Evaluation of cDNA microarray ratio calculating strategies and finding transcriptional factor binding sites for co-regulated genes by combining sequence over-representation with cross-species conservation

Jia, Hui

2007

Jia, H. (2007). Evaluation of cDNA microarray ratio calculating strategies and finding transcriptional factor binding sites for co-regulated genes by combining sequence over-representation with cross-species conservation. Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/6580>

<https://doi.org/10.32657/10356/6580>

Nanyang Technological University

Downloaded on 26 Apr 2023 18:36:25 SGT

**Evaluation of cDNA Microarray Ratio Calculating
Strategies
&
Finding Transcriptional Factor Binding Sites for
Co-regulated Genes by Combining Sequence
Over-representation with Cross-species
Conservation**

Jia Hui

School of Biological Sciences

A thesis submitted to the Nanyang Technological University

in fulfillment of the requirements for the degree of

Doctor of Philosophy

2007

Acknowledgments

I would like to express my deepest gratitude and appreciation to my supervisor, Dr. Li Jinming for his guidance, encouragement and patient assistance, without whom this work may never have been completed.

I also would like to thank my lab mates: Ms. Lu le, Ms. Xiao Jie, Mr. Hu Guangan, Mr. Lin Xin and Mr. Yang Ye for their valuable suggestions.

Table of contents

ACKNOWLEDGMENTS	I
TABLE OF CONTENTS	II
SUMMARY	IV
LIST OF FIGURES	VII
LIST OF TABLES.....	VIII
LIST OF TABLES.....	VIII
CHAPTER 1 PREPROCESSING OF DNA MICROARRAY DATA	1
1.1 DNA MICROARRAY	1
1.2 cDNA MICROARRAY	2
1.3 OLIGONUCLETIDE ARRAY	8
1.4 MICROARRAY IMAGE ANALYSIS	11
1.4.1 Addressing	14
1.4.2 Segmentation	14
1.4.3 Information extraction	15
1.5 QUALITY CONTROL	17
1.6 NORMALIZATION	18
CHAPTER 2 SIMULATION STUDY OF CDNA MICROARRAY RATIO CALCULATION STRATEGIES.....	21
2.1 INTRODUCTION	21
2.2 METHOD	28
2.3 RESULTS.....	35
2.4 DISCUSSION	59
2.5 CONCLUSION	62
CHAPTER 3 GENE EXPRESSION PROFILING	63
3.1 CLUSTERING	63
3.1.1 K-Means	65
3.1.2 Hierarchical clustering	66
3.2 MICROARRAY GENE EXPRESSION PROFILING AND DNA BINDING MOTIF FINDING	67
FINDING DNA BINDING MOTIFS.....	70
CHAPTER 4	70
4.1 TRANSCRIPTIONAL FACTORS AND DNA BINDING MOTIFS	70
4.2 REPRESENTATION OF DNA MOTIFS	70
4.2.1 Consensus sequence.....	70
4.2.2 PSSM.....	74

4.3 EXPERIMENTAL METHOD FOR IDENTIFYING TRANSCRIPTION FACTOR BINDING SITES	75
4.4 COMPUTATIONAL METHODS OF FINDING DNA BINDING MOTIFS	76
4.4.1 Expectation Maximization	77
4.4.2 Gibbs Sampling Algorithm	78
4.4.3 Limitation of EM and Gibbs sampling	79
4.4.4 Phylogenetic footprinting	79
CHAPTER 5 FINDING TF BINDING MOTIFS BY COMBINING SEQUENCE OVERREPRESENTATION AND CROSS-SPECIES CONSERVATION	81
5.1 MATERIAL	81
5.2 METHOD	83
5.2.1 Finding overrepresented sequence patterns	83
5.2.2 Relative entropy and Conservation criteria	85
5.2.3 Building a profile for a candidate motif	91
5.2.4 Building PSSM	92
5.2.5 Optimal motif length	93
5.3 RESULTS	94
5.4 DISCUSSION	112
5.5 CONCLUSION	116
5.6 IMPLEMENT OF THE METHOD	116
REFERENCES	117

Summary

The thesis consists of two parts: 1).Evaluation of cDNA Microarray Ratio Calculating Strategies; 2).Finding Transcriptional Factor Binding Sites for Co-regulated Genes by Combining Sequence Over-representation with Cross-species Sequence Conservation.

Part One: *Evaluation of cDNA Microarray Ratio Calculating Strategies*

There are different methods for calculating channel ratios for spots on two-color cDNA microarray slides. At image analysis step, microarray users have to choose one out of the available methods at their own discretion as no guidelines are provided by image analysis software such as *GenePix Pro*. The ratio values calculated using different methods to the same spot may be of substantial variations. Therefore, our study was to address one of the most frequently asked questions by microarray users: which ratio quantity provided by the image analysis software should be used? Standard microarray image analysis software such as the *Axon GenePix Pro* calculate the channel ratio out of the pixels that define a given feature using five different methods, i.e., Ratio of Means, Ratio of Medians, Mean of Ratios, Median of Ratios and Regression Ratio. We have evaluated the five different ratio calculation strategies using simulation approaches. Our results suggest that in most circumstances the Ratio of Means appears to be the prime approach, particularly when the CVs (Coefficient of Variance) of two channel pixel intensities are small (less than 0.5) and channel intensities are large. However the

Ratio of Medians and the Median of Ratios are more favorable when the CVs are large.

Part Two: *Finding Transcriptional Factor Binding Sites for Co-regulated Genes by Combining Sequence Over-representation with Cross-species Sequence Conservation*

One major challenge of modern biology is to understand how gene expression is regulated. An important step toward this understanding is to reliably identify the transcriptional factor binding sites (TFBSs) of co-regulated genes. Novel computational methods for finding TFBSs have long been sought owing to the arduous lab work. The current prevailing TFBS finding methods yield a large number of false positive predictions due to the short, variable nature of transcriptional binding sites (TFBSs). To reduce the false positive rate of motif finding we proposed a method here that combines sequence over-representation and cross-species sequence conservation to detect TFBSs in upstream regions of a given set of co-regulated genes. Putative motifs that are statistically overrepresented are selected to go through a filtering process which eliminates the candidates with low degree of conservation across multiple closely related species, and we proposed to measure the cross-species conservation of a putative motif by the average relative entropy of the corresponding blocks in the alignments of gene upstream sequences from multiple species. Only the candidate motifs exceeding certain threshold of relative average entropy will be kept for building the binding motif profile. We applied this method to 35 *S. cerevisiae* transcriptional factors

with known DNA binding motifs (with the support of orthologous sequences from genomes of *S. mikatae*, *S. bayanus* and *S. paradoxus*), and found that our method outperformed the single-genome based motif finding methods *MEME* and *AlignACE* as well as the multiple-genome based methods *PHYME* and *Footprinter* for the majority of these transcriptional factors.

