Gene expression data analysis using novel methods: Predicting time delayed correlations and evolutionarily conserved functional modules

Dissertation

zur
Erlangung des Doktorgrades
der Naturwissenschaften
(Dr. rer. nat.)

dem Fachbereich Biologie
der Philipps-Universität Marburg
vorgelegt von

Rajarajeswari Balasubramaniyan
aus Madurai, Tamilnadu, Indien

Marburg/Lahn 2005

Vom Fachbereich Biologie
der Philipps-Universität Marburg als Dissertation
angenommen am: 18-07-2005

Erstgutachter: Herr PD Dr. Jörg Kämper
Zweitgutachter: Herr Prof. Dr. Eyke Hüllermeier

Tag der mündlichen Prüfung am: 22-07-2005

The research pertaining this thesis was carried out at the Department of Organisimic Interactions of the Max-Planck-Institute for Terrestrial Microbiology, Marburg, from July 2002 to July 2005 under the supervision of PD Dr. Jörg Kämper.

Part of this thesis is published in:

Balasubramaniyan, R., Hüllermeier, E., Weskamp, N., Kämper, J. (2005). Clustering of Gene Expression Data Using a Local Shape-Based Similarity Measure, Bioinformatics *21*, 1069-1077.

Declaration

I hereby declare that the dissertation entitled “Gene expression data analysis using novel methods: Predicting time delayed correlations and evolutionarily conserved functional modules” submitted to the Department of Biology, Philipps-Universität, Marburg is the original and independent work carried out by me under the guidance of the PhD committee, and the dissertation is not formed previously on the basis of any award of Degree, Diploma or other similar titles.

(Date and Place)

(Rajarajeswari Balasubramaniyan)

On action alone be thy interest,
Never on its fruits.
Let not the fruits of action be thy motive,
Nor be thy attachment to inaction.

Bhagavad Gita

Synopsis

Microarray technology enables the study of gene expression on a large scale. One of the main challenges has been to devise methods to cluster genes that share similar expression profiles. In gene expression time courses, a particular gene may encode transcription factor and thus controlling several genes downstream; in this case, the gene expression profiles may be staggered, indicating a time-delayed response in transcription of the later genes. The standard clustering algorithms consider gene expression profiles in a global way, thus often ignoring such local time-delayed correlations. We have developed novel methods to capture time-delayed correlations between expression profiles: (1) A method using dynamic programming and (2) CLARITY, an algorithm that uses a local shape based similarity measure to predict time-delayed correlations and local correlations. We used CLARITY on a dataset describing the change in gene expression during the mitotic cell cycle in *Saccharomyces cerevisiae*. The obtained clusters were significantly enriched with genes that share similar functions, reflecting the fact that genes with a similar function are often co-regulated and thus co-expressed. Time-shifted as well as local correlations could also be predicted using CLARITY.

In datasets, where the expression profiles of independent experiments are compared, the standard clustering algorithms often cluster according to all conditions, considering all genes. This increases the background noise and can lead to the missing of genes that change the expression only under particular conditions. We have employed a genetic algorithm based module predictor that is capable to identify group of genes that change their expression only in a subset of conditions. With the aim of supplementing the *Ustilago maydis* genome annotation, we have used the module prediction algorithm on various independent datasets from *Ustilago maydis*. The predicted modules were cross-referenced in various *Saccharomyces cerevisiae* datasets to check its evolutionarily conservation between these two organisms. The key contributions of this thesis are novel methods that explore biological information from DNA microarray data.

Zusammenfassung

Die Mikroarray-Technologie ermöglicht es, die Expression von Genen im großen Maßstab zu analysieren. Einer der größten Anreize bei der Daten-Analyse besteht darin, Methoden zu entwickeln, um Gene mit einem ähnlichen Expressionsprofil in gemeinsamen Clustern zu gruppieren.

Bei Experimenten, in denen die Veränderung der Gen-Expression zeitabhängig verfolgt wird, ist es möglich, dass ein bestimmtes Gen für einen Transkriptionsfaktor die Expression weiterer Gene kontrolliert. Dadurch bedingt können die Profile einzelner Gene zueinander verschoben sein. Die Standard-Cluster-Algorithmen betrachten Gen-Expressionsprofile oftmals global, womit solche zeitversetzten Zusammenhänge in vielen Fällen ignoriert werden.

Wir haben neuartige Methoden entwickelt, um zeitversetzte Zusammenhänge zwischen Expressionsprofilen zu detektieren: (1) Eine Methode, die dynamische Programmierung verwendet und (2) CLARITY; ein Algorithmus, der über den Vergleich lokaler Ähnlichkeiten im der Kurvenform sowohl zeitversetzte als auch lokale Ähnlichkeiten entdecken kann. Wir haben CLARITY verwendet, um einen Datensatz, der die Veränderungen der Gen-Expression währen des Zellzyklus von *Saccharomyces cerevisiae* beschreibt, zu analysieren. Die erhaltenen Cluster zeigen eine signifikante Anreicherung mit Genen bestimmter Funktionen, was deutlich macht, dass Gene mit einer ähnlichen Funktion oft auch co-reguliert und damit co-exprimiert sind. Durch CLARITY wurden sowohl zeitversetzte als auch lokale Korrelationen entdeckt.

In Datensätzen, die verschiedene voneinander unabhängige Experimente miteinander kombinieren, versuchen Standard-Algorithmen oftmals, Cluster zu bilden, indem sie alle Bedingungen und alle Gene berücksichtigen. Diese Vorgehensweise erhöht den Hintergrund (Rauschen), was dazu führen kann, dass bestimmte Gene, die ihre Expression nur unter bestimmten, aber nicht allen Bedingungen ändern, nicht erfasst werden. Wir haben ein Programm zur Modul-Vorhersage entwickelt, das auf der Anwendung genetischer Algorithmen beruht, und das Gruppen von Genen identifizieren kann, die nur in einer Untergruppe der Bedingungen ihre Expression verändern. Mit dem Ziel, die funktionelle Annotierung des *Ustilago maydis* Genoms zu unterstützen, haben wir das Modul-Vorhersage Programm für die Analyse verschiedener unabhängiger Expressions- Datensätze von *U. maydis* verwendet. Die vorhergesagten Module wurden auf verschiedene Expressions-Datensätze von *S. cerevisiae* übertragen, um die evolutionäre Konservierung zwischen den beiden Organismen zu untersuchen.

Der Hauptbeitrag dieser Arbeit liegt in der Entwicklung neuartiger Methoden, die es ermöglichen, biologische Informationen in Mikroarray-Datensätzen zu untersuchen.

Summary of Terms

BLAST	Basic Local Alignment Search Tool
cDNA	Complementary DNA; complementary single stranded DNA copy of a messenger RNA, produced by reverse transcription
cRNA	Synthetic RNA produced by transcription from a specific DNA single stranded template
CLARITY	Clustering with Local shApe based similaRITY
CYGD	Comprehensive Yeast Genome Database
DNA	Deoxy riboNucleicAcid; carrier of the genetic information in organisms
EGAD	Expressed Gene Anatomy Database
EST	Expressed Sequence Tags; a small part of the active part of a gene made from cDNA which can be used to fish the rest of the gene out of the chromosome by matching base pairs with part of the gene
GA	Genetic Algorithm
GenProtEC	Genome and Proteome Database of *E. coli*
GEMS	Gene Expression Module Sampler
GO	Gene Ontology; a controlled vocabulary of terms relating to molecular function, biological process, or cellular components developed by the Gene Ontology Consortium
KEGG	Kyoto Encyclopedia of Genes and Genomes
MIPS	Munich Information Center for Protein Sequences
Min (X, Y)	Minimum between X and Y
mRNA	Messenger RNA; a complementary copy of a stretch of DNA encoding a gene
OPSM	Order Preserving Sub-Matrix
OP-cluster	Order Preserving Cluster
ORF	Open Reading Frame

P value	Probability value; The probability value (p-value) of a statistical hypothesis test is the probability of getting a value of the test statistic as extreme as or more extreme than that observed by chance alone, if the null hypothesis H_0, is true
PCR	Polymerase Chain Reaction; a method for amplifying a specific DNA sequence using DNA polymerase
PIR	Protein Information Resource
RNA	RiboNucleic Acid
rRNA	Ribosomal RNA
RT-PCR	Reverse Transcriptase Polymerase Chain Reaction
SAMBA	Statistical Algorithmic Method for Bicluster Analysis
SIM (X, Y)	Similarity between expression profiles X and Y
SOM	Self Organizing map
SRC	Spearman Rank Correlation
TM	Transcription Module
tRNA	transfer RNA

Contents

Chapter 1

Chapter 2

Chapter 3

Chapter 4

Chapter 1

Introduction

Recent developments in the biological sciences refreshed exploration in biological research. The genome sequencing projects of many organisms including human represents one of the largest scientific endeavors in the history of mankind. In 1995, *Haemophilus influenzae,* a gram negative human parasitic bacterium was the first free-living organism to have its entire chromosome sequenced. In eukaryotes, *Sacchromyces cerevisiae*, the brewers yeast was the first one to have its genome fully sequenced in 1996 (Goffeau et al., 1996). Recently in October 2004, the complete sequence of the human genome confronted us with the fact that the human genome contains approximately 20,000 to 25,000 genes, which was about 10,000 less than that indicated in the draft (International human genome sequencing consortium, 2004). The completion of the genomic sequences of 'model' organisms such as *Sacchromyces cerevisiae* and *Caenorbabditis elegans* provides us an idea about the genome, the complete blue print of the organism. Once whole genome sequencing information is available for an organism, the task turns to understanding the 'biological function' of genes. Although many genome sequencing projects have been completed, the biological function is not known for roughly half of the genes in every genome that has been sequenced to date (Stuart et al., 2003). Elucidating 'function' for these large fractions of genes referred to as 'functional genomics' poses the next major challenge in the post genomic era.

In terms of understanding the function of genes, knowing when, where and to what extent a gene is expressed is central to understanding the activity and biological roles of its encoded protein. The collection of genes that are expressed or transcribed from genomic DNA, referred to as an expression profile or the 'transcriptome', is a major determinant of cellular phenotype and function. The transcription of genomic DNA to produce mRNA is the first step in the process of protein synthesis, and differences in gene expression are responsible for both morphological and phenotypic differences as well as indicative of cellular response to environment stimuli and perturbations.

Traditional molecular biology followed the reductionist approach mostly concentrating on ‘one gene at a time’. In order to take full advantage of the large and rapidly increasing body of sequence information, new technologies are required. Among the most powerful and versatile tools for functional genomics are high-density arrays of oligonucleotides or complementary DNAs. One of the most important applications for arrays so far is simultaneous measurement of gene expression (mRNA abundance) of thousands of genes in a genome during important biological processes.

Elucidating the patterns hidden in gene expression data offers a tremendous opportunity for an enhanced understanding of functional genomics. The data generated from microarray experiments often consists of millions of gene expression measurements that raise the complexity of comprehending and interpreting the massive information. Although many algorithms (reviewed in Jiang et al., 2004) have been developed to analyze the massive data generated from microarrays, interpreting this vast amount of data often presents a challenge to those interested in studying the relationships among genes in a genome.

A major drawback of available data analysis algorithms is that they ignore additional information that could give improved understanding of transcriptional control (i.e., the controls that act on gene expression or transcription) within a genome.

Further, these standard algorithms often consider all conditions (or samples) that measure the gene expression of a large number of genes, perhaps all genes of an organism during different biological processes. Since the data come from diverse experiments, considering all conditions could simply add noise to the data.

This thesis presents new methods to overcome the problems stated above with the aim to increase the biological information obtained from gene expression data.

In this chapter, section 1.1 presents a brief description of gene expression and microarrays, section 1.2 and section 1.3 describes cDNA arrays and Affymetrix Genechip technology, respectively, and section 1.4 explains the topic of Microarray data analysis and provides the motivation for this dissertation.

1.1 Gene expression and microarrays

The foundation of genetic and molecular biological lies in the central dogma enunciated by Francis Crick in 1958 which states that genetic information from double stranded DNA is transformed to RNA through transcription, which in turn is used as a matrix to synthesize specific proteins via the translation machinery.

The process of making a particular type of a protein from a continuous stretch of genomic DNA, referred to as a gene, is called protein synthesis and has three essential stages: (1) transcription, (2) splicing and (3) translation.

(1) In the transcription phase one strand of DNA molecule is copied into a complementary pre-mRNA* by the protein complex RNA polymerase II.

(2) In eukaryotes, the pre-mRNA is trimmed by a process called splicing. Splicing removes stretches of the pre mRNA, called introns, while the remaining sections called exons are then joined together. 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.

(3) Translation is the process of making proteins by joining together amino acids in the order encoded by the mRNA. The order of the amino acids is determined by 3 adjacent nucleotides (triplets) in the DNA. This is known as the 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 both CAT and CAC encode the amino acid histidine. In the cytoplasm the mRNA forms a complex with ribosomes, which are large complexes of proteins and RNA molecules. The precise interactions and functions of all proteins in ribosomes are not yet fully understood.

Proteins, the final product of translation machinery, can be post-translationaly modified e.g., by addition of sugars or cleavage that might affect their location and function.

Due to alternative splicing and post-translational modifications, the paradigm of ‘one gene – one protein’ has changed; one gene can produce a variety of proteins.

*pre stands for preliminary and m for messenger

For a particular organism, the DNA content of most of the cells is the same. However, the amount of mRNA and the proteins vary between cells and also varies within a cell under different conditions. By systematically observing the changes in mRNA expression levels of all the genes in a genome under different experimental conditions, one can add to the knowledge of how these genes affect the function of the cells.

Traditionally, gene expression studies were done one gene at a time using Northern blots that compared the mRNA abundance between mRNA samples cross linked on a single membrane and hybridized with a labeled probe. *In vitro* transcribed RNA and oligonucleotides are normally used as hybridization probes in Northern blots.

Another sensitive technique called RT-PCR (Reverse Transcriptase Polymerase Chain Reaction) uses PCR amplified reverse-transcribed mRNA to measure very low mRNA levels from samples.

However, with the advent of genomics, more sophisticated methods are needed to understand the genetic and functional relationship among the genes. Concurrent measurement of mRNA expression levels of thousands of genes can help to make sense of the cell's response to a specific condition. The advent of microarrays helps to study the entire genome of an organism on a chip with the size of a microscope slide.

Microarrays exploit the preferential binding of complementary single-stranded nucleic acid sequences. “Probe” DNA strands are spotted onto the chip. The “Target” DNA mixture is then hybridized onto the chip to allow base pairing under conditions such that only highly complementary sequences will remain bound to their specific partners.

Several technologies have been developed for the simultaneous measurement of gene expression, most notably spotted microarrays and oligonucleotide arrays from Affymetrix* (Affymetrix Genechip arrays). These technologies mainly differ in two ways, (1) how probes are deposited on the chip and (2) the length of DNA sequences that are deposited. The next sections give a brief overview of these two technologies.

*Affymetrix, Inc, Santa Clara, CA, USA

1.2 cDNA Microarray technology

1.2.1 Principle of spotted DNA microarrays

The spotted DNA microarrays are a conceptually simple and cost effective method for monitoring the relative levels of expression of thousands of genes simultaneously (Schena et al., 1995). In an array experiment, probes (PCR-amplified cDNA or genomic fragments or specific oligonucleotides) are individually printed on glass microscope slides using a robotic arrayer. To compare the relative abundance of each of these molecules in DNA or RNA samples of two different populations, the two samples are first labeled using different fluorescent dyes, for example Cy3 and Cy5. The two samples are then mixed and hybridized with the arrayed DNA spots. Laser excitation of the incorporated targets yields an emission with a characteristic spectrum, which is measured using a scanning device. Monochrome images from the scanner are imported into software in which the images are pseudo-colored and merged. Data from a single hybridization experiment are viewed as a normalized ratio in which significant deviation from 1 (no change) are indicative of increased (>1) or decreased (<1) levels of gene expression relative to the reference sample. These measurements are used to determine the ratio, and in turn the relative abundance, of specific molecules in the two mRNA or DNA samples (Fig. 1.1).

1.2.2 Probe selection

Production of arrays begins with the choice of DNA fragments (probes) to be printed on the microarray. Probes are often chosen from databases like GenBank, dbEST and UniGene. Before the availability of complete or near-complete eukaryotic genome sequences, genes expressed in cells, tissues or organs were identified through sequence analysis of cDNA data banks. cDNA clones from the cDNA data banks of *Arabidopsis thaliana* and human peripheral blood lymphocytes were used in the construction of the first cDNA microarrays (Schena et al., 1995; Schena et al., 1996). Expressed sequence tags (ESTs) of an organism are also used as sources for arraying.

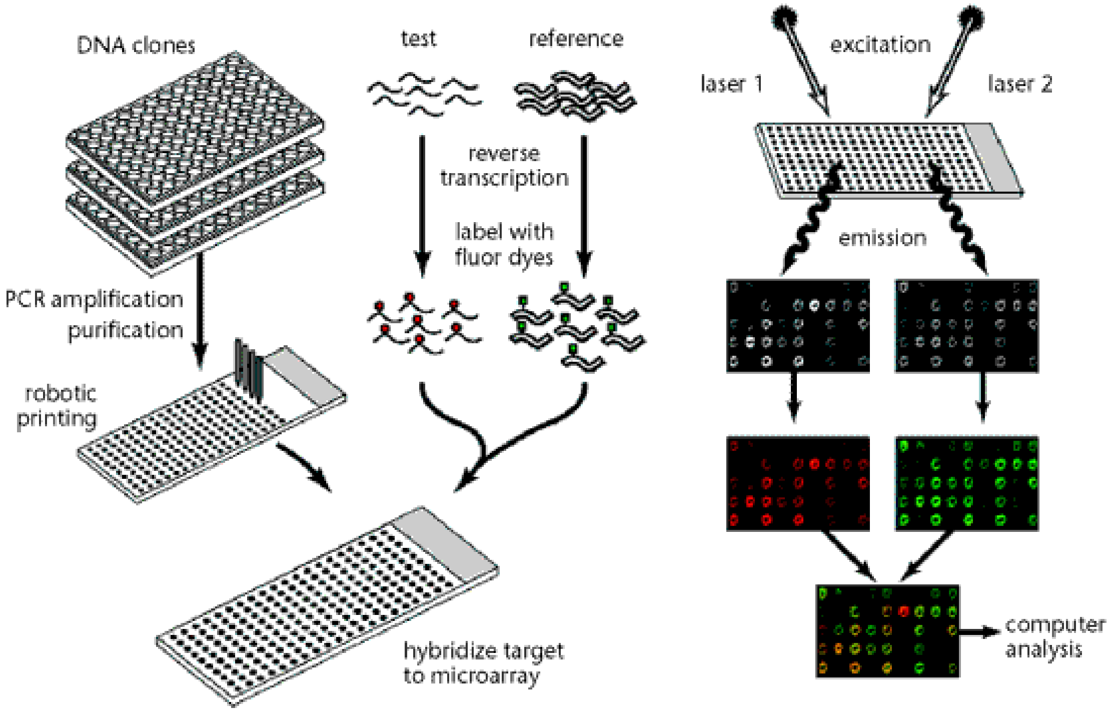


Fig. 1.1. cDNA Microarray Schema (Adapted from Duggan et al., 1999)

Ideally, every cDNA (or the sequenced part of the cDNA termed EST) should represent a unique gene or alternate splice variant. With the completion of whole genome sequences, new sets of probes are being assembled that include genomic clones representing predicted genes for which no EST has yet been identified. If a gene is abundantly transcribed in the cells, it will be represented often in the cDNA library producing redundant clones. Normalization procedures are used to reduce the frequent representation of highly expressed genes. Arrays for higher eukaryotes are typically based on ESTs, whereas for yeast and prokaryotes, probes are usually generated by amplifying genomic DNA with gene specific primers (Duggan et al., 1999). Spotted long oligonucleotide arrays were introduced as an alternative to spotted cDNA arrays and *in situ* synthesized oligonucleotide arrays (Kane et al., 2000). Spotted oligonucleotide arrays are produced by deposition (or spotting) of solution containing synthetic oligonucleotides, typically 40-90 bases long, on a solid substrate.

1.2.3 Amplification and printing

Probes are usually amplified by the use of Polymerase chain reaction (PCR), and spotted onto a coated glass microscope slide or a nitrocellulose or nylon membrane. Membranes are most suited to applications where radioactivity is used to label the respective target sample, while glass slides are used in florescence-based detection. Robots (arrayers) are required to place a large number of probes onto the slides. One platform used for printing microarrays is the common microscope slide, with dimensions of 25 mm x 75 mm. The latest robotic printers can easily fit 50,000 spots or elements onto one slide if the spots are 100 µm in diameter and spaced 50 µm apart (Barrett and Kawasaki, 2003)

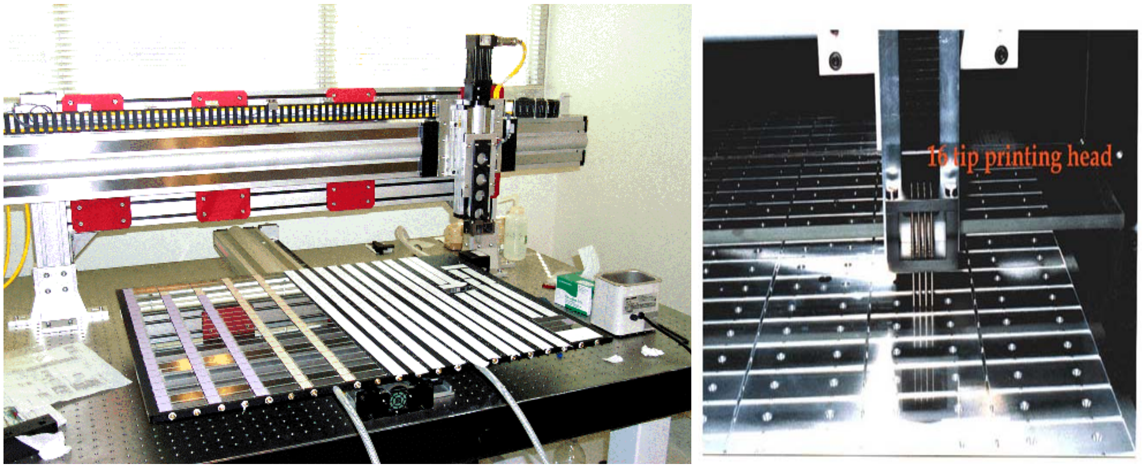


Fig. 1.2. Microarray printing robot from DeRisi microarrayer version II with 16 tip printing head. The microarray core from DeRisi Lab has recently assembled microarrayer version II. DeRisi's version II is capable of printing two batches per week, each batch containing 250 slides, with each slide containing a 40,000 spot array. (Adapted from DeRisi lab website http://derisilab.ucsf.edu/cshl/)

1.2.4 Target labeling, hybridization and image processing

The cellular mRNA is extracted from samples of interest. Target cDNA is prepared from extracted mRNA samples by reverse transcription. Typically reverse transcription from an oligo-dT primer is used for this purpose. Fluorescent dyes such as Cy3 and Cy5 are commonly used for labeling the cDNAs. These dyes are chemically coupled to the poly-dT oligonucleotide that is used to prime the polymerization. If a radioactive label is used, it is incorporated directly on one of the nucleotides. During the hybridization step, the DNA probes on the glass slides and the labeled cDNA target form hetroduplexes. As array technology has advanced, more sensitive and quantitative methods for target preparation are now available. Modern microarrays have been reported to detect the presence of even one mRNA per cell, that is, a concentration of one mRNA per ≥ 100,000 molecules (Barrett and Kawasaki, 2003).

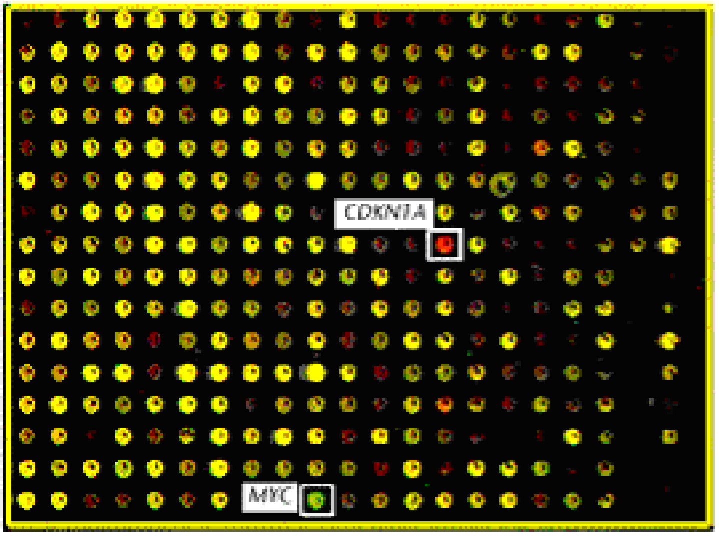


Fig. 1.3. A segment of cDNA microarray (Adapted from Duggan et al., 1999) to which targets from λ irradiated human leukemia-derived ML1 cells (red) and untreated ML1 cells (green) were hybridized. Highly differential hybridization is visible at the detectors for *CDKN1A* and *MYC* genes (boxed)

The final step is to produce an image of the surface of the hybridized array. When exposed to a light of appropriate wavelength the dyes used in the target probes are excited to a higher energy level by producing fluorescence. The florescence intensity produced by the dyes is captured by scanning the microarray slide.

In a single experiment, labeled cDNA samples derived from test and from reference cells are hybridized to the same microarray. This enables the determination of the relative amount of transcript present in test cells as compared to reference cells by the type of fluorescent signal generated. The expression level of any gene can be directly correlated by the log ratio of the measured fluorescent intensity of test versus reference cells as shown in equation 1.1.

$$E=\log (Cy5/Cy3) \qquad (1.1)$$

Higher fluorescence intensity for one spot on the array does not necessarily mean that the respective gene is expressed at a higher level than genes that produce weaker fluorescence signals. This is because the fluorescence intensity depends on many factors, including the length of the probe, the amount of label incorporated into the target sample during reverse transcription, and the efficiency of hybridization. Due to the above mentioned theoretical and experimental reasons, ratios are preferred as the standard for comparison of gene expression (Eisen et al., 1998). Further, competitive hybridization removes variation among arrays from the analysis.

1.3 Affymetrix Genechip arrays

1.3.1 Technology

The basic idea of the Affymetrix oligonucleotide arrays (or DNA chips) is similar to that of spotted DNA arrays. However, the oligonucleotides of length 25 bases are not spotted, but are synthesized on the chip using photolithographic techniques. Agilent*, another company producing arrays with 65 nucleotides, employs a technology that uses the inkjet printing technique to create microarrays. In Affymetrix arrays, each gene is represented by between 10 to 20 different oligonucleotides to control for variation in hybridization efficiency due to factors such as GC content. Since the oligonucleotides are shorter, these chips are usually denser. For instance, a chip with a dimension of 1cm^2 can easily contain 1 million oligonucleotides probes.

*Agilent Technologies, Inc. Palo Alto, CA 94306, USA

In order to control the cross-hybridization with similar short sequences in transcripts, a mismatch control that has a single base change at the 13th base position, labeled mismatch (MM) probe, is included adjacent to each probe called perfect match (PM). Under high stringency conditions this control should not hybridize. RNA from cells or tissues is extracted and the corresponding complementary DNA (cDNA) is generated using reverse transcription. cDNAs are transcribed *in vitro* by means of the T7 RNA- Polymerase and are labeled with biotin further to produce complementary RNA (cRNA) from the cDNA template.

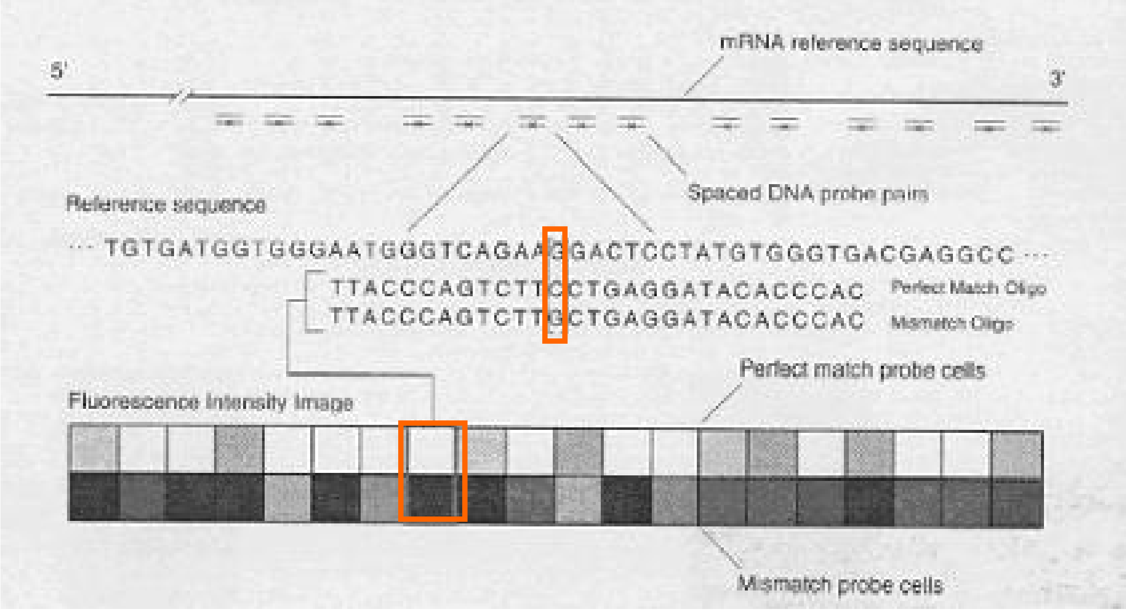

mRNA reference sequence
5'
3'
Spaced DNA probe pairs
Reference sequence
··· TGTGATGGTGGGAATGGGTCAGAAGGACTCCTATGTGGGTGACGAGGCC ···
TTACCCAGTCTTCCTGAGGATACACCCAC
Perfect Match Oligo
TTACCCAGTCTTGCTGAGGATACACCCAC
Mismatch Oligo
Perfect match probe cells
Fluorescence Intensity Image
Mismatch probe cells

Fig. 1.4. A probe set of 20 PM (Perfect match), MM (Mismatch) pairs. (Adapted from Lipshutz et al., 1999). Oligonucleotide probes are chosen based on uniqueness criteria and composition design rules. For eukaryotic organisms, probes are chosen typically from the 3´ end of the gene or transcript (nearer to the poly (A) tail) to reduce problems that may arise from the use of partially degraded mRNA. The use of the PM minus MM differences averaged across a set of probes greatly reduces the contribution of background and cross–hybridization and increases the quantitative accuracy and reproducibility of the measurements.

Labeled cRNA targets are hybridized to probes attached to the solid support. The hybridized samples are visualized after excited by the laser and the fluorescence intensity signal is measured for each hybridized probe. Difference between the perfect match and mismatch signal of the probes termed difference scores is calculated (PM – MM= difference score per probe set). Significant difference scores are used to calculate an "average difference" that directly correlate to the mRNA abundance of the gene.

1.3.2 Manufacturing and using oligonucleotide arrays

The high density arrays are produced, using photolithography and combinatorial chemistry. Oligonucleotides are built base by base on the surface of the array.

