Spectral Subtraction and Multi-Wavelet algorithms are 0.088 and 0.13 respectively.
The new strategy based on spectral subtraction method is adaptive and simple to implement while offering a substantial improvement of the SNR. Very few assumptions about the nature of the noise model and no assumptions about the deterministic signal components are made.
The Matlab code of SS denoising method with DREAM3 data could be downloaded from
http://home.k-space.org/FADL/Downloads/PhD/RECOMB_paper/spectral_subtraction_denoising

3.3 Data Partitioning: Clustering Algorithms

Detecting groups (clusters) of closely related objects is an important problem in bioinformatics and data mining in general. Laboratories apply every existing clustering method to their microarray data sets, hoping to find some significant genes or clusters[52]. In this section we first give basic background on clustering. We then describe two clustering algorithms used for gene expression analysis.

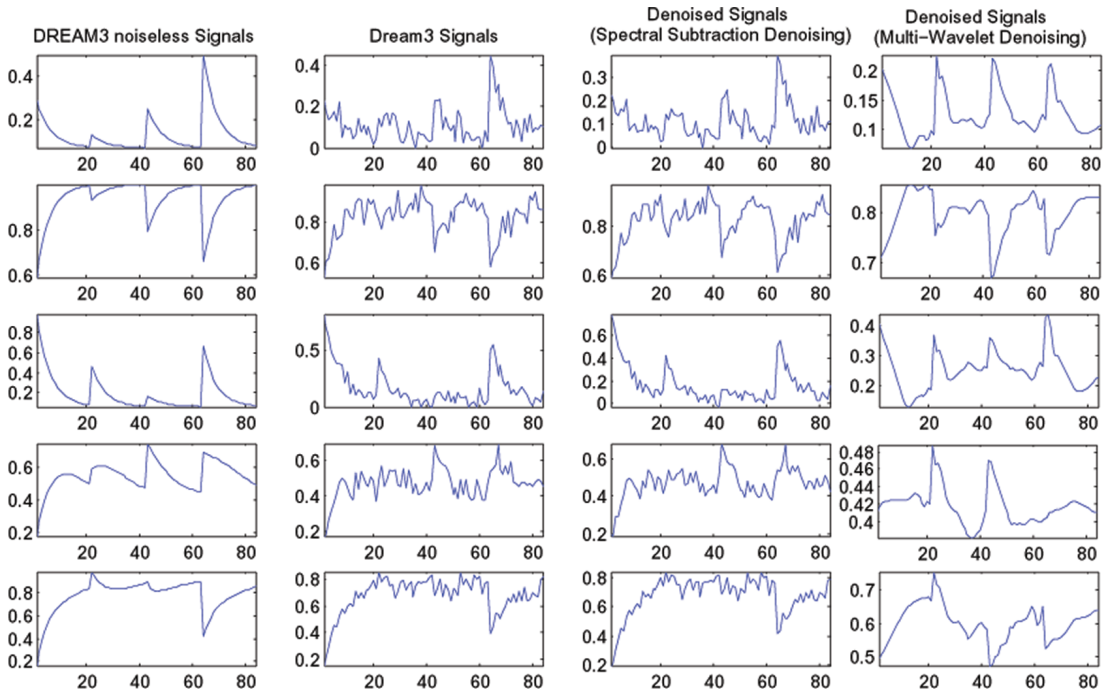


FIGURE 3.9: **Comparison of Spectral Subtraction and Multi-Wavelet Denosing Algorithms** Denoising of the DREAM3 In-Silico Network Challenge4 Signals, drawn represent gene expression time serious data a long 84 time points. Rows represent gene signals(shown 5 genes); columns represents Dream3 challenge 4 data as following: DREAM3 noiseless data (First Column);DREAM3 submitted data(Second Column); DREAM3 denoised data using Spectral Subtraction(Third Column); DREAM3 denoised data using Multi-Wavelet(Fourth Column). This Figure showed the high accuracy of Spectral Subtraction over Multi-Wavelet.

3.3.1 What is Clustering?

A large number of clustering definitions can be found in the literature. The simplest definition is shared among all and includes one fundamental concept: the grouping together of similar data items into clusters [60].

Clustering is an important explorative statistical analysis of gene expression data. It aims to identify and group genes that exhibit similar expression patterns over several conditions and also group the conditions based on the expression profiles across set of genes. The successful clustering approach should guarantee two criteria which are homogeneity high similarity between elements in the same cluster, and separation - low similarity between elements from different clusters. When homogeneity and separation are precisely defined, those are two opposing objectives: The better the homogeneity the poorer the separation, and vice versa [61]. Several algorithmic techniques were previously used for clustering gene expression data, including hierarchical clustering [62], self organizing maps [63], and graph theoretic approaches [64]. For more extensive

reviews and more information and background on clustering, see [65].

3.3.2 K-means

K-means is a classical clustering algorithm [9]invented in 1956 to classify or to group objects (genes) based on attributes or features (experimental conditions) into K number of groups (clusters). K is positive integer number and assumed to be known. Kmeans computational approach starts by placing K points into the space represented by the objects that are being clustered. These points represent initial group centroids. We can take any random objects as the initial centroids or the first K objects in sequence can also be used as the initial centroids. Then the K means algorithm will do the four steps below until convergence:

1. Determine the centroids coordinate.
2. Determine the distance of each object to the centroids using the Euclidean distance which is defined as: $\sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + (p_n - q_n)^2} = \sqrt{\sum_{i=1}^{n}(p_i - q_i)^2}$
   Where p is the object (gene expression) value of i condition, q is centroid point value of i condition and n is the total number of conditions.
3. Group the objects based on minimum distance.
4. Iterate the above steps till no object moves its assigned group.

Each iteration of k-means modifies the current partition by checking all possible modifications of the solution, in which one element is moved to another cluster. This is done by reducing the sum of distances between objects and the centers of their clusters. This procedure is repeated until no further improvement is achieved (No object move the group) and all the objects are grouped into the final required number of clusters. A disadvantage of K-means algorithm could be perceived in the need to specify the number of clusters K as a parameter value prior to running the algorithm. In cases where there is no expectation about K, user has to make trails with several values of K or use external techniques to guess the no of clusters may be exist.

3.3.3 Hierarchical clustering (HCL)

Hierarchical clustering does not partition the genes into subsets. Instead it builds a down-top hierarchy of clusters using agglomerative methods or top - down hierarchy of clusters using divisive methods. The traditional graphical representation of this hierarchy is called dendrogram tree. The divisive method begins at the root and starts to breaks up clusters whose having low similarity. Whereas, the Agglomerative method begins at the leaves of the tree and starts with an initial partition into single element clusters and successively merges clusters until all elements belong to the same cluster [66]. (See Figure 3.10) The agglomerative method is widely used than

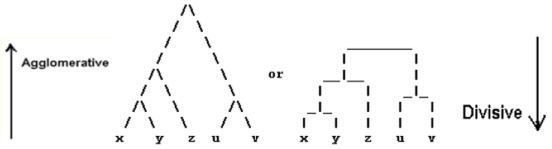


FIGURE 3.10: HCL:Agglomerative and Divisive Methods.

the divisive one which is not generally available, and rarely has been applied. The idea of the agglomerative method can be summarized as following: Given a set of N items (genes in our case) to be clustered, and an N*N distance (or similarity) matrix [67],

1. Assign each item to a cluster, so you have N clusters, each containing just one item.
2. Find the closest (most similar) pair of clusters and merge them into a single cluster.
3. Compute distances (similarities) between the new cluster and each of the old clusters.
4. Repeat steps 2 and 3 until all items are clustered into a single cluster of size N.

In Step 3, distance or similarity measurements between the merged clusters and all the other clusters can be calculated in one of three schemes: single-linkage, complete-linkage and average-linkage.

3.4 Biclustering Algorithms

Traditional clustering approaches such as k-means and hierarchical clustering put each gene in exactly one cluster based on the assumption that all genes behave similarly in

all conditions. However, recent understanding of cellular processes shows that it is possible for subset of genes to be co expressed under certain experimental conditions, and at the same time; to behave almost independently under other conditions. From this context, a new two mode clustering approach called biclustering or co-clustering has been introduced to group the genes and conditions in both dimensions simultaneously. This allows finding subgroups of genes that show the same response under a subset of conditions, not all conditions. Also, genes may participate in more than one function, resulting in one regulation pattern in one context and a different pattern in another. Example, if a cellular process is only active under specific conditions and there is a gene participates in multiple pathways that are differentially regulated, one would expect this gene to be included in more than one cluster; and this cannot be achieved by traditional clustering techniques.

Many biclustering methods exist in the literature [68]. Table 3.9 summarized some of promising biclustering algorithms developed during the last ten years. In brief we described some of these algorithms according to their prediction strength, their promising results, to what they extend in the community, whether an implementation was available, and the feedback from their authors to explain some ambiguous issues.

Table 3.9: Biclustering Algorithms Comparison

Algorithm	Approach	Time Complicity	Prediction ability
Bivisu [69]	Exhaustive Bicluster Enumeration	$O(m^2nlogm)$[a]	Coherent values
MSBE [70]	Greedy Iterative Search	$O((n+m)^2)$	Coherent values
Bimax[15]	Divide-and-Conquer	$O(nm\beta log\beta)$	Coherent values
ROBA [71]	Matrix algebra	$O(nmLN_b)$	Coherent Evolution
x-motif [20]	Greedy Iterative Search	$nm^{O(log(1/\alpha)/log(1/\beta))}$	Coherent Evolution
SAMBA [72]	Exhaustive Bicluster Enumeration	$O(n2^d)$	Coherent Evolution
OPSM [19]	Greedy Iterative Search	$O(nm^3I)$	Coherent Evolution
Plaid [73]	Distribution Parameter Identification	XXX[b]	Coherent values
ISA [21]	Iterative Signature Algorithm	XXX	Coherent values
CC [11]	Greedy Iterative Search	O((n + m)nm),	Coherent values

[a] n and m are the row and column sizes of the expression matrix
[b] not available

3.4.1 Cheng and Church (CC)

CC algorithm [11] is considered to be the first real biclustering implementation after the primary idea has been introduced by Hartigan [74] in 1972.

CC defines a bicluster as a subset of rows and a subset of columns with a high similarity. The proposed similarity score is called mean squared residue (H) and it is used to measure the coherence of the rows and columns in the single bicluster. Given the

gene expression data matrix A = (X;Y); a bicluster is defined as a uniform submatrix (I;J) having a low mean squared residue score as following:
The CC Mean Squared Residue:

$$H(I,J) = \frac{1}{\|I\|\|J\|} \sum_{i\in I, j\in J} (a_{ij} - a_{iJ} - a_{Ij} + a_{IJ})^2$$

Where: a_{ij} is gene expression level at row i and column j, a_{iJ} is the mean of row i, a_{Ij} is the mean of column j, a_{IJ} is the overall mean. CC algorithm will identify the submatrix as a bicluster if the score is below a level alpha which is a user input parameter to control the quality of the output biclusters. Generally; CC algorithm performs the following major steps:

1. Delete rows and columns with a score larger than alpha.
2. Adding rows or columns until alpha level is reached.
3. Iterate these steps until a maximum number of biclusters is reached or no bicluster is found [11].

3.4.2 Iterative Signature Algorithm (ISA)

The ISA algorithm [21, 22] is a novel method for the biclustering analysis of large-scale expression data. It is an efficient algorithm based on the iterative application of the signature algorithm presented in [21]. ISA considers a bicluster to be a transcription module which can be defined as a set of coexpressed genes together with the associated set of regulating conditions(Figure 3.11). Starting with an initial set of genes, all samples (conditions) are scored with respect to this gene set and those samples are chosen for which the score exceeds a certain threshold (usually defined by the user). In the same way, all genes are scored regarding the selected samples and a new set of genes is selected based on another user-defined threshold. The entire procedure is repeated until the set of genes and the set of samples converge and do not change anymore. Multiple biclusters can be discovered by running the ISA algorithm on several initial gene sets. This approach requires identification of a reference gene set which needs to be carefully selected for good quality results. In the absence of pre-specified reference gene set, random set of genes is selected at the cost of results quality [21].

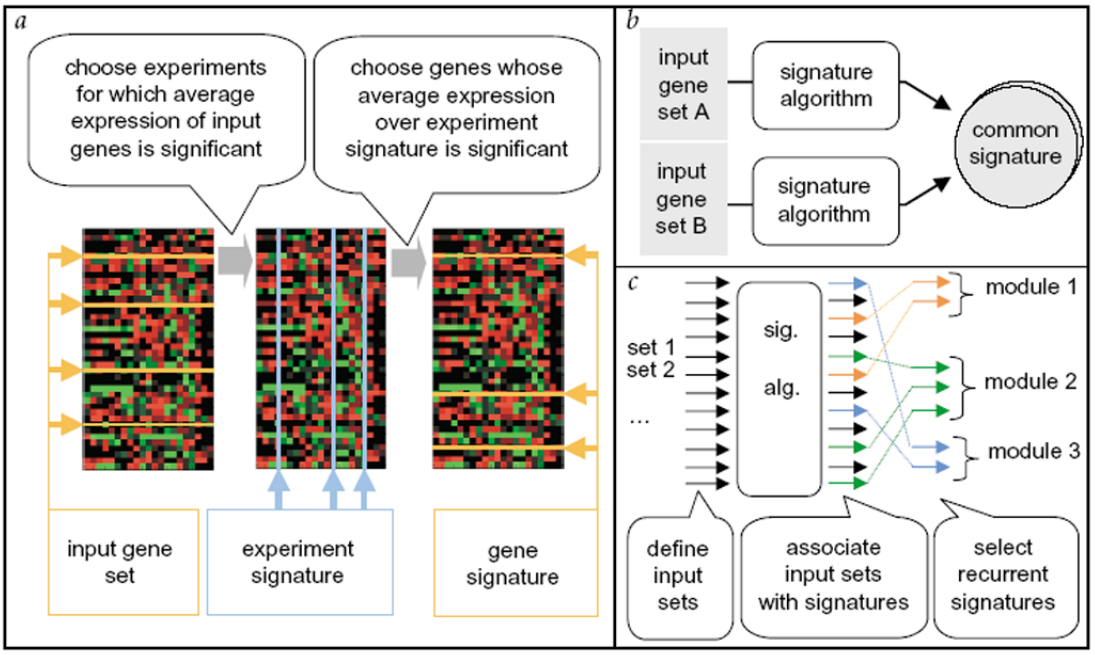


FIGURE 3.11: The recurrence signature method. a, The signature algorithm. b, Recurrence as a reliability measure. The signature algorithm is applied to distinct input sets containing different subsets of the postulated transcription module. If the different input sets give rise to the same module, it is considered reliable. c, General application of the recurrent signature method.Copyright © [21].

3.4.3 Biclusters Inclusion Maximal (Bimax)

Bimax [15] is a simple binary model and new fast divide-and-conquer algorithm used to cluster the gene expression data. It is presented in 2006 by Computer Engineering and Networks Laboratory ETH Zurich, Switzerland. Bimax discretized the gene expression data matrix and convert it into a binary matrix by identifying a threshold, so transcription levels (genes expression values) above this threshold become ones and transcription levels below become zeros (or vice versa). Then, it searches for all possible biclusters that contain only ones. This can be done by iterating these steps:

1. Rearrange the rows and columns to concentrate ones in the upper right of the matrix.
2. Divide the matrix into two sub matrices.
3. Whenever in one of the submatrices only ones are found, this sub matrix is returned.

So up or down-regulated constant biclusters are found. In order to get satisfying results the above steps have to be restarted several times with different starting points [15]

3.4.4 Order Preserving Submatrix(OPSM)

The order preserving submatrix (OPSM) algorithm [19] is a probabilistic model introduced to discover a subset of genes identically ordered among a subset of conditions. It focuses on the coherence of the relative order of the conditions rather than the coherence of actual expression levels. In other words, the expression values of the genes within a bicluster induce an identical linear ordering across the selected conditions. Accordingly, the authors define a bicluster as a subset of rows whose values induce a linear order across a subset of the columns. The time complexity of this model is $O(nm^3I)$ where n and m are the number of rows and columns of the input gene expression matrix respectively and I is the number of biclusters. A disadvantage of OPSM algorithm is that it takes long time for high dimensional datasets. And this is because its time complexity is cubic with regards to the number of columns (dimensions) of the input matrix [19].

3.4.5 Maximum Similarity Bicluster(MSBE)

MSBE Biclustering algorithm [70] is a novel polynomial time algorithm to find an optimal biclusters with the maximum similarity. The idea behind this algorithm is to find subset of genes that are related to a reference gene. The reference gene is known in advance. MSBE algorithm uses the similarity score for a sub-matrix to find the similar expressions in the microarray datasets. And the threshold of the average similarity score is a user input parameter in order to allow the user to control the quality of the biclustering results.

3.5 AGO:Automatic Gene Ontology Software Tool for Bicluster and Cluster Comparisons

The analysis of microarrays data poses a large number of exploratory statistical aspects including clustering and biclustering algorithms which help to identify similar patterns in gene expression data and group genes and conditions in to subsets that share biological significance.
During the last year, more than ten biclustering algorithms have been proposed(Table 3.9), but the question is: which algorithm is better? And do some algorithms have advantages over others? Generally, comparing different biclustering algorithms is not

straightforward as they differ in strategies, approaches, time complexity, number of parameters and prediction ability. They are strongly influenced by user-selected parameter values. For these reasons, the quality of biclustering results is also often considered more important than the required computation time. Although there are some analytical comparative studies to evaluate the traditional clustering algorithms [12–14], for biclustering; no such extensive comparison exist even after initial trails have been taken [15]. At the end, biological merit is the main criterion for evaluation and comparison between the various biclustering methods.

To our best knowledge, biclustering algorithms compassion toolbox has not been available in the literature. So, we have developed a comparative tool Automatic Gene Ontology (AGO)[15] [29] that includes the biological comparative methodology. The Goal of AGO is to enable researchers and biologists to compare between the different bi/clustering methods based on set of biological merits and draw conclusion on the biological meaning of the results. Also AGO help researchers in comparing and evaluating the algorithms results multiple times according to the user selected parameter values as well as the required biological perspective on various datasets.

AGO paper, program, help file and supplementary data could be downloaded from: http://home.k-space.org/FADL/Downloads/PhD/AGO_paper

3.5.1 Comparison Methodology

Internal indices such as homogeneity and separation have not been suggested for biclustering methods [75]. This is because these indices did not consider matching in two direction(genes and conditions). For example, if we have two biclusters contain the same genes set and they differ in conditions set. Even the two biclusters look bad at genes dimension for the separation index, they have good separation distance in the condition dimension. For this reason external indices are used to assess the methods under consideration as in most biclustering papers [15]. Also Gat-Viks et al. [75] and Handl et al. [76] recommend external indices for evaluating the performance of (bi)clustering methods.

There are four external indices in-order to test the enrichment of the bicluster via available databases. We said the bicluster is enriched if one of following has strong statistical evidence as following:

[15] F. M. Al-Akwaa and Y.M. Kadah. Automatic gene ontology software tool for bicluster and cluster comparisons. In IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology, Nashville, TN, USA, 2009. IEEE Computational Intelligence Society.

- The bicluster is enriched if its genes or some of them share the same function. This could be tested by applying the hypergeometric test of bicluster genes to the Gene Ontology (GO) data base.
- The bicluster is enriched if its genes or some of them participate in the same pathway. This could be tested by applying the hypergeometric test of bicluster genes to the Kyoto Encyclopedia of Genes and Genomes (KEGG) data base.
- The bicluster is enriched if its genes products(proteins) or some of them have biological interactions. This could be tested by analyzing the Protein Protein Interaction data base like BIOGRID.
- Bicluster is enriched if the promoter region of its genes and some of them have a conservative motif. This could be done by aligning the 50-100 base pair of genes DNA sequence upstream region.

Because PPI and KEGG databases are still incomplete, hypergometric test using GO data base is still the meaningful tool for biclustering comparison.
We have to define many important terms for comparing biclusters:

- The percentage of enriched or overrepresented biclusters This percentage is calculated for each algorithms with one or more GO term per multiple significance levels (p-values) for each algorithm using the below equation:

$$Percentage\ of\ Enriched\ Biclusters = \frac{Number\ of\ Enriched\ Biclusters}{Total\ Number\ of\ Biclusters} \text{ x } 100$$

- Percentage of annotated genes per each bicluster
  Some times even the bicluster is enriched, it contains few annotated genes. So we defined the percentage of annotated genes per each bicluster as more specific comparison metric as following:

$$Percentage\ of\ Annotated\ Genes\ per\ Each\ Bicluster = \frac{No\ of\ Genes\ Sharing\ GO{-}Term\ in\ aBicluster}{Total\ Number\ of\ Genes\ in\ this\ Bicluster}$$

3.5.2 Gene Ontology

In the last five years, biologists faced a problem of annotating the completed genome sequences especially for the Drosophila and the *S.cerevisiae* species and the organizations of the complex databases start to provide their own classification terminologies.
Consequently, these wide variations in terminologies and annotations inhibit effective searching by both computers and people [42]. For example, if biologist was searching for

new targets for antibiotics, If one database describes these molecules as being involved in 'translation', whereas another uses the phrase 'protein synthesis', it will be difficult for biologist and even harder for a computer to find functionally equivalent terms. Therefore formal and explicit specifications of the gene annotation terms (in the shape of well-structured and controlled vocabularies) used and the relationships between them have been defined [16]. This is called Gene Ontology and referred as GO. Using GO, biologists and researchers have systematic consistent classification of genes functions, in the form of a dictionary of functional terms that are hierarchically structured to allow both attribution and querying at different levels of granularity (See Figure 3.12). The

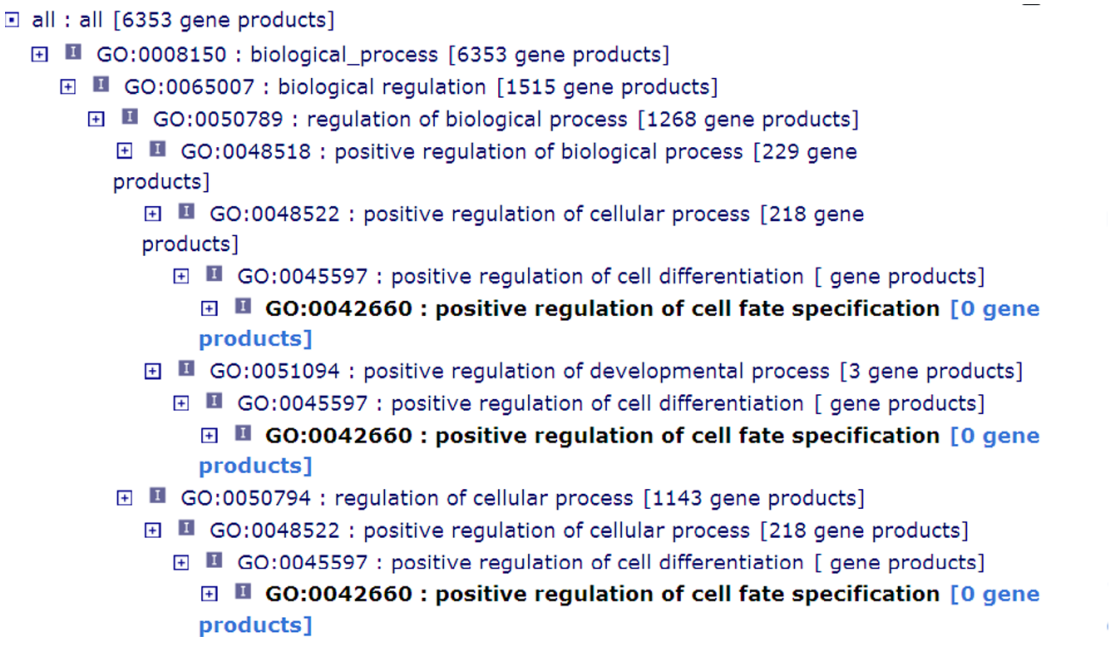


FIGURE 3.12: Tree view of Biological Process Gene Ontology Category of *S.cerevisiae*.

building blocks of the Gene Ontology are the terms (sometimes called functional classes or functional categories). Each GO term has a unique number and a textual name. E x, GO: 0042660: positive regulation of cell fate specification. Each GO term is assigned to one of the three subontologies(Figure 3.13) in GO: biological process, molecular function and cellular component.

1. Biological process(GO:0008150): A function represented in a series of events and activities of a living system, mediated by protein or RNA.

2. Molecular function(GO:0003674): A function associated with the biochemical activity (including specific binding to ligands or structures) of a gene product.

[16]http://www.geneontology.org/GO.tools.shtml.