List of Figures

Figure 1-1 cDNA Microarray.....	4
Figure 1-2 Four layers of a typical cDNA microarray project.....	7
Figure 1-3: oligonucleotide microarray	9
Figure 1-4: From dye to signal.....	12
Figure 1-5 A composite image of cDNA microarray	13
Figure 1-6 Different ways of defining local background region for a spot	16
Figure 2-1 Spot and pixels	23
Figure 2-2 Intensity distribution of pixels	30
Figure 2-3 Results of simulation 1 and simulation 2	38
Figure 2-4 High resolution results of simulation 1 and simulation 2	40
Figure 2-5 Results of simulation 3	43
Figure 2-6 Impact of intensity of spot on the choice of ratio calculating method ..	51
Figure 2-7 Impact of CV of pixels on the choice of ratio calculating method	52
Figure 2-8 Accuracy of different methods	55
Figure 2-9 Precision of different methods	56
Figure 2-10 Precision of different methods (CV)	58
Figure 4-1 A sequence logo.....	73
Figure 5-1 Conservation of motif sequence among close related species	88
Figure 5-2 minus logarithm of p -value versus average relative entropy	89
Figure 5-3 minus logarithm of p -value versus Z -value.....	90

List of Tables

Table 2-1 Ratios calculated with five different methods	27
Table 2-2 Results of simulation 4	48
Table 4-1 IUPAC code.....	72
Table 5-1 Transcription factors and the genes being regulated	99
Table 5-2 Known motifs that can be found by our method	100
Table 5-3 Comparison of our method with <i>AlignACE</i>	102
Table 5-4 Comparison of our method with <i>MEME</i>	104
Table 5-5 Comparison of our method with <i>PHYME</i>	106
Table 5-6 Rank frequencies of the 35 known TF binding motifs found with different methods.	108
Table 5-7 The TFs whose known binding sites only can be found by our method.....	109
Table 5-8 TFs whose binding site can not be found by our method	110
Table 5-9 Binding motifs of two <i>Drosophila</i> transcription factors.....	111

Chapter 1 Preprocessing of DNA Microarray Data

1.1 DNA microarray

Traditional methods in molecular biology generally work on a "one gene in one experiment" basis, which means that the throughput is very limited and the "whole picture" of gene function is hard to obtain. In the past decade, a new technology, called DNA microarray, has attracted tremendous interests among biologists[1]. This technology promises to monitor the whole genome on a single chip so that researchers can have a better picture of the interactions among thousands of genes simultaneously[2-5].

DNA microarray, or DNA chip is an orderly arrangement of samples. It provides a medium for matching known and unknown DNA samples based on base-pairing rules and automating the process of identifying the unknowns. An array is fabricated by high-speed robotics[6], generally on glass but sometimes on nylon substrates, for which probes with known identity are used to determine complementary binding, thus allowing massively parallel gene expression and gene discovery studies. The sample spot sizes in microarray are typically less than 200 microns in diameter. An experiment with a single DNA chip can provide researchers information on thousands of genes simultaneously - a dramatic increase in throughput. The property that DNA microarray can be used to measure

the expression levels of thousands of genes in parallel makes it widely adopted in many fields[4, 7, 8] such as drug discovery[9] and disease study[10].

There are two major variants of the DNA microarray technologies, in terms of the property of arrayed DNA sequences[11]. They are cDNA microarray and oligonucleotide microarray[12].

1.2 cDNA microarray

With cDNA microarray we measure the expression levels of genes using fluorescent dye intensities. The probes are pieces of DNA that are complementary to part of the genes under study. Usually cDNA probes for making the array can be generated from a commercially available cDNA library ensuring a close representation of the entire genome of an organism on the array. Alternatively, PCR using specific primers can be used to amplify specific genes from genomic DNA to generate the cDNA probes.

Doing a cDNA microarray[13] experiment we need to prepare two samples for hybridization to the array: a control sample and an experimental sample(see Figure1-1)[14]. These samples are prepared with mRNA extracted from cells and reverse transcribed into cDNA, and during the reverse transcription step a fluorescent dye is incorporated into the newly formed cDNA. Different dyes should be employed to label the different samples. For example, the control sample can be labeled with a green-fluorescing dye called Cy3 and experimental sample

labeled with a red-fluorescing dye called Cy5. Since the samples are labeled differently they can be combined and hybridized to the microarray together (see Figure 1-1). The two samples will competitively bind to the probes on the array and the sample containing more gene expression for a particular probe will win out. That is, if there is more of an mRNA transcript in the control sample than in the experimental sample (i.e. the gene is down-regulated in the experiment) then more Cy3 will bind to the probe on the array and the spot will fluoresce green. If there is more experimental transcript, the reverse will happen and the spot will fluoresce red. When the two samples have the same amount of transcript, the dyes will cancel each other out and the spot will have a yellow color in the composite image for two channels.

Figure 1-1 cDNA Microarray

- **Preparation of targets:** Amplification and purification of cDNA clones
- **Slide Surface Preparation:** Slide coating chemistry
- **Target DNA Attachment :** Binding and retention of arrayed DNA samples
- **Preparation of Probe:** Isolation of mRNA, and fluorescence dye labeling of the cDNA probe
- **Hybridization:** Probe depletion and mixing
- **Washing:** Eliminating non-hybridized and non specific binding probes
- **Scanning:** Obtaining fluorescent image of each spot

- **Data analysis:** Extraction of intensity from both color channel, determination of local background and variance, estimation of error and standard deviation in the ratios, selection and identification of differentially expressed genes. (excerpted from [14])

Four layers of a typical cDNA microarray project

A typical research project using cDNA microarray can be considered as having four different layers, namely pixel, spot, slide and project (Figure 1-2). A microarray project usually comprises multiple (tens or hundreds of) slides that use samples under different conditions (different cell types, different development stage ..., etc.). It is also quite often to make replicate slides for the same condition to make sure the microarray data are reproducible. A cDNA microarray slide consists of thousands of spots, and each spot usually represents one gene. A spot on microarray slide consists of hundreds of pixels. The two channel intensities of each pixel are detected by image analysis software, whereas the two channel intensities and the channel intensity ratio of the spot are estimated (after eliminating the effect of background signals) from the channel intensities of all the pixels in it.

Figure 1-2 Four layers of a typical cDNA microarray project

A microarray project usually comprises multiple slides that use samples under different conditions. A cDNA microarray slide consists of thousands of spots, and each spot usually represents one gene. A spot on microarray slide consists of hundreds of pixels.

1.3 Oligonucleotide array

Oligonucleotide array (trademarked as a GeneChip by Affymetrix) uses the same principle as cDNA but with different technology. In the oligonucleotide array system a known gene or potentially expressed sequence is represented on the chip by 11-16 unique oligomer probes, each 25 bases long. These probes are selected to have little cross-reactivity with other genes so that non-specific hybridization will be minimized. The synthesis for the probe Affymetrix arrays uses two techniques: photolithography and solid-phase DNA synthesis[12].

To deal with non-specific hybridization that may occur, a second probe is synthesized in an adjacent cell which is identical to the previous probe with the exception of a mismatched base at the central position. This is called the Perfect Match/Mismatch (PM/MM) probe strategy. Any background hybridization with the MM probe is subtracted from the PM probe signal which results in perfect hybridization.

After samples are prepared by extracting mRNA from a cell, the mRNA labeling process is initiated, in which the double stranded cDNA is synthesized from the mRNA, and the cDNA serves as a template in an in vitro transcription (IVT) reaction that produces biotin-labeled antisense mRNA which is called cRNA. Once labeled, the sample of cRNAs can be hybridized to the array and bound by the various oligonucleotide probes. Lastly, a staining reaction is performed in order to visualize the amount of hybridization. (Figure 1-3)[15]

Figure 1-3: oligonucleotide microarray

This figure is the flowchart of oligonucleotide microarray. After samples are prepared by extracting mRNA from a cell, double stranded cDNA is synthesized from the mRNA, and then biotin-labeled cRNA is transcribed using cDNA as template. Once labeled, the sample of cRNAs can be hybridized to the array and bound by the various oligonucleotide probes. Lastly, a staining reaction is

performed in order to visualize the amount of hybridization(The figure was excerpted from [15][15]).

1.4 Microarray Image analysis

With the help of a set of devices, including laser scanner and a charge coupled device (CCD) camera, we can obtain the fluorescent images. The scanner works in the flowing process (Figure 1-4)[16]: photons are excited by laser, after absorbing these photons, fluorescent molecules emit fluorescence photons. A fraction of these fluorescence photons are gathered by a lens. The detector of the scanner converts the emission photons into electric current. Photomultiplier tube (PMT) is the most popular detector. Finally, an analog to digital (A/D) converter is used to convert the electrons into a sequence of digital signals. In cDNA microarray experiment, the scanner produces two 16 bit tiff images, one for each fluorescence dye. Figure 1-5 is a composite image for two channels. The slide shown in Figure 1-5 consists of 16 blocks, and each block has 24×24 spots.