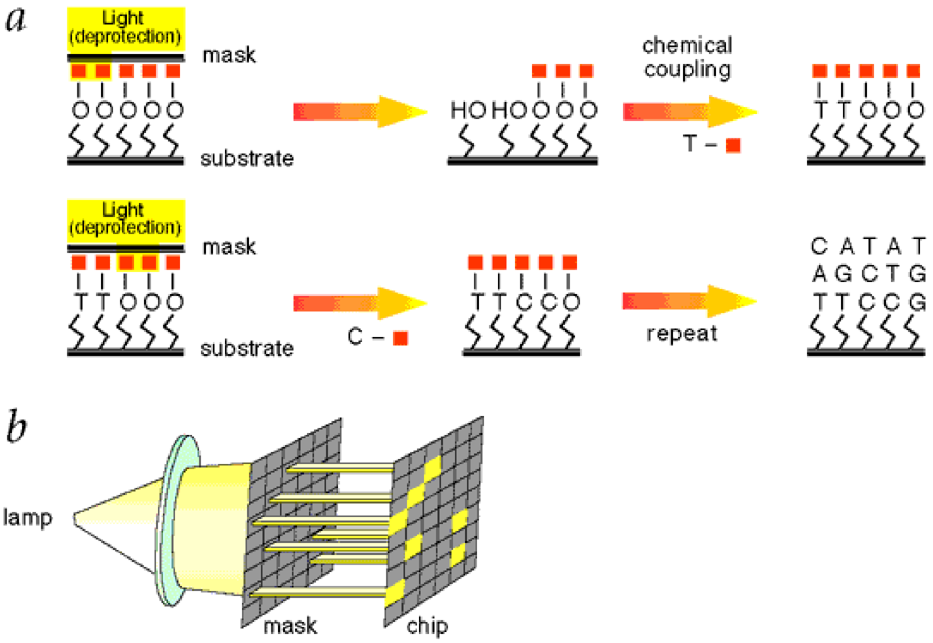


Fig. 1.5. Manufacturing of Affymetrix oligonucleotides arrays adapted from Lipshutz et al., (1999). ***(a)*** Light directed oligonucleotide synthesis. A solid support is derivatized with a covalent linker molecule terminated with a photolabile protecting group. Light is directed through a mask to deprotect and activate selected sites, and protected nucleotides couple to the activated sites. The process is repeated, activating different sets of sites and coupling different bases allowing arbitrary DNA probes to be constructed at each site. ***(b)*** Schematic representation of the lamp, mask and array.

This takes place by the covalent reaction between the 5' hydroxyl group of the sugar of one nucleotide to be attached and the phosphate group of the adjacent nucleotide. Each nucleotide added to the oligonucletide on the glass has a protective group on its 5' position to prevent the addition of more than one base during each round of synthesis. In each cycle, a localized flash of light "deprotects" the growing nucleotide chain just on that position where the next nucleotide should be added. By inserting a mask between the light and the chip, the localization of the light is achieved. The process is repeated until the probes reach their full length, usually 25 nucleotides. Several hundred thousands of oligonucleotides with their mismatch controls can be rapidly synthesized on thousands of identical chips.

1.4 Microarray Data Mining

Microarray experiments are providing unprecedented quantities of genome-wide data on gene expression patterns. The real power of microarrays is in their ability to study the relationship between genes or samples that behave in a similar or coordinated manner. Starting from microarray data, the first major computational task is to cluster genes into biologically meaningful groups according to their pattern of expression (Quackenbush, 2001). Those genes that share similar expression patterns could imply that they are co-regulated, which in turn may imply that these genes are involved in a similar biological function.

The results of DNA chip experiments are usually organized together in a gene expression matrix, with rows corresponding to genes and columns corresponding to conditions. Since each row is corresponding to a single gene measured over different conditions, generally each row is called as an expression profile and mathematically it represents a row or (gene) vector. There are two issues that we will be interested in while doing the gene expression data analysis: either we are interested in those genes that share similar expression profiles under a set of given conditions, or we are looking for those conditions that trigger the co-expression of a given group of genes. Expression data analysis can be loosely divided into 1) Internal analysis and 2) External analysis (Gerstein and Jansen, 2000). In the internal analysis, the numerical structure of the data is analyzed by doing clustering. In the external analysis, the expression measurements are related to other biological information like protein function, structure, and regulation and so on. Many standard algorithms have been proposed to perform internal analysis. This section gives a brief overview of the standard clustering algorithms but is not meant as a review. For detailed information refer to the review by Moreau et al., (2002).

1.4.1 Internal analysis

Internal analysis of gene expression data mainly involves normalization of the data, calculating a similarity measure and clustering or partitioning the data. The starting point of internal analysis is to normalize the data and then to define a measure of the similarity, for example, by means of a correlation coefficient between expression profiles.

1.4.2 Data normalization

Before calculating the similarity measure, it is common to center gene expression profiles to ensure that they have a mean equal to 0 and a standard deviation equal to 1. For an expression profile *x* having N measurements, the normalized profile X can be computed as a Z-score from the measured expression profile *x* through the relation

$$\mathrm{X}(\mathrm{k}) = \frac{x(\mathrm{k}) - x_{\mathrm{avg}}}{\sigma_x} \qquad (1.2)$$

where $\boldsymbol{x}_{\mathbf{avg}}$ denotes the average and $\boldsymbol{\sigma}_{\mathbf{x}}$ denotes the standard deviation of values in x, and **X (k)** and ***x* (k)** are the **k**th components of their respective profiles.

1.4.3 Correlation coefficient

A first step in calculating the similarity measure is to define a correlation coefficient between two expression profiles. The correlation should be a high value for co-expressed genes and low for genes with unrelated expression patterns.

The correlation coefficient takes a value between -1 and +1. A value of –1, representing a strong negative correlation between the two profiles, suggests that if one profile is expressed high the other will be expressed low. A value of +1, representing a strong positive correlation suggests that both profiles are going high or low simultaneously. A value of 0 represents no correlation between two profiles.

On the other way, it is common to describe the similarity between two profiles in terms of the distance d_{dis} between them in the high-dimensional space of gene expression or sample measurements. Distance d_{dis} is given by

$$d_{dis} = 1 - r \tag{1.3}$$

where r represents the similarity measure between two profiles.

Commonly used similarity measures include Euclidean distance, Pearson correlation, Jackknife correlation and rank correlation.

1. **Euclidean distance**

Euclidean distance is the simplest measure of similarity between two expression profiles. It calculates the straight line distance between two profiles by mapping them as vectors into *n* dimensional space called Euclidean space. The distance between profiles *x* and *y* in a Euclidean space is given by

$$d = | x - y | = \sqrt{\sum_{i=1}^{n} |x_i - y_i|^2} \tag{1.4}$$

2. Pearson correlation coefficient

In many cases, the Pearson correlation coefficient is used as a similarity measure. For two normalized expression ratio profiles X_i and X_j, each with an average of 0 and a standard deviation of 1, the Pearson correlation coefficient r_{ij} is given by the dot product of two respective profiles.

$$r_{ij} = \frac{1}{N - 1} X_i \cdot X_j \tag{1.5}$$

Given a group of G genes, we can compute the correlation coefficient matrix r, where each element (r_{ij}) of the matrix denotes the Pearson correlation coefficient between genes *i* and *j*.

3. Jackknife Coefficient

Jackknife coefficient is normally used in specific signal outlier cases. If the expression levels of two ORFs are completely unrelated at all but one condition, and both ORFs have a high peak or valley at the remaining conditions, then the correlation coefficient will be very high. An outlier of this type can occur because of an experimental error. For an expression profiles pair *i, j*, let ρ_{ij} denote the correlation of the pair *i, j*; also, let $\rho^{(1)}_{ij}$ denote the correlation of the pair *i, j* computed with the l^{th} observation deleted. For a data set with *t* observations, the Jackknife correlation is defined as J_{ij} and given by

$$J_{ij} = \min \{\rho^{(1)}_{ij} \ldots \rho^{(2)}_{ij}, \ldots \rho^{(t)}_{ij}, \ldots \rho_{ij}\} \tag{1.6}$$

4. Rank correlation

Spearman rank correlation r_s is a non-parametric correlation that calculates a measure of the strength of the association between two variables. The first step in finding r_s is to rank the values of each of the variables separately; ties are treated by averaging the tied ranks. Then, r_s is computed in exactly the same way as the simple correlation coefficient. The only difference is that the values of x and y that appear in the formula for r_s denote the ranks of the raw data rather than the raw data themselves. The Spearman rank correlation coefficient is defined as

$$r_s = \frac{SS_{xy}}{\sqrt{SS_{xx} SS_{yy}}} \tag{1.7}$$

where

$$SS_{xx} = \sum (x - \bar{x})^2, \quad SS_{yy} = \sum (y - \bar{y})^2, \quad SS_{xy} = \sum (x - \bar{x})(y - \bar{y})$$

when there are no ties, the formula for r_S, reduces to

$$r_s = 1 - \frac{6\sum d^2}{n(n^2 - 1)} \tag{1.8}$$

where d is the difference between the values of x and y corresponding to a pair of observations. This simple formula will provide a good approximation to r_S when the number of ties in the ranks is small.

Because it uses ranks, the Spearman rank correlation is much easier to compute. Spearman rank correlation is a measure of association that is used when the distribution of the data makes Pearson correlation coefficient undesirable or misleading. Hence this measure is the method of choice in our study.

1.4.4 Cluster Analysis

Cluster analysis aims at identifying subgroups or clusters of co-expressed genes in a collection of gene expression profiles. The most obvious use of grouping the genes into clusters is to get an improved understanding of transcription regulatory networks within genomes. Genes with similar profiles are most likely to be subject to the same or related transcriptional control. Co-expression of genes can be an important observation to infer the biological role of the genes. For example, co-expression of a gene of unknown biological function within a cluster of genes with known function indicates that the unknown functional gene might also function the same as that of the genes in the cluster (Bilu and Linial, 2002). The standard algorithms for grouping genes with related profiles include hierarchical clustering, K-means clustering and self-organizing maps (Moreau et al., 2002).

1. Hierarchical clustering

Hierarchical clustering is a widely used method for clustering gene expression data. Given the gene expression profile of N genes, these algorithms produce a hierarchy (dendogram) in which each node represents a gene to which the most similar genes are connected according to the correlation among them.

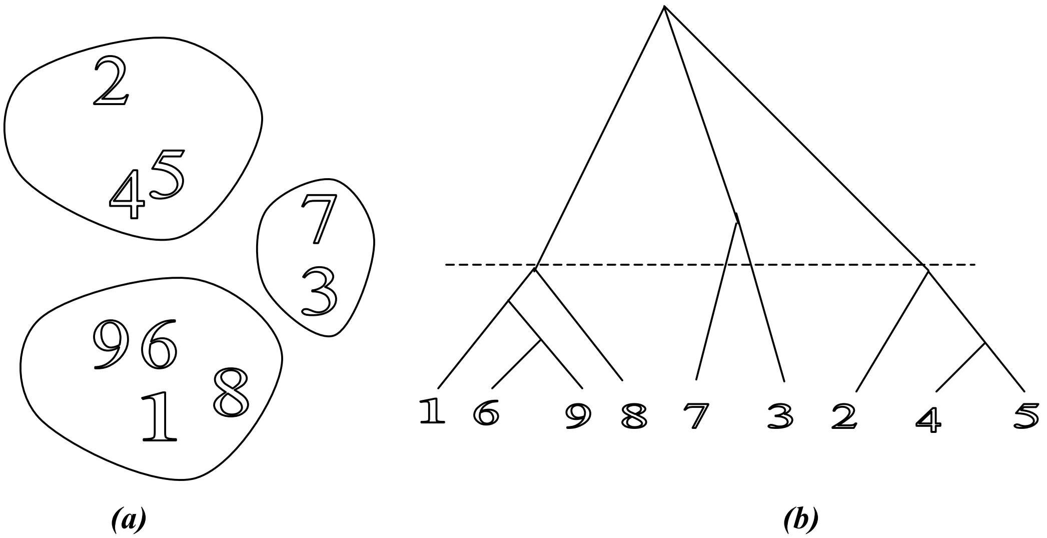


Fig. 1.6. Bottom-up agglomerative clustering. ***(a)*** Expression profiles are grouped according to similarity or distance between them. 3 groups of clusters each corresponding expression profiles of genes {2, 4, 5}, {9,6,8} and {7,3} are shown. ***(b)*** The genes with smaller distance are mapped as a leaves in a tree, for example the distance between 4 and 5 is smaller than that of any other pair, so 4 and 5 are put together and others are added iteratively.

Each level of the tree represents a partition of the input data into several nested clusters or groups. The advantage of hierarchical clustering is that the number of clusters does not to be specified beforehand. There are two styles of hierarchical clustering algorithms to build a tree from the input set of expression profiles: (a) Bottom-up approach (agglomerative clustering) and (b) Top-down approach (divisive clustering).

a) Bottom-up approach

In this approach, every gene expression profile is initially assigned to a single cluster. The similarity or distance between any couple clusters is calculated using any similarity measure discussed in section (1.4.3).

Clusters with the most similar expression profiles (or closest in distance) are merged first, and those with more diverse profiles are merged iteratively. This process gives rise to a tree structure (Fig. 1.6), where the height of the branch is proportional to the pair-wise similarity (or distance) between the clusters. The root of this tree corresponds to the whole input of gene expression profiles and each leaf corresponds to a single gene expression profile. Clusters are formed by cutting the tree at a certain level or height. Agglomerative clustering is the most commonly used hierarchical clustering method.

b) Top-down approach

Divisive clustering is the example of top-down hierarchical clustering. It clusters in the opposite way to that of agglomerative clustering. The entire input set of gene expression profiles are first considered to be one cluster and then they are broken down into smaller and smaller subsets until each subset consists of only a single entity (gene profile). Hierarchical clustering was the first method applied to gene expression data by Eisen et al., (1998). Serum cluster resulting from hierarchical clustering of human fibroblasts dataset from Eisen et al. is shown in Fig. 1.7.

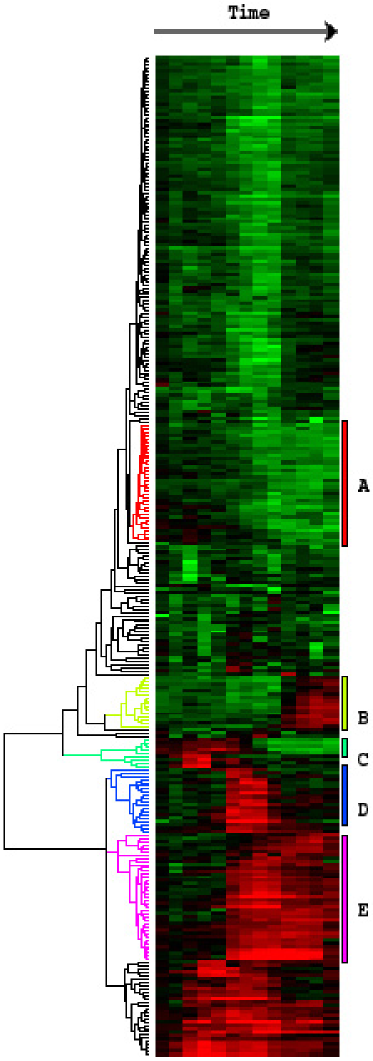


Fig. 1.7. Hierarchical clustering of serum cluster from Human fibroblasts microarray data adapted from Eisen et al., (1998). Clustered display of data from time course of serum stimulation of primary human fibroblasts is shown. Five separate clusters are indicated by colored bars and by identical coloring of the corresponding region of the dendrogram. These clusters contain multiple genes involved in (*A*) cholesterol biosynthesis, (*B*) the cell cycle, (*C*) the immediate-early response, (*D*) signaling and angiogenesis, and (*E*) wound healing and tissue remodeling. These clusters also contain named genes not involved in these processes and numerous uncharacterized genes.

2. K-means clustering

K-means clustering is a non-hierarchical approach for forming good clusters. It partitions the data into a predefined number of partitions or clusters. K-means clustering starts with a number of randomly divided K initial clusters from the gene expression profiles. The cluster center is calculated iteratively by taking the average of all the expression profiles in each cluster, followed by a reassignment of the gene expression vectors to the cluster with the closest cluster center (Fig. 1.8).

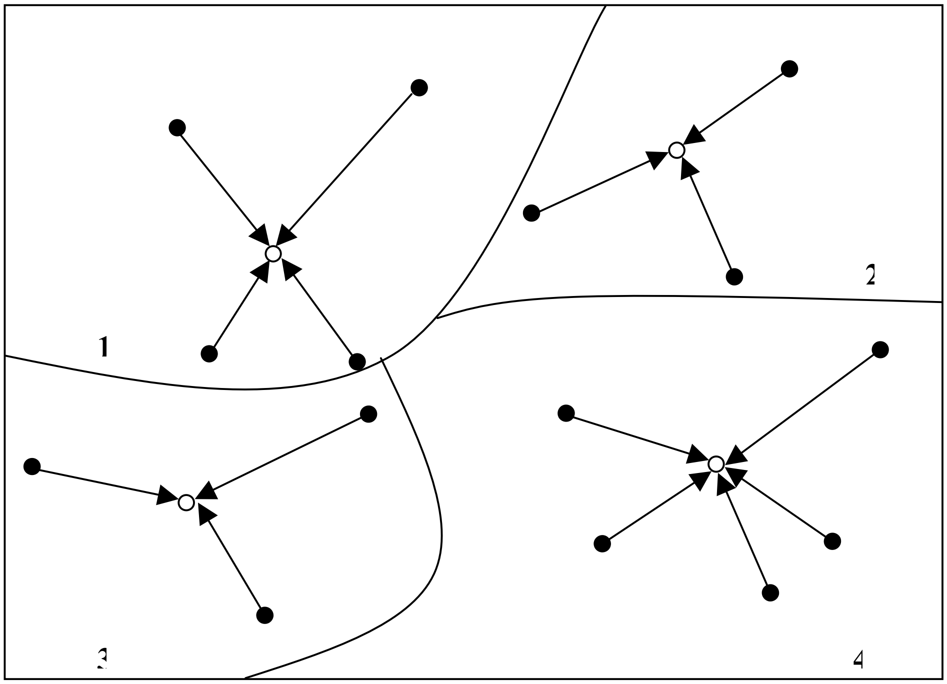


Fig. 1.8. K-means clustering example: The whole dataset is divided into defined number of 4 clusters. The center of the each cluster is calculated and each gene is assigned to its nearby cluster center.

Convergence is reached when the cluster center remains unchanged. In practice, the arbitrary definition of cluster number predefinition makes it necessary to use a trial and error approach.

3. Self organizing maps (SOMs) (Tamayo et al., 1999)

Self-organizing maps (Tamayo et al., 1999) are a data visualization technique that reduces the dimension of data through the use of self-organizing neural networks. In the first step of constructing SOMs, the geometry of the nodes is chosen. A SOM has a set of nodes with a simple topology (e.g., two-dimensional grid) and a distance function d (N1, N2) on the nodes. The nodes are mapped into k-dimensional gene expression space (in which the *i*th coordinate represents the expression level in the *i*th sample). In each iteration, the weight vector associated with a node G is randomly selected and nodes are moved towards G. The closest node N_G is moved the most, whereas other nodes are moved by smaller amounts depending on their distance from N_G in the initial geometry.

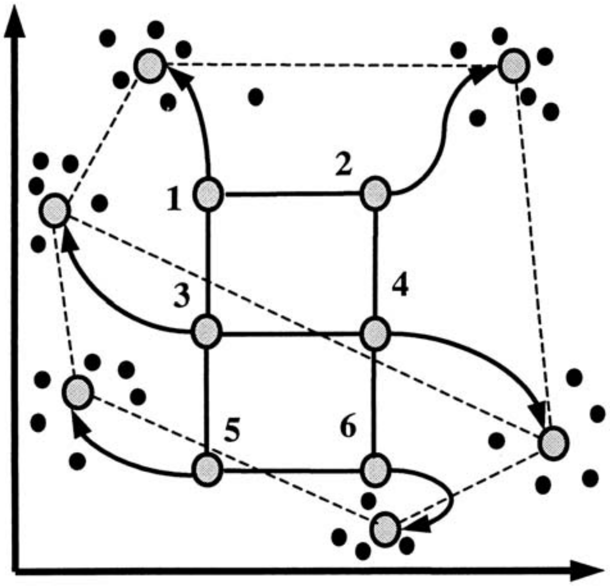


Fig. 1.9. Principle of SOMs (Adapted from Tamayo et al., 1999). Initial geometry of nodes in 3 × 2 rectangular grid is indicated by solid lines connecting the nodes. Hypothetical trajectories of nodes as they migrate to fit data during successive iterations of SOM algorithm are shown. Data points are represented by black dots, six nodes of SOM by large circles, and trajectories by arrows.

1.4.5 External Analysis

External analysis of gene expression data involves applying expert biological knowledge to interpret the clusters resulting from internal analysis. It is often done by explicitly integrating information like functional classification or transcriptional factor binding site information, directly into the data analysis to give more insight regarding the functional role of the genes in the clusters.

1.4.6 Functional classification

An immense amount of research has been reported on functional classification suggesting the ability of expression data to predict function, interaction or localization (Hughes et al., 2000; Bussemaker et al., 2001; Jansen et al., 2001; Brown et al., 2000; Altman and Raychaudhuri, 2001; Gerstein and Jansen, 2000). Similar expression profiles of genes might indicate similar functional and cellular roles.

There are a number of schemes for classifying protein function. The functional annotation information first appeared in databases of gene products such as SWISS-PROT or PIR. In these databases, protein entries were accompanied by careful human generated annotations of their empirically determined or predicted role (Bairoch and Apweiler, 1999; Barker et al., 1999). Many efforts have been made to classify such databases on the basis of their annotation (Tamames et al., 1998; Eisenhaber and Bork, 1999; Licciulli et al., 1999). Most of these classification schemes concentrate on a single organism, for example MIPS for *S. cerevisiae* (Mewes et al., 2002), GenProtEC for *E. coli* (Serres et al., 2004), FlyBase for *Drosophila melanogaster* (Drysdale et al., 2005) and EGAD for human ESTs (www.tigr.org/tdb/egad/egad.shtml).

There are other schemes available in the literature that classify a subsets of functions across many organisms, for example, ENZYME for enzyme function (Bairoch, 2000), Ecocyc for encyclopedia of *Escherichia coli* K12 genes and metabolism (Keseler et al., 2005), and KEGG for pathways (Kanehisa and Goto, 2000). The Gene Ontology project (GeneOntologyConsortium, 2000) is the new effort that focuses on merging the functional classifications for different organisms into one common source. The GO project produces a controlled vocabulary that can be applied to all organisms, even though knowledge of gene and protein roles in cells are accumulating and changing.

1.4.7 Transcriptional factor binding site information

In any organism, the gene expression is regulated by complex mechanisms that involve many transcription factors. These controls are much more complex in eukaryotes than in prokaryotes. Transcription factors bind to particular DNA sequences, called transcription factor binding sites. These sites are believed to occur within several hundred base pairs upstream of the respective ORFs, but may even bind to sites greater than 1Mb downstream of its target gene (Nobrega et al., 2003). The transcription factor binding sites are assumed to be of 5 – 25 base pairs long. The regulation of gene expression in eukaryotes often occurs through the coordinated action of multiple transcription factors. Combinatorial regulation of transcription has several advantages, including the control of gene expression in response to a variety of signals from the environment and the use of a limited number of regulators whose activities are modulated by a diverse set of conditions. Many studies showed combinatorial transcriptional control in several organisms (Kel et al., 1995; Quandt et al., 1996; Yuh et al., 1998; Wang et al., 1999; Halfon et al., 2000; Fickett and Wasserman, 2000). It is commonly believed that, those genes that share similar expression profiles probably have the same upstream element. This fact has been exploited by Roth et al., (1998), J. van Helden et al., (1998), Brazma et al., (1998) and Tavazoie et al., (1999) to search for new transcriptional factor binding site sequences.

1.5 Drawbacks of standard clustering methods

1.5.1 Dataset

In any typical array experiment, many genes of an organism are assayed under multiple conditions. These conditions may be of two types

1) Different time points during a biological process, such as the yeast cell cycle (Cho et al., 1998; Spellman et al., 1998), yeast sporulation (Chu et al., 1998) or *Drosophila melanogaster* development (White et al., 1999). The data produced from these types of microarray experiments are called 'time series' data.

2) There can be different tissue samples that are independent measures of gene expression with some common phenotype, such as tissue type or malignancy. The data produced from these experiments are called 'independent' condition data.

1.5.2 Drawbacks of clustering methods on time series data

A major drawback of the standard clustering methods is that they ignore many additional relationships inherent to time series data, for example the time delayed relationship between a transcription factor and the gene that is activated by transcription factor. For example, in yeast, the expression profiles of Arc35 and Arp2/3 genes show a time-delayed response with the expression of Arc35 being 20 minutes delayed compared to Arp3 that corresponds to one time point in the yeast cell cycle time course. Arp2 and Arp3 are highly conserved actin-related proteins, and are localized to the actin cytoskeleton. This complex is involved in endocytosis and actin cytoskeleton organization, and binds actin and profilin. Arc35 is one of the subunits of the Arp2/3 complex. Arc35 has been implicated as a regulator of calmodulin localization. Calmodulin is a calcium sensor in yeast localization. The results from Schaerer-Brodbeck and Reizeman, (2000) suggest that the calmodulin-dependent function of the Arp2/3 complex is mediated by the Arc35 subunit, although other subunits could be required as well. Arc35 works through two genetically separatable calmodulin functions to regulate the actin and tubulin cytoskeletons.

The standard clustering methods focus on global correlation over whole time series, by identifying clusters of genes whose expression changes simultaneously. However, they are prone to miss the local time-delayed relationships. Further, the standard similarity measures do not depend on the ordering of the measurements, i.e., they do not exploit this additional information.

1.5.3 Drawbacks of clustering methods on independent condition data

One of the important drawbacks of standard clustering methods on independent condition data is, clustering according to all experimental conditions. However, few conditions could trigger important processes in an organism. For example, in clinical studies gene expression measurements are often done on tissues taken from patients with a medical condition. Using assays, biologists have identified molecular fingerprints that can help in the classification and diagnosis of the patient status and guide treatment protocols (Alizadeh et al., 2000; Ramaswamy et al., 2001). In these studies, the focus is primarily on identifying profiles of expression over a subset of the genes that can be associated with clinical conditions and treatment outcomes, where the set of samples is equal in all stage of the disease. Application of standard clustering algorithms in a clinical dataset results in clusters that incorporate all clinical conditions. Clustering data over diverse conditions often miss those subset conditions that affect a subset of genes of clinical interest.

In order to overcome the problems stated above, sophisticated algorithms that increase the biological knowledge of gene expression data are needed.

Chapter 2

Predicting time delayed and local correlations

2.1 Method using dynamic programming

The first method we developed to address the drawbacks mentioned in the section 1.5.2 was a method using dynamic programming algorithm. A typical gene expression time course data consists of gene expression measurements of many genes under many time points.

Consider the data from the microarray time course experiments as a gene expression matrix *A*. The matrix element a_{ij} denotes the normalized expression change of a gene $i \in m$, at time point $j \in n$, where m represents number of genes taken for analysis and n represents the number of time points.

$$\mathrm{A} = \begin{pmatrix} a_{11} & a_{12} & . & a_{1\mathrm{n}} \\ a_{21} & a_{22} & . & a_{2\mathrm{n}} \\ . & . & & . \\ a_{\mathrm{m}1} & a_{\mathrm{m}2} & . & a_{\mathrm{mn}} \end{pmatrix}$$

By using vector notation, the expression matrix X can be viewed as a collection of row vectors.

$$\mathrm{A} = \begin{pmatrix} r_1 \\ r_2 \\ . \\ . \\ . \\ r_m \end{pmatrix}$$

where any row vector $r_i = (a_{\mathrm{i}1}, a_{\mathrm{i}2}, \ldots a_{\mathrm{in}})$ represents the gene expression profile for gene

$i \in m$. Thus, gene expression profiles can be viewed as *m* vectors of *n* dimensions. Here, *m* represents the number of genes and *n* represents the total number of time points.

Given the vector notation for gene expression profiles, a sub vector can be considered as a locally similar segment in *n* dimensional space between two row vectors. Optimal sub vector represents the best locally matching segment between two row vectors.

2.1.1 Method

In order to find the optimal sub vector between two expression profiles, the modified version of the dynamic programming algorithm, more specifically the Smith-Waterman algorithm (Smith and Waterman, 1981) without gaps for locally similar matches was adopted in our approach. The Smith-Waterman algorithm implements a dynamic programming technique that takes alignments of any length, at any location, in any sequence, and determines whether an optimal alignment can be found. In order to get a score for an optimal alignment, the following simple scoring scheme was used.

$$S_{i,j} = a_{\mathrm{xi}} * a_{\mathrm{yj}}\,, \qquad (2.1)$$

where x, y $\in m$ for any i, j $\in n$.

2.1.2 Dynamic programming algorithm

The dynamic programming algorithm takes three steps to compute an optimal alignment,

1. Initialization
2. Scoring
3. Trace back

1. Initialization step

In the first step, a matrix with i +1 columns and j+ 1 rows is created, where i, j ∈ T, corresponds to the length of two profiles (row vectors). The first row and first column of the matrix is filled initially with 0.

Consider two profiles a_1 = (1, 0.5, 2, -1, 0.6) and a_2 = (0.9, 0.6, 1.5, -1, 0.7) each of length *5*. The application of dynamic programming algorithm steps on this data would produce results like shown below.

Table 2.1. Dynamic programming initialization step.

		1	*0.5*	*2*	*-1*	*0.6*
	0	0	0	0	0	0
0.9	0					
0.6	0					
1.5	0					
-1	0					
0.7	0					

2. **Scoring step**

In the second step, the matrix is filled by calculating the positive score $P_{i,j}$ defined by the equations 2.2 below.

$$P_{i,j} = Maximum\ \{[P_{i-1,j-1} + S_{i,j}], 0\} \tag{2.2}$$

Here, $S_{i,j}$ represents the scoring scheme defined in equation 2.1. The scoring step for the two profiles from Table 2.1 is explained in Table 2.2. Starting from the upper left hand corner the matrix is filled for each position in the matrix until all the cells are filled.

Table 2.2. Dynamic programming positive scoring steps. The value at each cell is calculated as the score $P_{i,j}$ = Maximum $\{[S_{i,j} + P_{(i-1,j-1)}], 0\}$.

		1	*0.5*	*2*	*-1*	*0.6*
	0 ↘	0 ↘	0 ↘	0 ↘	0 ↘	0
0.9	0 ↘	*0+0.9* ↘	*0+0.45*	*0+1.8*	*0*	*0+0.54*
0.6	0 ↘	*0+0.6*	0.9+0.3 =1.2			
1.5	0 ↘	*0+1.5*				
-1	0 ↘	*0*				
0.7	0	*0+0.7*				

3. Trace back step

After the matrix fill step, the trace back step determines the actual segment(s) that result in the maximum score. Note that using the scoring scheme defined in equation 2.1, it is likely that there are multiple maximal segments. The one with the maximum score of all is considered as the optimal alignment and hence could represent the optimal sub vector. The trace back step begins in the x, y position of the matrix, i.e. the position that leads to the maximal score and traces back until the starting point.

Table 2.3. Dynamic programming trace back step. From the example the maximum value occurs at diagonal element $P_{5,\ 5}$ indicating a global correlation between the two profiles.