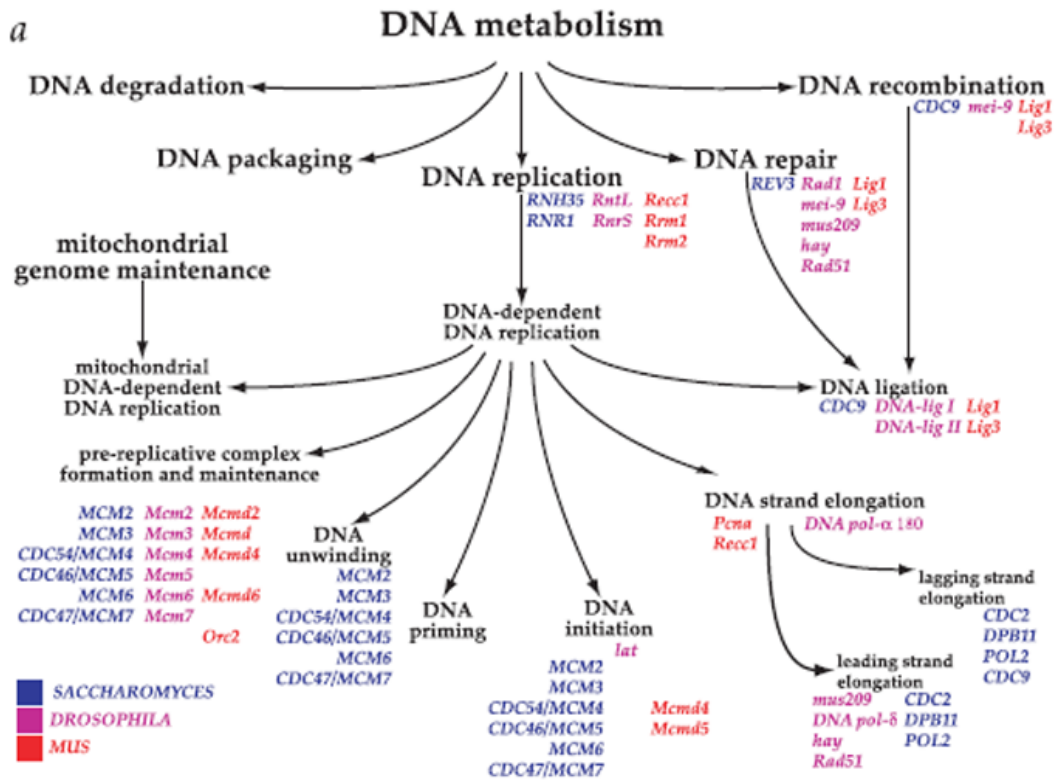


FIGURE 3.13: Example of Gene Ontology to Illustrate the Structure and Style used by GO to Represent the Gene Ontologies and to Associate Genes with Nodes within an Ontology(Copyright © [77]).

3. Cellular component(GO:0005575): A function refers to the place in the cell where a gene product is active. It can be a general term such as nucleus or a specific term such as ribosome.

Particularly, The GO project is a collaborative work across many laboratories and controlled by the gene ontology Consortium (set of model organism and protein databases and biological research communities actively involved in the development and application of the Gene Ontology) [77].

3.5.3 Hypergometric Test

If the bicluster we want to test its enrichment contains genes like $[g_1, ...g_n]$. The enrichment question is like this: Are there any GO terms that have a larger than expected subset of our bicluster genes in their annotation list? If so, these GO terms will give us insight into the functional characteristics of our bicluster. The hypergeometric test calculates the probability of drawing r genes with a certain GO function from a sample

of size k from a population of size n given that this GO function exists in fraction p in the population set of genes. The basic question answered by hypergometric test is as described by Steven et.al [78]

> when sampling X genes (test set) out of N genes (reference set, either a graph or an annotation), what is the probability that x or more of these genes belong to a functional category C shared by n of the N genes in the reference set?.

The hypergeometric test, in which sampling occurs without replacement, answers this question in the form of P-value. Its counterpart with replacement, the binomial test, which provides only an approximate P-value, but requires less calculation time. More details about hypergeometric test and its software implementation can be found in [79, 80].

3.5.4 GO Enrichment Programs

There are various tools (web based and standalone applications) introduced to analyze GO term enrichment in a given genes set. Some of these tools have been developed by the GO Consortium such as AmiGO and OBO-Edit, while other tools have been developed outside the GO Consortium for use with GO ontologies such as BiNGO [78], GeneMerge [81], GOEAST [80] and FuncAssociate [82]. A comprehensive list of all these tools can be found at GO website [17].
The shortcoming of these programs is that you should to enter each bi/cluster manually and then count the enriched and unriched clusters , which is consuming time and hard to do manually. AGO was proposed to overcome all of these shortcomings as described in the following sections.

3.5.5 AGO Implementation

We test AGO on a desktop PC with P4 1.8G CPU and 2.0 G memory running windos XP operating system and Matlab 7.2.
AGO block diagram is shown in Figure 3.14. First, As illustrated in this figure AGO input are the biclustering output files, which contains the biclusters results from one of available biclustering toolbox like BicAT toolbox [17],Bivisu program [69], MSBE

[17] http://www.geneontology.org/GO.tools.shtml.

package [70]). Second, function enrichment was analyzed for each biclusters/clusters using GeneMerge Perl program [81] by setting sufficient significance level and interested GO category. Third, As the number of generated biclusters varies strongly among the considered methods, a postprocessing filtration procedure, has been applied to the output of the algorithms to provide a common basis for the comparison. Finally, Using one of comparison methodology which were implemented in AGO, the user could test the performance of various algorithms.

AGO provides reasonable methods for comparing the results of different biclustering

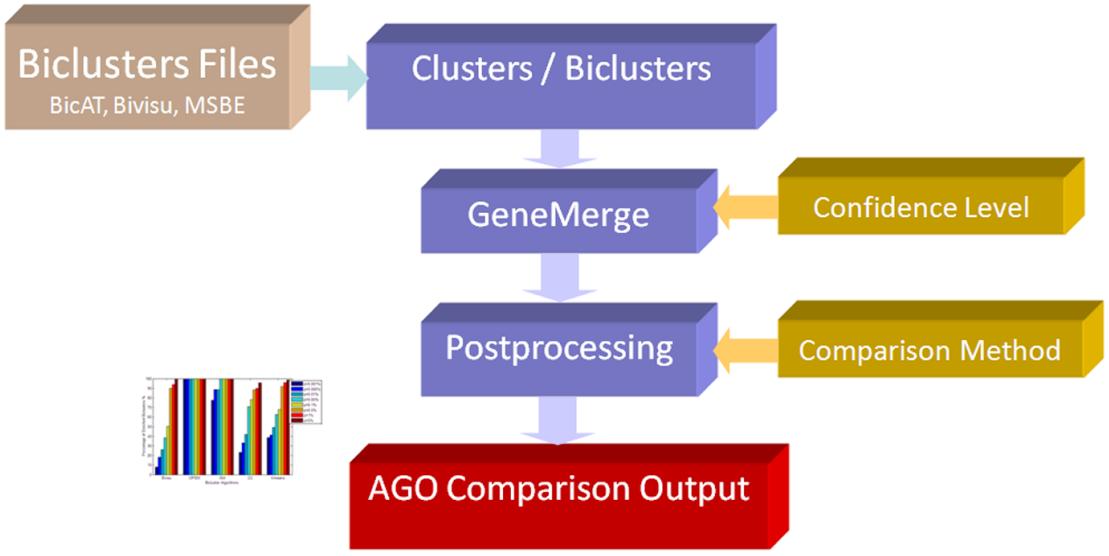


FIGURE 3.14: Blook diagram of the AGO.

algorithms by:

1. Identifying the percentage of enriched or overrepresented biclusters with one or more GO term per multiple significance levels for each algorithm. A bicluster is said to be significantly overrepresented (enriched) with a functional category if the p-value of this functional category is lower than the preset threshold P-value. The results are displayed using a histogram for the entire compared algorithms at the different preset significance levels, and the algorithm which gives higher proportion of enriched biclusters per all significance levels is considered to be the optimum one as it does group effectively the genes sharing similar functions in the same bicluster.

2. Identifying the percentage of annotated genes per each enriched bicluster.

3. Estimating the algorithms predictability power to recover interesting pattern. Genes whose transcription is responsive to a variety of stresses have been implicated in a general yeast response to stress. Other gene expression responses appear to be specific to particular environmental conditions. AGO Compare biclustering methods based on which of them could recover known patterns in experimental datasets. For example in Gasch et al [27] measure changes in transcript levels over time responding to panel of environmental changes. So it was expected to find enriched biclusters with one of response to stress (GO:0006950) Gene Ontology category like response to heat (GO:0009408), response to cold (GO:0009409) and response to glucose starvation(GO:0042149).

3.5.6 AGO Testing: Case Study

To test AGO, we run biclustering algorithms on the gene expression data of *S. cerevisiae* provided by Gasch et al [27]. The dataset contains 2993 genes and 173 conditions of diverse environmental transitions such as temperature shocks, amino acid starvation, and nitrogen source depletion.
Table 3.10 shows the biclustering algorithms parameters setting as authors recommended in their corresponding publications. There are three type of parameters. First, Parameters recommend by author which we could not alter it as in the previous publication [15, 21, 70]. Second, parameters depend on the data itself like noise threshold which equals data Standard Devision [69]. Third, parameters alter number of generated biclusters [11, 15] and biclusters size (min number of genes per each bicluster [15, 83]).
Table 3.11 demonstrates the statistical comparison of the biclusters output for each algorithm. They differ in the number of bicluster outputs, the number of genes and conditions within each bicluster and the ability to recover genes and conditions within its biclusters.

Comparing these algorithms using the percentage of the enriched biclusters histogram is shown in Figure 3.15. By comparing Figure 3.15 and Figure 3 in [15, 70], we found that the percentages of enriched biclusters for the matched algorithms are almost the same. This does validate the results of the proposed comparative tool. Investigating both figures, we observed that OPSM algorithm gave a high portion of functionally enriched biclusters at all significance levels (from 85% to 100 %). Next to OPSM, ISA shows relatively high portions of enriched biclusters.
According to many simulations, we found that most of the enriched biclusters contains low number of annotated genes. Figure 3.16 shows the percentage of enriched biclusters if at least half of their genes were annotated using any GO category. Figure 3.16 shows

Table 3.10: Parameters Setting of Biclustering Algorithms Applied to Gasch [27] Dataset

Algorithm	Parameters	Parameter Description
ISA	tg=2.0	Genes threshold level
	tc = 2.0	Condition threshold level
	SN= 500	Number of seeds
CC	Delta=0.5	Maximum of accepted score
	Alpha=1.2	Scaling factor
	M=100	Number of bicluster to be found
OPSM	l = 100	Number of passed models for each iteration
K-means	M=100	Number of Bicluster to be found
	IN=100	Number of Iteration
	RN=10	Number of replication
	DM=ED	Distance Metric is Euclidean Distance
Bivisu	NT=0.82	Data Noise threshold
	% NR=0.33	Minimum % of rows
	NC=5	Minimum number. of columns
	O%=25%	Maximum overlap allowed

Table 3.11: Statistical Comparison of Biclusters Produced by Applying Biclustering Algorithms to Gasch [27]Dataset

Biclustering Algorithm	No of Biclusters	BiclustersClusters Size		GeneCoverage%	ConditionCoverage%
		Min	Max		
ISA	9	50 x 35	155 x 37	25	97
CC	69	11 x 5	2259 x 134	100	100
OPSM	2	11 x 15	575 x 6	88.5	32.9
BiVisu	100	27 x 142	99 x 52	55	100
Kmeans	100	20 x 173	50 x 173	100	100

that OPSM and ISA have highly enriched biclusters that have large number of annotated genes. On the other hand, Bivisu biclusters are strongly affected by this filtration as they contains a lower number of annotated genes per each category. Figure 3.16 helps in identifying the powerful and most reliable algorithms which are able to group maximum numbers of genes sharing same functions in one bicluster.

Finally, given the ease of comparison allowed by the AGO, it was straightforward to do further analysis to address predictability power to recover interesting patterns. That is, to compare biclustering methods based on which of them could recover known patterns in the particular experimental dataset used. Table 3.12 shows the difference between the biclusters contents based on its predictability to recover response to stress category. Although OPSM showed high percentage level of enriched biclusters, it did not have

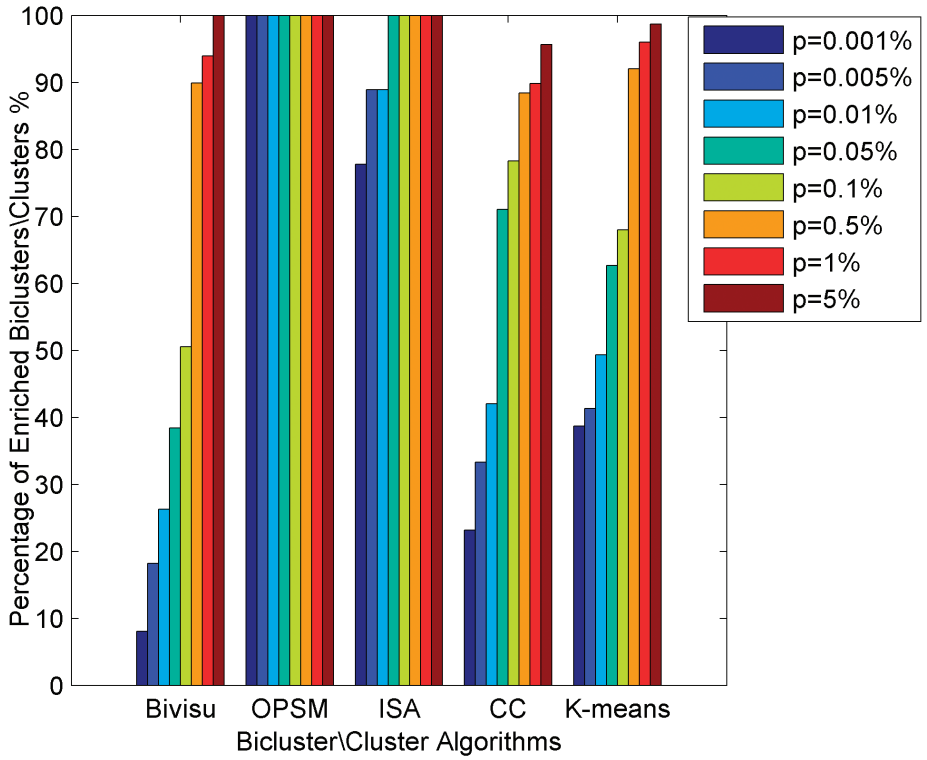


FIGURE 3.15: **Percentage of Enriched Biclusters**: This Figure draws the percentage of enriched biclusters for Biological Process GO annotations (y-axis) against the selected biclustering algorithms(x-axis) at different significance levels. The biclustering algorithms and k-means were applied to Gasch dataset [27] using parameter setting in Table 3.10 with GO annotations of Biological Process category. A bicluster is said to be significantly overrepresented (enriched) with a functional category if the p-value of this functional category is lower than the preset threshold P-value. OPSM algorithm gave a high portion of functionally enriched biclusters at all significance levels (from 85% to 100 %). Next to OPSM, ISA show relatively high portions of enriched biclusters.

any biclusters with genes matching any of the known GO categories for Gasch data set. Although the low number of ISA biclusters (9 biclusters) and GeneCoverage% (25%), it showed better performance with one of its biclusters having 11 genes matching response to oxidative stress (GO:0006979). We can see also that three methods(k-means, CC and ISA) were able to define biclusters that have 4 out of 5 genes in the cellular response to nitrogen starvation functional category, which is very remarkable. Finally, we can observe also that there are several methods assumed to be unique in detecting biclusters related to certain function categories. For example, ISA and CC detected 2 genes belong to response to cold and cellular response to starvation functions respectively.

The comparison methodology used in this study indicates that the present methods do not show a clear winner and in fact it seems that all methods should somehow be integrated together to capture the information in the data.

TABLE 3.12: Statistical Comparison of Biclusters Produced by Applying Biclustering Algorithms to Gasch [27]Dataset

GO Term /number of annotated genes	K-means	CC	ISA	Bivisu	OPSM
GO:0042493					
Response to drug (118)	4	5	7	6	0
GO:0006970					
Response to osmotic stress (83)	3	5	6	3	0
GO:0006979					
Response to oxidative stress (79)	2	7	11	0	0
GO:0046686					
Response to cadmium ion (102)	2	3	2	2	0
GO:0043330					
Response to exogenous dsRNA (7)	2	3	2	2	0
GO:0046685					
Response to arsenic (77)	2	0	2	2	0
GO:0006950					
Response to stress (532)	9	11	16	2	0
GO:0009408					
Response to heat (24)	3	0	2	2	0
GO:0009409					
Response to cold (7)	0	0	2	0	0
GO:0009267					
Cellular response to starvation (44)	0	2	0	0	0
GO:0006995					
Cellular response to nitrogen starvation (5)	4	4	4	0	0
GO:0042149					
Cellular response to glucose starvation (5)	0	2	0	0	0
GO:0009651					
Response to salt stress (15)	2	7	0	0	0
GO:0042542					
Response to hydrogen peroxide (5)	0	0	0	2	0
GO:0006974					
Response to DNA damage stimulus (240)	0	22	0	3	0
GO:0000304					
Response to singlet oxygen (4)	2	0	0	0	0

We test the predictability ability of different biclustering algorithms to recover gene ontology category within response to stress (GO:0006950). Rows represent the known gene ontology function categories under response to stress category and the different biclustering methods in the columns with the highest performance relevant cluster result as the entry for a given functional category and clustering method. Several interesting observation can be made.

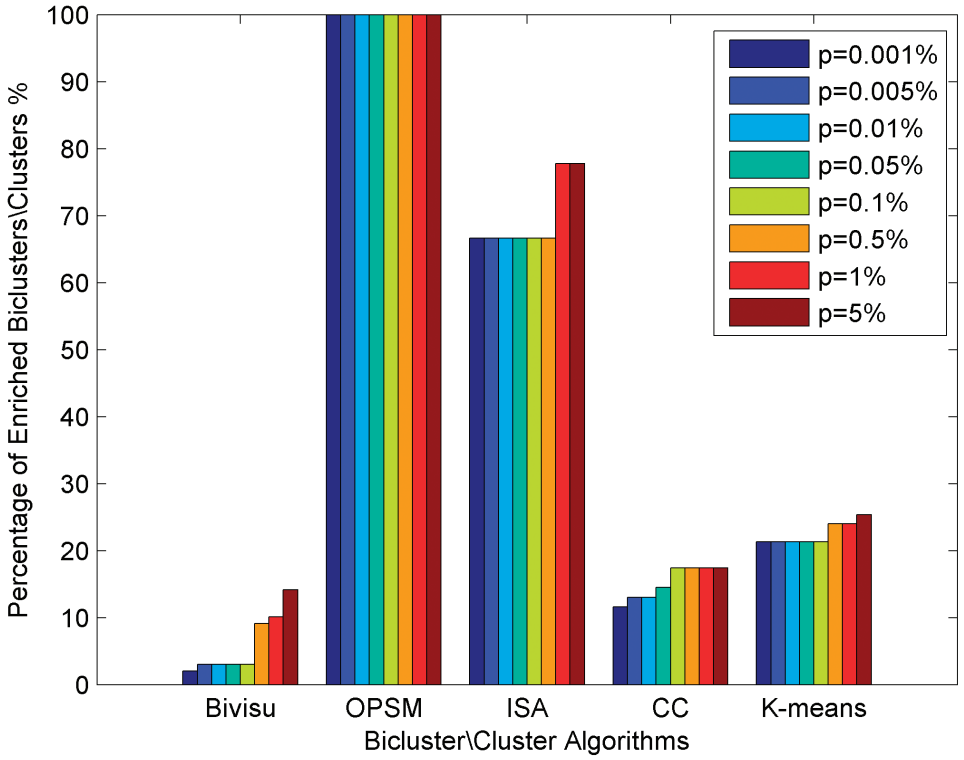


FIGURE 3.16: **Percentage of Enriched Biclusters using Restricted Criteria**: This Figure is similar to Figure 3.15 with restriction on enrichment definition. A bicluster is said to be significantly enriched if the p-value of any of GO category is lower than the preset threshold P-value and at least half of its genes was annotated with this GO category. Bivisu biclusters are strongly affected by this filtration as they contains a lower number of annotated genes per each category. This filtration criteria helps in identifying the powerful and most reliable algorithms which are able to group maximum numbers of genes sharing same functions in one bicluster.

3.6 BicAT-plus: An Automatic Comparative Java Tool For Bi/-Clustering Algorithms Used In Analysis And Visualization of Gene Expression Data Obtained Using Microarrays

To facilitate the comparison algorithms, it is preferable to implement AGO with one of wide biclustering available toolbox. So we incorporate AGO in BicAT toolbox [17]. BicAT [17] is a common biclustering analysis toolbox in which most important bi/clustering algorithms like k-means, HCL [18], Bimax [15] , OPSM [19], X-motif [20],CC [11], and ISA [21, 22] were implemented (Fig3.17). The new version of BicAT toolbox is called BicAT-Plus[18] [16] and manual file can be downloaded from:

[18]Al-Akwaa FM, Ali MH, Kadah YM. BicAT-Plus: An Automatic Comparative Tool For BiClustering of Gene Expression Data Obtained Using Microarrays. 26th National Radio Science Conference (NRSC) Cairo, Egypt, 2009.

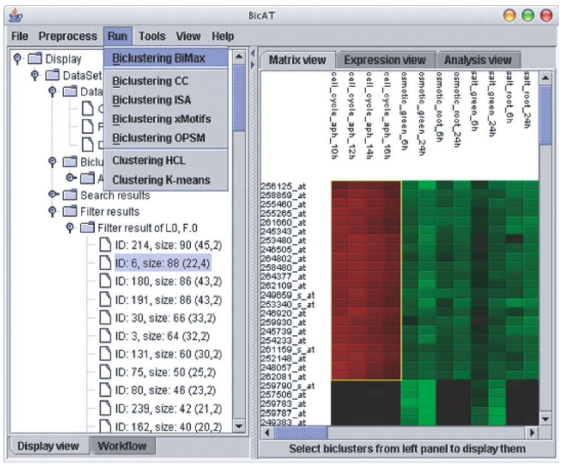


FIGURE 3.17: Bi/clustering Algorithms Employed by BicAT (Copyright©[17])

http://home.k-space.org/FADL/Downloads/PhD/Bicat-Plus_paper/

BicAT-plus has many features added to BicAT which could summarized in the following:

1. Adding more algorithms to the BicAT tool in order to have one software package that employs most of the commonly used bi/clustering algorithms. The additional algorithms are MSBE constant biclustering(Figure 3.18) and MSBE additive biclustering(Figure 3.19) [70].

2. Extending the BicAT to perform functional analysis using the three subontologies or categories of Gene Ontology (GO) (biological process, molecular function and cellular component)(Figure 3.20) and visualizing the enriched GO terms per each bi/cluster in a separate histogram.

3. Evaluating the quality of each bi/clustering algorithm (Figure 3.21) results after applying the GO functional analysis and displaying the percentage of the enriched bi/clusters at the standard P-values (significance levels) which are: 0.00001,0.00005,0.0001,0.0005,0.001,0.005,0.01 and 0.05.

4. Comparing between the different bi/clustering algorithms according to the percentage of the functionally enriched bi/clusters at the required significance levels, the selected GO category and with certain filtration criteria for the GO terms(Figure 3.20).

5. Evaluating and comparing the results of external bi/clustering algorithms (not included in the BicAT-plus current version). This gives the BicAT-plus the advantage to be a generic tool that doesn't depend on the employed methods only. For

example; it can be used to evaluate the quality of the new algorithms introduced to the field and compare against the existing ones.

6. Displaying the analysis and comparison results using graphical and statistical charts visualizations in multiple modes (2D and 3D)(Figure 3.22).

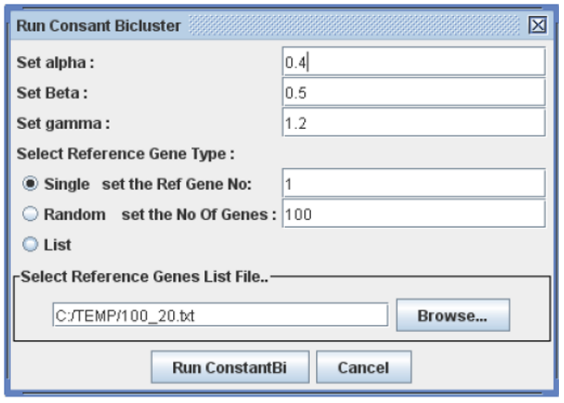


FIGURE 3.18: **Constant MSBE Biclustering Input Dialog Implemented in Our BicAT-Plus Toolbox [16]**: alpha, beta and gamma to be determined by the user and the number of reference genes.