Figure 1-4: From dye to signal

This figure is the flowchart of how the scanner converts the dye to digit signal.

Inside a scanner, fluorescence dyes absorb the energy from the excitation light given out by the laser and emit photons. A PMT detector then converts and amplifies the photons to electrons. An A/D converter finally converts the signal into a digital signal. (The figure was excerpted from [16])

Figure 1-5 A composite image of cDNA microarray

This figure is a composite image for two channels. The slide consists of 16 blocks, and each block has 24×24 spots.

After obtaining the microarray image, mRNA abundance can be evaluated by measuring the intensity of fluorescence of each spot on the array. This process is called image analysis. Image analysis is an important aspect of microarray experiments, which can have great effect on the downstream analyses such as clustering or the identification of differentially expressed genes. The purpose of image analysis is to extract for each spotted DNA sequence a measure of transcript abundance in the two labeled mRNA samples, as well as to obtain background estimates and quality measures.

There are mainly three steps in image analysis, *i.e.*, addressing, segmentation and information extraction[16, 17].

1.4.1 Addressing

Addressing is a fundamental step of microarray image analysis which detects grid structure for the accurate localization of each spot. Addressing deals mainly with separation between rows and columns of spots within each grid. Automating this part of the process can allow high throughput analysis.

1.4.2 Segmentation

The segmentation of an image allows the separation between foreground pixels or background pixels. There are different methods of microarray image segmentation, *e.g. fixed circle segmentation, adaptive circle segmentation, adaptive shape*

segmentation and *histogram*[16, 17]. Different microarray image analysis software use different regions to estimate the background intensity. For example, *Quantarray* consider the area between two concentric circles[18] (green circles in Figure 1.6). *ScanAnalyz* choose all the pixels in a square whose center is the spot but not within the circle region[19] (blue square). *Spot*[20] and *Genepix* use four diamond areas to estimate the background[21]. All of these methods calculate the median of the pixels in the background region as background.

1.4.3 Information extraction

Addressing and segmentation allow us to detect the location and size of each spot, as well as to determine the foreground and the background of a spot. The next step is to extract information such as foreground and background intensities, ratio of two channel signal intensities, quality factor, etc. for each spot on microarray slides.

Figure 1-6 Different ways of defining local background region for a spot

The region inside the red circle represents the spot mask and the other regions bounded by colored lines represent regions of local background used in different image analysis software. Green: used in *Quantaray*; blue: used in *ScanAlyze*; pink: used in *Spot* (The figure was excerpted from [16]).

1.5 Quality control

Microarray technology enables measurements of the expression levels of genes in parallel. On one hand, replication experiment can reduce the random error and make the data more credible. On the other hand, due to the variability created during microarray printing, hybridization and staining, there may be great difference between replications. So the quality of data might be incredulous. As erroneous measurements may have a drastic impact on the results by disturbing the normalization schemes and by introducing expression patterns that lead to incorrect conclusions, it is crucial to discard low quality observations in the early phases of a microarray experiment. It is essential to employ a quality control strategy since a microarray experiment, even under optimal conditions, results in several spots whose intensities may vary due to experimental variation. It is also crucial to exclude spots whose quality is poor in the early stages of a microarray study. There are several factors that affect the quality of a spot on a microarray.

1).Spot intensity: signal intensity is traditionally considered one of the most important features affecting spot quality. The reason is that if a signal is weak, it is very difficult to distinguish the actual signal from the background. Most weak spots are caused by the fact that many genes are physiologically expressed at very low levels, at or close to the sensitivity limit of the cDNA microarrays. In addition, experimental factors that may cause low signal intensities are: low amount of cDNA in the spot; suboptimal labeling; uneven or incomplete hybridization; signal bleaching, low sensitivity of the scanner.

2).Spot size: as microarray spots are produced by depositing an equal amount of liquid onto the microarray slide, all spots are expected to be of roughly equal size. Whether a spot is significantly larger or smaller than expected, it is an indication that there has been some error during manufacturing, which should be reflected when determining the quality of the spot.

3.) Spot morphology: spot are expected to be roughly circular in shape, but manufacturing of the microarrays may cause some variation to spot morphology.

Image analysis software such as *GenePix Pro* or *Quantarray* can automatically flag the spot as “bad” or “not found” according to a set of criteria related to pixels number, spot diameter, spot shape, signal vs. background ratio, etc. Moreover, for some spots the channel intensities after background subtraction may be less than 0, we eliminate these spots also from overall ratio calculation.

1.6 Normalization

For cDNA microarray, the purpose of dye normalization is to balance the fluorescence intensities of the two dyes (green Cy3 and Cy5 dye) as well as to allow the comparison of expression levels across experiments (slides). Dye bias can be most obviously seen in an experiment where two identical mRNA samples are labeled with different dyes and subsequently hybridized to the same slide. In

this situation, it is rare to have the dye intensities equal on average and often the intensities are higher for the green dye. This bias can stem from a variety of factors including physical properties of the dyes, efficiency of dye incorporation, experimental variability in probe coupling and processing procedures, and scanner settings at the data collection step. Many of these factors, whether internal or external to the sample, present unique difficulties to a global normalization procedure. Furthermore, the relative gene expression levels (as measured by log ratios) from replicate experiments may have different spreads due to differences in experimental conditions. Some scale adjustment may then be required so that the relative expression levels from one particular experiment do not dominate the average relative expression levels across replicate experiments.

The current available microarray data normalization methods can be classified into two categories: 1. total intensity normalization, 2. regression based normalization. These methods could be performed either locally or globally on microarray slides. For example, normalization method used by *GenePix Pro* is a kind of global total intensity method. It is based on the assumption that most genes on the array will not be differentially expressed, and therefore the arithmetic mean ratio of two channel signal intensities of every gene (feature) on a given array should be equal to 1.0. If the mean is not 1.0, a scaling factor is computed and used to the ratio of each spot so that their mean value returns to 1.0. We referred this scaling factor as normalization factor. In real practice, the gene expression data (the two channel

ratios) from microarray experiments are often transformed to log scale values. Transforming expression data to a log scale removes much of the proportional relationship between random error and signal intensity. Moreover, log transformation has the advantage of transforming the error model from a proportional to an additive one because $\log(a/b) = \log(a) - \log(b)$, although non-logged difference scores $(a-b)$ have also been recommended for low signals. Distributions of replicated logged expression values and consequently of log ratios tend to be normal.

To calculate normalization factor, *GenePix Pro* first takes the log of each ratio value, the geometric mean $\bar{x}$ of a set of n ratios $\{x\}$ is therefore:

$$\bar{x} = \exp \left[\frac{1}{n} \sum_{i=1}^n \ln x_i \right] ,$$

where n is the total number of spots on the slide subtracted by the number of spots eliminated from further analysis in the quality control step. The inverse $\frac{1}{\bar{x}}$ is then defined by *GenePix Pro* as normalization factor.

Chapter 2 Simulation Study of cDNA Microarray Ratio Calculation Strategies

2.1 Introduction

We usually refer a spot on microarray slides as a feature. An important task in information extraction stage of microarray image analysis is to calculate the ratio of two-channel intensities for a feature from the intensities of the foreground and background pixels that define the spot. Each of the above mentioned image analysis software offers one or more methods in quantifying the channel ratios of spots on a microarray slide. These methods can be roughly divided to three subgroups: those based on quantities derived from whole features (such as median and mean pixel intensities), those based on pixel-by-pixel ratios of intensities, and those based on regression of pixel intensities of two channels. In this chapter we focused on the evaluation of different ratio computing strategies as provided by *Axon GenePix Pro*. In order to check the general validity of our conclusion, we also used a series of mouse cDNA microarray data processed by another well known image analysis package *QuantArray* in our simulation studies.