		1	0.5	2	-1	0.6
	0	0	0	0	0	0
0.9	0 ↖	0.9	0.45	1.8	0	0.54
0.6	0	0.6↖	1.2	0.75	0.15	0.36
1.5	0	1.5	1.35↖	4.2	0	2.4
-1	0	0	1	0 ↖	5.2	0
0.7	0	0.7	0.35	2.4	0 ↖	5.62

For two simple profiles with time shifted local correlations a_1 = (1, 2, 3, 0, -1) and a_2 = (-2, -1, 1, 2, 3) each of length *5*. The application of dynamic programming algorithm steps on this data would produce results like shown below in Table 2.4.

Table 2.4. Dynamic programming steps for the time-delayed profiles. From the example the maximum value occurs not at the diagonal element indicating a local time delayed correlation between the two profiles.

		1	2	3	0	-1
	0	0	0	0	0	0
-2	0	0	0	0	0	0
-1	0	0	0	0	0	1
1	0	1	2	3	4	5
2	0	2	5	8	0	0
3	0 ↖	3	8	14	0	0

```
Algorithm: Method using dynamic programming
  Input: Gene expression time course data matrix A with G genes measured over T time points.
  Output: List of optimal sub vectors with their corresponding alignment scores consists of both direct and time delayed correlations.
  Init: Scan the input expression matrix and store number of time-points and row vectors.
  Iteration:
  Select any two row vectors $r_i = (a_{i1}, a_{i2}, \ldots a_{iT})$ and $r_j = (a_{j1}, a_{j2}, \ldots a_{jT})$, i, j $\in$ G.
  Start dynamic programming initialization step
  Calculate the positive score and negative score for each alignment
  Check for the maximum score and thus the optimal sub vector alignment
  Perform the trace back step.
```

Fig. 2.1. Pseudo code of the method using dynamic programming

2.1.3 Application of the method on the real datasets

In order to test the method, DNA microarray experiments from the two different time series array experiments summarized in Table 2.5 was used. The method was applied on these datasets, and the statistical significance of the output subvectors were tested by calculating the P value corresponding to alignment score.

2.1.4 Datasets

1) The dataset, that measured the gene expression change during yeast cell cycle from Spellman et al., (1998), was considered for analysis. In this experiment, the cell cycle of yeast cells was arrested by addition of the α factor. After 120 minutes the α factor was removed and the gene expression with the synchronized cell culture was arrayed every 7 minutes until 140 minutes.

2) The dataset from Gasch et al., (2000), that measured the gene expression change in yeast during various heat shock conditions was also considered for analysis. In this experiment yeast cells grown at various temperatures were given heat shock by shifting

the temperature from 25° C to 37° C, heat shock from various temperatures to 37° C and temperature shifts from 37° C to 25° C. Gene expression time course experiments were conducted for each experiment at different time points.

Table 2.5. Dataset taken for analysis

Experiment Name	**Time points**	**No of genes taken for analysis**
Cell Cycle Alpha Arrest	18	6075
Heat shock	29	6153

2.1.5 Statistical significance

As in sequence and structural alignments (Altschul et al., 1997; Pearson, 1998; Gerstein and Levitt, 1998) the statistical significance of optimal subvectors were given by P values from the distribution of alignment scores *D*. In order to estimate the P values for a given alignment score, random expression profiles were generated by shuffling the normalized expression values from gene expression matrix *A* at different time points, for example interchanging the expression level at time point 4 to time point 7, a_4 and a_7. The resulting randomly shuffled profiles were still normalized values with an average of 0 and a standard deviation of 1. The method was applied on these random data, and the distribution of the alignment scores *d*, predicted from optimal subvectors was calculated. This distribution was meant to be that of true negatives (true random subvectors). In sequence analysis, a random distribution of similarity scores is used to find a significance level associated with a computed similarity score. By integrating true random subvector distribution with that of original data distribution, the conventional P-value P ($d > D$) was calculated. This probability is defined as the probability of obtaining an alignment score *d* larger than *D* from the random profiles. The smaller the P value is, the greater is the significance of the alignment. The distribution of the match scores in comparisons to actual observed P (*D*) values for each of the datasets mentioned in the previous section are shown in Figures 2.2a and 2.2b.

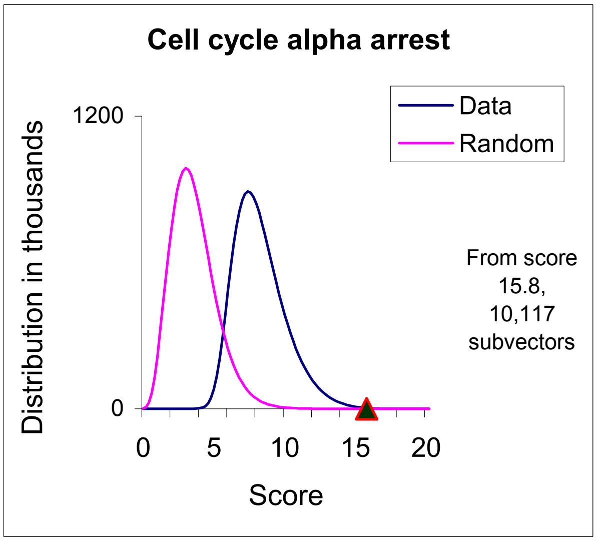


Fig. 2.2a. Relationship between match score D and P-value for the yeast cell cycle dataset and a random dataset.

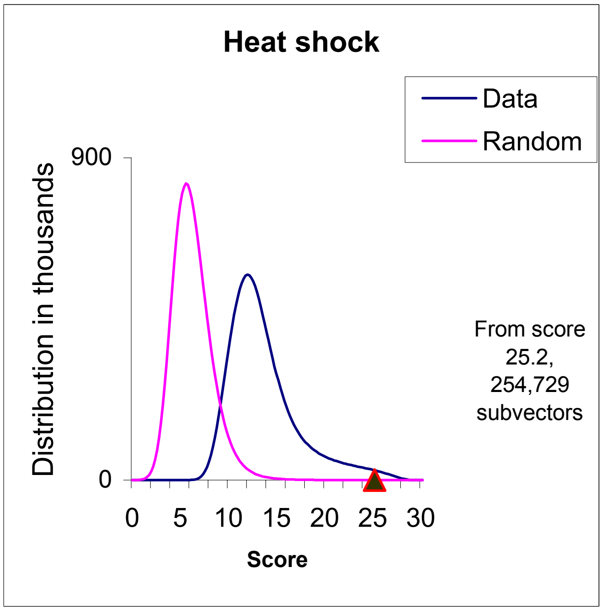


Fig. 2.2b. Relationship between match score D and P-value for the yeast heat shock dataset and a random dataset.

The number of significant optimal subvectors with rounded down probability (P value) ≈ 0* with alignment scores are summarized in Table 2.6. In the cell cycle dataset, 10,117 subvectors were predicted to be significant with alignment score of greater than 15.8. In the heat shock dataset, 254,729 subvectors were predicted to be significant with an alignment score greater than 25.2.

Table 2.6. Significant subvectors predicted from the algorithm with their corresponding alignment scores.

Experiment Name	**Number of subvectors predicted with significant P value**	**Alignment score**
Cell Cycle Alpha Arrest	10,117	15.8
Heat shock	254,729	25.2

2.1.6 The MIPS functional catalog

In depth analysis is usually done on genes and gene products in order to discover, confirm or clarify their function. The concept of 'function' is itself rather vague (Gerstein and Jansen, 2000). Function may represent the biochemical mechanism; at other times, 'function' means the involvement of the gene product in metabolic pathways or cellular processes. The function of a gene product is its *raison d'être* (Rison et al., 2000) understanding it is key to understanding how a limited number of gene products can generate life, from simple unicellular organisms to the complex multi-cellular vertebrates. Ontology systems that integrate various conceptualizations of function need to be established. Ontology systems need to facilitate cross query and annotation transfer as well as a variety of projects that entail interoperation of the ever-increasing biological databases (Lan et al., 2002).

* smaller P value might be due to empirical data distribution

The Munich Information Center for Protein Sequences (MIPS) developed a hierarchical representation of protein function for *Saccharomyces cerevisiae,* the first eukaryotic genome sequenced (Goffeau et al., 1996). Protein functions have been assigned by integrating the available information from significant homologues to functionally characterized proteins as well as data from the literature derived from biochemical, genetic, cellular and phenotypic experiments. MIPS Comprehensive Yeast Genome Database (CYGD) presents comprehensive information about all protein coding regions, as well as RNA-genes. Gene products are assigned to more than one functional category to account for multi functional proteins having more than one function. Biomipsmax funcat version 06.12.2001 had 29 main categories, each containing 3 to 5 levels of subcategories. In total, the catalog had more than 400 functional categories.

Category code	Category name
01	METABOLISM
01.01	Amino acid metabolism
01.01.01	Amino acid biosynthesis
01.05.01	C-compound and carbohydrate utilization
01.05.04	Regulation of C-compound and carbohydrate utilization
02	ENERGY
02.01	Glycolysis and gluconeogenesis
02.07	Pentose-phosphate pathway
02.13	Respiration
02.19	Metabolism of energy reserves (glycogen, trehalose)
03	CELL CYCLE AND DNA PROCESSING
03.01	DNA Processing
03.01.03	DNA synthesis and replication
03.01.05	DNA recombination and DNA repair
03.03	Cell cycle
03.03.01.01	Mitotic cell cycle
03.03.02	Meiosis
04	TRANSCRIPTION
04.01.04	rRNA processing
04.03.03	tRNA processing
04.05.01.04	Transcriptional control
04.05.05	mRNA processing (5’ – 3’- end processing, mRNA degradation)
04.05.05.01	Splicing
04.05.05.09	mRNA editing
04.05.99	Other mRNA-transcription activities
04.07	RNA transport

Fig. 2.3. An excerpt of MIPS functional catalog

In the version used for analysis, 3932 genes out of 6331 yeast genes had at least one function assigned, the remaining genes were assigned to the category unclassified proteins. An excerpt of the MIPS functional catalog is shown in Fig. 2.3. The Updated version of the functional catalog is currently available for downloading at the MIPS ftp site (ftp://ftpmips.gsf.de/yeast/catalogues/funcat/).

2.1.7 Network Topology and Clustering

The resulting subvectors from the algorithm with significant scores from each dataset were clustered using hierarchical clustering algorithm from *Pajek* (Batagelj and Mrvar, 2003). Pajek* is a program package for large network analysis. The resulting clusters were mapped to more than 400 yeast MIPS functional categories described in the previous section.

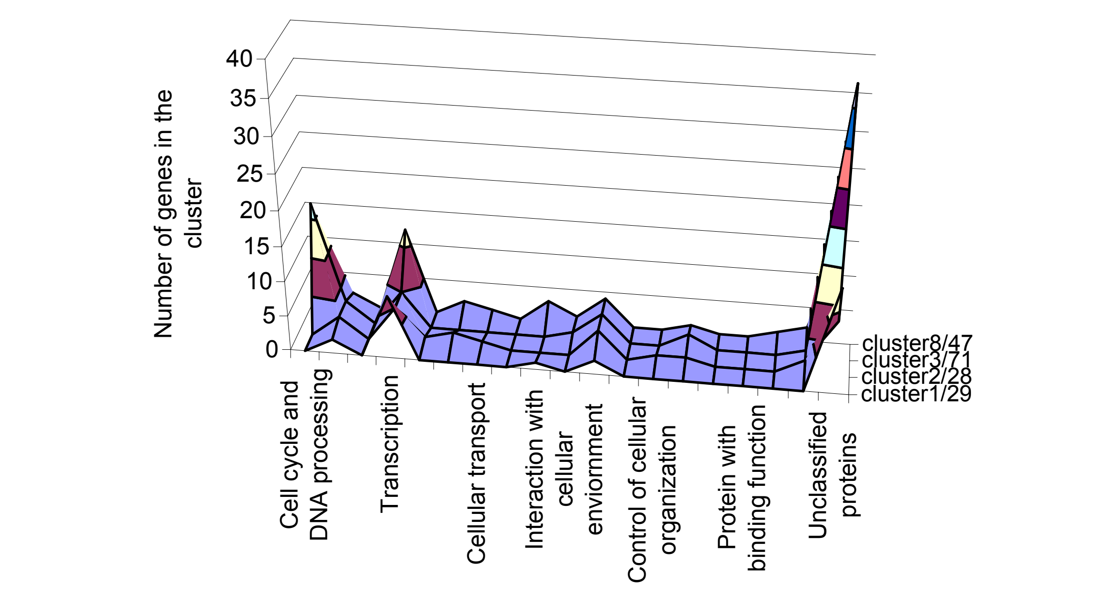


Fig. 2.4a. Functional distribution of selected clusters from cell cycle data set, showing Cluster 3 and Cluster 8 having significant number of genes from the functional categories 'Cell cycle and DNA processing' and 'Transcription' respectively.

* http://vlado.fmf.uni-lj.si/pub/networks/pajek/

Fig. 2.4a depicts the functional distribution of all genes from selected clusters. Some of the resulting clusters were supplemented with genes of similar functions, although each cluster had a significant number of proteins from the ‘Unclassified protein’ functional category.

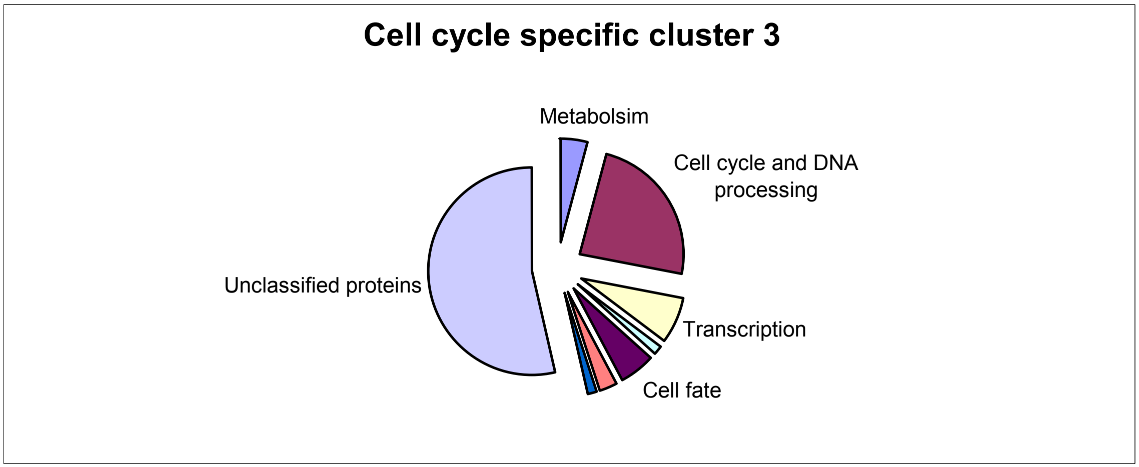


Fig. 2.4b. Functional distribution of genes in cluster 3 from cell cycle dataset**.**

In particular, cluster 3 had 17 genes out of 71 genes from the functional category, ‘Cell cycle and DNA processing’ (P value 3.5. 10^{-4})[*] (Fig. 2.4b) and cluster 8 had 12 genes out of 41 from the functional category ‘Transcription’ (P value (1.9. 10^{-3}))

In the heat shock dataset, clusters were enriched with genes of similar functional category (Fig. 2.5a). In particular cluster 1 had 63 genes out of 73 from the functional category ‘protein synthesis’ (P value 3.9. 10^{-54})(Fig. 2.5b). The clusters from these two datasets with significant P values are summarized in Table 2.7.

[*] P values were calculated using binomial distribution using hypergeometric approximation (Section 3.2.3)

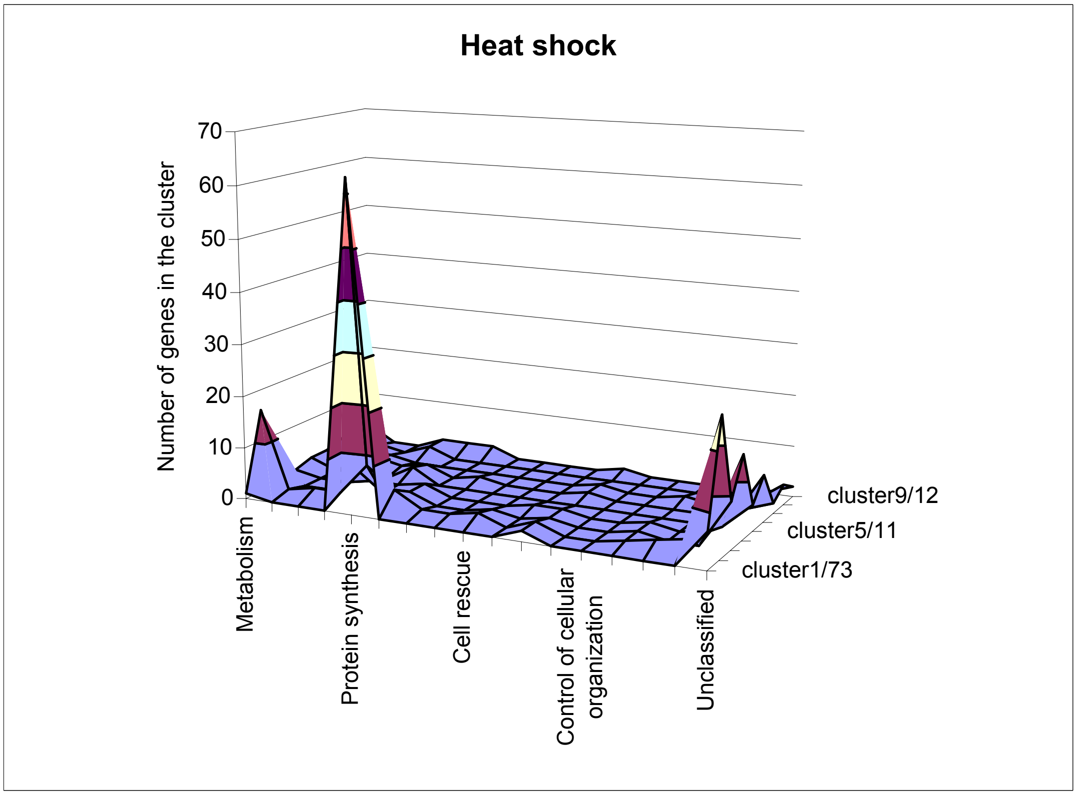


Fig. 2.5a. Functional distribution of selected clusters from the heat shock data set. Showing Cluster 1 having significant number of genes from the functional category 'Protein synthesis'.

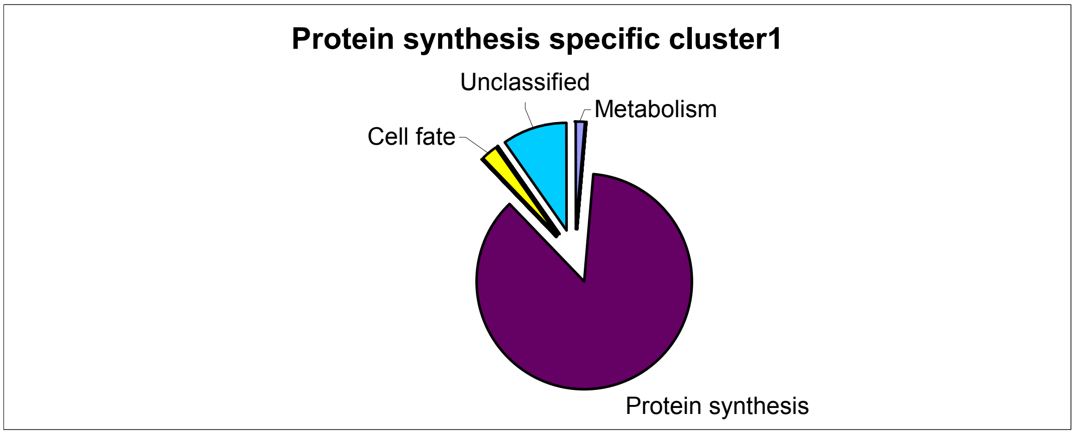


Fig. 2.5b. Functional distribution of genes in cluster 1 from the heat shock data set**.**

Table 2.7. Clusters with significant P values

Dataset	Cluster No.	Number of ORFs (n)	MIPS Category No. of ORFs (M)	MIPS category code	MIPS Category name	ORFs within Category (k)	P- value
Cell cycle	3	71	628	03	Cell cycle and DNA processing	17	$3.5. 10^{-4}$
Cell cycle	8	41	771	04	Transcription	12	$1.9. 10^{-3}$
Heat shock	1	73	359	05	Protein synthesis	63	$3.9. 10^{-54}$

2.1.8 Limitations

Although this method was designed to predict the time-delayed correlations (Section 2.2.10), it has the following limitation.

1) Local vs. global normalization

2) Time taken for analysis

1) Local vs. global normalization

Given the "raw" expression data matrix, it is usually common to perform the normalization step (Section 1.4.2) before proceeding with the correlation coefficient. Usually this normalization procedure is done globally by considering all the time points in case of time series data. But doing this type of global normalization is challenging while considering the locally similar regions. This is because different time points may affect the expression levels at a different scale. For example, consider the expression level corresponding to two time points t_1 and t_2. The first time point t_1 may induce or reduce the gene expression of a set of genes g_1 by a very large fold change for example a 50-fold increase or decrease (>> 1). While another time point may affect mainly the same set genes g_1, but shifts their expression level by a much smaller amount for example a 2-fold increase or decrease. Although the two time points affect the similar set of genes and are thus related, this relation is not explicit while observing the fold change in expression data. So in these cases performing a global normalization by considering both the t_1 and t_2 gives us normalized expression values.

Consider another two time points t_3 and t_4 that affects a different set of genes g_2, but shifts their expression level by a very large factor (more than 10 folds) (>> 1). By considering all 4 time points t_1, t_2, t_3 and t_4 together and performing a global normalization results in normalizing the data matrix according to all 4 time points. But, they are basically affecting different sets of genes g_1 and g_2 locally.

To illustrate the effect of global normalization that affects the locally similar regions, consider the two profiles shown in Fig. 2.7 that are locally similar while measuring the gene expression for time points 1 to 5 (Pearson correlation of 0.68). But they don’t have similar relationship when measuring the gene expression for time points 1 to 15 (Pearson correlation 0.22). After the global normalization, the Pearson correlation corresponding to the locally similar region of time points 1 to 5 reduces to 0.49 illustrated in Fig. 2.8.

Since the data matrix supplied, as input to the dynamic programming is a globally normalized one, looking for locally similar regions in this normalized data set might not reveal the interesting local time shifted correlations. This is one of the major limitations of this method.

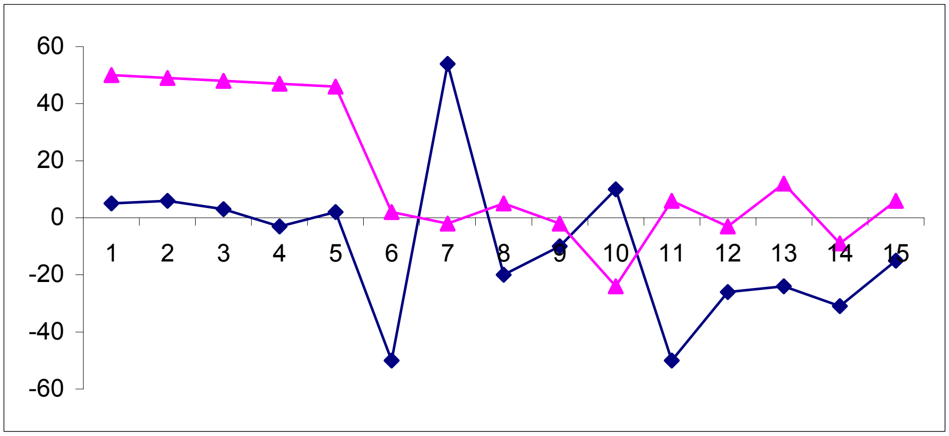


Fig. 2.7. Two example profiles before global normalization. The gene expression values from time point 1 to time point 5 have a Pearson correlation of 0.68.

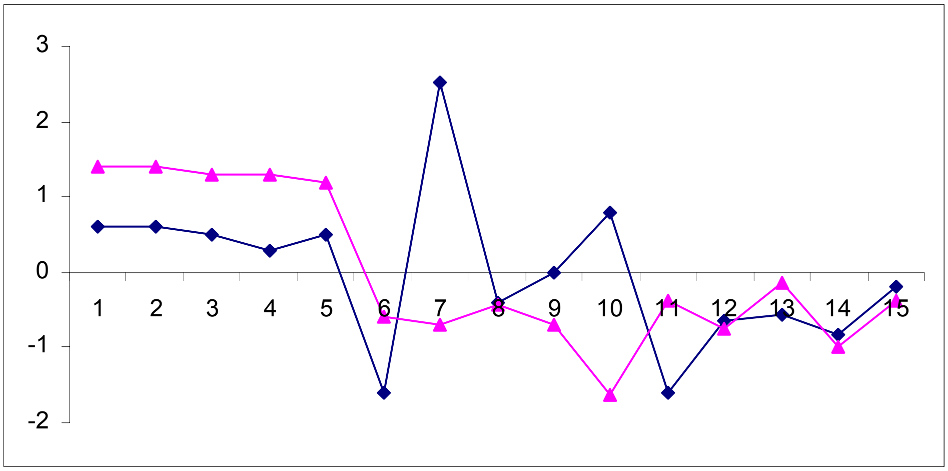


Fig. 2.8. Two example profiles after global normalization. Pearson correlation from time point 1 to time point 5 is 0.49.

2) Time taken for analysis

Since optimal sub vectors are predicted by using dynamic programming optimization, another main disadvantage comes from the time taken for computation using dynamic programming. The time taken for calculation is in the order of O (n^2) both in time and space, which is often too large for practical purposes, especially when it comes to large expression matrices of many genes measured over many time points.

In order to overcome the local similarity limitation of the method using dynamic programming, we extended it to another method called CLARITY that surmounts the problem of local similarity.

2.2 CLARITY

2.2.1 Method

Recall that we are looking for *local* relationships (similarities) between expression profiles and, furthermore, that we seek to incorporate the possibility of *time-shifts*. Thus, we consider two genes respective expression profiles X and Y, represented by sequences (x_1 ...x_n) and (y_1...y_n), as similar if there are similar subsequences X [i, j] and Y [k, l], where X [i, j] $=_{\mathrm{def}}$ (x_i, x_{i+1}...x_j) for $1 \leq i \leq j \leq n$. In analogy to sequence analysis, one can consider a *tuple* (X [i, j], Y [k, l]) as an (local) “alignment” (of length $j - i + 1 = l - k + 1$) (Section 2.2.2).

2.2.2 Exact Similarity Computation

The simplest approach for discovering local, time-shifted relationships between two profiles is to enumerate all possible alignments in a systematic way. Thus, the similarity **SIM** (***X, Y***) between two profiles ***X*** and ***Y*** of length *n* is computed as

$$\mathbf{SIM}\,(\boldsymbol{X},\boldsymbol{Y}) =_{\text{def}} \max_{k_{min} \leq k \leq n} SIM_{\text{k}}\,(X,\,Y), \qquad (2.3)$$

where $SIM_{\text{k}}\,(X,\,Y)$ is given by

$$\max_{1\leq i,\, j\leq n-k+1} \mathrm{S}(X\,[i,\, i+k-1],\, Y\,[j,\, j+k-1]) \qquad (2.4)$$

for an underlying basic similarity measure S. Our particular choice of S will be discussed and motivated below.

As can be seen, $SIM_{\text{k}}\,(X,\,Y)$ corresponds to the similarity of the best alignment of length k. In deriving the similarity between two profiles from very short local alignments is questionable and usually not statistically significant. Therefore, the parameter k_{min} specifies a lower bound to the length of an alignment.

2.2.3 Approximate Similarity Computation

A straightforward implementation of (2.3) leads to a "sliding window" algorithm (i.e., $SIM_{\text{k}}\,(X,\,Y)$ is computed by sliding two windows of size *k* over *X* and *Y*) whose time complexity is $O\,(n^3)$. Note that the complexity is reduced to $O\,(n^2)$ if no time-shifts are allowed (and, hence, i = j in (2.4)). Both cases are completely acceptable for small *n*. For longer expression profiles, however, the exact computation of **SIM** (***X, Y***) might become too expensive. In that case, we suggest the use of an approximate algorithm that is inspired by the well-known BLAST (Altschul et al., 1990) method for sequence alignment. The idea of this approach is to find an initial "hit" in the form of a short optimal alignment. Then, in a second step, this alignment is extended in both directions. More precisely, our heuristic approach works as follows:

1. **Hit:** $SIM_k\,(X,\ Y)$ is computed for $k = k_{min}$. Suppose that this similarity degree is obtained for the "best match" $X[a_x, b_x], Y[a_y, b_y]$, i.e.

   $SIM_k\,(X,\ Y) = S\,(X\,[a_x, b_x],\ Y\,[a_y, b_y])$.

   If the "best match" is not unique, the second step is carried out for all other candidates as well.

2. **Extend:** The similarity degrees
   $S\,(X\,[a_x\ \text{-}u,\ b_x + v],\ Y\,[a_y - u,\ b_y + v])$
   are derived for all
   $0 \leq u \leq \mathbf{min}\ \{d,\ a_x - 1,\ a_y - 1\}$
   $0 \leq v \leq \mathbf{min}\ \{d,\ n - b_x,\ n - b_y\}$
   and the best match $(X\,[a_x - u^*,\ b_x + v^*], Y\,[a_y - u^*, b_y + v^*])$ is determined.
   In the case of ties, longer matches are preferred to shorter ones.
   If there are still several optimal matches, one is chosen at random.

3. **Iterate:** The optimal local alignment is updated by setting
   $a_x \longleftarrow a_x - u^*,\ b_x \longleftarrow b_x + v^*,\ a_y \longleftarrow a_y - u^*,\ b_y \longleftarrow b_y + v^*,$
   and the second step is repeated. This process is iterated until the optimal alignment does not change ($u^* = v^* = 0$).

   The parameter d in the second step is a pre-specified constant that determines the size of the neighborhood to be searched and, hence, the complexity of this step, which is obviously $O\,(d^2)$. Note that the "myopic" strategy obtained for d =1 carries a high risk of getting caught in local maxima. On the other hand, experience has shown that large values for d are usually not necessary for this type of "look-ahead search". Most often, sufficiently good approximations or even exact results are already obtained for d =2.

2.2.4 Basic Similarity Measure

So far, the algorithm outlined above is completely independent of the basic similarity measure S. As noted before, measures commonly used in gene expression analysis include the Euclidean distance and the Pearson correlation. Such numerical measures are easy to compute but suffer from some disadvantages. Particularly, they are quite sensitive toward outliers and measurement errors, a point of critical importance in connection with gene expression data. Moreover, in the context of expression analysis, we prefer a concept of similarity that is based on the qualitative behavior or, say, the shape of the profiles to one that is very sensitive to the precise values of fold changes.

Our similarity measure S is therefore defined by the *Spearman rank correlation* (SRC). The SRC between two profiles *X* and Y is given by

$$\mathrm{SRC}\ (X, Y) = 1 - \frac{6}{n\ (n^2-1)} \sum_{i=1}^{n} (\mathrm{r}_X\ (x_i) - \mathrm{r}_Y\ (y_i))^{\ 2} \qquad (2.5)$$

where $\mathrm{r}_X\ (x_i)$ is the rank of x_i in the profile $(x_1\ ...\ n_n)$: $\mathrm{r}_X\ (x_i) = k \Leftrightarrow |\{ j \mid x_j < x_i\}| = k$ - 1. Actually, we used an extended version of the SRC **(**Press et al., 2002) which takes the possibility of ties, i.e. $x_i = x_j$ for $i \neq j$, into account. The SRC satisfies $-1 \leq \mathrm{SRC}\ (X, Y) = 1$ for all *X, Y*.