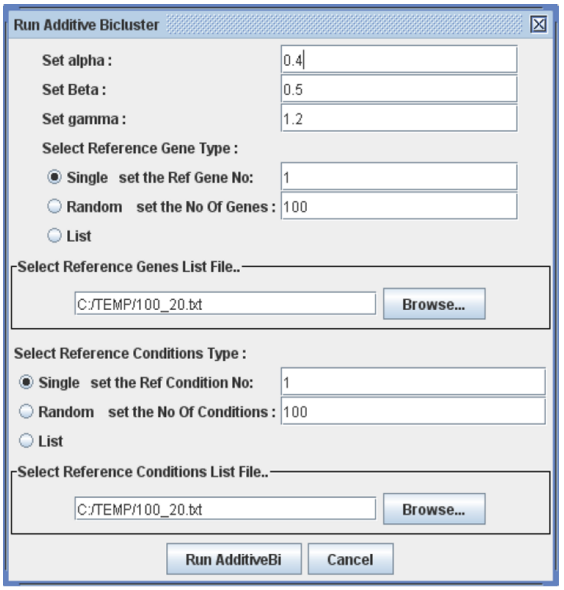


FIGURE 3.19: **Additive MSBE Biclustering Input Dialog Implemented in Our BicAT-Plus Toolbox [16]**:alpha, beta and gamma to be determined by the user and the number of reference genes and conditions.

3.6.1 BicAT-Plus Development and Architecture

Before using the BicAT-plus, Active Perl version 5.10 and Java Runtime Environment (JRE).version 6 are required to be installed on your machine. BicAT-plus has been

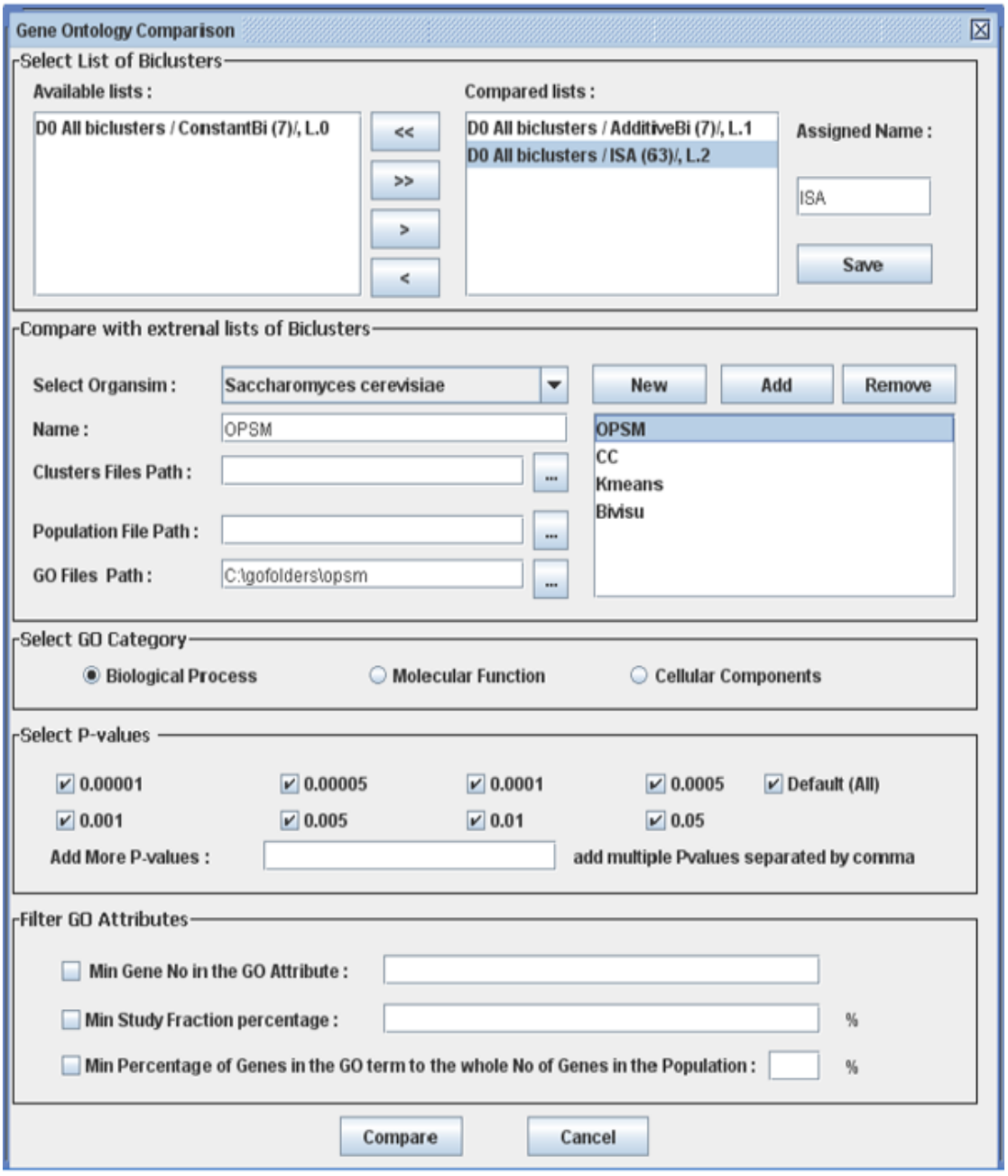


FIGURE 3.20: Algorithms required to compare could dragged from available list to compared list. External biclustering results for other algorithms could be included in the comparison process. Also organism model, selectable significance level, GO category should be selected. Finally Comparison criteria have to be selected based on the user biological metric.

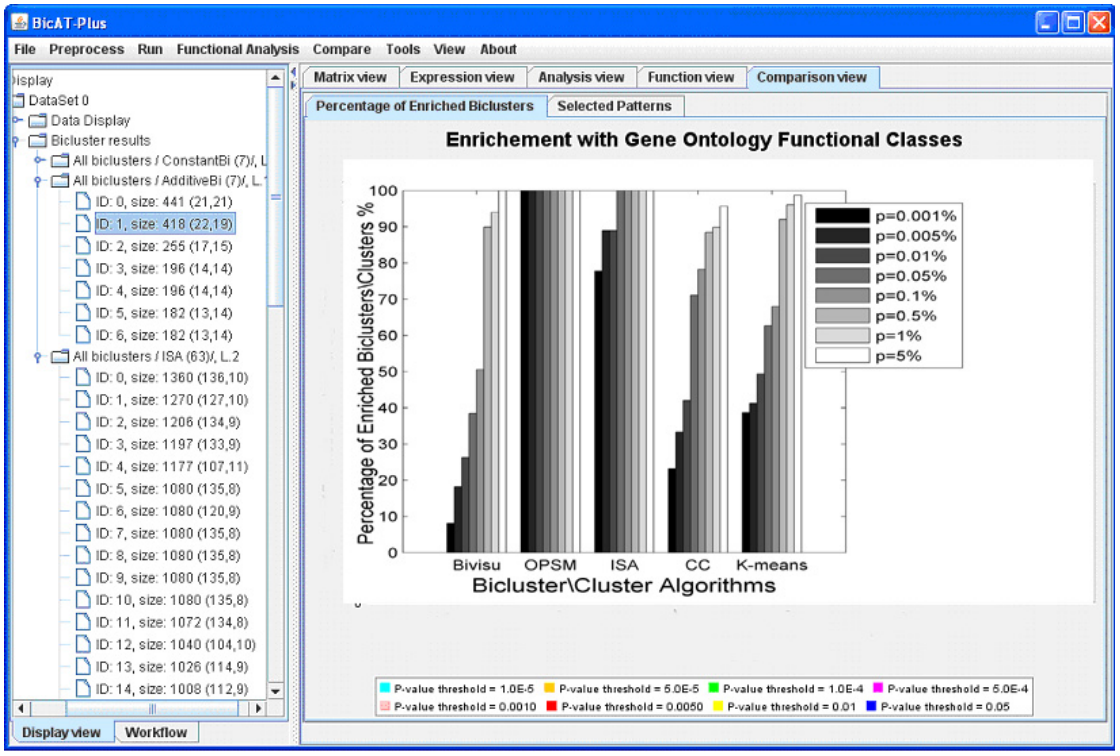


FIGURE 3.21: **Percentage of Enriched Biclusters**: This Figure draws the percentage of enriched biclusters for Biological Process GO annotations (y-axis) against the selected biclustering algorithms(x-axis) at different significance levels.

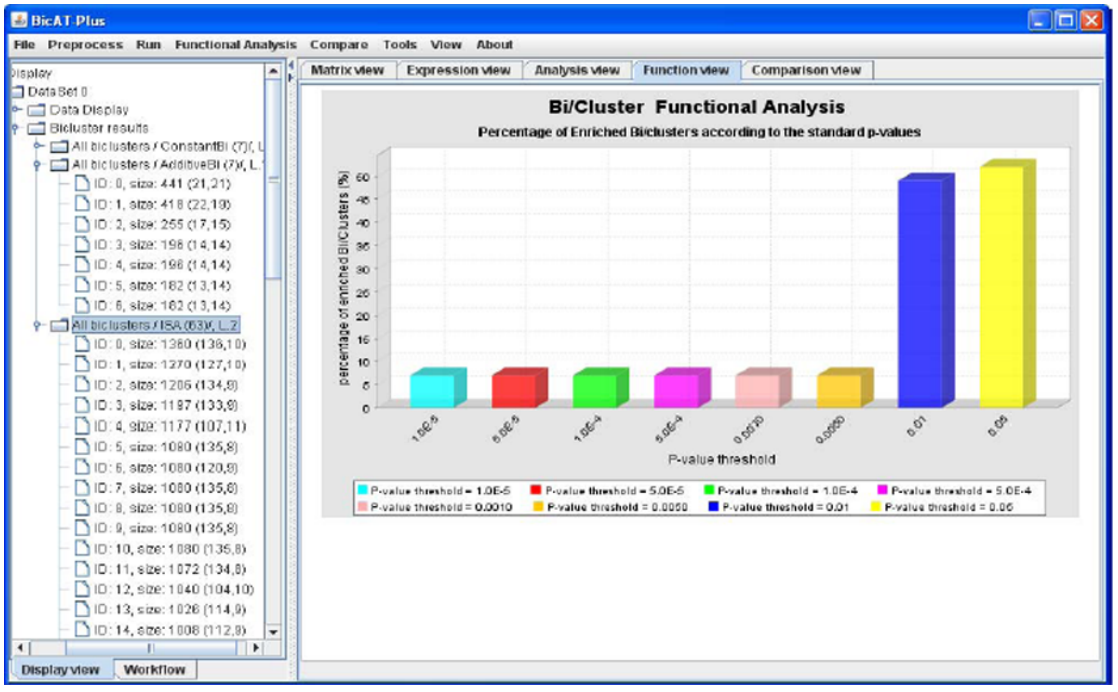


FIGURE 3.22: Functional analysis of the selected algorithm results

tested and show good performance on a PC machine with the following configurations: CPU: Pentium 4, 1.5 GHZ, RAM: 2.0 GB, Platform: windows XP professional with SP2. BicAT-plus is structured in the hierarchy of packages which are shown in Fig 3.23. The highlighted blocks with red color are the additional modules developed for the comparative tool while the black ones are the original modules of the BicAT program. We faced many problems during the implementations like:

1. lack of documentation of the BicAT tool which influenced the planned time to understand the source code and extend it.
2. All bugs reported about BicAT should be fixed in order to avoid its effect on the comparative tool. Ex: delete node from the navigation tree.
3. Technical problems like calling GeneMerge Perl script from java code. The used solution was to save the Perl commands in a batch file, then call the batch file from the java code using the Runtime class provided by SUN.
4. One of the objectives of this research was to enrich the BicAT (written using java) with more biclustering algorithms. But, some of these algorithms are written using C and C++. Thus, to solve such a compatibility problem, we converted the C files to dynamic link library (DLL) file then loaded it to the system class path library. Another possible solution was to use the Java native interface (JNI) to call the C files.

3.6.2 BicAT-Plus Comparison Process Steps

The following process diagram shown in Fig 3.24 summarizes the required steps by the user to compare between the different algorithms using the BicAT-plus.

1. Download BicAT-plus from our site (http://home.k-space.org/BicAT-plus.zip).
2. Load Gene Expression Data to BicAT-plus then run the selected five prominent biclustering methods with setting parameters as Table 3.10.
3. Run GO comparison tool in the BicAT-plus and add the available bi/clustering algorithms to the compared list as shown in Figure 3.20.
4. Select one of the available GO category e.g. biological process, molecular function and cellular components as in Figure 3.20.
5. Select the P-values e.g. 0.00001, 0.0001, 0.01, 0.005, and 0.05.

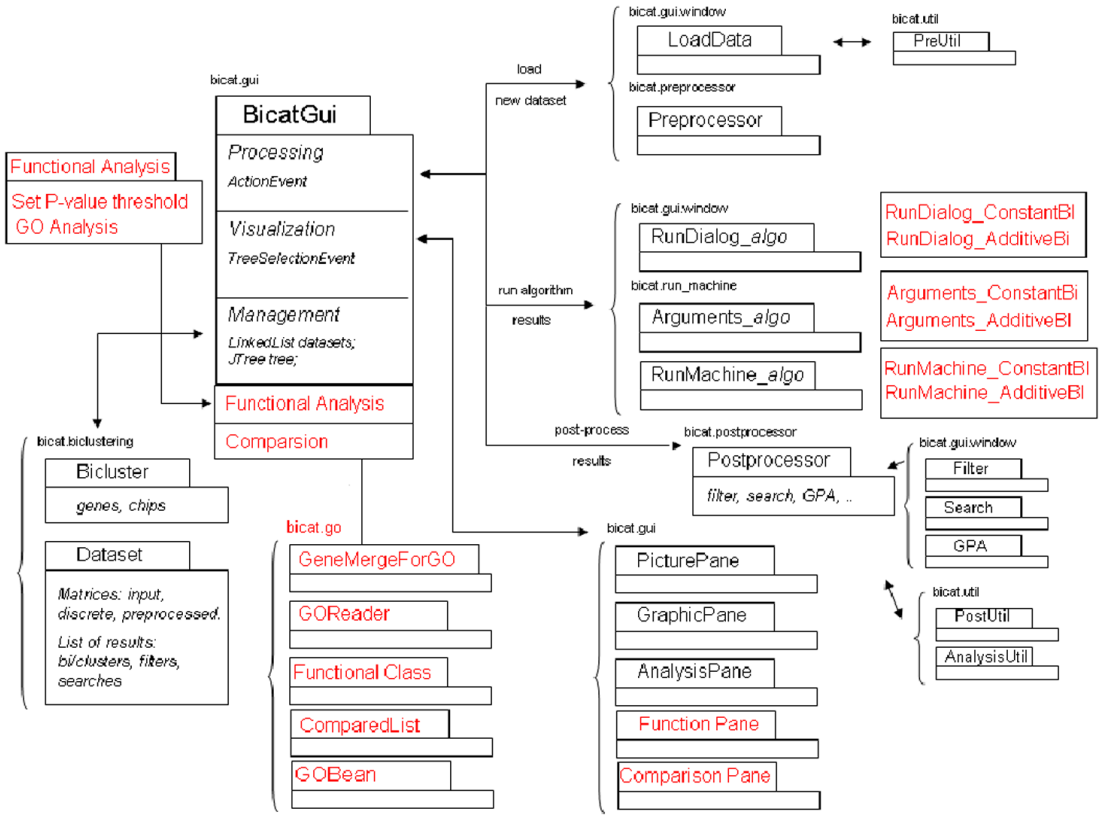


FIGURE 3.23: The general design of the BicAT-Plus. Red color for the comparative tool packages and classes. The black entities are the original packages and interfaces of the BicAT program. (Modified from [84]).

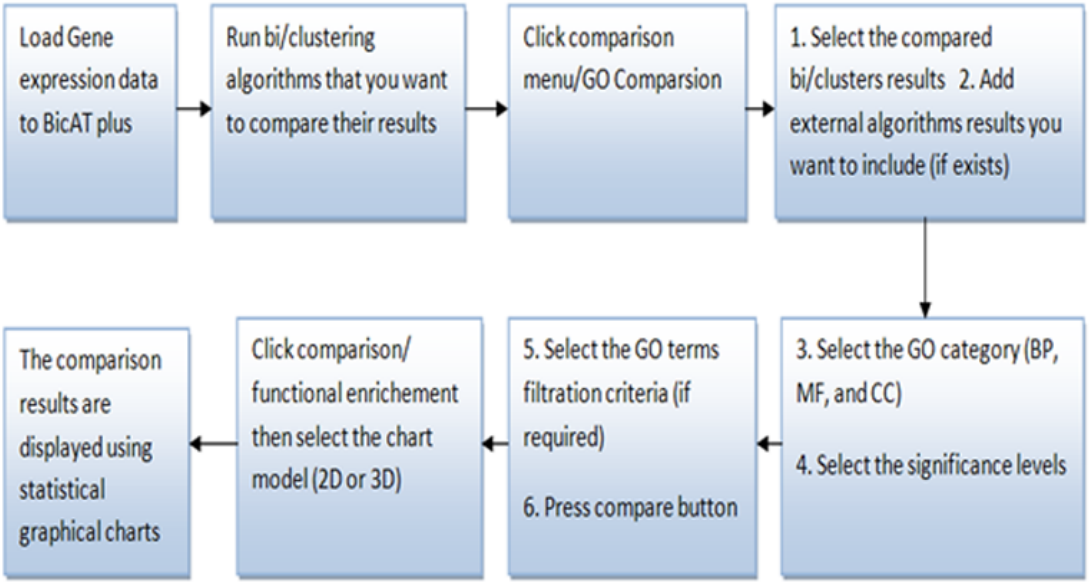


FIGURE 3.24: BicAT-Plus Comparison Process Steps.

6. Press compare button.

7. Press comparison menu, Functional enrichment and select 2D or 3D charts see Figure 5.(Figure 3.22)

Chapter 4

Bayesian Network

In the last years numerous methods have been developed and applied to reconstruct the structure and dynamic rules of gene-regulatory networks from different high-throughput data sources such as gene expression data. In this chapter we summarized some of the promising modeling approaches(Section 4.1) to obtain a better understanding of their relative strengths and weaknesses. We focus on probabilistic models(Bayesian Network)(Section 4.4) that use stochasticity to account for measurement noise, variability in the biological system, and aspects of the system that are not captured by the model. Finally, in Section 4.5 we compare between different Bayesian Network Structure Learning algorithms.

4.1 Reverse Engineering Approach

Many approaches have been developed to reverse the gene network. Many review papers [85, 86, 86–93] had been published comparing these approaches.
We choose four modeling approach which are Bayesian Network (BYN), Boolean Network (BNN), Non Linear Ordinary Differential Equation (NLODE), and Association Networks (AN), based on their promising results, to they extend on the community and the availability of its implementation which makes an easy for the reader to test each approach on synthetic or read data set with out involving in the implementation complicity. The available software for each method are Bayesnet Toolbox [94], Probabilistic Boolean Network [95], Genetic Network Analyzer (GNA) [96] and ARACNE [97] respectively.
Table4.1 compares between promising GRN modularity approaches as following:

TABLE 4.1: Comparison Between GRN Modeling Approaches

Approach	Static(s)/ dynamic(d)	Discreet(d)/ Continuous(c)	Deterministic(d) / Stochastic(s)	Qualitative(ql)/ quantitative(qn)
Bayesian Network	s	d,c	s	qn
Boolean Network	d	d	d	ql
NLDE[a]	d	c	d	qn
Association Network	s	c	d	qn

[a] Non Linear Definitional Equations

- Discrete or continuous Ivan et al [98] Compared fine-scale stochastic-differential equation models with coarse-scale discrete models in the context of currently available data and with respect to their description of switch-like behavior among specific groups of genes. They find that a discrete model has predictive power comparable to that of the stochastic differential equation model under the assumption of complete knowledge of the parameters of the fine-scale model.

- Deterministic or Stochastic In deterministic models we assume that the next state of the system is determined by the current state and the external inputs. However, in real world systems stochastic effects may play an important role. For instance, for some genes in yeast the number of mRNA molecules is close to one copy per cell [99]. This means that it is likely that there is a considerable intrinsic noise element present - some cells apparently have more mRNA molecules of the given species present than others. Thus modeling a cell by using continuous concentrations effectively means modeling an ensemble of cells by mean values of stochastic variables. Simulating a stochastic model is computationally more expensive, because the simulations have to be run several times to provide a good impression of the system behaviour. But stochastic models are not always necessary; it depends on the system that is to be modeled. If the number of molecules involved is small and if important processes depend on random effects, stochastic models might be the best choice.

- Data Discretization Reconstructing regulatory networks from gene expression profiles is a challenging problem of functional genomics. In microarray studies the number of samples is often very limited compared to the number of genes, thus the use of discrete data may help reducing the probability of finding random associations between genes. On other hand the previous studies [55] were suggested that discretization of the continuous data leads to a large information loss. Barbara Di Camillo et.al [56] confirmed that the use of discrete rather than continuous data is advantageous when few samples are available. Continuous

approaches are likely to become advantageous with increasing number of samples.

4.1.1 Boolean Network

The simplest dynamic models synchronous Boolean network models were used as a model for gene regulatory networks already in the 1960's by Stuart Kauffman [3]. Boolean networks are based on the assumption that binary on/off switches functioning in discrete time steps can describe important aspects of gene regulation. In synchronous Boolean network models all genes switch states simultaneously (Figure 4.1). We can

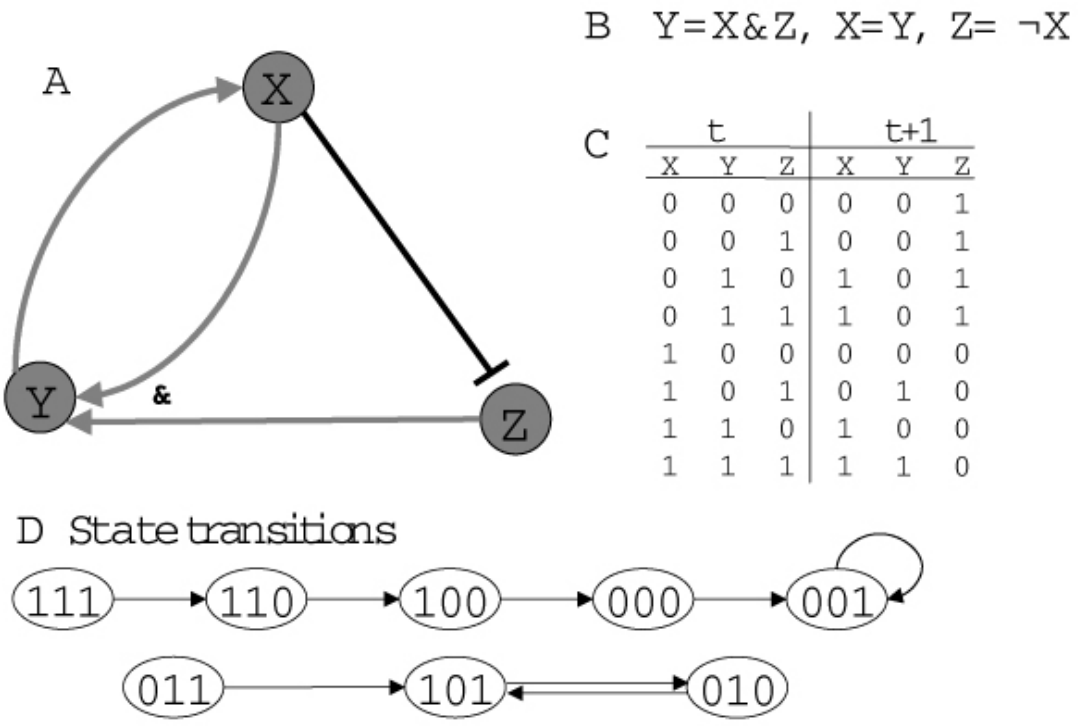


FIGURE 4.1: Example for a small Boolean network consisting of 3 genes X, Y, Z. There are different ways for representing the network: A as a graph, B Boolean rules for state transitions, C a complete table of all possible states before and after transition, or D as a graph representing the state transitions. Copyright © [85].

introduce the concept of the state of the network defined as an n-tuple of 0s and 1s describing which genes in the network are or are not expressed at the particular moment (Figure 4.1). As time progresses, the network navigates through the 'state space', switching from one state to another, as shown in Figure4.1 D. For a network of n genes, in total there are 2n possible different states, for instance, for a three gene network the possible states are (0,0,0), (0,0,1), ..., (1,1,1). We can follow the succession of states with time and study which states are reached. Some states might never be

reached. It is possible to look for attractors: these are states or series of states that once reached will not be left anymore. The small example network in Figure 4.1 has two attractors: one attractor is a single state (0,0,1), and the second attractor consists of two alternating states (1,0,1) and (0,1,0). This approach has been generalized in a number of ways. Randomly generated networks are used to study the dynamics of complex systems [100]. Stochastic extensions to deterministic Boolean networks were proposed so-called noisy networks by Akutsu et al. [88] and Probabilistic Boolean Networks by Shmulevich et al. [101].