For each feature on the slide, *Axon GenePix Pro* extracts information such as feature intensities, background intensities, channel ratio, quality flag, etc. *Axon GenePix Pro* utilizes five different methods to calculate the channel ratio quantities

for an individual spot from image pixels that define the feature (see Figure 2-1).

They are Ratio of Medians (*Romd*), Ratio of Means (*Romn*), Median of Ratios (*Mdor*), Mean of Ratios (*Mnor*) and Regression Ratio (*Rg*).

Figure 2-1 Spot and pixels

This figure displays an amplified feature (spot) on a cDNA microarray slide. Each feature is defined by hundreds of pixels.

Computing Ratio Quantities

In the following discussion, I_{P,λ_k} is a single raw pixel intensity at channel k ($k=1,2$), I_{B,λ_k} is a single background pixel intensity at channel k ($k=1,2$), $\langle(S)_i\rangle_{med}$ is the median of dataset S that contains i terms, n is the number of foreground pixels in a feature, and m is the number of background pixels in a feature[21].

Axon GenePix Pro computes the channel ratio quantities for each of the spot on the microarray slide with the following five different methods:

Ratio of Medians (*Romd*)

The *Romd* is the ratio of the background subtracted median pixel intensity of the two channels (channel 2 vs. channel 1)[21]:

$$r_1 = \frac{\langle(I_{P,\lambda_2})_n\rangle_{med} - \langle(I_{B,\lambda_2})_m\rangle_{med}}{\langle(I_{P,\lambda_1})_n\rangle_{med} - \langle(I_{B,\lambda_1})_m\rangle_{med}}$$

Ratio of Means (*Romn*)

The *Romn* is the ratio of the arithmetic means of the background subtracted pixel intensities of the two channels (channel 2 vs. channel 1)[21]:

$$r_2 = \frac{\sum_{i=1}^n \left(I_{P,\lambda_2} - \left\langle \left(I_{B,\lambda_2} \right)_m \right\rangle_{med} \right)_i}{\sum_{i=1}^n \left(I_{P,\lambda_1} - \left\langle \left(I_{B,\lambda_1} \right)_m \right\rangle_{med} \right)_i}$$

Median of Ratios (*Mdor*)

The *Mdor* is the median of the pixel-by-pixel ratios of the pixel intensities (Channel 2 vs. Channel 1) that have the median background intensity subtracted[21]:

$$r_3 = \left\langle \left(\frac{I_{P,\lambda_2} - \left\langle \left(I_{B,\lambda_2} \right)_m \right\rangle_{med}}{I_{P,\lambda_1} - \left\langle \left(I_{B,\lambda_1} \right)_m \right\rangle_{med}} \right)_n \right\rangle_{med}.$$

Mean of Ratios (*Mnor*)

The *Mnor* is the arithmetic mean of the pixel-by-pixel ratios of the pixel intensities (Channel 2 vs. Channel 1) that have the median background intensity subtracted[21]:

$$r_4 = \frac{1}{n} \sum_{i=1}^n \left(\frac{I_{P,\lambda_2} - \left\langle \left(I_{B,\lambda_2} \right)_m \right\rangle_{med}}{I_{P,\lambda_1} - \left\langle \left(I_{B,\lambda_1} \right)_m \right\rangle_{med}} \right)_i$$

Regression Ratio (*Rg*)

The *Rg* is a method that does not require rigidly defining background and foreground pixels. Every pixel within a circle of twice the diameter of the feature

is used. The relationship between Channel 1 and Channel 2 is determined by a linear regression between two channel intensities of pixels. The slope of the best fitting straight line is the regression ratio quantity r_5 for the given feature[21].

Feature (EST)	<i>Romd</i>	<i>Romn</i>	<i>Mdor</i>	<i>Mnor</i>	<i>Rg</i>
R30520	1.72	2.01	1.44	1.55	0.32
H76742	0.37	1.97	0.56	0.63	0.80
H37057	0.12	0.65	0.64	0.78	1.44
R64931	0.20	0.39	0.97	1.31	21.15
R83976	0.50	0.26	0.86	1.00	1.26
H37434	0.05	0.32	0.72	0.72	38.30
T45160	0.08	0.28	0.26	0.26	0.67

Table 2-1 Ratios calculated with five different methods

The ratio values calculated using five different methods for selected features from a real GPR file (accessing <http://www.ntu.edu.sg/sbs/staff/jml/simu.html> for GPR files). There are substantial differences in the ratio values calculated using the five methods. Thus choosing the most appropriate method in response to different data is of paramount importance.

For each feature there are five ratio quantities listed in the *Genepix Pro Report* file (GPR file). Before proceeding further microarray data analysis, a user has to decide which ratio quantity to be used. However, the software does not advice users on which ratio quantities to be chosen and the ratio quantities computed using different methods may vary substantially. Thus a frequently asked question by the cDNA microarray users is: Which computing method will confer the most reliable (accurate and precise) ratio quantity for a feature on a microarray slide? There is no answer for this question yet and a user has to select a method out of the five provided by the software at their own discretion. In Table 2-1 we have listed the five ratio quantities for the ESTs selected from an *Arabidopsis thaliana* cDNA microarray slide which compares the light regulated gene expression under two different light conditions, *i.e.* white light vs. wild type[22, 23]. The table clearly shows that there are substantial differences among the five ratio values.

In this study, we have implemented simulation approach to evaluate the five ratio computing methods. Our major objective is to provide microarray users, especially *GenePix Pro* users, with a general guideline on choosing an appropriate ratio quantity for a given feature during the cDNA microarray data processing.

2.2 Method

For a given feature the channel ratio can be computed (estimated) with different methods, but the optimal method is not known as the true channel ratio value is

inaccessible.

We were evaluating the five methods using simulation approach. The true values of the feature intensities and the background intensities have been assigned for both channels to a given spot. Then the true ratio can be calculated by

$$\text{True ratio} = (\text{intensity1-background1}) / (\text{intensity2-background2}).$$

To perform simulation study we need to assume that pixel intensities of a given feature follow certain probability distribution, so that we can generate simulated datasets based on these distributions. We examined the histograms of pixel intensities of features on microarray slides using *GenePix Pro Feature Viewer*, and we found that the majority of features have an approximate normal distribution of pixel intensities, and for some of the features the pixel intensities are more likely to follow a log-normal distribution or a mixture of normal distributions (see Figure 2-2 for examples). Since the distribution of pixel intensities depends on many factors in microarray experiment stage and image analysis stage, these three types of distributions may not cover all the possible cases.

Figure 2-2 Intensity distribution of pixels

Histograms of pixel intensities for three different features selected from an *Arabidopsis* cDNA microarray slide. The histograms were obtained using *GenePix Pro Feature Viewer*, and they are examples for three major types of pixel intensity distributions, i.e., **a.** normal distribution; **b.** log-normal distribution and **c.** a mixture of two normal distributions.

We firstly assigned true feature (foreground) intensities and background intensities to a given spot, and then we generated dataset of two channel pixel intensities for the spot under different distribution assumptions, namely normal distribution, log-normal distribution and a mixture of normal distributions, where the parameters of the distributions can be set or estimated with reference to the feature channel intensities and their standard deviations obtained from the output file of microarray image analysis software such as *GenePix Pro* and *QuantArray*. For the convenience of our simulation, we also assumed that the median of pixel background intensities for each channel equals to the true value of respective channel background intensity. Since the true values of the feature channel intensities and the background intensities have been assigned for a given spot, we can apply various simulated datasets generated under different conditions to the evaluation of the various ratio computing methods.