To exemplify the aforementioned difference between the Pearson correlation and SRC, Fig. 2.9, shows two profiles (of length 7), which are highly correlated according to the latter but almost uncorrelated according to the former. This is mainly caused by the comparatively large value of the third fold change in one of the sequences.

As opposed to this, the SRC correctly reflects the fact that both profiles have a rather similar shape. In fact, even though SRC ignores some information, it seems that it retains only the relevant information, making it much more robust than the Pearson correlation.

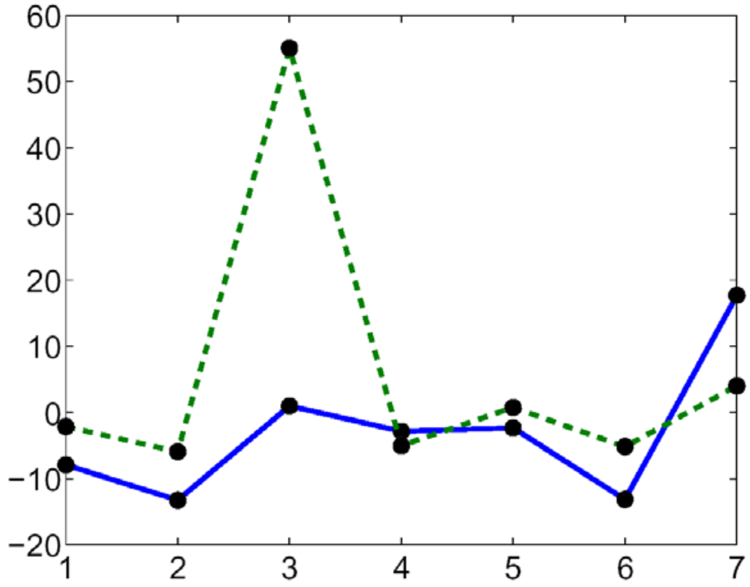


Fig. 2.9. The SRC for the two profiles is 0.93 whereas the Pearson correlation is 0.3

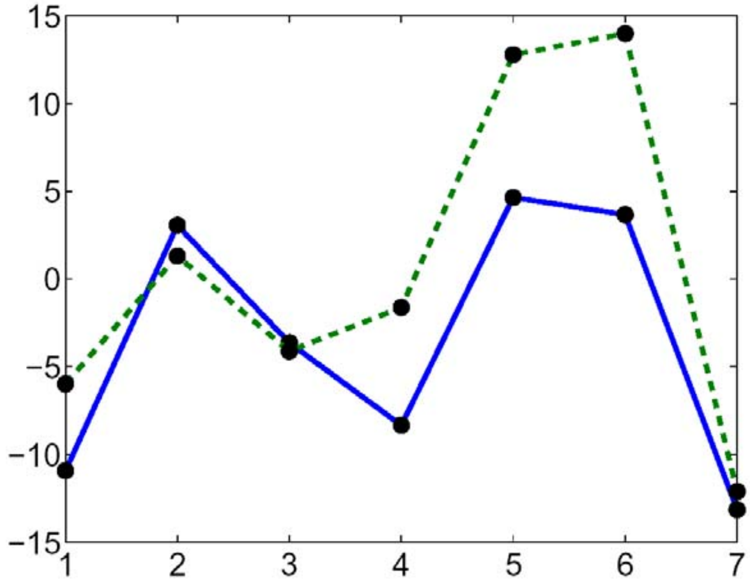


Fig. 2.10. The SRC for the two profiles is 0.93 whereas the Pearson correlation is 0.3

In this connection, it should also be noted that SRC retains more information than a frequently used qualitative measure that compares the sign of the first-order differences (i.e. the “ups” and “downs”):

$$\text{SRC}\ (X, Y) = 1\ -\ \frac{1}{n}\ \sum_{i=1}^{n-1} \mathbf{sgn}\ ((x_{i+1} - x_i) - (y_{i+1} - y_i)), \qquad (2.6)$$

Where sgn (z) is the sign of z. For example, this measure suggests a similarity of only 0.3 for the two profiles in Fig. 2.10, even though both profiles do again have a rather similar shape. Fig. 2.11 shows an example obtained from the mitotic cell cycle time course experiment (Cho et al., 1998 see below) of two expression profiles where SRC yields a similarity of 0.8, whereas Pearson correlation gives only 0.4.

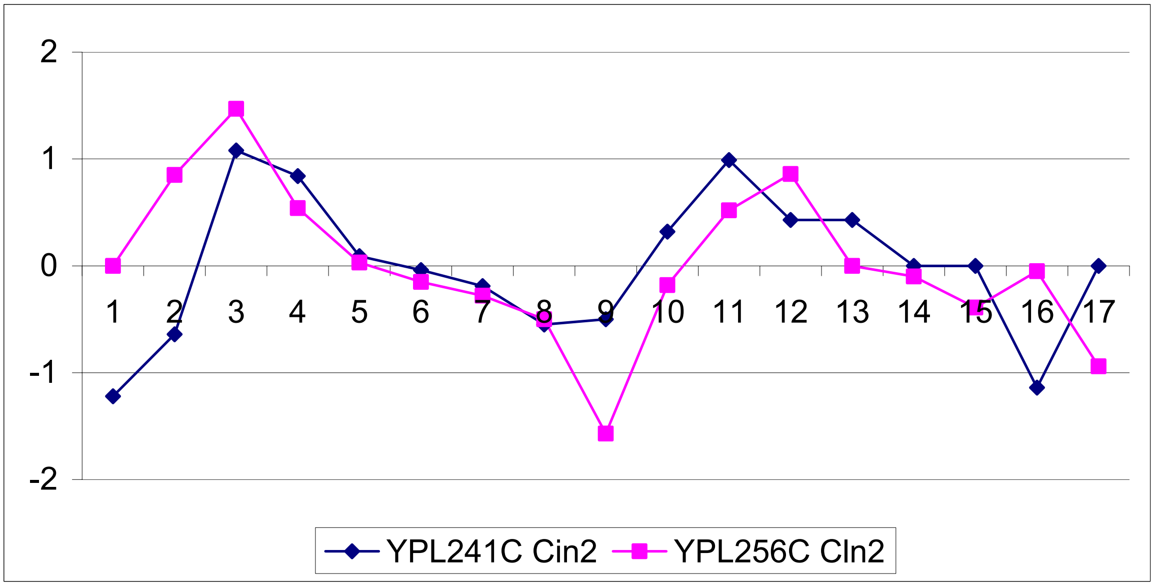


Fig. 2.11. The SRC for this pair of genes is 0.8 over the first 15 time points, whereas the global Pearson correlation is only 0.4. For each of the time points, the expression ratio is plotted. The genes Cin2 and Cln2 were detected as co-induced in a Cdc28-13 mutant during late G1 phase of the cell cycle (Cho et al., 1998; Jiang et al., 2004).

The overall similarity of two profiles *X* and *Y*, as defined by (2.3), is the maximum of similarity degrees for sequences of different lengths *k*. In order to guarantee the comparability of the similarities SIM_k (*X, Y*), $k_{\min} \leq k \leq n$, these similarities have been defined by their corresponding P -value rather than by the SRC directly. Thus, if S* denotes the optimal SRC that has been found for sequences of length *k*, then SIM_k is given by the probability to obtain a correlation of at most **S*** under the null hypothesis of completely unrelated profiles (of length *n*). Note that the statistical distribution of this measure is an extreme value distribution that depends on the parameters *k* and *n*. As there is apparently no simple analytical expression for this distribution, we derived approximations from simulated data. Fig. 2.12 shows the result of such a simulation for profiles with 17 time points.

In order to decide whether or not two profiles *X, Y* are "significantly similar", we also need the P -value of the overall similarity SIM (*X, Y*). Again, we derived approximations of these P -values from simulated distributions.

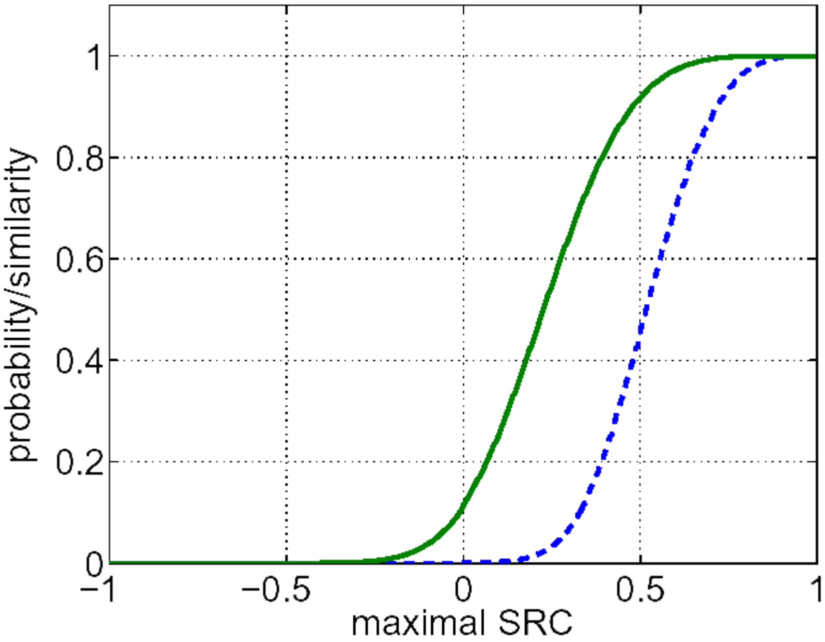


Fig. 2.12. The empirical distribution functions of the (maximal) SRC obtained from simulations for n = 17 and k = 15 (solid line in green)

2.2.5 Data and clustering

In order to test CLARITY on a real dataset, a dataset from a mitotic cell cycle time course experiment in the yeast *Saccharomyces cerevisiae* that included 6331 open reading frames and has been measured over 17 time points by Cho et al., (1998) was used. The yeast cell cultures were synchronized using so-called Cdc28 gene arrest and sampled at uniform intervals covering nearly two complete cycles of cell cycle. The experiment were done using Affymetrix oligonucleotide array. The data is scaled to account for the experimental differences between arrays used. As a first step some data points that appeared to be aberrant were eliminated. The dataset was then converted to ratio style measurements by dividing each measurement by the average value of the measurements for that gene as described in Spellman et al., (1998). 6145 genes were taken for further analysis.

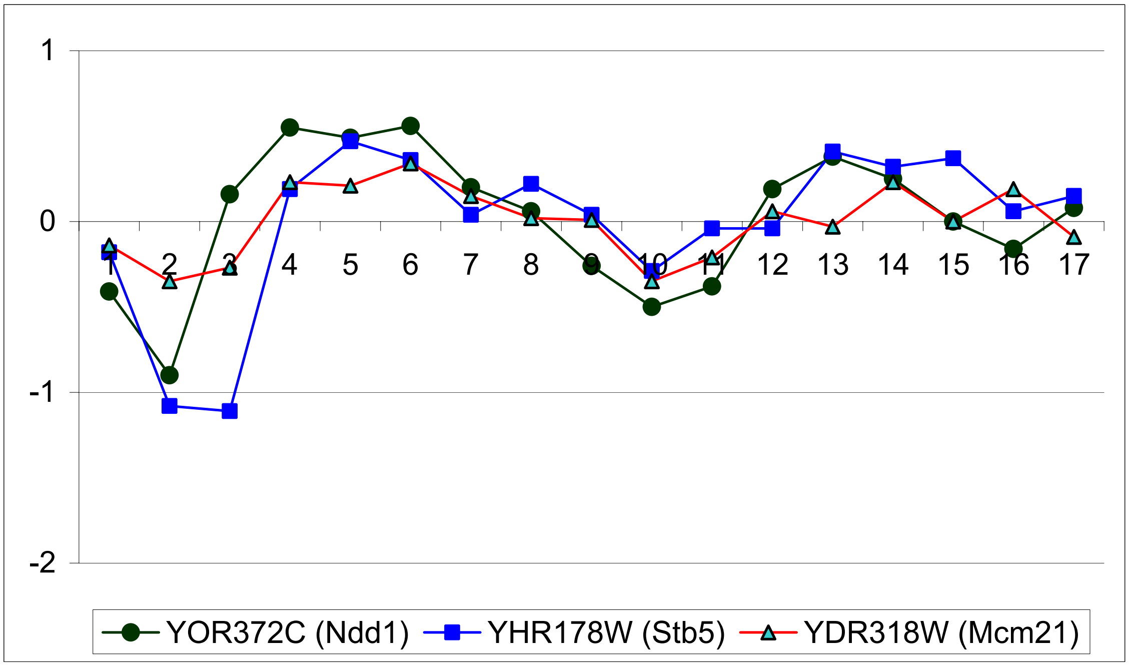


Fig. 2.13. An example for a time-shifted relationship that was also identified by Yu et al., (2003). For each of the time points, the expression ratio is plotted. The profiles are shifted one time point, starting from time point 2. These genes are found in the same cluster 19 if one uses $k_{min} = 15$. If one sets $k_{min} = 17$, they are found in different clusters.

We first applied CLARITY with $k_{min} = 15$ in order to calculate the pairwise similarity matrix between the expression profiles of the individual genes. i.e., time-shifts of maximally two time-points is allowed as longer shifts are hard to explain from a biological point of view. Additionally, it is difficult to obtain significant degrees of similarity if much shorter sub-profiles are used as can be seen from the simulations described above.

Fig. 2.13 shows an example of a time-shifted relationship among genes that was described by Yu et al., (2003) and that was also successfully identified by our approach.

Clusters of genes were derived from the similarity matrix thus obtained using CLUTO, a program package for graph based clustering (Karypis, 2002). CLUTO first constructs a graph where each gene is represented by a node and edges between nodes are labeled with corresponding similarity degrees. Dense regions in this graph correspond to sets of genes that are highly co-expressed and thus form good candidates for clusters.

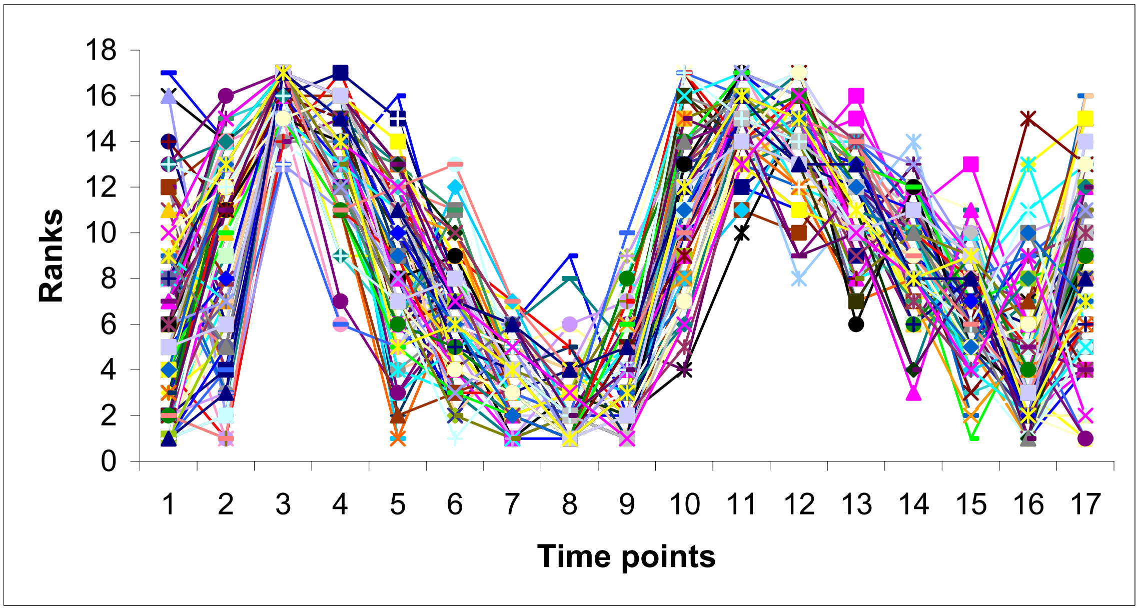


Fig. 2.14. The profiles of a generated cluster that are highly related to the cell cycle. For each of the time points, the ranks of the expression value are plotted to clarify why the method assigned the genes to the same cluster.

For computational efficiency and in order to avoid bias of the results due to insignificant relationships between genes, we simplified the graph in a preprocessing step: The edge between two nodes is deleted whenever the corresponding similarity degree falls below a similarity threshold. In order to derive a clustering structure, CLUTO partitions the graph obtained by means of optimal (minimal) cuts. This is repeated in a recursive manner until a pre-specified number of clusters have been constructed. As can be seen, two critical parameters have to be specified for the clustering approach, namely the number of clusters and the similarity threshold. Fortunately, we found that in my case the clustering results are quite robust toward variations of the above parameters. More specifically, computations with various similarity thresholds showed that thresholds above 0.7 yield almost identical clustering structures. Likewise, we found that the clustering structure did not change appreciably if the number of clusters was raised above 25. We therefore decided to use this number to obtain a maximal number of "independent" sets of genes. Additionally, the generated clusters appear to be quite homogeneous and show a high internal similarity. See Fig. 2.14 for an example.

2.2.6 Functional evaluation

In several cases it has been shown that genes with similar function can be co-expressed (Eisen et al., 1998). To elucidate the biological significance of the clusters generated by our procedure, the clusters produced from CLUTO were mapped to the 400 different MIPS functional categories (section 2.1.6), and one or several predominant categories were assigned to each cluster. Moreover, in order to prove that the occurrence of a predominant category is statistically significant, we derived corresponding P-values for each cluster. The probability of observing at least *m* ORFs from a functional category within a cluster of size *n* is given by

$$P = 1 - \sum_{i=0}^{m-1} \frac{\binom{f}{i}\binom{N-f}{n-i}}{\binom{N}{n}} \qquad (2.7)$$

where f is the total number of genes within a functional category and N is the total number of genes within the genome (6331). Since this P- value obviously ignores the problem of multiple-hypothesis testing, it should be therefore interpreted with caution. For reasons of numerical stability, we computed the above value using the binomial distribution as an approximation of the hypergeometric distribution.

2.2.7 Results

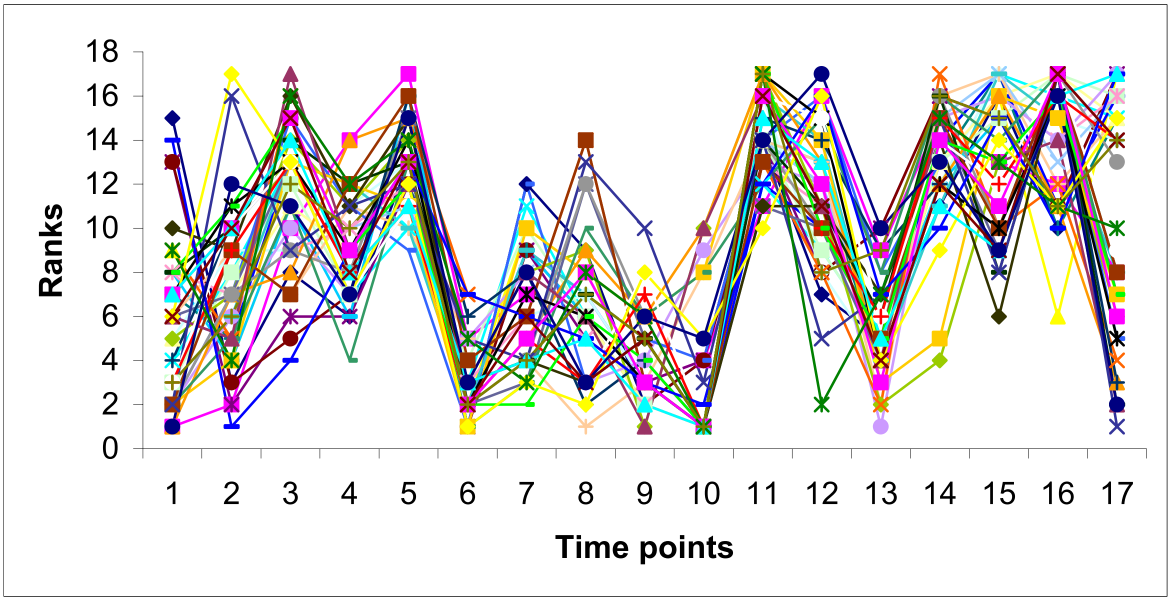
Clusters with P-values less than 10^{-4} and average similarity among cluster members smaller than 0.5 were not reported in this thesis. We found several clusters to be significantly enriched with genes of similar function. The results were summarized in Table 2.8.

In general, the clusters can be divided into those that do not follow the periodicity of the cell cycle (clusters 0, 5, 6, 15, 16 and 21 see Fig. 2.15 (a), (b), (c), (d), (e)) and those that are cell cycle related (clusters 3, 4, 8; See Fig. 2.16 (a), (b) (c)).

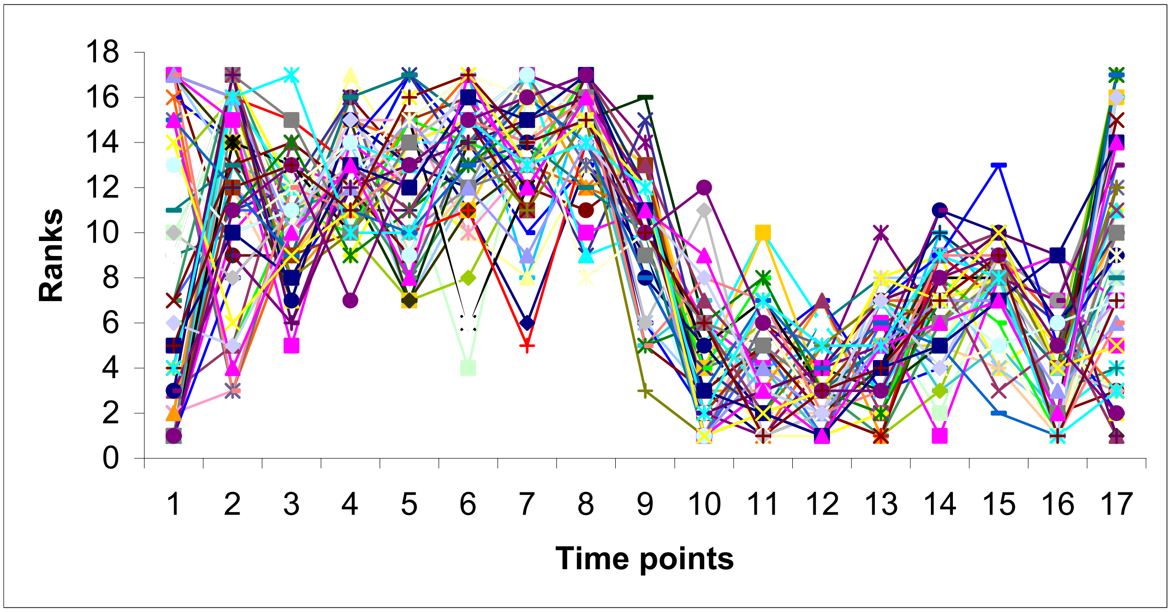
2.2.8 Non-periodic clusters

Among the clusters with non-periodically expressed genes, the most significant functional grouping expressed genes, the most significant grouping occurs in cluster 0. This cluster consists of 43 genes, 33 of them being associated with ribosome biogenesis (P- value 5.0. 10^{-34}).

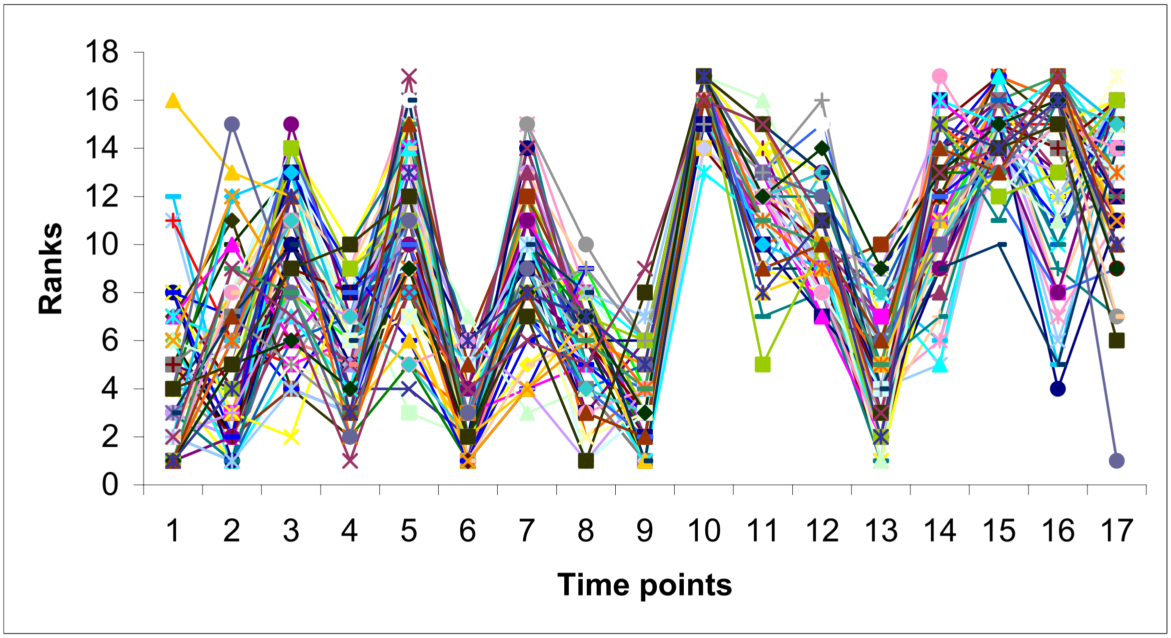
Cluster 16 also contains significant number of protein synthesis related genes (46 out of 168, P value 10^{-18}) including 29 ribosomal genes (P-value 1.2. 10^{-20}). In addition, this cluster contains 14 genes related to amino acid metabolism (P value 1.5. 10^{-4}), indicating that genes within this cluster might play a role in protein synthesis and related functions.



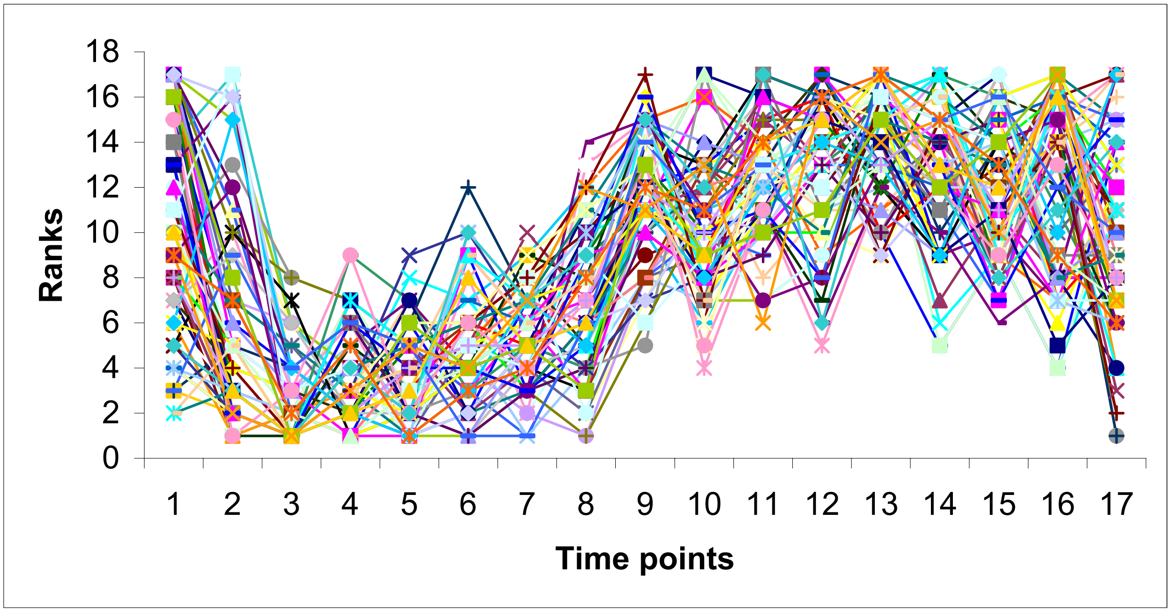
(a) Cluster 0 contains 43 profiles, 33 of them associated to ribosome biogenesis



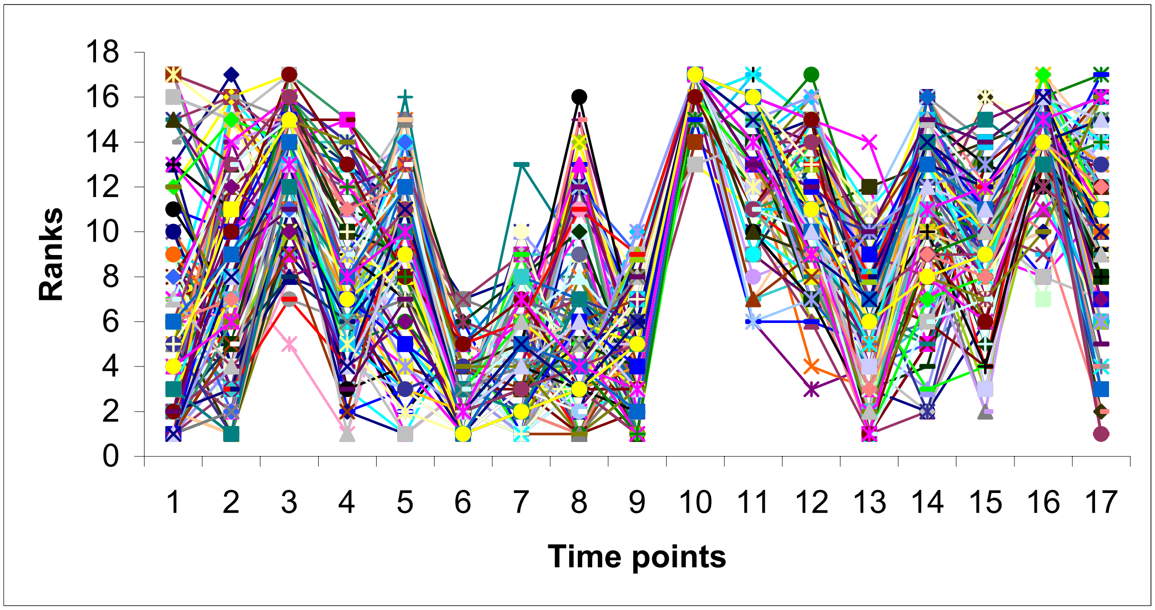
(b) Cluster 5 contains 61 profiles that are mainly related to transcription, rRNA synthesis and tRNA synthesis.