4.1.2 Non Linear Definitional Equations

Nonlinear ordinary differential equations are probably the most-widespread formalism for modeling genetic regulatory networks. They represent the concentration of gene products mRNAs or protein by continuous, time-dependent variables, that is, $x(t)$, $t \in T$, T being a closed time interval ($T \in R_{\geq}0$). The variables take their values from the set of nonnegative real numbers ($x : T \rightarrow R_{\geq}0$), reflecting the constraint that a concentration cannot be negative. In order to model the regulatory interactions between genes, functional or differential relations are used.
More precisely, gene regulation is modeled by a system of ordinary differential equations having the following form:

$$\frac{dx}{dt} = f(x)$$

where x = $[x_1, ..., x_n]'$ is the vector of concentration variables of the system, and the function f =$[f_1, ..., f_n]'$, usually highly nonlinear, represents the regulatory interactions. The above system does not include the delays resulting from the time it takes to complete transcription, translation, and the other stages of the synthesis and the transport of proteins you can see [102] for more details.
The above definitions can be illustrated by means of a simple network in Figure 4.2. Each of the genes encodes a regulatory protein that inhibits the expression of the other gene, by binding to a site overlapping the promoter of the gene.
An ordinary differential equation model of the network in Figure 4.2 is shown in Figure 4.3. The variables x_a and x_b represent the concentration of proteins A and B, encoded by genes a and b, respectively. The temporal derivative of x_a is the difference between the synthesis term $k_a h^-(x_b, \Theta_b, m_b)$ and the degradation term $\gamma_a x_a$. The first term expresses that the rate of synthesis of protein A depends on the concentration of protein B and is described by the function h^-. This so called Hill function is monotonically decreasing.

It takes the value 1 for $x_b = 0$, and asymptotically reaches 0 for $x_b \leq \infty$. It is characterized by a threshold parameter θ_b and a cooperativity parameter m_b (Figure 4.3b). For m_b >1, the Hill function has a sigmoidal form that is often observed experimentally [103]. The synthesis term $k_a h^-(x_b, \Theta_b, m_b)$ thus means that, for low concentrations of protein B, gene a is expressed at a rate close to its maximum rate $k_a (k_a > 0)$, whereas for high concentrations of B, the expression of the gene is almost completely repressed. The second term of the differential equation, the degradation term, expresses that the degradation rate of protein A is proportional to its own concentration x_a, γa being a degradation parameter ($\gamma_a > 0$). Unfortunately, they are difficult to treat mathematically for networks comprising more than two genes, in which case we have to take recourse to numerical simulation. However, the application of numerical techniques is often difficult in practice, due to the absence of numerical values for the parameters in the model. A possible alternative is the use of linear ordinary differential equations. Powerful techniques for solving these equations exist, as well as techniques for estimating parameter values from experimental data.

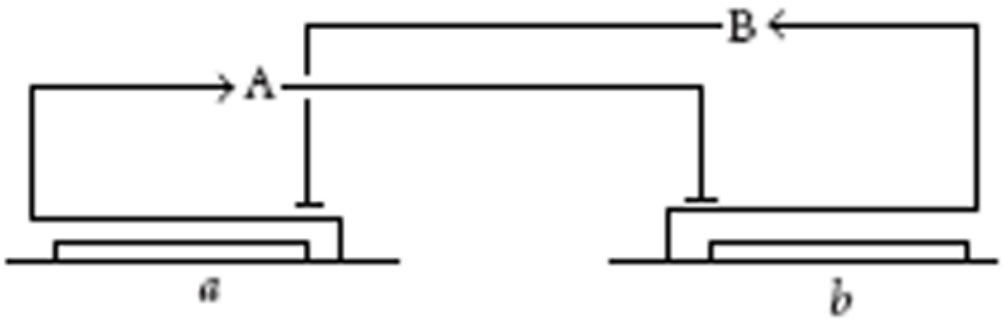


FIGURE 4.2: Example of a simple genetic regulatory network, composed of two genes a and b, the proteins A and B, and their regulatory interactions. Copyright © [104].

4.1.3 Stochastic Differential Equation

Real genetic networks are subject to considerable noise, and hence ideally should be modeled them using stochastic differential equations, or some other type of random process [86]. However, as far as we are aware, due to the complexity involved in estimating, solving and analysing stochastic models, these are rarely used to model real networks of more than two or three genes and it is beyond this study.

$$\frac{dx_a}{dt} = \kappa_a h^-(x_b, \theta_b, m_b) - \gamma_a x_a$$
$$\frac{dx_b}{dt} = \kappa_b h^-(x_a, \theta_a, m_a) - \gamma_b x_b$$
$$h^-(x, \theta, m) = \frac{\theta^m}{x^m + \theta^m}$$

(a)

(b)

FIGURE 4.3: Nonlinear ordinary differential equation model of the mutual-inhibition network (Figure 4.2). The variables x_a and x_b correspond to the concentrations of proteins A and B, respectively, parameters k_a and k_b to the synthesis rates of the proteins, parameters γ_a and γ_b to the degradation constants, parameters Θ_a and Θ_b to the threshold concentrations, and parameters m_a and m_b to the degree of cooperativity of the interactions. All parameters are positive. (b) Graphical representation of the characteristic sigmoidal form, for m ¿ 1, of the Hill function $h^-(x, \Theta, m)$. Figure Copyright © [104].

4.1.4 Association Network

If two genes show similar expression profiles, they are supposed to follow the same regulatory regime. To put it more pointedly: coexpression hints at coregulation. Coexpression networks (also known as relevance networks) are constructed by computing a similarity score for each pair of genes. If similarity is above a certain threshold, the gene pair gets connected in the graph, if not, it remains unconnected. Networks of coexpressed genes provide a widely applicable framework for assigning gene function [105]. Also, the coexpression agrees well with functional similarity as it is encoded in the Gene Ontology [77]. The first critical point in building a coexpression network is how to formalize the notion of similarity of expression profiles. Several measures have been proposed, the most simple of which is correlation. In a Gaussian model, zero correlation corresponds to statistical independence. The second critical step in building coexpression networks is assessing the significance of results. Many pairs of genes show similar behavior in expression profiles by chance even though they are not biologically related. Even high similarity of expression tells us little about the underlying biological mechanisms. Coexpression networks include regulatory relationships, but we cannot distinguish direct from indirect dependencies based on the similarity of expression patterns. Figure 4.4 exemplifies this problem on a small set of three highly coexpressed genes, which form a clique (a completely connected subgraph) in a coexpression network.

Figure 4.4 shows that several regulatory mechanism can explain this observation, and from coexpression data alone we have no way of choosing between them. There are

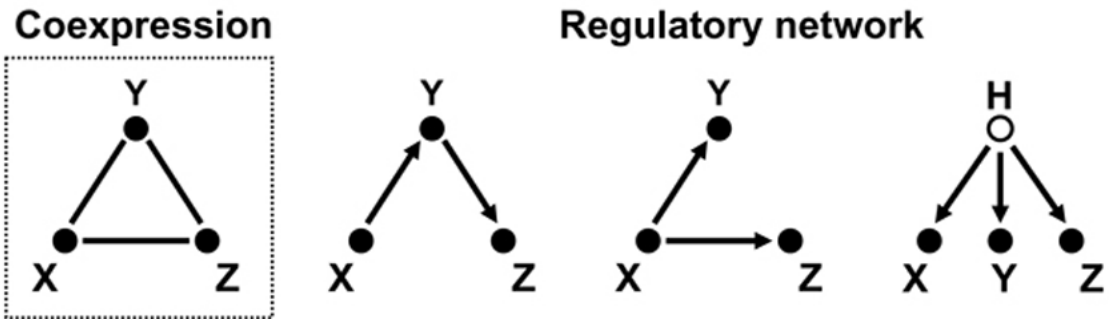


FIGURE 4.4: Different mechanisms can explain coexpression. The left plot in the dashed box shows three coexpressed genes forming a clique in the coexpression graph. The other three plots show possible regulatory relationships that can explain coexpression: The genes could be regulated in a cascade (left), or one regulates both others (middle), or there is a common "hidden" regulator (right), which is not part of the model. Figure Copyright © [93].

two possible solutions. Functional genomics has a long tradition of perturbing the natural state of a cell and inferring a gene's function from the observed effects. These interventions allow us to distinguish between the three models in Figure 4.4, because each model results in different predictions of effects, which can be compared to those obtained in experiments. For example, perturbing gene Y in the cascade $X \leq Y \leq Z$ will only have an effect on gene Z but none on gene X. In the case where Y regulates both X and Z, perturbing it will result in changes at both regulatees. In the last case, where all three genes are regulated by a hidden regulator, perturbing one of them will not lead to changes at the other two. In the absence of perturbation data statistical methods may be used to find which of the possibilities is most likely. The theoretical background is the concept of conditional independence. Please see [93] for more details.

4.2 Which Model Should I Select?

We need to say that these models without real and clear problem statement are like computer games simulation. i.e you should to fit your problem with one of these models; not all models work perfectly. For example If we are not interested in predicting the exact concentrations of different substances, but only in the patterns of the systems behaviour such as steady states, we can often use simplified Boolean-type networks instead of differential equations [106].

4.3 Data Sources and Requirements

Gene network inference techniques are data-hungry [107]. Time series and steady-state data are the available gene expression data. Time series has the advantage of being able to identify causal relations, i.e. gene-regulatory relations, between genes without the need of actively perturbing the system. Spellman et al. [28] generated time series data under different culture conditions and using different mutant backgrounds in order to reveal a more comprehensive picture about gene regulation during the yeast cell cycle. Table 4.2 shows how many data points do we really need to infer a gene network on N genes depends on the model used to do the inference [107].

The need for large numbers of data points, and many different conditions, implies that

TABLE 4.2: Data Requirements of Difference Reverse Engineering Approach: Data Size to Recover Gene Networks with N Genes and Connectivity K

Model	Data needed
Boolean, fully connected[a]	2N[b]
Boolean, connectivity K[c]	2Klog(N)
Boolean, connectivity K, lin. sep.[d]	Klog(N/K)
Continuous, fully connected, additive	N
Continuous, connectivity K, additive[e]	Klog(N/K)
Pairwise correlation	log(N)

[a] Each gene can receive regulatory inputs from all other genes
[b] No of genes
[c] Maximum regulatory inputs per gene
[d] Linearly separable, for Boolean functions
[e] Regulation can be modeled as a weighted sum

successful modeling efforts will probably have to use data from different sources like from different high-throughput data sources, mainly microarray based gene expression analysis, promoter sequence information, Chromatin immunoprecipitation (ChIP) and protein-protein interaction assays.

The requirements on data size and data quality that must be met by a successful network reconstruction could be summarized as the following:

- Short time series generated under transcription factor knock-out are optimal experiments in order to reveal the structure of gene regulatory networks.
- The benefit of using of prior knowledge within a Bayesian learning framework is found to be limited to conditions of small gene expression data size.

- The results suggest that discretization of the continuous data leads to a large information loss.
- Results indicate, that network reconstruction with currently available data will still give rise to many false predictions (FDR 50%).

4.4 Bayesian Network

While a variety of computational methods have been considered for reconstructing gene networks from observational gene expression data, Bayesian network (BN) based approaches have shown great promise to infer causal relationships between genes and receive increasing attention. One of the first seminal papers promoting this approach aimed to learn gene regulatory networks in *Saccharomyces Cerevisiae* from gene expression profiles with Bayesian networks [4].
BN are especially suitable for learning genetic regulatory networks for the following reasons: (1) the sound probabilistic semantics allows BNs to deal with the noises that are inherent in experimental measurements; (2) BNs can handle missing data and permit the incomplete knowledge about the biological system and (3) BNs are capable of integrating prior biological knowledge into the system [108].

4.4.1 Bayesian Networks Representation

Consider a finite set x=$X_1, X_2, ..., X_n$of random variables where each variable X_i may take on a value $x_i$from the domain Val(X_i). we use capital letters, such as *X,Y,Z* for variable names and lowercase letters *x,y,z* to denote specific values taken by those variables. Sets of variables are denoted by boldface capital letters ***X,Y,Z*** and assignments of values to the variables in these sets are denoted by boldface lowercase letters***x,y,z***. We denote I(X;Y—Z) to mean ***X*** is independent of ***Y*** conditioned on ***Z***. A *Bayesian network* is a representation of a joint probability distribution. This representation consists of two components. The first component, G , is a directed acyclic graph (DAG) whose vertices correspond to the random variables$X_1, X_2, ..., X_n$. The second component, Θ describes a conditional distribution for each variable, given its parents in G . Together, these two components specify a unique distribution on $X_1, X_2, ..., X_n$.
The graph G represents conditional independence assumptions that allow the joint distribution to be decomposed, economizing on the number of parameters. The graph G encodes the Markov Assumption:
(*) Each variable X_i is independent of its non-descendants, given its parents in G.

By applying the chain rule of probabilities and properties of conditional independencies, any joint distribution that satisfies (*) can be decomposed into the product form:

$$p(x_1, ..., x_n) = \prod_{i=1}^{n} P(x_i | pa^G(X_i))$$

Where $pa^G(X_i)$is the set of parents of X_i in G.Figure 4.5 shows an example of a graph G , lists the Markov independencies it encodes, and the product form they imply.

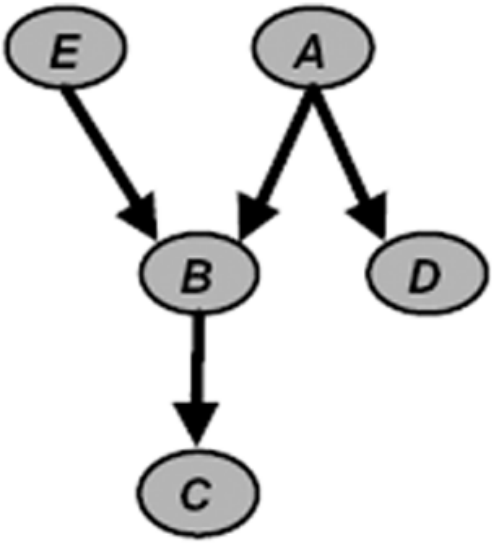


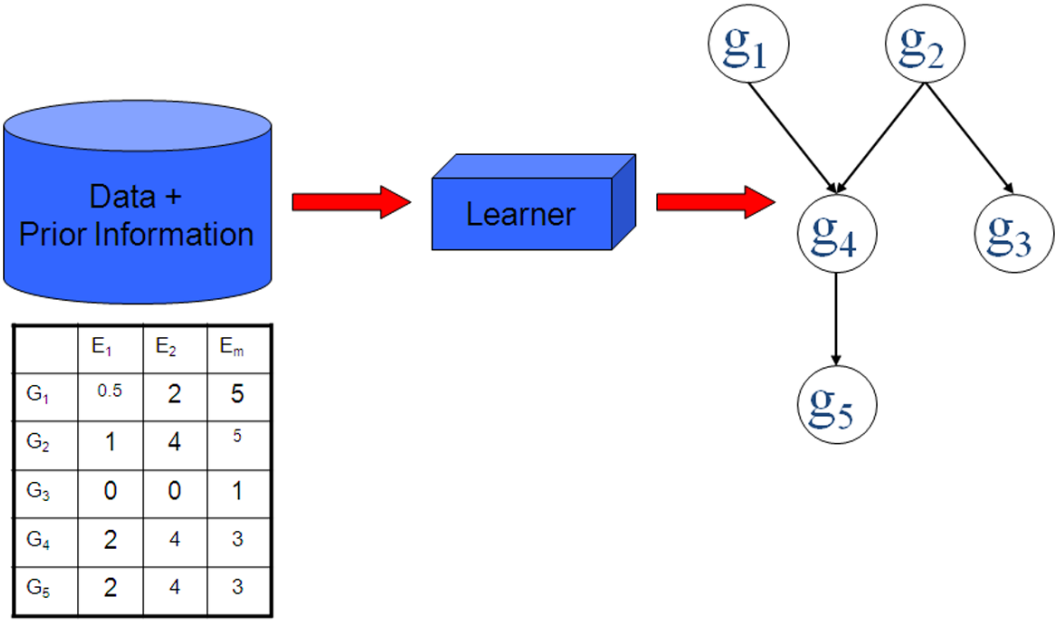
FIGURE 4.5: An example of a simple Bayesian network structure. This network structure implies several conditional independence statements:$I(A; E), I(B; D|A, E), I(C; A, D, E|B), I(D; B, C, E|A), I(E; A, D)$ The network structure also implies that the joint distribution has the product form $P(A, B, C, D, E) = P(A)P(B|A, E)P(C|B)P(D|A)P(E)$. Figure Copyright © [4]

4.4.2 Bayesian Networks Structure Learning

The theory of learning networks structure from data has been examined extensively over the last decade(see Figure 4.6).

The problem of learning a Bayesian network can be stated as follows. Given a training set $D = X^1, X^2, ..., X^N$ of independent instances of X , find a network $B = (G, \Theta)$that best matches D. More precisely, we search for an equivalence class [1]of networks that best matches D [4].

[1]Set of graph which can imply exactly the same set of independencies.



	E_1	E_2	E_m
G_1	0.5	2	5
G_2	1	4	5
G_3	0	0	1
G_4	2	4	3
G_5	2	4	3

FIGURE 4.6: **Bayesian Network Structure Learning Problem**: From the Expression Level of Five Genes Shown(left), It was Required to Construct the Network Structure Between Them(Right)

Many *structure learning* methods have been proposed in the literature, and it is important to understand their relative merits and shortcomings. They can be categorized as either conditional independence (CI) test-based methods or scoring-based methods. The CI-based methods analyze the dependence and independence relationships among variables via CI tests and construct the networks that characterize these relationships. The scoring-based methods consist of two components: (1) a scoring function that assesses how well a network fits the data and (2) a search method to find networks with high scores.

4.4.2.1 Scoring Function

Learning a BN structure is to find a DAG that best matches the dataset. The common method of structure learning is to define a scoring function that evaluates how well the DAG explains the data and then to search for the best DAG that optimizes the scoring function. NormalGamma, MeanSquareError, BIC (Bayesian Information Criterion) and BDe are the common scoring function were used. For all three scoring functions used, the component scores and the total network score are always negative numbers; a better network has a higher score, i.e. a negative score of smaller magnitude.
A commonly used scoring function for discrete data is called BDe scoring metric which

computes the posterior probability of a network for the given data [109].
In this score, we evaluate the posterior probability of a graph given the data:

$$S(G:D) = logP(G|D) = logP(D|G) + logP(G) + C$$

where C is a constant independent of G and $P(D|G) = \int P(D|G,\theta)P(\theta|G)d\theta$ is the marginal likelihood which averages the probability of the data over all possible parameter assignments to G. The particular choice of priors $P(G)$ and $P(\theta|G)$ for each G determines the exact Bayesian score.

4.4.2.2 Heuristic Search

The number of DAGs as a function of the number of nodes, G(n), is super-exponential in n, and is given by the following recurrence:

$$G(n) = \sum_{k=1}^{n} (-1)^{k+1} \binom{n}{k} 2^{k(n-k)} G(n-k)$$

For example DAG space of 10 genes are $4.2x10^18$. Also, It takes around 50 hours using a Sun Fire 15K supercomputer with 96 CPUs, 900MHz each, to compute the gene regulatory network of just 20 genes [110]. Since the number of DAGs is super-exponential in the number of nodes, we cannot exhaustively search the space, so we either use a local search algorithm (e.g., greedy hill climbining, perhaps with multiple restarts) or a global search algorithm (e.g., Markov Chain Monte Carlo). These algorithms were compared in the below section.

4.4.2.3 Model Averaging

Instead of taking the best network which have the best score or posterior probability, we can consider the average of the predicted networks. Using available dataset there are many different networks that score approximately equally well(Figure 4.7).Each predicted networks have common edges. Important edges are those appeared in a majority of the search results. So we have to average the produced networks to get the final network with high confidence level(Figure 4.8). For example, we could build the network which its edges appear in the results of more than half the searches. Also, we could generalized like this an edge will then appear in the final network if it appears

in the results of more than N% of the searches. (100% means the edge appeared in the results of all runs; 95% means it appeared in 95% of runs; etc.).

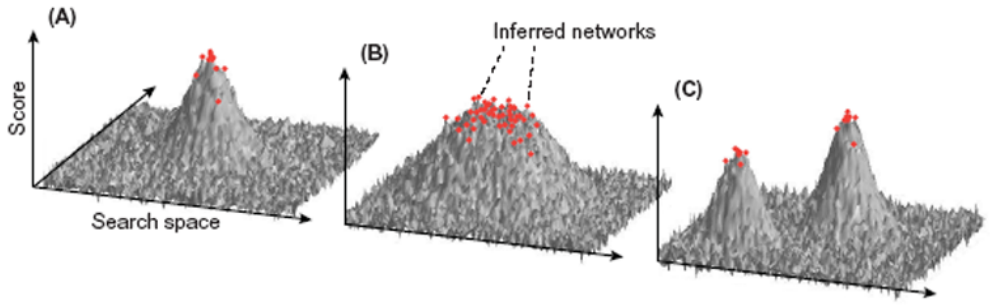


FIGURE 4.7: **Schematic Representation of Possible Posterior Distributions in a Reverse Engineering Problem**: The horizontal plane represents the search space of all possible networks and the vertical axis corresponds to the score (e.g., the posterior probability). The dots are tentative networks inferred by a reverse engineering algorithm. (A) The data is sufficient to identify a unique, distinctive global optimum. (B) The problem is underdetermined by the available datathere are many different networks that score approximately equally well. (C) There are several distinctive classes of networks that fit the data well. Figure Copyright © [111]

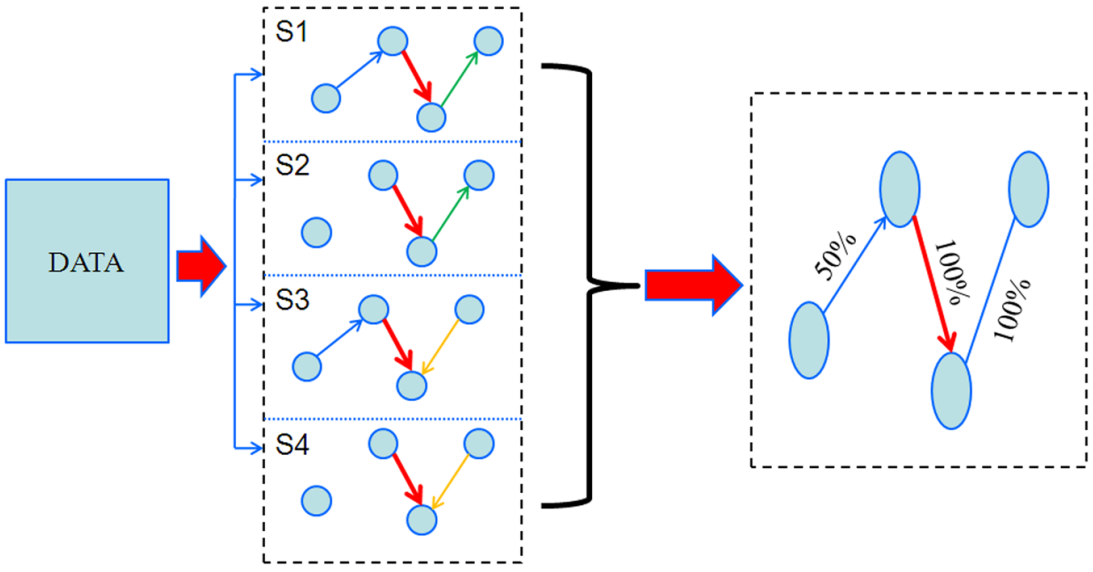


FIGURE 4.8: **Bayesian Networks Averaging**: From Data(left) using Multiple Searches, we have four DAGs. Averaging Them to Get the Final Network. Edge Assign 100% Means the Edge Appeared in the Results of All Runs

4.5 Performance Comparison of the Structure Learning Bayesian Network Algorithms Using Gene Expression Data

Many *structure learning* methods have been proposed in the literature, and it is important to understand their relative merits and shortcomings. They can be categorized as either conditional independence (CI) test-based methods or scoring-based methods. The CI-based methods analyze the dependence and independence relationships among variables via CI tests and construct the networks that characterize these relationships. The scoring-based methods consist of two components: (1) a scoring function that assesses how well a network fits the data and (2) a search method to find networks with high scores.
In this section we apply currently available Structure learning algorithms on actual microarray data to obtain a better understanding of their relative strengths and weaknesses on the system biology community and we have carried out a series of experiments to evaluate their behavior from different perspectives. The structure learning algorithms were used in this comparison are: K2 algorithm[5], Markov Chain Monte Carlo (MCMC) [112], Bayesian Network Power Constructor (BNPC) [113] and Greedy Search in the Markov Equivalent Space (GSMES) [114]. An overview of these algorithms is presented in [113].