Simulation under the assumption of normally distributed pixel intensities

Assuming that pixel intensities of a given feature follow normal distribution, the mean of the pixel intensities should equal the true value of the feature intensity. We assigned the estimated feature (foreground) intensities, which we obtained from output files of image analysis software such as *GenePix Pro* and *QuantArray*, to μ_1 and μ_2 . Actually image analysis software use two methods to estimate the feature intensity, namely the mean of all pixel intensities or the median of all pixel intensities, and we only used the mean of pixel intensities to the value assignment

of μ_1 and μ_2 , and took μ_1 and μ_2 as the true feature intensities. We also assigned the estimated background intensities listed in the image analysis output to β_1 , β_2 , and took β_1 , β_2 as the true background intensities of the feature.

So the true ratio of the net signal intensity of channel 1 over that of channel 2 was given by:

$$r = \frac{\mu_1 - \beta_1}{\mu_2 - \beta_2} \quad (2.1)$$

The two channel pixel intensities follow normal distributions as assumed, where μ_1 and μ_2 are the means and the standard deviations σ_i ($i=1, 2$) were set to the estimated standard deviations of pixel intensities listed in image analysis output.

Without loss of generality, we assumed that each given feature consists of 100 pixels. In order to evaluate the effect of the variances (σ_1^2 and σ_2^2) of two channel pixel intensities we have also set the values of σ_1^2 and σ_2^2 at different levels instead of just fixing them at the variance level obtained in image analysis output. So the pixel intensities X_i ($i=1, 2$) follow a normal distribution with mean μ_i and variance σ_i^2 ($i=1, 2$). A dataset $S = \{(x_{1i}, x_{2i}) | i = 1, 2, \dots, 100\}$ has been generated using a computer program written in programming language C, which contains channel intensities of 100 pixels, where x_{1i} and x_{2i} were drawn randomly from the distribution of $X_1 \sim N(\mu_1, \sigma_1^2)$, and $X_2 \sim N(\mu_2, \sigma_2^2)$ respectively.

Since all the first four ratio computing methods took the median of pixel

background intensities as feature background intensity, the pixel background intensities of a feature were set to constants β_1 and β_2 in all three types of simulations.

Simulation under the assumption of log-normally distributed pixel intensities

Let X_i ($i=1,2$) be the random variables of pixel intensity distributions with mean μ_i and variance σ_i^2 ($i=1, 2$). We assumed that $\log(X_i)$ ($i=1, 2$) follows normal distributions with mean ν_i and variance γ_i^2 ($i=1, 2$). If we set the values of μ_i and σ_i ($i=1, 2$) respectively by the estimated feature intensity and the estimated standard deviation of pixel intensities, which can be found in the image analysis output, then we can estimate ν_i and variance γ_i^2 using formula (see http://www.riskglossary.com/link/lognormal_distribution.htm):

$$\nu_i = \log \frac{\mu_i^2}{\sqrt{\mu_i^2 + \sigma_i^2}} \quad (2.2)$$

and

$$\gamma_i^2 = \log \left[\left(\frac{\sigma_i}{\mu_i} \right)^2 + 1 \right]. \quad (2.3)$$

We took ν_i ($i=1, 2$) as true log-intensity, so the true feature intensity should be e^{ν_i} ($i=1, 2$). So the true ratio can be calculated using formula

$$r = \frac{e^{\nu_1} - \beta_1}{e^{\nu_2} - \beta_2}. \quad (2.4)$$

Simulated dataset $S = \{(x_{1i}, x_{2i}) | i = 1, 2, \dots, 100\}$ has been generated in following way: y_{1i} and y_{2i} ($i=1, 2, \dots, 100$) were firstly drawn randomly from the

distribution $\log(X_1) \sim N(\nu_1, \gamma_1^2)$ and $\log(X_2) \sim N(\nu_2, \gamma_2^2)$ respectively, then we got $x_{1i} = e^{y_{1i}}$ and $x_{2i} = e^{y_{2i}}$, where $i=1,2,\dots,100$.

Simulation under the assumption of a normal mixture distribution of pixel intensities

In this case, the existence of a small proportion (p_1 for channel 1, and p_2 for channel 2) of pixels with improbable signal intensities (outlier pixels) was allowed. Without the loss of the generality, the intensities of these outlier pixels were also assumed to follow certain normal distribution. Hence, we generated $S = \{(x_{1i}, x_{2i}) | i = 1, 2, \dots, 100\}$ from distributions $X_1 \sim N(\mu_1, \sigma_1^2)$ or $X_1^* \sim N(\Lambda_1, \Sigma_1^2)$, and $X_2 \sim N(\mu_2, \sigma_2^2)$ or $X_2^* \sim N(\Lambda_2, \Sigma_2^2)$, where X_k^* ($k=1,2$) denotes the signal intensity distribution of the outlier pixels with mean Λ_k and variance Σ_k^2 ($k=1,2$). To generate dataset S we have taken x_{1i} from distribution X_1 and X_1^* (with proportion $1-p_1$ and p_1 respectively), and x_{2i} from distribution X_2 or X_2^* (with proportion $1-p_2$ and p_2 respectively). The true ratio was still calculated using formula (2.1).

For various datasets that have been generated, the channel intensity ratio r_j ($j=1,2,3,4,5$) could be calculated using the above mentioned five computing methods. In this way, we were able to evaluate the five methods by studying the absolute difference d_j ($j=1, 2, 3, 4, 5$) between the logarithm of true ratio value and the logarithm of the ratio quantity obtained using each method:

$$d_j = |\ln(r_j) - \ln(r)|, \quad j=1, 2, 3, 4, 5. \quad (2.5)$$

The advantage of using the difference of two log-ratios (instead of using the difference of two ratios) to define the error d_j ($j=1, 2, \dots, 5$) is that this provides a more balanced consideration of two channel intensities. In our study each simulation were repeated 100 times, so in the following text we actually used d_j ($j=1, 2, \dots, 5$) to denote the average of the 100 absolute differences, and we call this the *error* of the respective ratio computing method at the given feature.

2.3 Results

Simulation 1: Assuming normal distribution of pixel intensities and assigning the variances to different levels

To generate simulated datasets S , we set the mean μ_i ($i=1,2$) of pixel intensities at different levels, and for fixed pairs of μ_1 and μ_2 , we found that, with the accretion of the respective variance (σ_i^2 , $i=1, 2$), the error d_4 of *Mnor* is always much greater than the error (d_j , $j = 1,2,3,5$) of any other methods (see Figure 2-3a). Moreover, the error d_5 of *Rg* was significantly greater than d_1 of *Romn*, d_2 of *Romd* and d_3 of *Mdor* for most of the datasets that have been generated in our simulation study (see Figure 2-3a for an example), even though *Rg* also appeared to be the best method (d_5 is the smallest among the five errors (d_j , $j=1, 2, 3, 4, 5$)) for a small proportion of the datasets. Thus we have decided to focus on the

comparison of *Romn*, *Romd* and *Mdor* with a higher resolution (Figure 2-4a).