(c) Cluster 6 contains 86 profiles that are mainly related to protein fate



(d) Cluster 15 contains 77 profiles that are mainly related to amino acid metabolism



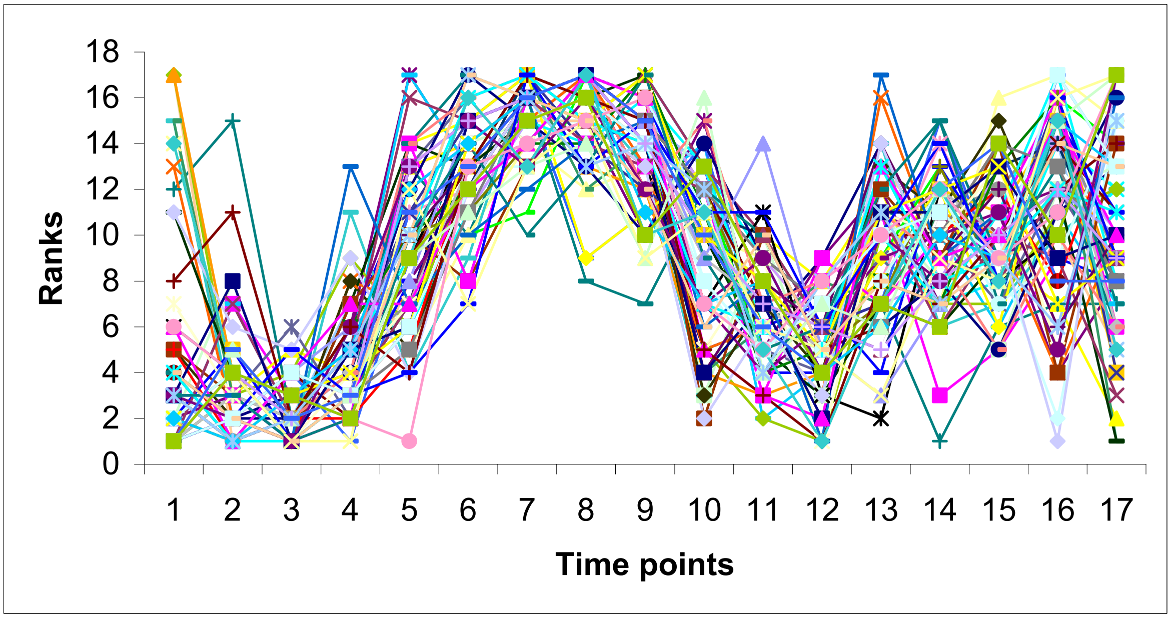
(e) Cluster 16 contains 168 profiles that are mainly related to protein synthesis

Fig. 2.15. The expression profiles of non cell cycle related clusters

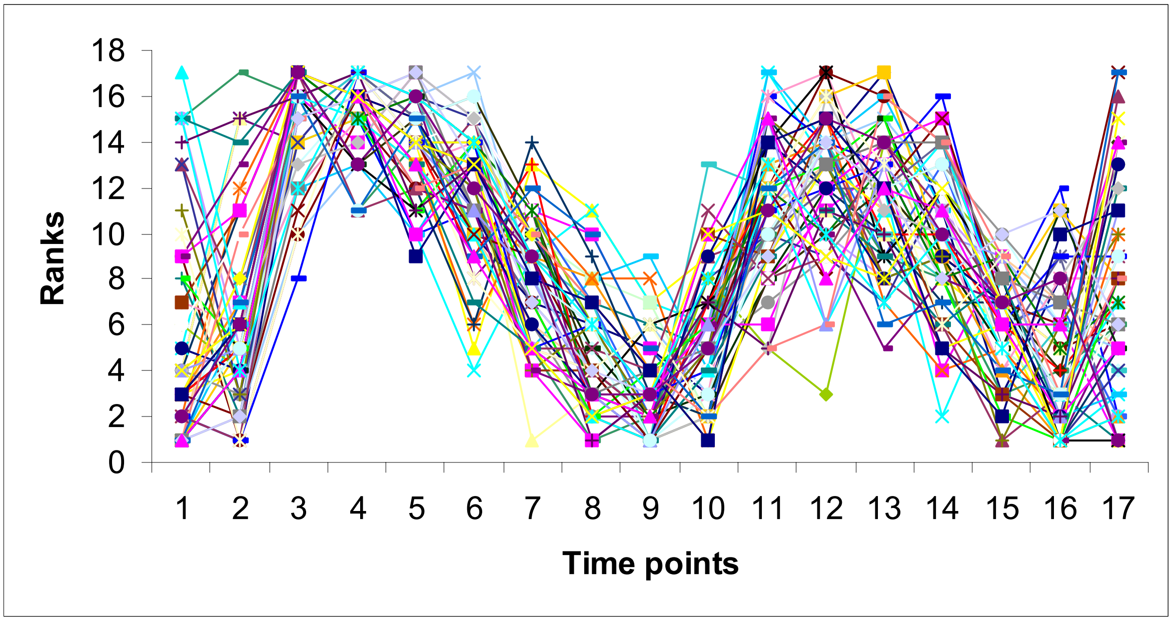
Cluster 21 has a significant enrichment of genes that can be related to energy (P-value 4.5. 10^{-12}) in the broader sense, including genes related to mitochondrial protein synthesis (4.9. 10^{-11}), mitochondrial organization (P- value 5.8. 10 $^{-15}$), and energy and carbohydrate metabolism (P value 2.5. 10^{-7}). Six among the seven genes for the nuclear encoded proteins of the cytochrome C oxidase protein complex 1V (Cox4. Cox5a, Cox7, Cox8, Cox12 and Cox13) are present within this cluster, and similarly several genes encoding proteins of the mitochondrial protein synthesis turnover complex (Mrpl10, Mrpl17, Mrpl24, Mrpl28, Mrpl3, Ydr116C and Ypr100w). These findings support the functional relationships of these genes; the proteins encoded should be co-expressed in stoichiometric amounts required for the assembly of the respective protein complexes.

2.2.9 Periodic clusters

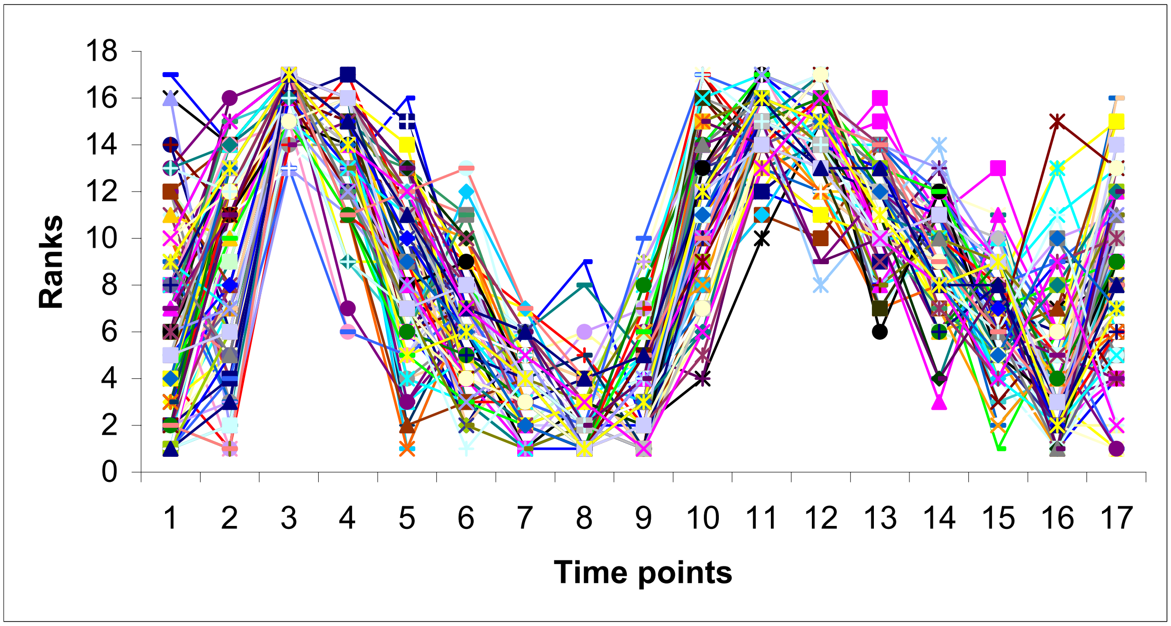
As is expected, among the clusters with a periodic profile many of the genes encode proteins with functions in cell cycle dependent processes, like DNA- processing, DNA synthesis and DNA- replication. These periodic clusters are defined by the timing of the maximum expression of the genes within the cluster. For example, cluster 8 can be considered as G1 specific as it harbors 95 out of 300 reported genes that peak during the G1 phase of cell cycle, while cluster 3 includes 33 out of 197 genes regulated in the M phase (Spellman et al., 1998).



(a) Cluster 3 contains 74 profiles that are mainly related to cell cycle



(b) Cluster 4 contains 60 profiles that are mainly related to DNA processing, cell cycle and mitotic cell cycle control.



(c) Cluster 8 contains 113 profiles that are mainly related to cell cycle, mitosis and DNA processing.

Fig. 2.16. The expression profiles of cell cycle related clusters

2.2.10 Time delayed correlations

One of the advantages of the CLARITY algorithm is that time-shifted relations can be discovered. The time-shifted relations constitute up to 55.4% of the total number of relationships within individual clusters (Table 2.9).

To elucidate whether the implementation of time-shifted relations can aid the discovery of biological implications, we analyzed cluster 15, comprising the highest portion of time-shifted correlations, in more detail. We compared this cluster with the clusters from Tavazoie et al., (1999) which have been generated using Euclidean distance and K-means clustering. From 77 genes in cluster 15, 38 were found in clusters 4, 5, and 8 of Tavazoie et al., the remaining 39 genes have not been assigned to any cluster. Among such genes is Tps3, encoding the regulatory component of the trehalose-6-phosphate synthase/phosphatase complex consisting of Tps1p, Tps2p and Tps3p. Although Tps1 encoding the trehalose-6-phosphatese synthase is present in cluster 8 of

Tavazoie et al., the time shifted relations with Tps3 has not been detected by these authors. (See Fig. 2.17).

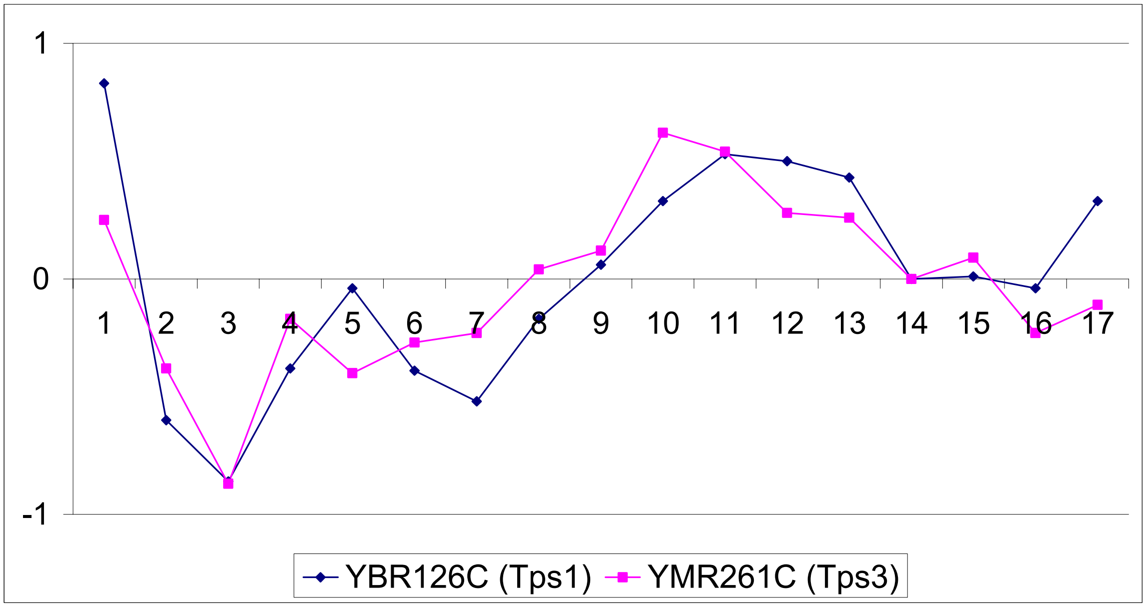


Fig. 2.17. Example of time shifted correlation detected by CLARITY between Tps1 and Tps3.

Table 2.9. Analysis of the relations among *n* entries of the respective clusters. For each of the $\frac{1}{2}$. n . (n -1) possible relationships, the number of time shifted relationships found by CLARITY is shown. Some of the clusters are constituted heavily by time-shifted relationships. Disregarding such relations would thus likely lead to a different cluster structure.

Cluster no	No of genes	No of relationships	No of time-shifted relationships	Percentage
0	43	903	20	2.2
3	74	2701	822	30.4
4	60	1770	368	20.8
5	61	1830	407	22.2
6	86	3655	2	0.08
7	113	6328	711	11.2
15	77	2926	1622	55.4
16	168	14028	653	4.7
19	130	8385	3242	38.7
21	201	20100	7062	35.0

2.2.11 Local correlations

As an example for local correlations, Put1 (Proline oxidase) and Put2 (Delta-1-pyrroline-5-carboxylate dehydrogenase) are found to be in cluster 15, but were found to be in different clusters by Tavazoie et al., (1999). Put2p in conjunction with Put1p converts praline to glutamate in the mitochondrion. In addition, cluster 15 harbors several other genes involved in glutamate metabolism that show local similarities with Put1 and Put2: Put4, a high affinity praline permease, Agp1, the principal transported of asparagines and glutamine, Dip5, an amino acid permease for the transport of alanine, glycine, serine, asparagines and glutamine, and the Glutamate decarboxylase Gad1 (see Fig. 2.18).

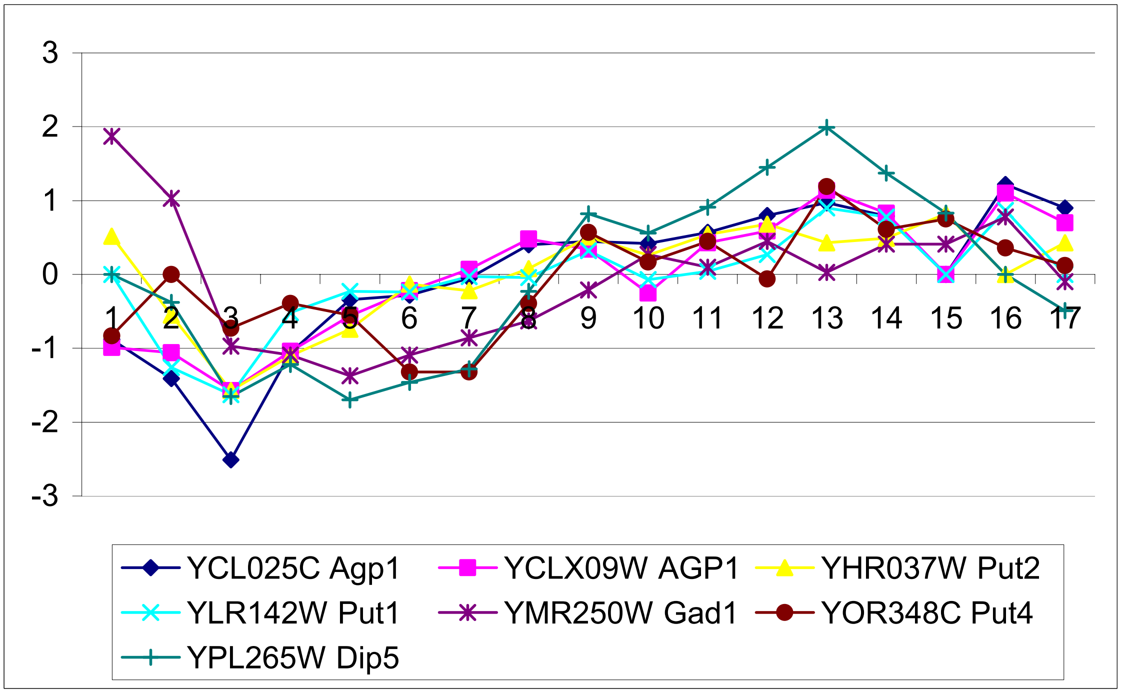


Fig. 2.18. Expression profiles of genes Put1 and Put2, together with other genes involved in glutamate metabolism.

Table 2.8. Results of the clustering analysis. For each of the generated cluster, we show some MIPS functional categories that were found to be significantly enriched.

Cluster No.	Internal Similarity	Number of ORFs (*n*)	MIPS Category No. of ORFs (*M*)	MIPS category code	MIPS Category name	ORFs within Category (*k*)	*P*- value
0	0.737	43	380	05	Protein synthesis	33	$7.5.\ 10^{-27}$
			234	05.05	Ribosome biogenesis	33	$5.0.\ 10^{-34}$
3	0.766	74	492	03.03	Cell cycle	16	$9.5.\ 10^{-5}$
4	0.740	60	271	03.01	DNA processing	10	$6.9.\ 10^{-5}$
			157	03.01.05	DNA recombination and repair	8	$2.1.\ 10^{-5}$
			492	03.03	Cell cycle	19	$9.3.\ 10^{-8}$
			354	03.03.01	Mitotic cell cycle control	11	$1.8.\ 10^{-4}$
5	0.755	61	832	04	Transcription	24	$1.0.\ 10^{-6}$
			75	04.01.01	rRNA synthesis	10	$1.9.\ 10^{-10}$
			31	04.03.01	tRNA synthesis	3	$2.2.\ 10^{-4}$
6	0.803	86	624	06	Protein fate	19	$4.7.\ 10^{-4}$
8	0.824	113	271	03.01	DNA processing	35	$1.7.\ 10^{-20}$
			103	03.01.03	DNA synthesis and replication	17	$4.3.\ 10^{-20}$
			157	03.01.05	DNA recombination and repair	17	$6.0.\ 10^{-10}$
			492	03.03	Cell cycle	29	$1.0.\ 10^{-8}$
			354	03.03.01	Mitotic cell cycle control	19	$8.3.\ 10^{-6}$
15	0.732	77	186	01.01	Amino acid metabolism	9	$1.0.\ 10^{-4}$
16	0.722	168	186	01.01	Amino acid metabolism	14	$1.5.\ 10^{-4}$
			380	05	Protein synthesis	46	$4.7.\ 10^{-18}$
			234	05.01	Ribosome biogenesis	39	$1.2.\ 10^{-20}$
19	0.622	130	492	03.03	Cell cycle	33	$1.5.\ 10^{-9}$
			354	03.03.01	Mitotic cell cycle control	18	$1.7.\ 10^{-4}$
			430	04.05.01	mRNA synthesis	19	$7.3.\ 10^{-4}$
			343	04.05.01.04	Transcription control	16	$8.9.\ 10^{-4}$
			186	01.01	Amino acid metabolism	15	$1.9.\ 10^{-6}$
			121	01.01.01	Amino acid biosynthesis	10	$4.4.\ 10^{-5}$
21	0.800	201	260	02	Energy	33	$4.5.\ 10^{-12}$
			93	02.13	Energy: Respiration	14	$2.5.\ 10^{-7}$
			37	02.01	Energy: Glycolysis	6	$1.4.\ 10^{-4}$
			25	02.10	TCA cycle	6	$9.4.\ 10^{-6}$
			378	01.05	C- compound and carbohydrate metabolism	32	$2.5.\ 10^{-7}$
			380	05	Protein synthesis	29	$6.4.\ 10^{-6}$
			234	05.01	Ribosome biogenesis	20	$2.3.\ 10^{-5}$
			52	05.01.01	Mitochondrial ribosomal proteins	14	$4.9.\ 10^{-11}$
			100	06.13.01	Cytoplasmic and nuclear degradation	11	$8.5.\ 10^{-5}$
			128	30.16	Control of cellular organization: Mitochondria	26	$5.8.\ 10^{-15}$

Chapter 3

Predicting evolutionarily conserved functional modules using genetic algorithms

3.1 Bicluster

In order to address the shortcomings of independent condition dataset stated in section 1.5.3, the concept of ‘biclustering’ was introduced to gene expression analysis. The biclustering concept was first defined by Hartigan, (1975). Since then it has been applied to several domains before Cheng and Church, (2000) introduced the concept in gene expression data analysis. The concept of bicluster corresponds to a subset of genes and a subset of conditions with a high similarity score. Given a gene expression matrix, a biclustering algorithm searches for sub matrices, (or biclusters) which are tightly co-regulated according to some scoring criterion (Fig. 3.1 (a) (b) and (c)). Similarity is not treated as a function of pairs of genes or pairs of conditions. Instead it is a measure of coherence of genes and conditions in the bicluster. This measure can be a symmetric function of the genes and conditions involved and thus the finding of bicluster is a process that groups genes and conditions simultaneously.

Since each bicluster consists of a set of genes that are expressed similarly under given conditions they might be responsible for inducing certain transcriptional or functional modules. Thus the tightly clustered predicted biclusters can be referred to as condition specific functional modules. For a detailed survey of biclustering refer to Madeira and Oliveira, (2004).

In order to address the drawbacks regarding the independent gene expression data analysis (stated in section 1.5.3) by predicting biclusters or modules (section 2.2.1), we adopted a genetic algorithmic approach. This section introduces genetic algorithms in general as well as in the context of gene expression. It also explains the fitness function and selection method.

E(i, j)	c1	c2	c3	c4	c5	c6	c7	c8	c9	c10
Gene 1	-0.42	-0.08	0.68	0.55	0.33	-0.18	-0.1	0.29	0.25	0.02
Gene 2	0.01	**-0.82**	-0.5	**-1.02**	-0.23	**1.40**	0.05	-2.1	0.31	**0.8**
Gene 3	0.21	0	-0.5	-0.25	0.33	0.12	0.18	0.3	0.13	0.38
Gene 4	0.04	-0.12	-0.3	-0.13	-0.03	-0.04	0.21	0.43	0.26	0.09
Gene 5	0.24	0.02	-0.4	-0.23	-0.05	-0.02	-1.3	0.6	0.49	0.71
Gene 6	0	0.43	-0.6	-0.2	0.78	0.44	0.45	0.74	0.22	-0.42
Gene 7	0.01	-0.01	0.04	-0.1	-0.07	-0.02	0.03	0.25	0.06	0.35
Gene 8	0.87	**-0.91**	0.04	**-1.56**	-0.5	**2.3**	-0.6	3.2	-0.26	**0.9**
Gene 9	0.9	0.25	0	0.16	-0.32	-0.22	-0.4	-0.3	-0.44	0.34
Gene 10	0.15	-0.168	-0.088	-0.011	0.081	-0.038	-0.15	0.254	0.11	0.388

***m1*= {g2, g8} {c2, c4, c6, c10}**

-0.82	**-1.02**	**1.40**	**0.8**
-0.91	**-1.56**	**2.3**	**0.9**

Fig. 3.1a. Representation of a gene expression matrix of 10 genes that has been measured over 10 different independent conditions. A sample bicluster (module) m1 that consists of gene2 and gene8 that change the expression simultaneously under the set of conditions {c2, c4, c6, c10} , representing a bicluster of dimension 2 x 4, is shown in the bottom of the matrix.

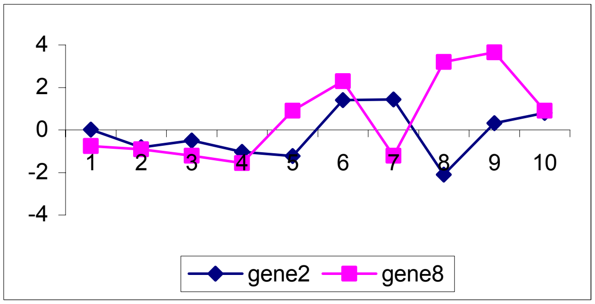


Fig. 3.1b. Graphical representation of expression profiles from gene2 and gene8 including all conditions. Pearson correlation is equal to –0.03.

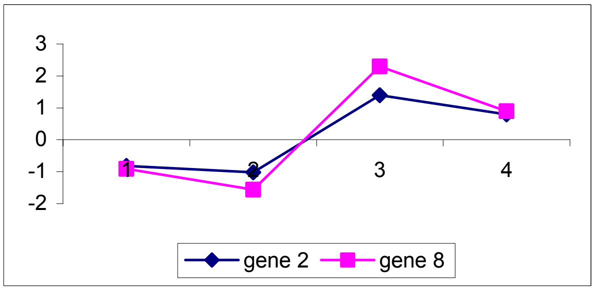


Fig. 3.1c. Graphical representation of the expression profiles from module *m1* including only subset of conditions. Pearson correlation is equal to 0.98.

3.2 Optimization problem

Optimization problems are abundant in everyday life. In computational biology, well-known optimization problems of finding the low energy protein conformations, protein structure comparison and finding the best optimal multiple alignment. In our present case, searching for the best bicluster that has a set of genes co-expressed under a set of conditions representing a functional module is an optimization problem.

The metaphor of a mountain landscape is often used to represent the optimization problems. Here, the landscape corresponds to the fitness function. The landscape might have many peaks representing possible solutions of the problem, which makes it difficult to determine the highest peak representing the best solution to the problem. Many optimization problems are so complex that a mathematical optimization of all possible solutions is not feasible. Especially when local optima are present it is very hard to find the global optimum. In such a case, optimization strategies must be used. The simplest strategy is random search like Monte Carlo search, in which, one randomly samples from the space of potential solutions and selects the one that appeared to be the best. The nature of random searching strategies for near-optimal solutions involves a large degree of potentially wasteful computation through sampling unfavorable regions of parameter space. A general, non problem-specific optimization method ideally should combine robust exploration of the parameter space and efficient exploitation of the information provided from sampling.

An alternative is heuristic search, in which rules of thumb are used to guess a solution that is at least acceptable. Heuristic strategies start from a trial solution and go on to a next solution through small modifications, perhaps after a small assessment of the best direction. Methods like gradient descent and simulated annealing belong to this category. Simulated annealing is able to escape from local optima by accepting moves to a worse state with a small probability, whereas the others are likely to find the peak nearest to the starting point, instead of the highest.

3.3 Evolutionary optimization

Another class of optimization methods that can be applied to complex optimization problems is formed by evolutionary algorithms such as Genetic Algorithms (GAs) (Goldberg, 1989; Davis, 1991; Holland, 1992). The Genetic algorithms were formally introduced in the United States in the 1970s by John Holland at the University of Michigan basically adopts the idea from Charles Darwin's theory of evolution-often paraphrased as "survival of the fittest".

The principle of natural selection from Darwin states that, given the available resources, individuals better adapted to their environment can possibly survive, and will, on average leave behind more offspring than those members who cannot adapt to their environment. This implies that unfit members will die from attrition before they have a chance to reproduce.

For natural selection to lead to evolution, at least two essential features are required: (1) Recombination or crossover: By making crossover, the offspring retain at least some of the features that made their parents fitter than the average; (2) Mutation: By permitting a mutation at any given time results in a population of individuals of varying fitness, making the natural selection to operate on. The crossover process allows offspring to have a combination of the parent's characteristics. Mutation is a random process that also provides the opportunity to introduce new characteristics unrelated to the parents. In the general scheme of evolution, mutation generally is regarded as secondary crossover. In part, this is because mutation occurs relatively infrequently but, more importantly, it is a less efficient optimizing process because it fails to exploit the information contained in the parent structures which contribute to successful organisms. From an algorithmic point of view, mutation is extremely important since it guarantees the diversity.

The principal characteristics of evolutionary algorithms are that they consider populations of solutions rather than one solution at a time. By a reproduction process that is biased towards better solutions the next population is formed, containing new and hopefully better solutions. If one is interested in more than one solution, it is difficult to prevent individual based approaches from finding the same optimum over and over again; in evolutionary optimization one can force diversity in the population. Another advantage of evolutionary algorithms is that the machinery of the algorithm determines the next population of trial solutions. In individual-based approaches, the user has to define how to proceed from one state to the next. Since genetic algorithms evaluate multiple points in the solution space simultaneously, they have the potential to converge on the global optimum.

3.4 Genetic algorithms

The idea of searching among a collection of candidate solutions for a desired solution is common in computer science. The space that consists of possible solutions to the problem is defined as the "search space". The basic idea in using genetic algorithms as an optimization method is to represent a population of possible solutions, in a chromosome-type encoding, and manipulate these encoded solutions through simulated reproduction, crossover and mutation. Those sample solutions with more characteristics in common with the correct solution would tend to survive during the evolutionary process, where the less successful solutions would die off, in a manner analogous to the survival of the fittest in nature.

3.4.1 The basic genetic algorithm

Genetic algorithms produce an initial population of solutions, and simple manipulations or operators are applied to the population of solutions. The result of applying the operators to a population is to produce a new population of solutions. This process is repeated a number of times until a suitable solution or group of solutions evolve. Fig. 3.2 represents the basic genetic algorithm steps.

In genetic algorithmic terminology, the current population is referred to as parents; the new population as offspring, and every iteration represents a successive generation.

```
Initialise Population

Repeat until Convergence
    Evaluation
    Reproduction
    Crossover and Mutation
    Update Population
```

Fig. 3.2. Basic genetic algorithm

3.4.2 Model representation

A key aspect of genetic algorithms is the representation of complex solution by simple encoding. The encoding adopted by Holland, (1975) is the representation of a solution by binary digits or bit strings. There are many types of encoding other than binary encoding. As stated by both Goldberg, (1989) and Davis, (1991), the best encoding is problem specific and may require some experimentation and modification of the crossover and mutation operators. This study uses the binary encoding.

Our initial population consists of *Q* biclusters or modules. Each bicluster m_q has dimension of g x c, $1 \leq g \leq$ G and $1 \leq c \leq$ C, where $q \in 1 \ldots$ Q, G represents the number of genes taken for analysis and C represents the number of conditions under which the experiments have been conducted (Fig. 3.3a and 3.3b). In order to avoid complications, the dimension of every bicluster was kept constant.

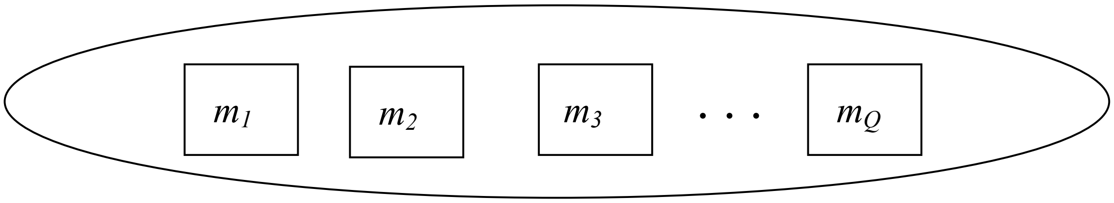


Fig. 3.3a. Initial population of m_N biclusters

$$m_1 = \{g_2, g_3\}\ \{c_2, c_{10}\}, \qquad m_1 = \begin{pmatrix} g_2c_2 & g_2c_{10} \\ g_3c_2 & g_3c_{10} \end{pmatrix}$$

Fig. 3.3b. Representation of a *2 x 2* bicluster m_1 (module).

3.4.3 Model evaluation

The other important aspect of genetic algorithms is to specify the model evaluation criterion, referred to as the fitness. While invoking the genetic algorithm, we are looking for the fittest biclusters in the sense of the survival of the fittest. For many problems of interest, this is reasonably straightforward as we are concerned with observed data, and are interested in determining a set of parameters that provide a good prediction of the observed data. We used the average Pearson correlation as the fitness function in this study. Consider a bicluster of dimension g x c, where g and c represents the number of rows and columns of the bicluster, respectively. The average Pearson correlation R_{avg} of the bicluster is given as

$$R_{avg} = \frac{1}{\mathrm{N}} \sum_{i,j=1}^{\mathrm{N}} r_{(i,j)} \tag{3.1}$$

$r_{(\mathrm{i,\,j})}$ represents the Pearson correlation of any two rows i, j, where $i, j \in (1 \ldots g)$ and N is given by $\mathrm{N} = g*(g\text{-}1)/2$.