4.5.1 K2 algorithm

The K2 Algorithm [109] is a greedy search algorithm that learns the network structure of the BN from the data presented to it. It attempts to select the network structure that maximizes the network's posterior probability given the experimental data. The K2 algorithm reduces this computational complexity by requiring a prior ordering of nodes as an input, from which the network structure will be constructed. The ordering is such that if node X_i comes prior to node X_j in the ordering, then node X_j cannot be a parent of node X_i. In other words, the potential parent set of node X_i can include only those nodes that precede it in the input ordering.

4.5.2 MCMC

Markov chain Monte Carlo (MCMC) methods, are a class of algorithms for sampling from probability distributions based on constructing a Markov chain that has the desired distribution as its equilibrium distribution. The state of the chain after a large number

of steps is then used as a sample from the desired distribution. The quality of the sample improves as a function of the number of steps. We can use a Markov Chain Monte Carlo (MCMC) algorithm called Metropolis-Hastings (MH) to search the space of all DAGs [112].

4.5.3 BNPC

The BN Power Constructor (BNPC), uses independence tests and mutual information [113]. This algorithm has a three-phase operation: drafting, thickening, and thinning. In the first phase, the algorithm computes mutual information of each pair of nodes as a measure of closeness, and creates a draft based on this information. In the second phase, the algorithm adds arcs when the pairs of nodes are not conditionally independent on a certain conditioning set. In the third phase, each arc is examined using conditional independence tests and will be removed if the two nodes of the arc are conditionally independent.

4.5.4 GSMES

Recent works have shown the interest of searching in the Markov equivalent space . It has proved that a greedy search in this space (with an equivalent score) is more likely to converge than in the DAGs space [114]. This method works in two steps. First, it starts with an empty graph and adds arcs until the score cannot be improved, and then it tries to suppress some irrelevant arcs.

4.5.5 The dataset

A powerful approach to test our understanding of gene regulatory networks is to build new networks from scratch in an approach called synthetic biology(Figure4.9). Then we could compare model predictions with networks output. This approach allows us to investigate in depth the effect of noise, data size and hidden variables in the form of unobserved processes on the reconstruction of gene regulatory network[115].

The structure learning algorithms was tested with synthetic data samples randomly generated from Raf signaling network, depicted in Figure 1. The random generation of data samples was done to ensure the robustness of the algorithms. We used the sampling function which was implemented in Bayesnet Toolbox [94]. Raf network includes 11 nodes and 20 arcs. Raf is a critical signalling protein involved in regulating cellular proliferation in human immune system cells. The deregulation of the Raf pathway can

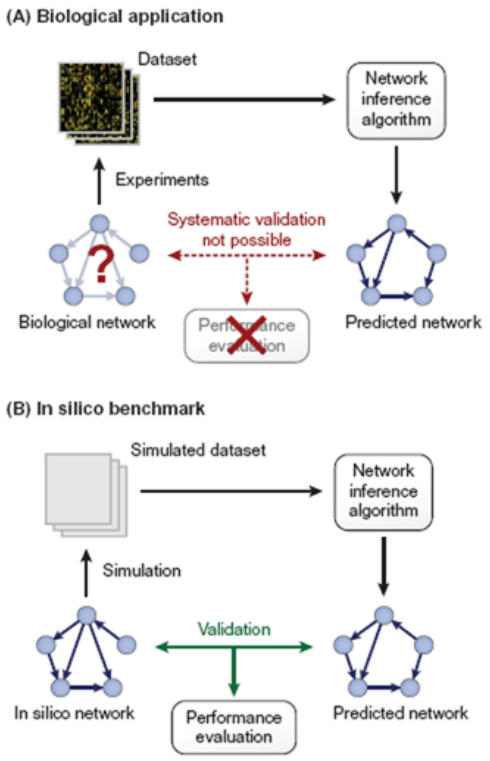


FIGURE 4.9: Validation strategies for network inference methods. (A) The true network structure of biological gene networks is in general unknown or only partly known, which hinders systematic performance evaluation. (B) Since the structures of in silico networks are known, predictions can be validated. Figure Copyright © [116]

lead to carcinogenesis, and the pathway has therefore been extensively studied in the literature [117].

4.5.6 Comparison Methodology

The comparison methodology used in this paper is similar with the method was used in [108]. The existence of the known network structures allows us to define three important terms, which indicate the performance of the algorithm (in terms of the number of graphical errors in the learnt structure).

- Correct edges(C): Edges present in the original network and in the learnt network structure.
- Missing edges (M): Edges present in the original network but not in the learnt network structure.

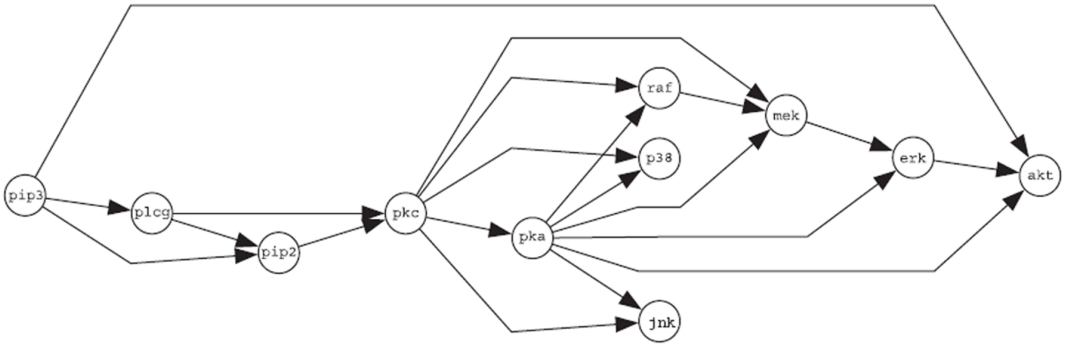


FIGURE 4.10: **Raf signalling pathway**: The graph shows the currently accepted signalling network. Nodes represent proteins, edges represent interactions and arrows indicate the direction of signal transduction. In the interventional studies, the following nodes were targeted. Activations: PKA and PKC. Inhibitions: PIP2, AKT, PKC and MEK..Figure Copyright © [117]

- Wrongly oriented edges (WO): Edges present in the learnt network structure, but having opposite orientation when compared with the corresponding edge in the original network structure.
- Wrongly connected edges (WC): Edges not present in the original network but included in the learnt network structure.

TABLE 4.3: Bayesian Structure Learning Algorithms Parameters Setting

Learning Algorithm	Parameters setting
K2 (known order)	xx[a]
K2(order from MWST[b])	max-fan-in = 2
K2 (random order)	xx
MCMC	Nsamples=100*11; burnin=5*11
GSMES	xx
BNPC	epsilon=0.05

[a] No Parameters
[b] Maximum Weight Spanning Tree [118]

simulations of these structure learning algorithms in our comparative evaluation study were carried out with the Bayesnet Toolbox [94] and Structure Learning Package [119]. The tests are carried on an Intel Core Due 1.8 GHz CPU and 1 GB RAM. Table I shows the parameters for each candidate learning algorithms. Tables 4.4,4.5 show the performances of the algorithms for the Raf networks with 1000 and 100 data samples generated randomly 100 times , respectively. Tables4.4,4.5 report the mean results (the results averaged over 100 trial runs).

Tables 4.4,4.5 show that these algorithms differ significantly in their predictability power

and how could using larger data set improve algorithms performance except for BNPC and GSMES which is against our expectation. We attempt to contact the corresponding authors to explain these results. Also the low performance of the small data set promote the importance of solving the dimensionality reduction of the gene reverse engineering algorithms where the number of experiments are minimal.

For the k2 algorithm we present the results obtained with the correct order (of which we have the knowledge, since the network structure is known), order known from Maximum Weight Spanning Tree (MWST) [120] and with the random order. The results for K2 with correct order are the optimal results one can get. K2 algorithms outperforms the learning algorithms. For its result with known order about 17 over 20 edge were covered perfectly. Also its result with random order outperform the tested algorithms. Moreover the results of k2 algorithm getting order from MWST directed the authors to develop a new algorithm to get network order. GSMES is the only method which have wrong orientation edges.

TABLE 4.4: **Bayesian Structure Learning Algorithms Comparison Results**: These Learning Algorithms Were Applied to Raf Network [117] with 1000 Data Samples Generated Randomly 100 Times.

Learning Algorithm	C	M	WO	WC
K2 (known order)	17.12	2.88	0	0.16
K2(order from MWST)	12.49	7.51	0	7.35
K2 (random order)	8.43	11.57	0	10.86
MCMC	5.86	14.14	0	13.84
GSMES	9.82	10.18	1.72	10.31
BNPC	2.35	17.65	0	5.08

TABLE 4.5: **Bayesian Structure Learning Algorithms Comparison Results With Small Data Samples**: These Learning Algorithms Were Applied to Raf Network [117] with 100 Data Samples Generated Randomly 100 Times.

Learning Algorithm	C	M	WO	WC
Learning Algorithm	C	M	WO	WC
K2 (known order)	12.82	7.18	0	2.82
K2(order from MWST)	8.81	11.19	0	6.29
K2 (random order)	5.76	14.24	0	9.51
MCMC	3.98	16	0	12.19
GSMES	9.18	10.82	1.51	8.91
BNPC	1.97	18.03	0	2.1

In this section we aim to compare the structure learning algorithms performance on a gene expression data. We see how could the data set size alter their performance. Also we show the importance of developing the correct network order algorithms. For

simulated data was used here, the true structure of the regulatory network is known; this allows us, in principle, to faithfully evaluate the prediction results. However, the sampling approach used for data-generation is a simplification of real molecular biological processes, and this might lead to systematic deviations and a biased evaluation. We can overcome this using real laboratory data.

4.6 Dream Project

DREAM is a **D**ialogue for **R**everse **E**ngineering **A**ssessments and **M**ethods. Its main objective is to catalyze the interaction between experiment and theory in the area of cellular network inference. The fundamental question for DREAM is simple: How can researchers assess how well they are describing the networks of interacting molecules that underlie biological systems? The answer is not so simple. Researchers have used a variety of algorithms to deduce the structure of very different biological and artificial networks, and evaluated their success using various metrics. What is still needed, and what DREAM aims to achieve, is a fair comparison of the strengths and weaknesses of the methods and a clear sense of the reliability of the network models they produce. The reader could refer to the recent previous DREAM conference meeting to look for new reverse engineering approaches [121]. The purpose of DREAM is not to produce the best possible network, but to evaluate the best tools for producing networks. The choice of tools depends in part on the nature of the available data. The uploaded results with DREAM2 challenge show that the networks inferred from the data differed significantly from the real network, which is precisely known. What is not known is whether the data given are, by themselves, sufficient to distinguish the networks. An interesting blinded competition on DREAM3 2008 assess the ability of scientists and their computer servants to infer networks from experimental data, by comparing their predictions to "gold-standard" networks whose structure is thought to be known. Predictors could know their ranking online.
At the basis of any modelling, including network modelling, there is a realisation and acceptance that a model describes only some properties of the 'real world' system, and ignores others. Thus it emphasizes particular aspects of reality, leaving out details that are not relevant for the purpose of the study. How far are we from being able to build realistic cell models? The availability of large-scale data sets such as microarray gene expression and genomic localisation data triggered the search for suitable approaches to model complex biological systems. By prediction gene network just from gene expression data we were ignoring the last 30 years of molecular biology literature in the design of the network. The question is how to make predictions in addition of what is known. We need also to standardize the methods of comparing gene network

models. What is not known is whether the data given are, by themselves, sufficient to distinguish the networks. Finally from the Gene Ontology project the function of about one third of all genes is still unknown for the yeast *Saccharomyces cerevisiae* despite it being one of the best-studied organisms. And even for many of the better-known genes and core processes that have been studied for decades, like the cell cycle, there is still not enough data available to exactly know all changes in concentration and activation patterns. Currently it seems not feasible to simulate even relatively simple cells like yeast. Mechanisms like RNA interference, regulated degradation of mRNAs and proteins, chemical modifications of key molecules and others might play a larger role than anticipated in current models, other processes might still be unknown. It is obvious that the separation into gene regulatory networks, metabolic networks and protein interaction networks is possible only up to a certain degree. To what extent can the transcription regulation networks be decoupled from other networks, such as signal transduction networks? We need to integrate many types of information if we want to build realistic dynamic models, however, for current modelling approaches we have to limit the complexity of the systems we are dealing with[2].

[2]http://wiki.c2b2.columbia.edu/dream/discuss/

Chapter 5

Results

In this chapter we applied our algorithm to Spellman [28] dataset and generated GRN network. The goal of this chapter are to make confident of our approach by comparing the generated network via previous algorithms(see section 5.1) and existing interactome databases(see section 5.2). Also we asses the credibility of this network by analyzing the network topology(see section 5.5.1) and finding putative modules(see section 5.5.2). The results in this chapter were submitted to the RECOMB conference [1] which will be hold in MIT, USA during 2 Dec-4 Dec 2009 and could be downloaded from:
http://home.k-space.org/FADL/Downloads/PhD/RECOMB_paper/
.

5.1 Comparison to Previous Algorithms

We need to test the algorithm performance via previous algorithms. To accomplish this task we face many challenges listed as following:

1. Network generated from some of inference algorithms are not available like the Inferelator [23] which was developed at Institute of System Biology, Seattle.
2. Part of network inference algorithm are not freely distributed.
3. Part of network inference algorithms are just applicable for certain organism like *E.Coli*. For example of these algorithms are the CLR [123] which was developed at Bioinformatics Program, Boston University.

[1] F. M. Al-Akwaa, N. H. Solouma, and Y. M. Kadah, "SSBBN: Gene Regulatory Network Construction using Spectral Subtraction Denoising, Biclustering and Bayesian Network," in The 6th Annual RECOMB Satellite on Regulatory Genomics, the 5th Annual RECOMB Satellite on Systems Biology, and the 4th Annual DREAM reverse engineering challenges., MIT, 2009.

4. Part of network inference algorithm used prior biological knowledge or constraints. For example CLR algorithm assumed that the edges in the network are just created between Transcription Factor (TF) and non TF genes. For clarity we assumed we have not any prior knowledge about genes.

5. Part of the network inference algorithm used expression dataset which is not available.

6. Part of network inference algorithm required large number of samples which is not applicable with the dataset which was used in this study.

From the above challenges we could recognize that Friedman network[2] is suitable for our comparison. Friedman [4] developed a new framework for discovering interactions between genes based on multiple expression measurements which are capable of discovering causal relationships, interactions between genes other than positive correlation, and finer intra-cluster structure.
Friedman used SparseCandidate algorithm where a relatively small number of candidate parents could be identified for each gene based on simple local statistics (such as correlation). Using SparseCandidate algorithm the search space is restricted to networks in which only the candidate parents of a variable can be its parents, resulting in a much smaller search space in which a good structure quickly hope to be found. To overcome with small sample size Friedman used bootstrap method where the perturbed versions of original data set was generated, and learned network from them. In this way many networks were collected, all of which are fairly reasonable models of the data [4].
Friedman applied his approach to the data of Spellman et al. [28], containing 76 gene expression measurements of the mRNA levels of 6177 *S. cerevisiae* ORFs. These experiments measure six time series under different cell cycle synchronization methods. Spellman et al. [28] identified 800 genes whose expression varied over the different cell-cycle stages. For computational reason Friedman applied his algorithm on only these 800 cell cycle genes. 702 genes over these 800 genes are not singleton genes which they have 1163 edges. To work with these 702 genes we have faced many problems as following:

1. Some of Friedman genes are alias for different genes, for instance ALPHA1 is alias of HMLALPHA1(YCL066W) and MATALPHA1 (YCR040W). We do not know which genes Friedman used. As this genes has low edge connectivity in Friedman network, we deleted these genes from evaluation.

[2]http://www.cs.huji.ac.il/ nirf/GeneExpression/top800/

2. We did not find any biological information for some of potential connected gene in Friedman network like EXPERIM (12 edges) and PHASE (12 edges) in the literature. Even these gene become alias for other genes, it should be mentioned in the SGD data base, so we have to delete these genes from further comparison.

3. Some gene names used by Friedman were been become retired names by SGD curator like HSN1 and HDR1. (A ' Retired name ' is a gene name that was reserved for an ORF by a member of the yeast community, but never published. A gene name reservation is good for one year. After this time, if SGD is unable to determine that the gene name has been published and unable to contact the person who made the reservation or if the submitter of the reserved gene name requests that SGD discontinue/delete the gene name reservation, such gene names become Retired names. SGD retains such gene names rather than deleting them since these names have existed in the database for a significant period of time (usually more than 2 months). When this occurs, it is documented with a note in the Locus History Page of the relevant ORF)[3].

4. Some genes were written wrongly like PST1 was wrote PTS1. It needs a lot time and effort to filter all the gene names.

5. Friedman network contains genes which were merged to other genes. for instance, YCL012W, YCL060C were merged to YCL014W & (YCL061C) respectively. All edges corresponding to the removed genes were removed from the Friedman network. For more details about merged genes see section 3.2.1.

6. Some of Friedman network genes were deleted by SGD curator like YCLX09W, so all its corresponding edges are removed from comparison. For more details about merged genes see section 3.2.1.

7. Finally, Friedman network contains genes not included in Spellman data like SNR17A which is small nucleolar RNA. We do not know why Friedman included them unless he mentioned that his network will base on Spellman cell cycle genes.

692 genes of Friedman network were passed the above filtration criteria and were considered in our evaluation.

5.2 Comparison to Literature

During the last 5 years, interactome databases are continuously increasing. The term interactome denotes the complex network (pathways) of intermolecular interactions

[3]http://www.yeastgenome.org/help/glossary.html#verified

that wires together the vast number of genes, proteins and small molecules. Information about the interactome are very promising to assist the GRN inference and or to validate the obtained networks. There are two important different type of interaction [124]as following:

- Protein-DNA interactions are those that occur between TF and their DNA binding sites. The development of large-scale experiments such as ChIP-on-chip (chromatin immunoprecipitation combined with microarray technology) allows to obtain such TF-DNA interactions (also called DNA binding location data) for a given TF. Thereby, the ChIP-on-chip experiment identifies the regions of a genome that are bound by this TF in vivo. Afterwards, this information can be used to predict its potential gene regulatory effects (i.e. its target genes).
- Protein-protein interactions (PPIs) play a major role for intercellular signaling and can be experimentally identified by methods such as yeast two-hybrid arrays. The protein interaction network in *S. Cerevisiae* is the best-studied PPI network today, but information for other organisms are continuously increasing too. Given the existing data sets for yeast proteins a total of 10.000-30.000 pairwise interactions are estimated, i.e. roughly 3-10 interactions per protein [125].

Molecular interaction information can be extracted from different sources. Pathguide [126], a so-called metadatabase, provides an overview of more than 230 web-accessible biological pathway and network databases[4]. Pathguide distinguishes 8 approximate categories based on the content of databases (see table 5.1).

The Interaction databases (table 5.1) use different identifiers to identify the same gene (GI, SwissProt, internal identifiers, etc.) requiring the resolution of synonymous names/IDs across databases. So, we want to integrate molecular interactions and other types of high-throughput data from different public databases to build biological networks automatically. For this purpose we used BioNetBuilder [51] which is an open-source client-server Cytoscape plug-in that offers a user-friendly interface to create biological networks integrated from several databases. The BioNetBuilder is available as a Java Webstart, providing a platform-independent network interface to these public databases(Figure 5.1). Figure 5.1 shows the number of interaction for the *S. Cerevisiae* from [(BIND,16244);(BioGrid,99485); (DIP,17465);(IntAct; 14331);(Interologger,5395); (KEGG,5478);(MINT,11907)].
For 692 filtered genes(see section 5.2) only 635 genes have interactions from BioNetBuilder, so the gold standard network we will compare with consists of 635 genes and 2611 edges.

[4]www.pathguide.org

TABLE 5.1: Categories of interaction databases presented in Pathguide as of 12/2008.(modified from [124])

Category	Content	# resources	Examples
Protein Protein Interaction	pairwise interaction between proteins	93	BID BIND BioGRID BRITE DIP
Metabolic Pathways	biochemical reaction in metabolic pathway	50	KEGG, GO, ExPASY , Reactome
Signaling Pathways	molecular interactions and chemical modifications in regulatory pathways	49	STKE, Reactome, TRANSPATH
Transcription Factors\ Gene Regulatory Networks	transcription factor and genes they regulate	33	GeNet, SCPD, TFe, YEASTRACT, RegulonDB
Pathway Diagrams	hyperlink pathway images	27	KEGG, HPRD, SPAD
Protein Compound Interactions	interactions between protein and compounds	19	DrugBank, PLD, TTD
Genetic Interaction Networks	genetic interactions, such as epistasis	6	BIND, BioGRID
Protein Sequence Focused	diverse pathway information in relation with sequnce data	12	TGDB MotifMap

Because Bayesian network algorithms are not able to detect self edges(i.e. Is the gene regulate itself?), we have to remove all the self regulation edges, which make the Gold Standard Network (635 genes and 2194 edges).`http://home.k-space.org/FADL/Downloads/PhD/RECOMB_p`

5.3 Network Generation

Now we applied our algorithm to these 635 Spellman genes. First, the genes were partitioning using our biclustering toolbox (BicAT-plus) [16](section 5.3.1). Second the biclusters were learned using GreedyHillClimbing learning algorithms to generate different subnetworks for each biclustering algorithms(section 5.3.2). These subnetworks were integrated to produced the whole network per each biclustering algorithm.
The generation Matlab code with all the generated networks could be downloaded from:
`http://home.k-space.org/FADL/Downloads/PhD/RECOMB_paper/bolearn_results/`

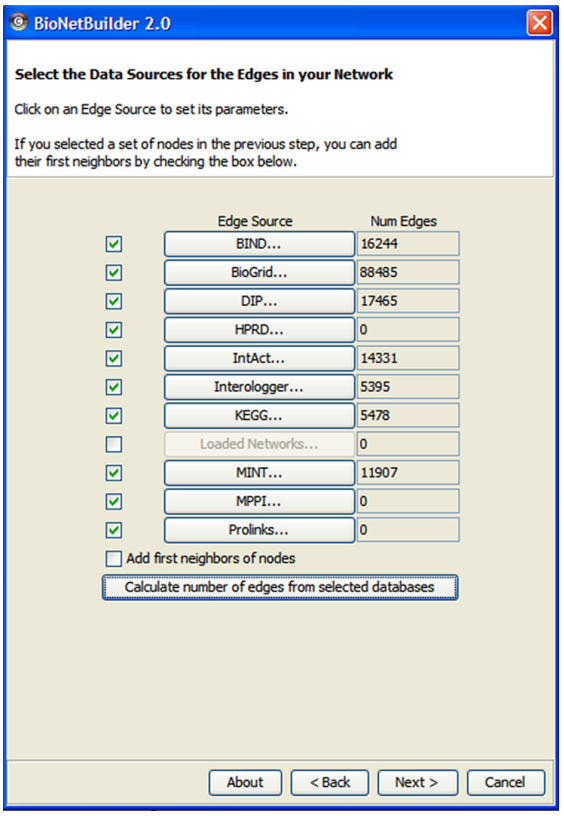


FIGURE 5.1: Gene Regulatory Network Extracted From Interactome Databases using Bionetbuilder Cytoscape Plug-in [51].

5.3.1 Biclustering Phase

We applied the biclutering algorithms implemented in our modified biclustering comparison toolbox(BicAT-plus) [16] to the Spellman experiments for these 635 genes. Table 5.2 shows biclustering algorithm parameters setting as authors recommended in their corresponding publications. The statistical comparison of the 683 produced biclusters/clusters are shown in table 5.3.
It is important to focus on how ISA genes coverage percentile using Spellman dataset(91%)(See Table 5.3) is larger than using Gasch dataset(25%)(See Table 3.11) for the same parameter setting.