When the variance of the pixel intensities were small, d_2 was smaller than d_1 and d_3 , so *Romn* was the superior method. However, when the variance increased gradually, *Romd* and *Mdor* became the preferable methods (see Figure 2-4a).

a **b**

Figure 2-3 Results of simulation 1 and simulation 2

Comparing all the five ratio computing methods by setting the variances of the pixel intensities or the log pixel intensities at different levels. **a.** assuming normal distribution of pixel intensities. $\mu_1 = 3000$, $\beta_1 = 100$, $\mu_2 = 8000$, $\beta_2 = 100$, σ_2 took value 2000, 3000, 4000, 5000, and 6000 respectively, and σ_1 was set at 500; **b.** assuming log normal distribution of pixel intensities, $\nu_1 = 7.3$, $\beta_1 = 70$, $\nu_2 = 7.9$, $\beta_2 = 250$, γ_2 took value 0.23, 0.32, 0.41, 0.53, 0.61 respectively and γ_1 was set at 0.18.

a

b

Figure 2-4 High resolution results of simulation 1 and simulation 2

Comparing three ratio computing methods (*Romn*, *Romd* and *Mdor*) by setting the variances of the pixel intensities or the log pixel intensities at different levels. **a.** assuming normal distribution of pixel intensities, for each simulation $\mu_1 = 3000$, $\beta_1 = 100$, $\mu_2 = 8000$, $\beta_2 = 100$, σ_2 took value 2000, 3000, 4000, 5000 and 6000 respectively, and σ_1 was set at 500(**a1**), 750(**a2**), 1000(**a3**), 1250(**a4**) respectively. **b.** assuming log normal distribution of pixel intensities, $\nu_1 = 7.3$, $\beta_1 = 70$, $\nu_2 = 7.9$, $\beta_2 = 250$, γ_2 took value 0.23, 0.32, 0.41, 0.53, 0.61 respectively and γ_1 was set at 0.18 (**b1**), 0.29(**b2**), 0.40(**b3**), 0.50(**b4**) respectively.

Simulation 2: Assuming log-normal distribution of pixel intensities and assigning the variances to different levels

Based on the values of μ_i and σ_i^2 ($i=1, 2$) assigned in the same way as described in *simulation 1* we estimated the values of v_i and γ_i^2 ($i=1,2$) using formulae (2.2) and (2.3), and the true ratio was calculated using formula (2.4). The results were similar to that of *Simulation 1*. For the datasets generated under the assumption of log-normal distribution of pixel intensities, the errors d_4 and d_5 are much greater than the errors ($d_j, j=1,2,3$) of the other three methods (see Figure 2-3b), and the error d_5 of *Rg* was the largest among the five (see Figure 2-3b). Further comparison of *Romn*, *Romd* and *Mdor* (Figure 2-4b) suggested that when the variance of log pixel intensities were relatively small for both channels, d_2 was smaller than d_1 and d_3 , so *Romn* was the superior method (Figure 2-4b1, Figure 2-4b2). However, when the variance of log pixel intensities in one channel or in the both two channels became large enough, *Romd* and *Mdor* became the preferable methods (see Figure 2-4b3, Figure 2-4b4).

Simulation 3: Assuming a mixture of normal pixel intensity distributions and assigning the outlier percentages to different levels

If there were a small proportion of outliers (pixels with extremely large or small intensities) for a given feature, our simulation suggested that there were substantial advantages for the median based methods *Romd* and *Mdor* (see Figure 2-5),

since the mean statistics could be easily biased due to the existence of the outlier pixels. Figure 2-5 also suggested that the errors d_4 of *Mnor* and d_5 of *Rg* were greater than the errors of the other three methods, and the error d_2 of *Romn* was moderate.

Figure 2-5 Results of simulation 3

Comparing the five ratio computing methods for different percentages of outlier pixels under the assumption of a mixture of normal distributions of pixel intensities. In our simulation $\mu_1 = 1700$, $\sigma_1 = 300$, $\mu_2 = 3000$, $\sigma_2 = 700$ for regular pixels, and $\Lambda_1 = 4000$, $\Sigma_1 = 2000$, $\Lambda_2 = 7000$, $\Sigma_2 = 3000$ for outlier pixels, $p_2 = 10\%$, and p_1 took value 4%, 7%, 10%, 13%, 17%, 20% respectively.

Simulation 4: Evaluating the five ratio computing methods in terms of all features on microarray slides

On a typical microarray slide there are thousands of features, the magnitudes of the pixel intensities and the variances of the pixel intensities are generally of great differences between two features. For example, in the *GenePix Pro* output of one of our microarray slides with 9216 features, the estimated feature intensities ranges from 138 to 62541 with mean 4807 for the channel at wavelength 635nm, and from 115 to 52334 with mean 5066 for the channel at wavelength 532nm respectively, and the estimated standard deviation of pixel intensities ranged from 41 to 22613 with mean 1852 for the channel 635nm, and from 23 to 25469 with mean 2005 for the channel 635nm respectively. In order to cover all possibilities and make our conclusions as general as possible, we carried out simulations for nearly all the spots on each of the eight microarray slides used in this study. The eight slides can be divided into two groups. The first group consists of four *Arabidopsis thaliana* cDNA microarray slides that were kindly provided by Deng's Laboratory from the Department of Molecular, Cellular and Developmental Biology, Yale University[22, 23], and these four slides were scanned and processed by *Axon GenePix* scanner and *GenePix Pro* image analysis software respectively, with the output file in GPR format. The second group consists of four mouse cDNA microarray slides obtained from Yale Keck Microarray Facility, and these four slides were scanned using *GSI Lumonics (Packard/Perkin Elmer) ScanArray*

platform and images were processed using *QuantArray* package. On each of these microarray slides there were 9216 features. A small proportion of the spots on each slide have been eliminated due to the negative net channel intensities. Our simulations were done under two different assumptions: 1). normal distributions of pixel intensities for both channels; 2). log-normal distributions of pixel intensities for both channels. For each feature on a microarray slide, we could extract information such as estimated feature (foreground) intensities, estimated background intensities and the estimated variances of pixel intensities from image analysis output. We assigned these quantities to μ_1 , μ_2 , β_1 , β_2 , σ_1^2 , σ_2^2 , and generated simulated dataset S for each feature as described in *simulation 1* under the normal assumption. We estimated v_i and γ_i^2 ($i=1,2$) using formulae (2.2) and (2.3), and then generated simulated dataset S for each feature as described in *simulation 2* under the log-normal assumption.

In order to compare the five ratio computing methods we have calculated the error d_j ($j=1, 2, \dots$) using simulated datasets generated for each feature.. The five error quantities were ranked in ascending order to indicate the most to the least favorable methods, for example, the first represents the best method and the fifth represents the worst method. Table 2-2a and Table 2-2b summarized the results of simulations under normal assumption, whereas 2-2c and Table 2-2d listed the results of simulations under log-normal assumption. Table 2-2a and Table 2-2c display the distribution of the best methods whereas Table 2-2b and Table 2-2d

display the distribution of the worst methods under two different assumptions. Under both assumptions of normal and log-normal pixel intensity distributions, *Romn* appears to be the best method in most cases (Table 2-2a and Table 2-2c) on all eight microarray slides, and it has very little chance to be the worst method (Table 2-2b and Table 2-2d). Although *Rg* is relatively good as it ranked first for a substantial proportion of features on the eight arrays (Table 2-2a and Table 2-2c), the chances of it to turn up to be the worst method are also very high (Table 2-2b and Table 2-2d). Under both assumptions of normal and log-normal pixel intensity distributions, *Mnor* has large chance to be the worst method (Table 2-2b and Table 2-2d). Therefore, we advise users to be cautious on using this *Rg* and *Mnor* methods. Our simulation suggested that *Romn*, *Mdor* or *Rg* ranked first for the most of the features on the eight microarray slides.