3.4.4 Genetic algorithm operators

Having defined a population of Q biclusters and calculated the value of fitness function R_{avg} for each bicluster, an iteration of our genetic algorithm proceeds in three stages, corresponding to the operations as mentioned here.

1. Reproduction using selection

This stage selects an interim population of Q biclusters via rank a based fitness selection assignment. As the aim is to propagate better or fitter biclusters, those biclusters with higher values of the fitness function should have a higher probability of proceeding to the next generation. In the selection process the offspring-producing individuals or parents are chosen. The first step is the fitness assignment. Each individual in the selection pool receives a reproduction probability depending on the own fitness value and the fitness value of all other individuals in the selection pool. This fitness is used for the actual selection step afterwards.

In rank based fitness assignment, the population is sorted according to the fitness values. The probability assigned to each individual depends only on its position in the individual rank and not on the actual fitness value. Rank-based fitness assignment overcomes the 'scaling problems' of the proportional fitness assignment such as the effect of fitness on one or two extreme individuals. The fitness values of these extreme individuals will be neglected irrespective of how much greater or less their fitness is than the rest of the population. The reproductive range is limited, so that no individuals generate an excessive number of offspring. Ranking introduces a uniform scaling across the population and provides a simple and effective way of controlling selective pressure. Rank-based fitness assignment behaves in a more robust manner than proportional fitness assignment and, thus, is the method of choice (Bäck and Hoffmeister, 1991;Whitley, 1989).

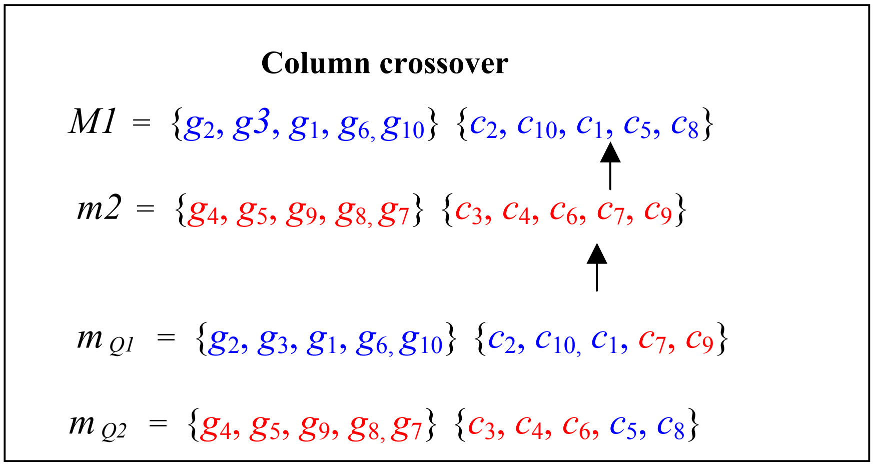
In a population of Q individuals, let Pos represents the position of an individual in this position (least fit individual has Pos = 1, the fittest individual Pos = Q), Let SP be the selective pressure represents the probability of the best individual being selected compared to the average probability of selection of all individuals. The fitness value for an individual can be calculated in two ways using linear ranking or non-linear ranking as given below. In this study, we used linear ranking as the method of selection.

$$Fitness\ (Pos) = 2 - SP + 2*(SP\text{-}1)\ \frac{(Pos - 1)}{(Q\text{-}1)} \qquad (3.2)$$

Linear ranking allows values of selective pressure in [1.0, 2.0]. The probability of each individual being selected for mating depends on its fitness normalized by the total fitness of the population.

2. Crossover

Having selected an interim population of individuals called parent population, we want to produce an offspring population and this is achieved by randomly paring off members of the parent population. Once all parents have been paired off to form *Q/2* pairs, each pair is selected randomly out of the interim population and crossed over progressively. The crossover operation involves the random selection of two positions x_{Nc} and x_{Ng} as a first step for column crossover corresponding to conditions and for row crossover corresponding to rows respectively, where *Ng* x *Nc* represents the dimension of the bicluster. In the second step, from the selected positions parts are exchanged from one parent with the equivalent part in the other parent. For example, consider two-selected bicluster parents m_1 and m_2 from a population, of dimension 5 x 5, and x_{Nc} and x_{Ng} be 3 and 2 respectively, then the column crossover followed by row crossover to produce offspring m_{Q1} and m_{Q1} is illustrated in Fig. 3.4 and Fig. 3.5.



Column crossover

$M1 = \{g_2, g3, g_1, g_6, g_{10}\}\ \{c_2, c_{10}, c_1, c_5, c_8\}$

$m2 = \{g_4, g_5, g_9, g_8, g_7\}\ \{c_3, c_4, c_6, c_7, c_9\}$

$m_{Q1} = \{g_2, g_3, g_1, g_6, g_{10}\}\ \{c_2, c_{10}, c_1, c_7, c_9\}$

$m_{Q2} = \{g_4, g_5, g_9, g_8, g_7\}\ \{c_3, c_4, c_6, c_5, c_8\}$

Fig. 3.4. Representation of Column cross over

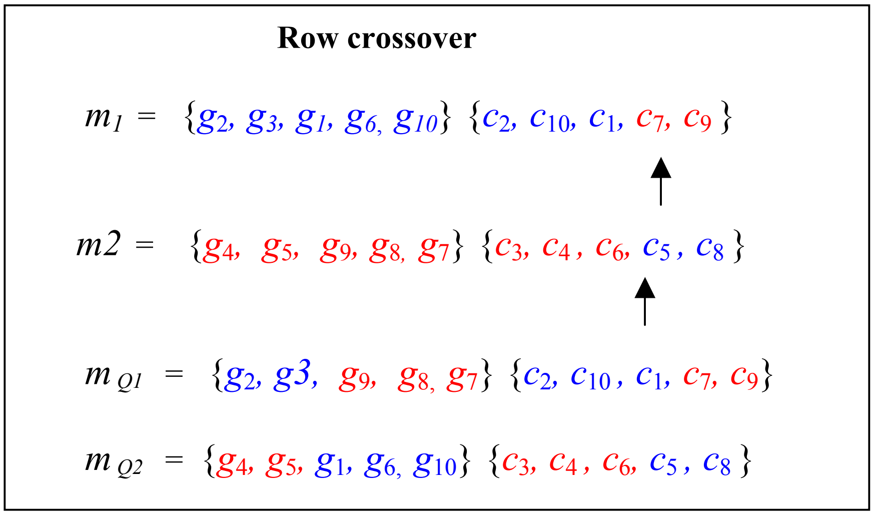

Row crossover
m_1 = {g_2, g_3, g_1, g_6, g_{10}} {c_2, c_{10}, c_1, c_7, c_9 }
$m2$ = {g_4, g_5, g_9, g_8, g_7} {c_3, c_4 , c_6, c_5 , c_8 }
m_{Q1} = {g_2, g_3, g_9, g_8, g_7} {c_2, c_{10} , c_1, c_7, c_9}
m_{Q2} = {g_4, g_5, g_1, g_6, g_{10}} {c_3, c_4 , c_6, c_5 , c_8 }

Fig. 3.5. Representation of Row cross over

3. Mutation

The mutation operator randomly selects a gene $g_m \in 1 \ldots G$ or condition $c_m \in 1 \ldots C$ for any set of gene G and condition C and selects positions x_{Ng} or x_{Nc} from any bicluster m_q and swaps with the selected gene or condition in a bicluster. For example, let $g_m = g_6$ and $x_{Ng} = 5$, after mutation the offspring m_{Q1} from the previous section is illustrated in Fig. 3.6.

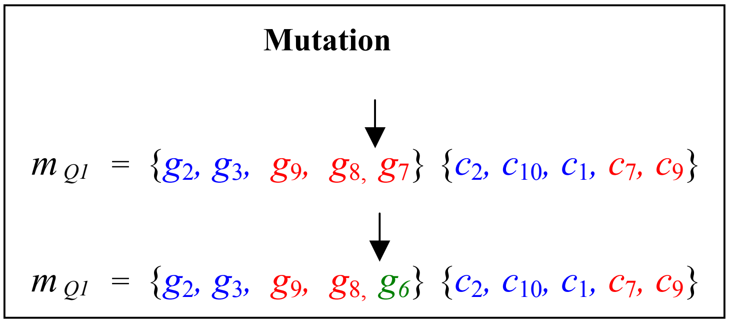

Mutation
m_{Q1} = {g_2, g_3, g_9, g_8, g_7} {c_2, c_{10}, c_1, c_7, c_9}
m_{Q1} = {g_2, g_3, g_9, g_8, g_6} {c_2, c_{10}, c_1, c_7, c_9}

Fig. 3.6. Representation of Mutation

Mutation is important as it introduces diversity in the model population, which reproduction and crossover cannot achieve.

3.5 Testing the module prediction algorithm on a synthetic dataset

In order to test the genetic algorithm based module predictor, we used a synthetic dataset. Synthetic data was created in two steps. In the first step, random expression profiles were generated as follows. We considered the normalized gene expression matrix of size 6152 X 173 from Gasch et al., (2000) as the input to create the random dataset. This dataset measured the change in expression pattern during various environmental conditions in yeast. From this dataset, a small gene expression matrix E of dimension 100 X 50 was created by randomly selecting expression values from the data matrix of Gasch et al. In the second step, expression profiles of 10 highly co-expressed (correlation coefficient 0.99) ribosomal genes during the heat shock condition time course from Gasch et al. were chosen.

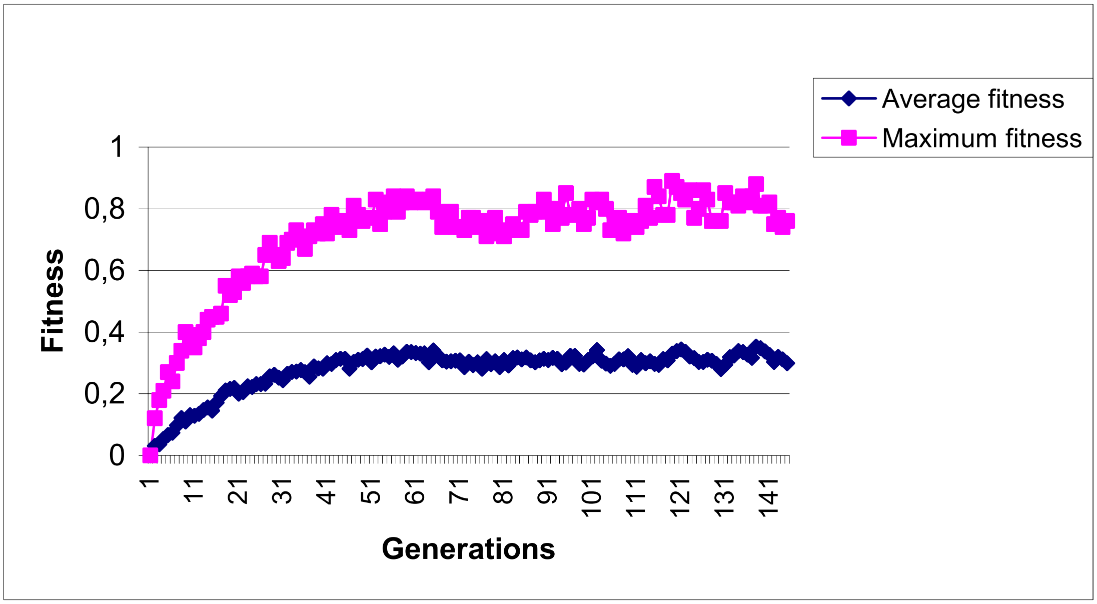


Fig. 3.7. Plot representing fitness vs. generations

These genes were spiked in to the randomly generated gene expression matrix E at positions specified below.

$\{g_1, g_6, g_{14}, g_{22}, g_{32}, g_{41}, g_{48}, g_{54}, g_{62}, g_{92}\}\{c_1, c_5, c_{11}, c_{15}, c_{20}, c_{24}, c_{28}, c_{34}, c_{38}, c_{42}\}$

The initial population was considered as 20 modules and the size of the module was 10 X 10. During evolution, one would expect an increase in the fitness during every generation.

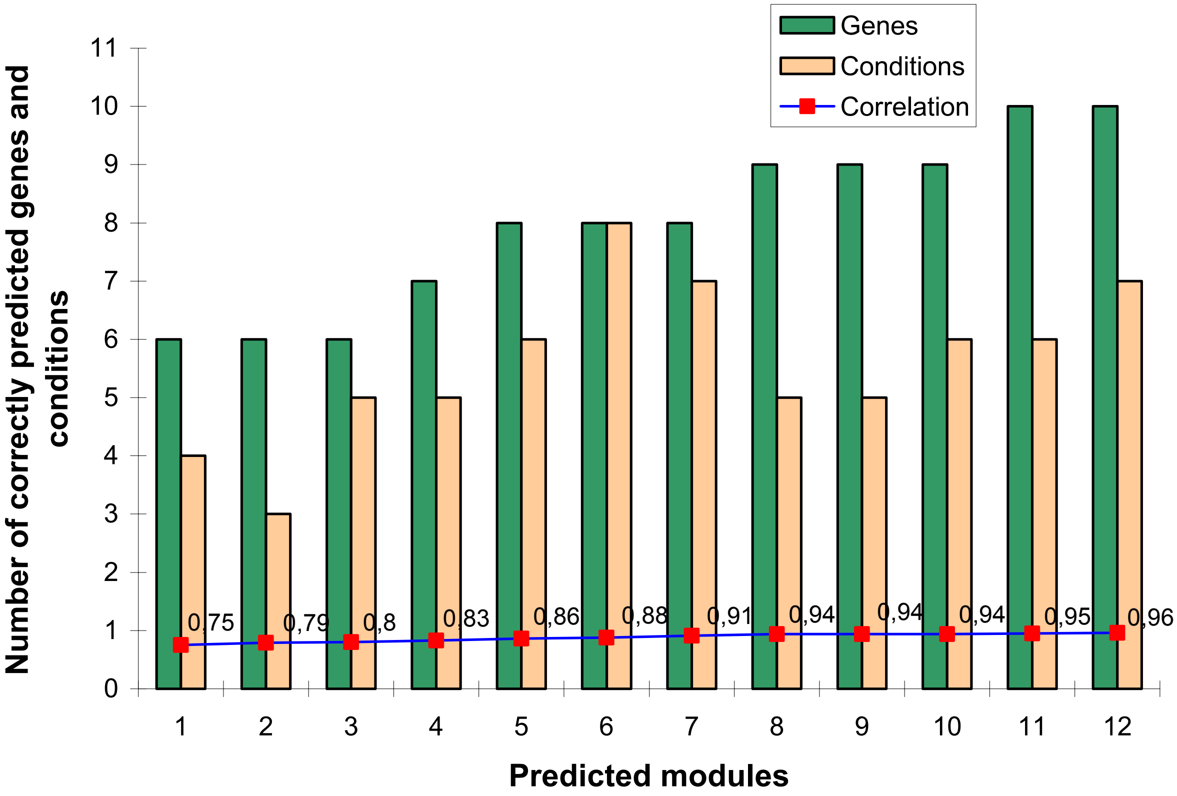


Fig. 3.8. Plot representing correctly predicted genes and conditions with their corresponding correlations.

The GA based module prediction algorithm was applied to this synthetic data for 150 generations. The fitness during each generation is plotted as shown in Fig. 3.7. This Figure depicts the fitness evolution during the GA optimization process.

It can be seen that fit modules could be found within 50 generations. The fittest modules were harvested after 150th generation.

If the evolution works fine, one would expect to find all spiked genes and their respective conditions as fittest modules. The correctly predicted genes and conditions from these modules with their respective correlations were plotted in Fig. 3.8. From the figure, it is clear that as the correlation increases, each predicted module consists of an increased number of spiked genes and conditions. Once it reaches the highest correlation of 0.96, nearly all spiked genes with almost all conditions were predicted as the fittest module, thus proving the algorithm as well as the functioning of evolution on this synthetic dataset.

3.6 *Ustilago maydis* genome and DNA microarray

In order to apply the module prediction algorithm on a real dataset, we considered the DNA microarray datasets from *Ustilago maydis*. *Ustilago maydis*, the causal agent of corn smut disease, has been used the last decades as a model system for studying genetics and pathogen-host interactions. Recently, the fungus has emerged as an excellent experimental model for the molecular genetic analysis of phytopathogenesis, particularly in the characterization of infection-specific morphogenesis in response to signals from host plants.

With the cooperation of BayerCropScience AG*, the *U. maydis* genome sequencing for the strain Um521 was made available to our department. Um521 has an estimated genome size of 20.5 Mb, with 23 chromosomes ranging in size between 350 kb and 2.4 Mb. The *Ustilago maydis* sequence project is also a part of the Broad Institute's Fungal Genome Initiative. The goal of the Broad Institute's± *U. maydis* sequencing project is to release 10X genomic coverage for *U. maydis* strain Um521. The BayerCropScience assembly consists of 28 physical contiguous sequence blocks covering 17.4 Mb.

± http://www.broad.mit.edu/annotation/fungi/ustilago_maydis/ *Bayer Crop Science AG, 40789 Manheim am Rhein, Germany

Taking into consideration that the repetitive elements were not sequenced, sequence information was available for about 93% of the genome and 6297 genes were predicted. On the basis of the gene prediction, a DNA gene chip was designed by Affymetrix that allows the detection of the transcript levels of 6200 *U. maydis* genes.

3.7 Comparison between *U. maydis* and Yeast genomes

Elucidating the functions of the large fraction of *U. maydis* genes, whose functions are currently unknown, is a challenging process. DNA microarrays could provide us with the first step toward the goal of uncovering gene function on a global scale. With the assumption that genes that encode proteins that participate in the same pathway or are part of the same protein complex are often co-expressed, predicting the co-expressed cluster of genes is often the first step towards assigning functions for unknown genes. However, co-expression does not necessarily imply that genes are functionally related. For example, it would be difficult to distinguish accidentally expressed genes from those that are physiologically important. On the other hand, evolutionary conservation is a powerful criterion to identify genes that are functionally important from a set of co-expressed genes (Stuart et al., 2003). With this assumption, we considered looking for conserved modules among evolutionarily distant organisms that consists of a similar set of co-expressed genes under diverse conditions.

With the aim of finding evolutionarily conserved modules between evolutionarily conserved organisms, we considered *Saccharomyces cerevisiae* and *Ustilago maydis* for the analysis. We associated genes from yeast with similar genes of *U. maydis*. Similar genes were identified by performing all-against-all BLAST search (Altschul et al., 1990) between every pair of gene sequences from *U. maydis* with yeast. Around 2966 genes were identified to be similar with an E-value $\leq 10^{-10}$, and around 330 similar genes were identified to be similar with an E-value of 10^{-7} and 10^{-10}. Since the similarity searches were done on the gene level, we considered those highly similar 973 genes whose similarity E-value is $\leq 10^{-65}$ for further analysis.

3.8 Application of the program on a dataset

We sought to identify evolutionarily conserved sets of similar genes from the 973 genes comprising a module that are co-expressed not only in one experiment and in one organism but that also show co-expression in diverse experiments in both organisms. We extracted data corresponding to 973 genes from diverse sets of DNA microarray experiments from yeast and *U. maydis*. Forty-four DNA microarray experiments from yeast and 27 diverse experiments from *U. maydis* were considered (Appendix I).

3.8.1 *U. maydis* microarray data normalization

The data was first scaled to account for the experimental differences between the arrays used by taking the mean of the replicate experiments. Some data points that appeared to be aberrant in all the replicates (Absent signal) were reduced to 0. In order to compare gene expression results from 27 diverse experiments in *U. maydis* Affymetrix Genechips, it was necessary to normalize the microarray data. As the normalization step, 'Per-gene normalization' which compares the results for a single gene across all the samples was done as explained in the section 1.4.2, such that gene expression profiles are ensured to have mean equal to 0 and a standard deviation equal to 1. Since our goal was to identify the genes whose expression change under different conditions, per-gene normalization was necessary to compare the gene expression profiles of genes that may be expressed at very different levels. In addition, it gives information about the expression fold change at each condition with respect to the mean of all conditions.

3.8.2 Yeast microarray data normalization

The fluorescence signal from 44 diverse experimental conditions cDNA microarrays from yeast (Gasch et al., 2000) was first background corrected for red (from dye Cy5) and green (from dye Cy3) intensities for each spot.

Normalization is usually applied to the log-ratios of expression, which is written as

$$M = \log_2 (R / G) \tag{3.3}$$

The data were further normalized as explained in section 1.4.2.

3.9 Results: Evolutionarily conserved modules

In order to find the evolutionarily conserved modules between two organisms, we adopted the following strategy. First, we applied the module prediction algorithm to the microarray dataset of yeast. The *Q* modules predicted from the algorithm were considered as prototype. As a second step, the microarray data corresponding to those co-expressed genes in *Q* modules were collected from *U. maydis*. The module prediction algorithm was applied to this restricted dataset. This restricted analysis was done to test the evolutionary conservation of the modules. If the modules were evolutionarily conserved, one would expect a significant number of co-expressed sets of genes from every module from *Q* modules in other organism. Similar strategy was followed in the vice versa case, in which modules from another dataset are considered as prototypes for *U. maydis*, and the same procedure described above had been followed to predict evolutionarily conserved modules.

With this idea, we first considered the *U. maydis* microarray dataset. The Module prediction algorithm was applied on this dataset. The modules were harvested after the 200th generation and those modules with an average fitness greater than 0.7 were analyzed further. The size of the module was kept constant. Four modules were predicted to satisfy the criteria of an average fitness of greater than 0.7. All 4 modules were found to have a similar set of genes co-expressed among a similar set of conditions.

The best module with a fitness of 0.81 is shown in Fig. 3.9a. The conditions inducing the co-regulation are shown in Fig. 3.9b. The functional roles of the genes that participate in this module are summarized in Table 3.1. This module has

29 genes co-expressed under 11 different conditions in the *U. maydis* dataset. After introducing the threshold parameter of at least 0.5 fold gene expression change, this module reduces to 20 genes co-regulated over 11 conditions.

In order to examine whether this is an evolutionarily conserved module, the microarray expression data corresponding to the 20 genes in this module were collected for all 44 conditions from yeast. If this was an evolutionary conserved module, one would expect a co-regulation of genes from the yeast dataset under different conditions. We applied the module prediction algorithm on the yeast dataset corresponding to the 20 genes from 44 different conditions. The modules were collected after the 200th generation and those modules with an average correlation greater than 0.7 were analyzed further. As a result we found only one module with an average fitness of 0.81 that has 17 genes (Figure 3.10a and 3.10b) that includes 14 ribosomal related genes out of 20 genes from the prototype module from *U. maydis*. These 17 genes were found to be co-regulated over 22 different conditions in yeast and are indicated in red in Table 3.1. This implies that this module is related to protein synthesis and is evolutionarily conserved.

Next we considered the vice versa case. The yeast microarray dataset corresponding to 44 different conditions is taken as the prototype dataset and the module prediction algorithm is applied to the dataset. The modules were harvested after the 200th generation and those modules with average fitness greater than 0.7 were analyzed further. The size of the module was kept constant. There were 3 predicted modules of overlapping genes co-regulated among different conditions. The best module with the average fitness of 0.88 is shown in Fig. 3.11a, the conditions that induce co-regulation are summarized in Fig. 3.11b. The functional roles of the genes that participate in this module are summarized in Table 3.2. This module has 29 genes co-expressed under 19 different conditions in the yeast dataset

In order to examine whether this is an evolutionarily co-expressed module, the microarray data corresponding to genes in this module were collected for all 27 conditions from *U. maydis*. We applied the module prediction algorithm on the *U. maydis* dataset corresponding to the 29 genes from 27 different conditions. The modules were collected after the 200th generation and those modules with an average correlation greater than 0.7 were analyzed further. The best module with an average fitness of 0.73 is shown in Figure 3.12a, the conditions that induce co-regulation of this module is shown in Fig. 3.12b. As a result we found 15 genes that includes 11 amino acid biosynthesis genes out of the 29 genes from the prototype module from yeast were found to be co-expressed over 7 different conditions in *U. maydis* are indicated in red in Table 3.2. This implies that this module is amino acid biosynthesis related and is evolutionarily conserved.

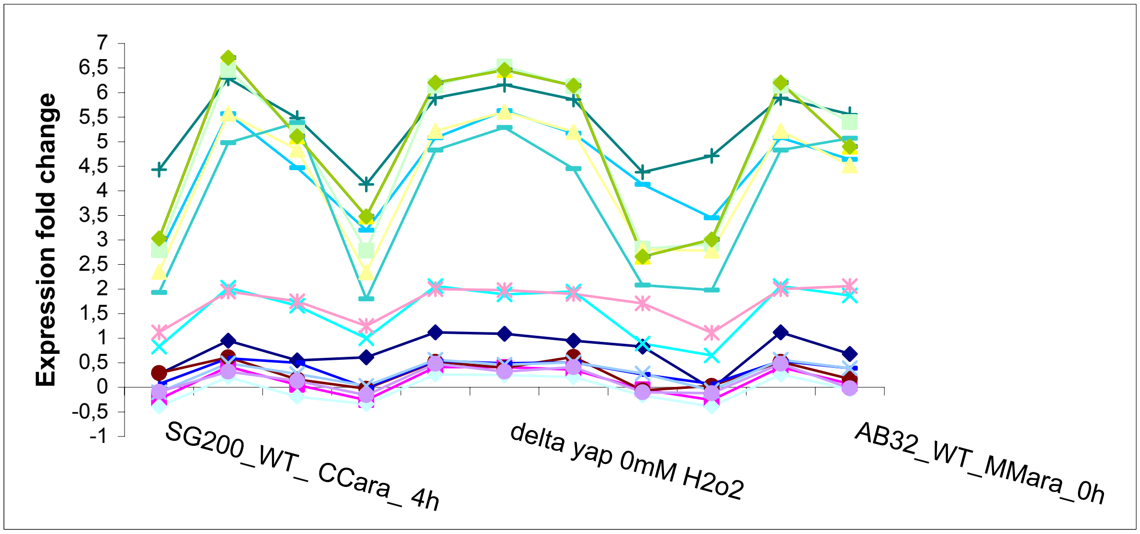


Fig. 3.9a. Co-expressed genes from a protein synthesis specific module in *U. maydis* with correlation of 0.81.

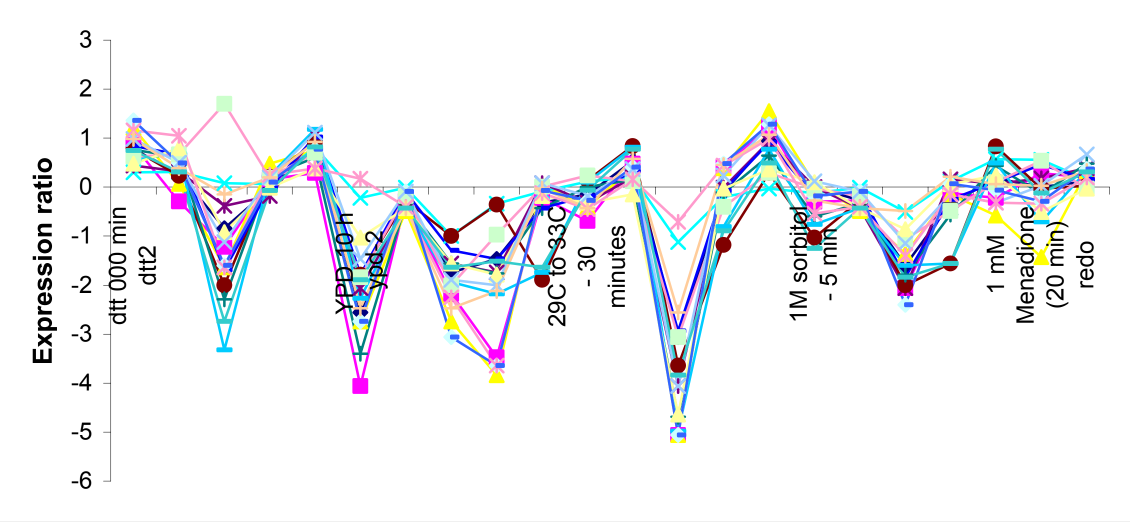
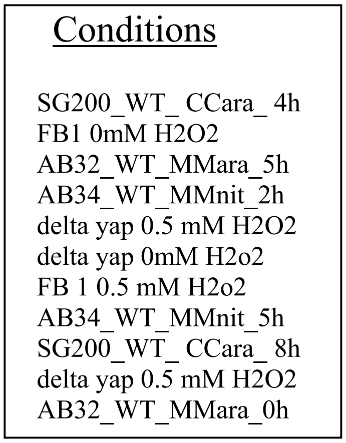


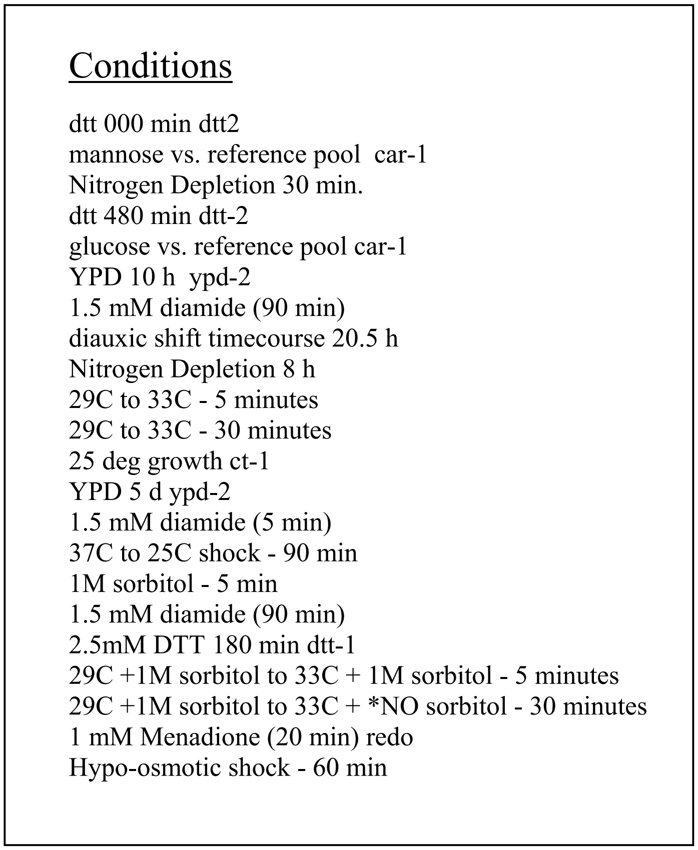
Fig. 3.10a. Evolutionarily conserved cluster from yeast that has 17 genes co-expressed in a protein synthesis specific module from *U. maydis*.