5.3.2 Learning Phase

In this phase, we first learn the produced biclusters/clusters from the biclustering phase to get small overlapped networks or submodule networks. Then, we integrate these submodule networks to get the whole network for each bicluster algorithm. We used

TABLE 5.2: Parameters setting of biclustering algorithms implemented in BicAT-Plus toolbox [16] applied to Spellman [28] cell cycle data set. for more details about these parameters please see corresponding publication

Algorithm	Parameters	Parameter Description
ISA	tg=2.0	Genes threshold level
	tc = 2.0	condition threshold level
	SN= 500	No of seeds
CC	Delta=0.5	Maximum number of accepted score
	Alpha=1.2	Scaling factor
	M=100	Number of bicluster to be found
OPSM	l = 100	Number of passed models for each iteration
BIMAX	Minr=10	Minimum row size of resulting bicluster.
	Minc=5	Minimum column size of resulting bicluster
	M=100	Number of Bicluster to be found.
	Dth= -0.0950	Discredited Threshold
K-means	M=100	Number of Bicluster to be found
	IN=100	Number of Iteration
	RN=10	Number of replication
	DM=ED	Distance Metric is Euclidean Distance
HCL	M=100	Number of Bicluster to be found
	LM=AL	Linkage Mode is Average Linkage
	DM=ED	Distance Metric is Euclidean Distance
Bivisu	NT=0.5819	Data Noise threshold
	% NR=1.57	Minimum % of rows
	NC=5	Minimum number of columns
	O%=25%	Maximum overlap allowed
MSBE	alpha = 0.4	similarity threshold
	beta = 0.5	bonus similarity threshold
	gamma=1.2	The threshold of the average similarity score
SAMBA	MHS=100	Maximal memory allocated for hashing stage
	KHS1=4	Maximal kernel size in the hashing stage
	PC=100	Minimal number of responding probes per condition
	KHS2=4	Minimal kernel size in the hashing stage
	O%=25%	Maximum overlap between two biclusters

TABLE 5.3: Statistical Comparison of Biclusters Produced by Applying Bicluster Algorithms Implemented in BicAT-Plus [16] to Spellman [28] Cell Cycle Dataset with Parameter Settings Shown in Table 5.2

Biclustering Algorithm	No of Biclusters	BiclustersClusters Size		GeneCoverage%	ConditionCoverage%
		Min	Max		
ISA	9	50 x 35	155 x 37	25	97
CC	69	11 x 5	2259 x 134	100	100
OPSM	2	11 x 15	575 x 6	88.5	32.9
BiVisu	100	27 x 142	99 x 52	55	100
Kmeans	100	20 x 173	50 x 173	100	100

GreedyHillClimbing search algorithm and BDe Scoring Function implemented in Bioleam [127] at Department of Biological Sciences, Columbia University.
Table 5.4 shows the result of integration each sub-networks generated per each bicluster algorithm. They differ significantly in the number of interaction edges. In the next section we compare the performance of these network via the gold standard network(section 5.2) and Friedman network(section 5.1).
The result in Table 5.4 looks unreasonable. From Table 5, Kmeans and CC covers 100%

TABLE 5.4: Edge Number of Networks Generated from Biclustering Algorithms Implemented in BicAT-Plus [16] toolbox

Network source	Number of Edges
Friedman network	947
K-means network	380
ISA network	2558
OPSM network	220
CC network	590
Bivisu network	1515
MSBE network	735
SAMBA network	1611

genes while ISA covers just 25% genes. However, Kmeans and CC networks have quite less number of edges than ISA has [5]. The explanation for this conflict that k-means and CC produce biclusters with size equal all dataset genes. This cluster have no biological meaning and even any existing learning algorithms ,restricted to learn with 100 genes as maximum. So we have to neglect these large biclusters from the learning stage.

5.4 Evaluation Methodology

After we get all these networks, we need a methodology to fairly score each network and conclude the performance of our algorithm via other algorithms and existing increasing databases.
receiver operator characteristic (ROC) curve are commonly used to present results for binary decision problems in machine learning, which show how the number of correctly classified positive examples varies with the number of incorrectly classified negative examples. However, when dealing with highly skewed datasets, precision-recall (PR) curves give a more informative picture of an algorithm's performance. PR curves have been cited as an alternative to ROC curves for tasks with a large skew in the class distribution [128].

[5]This is part of RECOMB revision on our paper

We can consider Gold standard Network produced at section 5.2 as a space with T elements. These elements belong to one of two classes: the class of positive examples(there is edges between two genes), with P elements, and the class of negative examples(there is no edges between these two genes), with N elements. Clearly T=P+N. The fact that we know the class to which each example belongs makes this space a gold-standard.
A prediction is made in the form of an ordered list of L samples taken from our gold space. This list is ordered such that the examples at the top of the list are the ones which we have the higher confidence that they belong to the positive class. We will assume that the list contains TPL true positive predictions and FPL false positive predictions. Clearly L = TPL + FPL. We now add in random order the remaining T-L samples (on which no prediction was made) to the bottom of the original list with L examples. We want to compute the precision and recall corresponding to the prediction that the k (k¿L) first samples in the resulting list are positive [59].
we have to define important score terms here:

$$TPR(Sensitivity) = \frac{TP}{TP+FN}$$
$$FPR = \frac{FP}{FP+TN}$$
$$Recall = \frac{TP}{TP+FN}$$
$$Precision = \frac{TP}{TP+FP}$$
$$Specificity = \frac{TN}{TN+FP}$$

In ROC space, one plots the False Positive Rate (FPR) on the x-axis and the True Positive Rate (TPR) on the y-axis. The FPR measures the fraction of negative examples that are misclassified as positive (FPR = 1-specificity). The TPR or the sensitivity measures the fraction of positive examples that are correctly labeled or (the fraction of correctly identified interactions in relation to the number of expected interactions). In PR space, one plots Recall on the x-axis and Precision on the y-axis. Recall is the same as TPR, whereas Precision measures that fraction of examples classified as positive that are truly positive or (the fraction of correctly identified interactions out of all predicted interactions) [128]. Further commonly used scores is the false discovery rate (FDR = 1-precision) and the specificity which measures the proportion of non-existing edges (number of potential edges number of inferred edges) which are correctly identified.
Note that each of the above scores is calculated only from two numbers out of FN, FP, TP, TN, i.e. each score is hardly informative when used alone. For instance, an inferred fully connected network will result in a recall equal to 1, but is obviously not biologically meaningful [87]. We used the evaluation script algorithm was used in DREAM2 [6](Figure 5.2) to compute the area under ROC curve (AUROC), and area under

[6]http://wiki.c2b2.columbia.edu/dream/index.php/DREAM2conf

PR curve (AUPR). The evaluattion Matlab code were used here could be downloaded from: http://home.k-space.org/FADL/Downloads/PhD/RECOMB_paper/comparsion_methdology/. An AUROC close to 0.5 corresponds to a random forecast, AUROC ≤ 0.7 is considered poor, AUROC) ≥ 0.8 good [87].

Figure 5.3 and Table 5.5 show the performance of the network generated from biclustering algorithms via gold standard network and Friedman network.

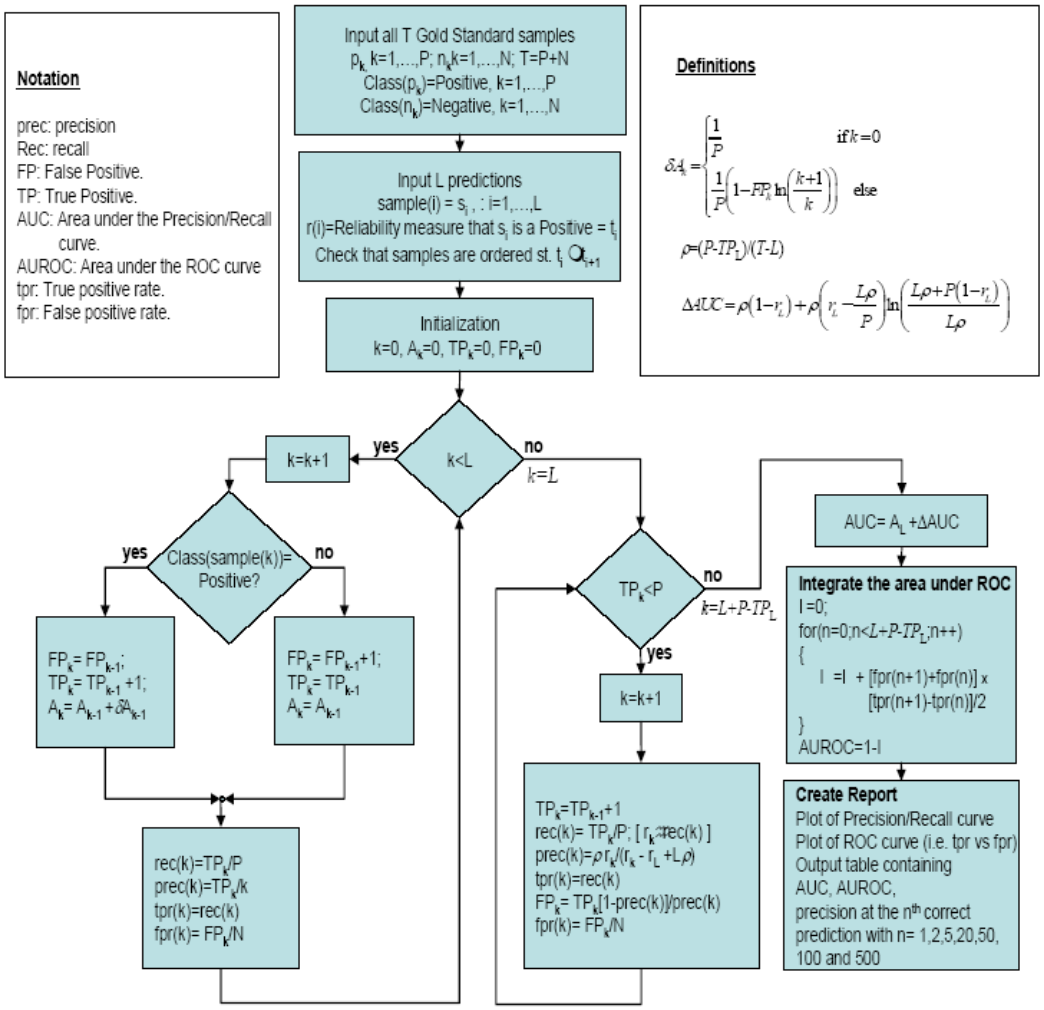


FIGURE 5.2: Pseudocode of the process to Evaluate the Predictions of the Dialogue for Reverse Engineering Assessments and Methods (DREAM2) challenges [59].

Inspection Figure 5.3 and Table 5.5 reveal that neither the generated networks from each bicluster algorithms nor the generated network from the whole biclusters integration perform well.

There is important note to be considered when interpreting the results of this comparison. First the interactions documented are either physical or genetic, which implies that they may not be direct interactions. The precision may be lower than the actual

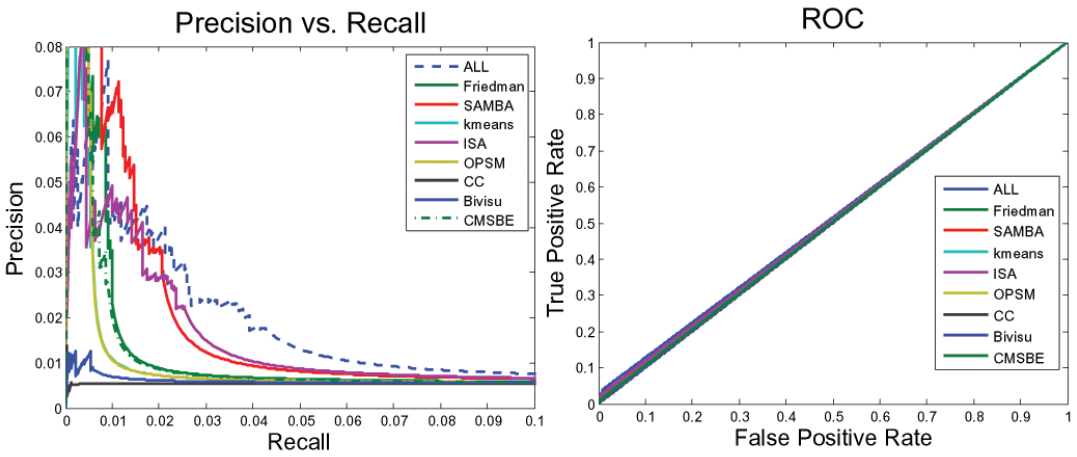


FIGURE 5.3: Performance of the Networks Generated from Corresponding Biclustering Algorithms(ALL: network produced from integrating all bicluster subnetworks ; Friedman: Friedman Network [4]; SAMBA:network generated from SAMBA [72] biclusters; Kmeans:network generated from k-means clusters; ISA:network generated from ISA [21] biclusters; OPSM:network generated from OPSM [19] biclusters; CC:network generated from CC [11] biclusters; Bivisu:network generated from Bivisu biclusters [83]; CMSBE:network generated from MSBE biclusters [70]).

TABLE 5.5: Statistical Comparison of Networks Produced from Biclustering Algorithms via Friedman Network and Gold Standard Network. EdgeCount: number of edges; TP:number of true positive edges; TN: number of true negative edges; FP:number of false negative edges; AUROC:area under ROC curve; AUPR:area under precision recall curve

Methods	EdgeCount	TP	FP	TN	FN	AUROC	AUPR
Gold	2194	2194	0	400396	0	1	1
ALL	5440	94	5346	395050	2100	0.5148	0.0073
SAMBA	1611	46	1565	398831	2148	0.5085	0.0072
ISA	2558	56	2502	397894	2138	0.5097	0.0067
OPSM	220	12	208	400188	2182	0.5025	0.0067
Friedman	947	22	925	399471	2172	0.5039	0.0065
CMSBE	735	20	715	399681	2174	0.5037	0.0063
K-means	380	13	367	400029	2181	0.5025	0.0061
Bivisu	1515	13	1502	398894	2181	0.5011	0.0055
CC	590	3	587	399809	2191	0.5000	0.0054

precision since links may be missing in the interactome databases; and the recall may be lower than the actual recall in part because some of the links reported in the interactome databases may be indirect rather than the direct [129].
Second, some presently unsupported edges in the constructed network may find experimental evidence in the future. Therefore, these unsupported edges are not necessarily false ones [108].
For the above reasons the False Positive (FP) edges could be consider as True Positive (TP) if it has strong evidence in the literature (gold network). for example if the inference network include edge between gene1 and gene3 which does not exists in gold network and if these two genes connected indirectly via another intermediate gene like gene2 we can now consider the edge between gene1 and gene3 as true positive edge.
Table 5.6and figure 5.3 show the biclustering networks performance improvement after taking in our consideration the above evaluation modification. Furthermore they show how almost the false positive edges in these networks have an evidence in the gold network.

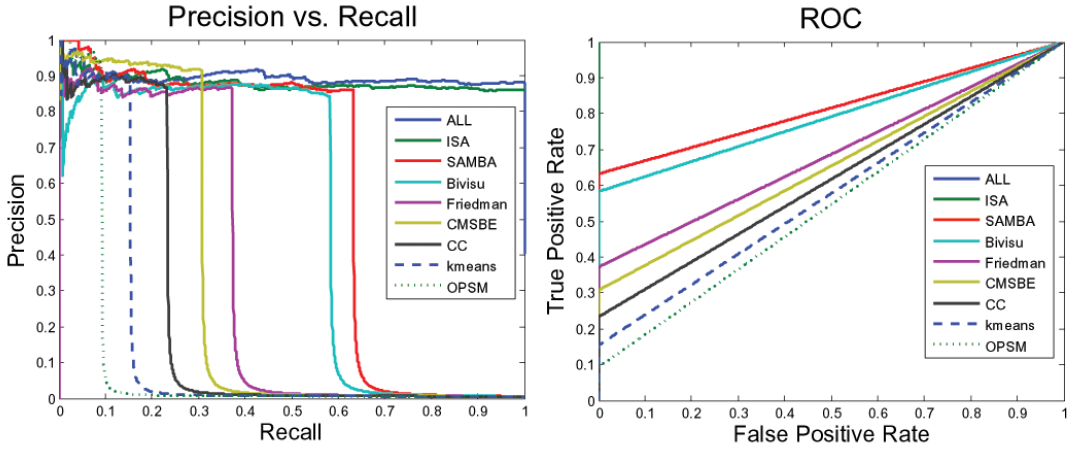


FIGURE 5.4: ROC and PR curves of Networks Produced from Biclustering Algorithms when False Positive Edges could be Consider as True Positive Edges if it has Strong Evidence from the Gold Network(see the text).

Also it should be mentioned that as we expected the sparseness nature of gene regulatory network,make using biclustering techniques(ISA, SAMBA, Bivisu) outperform the performance of the Friedman network. This will open the usage of biclustering algorithms to overcome the dimensionality reduction of the GRN inference problem.
Table 5.6 column 8 shows the percentage of false positive edges per each algorithm which could be consider as true positive (i.e have evidence in the gold standard network). for example 85% of ISA false positive edges have an evidence from the gold network.

TABLE 5.6: Statistical Comparison of Networks Produced from Biclustering Algorithms via Friedman Network and Gold Standard Network using new Evaluation Criteria (see the text).

Methods	#Edges	TP	FP	TN	FN	FP to TP	(FP to TP)%	AUROC	AUPRC
Gold	2194	2194	0	400396	0	0	XX	1.0000	1.0000
ALL	5440	94	5346	395050	2100	4623	86.48	0.9997	0.8926
ISA	2558	56	2502	397894	2138	2141	85.57	0.9996	0.8795
SAMBA	1611	46	1565	398831	2148	1340	85.62	0.8156	0.5709
Bivisu	1515	13	1502	398894	2181	1265	84.22	0.7910	0.5097
Friedman	947	22	925	399471	2172	794	85.84	0.6858	0.3316
CMSBE	735	20	715	399681	2174	653	91.33	0.6533	0.2969
CC	590	3	587	399809	2191	507	86.37	0.6161	0.2160
k-means	380	13	367	400029	2181	323	88.01	0.5765	0.1477
OPSM	220	12	208	400188	2182	190	91.35	0.5460	0.0965

It should be to note that even ISA network outperforms SAMBA network, number of produced biclusters from SAMBA and the percentage of the genes recovered by SAMBA are smaller than ISA(see table 5.3).

Figure 5.5 suggests the performance equality of the ISA network performance using NormalGamma and BDe scoring function.

Figure 5.6 demonstrates the ISA network performance using SparseCandidate with different size of the candidate sets and GreedyHillClimbing algorithms. Decreasing or increasing the size of the candidate sets beyond 5 worse the network performance.

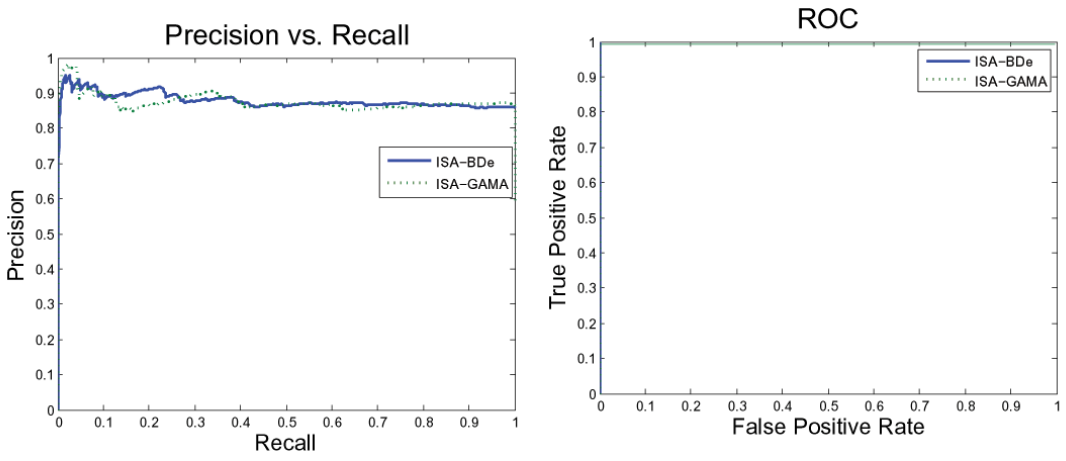


FIGURE 5.5: Performance of the ISA network using BDe (solid line) and NormalGamma(dotted line) Scoring Function.

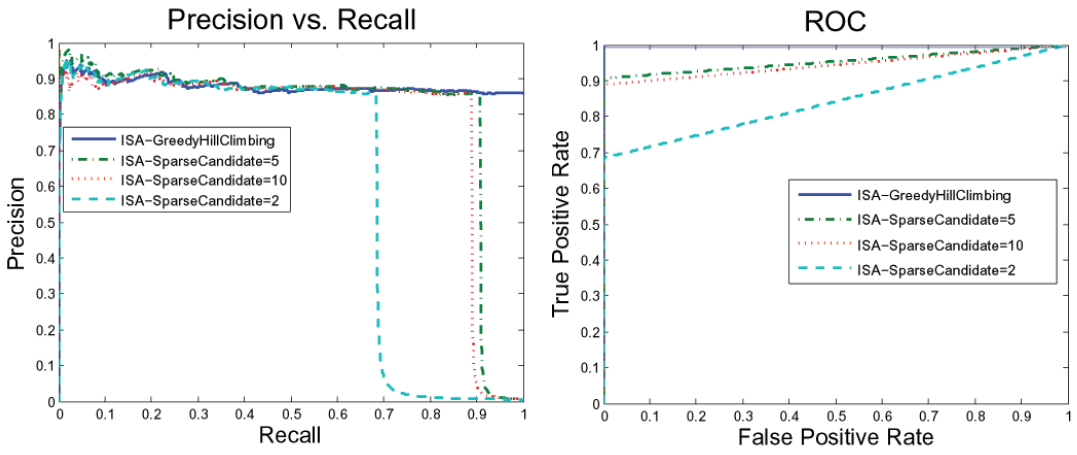


FIGURE 5.6: Performance of the ISA network using Greedy Hill Climbing and Sparse-Candidate Learning Algorithm with Different Size of the Candidate Sets.

5.5 Network Analysis and Validation

There are many reasons to perform analyses on GRN, and many methods can be used. One unique problem with transcriptomic datasets is that they are "short and wide," meaning that many characteristics are measured on relatively few samples. For example, current microarrays offer the quantitation of up to 60,000 expressed sequence tags (ESTs) in any given sample, but current costs may limit a single experiment to 10 to 100 samples [52]. Because of this problem, these data sets are essentially underdetermined, meaning that there are many correct ways to mathematically describe the clusters and genetic regulatory networks contained within them. Thus, some computational validation is required so that computationally sound but biologically spurious or improbable hypotheses are screened out.

One of the greatest methods to validate the generated network is to assess its accumulated information using information published in the known biological literature.

From the discussion in the previous sections, we found that the network produced from ISA subnetworks has significance performance comparable with Friedman network(section 5.1) and biological literature(section 5.2).

Figure 5.7 shows circular layout of ISA network(570 nodes and 2324 edges) generated by Cytoscape [130].