Comparison of Ration Calculating Methods

a

slide	<i>Romn</i>	<i>Romd</i>	<i>Mnor</i>	<i>Mdor</i>	<i>Rg</i>	Total
GP1	8369(92.7%)	4(0.0%)	4(0.0%)	131(1.5%)	522(5.8%)	9030
GP2	5390(62.2%)	3(0.0%)	24(0.3%)	133(1.5%)	3114(36.0%)	8664
GP3	7918(88.8%)	86(1%)	431(4.8%)	193(2.2%)	293(3.3%)	8921
GP4	5604(64.8%)	20(0.2%)	37(0.4%)	577(6.7%)	2423(28.0%)	8661
QA1	6960(75.7%)	5(0.0%)	7(0.1%)	3(0.0%)	2223(24.2%)	9198
QA2	7491(81.5%)	12(0.1%)	0(0.0%)	1(0.0%)	1686(18.3%)	9190
QA3	7498(81.6%)	12(0.1%)	1(0.0%)	0(0.0%)	1679(18.3%)	9190
QA4	7488(81.5%)	12(0.1%)	0(0.0%)	0(0.0%)	1690(18.4%)	9190

b

slide	<i>Romn</i>	<i>Romd</i>	<i>Mnor</i>	<i>Mdor</i>	<i>Rg</i>	Total
GP1	4(0.0%)	12(0.1%)	8983(99.5%)	7(0.1%)	24(0.3%)	9030
GP2	11(0.1%)	88(1%)	8269(95.4%)	10(0.1%)	284(3.3%)	8664
GP3	4(0.0%)	140(1.6%)	1752(19.6%)	20(0.2%)	7005(78.5%)	8921
GP4	10(0.1%)	51(0.6%)	8159(94.2%)	7(0.1%)	434(5%)	8661
QA1	0(0.0%)	450(4.9%)	3380(36.7%)	15(0.2%)	5353(58.2%)	9198
QA2	0(0.0%)	468(5.1%)	3100(33.7%)	0(0.0%)	5622(61.2%)	9190
QA3	0(0.0%)	471(5.1%)	3099(33.7%)	0(0.0%)	5620(61.2%)	9190
QA4	0(0.0%)	462(5.0%)	3109(33.8%)	0(0.0%)	5619(61.2%)	9190

Comparison of Ration Calculating Methods

c

slide	<i>romn</i>	<i>romd</i>	<i>mnor</i>	<i>mdor</i>	<i>rg</i>	Total
GP1	8304(92.0%)	23(0.3%)	14(0.2%)	112(1.2%)	569(6.3%)	9022
GP2	4332(50.3%)	46(0.5%)	35(0.4%)	517(6.0%)	3690(42.8%)	8620
GP3	6379(71.8%)	1309(14.7%)	419(4.7%)	412(4.6%)	365(4.1%)	8884
GP4	4730(55%)	347(4.0%)	59(0.7%)	905(10.5%)	2561(29.8%)	8602
QA1	6600(71.8%)	22(0.2%)	6(0.1%)	405(4.4%)	2165(23.5%)	9198
QA2	6640(72.3%)	27(0.3%)	0(0.0%)	702(7.6%)	1819(19.8%)	9188
QA3	6640(72.2%)	28(0.3%)	1(0.0%)	713(7.8%)	1810(19.7%)	9192
QA4	6655(72.4%)	21(0.2%)	1(0.0%)	710(7.7%)	1803(19.6%)	9190

d

Slide	<i>Romn</i>	<i>Romd</i>	<i>Mnor</i>	<i>Mdor</i>	<i>Rg</i>	Total
GP1	4(0.0%)	18(0.2%)	7359(81.6%)	36(0.4%)	1605(17.8%)	9022
GP2	14(0.2%)	46(0.5%)	7808(90.6%)	50(0.6%)	702(8.1%)	8620
GP3	6(0.0%)	117(1.3%)	1149(12.9%)	40(5.0%)	7572(85.2%)	8884
GP4	5(0.1%)	51(0.6%)	7227(84%)	44(0.5%)	1275(14.8%)	8602
QA1	10(0.1%)	102(1.1%)	3305(35.9%)	22(0.2%)	5759(62.6%)	9198
QA2	8(0.1%)	71(0.8%)	2828(30.8%)	2(0.0%)	6279(68.3%)	9188
QA3	7(0.0%)	71(0.8%)	2827(30.8%)	1(0.0%)	6286(68.4%)	9192
QA4	8(0.1%)	73((0.8%)	2826(30.8%)	2(0.0%)	6281(68.3%)	9190

Table 2-2 Results of simulation 4

On each microarray slide there are 9216 features. Some features were eliminated during the data quality control process (see supplement webpage for GPR files and *QuantArray* files used here); **a.** the number (percentage) of spots of each method being the best among the five (Normal distributions of pixel intensities assumed); **b.** the number (percentage) of spots of each method being the worst among the five (Normal distributions of pixel intensities assumed); **c.** the number (percentage) of spots of each method being the best among the five (Log-normal distributions of

pixel intensities assumed); **d.** the number (percentage) of spots of each method being the worst among the five (Log-normal distributions of pixel intensities assumed).

Figure 2-6 is a scatter plot that reveals the comparison of *Romn*, *Mdor* and *Rg* in terms of the estimated two channel intensities (obtained from image analysis output files) of the features on the slides. It is a plot of the estimated Channel 1 intensity versus estimated Channel 2 intensity. Only the features that conferred one of the three methods ranked first were disclosed in the scatter plots. As is apparent from this figure, when the intensities of the two channels are over 10000, *Romn* is the best method for nearly all the features on the slide.

We defined coefficient of variance (CV) of feature pixel intensities as their standard deviation divided by the mean. In practice we can calculate CV using estimated standard deviation of feature pixel intensities divided by the estimated feature intensity(the average of pixel intensities), and for a given feature on a slide, both these two quantities can be found from the output file of image analysis software such as *GenePix Pro* and *QuantArray*. Figure 2-7 is a scatter plot that demonstrates the comparisons of *Romn*, *Mdor* and *Romd* in terms of CVs of the two channel pixel intensities of the features that gave rise to one of the three methods ranked first. As seen in Figure 2-7, when *Romd* is the best method for a given feature, the CVs of both of the channel intensities are most likely to be greater than 0.5. This suggests the advantage of *Romd* when the CVs of the pixel intensities are large.

Figure 2-6 Impact of intensity of spot on the choice of ratio calculating method

Scatter plots that reveal the comparisons of *Romn*, *Mdor* and *Rg* in terms of the two channel intensities of the features on the slide. Only the features that conferred *Romn*, *Mdor* or *Rg* ranking first were disclosed respectively in the scatter plot **a**, **b** or **c**. When the intensities of the two channels are over 10000, *Romn* is the most favorable method for nearly all the features on the slide.

Figure 2-7 Impact of CV of pixels on the choice of ratio calculating method

Scatter plots that demonstrate the comparisons of *Romn*, *Mdor* and *Rg* in terms of the CVs (Coefficient of Variances, which is the standard variation divided by the mean) of the two channel pixel intensities of the features on the slide. Only the features that gave rise to *Romn*, *Mdor* or *Rg* ranking first were disclosed respectively in the scatter plot **a**, **b** or **c**. When *Romd* is the best method, the CVs of both channel intensities are most likely to be greater than 0.5. This suggests the advantage of *Romd* when the CVs of the pixel intensities are large.