Conditions

SG200_WT_ CCara_ 4h
FB1 0mM H2O2
AB32_WT_MMara_5h
AB34_WT_MMnit_2h
delta yap 0.5 mM H2O2
delta yap 0mM H2o2
FB 1 0.5 mM H2o2
AB34_WT_MMnit_5h
SG200_WT_ CCara_ 8h
delta yap 0.5 mM H2O2
AB32_WT_MMara_0h

Fig. 3.9b. Experimental conditions in *U. maydis* under those co-expression modules are detected



Conditions

dtt 000 min dtt2
mannose vs. reference pool car-1
Nitrogen Depletion 30 min.
dtt 480 min dtt-2
glucose vs. reference pool car-1
YPD 10 h ypd-2
1.5 mM diamide (90 min)
diauxic shift timecourse 20.5 h
Nitrogen Depletion 8 h
29C to 33C - 5 minutes
29C to 33C - 30 minutes
25 deg growth ct-1
YPD 5 d ypd-2
1.5 mM diamide (5 min)
37C to 25C shock - 90 min
1M sorbitol - 5 min
1.5 mM diamide (90 min)
2.5mM DTT 180 min dtt-1
29C +1M sorbitol to 33C + 1M sorbitol - 5 minutes
29C +1M sorbitol to 33C + *NO sorbitol - 30 minutes
1 mM Menadione (20 min) redo
Hypo-osmotic shock - 60 min

Fig. 3.10b. Experimental conditions corresponding to yeast under those co-expression protein synthesis specific modules are detected

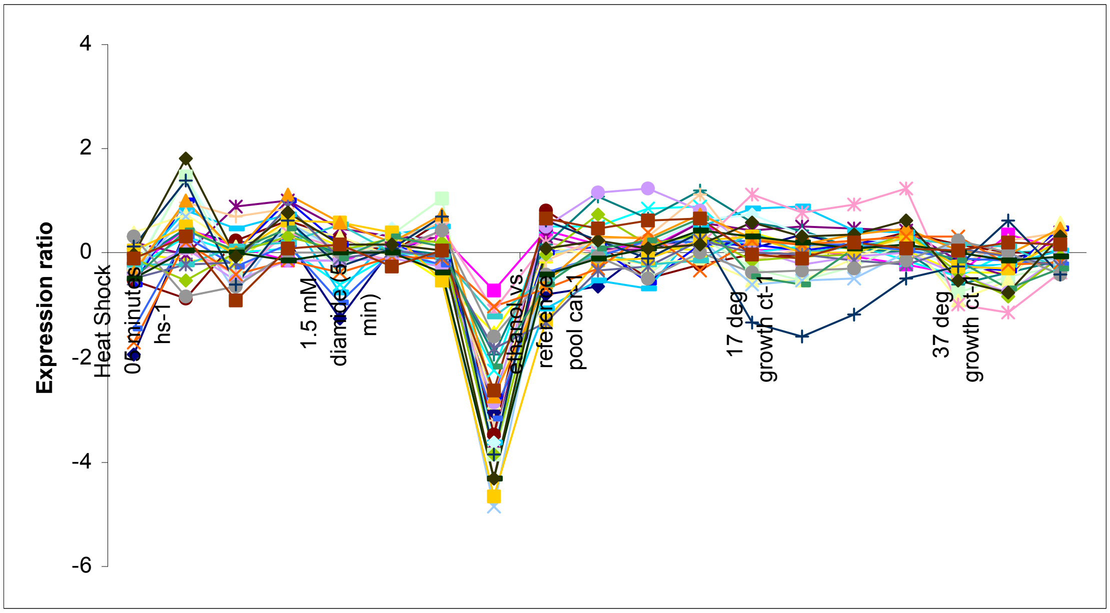


Fig. 3.11a. Co-expressed genes in an amino acid biosynthesis specific module from yeast with a correlation of 0.88.

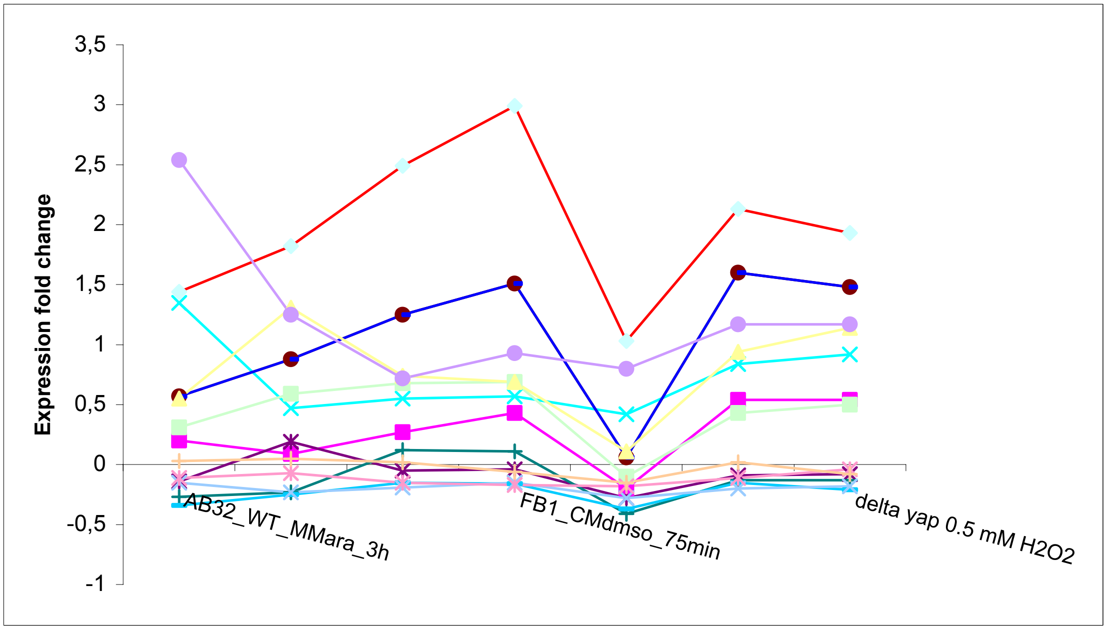


Fig. 3.12a. Evolutionarily conserved cluster from *U. maydis* that has 15 genes co-expressed in an amino acid biosynthesis specific module from yeast.

Conditions

Heat Shock 05 minutes hs-1
37C to 25C shock - 15 min
2.5mM DTT 005 min dtt-1
dtt 000 min dtt-2
1.5 mM diamide (5 min)
Diauxic Shift Time course - 0 h
YPD 2 h ypd-2
YPD 5 d ypd-2
ethanol vs. reference pool car-1
galactose vs. reference pool car-1
raffinose vs. reference pool car-1
sucrose vs. reference pool car-1
17 deg growth ct-1
21 deg growth ct-1
25 deg growth ct-1
29 deg growth ct-1
37 deg growth ct-1
29C +1M sorbitol to 33C + 1M sorbitol - 30 minutes
Hypo-osmotic shock - 60 min

Fig. 3.11b. Experimental conditions in yeast under those co-expression modules are detected.

Conditions

AB32_WT_MMara_3h
AB34_WT_MMnit_1h
FB1_CMa2_75min
FB1_CMdmso_75min
SG200_WT_ CCara_ 8h
FB1 0mM H2O2
delta yap 0.5 mM H2O2

Fig. 3.12b. Experimental conditions corresponding to *U. maydis* under that co-expression of amino acid biosynthesis specific modules are detected

Table 3.1. List of genes in the best co-expressed module from *U. maydis*. Evolutionarily conserved genes from both yeast and *U. maydis* are indicated in red.

Yeast Acc no	*U. maydis* Acc no	Functional class	Function
yor063w	W30UM029G	Protein synthesis	Ribosomal protein L3
ydr023w	C75UM020G	Protein synthesis	tRNA Synthetase, Seryl
ylr340w	W65UM075G	Protein synthesis	Ribosomal protein L10, Acidic
yol097c	W25UM092G	Protein synthesis	tRNA Ligase, Tryptophan
yll045c	C55UM166G	Protein synthesis	Ribosomal protein L8B
ymr121c	C20UM015G	Protein synthesis	Ribosomal protein L15B
ynl178w	C95UM023G	Protein synthesis	Ribosomal protein S3
ybr143c	C30UM066G	Protein synthesis	Translation release factor ERF1 SUBUNIT
yjl138c	C90UM185G	Protein synthesis	Translation initiation factor EIF4A
yer025w	W40UM044G	Protein synthesis	Translation initiation factor EIF2 GAMMA
yor063w	W30UM029G	Protein synthesis	Ribosomal protein L3
yor206w	C107UM044G		Protein involved in biogenesis of the 60S ribosome
ylr276c	C75UM097G		Member of the DEAD-box RNA helicase family, functions in rRNA processing to the precursor of 60S ribosomal subunits, interacts with Dbp6p
yhr066w	W40UM029G		Protein involved in 27S rRNA processing required for the maturation of 25S and 5.8S rRNA products, contains a BRIX domain and is a member of the Imp4p superfamily containing a sigma70-like motif
ypr110c	W15UM070G	Transcription	RNA Polymerase III 40 KD SUBUNIT
ydr390c	C35UM116G	Protein degradation, ubiquitin mediated	Subunit of a heterodimeric enzyme consisting of Uba2p and Aos1p, activates the ubiquitin-like Smt3p for conjugation to other proteins
ymr300c	W60UM216G	Purine biosynthesis	Amidophosphoribosyltransferase, (glutamine phosphoribosylpyrophosphate amidotransferase), catalyzes the first step in de novo purine biosynthesis
ycl030c	C40UM055G	Histidine biosynthesis	Phosphoribosyl-AMP cyclohydrolase / phosphoribosyl-ATP pyrophosphohydrolase / histidinol dehydrogenase, second, third, and tenth steps of histidine biosynthesis pathway
ygl148w	W45UM223G	Aromatic amino acid biosynthesis	Chorismate synthase, bifunctional enzyme with a flavin eductase activity that acts in the phenylalanine, tyrosine and tryptophan biosynthesis
yal036c	C70UM120G		Protein that contains a GTP1/OBG GTP-binding domain

Table 3.2. List of genes in the amino acid biosynthesis specific evolutionarily conserved module from yeast. Evolutionarily conserved genes from both yeast and *U. maydis* are indicated in red.

Yeast Acc no	***U. maydis* Acc no**	**Functional class**	**Function**
yil094c	W70UM127G	Lysine biosynthesis	Homoisocitrate dehydrogenase, converts homoisocitrate to alpha-ketoadipate, the fourth step in the lysine biosynthesis pathway
ynr050c	C145UM005G	Lysine biosynthesis	Saccharopine dehydrogenase (saccharopine reductase; NADP+, L-glutamate forming), catalyzes the seventh step in the lysine biosynthesis pathway
ycl009c	C110UM170G	Isoleucine and valine biosynthesis	Acetolactate synthase regulatory subunit
ygr155w	W60UM043G	Methionine biosynthesis	Cystathionine beta-synthase (beta-CTSase), converts serine and homocysteine to cystathionine
ygr012w	C175UM281G		Protein with similarity to Cys4p Cysteine synthase activity, Biological Process: Amino acid metabolism
yil020c	C65UM085G	Histidine biosynthesis	Phosphoribosyl imidazolecarboxamide isomerase
yer052c	C40UM020G	Methionine and Threonine biosynthesis	Aspartate kinase (L-aspartate 4-P-transferase), catalyzes the first step in the common pathway for methionine and threonine biosynthesis
yor323c	W50UM245G	Proline biosynthesis	Gamma-glutamyl phosphate reductase (phosphoglutamate dehydrogenase), proline biosynthetic enzyme
ybr249c	W30UM026G	Aromatic amino acid biosynthesis	2-Dehydro-3-deoxyphosphoheptonate aldolase (3-deoxy-D-arabino-heptulosonate-7-phosphate synthase or DAHP synthase), inhibited by tyrosine
ylr058c	C55UM072G	L-serine biosynthesis	Serine hydroxymethyltransferase (glycine hydroxymethyltransferase), cytosolic isoform, catalyzes the transfer of the hydroxymethyl group of serine to tetrahydrofolate to form 5,10-methylenetetrahydrofolate and glycine. Glycine hydroxymethyltransferase activity. Biological Process: Formate metabolism; L-serine biosynthesis; Amino acid metabolism
ylr359w	C90UM033G	Purine biosynthesis	Adenylosuccinate lyase, carries out the eighth step in de novo purine biosynthesis
ynl169c	C35UM012G	Phospholipid metabolism	Phosphatidylserine decarboxylase, mitochondrial isozyme, converts phosphatidyl-L-serine to phosphatidylethanolamine

Yeast Acc no	***U. maydis* Acc no**	**Functional class**	**Function**
ylr172c	C50UM258G	Diphthamide biosynthesis	Diphthamide methyltransferase, required for diphthamide biosynthesis S-adenosylmethionine-dependent methyltransferase activity ;Diphthine synthase activity. Biological Process: Peptidyl-diphthamide biosynthesis from peptidyl-histidine.
ygl171w	C90UM186G	rRNA processing	ATP-dependent RNA helicase required for rRNA processing, member of DEAD-box family of RNA helicases
yfl002c	C60UM079G	rRNA processing 25S	ATP-dependent RNA helicase of DEAD-box family, required for processing of 25S ribosomal RNA precursor
ynl161w	W20UM155G	Cell wall biosynthesis (putative)	Serine/threonine protein kinase required for sporulation and production of daughter specific proteins
yer164w	C32UM196G	Transcription	Protein involved in ATP-dependent nucleosome remodeling and DNA replication- independent nucleosome assembly, member of the Chromodomain-Helicase-DNA-binding (CHD) family
ybr118w	W85UM090G	Protein synthesis	Translation elongation factor EF-1alpha, identical to Tef1p
yor168w	C75UM133G	Protein synthesis	Glutaminyl-tRNA synthetase for the cytoplasm and mitochondria
yhr200w	W50UM091G	Protein degradation	Non-ATPase component of the 26S proteasome complex that also functions in RNA polymerase II transcription elongation
yer125w	W50UM054G	Protein degradation, ubiquitin-mediated	Essential ubiquitin-protein ligase (E3 enzyme), a member of HECT domain family of ligases, may be involved in the maintenance and remodeling of actin cytoskeleton during endocytosis
ycl059c	W50UM080G		Component of 90S preribosomal particles in association with small nucleolar RNAs, essential for cell division and spore germination
yil109c	C10UM236G	Secretion	Component of the COPII coat of vesicles, involved in endoplasmic reticulum to Golgi transport
ypr029c	C65UM054G	Secretion	Gamma-Adaptin, large subunit of the clathrin-associated protein (AP) complex
ylr293c	C80UM179G	Nuclear protein targeting	Ran, a GTP-binding protein of the ras superfamily involved in trafficking through nuclear pores
yfl037w	W105UM005G	Cytoskeleton	Tubulin beta chain, required for mitosis and karyogamy
yil103w	C20UM250G		Protein involved in susceptibility to *K. lactis* killer toxin
yor209c	W34UM052G	NAD Biosynthesis	Nicotinatephosphoribosyltransferase (NAPRTase), catalyzes the first step in the Preiss-Handler pathway leading to the synthesis of nicotinamide adenine dinucleotide (NAD)
yor175c	W26UM223G		Member of the membrane bound O-acyl transferase (MBOAT) family, which are found in acyltransferase enzymes, has high similarity to uncharacterized C. glabrata Cagl0l04642gp

Chapter 4

Summary and discussion

The contributions of this thesis are novel methods that exploit biological information from DNA microarray data. DNA microarray technology produces data on a large scale. The comprehensive analysis of this data is often done using the standard clustering algorithms presented in Chapter 1. However, the major drawback of most of these algorithms is th e lack of ability to predict time delayed and local correlations in time course datasets. We proposed two novel approaches to address the problems. The first proposed method is using dynamic programming, and the other is an extension of this method, solving the limitation of local versus global normalization.

In independent datasets, most standard algorithms cluster the data according to all conditions by missing those genes that change their expression only under a limited number of conditions. In order to address this problem, we developed the genetic algorithm based module prediction algorithm to predict modules consisting of co-regulated genes and subset of conditions that induce co-regulation.

4.1 Predicting time delayed correlations

Various methods have been developed in order to extract useful information from gene expression time course datasets. Herwig et al., (1999) used the mutual information between genes as a similarity measure. This measure separates the expression pattern into three states, unchanged normal expression, increased expression and decreased expression. These defined states were then used to estimate the mutual information between the genes. Although in principle three states of expression are sufficient to characterize an expression pattern, a slight variation in the expression level can lead to a dramatic change in the mutual information estimation. Spellman et al., (1998) used Fourier transformations on the time series data to calculate the similarity between the gene expression profiles. However, this method is only suitable for cyclic data such as cell cycle time series that has been analyzed by them.

A method for approximating an ideal similarity measure by training a neural network by learning from a pre-specified target gene expression pattern has been suggested in Sawa and Ohno-Machado, (2003). Qian et al., (2001) addressed the problem of identifying local similarities in gene expression time course data by means of a method based on the Smith-Waterman algorithm for (local) sequence alignment. However, in this approach the data is normalized by converting each expression value to its z score. Such global normalization is critical since it is not in agreement with our goal to discover local similarities. In fact, once should realize that normalizing a complete profile means that two sub-profiles cannot be compared independently of all other expression values.

All of the aforementioned algorithms compare concrete values for the change of gene expression profiles. Wen et al., (1998) suggested a shape based similarity measure that compares two profiles on the basis of qualitative changes of expression values. Thus, two sequences are considered as similar if they increase and decrease more or less simultaneously. However, this measure is still a global one in the sense that all time points are taken into consideration and missing the local time shifted relationships as discussed in Section 1.5.2.

Filkov et al., (2002) proposed a kind of "edge detection" method for periodic datasets with small sequences. This method searches for local regions in pairs of expression profiles where major changes in expression occur (edges). The profiles are regarded as similar if they do have similar edges. Kwon et al., (1999) suggested an "event-based" edge detection method. An event in specific time interval is considered as the directional change of the gene expression curve at that instant. This method converts the raw data to a string of events, such as: "R" representing changes greater than a certain (upper) threshold value, "F" for changes less than a "lower" threshold and "C" for insignificant changes. The event strings are then aligned using a modified version of the Needleman-Wunsch algorithm for global sequence alignment.

By converting a time series into a sequence of "events" such as an increase or decrease, tend to oversimplify the original data. This makes the methods robust toward noise and outliers, but also looses a lot of information contained in the original times series.

The methods that were developed here were aimed at predicting time delayed as well as local correlations from DNA microarray time course data. The method using the dynamic programming predicts clusters with a significant enrichment of genes of similar functions. However, this method has the limitation of local versus global normalization (Section 2.1.8).

Chapter 3.1 introduces the method we developed using dynamic programming algorithm to predict time delayed and local correlation from DNA microarray time course data. We used this method on two different datasets from yeast, mitotic cell cycle (Cho et al., 1998) and heat shock conditions (Gasch et al., 2000). Some of the resulting clusters had a significant number of genes with similar functions. For example, the best significant cluster from heat shock conditions had 63 genes out of 73 genes from the functional category 'Protein synthesis' with the P value of 3.9. 10^{-54}. However, this method has the limitation of local vs. global normalization. Since one cannot normalize locally using the dynamic programming algorithm, the dataset was already normalized globally by converting each expression value to its z score. In this globally normalized dataset, looking for the local similarities is critical.

A new approach to address the problem of finding regions with local similarity in expression profiles is presented in Chapter 3.2. These local regions can be time-shifted to allow for example for the detection of transcription control relationships. The measure of similarity is based on the Spearman rank correlation and can be seen as a good compromise between numerical measures (like Pearson correlation or Euclidean distance) and simple qualitative measures (like measures that consider only "ups" and downs of a time series) that ignore much of the relevant information. Simulations were performed to assess the statistical significance of the obtained degrees of similarity.

The actual comparison of the profiles is then performed with a heuristic sliding-window approach. Using this approach has the advantage that it does not impose restrictions on properties of the similarity measure as do methods that rely on dynamic programming. For example, the Spearman rank correlation could not be used with the approach of Qian et al., (2001), as it does not allow one to calculate the similarity of two profiles directly from the similarities of its sub-profiles.

CLARITY was applied to a dataset of gene expression profiles from the yeast *Saccharomyces cerevisiae* that was measured to study the mitotic cell cycle (Cho et al., 1998). The similarities among the profiles were then used to assign co-expressed genes to clusters. The obtained clusters were divided into two categories, periodic clusters and non-periodic clusters. The periodic clusters contain mainly cell cycle related genes that show as expected, a periodic behavior. The non-periodic clusters contain genes that have a non-periodic expression profile and thus are not directly related to the cell cycle. This result is expected, as the approach considers profiles similar that show a similar shape- and a cyclic behavior is just one of the many possible shapes of a profile.

The obtained clusters were then compared against an existing functional classification of the genes of *Saccharomyces cerevisiae*. Among the clusters with periodic behavior, many of genes encode proteins with cell cycle dependent functions, like DNA-processing, DNA synthesis and DNA replication, reflecting the fact that genes with a similar function are often co-regulated and thus co-expressed. On a reasonable level, one would not expect all genes in clusters to be simple correlations, but considerably more likely than random expectation to have a similar function or a similar cellular role.

One of the advantages of the CLARITY algorithm is that time-shifted as well as local correlations can be discovered. Apart from the proposed time shifted correlations between Ndd1, a cell cycle regulator during S and G2/M transition, Stb5, another transcription factor and Mcm21, a kinetochore protein required for normal cell growth from later S to early M phase, CLARITY predicted the new time delayed correlations between Tps3, and Tps1. Tps3 encodes the regulatory component of the trehalose-6-phosphate synthase/phosphatase complex consisting of Tps1. Further, CLARITY predicted local correlations between Put1 and Put2. Put2p in conjunction with Put1p converts proline to glutamate in the mitochondrion. Further, Put4, Agp1, Dip5 and Gad1, all genes involved in glutamate metabolism, were also predicted to have local similarities with Put1 and Put2 by CLARITY.

Although there is an obvious justification from published biological literature for the time delayed and local relationships like the one between Ndd1, Stb5, and Mcm21, many additional pairs of genes whose functions and relationships need to be further explored experimentally in order to have a better understanding of the gene interactions.

Thus the novel relationships we proposed here should be viewed as a potential hypothesis until they are validated by appropriate biological experiments. This type of hybrid computational and experimental analysis may allow one to investigate more of the gene networks or regulatory networks in future.

4.2 Predicting evolutionarily conserved functional modules

Extracting a set of genes that change their expression over a set of conditions called modules or biclusters can be seen basically as an optimization problem. The first application of the biclustering concept to gene expression data was done by Cheng and Church, (2000). They used a mean squared residue score as similarity score and used a greedy algorithmic approach to find one bicluster, combined iteratively to produce a collection of biclusters. The mean squared residue score is the variance of the set of all elements in the bicluster plus the mean row variance and the mean column variance. Their aim was to find biclusters with low mean squared residue scores, in particular large and maximal ones with scores below a certain threshold. The lowest mean residue score equaling 0 could indicate that the gene expression levels fluctuate in unison. This trivial or constant biclusters were discovered and masked.

Ihmels et al., (2002) proposed a signature algorithm in order to find meaningful modules of biclusters. As a first step, the algorithm receives a set of genes as input and identifies experimental conditions under which the input genes are co-regulated most tightly. This is done by calculating the average change in the expression (or condition scores) of the input genes for each condition and selecting those conditions with large absolute conditional score. In the second step, the algorithm selects those genes that show a significant change in expression under the conditions selected in the first step from the gene expression profiles of all the genes in the genome. This central idea of this work was to integrate prior biological information such as the function or sequence of known genes into the gene expression data analysis.

In Bergmann et al., (2003) the authors presented a complementary method to that of Ihmels et al., (2002) that does not require any prior biological knowledge. They introduced the term called transcription module (TM). A TM contains both set of genes and conditions. The conditions of the TM induce co-regulated expression of the genes belonging to this TM. The degree of similarity is determined by a pair of threshold parameters, the gene threshold and the conditions threshold. They proposed an iterative signature algorithm that searches for transcription modules encoded in the data by iteratively refining sets of genes and conditions until they match their definition of transcription module.

In order to identify “patterns” from gene expression data of cells characterized by a given phenotype and of control cells, Califano et al., (2000) proposed a supervised learning based method. The algorithm randomly selects genes and conditions, and assigns the corresponding bicluster. The chosen bicluster can be considered as a δ valid gene-condition “pattern” if each column in the bicluster is tightly clustered in an interval of size up to δ for a given $\delta > 0$.

Tang and Zhang, (2003) developed a heuristic searching method that adopts the simulated annealing technique to predict the empirical phenotypes and hidden phenotype structures of clinically interested microarray data. Their definition of a phenotype corresponds to a particular macroscopic phenotype such as the presence or absence of clinical syndromes or cancer types. For example, if the gene expression levels in the matrix are discretized into three-level ordinal values i.e., either “high”, “intermediate” or “low”; empirical phenotypes of samples can be discriminated through a small subset of genes whose expression levels strongly correlate with the phenotype distinction possible results could reveal expression levels that are low for one phenotype, intermediate for another phenotype and high for the third empirical phenotype. Their heuristic searching algorithm dynamically measures and manipulates the relationship between conditions and genes while conducting an iterative adjustment of the candidate phenotype structures to approximate the best quality.

Murali and Kasif, (2003) suggested another representation of gene expression data called gene expression motifs or xMOTIFs. A gene’s expression level is conserved across a set of samples if the gene is expressed with the same abundance in the entire sample. A conserved gene expression motif is a subset of genes that is simultaneously conserved across a subset of samples. They employed a heuristic approach to discover large and conserved gene expression motifs that cover all the samples and classes in the data.

A new approach for biclustering based on the Gibbs sampling paradigm called GEMS (Gene Expression Module Sampler) was applied on gene expression data by Wu et al., (2004). Their algorithm starts from a randomly selected module (bicluster) that matches a defined subset of conditions (samples) such as 10 conditions in each module, and uses Gibbs sampling to iteratively update this condition subset to maximize the number of genes in the module.

Tanay et al., (2002) proposed a fast biclustering method called SAMBA (Statistical Algorithmic Method for Bicluster Analysis). SAMBA uses a graph theory approach to find statistically significant clusters. In their approach, the expression data is modeled as a bipartite graph whose two parts correspond to conditions and genes respectively, with edges for significant expression changes. They assigned weights to the vertex pairs of the bipartite graph according to a statistical model so that heavy sub graphs correspond to significant biclusters. They used a polynomial algorithm that reduces under a defined scoring scheme, to find the significant, heaviest sub graphs in the bipartite graph.

Lasseroni and Owen, (2000) proposed "plaid models", in which the expression matrix is considered as a sum of "plaids" – signals that dominate the submatrix. They were interested in finding plaids so that the difference between their sum and the observed signal is only noise. In order to explain "plaids", consider coloring each element in the gene expression matrix with a specific color with each cell colored according to the expression value. The ordering of the rows and the columns is usually arbitrary. By considering ways of reordering the rows and columns in order to group together similar rows and similar columns thus forming an image with blocks of similar color called a 'plaid' pattern. A set of genes behaving similarly in a set of samples defined as a 'layer' in a plaid model context is similar in definition to that of a bicluster.

Ben-Dor et al., (2002) defined a bicluster as an order-preserving sub-matrix (OPSM). According to them, a bicluster is a group of rows whose values induce a linear order across a subset of columns. Their work focuses on the relative order of the columns in the bicluster rather than on the uniformity of the actual values in the data matrix as the plaid model did. They wanted to identify large OPSMs. A submatix is order preserving if there is a permutation of its columns under which the sequence of values in every row is strictly increasing. Liu and Wang, (2003) followed the OPSM idea to define a bicluster as an OP-cluster (Order Preserving Cluster). Their goal was also to discover biclusters with coherent evolutions on the columns.

In this thesis, a genetic algorithm based method for predicting meaningful condition specific biclusters (modules) from gene expression data is proposed. This algorithm was used to compare the common functional modules from gene expression data of two different organisms, *Saccharomyces cerevisiae* and *Ustilago maydis*.

The method was developed to compensate for some of the drawbacks of standard clustering methods that use the biclustering approach on an independent condition dataset. The aforementioned biclustering algorithms use different optimization strategies to predict significant biclusters. We used genetic algorithms to address the problem of biclustering. The first and important point is that genetic algorithms are intrinsically parallel. They consider populations of solutions rather than one solution at a time. Most of aforementioned biclustering algorithms are serial and can only explore the solution space of a given problem in one direction at a time, and if the solution they discover turns out to be sub-optimal, there is nothing can be done but to abandon all work previously completed and start over. However, since GAs have multiple offspring, they can explore the solution space in multiple directions at once. If one path turns out to be a dead end, they can easily eliminate it and continue work on more promising avenues, giving them a greater chance of finding the optimal solution in each run.

Bleuler et al., (2004) proposed a hybrid evolutionary framework that can be coupled to existing biclustering methods. They used the top-down biclustering approach that starts with the entire gene expression matrix that iteratively partitions it to smaller biclusters. Compared to Bleuler et al., (2004), we used a bottom-up approach that start with a population of biclusters that are iteratively modified, until no local improvement is possible anymore. Bleuler et al., (2004) started with the whole gene expression matrix and searched for the largest bicluster, that is then optimized. Since they are not considering all possible solutions to the problem, smaller biclusters might be overseen. In our approach, we are selecting an initial population of solutions by that maintaining diversity in a better way. Moreover, we used different model evaluation, operators and selection strategies compared to them.

We have used the module prediction algorithm on the *U. maydis* DNA microarray independent dataset to predict significant biclusters. Since the *U. maydis* genome lacks proper annotation, the resulting modules could not be functionally compared.

Aforementioned biclustering algorithms have been applied on at least annotated genome DNA datasets like *Saccharomyces cerevisiae.*

Up to now, co-expression of sets of genes under large number of conditions has been considered as one of the criteria in determining function. Stuart et al., (2003) used phylogenetic conservation as a very strong criterion to identify functionally relevant co-expression links among genes. Significant co-expression of two or more orthologous genes across evolutionarily distant organisms is very likely due to selective advantage, strongly suggesting a functional relation. In their approach they selected big datasets from four divergent organisms: *Homo sapiens*, *Saccharomyces cerevisiae*, *Drosophila melanogaster* and *Caenorhabditis elegans*. The orthologous genes among these organisms were selected by performing reciprocal BLAST searches. They wanted to identify pairs of orthologous genes that were co-expressed among multiple organisms showing correlation with respect to diverse experiments. They used the Pearson correlation as a correlation measure and then ranked all genes according to it. They constructed a multiple species co-expression network using the probability of observing the gene-gene correlation by chance using the technique of order statistics. Pellegrino et al., (2004) used a similar strategy as that of Stuart et al., (2003) to predict putative functional relationships among genes from two closely related organisms, human and mouse. They used orthologous ESTs instead of genes, and proposed a data mining method to predict similar ESTs. In these methods, they used huge datasets to find the correlation. This increases the background noise and can lead to missing of those important conditions that change the expression of orthologous genes significantly. If the available data is limited as in the case of *U. maydis* then one needs to find an alternative solution.

Interestingly, Bergmann et al., (2004) presented the comparison of homologous modules among six different organisms. In this approach, starting from a list of co-expressed genes associated with a particular function from one organism as seed, they identified the homologues in another organism by using BLAST. The co-expressed homologues were selected further by their signature algorithm, and genes that were not identified based on sequence homology but share similar expression profiles were predicted further. By restricting the analysis only to a list of functionally similar genes, other orthologous genes are missed in this approach. Systematic study of all orthologous genes could help exploring all conserved modules among evolutionarily distant organisms.