Figure 5.8 shows node color and size mapping using VizMapper Cytoscape plug-in [130]

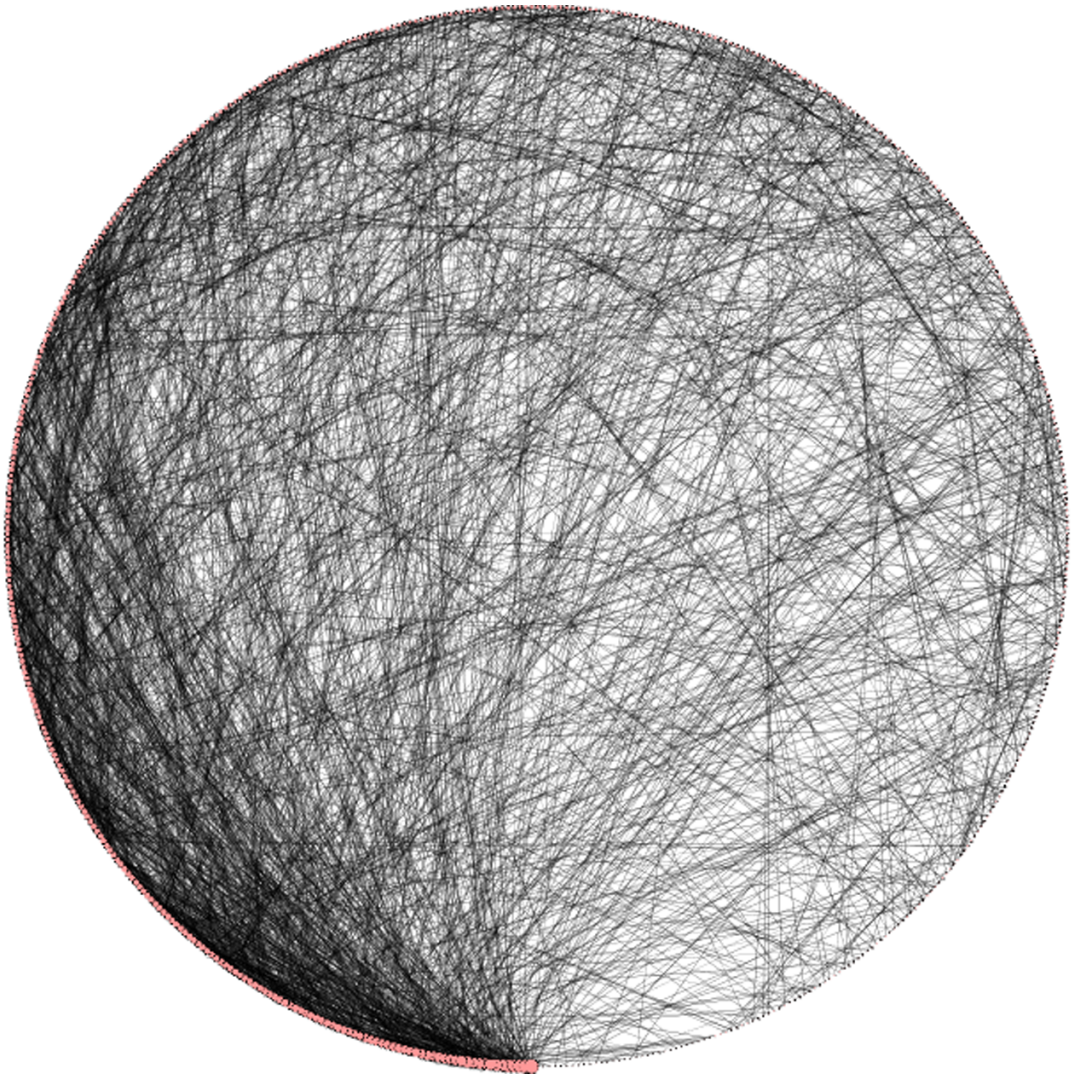


FIGURE 5.7: ISA Gene Regulatory Network, Generated from ISA Subnetworks.

5.5.1 Network Topology

The characterization of biological networks by means of graph-topological properties has become very popular for gaining insight into the global network structure. In this section we compare the topological parameters of ISA networks and gold network. We used NetworkAnalyzer Cytocape [131] plug-in developed at Max Planck center. There are many important topological parameters, like: number of nodes, edges, and connected components, the network diameter, radius, density, centralization, heterogeneity, clustering coefficient, and the characteristic path length. The definition and biologically importance of these parameters could be found in [7] and beyond this research. Closeness centrality is a measure of how fast information spreads from a given node to other reachable nodes in the network

[7] http://med.bioinf.mpi-inf.mpg.de/netanalyzer/help/2.6.1/index.html

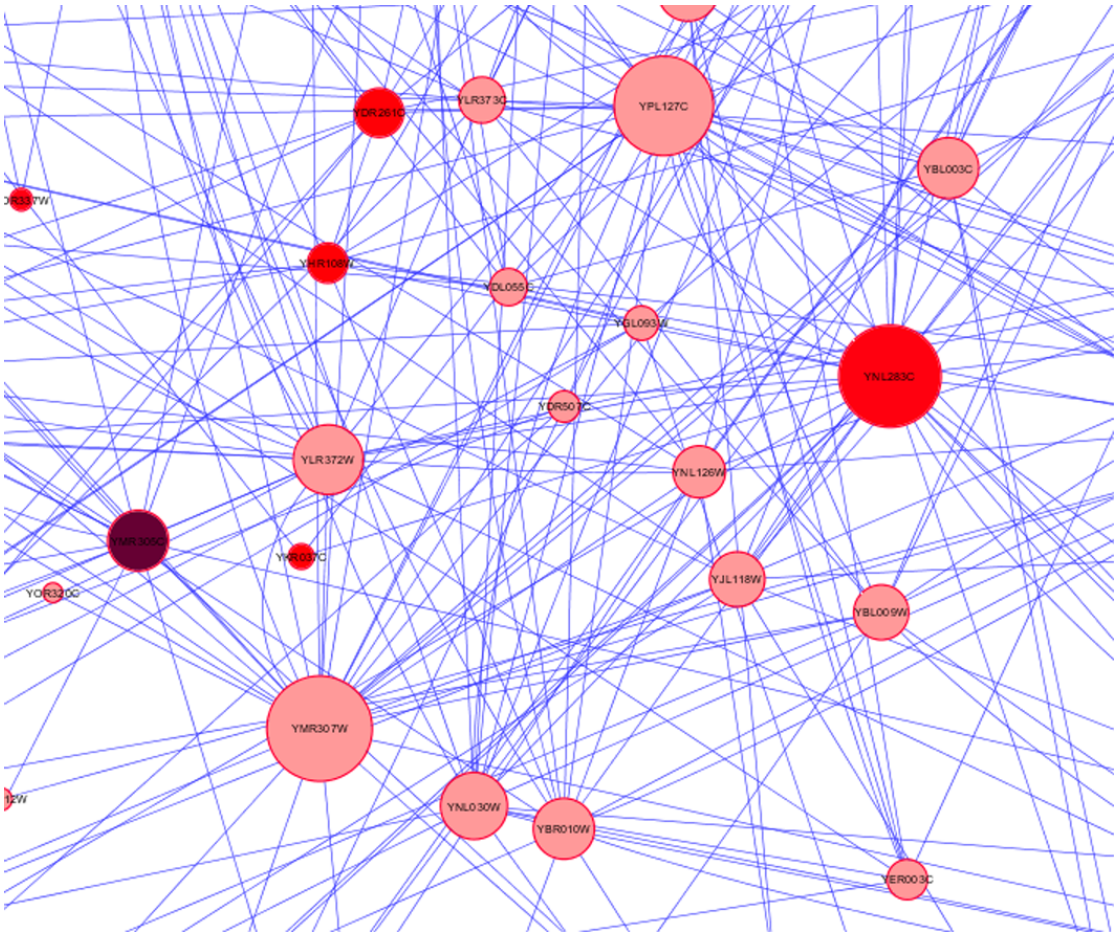


FIGURE 5.8: ISA network: Gene Degree(number of edges in and out)Mapped as Node Size and Gene Expression Values Mapped as Node Color with Adjacent Color Bar.

we can divided these parameters on to:

- **Parameters related to shortest paths** The length of a path is the number of edges forming it. There may be multiple paths connecting two given nodes. The shortest path length, also called distance, between two nodes n and m is denoted by L(n,m). The network diameter is the largest distance between two nodes. If a network is disconnected, its diameter is the maximum of all diameters of its connected components. The diameter can also be described as the maximum node eccentricity (Node Eccentricity= maximum shortest path between node i and other nodes in network). The network radius, on the other hand, is the minimum among the non-zero eccentricites of the nodes in the network. The average shortest path length, also known as the characteristic path length, gives the expected distance between two connected nodes. The closeness centrality Cc(n) of a node n is defined as the reciprocal of the average shortest path length.

Closeness centrality is a measure of how fast information spreads from a given node to other reachable nodes in the network

- **Parameters related to neighborhood** The neighborhood of a given node n is the set of its neighbors. The connectivity of n, denoted by kn, is the size of its neighborhood. The average number of neighbors indicates the average connectivity of a node in the network. A normalized version of this parameter is the network density. The density is a value between 0 and 1. It shows how densely the network is populated with edges (self-loops and duplicated edges are ignored). A network which contains no edges and solely isolated nodes has a density of 0. In contrast, the density of a clique is 1.

Table 5.7 shows the topological parameters of ISA network via gold network. It confirm the credibility of our algorithm.

TABLE 5.7: Topological Parameters of ISA Network and Gold Network using Network-Analyzer [131]

Parameters	Gold Standard	ISA Network
Network Diameter	8	9
Network Density	0.011	0.012
Avg no of neighbors	6.91	6.933

5.5.2 Finding Network Module

It was observed that highly interconnected, or dense, regions of the network may represent complexes [132]. One of the greatest methods to validate network is by assess its accumulated information with the known biological literature. Clustering algorithm was used to identify molecular complexes or modules in a large protein interaction network through network connectivity [133]. A network module is a group of nodes in the network that work together to execute some common function. We used in this section the MCODE Cytoscape plug-in [133] "Molecular Complex Detection" developed by Gary Bader at the University of Toronto. MCODE is a novel graph theoretic clustering algorithm, that detects densely connected regions in large protein-protein interaction networks that may represent molecular complexes.it is based on vertex weighting by local neighborhood density and outward traversal from a locally dense seed protein to isolate the dense regions according to given parameters.
Running MCODE on ISA network recovered 39 modules. Figure 5.9 shows the highly scores modules with the number of nodes, edges and the topology of each discovered

modules.

A significance number of modules with high score and small number of nodes and edges. To validate the significance of the recovered modules, the nodes of these modules are a portion of a complex, then there should be some process in which they all operate. Thus, if we explore Gene Ontology (GO) term enrichment using any of functional enrichment tools like BINGO [78], we should see some biological process with significant enrichment for these nodes [130].

Figure 5.10 demonstrate module with rank 1 functional enrichment analysis using BINGO[78], which indicates that four gene of this module share three related biological process which are Chromatin assembly or disassembly,DNA Packaging and Establishment and/or Maintenance of Chromatin Architecture (see Figure 5.10).

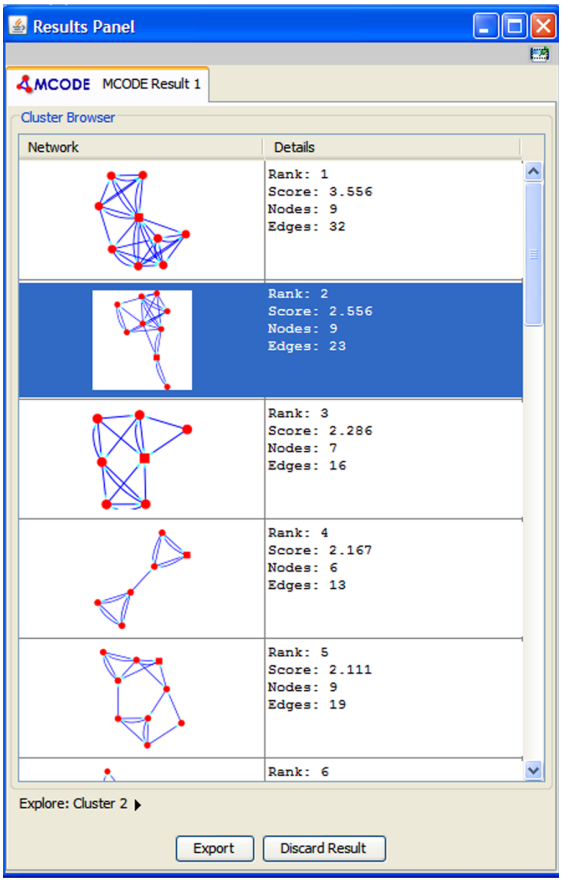


FIGURE 5.9: The Putative Complexes Through Network Connectivity from ISA Network using MCODE [133]. Module Credibility is Increased as the Increasing of the Module Score.

Lastly, our approach showed improvement of network accuracy. This is because of the sparseness nature of real gene regulatory network and also the noise decrement of gene

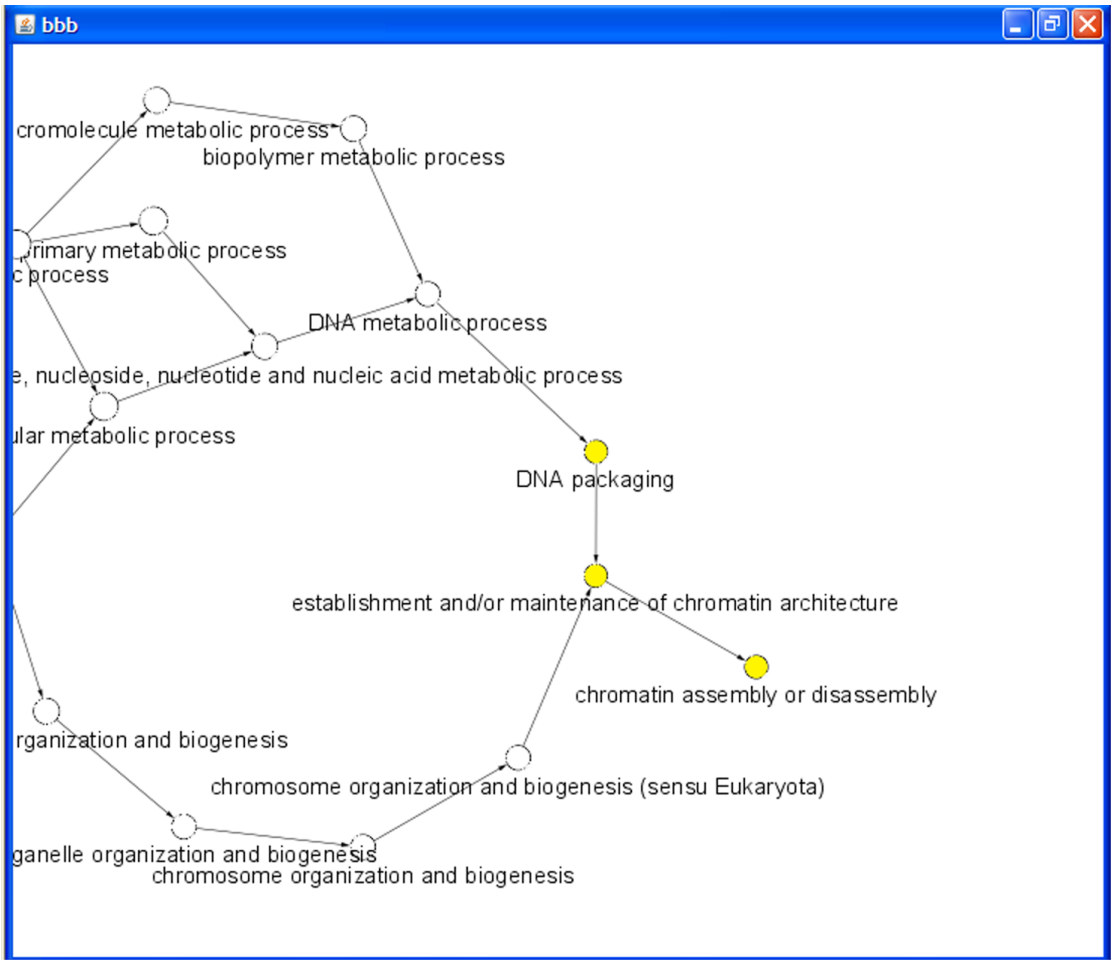


FIGURE 5.10: Biological Process Function Category of Highly Ranked Module Discovered from ISA Network(See Figure 5.9) .

expression data within each bicluster. Also Bayesian network allows to deal with the noises that are inherent in this data; and to model the hidden variable in the data.

Chapter 6

Conclusion and Future Work

Understanding gene interactions in complex living systems can be seen as the ultimate goal of the systems biology revolution. Hence, to fully understand disease ontology and to reduce the cost of drug development gene regulatory network (GRN) have to be constructed. During the last decade, many GRN inference algorithms that are base on genome-wide data have been developed to unravel the complexity of gene regulation. Transcriptomic data measured by genome-wide DNA microarrays are traditionally used for GRN modelling. This is because RNA molecules are easily accessible in comparison to proteins and metabolites. One of the major problems with microarrays is that a dataset consists of relatively few time points with respect to a large number of genes. The dimensionality and high degree of noise are interesting problems in GRN modelling. The most common and important design rule for modelling gene networks is that their topology should be sparse. This means that each gene is regulated by only a small number of other genes.

In this thesis a new gene regulatory network (GRN) construction system from microarray large dataset and prior biological information was proposed. As we expected the sparseness nature of GRN make biclustering techniques to show significance results compared to Friedman network. In this thesis we show the impact of using biclustering algorithms in GRN construction. A sophisticated filtration procedure(data filtration,missing values imputation, normilization, discretization) were used to reduce the number of expression profiles to some subset that contains the most significant genes. Also, we used a novel denoising method (Spectral Subtraction) which accurately may account for the low SNR and able to suppresses random noise or removes some of its components. It is clear from comparison SS with previous denoising methods like Multi-Wavelet that the spectral subtraction denosing method outperforms the Multi-Wavelet method and offering a substantial improvement of the SNR.

Also, The Biclustering comparison toolbox (BicAT-Plus) implemented in this thesis confirms that the bicluster and cluster algorithms can be considered as integrated modules; there is no certain algorithm that can recover all the interesting patterns, what algorithm A success to recover in certain data sets, Algorithm B might fail, and vice verse. we can identify the highly enriched bi/clusters of the whole compared algorithms, Integrating them to solve the dimensionality reduction problem of the gene regulatory network construction.
Moreover, the study in this thesis confirms the ability of the Bayesian Networks(BNs) structure algorithms to capture the structure of the real gene regulatory network. BNs allow to deal with the noises that are inherent in experimental measurements; and to model the hidden variable in the data.
Surprisingly, the generated networks from this study shows sufficient accuracy when comparing it via previous works and existing biological databases like BIOGRIDE.
Also, network validation of the generated network using popular vaildation algorithms like MCODE and NetworkAnalyzer adds more credibility on our algorithm. The data used in vaildation step not used for modelling. On other hand putative modules were recovered from our method, which suggest more analysis to recover and test unknown complex module.
As the consequence of development and emergence of new high throughput data technology, therefore it seems overly ambitious to imagine that within the next decade we will be able to generate robust predictive models that are able to accurately predict the interactions of thousands or millions of heterogeneous molecules and the ways in which they modulate the transcription of RNA and the translation of messenger RNA (mRNA) into protein and the subsequent functions of these proteins [52].
Several areas of work for future research are indicated:

1. Enriching the BicAT-Plus with more comparative methodologies beside GO. For example, KEGG and promoter analysis by identifying the transcription factors of the bi/clustered genes.

2. Extending the BicAT-plus to provide users with multiple export options for the interested enriched bi/clusters.

3. Embeding the BicAT-plus as a plug-in in the cytoscape platform which is an open source bioinformatics software for visualizing molecular interaction networks and biological pathways and integrating these networks with annotations, gene expression profiles and other state data. Thus, very promising challenge is to get use of the highly enriched bi/clusters identified by the BicAT-Plus in solving these integrated networks in the cytoscape.

4. incorporating the learning stage to the BicAT-Plus.

5. Integration more Biological data such as ChIP-chip Genome localization data and Protein similarity data. The need for large numbers of data points, and many different conditions, implies that successful modeling efforts will probably have to use data from different sources like from different high-throughput data sources, mainly microarray based gene expression analysis, promoter sequence information, Chromatin immunoprecipitation (ChIP) and protein-protein interaction assays.

6. Using New emerging learning algorithms Like evolutionary algorithms.

7. Using signal processing techniques to Remove non informative misleading profile genes. The tools to extract knowledge from data collected from all of these types of experiments are still in their infancy, and novel tools are still needed to sift through the enormous databases of simultaneous RNA expression to find the true nuggets of related function.

Bibliography

[1] J. Smith and E. H. Davidson, "Gene regulatory network subcircuit controlling a dynamic spatial pattern of signaling in the sea urchin embryo," *Proceedings of the National Academy of Sciences*, vol. 105, no. 51, pp. 20 089–20 094, 2008. [Online]. Available: http://www.pnas.org/content/105/51/20089.abstract

[2] G. Cuccato, G. G. Della, and D. d. . Bernardo, "Systems and Synthetic biology: tackling genetic networks and complex diseases," *Heredity*, vol. 102, pp. 527–532, March 2009.

[3] S. Kauffman, "Homeostasis and differentiation in random genetic control networks," *Nature*, vol. 224, pp. 177–178, 1969.

[4] N. Friedman, M. Linial, I. Nachman, and D. Pe'er, "Using Bayesian Networks to Analyze Expression Data," *Journal of Computational Biology*, vol. 7, no. 3-4, pp. 601–620, 2000, pMID: 11108481. [Online]. Available: http://www.liebertonline.com/doi/abs/10.1089/106652700750050961

[5] C. Wolfe, I. Kohane, and A. Butte, "Systematic survey reveals general applicability of "guilt-by-association" within gene coexpression networks," *BMC Bioinformatics*, vol. 6, no. 1, p. 227, 2005. [Online]. Available: http://www.biomedcentral.com/1471-2105/6/227

[6] T. Chen, L. H. Hongyu, and G. M. Church, "Modeling gene expression with differential equations," in *4th Pacific Symposium on Biocomputing*, Big Island of Hawaii, Hawaii, USA, 1999.

[7] P. D'haeseleer, X. Wen, S. Fuhrman, and R. Somogyi, "Linear modeling of mRNA expression levels during CNS development and injury," in *4th Pacific Symposium on Biocomputing*, Big Island of Hawaii, Hawaii, USA, 1999, p. 4152.

[8] P. D'haeseleer, S. Liang, and R. Somogyi, "Genetic network inference: from co-expression clustering to reverse engineering," *Bioinformatics*, vol. 16, no. 8, pp. 707–726, 2000. [Online]. Available: http://bioinformatics.oxfordjournals.org/cgi/content/abstract/16/8/707

[9] S. Tavazoie, J. Hughes, M. Campbell, R. Cho, and G. Church, "Systematic determination of genetic network architecture," *Nature Genetics*, vol. 22, pp. 281–285, 1999.

[10] R. Guthke, U. Moller, M. Hoffmann, F. Thies, and S. Topfer, "Dynamic network reconstruction from gene expression data applied to immune response during bacterial infection," *Bioinformatics*, vol. 21, no. 8, pp. 1626–1634, 2005. [Online]. Available: http://bioinformatics.oxfordjournals.org/cgi/content/abstract/21/8/1626

[11] Y. Cheng and G. Church, "Biclustering of expression data," *Proceedings of 8th International Conference on Intelligent Systems for Molecular Biology*, pp. 93–103, 2000.

[12] K. Y. Yeung, D. R. Haynor, and W. L. Ruzzo, "Validating clustering for gene expression data ," *Bioinformatics*, vol. 17, no. 4, pp. 309–318, 2001.

[13] S. Datta and S. Datta, "Comparisons and validation of statistical clustering techniques for microarray gene expression data," *Bioinformatics*, vol. 19, no. 4, pp. 459–466, 2003.

[14] F. Azuaje, "A cluster validity framework for genome expression data ," *Bioinformatics*, vol. 18, no. 2, pp. 319–320, 2002.

[15] A. Prelic, S. Bleuler, P. Zimmermann, A. Wille, P. Buhlmann, W. Gruissem, L. Hennig, L. Thiele, and E. Zitzler, "A systematic comparison and evaluation of biclustering methods for gene expression data," *Bioinformatics*, p. btl060, 2006.

[16] F. M. Al-Akwaa, M. H. Ali, and Y. M. Kadah, "BicAT-Plus: An Automatic Comparative Tool For Bi/Clustering of Gene Expression Data Obtained Using Microarrays," in *26th National Radio Science Conference (NRSC)*, 2009.

[17] S. Barkow, S. Bleuler, A. Prelic, P. Zimmermann, and E. Zitzler, "BicAT: a biclustering analysis toolbox," *Bioinformatics*, vol. 22, no. 10, pp. 1282–1283, 2006.

[18] L. Szeto, A. Liew, H. Yan, and S. Tang, "Gene Expression data clustering and visualization based on a binary hierarchical clustering framework," *Special issue on Biomedical Visualization for Bioinformatics, Journal of Visual Languages and Computing*, vol. 14, pp. 341–362, 2003.

[19] A. Ben-Dor, B. Chor, R. Karp, and Z. Yakhini, "Discovering local structure in gene expression data: the order-preserving submatrix problem," *Journal of Computational Biology*, vol. 10, no. 3-4, pp. 373–384, 2003.

[20] T. Murali and S. Kasif, "Extracting conserved gene expression motifs from gene expression data," *Pac. Symp. Biocomput*, vol. 8, pp. 77–88, 2003.

[21] J. Ihmels, G. Friedlander, S. Bergmann, O. Sarig, Y. Ziv, and N. Barkai, "Revealing modular organization in the yeast transcriptional network," *Nature Genetics*, vol. 31, pp. 370–377, 2002.