In order to supplement the *U. maydis* annotation, we used the phylogenetic conservation criteria from Stuart et al., (2003). The orthologous genes between yeast and *U. maydis* were found by all against all BLAST search. We have chosen 973 genes (Appendix II) with the E value of 10^{-65} for further analysis, reciprocal BLAST was performed on these genes to check the consistency. Stuart et al. used the cut off E value of 10^{-5} for the best BLAST hit and found that about 2000 genes from yeast had orthologous from other 3 organisms. By comparing the orthologous gene list from Stuart et al., (2003), we found that 612 genes out of 2000 genes occurred in our 973 orthologous gene list. The remaining 361 genes out of 973 in our list could be those genes that were not conserved among other 3 organisms.

In sequence analysis, significant sequence similarity that may reflect functional conservation of the protein is often considered to be at least 25% in protein level. Considering a small E value of 10^{-5} as in Stuart et al. will be critical. Since we consider a higher cut off E value of 10^{-65} for similarity, the remaining ≈1400 genes from the orthologous gene list of Stuart et al. could be those genes that have E value greater than 10^{-65}.

The microarray datasets corresponding to 973 orthologous genes between both organisms were extracted. The module prediction algorithm was applied to the microarray dataset from one organism. In order to check the evolutionary conservation of the modules between organisms, the modules predicted for one organism were checked further on the microarray dataset of another organism and vice versa. The predicted modules from the algorithm from two organisms had genes of similar function. Most notably, the module consist of 20 genes predicted from the *U. maydis* dataset co-expressed over 11 different conditions with a protein synthesis specific function, was found to be an evolutionarily conserved module with that of the yeast dataset over 22 different conditions. Similarly, another module that consists of 29 genes from yeast, co-expressed in over 19 different conditions, with an amino acid biosynthesis specific function, was found to be evolutionarily conserved with that of the *U. maydis* dataset over 7 different conditions. Since we have limited array data from *U. maydis*, the number of predicted modules is small. Although we could predict the conserved functional modules between two organisms, further modules could be predicted based on more experimental data.

Compared to the approach of Bergmann et al., (2004), we followed a reciprocal approach. We have predicted modules from the list of orthologous genes from two organisms. Bergmann et al. selected a subset of functionally similar genes from one organism and searched for homologues in other organisms as well as their co-expression behavior. By comparing our conserved modules to that of Bergmann et al. we found that only two genes from our protein synthesis specific module were predicted by them in the homologous module of ribosomal proteins in *S. cerevisiae*. Other genes from our protein synthesis specific module as well as amino acid biosynthesis specific module were not predicted by Bergmann et al. as homologous modules. Since Bergmann et al. started the analysis with subsets of similar genes and not with all orthologous genes among the 6 organisms, the probability of missing other conserved modules like our protein synthesis specific module as well as amino acid biosynthesis is significant.

Further, one should consider the fact that these predicted modules from our approach were from entirely different microarray experimental setups from the two organisms. Nevertheless, the modules predicted by the algorithm have functional similarity. This could entail that if a set of genes is co-expressed under set of conditions from one organism they can also be co-expressed under a different set of experimental conditions in other organisms, thus having a selective advantage implying they could be functionally similar.

4.2.1 Future directions

The final results of the module prediction algorithm appear to be ad-hoc, having many ‘tuning parameters’ determined by hand. While checking the algorithm on synthetic data, we have considered the dimension of the dataset as 100 x 50. In order to have a complete picture of the threshold dataset size and population size a systematic investigation is needed.

Further, in our approach parameters like the size of the module is kept constant whenever the algorithm is applied on the dataset. This could restrict those other genes that also behave similarly over the same set of conditions or, on the other hand those conditions that induce co-regulation over the same set of genes. An upgrade such as increasing the size of the module to a bigger one is needed. This could be done one way by first having a fixed module size say m x n, and comparing the resulting modules for overlapping genes between the modules. Defining a threshold parameter say p, and if the number of overlapping genes between modules is above x then one could combine both modules to $(2m-p)$ x n, conversely, if the number of overlapping conditions between modules is above p, one could combine both modules to m x $(2n-p)$.

Further other fitness functions such as multiple correlation coefficients can be used in future. ‘Multiple correlation’ represented as R^2 provides the simultaneous calculation of the correlation coefficient of several variables. Since in our case the fitness of the module depends on number of genes or conditions in the module, applying multiple correlations will be useful.

Since the annotation of *U. maydis* is not complete, one can transfer the yeast annotation to *U. maydis* for the predicted evolutionarily conserved modules, further experimental investigations are needed to prove these proposed modules.

Appendix I

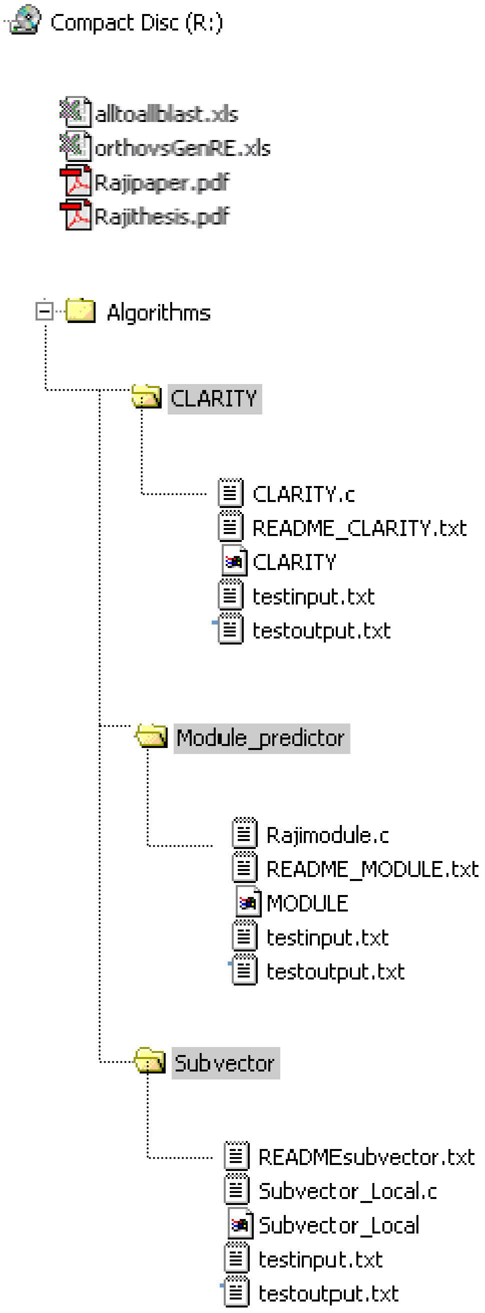
Table A-1. Yeast and *U. maydis* microarray experiments considered for analysis

Experimental Conditions from *U. maydis* (Min: minutes, h: hours, d: days)	Experimental Conditions from yeast (Min: minutes, h: hours, d: days)
AB32_WT_MMara_0h	Heat Shock 05 min hs-1
AB32_WT_MMara_12h	37C to 25C shock - 15 min
AB32_WT_MMara_1h_1	29C to 33C - 5 min
AB32_WT_MMara_2h_1	29C +1M sorbitol to 33C + 1M sorbitol - 5 min
AB32_WT_MMara_3h	29C +1M sorbitol to 33C + *NO sorbitol - 5 min
AB32_WT_MMara_5h	Constant 0.32 mM H2O2 (10 min) redo
AB34_WT_MMnit_0h	1 mM Menadione (10 min) redo
AB34_WT_MMnit_12h	2.5mM DTT 005 min dtt-1
AB34_WT_MMnit_1h	dtt 000 min dtt-2
AB34_WT_MMnit_2h	1.5 mM diamide (5 min)
AB34_WT_MMnit_3h	1M sorbitol - 5 min
AB34_WT_MMnit_5h	Hypo-osmotic shock - 5 min
AJ1-2 NM	Amino acid starvation 0.5 h
AJ38 -33 -2	Nitrogen Depletion 30 min.
FB1 I NM	Diauxic Shift Time course - 0 h
FB1_CMa2_360min	YPD 2 h ypd-2
FB1_CMa2_75min	YPD 5 d ypd-2
FB1_CMdmso_360min	Ethanol vs. reference pool car-1
FB1_CMdmso_75min	Galactose vs. reference pool car-1
GE38 NM	Glucose vs. reference pool car-1
SG200_WT_ CCara_ 4h	Mannose vs. reference pool car-1
SG200_WT_ CCara_ 8h	Raffinose vs. reference pool car-1
FB1 0mM H2O2	Sucrose vs. reference pool car-1
FB 1 0.5 mM H2o2	17 deg growth ct-1
FB1 5 mM H2o2	21 deg growth ct-1
Δ yap 0mM H2o2	25 deg growth ct-1
Δ yap 0.5 mM H2O2	29 deg growth ct-1
Δ yap1 5 mM	37 deg growth ct-1
	Heat Shock 060 minutes hs-2
	37C to 25C shock - 90 min
	29C to 33C - 30 minutes
	29C +1M sorbitol to 33C + 1M sorbitol - 30 min
	29C +1M sorbitol to 33C + *NO sorbitol - 30 min
	Constant 0.32 mM H2O2 (160 min) redo
	1 mM Menadione (20 min) redo
	2.5mM DTT 180 min dtt-1
	dtt 480 min dtt-2
	1.5 mM diamide (90 min)
	1M sorbitol - 120 min
	Hypo-osmotic shock - 60 min
	Amino acid starvation 6 h
	Nitrogen Depletion 8 h
	Diauxic shift time course 20.5 h
	YPD 10 h ypd-2
	1 mM Menadione (160 min) redo

Appendix II

Supplementary data and Program codes

The supplementary data and program codes of the three developed methods are presented in the CD-ROM. The data is presented in the following order.



```
Compact Disc (R:)

    alltoallblast.xls
    orthovsGenRE.xls
    Rajipaper.pdf
    Rajithesis.pdf

  Algorithms
      CLARITY
            CLARITY.c
            README_CLARITY.txt
            CLARITY
            testinput.txt
            testoutput.txt

      Module_predictor
            Rajimodule.c
            README_MODULE.txt
            MODULE
            testinput.txt
            testoutput.txt

      Subvector
            READMEsubvector.txt
            Subvector_Local.c
            Subvector_Local
            testinput.txt
            testoutput.txt
```

In these files, alltoallblast.xls represents our all to all reciprocal BLAST search results and our orthologous genes list. Further, this excel sheet has the comparison of our results with that of Stuart et al., (2003). OrthovsGenRE.xls represents the comparison of our orthologous genes list with that the result of GenRE* results from MIPS. The files, Rajipaper.pdf and Rajithesis.pdf represents the published paper and the pdf version of this respectively. The algorithm folder is divided into 3 subfolders: 1) CLARITY, 2) Module and 3) Subvector. Each subfolder has the respective source code, executable code, sample input file, sample output file and README file.

* http://mips.gsf.de/genre/proj/ustilago/

Bibliography

Alizadeh, A. A., Eisen, M. B., Davis, R. E., Ma, C., Lossos, I. S., Rosenwald, A., Boldrick, J. C., Sabet, H., Tran, T., Yu, X.*, et al.* (2000). Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling, Nature *403*, 503-511.

Altman, R. B., and Raychaudhuri, S. (2001). Whole-genome expression anaylsis: challenges beyond clustering, Current Opinion in Structural Biology *11*, 340-347.

Altschul, S. F., Gish, W., Miller, W., Meyers, E. W., and Lipman, D. J. (1990). Basic Local Alignment Search Tool, Journal of Computational biology *215*, 403-410.

Altschul, S. F., Madden, T. L., Schäffer, A. A., Zhang, J., Zhang, Z., Miller, W., and Lipman, D. J. (1997). Gapped BLAST and PSI-BLAST: a new generation of protein database search programs, Nucleic Acids Research *25*, 3389-3402.

Bäck, T., and Hoffmeister, F. (1991). Extended Selection Mechanisms in Genetic Algorithms. Paper presented at: Proceedings of the Fourth International Conference on Genetic Algorithms (San Mateo, California, USA, Morgan Kaufmann Publishers).

Bairoch, A. (2000). The ENZYME database in 2000, Nucleic Acids Research *28*, 304-305.

Bairoch, A., and Apweiler, R. (1999). The SWISS-PROT protein sequence data bank and its supplement TrEMBL in 1999, Nucleic Acids Research *27*, 49-54.

Barker, W., Garavelli, J., McGarvey, P., Marzec, C., BC Orcutt, Srinivasarao, G., Yeh, L., Ledley, R., Mewes, H., Pfeiffer, F.*, et al.* (1999). The PIR-International Protein Sequence Database, Nucleic Acids Research *27*, 39-43.

Barrett, J. C., and Kawasaki, E. S. (2003). Microarrays: the use of oligonucleotides and cDNA for the analysis of gene expression, Drug Discovery Today *8*, 134-141.

Batagelj, V., and Mrvar, A. (2003). Pajek - Analysis and Visualization of Large Networks. In Graph Drawing Software, P. Mutzel, M. Junger, S. o. G. D. Vienna, and M. Junger, eds. (Berlin, Springer), pp. 77-103.

Ben-Dor, A., Chor, B., Karp, R., and Yakhini, Z. (2002). Discoveing local strucutre in gene expression data: The order-preserving submatrix problem. Paper presented at: Proceedings of the 6th International Conference on Computational Biology (RECOMB'02) (Washington DC, USA).

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

Bergmann, S., Ihmels, J., and Barkai, N. (2004). Similarities and Differences in Genome-Wide Expression Data of Six Organisms, PLoS Biology *2*, 85-93.

Bilu, Y., and Linial, M. (2002). The advantage of functional prediction based on clustering of yeast genes and its correlation with non-sequence based classification, Journal of Computational biology *9*, 193-210.

Bleuler, S., Prelic, A., and Zitzler, E. (2004). An EA Framework for Biclustering of Gene Expression Data. Paper presented at: Congress on Evolutionary Computation (CEC 2004).

Brazma, A., Jonassen, I., Vilo, J., and Ukkonen, E. (1998). Predicting gene regulatory elements in silico on a genomic scale, Genome Research *8*, 1202-1215.

Brown, M. P. S., Grundy, W. N., Lin, D., Cristianini, N., Sugnet, C. W., Furey, T. S., Jr, M. A., and Haussler, D. (2000). Knowledge-based analysis of microarray gene expression data by using support vector machines, Proc Natl Acad Sci USA *97*, 262-267.

Bussemaker, H. J., Li, H., and Siggia, E. D. (2001). Regulatory element detection using correlation with expression, Nature Genetics *27*, 167-171.

Califano, A., Stolovitzky, G., and Tu, Y. (2000). Analysis of gene expression microarrays for phenotype classification, Proc ISMB, 75-85.

Cheng, Y., and Church, G. M. (2000). Biclusteiring of expression data, Proc ISMB *8*, 93-103.

Cho, R. J., Campbell, M. J., Winzeler, E. A., Steinmetz, L., Conway, A., Wodicka, L., Wolfsberg, T. G., Gabrielian, A. E., Landsman, D., Lockhart, D. J., and Davis, R. W. (1998). A Genome-Wide Transcriptional Analysis of the Mitotic Cell Cycle, Molecular Cell *2*, 65-73.

Chu, S., DeRisi, J., Eisen, M., Mulholland, J., Botstein, D., Brown, P. O., and Herskowitz, I. (1998). The Transcriptional Program of Sporulation in Budding Yeast, Science *282*, 699-705.

Davis, L. (1991). Handbook of Genetic Algorithms (New York, Van Nostrand Reinhold).

Drysdale, R. A., Crosby, M. A., and Consortium, T. F. (2005). FlyBase: genes and gene models, Nucleic Acids Research *33*, D390-D395.

Duggan, D. J., Bittner, M., Chen, Y., Meltzer, P., and Trent, J. M. (1999). Expression profiling using cDNA microarrays, Nature Genetics *21*, 10-14.

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

Eisenhaber, F., and Bork, P. (1999). Evaluation of human-readable annotation in biomolecular sequence databases with biological rule libraries, Bioinformatics *15*, 528-535.

Fickett, J. W., and Wasserman, W. W. (2000). Discovery and modeling of transcriptional regulatory regions, Current Opinion in Biotechnology *11*, 19-24.

Filkov, V., Skiena, S., and Zhi, J. (2002). Analysis techniques for microarray time-series data, Journal of Computational biology *9*, 317-330.

Gasch, A. P., Spellman, P. T., Kao, C. M., Carmel-Harel, O., Eisen, M. B., Storz, G., Botstein, D., and Brown, P. O. (2000). Genomic Expression Programs in the Response of Yeast Cells to Environmental Changes, Molecular Biology of the Cell *11*, 4241–4257.

GeneOntologyConsortium (2000). The Gene Ontology (GO) database and informatics resource, Nucleic Acids Research *32*, D258-D261.

Gerstein, M., and Jansen, R. (2000). The current excitement in bioinformatics-analysis of whole geneme expression date: how does it relate to protein structure and function?, Current Opinion in Structural Biology *10*, 574-584.

Gerstein, M., and Levitt, M. (1998). Comprehensive assessment of automatic structural alignment against a manual standard, the scop classification of proteins, Protein science *7*, 445-456.

Goffeau, A., Barrell, B. G., Bussey, H., Davis, R. W., Dujon, B., Feldmann, H., Galibert, F., Hoheisel, J. D., Jacq, C., Johnston, M.*, et al.* (1996). Life with 6000 Genes, Science *274*, 546-567.

Goldberg, D. E. (1989). Genetic Algorithms in Search, Optimization, and Machine Learning, Addison Wesley.

Halfon, M., Carmena, A., Gisselbrecht, S., Sackerson, C., Jiménez, F., Baylies, M., and Michelson, A. (2000). Ras pathway specificity is determined by the integration of multiple signal-activated and tissue-restricted transcription factors, Cell *103*, 63-74.

Hartigan, J. A. (1975). Clustering Algorithms, John Wiley and Sons.

Herwig, R., Poustka, A. J., Müller, C., Bull, C., Lehrach, H., and O'Brien, J. (1999). Large-Scale Clustering of cDNA-Fingerprinting Data, Genome Research *9*, 1093-1105.

Holland, J. H. (1975). Adaptation in natural and artificial systems, The University of Michigan Press.

Holland, J. H. (1992). Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence (Cambridge, MA, MIT Press).

Hughes, J. D., Estep, P. W., Tavazoie, S., and Church, G. M. (2000). Computational identification of Cis-regulatory elements associated with groups of functionally related genes in *Saccharomyces cerevisiae*, Journal of Molecular Biology *296*, 1205-1214.

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

International human genome sequencing consortium (2004). Finishing the euchomatic sequence of the human genome, Nature *431*, 931-945.

J. van Helden, B. André, and Collado-Vides., J. (1998). Extracting Regulatory Sites from the Upstream Region of Yeast Genes by Computational Analysis of Oligonucleotide Frequencies, Journal of Molecular biology *281*, 827-842.

Jansen, R., Greenbaum, D., and Gerstein, M. (2001). Relating Whole-Genome Expression Data with Protein-Protein Interactions, Genome Research *12*, 37-46.

Jiang, D., Chun Tang, and Zhang, A. (2004). Cluster Analysis for Gene Expression Data: A Survey, IEEE transactions on knowledge and data engineering *16*, 1370-1386.

Kane, M. D., Jatkoe, T. A., Stumpf, C. R., Lu, J., Thomas, J. D., and Madore, S. J. (2000). Assessment of the sensitivity and specificity of oligonucleotide (50mer) microarrays, Nucleic Acids Research, *28*, 4552-4557.

Kanehisa, M., and Goto, S. (2000). KEGG: Kyoto Encyclopedia of Genes and Genomes, Nucleic Acids Research *28*, 27-30.

Karypis, G. (2002). CLUTO- a clustering toolkit. In Technical report 02-017 (Department of computer science, University of Minnesota).

Kel, O. V., Romaschenko, A. G., Kel, A. E., Wingender, E., and Kolchanov, N. A. (1995). A compilation of composite regulatory elements affecting gene transcription in vertebrates, Nucleic Acids Research *23*, 4097-4103.

Keseler, I. M., Collado-Vides, J., Gama-Castro, S., Ingraham, J., Paley, S., Paulsen, I. T., Peralta-Gil, M., and Karp, P. D. (2005). EcoCyc: a comprehensive database resource for *Escherichia coli*, Nucleic Acids Research *33*, D334-D337.

Kwon, A. T., Hoos, H. H., and Ng, R. (1999). Inference of transcriptional relationships from gene expression data, Bioinformatics *19*, 905-912.

Lan, N., Jansen, R., and Gerstein, M. (2002). Towards a systematic definition of protein function that scales to the genome level: Defining function in terms of interactions, Proceedings of the IEEE *90*, 1848-1858.

Lasseroni, L., and Owen, A. (2000). Plaid models for gene expression data. In Technical report (Standford University).

Licciulli, F., Catalano, D., D'Elia, D., Lorusso, V., and Attimonelli, M. (1999). KEYnet: a keywords database for biosequences functional organization, Nucleic Acids Research *27*, 365-367.

Lipshutz, R. J., Fodor, S. P. A., Gingeras, T. R., and Lockhart, D. J. (1999). High density synthetic oligonucleotide arrays, Nature Genetics *21*, 20-24.

Liu, J., and Wang, W. (2003). OP-Cluster: Clustering by Tendency in High Dimensional Space. Paper presented at: Third IEEE International Conference on Data Mining (Melbourne, Florida).

Madeira, S. C., and Oliveira, A. L. (2004). Biclustering algorithms for biological data analysis: a survey, IEEE/ACM Transactions on Computational Biology and Bioinformatics *1*, 24-45.

Mewes, H. W., Frishman, D., Güldener, U., Mannhaupt, G., Mayer, K., Mokrejs, M., Morgenstern, B., Münsterkötter, M., Rudd, S., and Weil, B. (2002). MIPS: a database for genomes and protein sequences, Nucleic Acids Research *30*, 31-34.

Moreau, Y., Smet, F. D., Thijs, G., Marchal, K., and Moor, B. D. (2002). Functional bioinformatics of microarray data: from expression to regulation, Proceedings of the IEEE *90*, 1722-1743.

Murali, T. M., and Kasif, S. (2003). Extracting conserved gene expression motifs from gene expression data, In proceedings of the Pacific symposium on biocomputing *8*, 77-88.

Nobrega, M. A., Ovcharenko, I., Afzal, V., and Rubin, E. M. (2003). Scanning Human Gene Deserts for Long-Range Enhancers, Science *302*, 413.

Pearson, W. R. (1998). Empircal statistical estimates for sequence similarity searches, Journal of Molecular biology *276*, 71-84.

Pellegrino, M., Provero, P., Silengo, L., and Cunto, F. D. (2004). CLOE: Identification of putative functional relationships among genes by comparision of expression profiles between two species, BMC Bioinformatics *5*, 179.

Press, W. H., Teukolsky, S. A., vetterling, W. T., and Flannery, B. P. (2002). Nonparametric or Rank Correlation. In Numerical recipies in C, pp. 639-645.

Qian, J., Dolled-Filhart, M., Lin, J., Yu, H., and Gerstein, M. (2001). Beyond synexpression relationships: Local clustering of time-shifted and inverted gene expression profiles identifies new, biologically relevant interactions, Journal of Molecular biology *314*, 1053-1066.

Quackenbush, J. (2001). Computational analysis of microarray analysis, Nature Reviews Genetics *2*, 418-427.

Quandt, K., Grote, K., and Werner, T. (1996). GenomeInspector: basic software tools for analysis of spatial correlations between genomic structures within megabase sequences, Genomics *33*, 301-304.

Ramaswamy, S., Tamayo, P., Rifkin, R., Mukherjee, S., Yeang, C.-H., Angelo, M., Ladd, C., Reich, M., Latulippe, E., Mesirov, J. P*., et al.* (2001). Multiclass cancer diagnosis using tumor gene expression signatures, Proc Natl Acad Sci USA *98*, 15149-15154.

Rison, S. C. G., Hodgman, T. C., and Thornton, J. M. (2000). Comparision of functional annotation schemes for genomes, Funct Intergr Genomics *1*, 56-69.

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

Sawa, T., and Ohno-Machado, L. (2003). A neural network-based similarity index for clustering DNA microarray data, Computers in biology and medicine *33*, 1-15.

Schaerer-Brodbeck, C., and Reizeman, H. (2000). *Saccharomyces cerevisiae* Arc35p works through two genetically separable camodulin functions to regulate the actin and tubulin cytoskeletons, Journal of Cell Science *113*, 521-532.

Schena, M., D., S., W., D. R., and PO, B. (1995). Quantitatice monitoring of gene expression patterns with a complementary DNA microarray, Science *270*, 467-470.

Schena, M., Shalon, D., Heller, R., Chai, A., Brown, P. O., and Davis, R. W. (1996). Parallel human genome analysis: Microarray-based expression monitoring of 1000 genes, Proc Natl Acad Sci USA *93*, 10614-10619.

Serres, M. H., Goswami, S., and Riley, M. (2004). GenProtEC: an updated and improved analysis of functions of *Escherichia coli* K-12 proteins, Nucleic Acids Research *32*, D300-D302.

Smith, T. F., and Waterman, M. S. (1981). Identification of common molecular subsequences, Journal of Molecular biology *147*, 195-197.

Spellman, P. T., Sherlock, G., Zhang, M. Q., Iyer, V. R., Anders, K., Eisen, M. B., Brown, P. O., Botstein, D., and Futcher, a. B. (1998). Comprehensive Identification of

Cell Cycle-regulated Genes of the Yeast *Saccharomyces cerevisiae* by Microarray Hybridization, Molecular Biology of the Cell *9*, 3273-3297.

Stuart, J. M., Segal, E., Koller, D., and Kim, S. K. (2003). A Gene-Coexpression network for Global Discovery of Conserved Genetic Modules, Science *302*, 249-255.

Tamames, J., Ouzounis, C., Casari, G., Sander, C., and Valencia, A. (1998). EUCLID: Automatic Classification of Proteins in Functional Classes by Their Database Annotations, Bioinformatics *14*, 6542-6543.

Tamayo, P., Slonim, D., Mesirov, J., Zhu, Q., Kitareewan, S., Dmitrovsky, E., Lander, E., and Golub, T. (1999). Interpreting gene expression with self-organizing maps: Methods and application to hematopoietic differentiation, Proceedings of the National Academy of Sciences, USA *96*, 2907-2912.

Tanay, A., Sharan, R., and Shamir, R. (2002). Discovering statistically significant biclusters in gene expression data, Bioinformatics *18*, S136-S144.

Tang, C., and Zhang, A. (2003). Mining Multiple Phenotype Structures Underlying Gene Expression Profiles. Paper presented at: In Proceedings of 12th ACM International Conference on Information and Knowledge Management (CIKM 2003) (New Orleans, Louisiana).

Tavazoie, S., Hughes, J. D., Campbell, M. J., J.Cho, R., and Church, G. M. (1999). Systematic determination of genetic network architecture, Nature Genetics *22*, 281-285.

Wang, J., Ellwood, K., Lehman, A., Carey, M. F., and She, Z.-S. (1999). A mathematical model for synergistic eukaryotic gene activation, Journal of Molecular biology *286*, 315-325.

Wen, X., Fuhrman, S., Michaels, G. S., Carr, D. B., Smith, S., Barker, J. L., and Somogyi, R. (1998). Large-scale temporal gene expression mapping of central nervous system development, Proc NatlAcadSci USA *95*, 334-339.

White, K. P., Rifkin, S. A., Hurban, P., and Hogness, D. S. (1999). Microarray Analysis of *Drosophila* Development During Metamorphosis, Science *286*, 2179-2184.

Whitley, D. (1989). The GENITOR Algorithm and Selection Pressure: Why Rank-Based Allocation of Reproductive Trials is Best. Paper presented at: Proceedings of the Third International Conference on Genetic Algorithms (San Mateo, California, USA, Morgan Kaufmann Publishers).

Wu, C.-J., Fu, Y., Murali, T. M., and Kasif, S. (2004). Gene expression module discovery using gibbs sampling, Genome Informatics *15*, 239-248.

Yu, H., Luscombe, N., Qian, J., and Gerstein, M. (2003). Genomic analysis of gene expression realtionships in transciptional regulatory networks, Trends in Genetics *19*, 422-427.

Yuh, C.-H., Bolouri, H., and Davidson, E. H. (1998). Genomic Cis-Regulatory Logic: Experimental and Computational Analysis of a Sea Urchin Gene, Science *279*, 1896-1902.

Acknowledgements

I thank my husband Karthik for his unconditional love, perseverance for more than two years and encouragement during every step in my PhD. Without his support I could not imagine completing this thesis. I sincerely thank my parents, Balasubramaniyan, Jeyalakshmi Balasubramaniyan and my brother Vijayan Balasubramaniyan for their constant love and support in all my endeavors. I especially thank my mother for being so kind and loving during all my hardest times in life.

I thank my mentors PD Dr. Jörg Kämper and Prof. Dr. Eyke Hüllermeier for providing me scientific advice and resources to complete this thesis. Most of all, I credit them for providing me with ample academic independence to proceed into areas such as bioinformatics, which had not been an area of expertise for their laboratory.

I personally thank Jörg for all his help and support during many unfavorable situations in my personal life. I thank him for being so friendly, jovial and helpful. I thank Dr. Mario Scherer and Miroslav Vranes for many useful suggestions and help regarding microarray data. I personally thank Miro for always being a good friend to me. I thank all my lab mates for their fun and support.

I would like to thank my other mentors, Prof. Hans-Werner Mewes from MIPS, and Dr. Sabine Tornow from Universität Augsburg for introducing me to gene expression data analysis and for their constant advice and support.

I would like to thank Prof. Dr. Regine Kahmann for providing me opportunity to work in her department. I thank Nils Weskamp for his advice and suggestions that helped me to complete this thesis. I thank my thesis committee members, Prof. Dr. Michael Bölker and Prof. Dr. Uwe Maier for their kind acceptance and valuable comments. I thank Max-Planck Society for providing me fellowship during my PhD at Max-Planck institute for terrestrial microbiology.

I thank my friend Bernadette Heinze for her encouragement and support that made it possible to finish my thesis. I thank my friend Elamparithi Jayamani for his help and advice during many occasions. I thank my friends Nicole, Lazaro, Artemeo and Fernanda for those memorable moments and funny jokes.

CURRICULUM VITAE

Name	Rajarajeswari Balasubramaniyan
Birth place	Madurai, Tamilnadu, India
Date of Birth	15th July 1976
Gender	Female
Marital status	Married

Current Position

From July 2002 to till now	Pursuing PhD in the field of "Gene expression data analysis using novel methods: Predicting time delayed correlations and genetically conserved condition specific functional modules" from Max-Planck-Institute for Terrestrial Microbiology, Karl-von-Frisch-Strasse, D-35043 Marburg, Germany.
Jan 2002-Jun 2002	Research student at Max-Planck Institute for Terrestrial microbiology, D-35043 Marburg, Germany.
April 2001-Dec 2001	Bioinformatic position at DSQ Biotech limited, Bangalore, India.
August 1999-March 2001	Bioinformatic position at Avesthagen graine limited, Bangalore, India.
August 1998- August 1999	Postgraduate diploma in Bioinformatics, First class. School Of Biotechnology, Madurai Kamaraj University, Madurai-625021, India.
July 1996- June 1998	Master of Science in Physics, 1996-1998. First class. Madura College, Madurai Kamaraj University, Madurai-625021, India
June 1993- June 1996	Bachelor of Science in Physics, 1993-1996. First class. Meenakshi College, Madurai Kamaraj University, Madurai-625002, India.