[22] J. Ihmels, S. Bergmann, and N. Barkai, "Defining transcription modules using large-scale gene expression data," *Bioinformatics*, vol. 20, pp. 1993–2003, 2004.

[23] R. Bonneau, D. Reiss, P. Shannon, M. Facciotti, L. Hood, N. Baliga, and V. Thorsson, "The Inferelator: an algorithm for learning parsimonious regulatory networks from systems-biology data sets de novo," *Genome Biology*, vol. 7, no. 5, p. R36, 2006. [Online]. Available: http://genomebiology.com/2006/7/5/R36

[24] D. Reiss, N. Baliga, and R. Bonneau, "Integrated biclustering of heterogeneous genome-wide datasets for the inference of global regulatory networks," *BMC Bioinformatics*, vol. 7, no. 1, p. 280, 2006. [Online]. Available: http://www.biomedcentral.com/1471-2105/7/280

[25] E. Capobianco, "Denoising and dimensionality reduction of genomic data," in *Proc. SPIE*, 2005, pp. 69–80.

[26] M. Scott, J. Tzyy-Ping, and G. Dara, "Independent Component Analysis of Simulated ERP Data," Naval Health Research Center, Tech. Rep., 2000.

[27] A. P. Gasch, P. T. Spellman, C. M. Kao, O. Carmel-Harel, M. B. Eisen, G. Storz, D. Botstein, and P. O. Brown, "Genomic Expression Programs in the Response of Yeast Cells to Environmental Changes," *Mol. Biol. Cell*, vol. 11, no. 12, pp. 4241–4257, 2000.

[28] P. T. Spellman, G. Sherlock, M. Q. Zhang, V. R. Iyer, K. Anders, M. B. Eisen, P. O. Brown, D. Botstein, and B. Futcher, "Comprehensive Identification of Cell Cycle-regulated Genes of the Yeast Saccharomyces cerevisiae by Microarray Hybridization," *Mol. Biol. Cell*, vol. 9, no. 12, pp. 3273–3297, 1998. [Online]. Available: http://www.molbiolcell.org/cgi/content/abstract/9/12/3273

[29] F. M. Al-Akwaa and Y. M. Kadah, "Automatic Gene Ontology Software Tool for Bicluster and Cluster Comparisons," in *IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology*. Nashville, TN, USA: IEEE Computational Intelligence Society, 2009.

[30] EBI, "Bioinformatics and drug discovery," www.ebi.ac.uk/2can/disease/genes11.html.

[31] "Bioinformatics Defibation," www.biochem.northwestern.edu/holmgren/Glossary/Definitions/D
B/Bioinformatics.html.

[32] M. Dayhoff, "Computer analysis of protein evolution," *Sci Am*, vol. 222, no. 1, pp. 86–95, 1969.

[33] R. Doolittle, M. Hunkapiller, L. Hood, S. Devare, K. Robbins, S. Aaronson, and H. Antoniades, "Simian sarcoma virus onc gene, v-sis, is derived from the gene (or genes) encoding a platelet-derived growth factor," *Science*, vol. 221, no. 4607, pp. 275–277, 1983. [Online]. Available: http://www.sciencemag.org/cgi/content/abstract/221/4607/275

[34] M. Waterfield, G. Scrace, N. Whittle, P. Stroobant, A. Johnsson, A. Wasteson, B. Westermark, C. Heldin, J. Huang, and D. TF., "Platelet-derived growth factor is structurally related to the putative transforming protein p28sis of simian sarcoma virus," *Nature*, vol. 304, no. 5921, pp. 35–9, 1983.

[35] www.ibpassociation.org/IBPA_articles/aug1_2005/BIOINFORMATICS%
20Article.doc".

[36] S. OVERBY, "Drug Companies on Speed," *CIO Magazine*, 2001.

[37] P. Larranaga, B. Calvo, R. Santana, C. Bielza, J. Galdiano, I. Inza, J. A. Lozano, R. Armananzas, G. Santafe, A. Perez, and V. Robles, "Machine learning in bioinformatics," *Brief Bioinform*, vol. 7, no. 1, pp. 86–112, 2006. [Online]. Available: http://bib.oxfordjournals.org/cgi/content/abstract/7/1/86

[38] A. Brazma, H. Parkinson, T. Schlitt, and S. Mohammadreza, "A quick introduction to elements of biology - cells, molecules, genes, functional genomics, microarrays," http://www.ebi.ac.uk/microarray/biology_intro.html.

[39] B. Alistair J.P and T. Mick F, *Methods in Microbiology: Yeast Gene Analysis*. Academic Press, 1996, vol. 26.

[40] T. BROWN, *Genome2*. BIOS Scientific Publishers Ltd, 2002.

[41] A. Tanay, R. Sharan, M. Kupiec, and R. Shamir, "Revealing modularity and organization in the yeast molecular network by integrated analysis of highly heterogeneous genomewide data," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 101, no. 9, pp. 2981–2986, 2004. [Online]. Available: http://www.pnas.org/content/101/9/2981.abstract

[42] A. Lesk, *Introduction to Bioinformatics*. Oxford Press, 2008, ch. Archieves on information retrieval, 206 - 209.

[43] R. Jeremy, *Bioinformatics*, 2nd ed., ser. Computational Biology Series. Springer Verlag GmbH, April 2009, vol. 10.

[44] R. Kothapalli, S. Yoder, S. Mane, and T. Loughran, "Microarray results: how accurate are they?" *BMC Bioinformatics*, vol. 3, no. 1, p. 22, 2002. [Online]. Available: http://www.biomedcentral.com/1471-2105/3/22

[45] D. Sorin, K. Purvesh, E. Aron C, and S. Zoltan, "Reliability and reproducibility issues in DNA microarray measurements," *Trends Genet*, vol. 22, no. 2, pp. 101–109, 2006.

[46] "Genomebrowser in Wikipedia," http://en.wikipedia.org/wiki/Genomebrowser.

[47] "SGD Gene Nomenclature Conventions," http://www.yeastgenome.org/help/yeastGeneNomencl

[48] "SGD Help: Systematic Names - Protein Coding ORFs," http://www.yeastgenome.org/help/SystematicNamesHelp.html.

[49] "SGD Advanced Search," http://www.yeastgenome.org/cgi-bin/search/featureSearch.

[50] "Gasch Dataset," http://genome-www.stanford.edu/yeast_stress.

[51] I. Avila-Campillo, K. Drew, J. Lin, D. J. Reiss, and R. Bonneau, "BioNetBuilder: automatic integration of biological networks," *Bioinformatics*, vol. 23, no. 3, pp. 392–393, 2007. [Online]. Available: http://bioinformatics.oxfordjournals.org/cgi/content/abstract/23/3/392

[52] S. K. Isaac, K. Alvin, and J. B. Atul, *Microarrays for an Integrative Genomics*. The MIT Press, 2003.

[53] O. Troyanskaya, M. Cantor, G. Sherlock, P. Brown, T. Hastie, R. Tibshirani, D. Botstein, and R. B. Altman, "Missing value estimation methods for DNA microarrays," *Bioinformatics*, vol. 17, no. 6, pp. 520–525, 2001. [Online]. Available: http://bioinformatics.oxfordjournals.org/cgi/content/abstract/17/6/520

[54] J. H. Laurie, K. Semyon, and Y. Shibu, "Exploring Expression Data: Identification and Analysis of Coexpressed Genes," *Genoimc Research*, vol. 9, pp. 1106–1116, 1999.

[55] F. Geier, J. Timmer, and C. Fleck, "Reconstructing gene-regulatory networks from time series, knock-out data, and prior knowledge," *BMC Systems Biology*, vol. 1, no. 1, p. 11, 2007. [Online]. Available: http://www.biomedcentral.com/1752-0509/1/11

[56] B. Di Camillo, F. Sanchez-Cabo, G. Toffolo, S. Nair, Z. Trajanoski, and C. Cobelli, "A quantization method based on threshold optimization for microarray short time series," *BMC Bioinformatics*, vol. 6, no. Suppl 4, p. S11, 2005.

[57] Y. M. Kadah, "Adaptive denoising of event-related functional magnetic resonance imaging data using spectral subtraction," *IEEE Transactions on Biomedical Engineering*, vol. 51, no. 11, pp. 1944–1953, 2004.

[58] S. V. Vaseghi, *Advanced Signal Processing and Digital Noise Reduction*. New York: Wiley, 1996.

[59] G. Stolovitzky, R. Prill, and A. Califano, "Lessons from the DREAM2 Challenges," *Annals of the New York Academy of Sciences*, pp. 1158:159–95, 2009.

[60] G. Fung, "A Comprehensive Overview of Basic Clustering Algorithms," The University of WISCONSIN MADISON, Tech. Rep., 2001.

[61] R. Sharan, R. Elkon, and R. Shamir, "Cluster analysis and its applications to gene expression data," *Ernst Schering Res Found Workshop*, pp. 83–108, 2002.

[62] M. B. Eisen, P. T. Spellman, P. O. Brown, and D. Botstein, "Cluster analysis and display of genome-wide expression patterns," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 95, no. 25, pp. 14 863–14 868, 1998. [Online]. Available: http://www.pnas.org/content/95/25/14863.abstract

[63] P. Tamayo, D. Slonim, J. Mesirov, Q. Zhu, S. Kitareewan, E. Dmitrovsky, E. S. Lander, and T. R. Golub, "Interpreting patterns of gene expression with self-organizing maps: Methods and application to hematopoietic differentiation," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 96, no. 6, pp. 2907–2912, 1999. [Online]. Available: http://www.pnas.org/content/96/6/2907.abstract

[64] R. Shamir and R. Sharan, "Click: a clustering algorithm for gene Expression analysis," *Proceedings of the 8th International Conference on Intelligent Systems for Molecular Biology*, pp. 307–316, 2000.

[65] ——, *Current topics in computational biology*. MIT Press., 2001, ch. Algorithmic approaches to clustering gene expression data.

[66] M. Eisen, P. Spellman, P. Brown, and D. Botstein, "Cluster analysis and display of genome-wide expression patterns," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 95, pp. 14 863–14 868, 1998.

[67] S. Johnson, "Hierarchical clustering schemes," *Psychometrika*, vol. 32, no. 3, pp. 241–254, 1967.

[68] S. Madeira and A. Oliveira, "Biclustering algorithms for biological data analysis: a survey," *IEEE/ACM Trans Comput Biol Bioinform*, vol. 1, no. 1, pp. 24–45, 2004.

[69] K. Cheng, N. Law, W. Siu, and T. Lau, "BiVisu: software tool for bicluster detection and visualization," *Bioinformatics*, vol. 23, no. 17, pp. 2342–2344, 2007.

[70] X. Liu and L. Wang, "Computing the maximum similarity bi-clusters of gene expression data," *Bioinformatics*, vol. 23, pp. 50–56, 2007.

[71] A. Tchagang and A. Twefik, "Robust biclustering algorithm (ROBA) for DNA microarray data analysis," in *IEEE/SP 13th Workshop on Statistical Signal Processing*, 2005, pp. 984–989.

[72] A. Tanay, R. Sharan, M. Kupiec, and R. Shamir, "Revealing modularity and organization in the yeast molecular network by integrated analysis of highly heterogeneous genomewide data," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 101, no. 9, pp. 2981–2986, 2004. [Online]. Available: http://www.pnas.org/content/101/9/2981.abstract

[73] H. Wang, W. Wang, J. Yang, and P. S. Yu, "Clustering by Pattern Similarity: the pCluster Algorithm," in *SIGMOD*, 2002.

[74] J. Hartigan, "Direct Clustering of a data matrix," *Journal of the American Statistical Association*, vol. 67, pp. 123–129, 1972.

[75] I. Gat-Viks, R. Sharan, and R. Shamir, "Scoring clustering solutions by their biological relevance," *Bioinformatics*, vol. 19, no. 18, pp. 2381–2389, 2003. [Online]. Available: http://bioinformatics.oxfordjournals.org/cgi/content/abstract/19/18/2381

[76] J. Handl, J. Knowles, and D. B. Kell, "Computational cluster validation in post-genomic data analysis," *Bioinformatics*, vol. 21, no. 15, pp. 3201–3212, 2005. [Online]. Available: http://bioinformatics.oxfordjournals.org/cgi/content/abstract/21/15/3201

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

[78] S. Maere, K. Heymans, and M. Kuiper, "BiNGO: a Cytoscape plugin to assess overrepresentation of Gene Ontology categories in Biological Networks," *Bioinformatics*, vol. 21, pp. 3448–3449, 2005.

[79] P. Khatri and S. Draghici, "Ontological analysis of gene expression data: current tools, limitations, and open problems," *Bioinformatics*, vol. 21, no. 18, pp. 3587–3595, 2005. [Online]. Available: http://bioinformatics.oxfordjournals.org/cgi/content/abstract/21/18/3587

[80] S. Falcon and R. Gentleman, "Using GOstats to test gene lists for GO term association," *Bioinformatics*, vol. 23, no. 2, pp. 257–258, 2007. [Online]. Available: http://bioinformatics.oxfordjournals.org/cgi/content/abstract/23/2/257

[81] C. Castillo-Davis and D. Hartl, "GeneMerge - post-genomic analysis, data mining, and hypothesis testing," *Bioinformatics*, vol. 19, no. 7, pp. 891–892, 2003.

[82] G. F. Berriz, O. D. King, B. Bryant, C. Sander, and F. P. Roth, "Characterizing gene sets with FuncAssociate," *Bioinformatics*, vol. 19, pp. 250–25 024, 2003.

[83] K.-O. Cheng, N.-F. Law, W.-C. Siu, and A. Liew, "Identification of coherent patterns in gene expression data using an efficient biclustering algorithm and parallel coordinate visualization," *BMC Bioinformatics*, vol. 9, no. 1, p. 210, 2008.

[84] "Bicat Web Site," http://www.tik.ee.ethz.ch/sop/bicat/?page=developersGuide.php.

[85] S. Thomas and B. Alvis, "Current approaches to gene regulatory network modelling," *BMC Bioinformatics*, vol. 8, no. Suppl 6, p. S9, 2007. [Online]. Available: http://www.biomedcentral.com/1471-2105/8/S6/S9

[86] J. Stark, D. Brewer, M. Barenco, D. Tomescu, R. Callard, and M. Hubank, "Reconstructing gene networks: what are the limits?" *Biochem Soc Trans*, vol. 31, pp. 1519 – 1525, 2003.

[87] H. Michael, L. Sandro, T. Susanne, v. S. Eugene, and G. Reinhard, "Gene regulatory network inference: Data integration in dynamic modelsA review," *Biosystems*, vol. 96, no. 1, pp. 86–103, 2009.

[88] T. Akutsu, S. Miyano, and S. Kuhara, "Inferring qualitative relations in genetic networks and metabolic pathways," *Bioinformatics*, vol. 16, pp. 727–734, 2000.

[89] Wessels, E. v. Someren, and M. Reinders, "A comparison of genetic network models," in *Pac Symp Biocomput*, 2001, pp. 508 – 519.

[90] A. V. Werhli, M. Grzegorczyk, and D. Husmeier, "Comparative evaluation of reverse engineering gene regulatory networks with relevance networks, graphical Gaussian models and Bayesian networks," *Bioinformatics*, vol. 22, pp. 2523 – 2531, 2006.

[91] N. Friedman, "Inferring Cellular Networks Using Probabilistic Graphical Models," *Science*, vol. 303, pp. 799–805, 2004.

[92] A. Wagner, "How to reconstruct a large genetic network from n gene perturbations in fewer than n2 easy steps," *Bioinformatics*, vol. 17, pp. 1183 – 1197, 2002.

[93] F. Markowetz and R. Spang, "Inferring cellular networks - a review," *BMC Bioinformatics*, vol. 8, p. 55, 2007.

[94] K. P. Murphy, "Bayesnet Toolbox," http://www.cs.ubc.ca/ murphyk/Software/BNT, 1999.

[95] I. Shmulevich, "Probabilistic Boolean Network," http://personal.systemsbiology.net/ilya/PBN/PBN.htm., 2003.

[96] B. B, D. E., J. H. d., M. P., and M. Page, "Genetic Network Analyzer," http://www-helix.inrialpes.fr/article122.html., 2000.

[97] A. Margolin, I. Nemenman, K. Basso, C. Wiggins, G. Stolovitzky, R. Favera, and A. Califano, "ARACNE:Algorithm for the Reconstruction of Accurate Cellular Networks," http://amdec-bioinfo.cu-genome.org/html/ARACNE.htm, 2006.

[98] I. vanov and E. R. Dougherty, "Modeling Genetic Regulatory Networks: Continuous or Discrete?" in *Biological Systems*, 2006, pp. 219–229.

[99] F. Holstege, E. Jennings, J. Wyrick, T. Lee, C. Hengartner, M. Green, T. Golub, E. Lander, and R. Young, "Dissecting the regulatory circuitry of a eukaryotic genome," *Cell*, vol. 95, no. 5, pp. 717–728, 25 November 1998.

[100] U. Paul, V. Kaufman, and B. Drossel, "Properties of attractors of canalyzing random Boolean networks," *Physical Review E*, vol. 73, no. 2, 2006.

[101] I. Shmulevich, E. R. Dougherty, S. Kim, and W. Zhang, "Probabilistic Boolean networks: a rule-based uncertainty model for gene regulatory networks," *Bioinformatics*, vol. 18, pp. 261–274, 2002.

[102] P. Smolen, D. A. Baxter, and J. H. Byrne, "Modeling transcriptional control in gene networks methods, recent results, and future directions," *Bull. Math. Biol*, vol. 62, no. 2, pp. 247–292, 2000.

[103] G. Yagil and E. Yagil, "On the relation between effector concentration and the rate of induced enzyme synthesis," *Biophys. J*, vol. 11, pp. 11–27, 1971.

[104] d. J. Hidde and G. Johannes, *Genomic Signal Processing and Statistics*. Hindawi Publishing Corporation, 2005, ch. Modeling and simulation of genetic regulatory networks.

[105] C. J. Wolfe, I. S. Kohane, and A. J. Butte, "Systematic survey reveals general applicability of guilt-by-association within gene coexpression networks," *BMC Bioinformatics*, vol. 6, p. 227, 2005.

[106] R. Thomas, "Boolean formalization of genetic control circuits," *J Theor Biol*, vol. 42, no. 3, pp. 563–85, 1973.

[107] P. D'haeseleer, "Data Requirements for Inferring Genetic Networks from Expression Data," in *4th Pacific Symposium on Biocomputing*, Big Island of Hawaii, Hawaii, USA, 1999.

[108] X.-w. Chen, G. Anantha, and X. Wang, "An effective structure learning method for constructing gene networks," *Bioinformatics*, vol. 22, no. 11, pp. 1367–1374, 2006. [Online]. Available: http://bioinformatics.oxfordjournals.org/cgi/content/abstract/22/11/1367

[109] G. F. Cooper and E. Herskovits, "A Bayesian method for the induction of probabilistic networks from data," *Mach. Learn*, vol. 9, pp. 309–347, 1992.

[110] S. Ott, S. Imoto, and S. Miyano, "Finding Optimal Models for Small Gene Networks," in *Pacific Symposium on Biocomputing*, 2004, pp. 557–567.

[111] D. Marbach, C. Mattiussi, and D. Floreano, "Combining Multiple Results of a Reverse Engineering Algorithm: Application to the DREAM Five Gene Network Challenge," *Annals of the New York Academy of Sciences*, vol. 1158, pp. 102–113, 2009.

[112] K. P. Murphy, "Active learning of causal Bayes net structure," 2001, bayesian Network.

[113] J. Cheng, D. Bell, and W. Liu, "Learning Bayesian networks from data: an information-theory based approach," *Artif. Intell. Res*, vol. 137, pp. 43–90, 2002.

[114] D. Chickering, "Optimal structure identification with greedy search," *J. Mach. Learn. Res*, vol. 3, pp. 507–554, 2002.

[115] P. Mendes, "GEPASI: a software package for modelling the dynamics, steady states and control of biochemical and other systems," *Comput. Appl. Biosci*, vol. 9, no. 5, pp. 563–571, 1993.

[116] D. Marbach, T. Schaffter, C. Mattiussi, and D. Floreano, "Generating Realistic In Silico Gene Networks for Performance Assessment of Reverse Engineering Methods," *Journal of Computational Biology*, vol. 16, no. 2, pp. 229–239, 2009. [Online]. Available: http://dx.doi.org/10.1089/cmb.2008.09TT

[117] K. Sachs, O. Perez, D. Pe'er, D. A. Lauffenburger, and N. G.P, "Causal Protein-Signaling Networks Derived from Multiparameter Single-Cell Data," *Science*, vol. 308, pp. 523–529, 2005.

[118] C. Chow and C. Liu, "Approximating discrete probability distributions with dependence trees," *IEEE Transactions on Information Theory*, vol. 14, pp. 462–467, 1968.

[119] P. LeRay, "BNT Structure Learning Package," http://banquiseasi.insa-rouen.fr/projects/bnt-slp, 2003.

[120] C. Chow and C. Liu, "Approximating discrete probability distributions with dependence trees," *IEEE Transactions on Information Theory*, vol. 14, pp. 462–467, 1968.

[121] G. Stolovitzky, "DRAEM 3rd Conference," http://wiki.c2b2.columbia.edu/dream/index.php/The_3 2008.

[122] F. M. Al-Akwaa, N. H. Solouma, and Y. M. Kadah, "Ssbbn: Gene regulatory network construction using spectral subtraction denoising, biclustering and bayesian network," in *The 6th Annual RECOMB Satellite on Regulatory Genomics, the 5th Annual RECOMB Satellite on Systems Biology, and the 4th Annual DREAM reverse engineering challenges*, MIT, 2009.

[123] J. J. Faith, B. Hayete, J. T. Thaden, I. Mogno, J. Wierzbowski, G. Cottarel, S. Kasif, J. J. Collins, and T. S. Gardner, "Large-Scale Mapping and Validation of Escherichia coli Transcriptional Regulation from a Compendium of Expression Profiles," *PLoS Biol*, vol. 5, no. 1, p. e8, 01 2007.

[124] M. Hecker, "Gene Regulatory Network Reconstruction Best Practice Guide," Bio-Control Jena GmbH, Tech. Rep., 2007.

[125] P. Bork, L. Jensen, C. vonMering, A. Ramani, I. Lee, and E. Marcotte, "Protein interaction networks from yeast to human," *Curr Opin Struct Biol*, vol. 14, no. 3, pp. 292–299, 2004.

[126] G. D. Bader, M. P. Cary, and C. Sander, "Pathguide: a Pathway Resource List," *Nucl. Acids Res.*, vol. 34, no. suppl_1, pp. 504–506, 2006. [Online]. Available: http://nar.oxfordjournals.org/cgi/content/abstract/34/suppl_1/D504

[127] P. Dana, "Bayesian Network Analysis of Signaling Networks: A Primer," *Sci. STKE*, vol. 2005, no. 281, pp. pl4–, 2005. [Online]. Available: http://stke.sciencemag.org/cgi/content/abstract/sigtrans;2005/281/pl4

[128] D. Jesse and W. Madison, "The relationship between Precision-Recall and ROC curves," in *Proceedings of the 23rd international conference on Machine learning*. ACM New York, NY, USA, 2006, pp. 233 – 240.

[129] A. C. Lozano, N. Abe, Y. Liu, and S. Rosset, "Grouped graphical Granger modeling for gene expression regulatory networks discovery," *Bioinformatics*, vol. 25, no. 12, pp. i110–118, 2009. [Online]. Available: http://bioinformatics.oxfordjournals.org/cgi/content/abstract/25/12/i110

[130] P. Shannon, A. Markiel, O. Ozier, N. Baliga, J. Wang, D. Ramage, N. Amin, B. Schwikowski, and T. Ideker, "Cytoscape: a software environment for integrated models of biomolecular interaction networks," *Genome Res*, vol. 13, no. 11, pp. 2498–504, 2003.

[131] Y. Assenov, F. Ramirez, S.-E. Schelhorn, T. Lengauer, and M. Albrecht, "Computing topological parameters of biological networks," *Bioinformatics*, vol. 24, no. 2, pp. 282–284, 2008. [Online]. Available: http://bioinformatics.oxfordjournals.org/cgi/content/abstract/24/2/282

[132] A. Tong, B. Drees, G. Nardelli, G. Bader, B. Brannetti, and L. Castagnoli, "A combined experimental and computational strategy to define protein interaction networks for peptide recognition modules." *Science*, vol. 295, pp. 321–324, 2002.

[133] G. Bader and C. Hogue, "An automated method for finding molecular complexes in large protein interaction networks," *BMC Bioinformatics*, vol. 4, no. 1, p. 2, 2003. [Online]. Available: http://www.biomedcentral.com/1471-2105/4/2