Accuracy and Precision

To evaluate the accuracy of each method we took average of errors of all features on four microarray slides (see Figure 2-8). Two of these slides were processed using *GenePix Pro*, and the other two were processed using *QuantArray* package. The results suggested that in average *Mnor* and *Rg* had much poorer accuracy comparing with the other three methods. *Romn*, *Romd* and *Mdor* are comparable in average accuracy, and *Romn* had slightly higher accuracy than the other two. The results were consistent for all the four slides under the assumptions of both normal distribution (Figure 2-8a) and log-normal distributions (Figure 2-8b) of pixel intensities.

a

b

Figure 2-8 Accuracy of different methods

Mean errors of each method for all features on microarray slides. Four microarray slides were used in the study. **a.** under the assumption of normal distributions of pixel intensities; **b.** under the assumption of log-normal distributions of pixel intensities.

a

b

Figure 2-9 Precision of different methods

Variances of errors (d_i) of 100 repeated simulations for 4 randomly given spots, $i=1,2,3,4, 5$; **a.** assuming normal distributions of pixel intensities, $\beta_1=100$, $\beta_2=100$, the values set to the four spots are $\mu_1=3000$, $\sigma_1=500$, $\mu_2=8000$, $\sigma_2=2000$ for Spot 1; $\mu_1=1000$, $\sigma_1=300$, $\mu_2=5000$, $\sigma_2=2000$ for Spot 2; $\mu_1=3000$, $\sigma_1=500$, $\mu_2=2000$, $\sigma_2=600$ for Spot 3 and $\mu_1=5000$, $\sigma_1=2000$,

$\mu_2=4000$, $\sigma_2=1000$ for Spot 4; **b.** assuming log-normal distributions of pixel intensities, $\beta_1=70$, $\beta_2=300$, the values set to the four spots are $v_1=7.42$, $\gamma_1=0.18$, $v_2=7.98$, $\gamma_2=0.23$ for Spot 1; $v_1=8.96$, $\gamma_1=0.25$, $v_2=9.87$, $\gamma_2=0.25$ for Spot 2; $v_1=8.14$, $\gamma_1=0.23$, $v_2=8.66$, $\gamma_2=0.28$ and $v_1=6.65$, $\gamma_1=0.25$, $v_2=7.75$, $\gamma_2=0.34$ for Spot 4.

a

b

Figure 2-10 Precision of different methods (CV)

CVs (Coefficient of Variance, defined as standard deviation divided by the mean) of errors (d_i) of 100 repeated simulations for four randomly given spots, $i=1, 2, 3, 4, 5$; **a.** assuming normal distributions of pixel intensities, and the values set to the four spots are the same as that in Figure 2-9a; **b.** assuming log-normal distributions of pixel intensities, and the values set to the four spots are the same as that in Figure 2-9b.

We did simulation study for four randomly given spots to evaluate the precision of each method, each simulation repeated 100 times, and the average of the 100 ratio values and their standard deviation, as well as the true ratio were displayed in Figure 2-9, and we found again that *Rg* and *Mnor* had relative low accuracy under the assumptions of both normal (Figure 2-9a) and log-normal (Figure 2-9b) distributions of pixel intensities. *Rg* method had the highest precision (the lowest standard deviation of 100 ratios calculated for each spot), whereas the rest of methods are comparable in precision. In order to further compare the different methods with regard to their precision, we displayed in Figure 2-10 the CV of 100 ratios calculated for each method in our simulation study, where CV (coefficient of variance) were defined as the standard deviation of the 100 ratios divided by the mean of these ratio values. Under both normal assumption and log-normal assumption we found that *Rg* method had significantly higher precision than the other four methods, and *Romn* had the highest precision among the rest four methods (Figure 2-10). Under the normal assumption, *Mnor* had poorer precision than other four methods (Figure 2-10b).

2.4 Discussion

In DNA microarray image analysis, the channel ratios could be calculated by different methods for a given feature. These ratio quantities could be of great differences for some features on a microarray slide (see Table 2-1). However,

image analysis software such as *Axon GenePix Pro* and *QuantArray* provide no advice on which ratio quantity to be used for a given feature. In order to deal with this ambiguity, we have used simulation approaches to evaluate the five ratio computing methods in *Axon GenePix Pro* software. Current methods used to the ratio calculation of microarray slides can be roughly divided to three subgroups: those based on quantities derived from whole features (such as median and mean pixel intensities), those based on pixel-by-pixel ratios of intensities, and those based on regression of pixel intensities of two channels. The five methods used by *GenePix Pro* actually provide representative for each of the three subgroups. Based on the simulation studies, we have worked out a rough guidance for cDNA microarray users to choose an appropriate ratio computing method when using image analysis software. Although the simulations here were carried out for the output files (GPR file) of the microarray image processing software *Axon GenePix Pro* and *QuantArray*, the conclusions drawn from this study can be extended to the microarray data processed by other software, because most image processing software calculate the channel ratio quantities in terms of the intensities of foreground and background pixels that define a spot, so the selection of appropriate ratio computing method depends actually on the distribution of pixel intensities. In theory, when a spot has a clear boundary and the hybridization is uniform, the pixel intensities of the feature would be approximately normally distributed[24]. But in practical situations, the distributions of the pixel intensities of some features might be skewed due to various experimental or instrumental

factors, for example, due to a small proportion of the outliers occurred during the slide printing process or hybridization process. We examined the histogram of pixel intensity distribution of feature on microarray slides, and found that the majority of features have an approximate normal distribution of pixel intensities, and for some of the features the pixel intensities are more likely to follow a log-normal distribution or a mixture of normal distributions (see Figure 2-2 for examples). Therefore our simulation studies take into consideration of these three cases. Our simulation study was conducted in two different ways. Firstly (*Simulation 1*, *Simulation 2* and *Simulation 3*), fixed values of the means and variances of the pixel intensities for a feature have been assigned to different levels solely in terms of our observation of the real microarray datasets. The background intensity for each channel was set to a fixed value within a reasonable range. One of the advantages of this type of simulation is that the value of the parameters in the distribution could be assigned according to our intention. However, the coverage of the simulation study using this approach is inadequate. Secondly (*Simulation 4*), from the real microarray data (*Axon GenePix Pro* or *QuantArray* output files) we have extracted the information about estimated feature/background intensities and the variances of the pixel intensities of both channels for each spots on the slide. Based on this information we set parameter values to generate simulated datasets under different distribution assumptions. The data generated using the second approach have a much more extensive coverage of real situations.

It is obvious that none of the ratio computing methods is universally optimal for all features on a microarray slide. For different features we have to select different ratio computing methods. Which method to be chosen for a given feature, depends on the intensity distributions of feature foreground and background pixels. The guidance suggested here might not be perfect and complete since there were thousands of features and they would form thousands of different possibilities. Thus it was very difficult for us to cover every case in our simulation study. Nevertheless we believe that it will serve its ultimate aim by facilitating users in selecting the appropriate methods for each feature within the pool of information in the output file of microarray image processing software. Last but not least, our simulation studies have been carried out for the *Axon GenePix Pro* and *QuantArray* output files, but the results could also be extended to the output files of other microarray image analysis software.

2.5 Conclusion

Based on the simulation studies we recommend *Romn* as the prime cDNA microarray ratio computing method, especially if the CVs of two channel pixel intensities of a given feature are less than 0.5. However, if the CVs of pixel intensities are much greater than 0.5, *Romd* is recommended. Moreover, we also advise cDNA microarray users to be cautious on using *Mnor* and *Rg*.

Chapter 3 Gene Expression Profiling

By measuring transcription levels of genes using microarrays in cells under various conditions, at different developmental stages and in different tissues, we can build a profile which characterizes the expression pattern of each gene in the genome. Here, the data from microarray experiments are usually in the form of large matrices of gene expression levels under different experimental conditions, and we call the analysis of this kind of data gene expression profiling.

3.1 Clustering

After we have processed the microarray raw data into gene expression profiles, the next task is to extract useful information about the underlying biological processes from the profile data. Clustering analysis is a major approach in gene expression profile analysis. A clustering algorithm groups genes or samples together by certain similar properties. Clustering in terms of microarray gene expression profile can be conducted for either grouping genes or classifying samples. The former is called gene based clustering in which genes are put into different partition based on their expression patterns. The latter is called sample based clustering in which samples can be partitioned into homogeneous groups. Clustering analysis of gene expression profile data facilitates greatly our understanding on gene functions, gene regulations, cellular processes and different subtypes of cells.

Clustering algorithms group the objects with similar property, the objects in the same group have the similar property and the objects in different group have dissimilar property. So how to measure the similarity (or dissimilarity) level of two different objects is critical in clustering algorithms.

Euclidean distance is one of the most commonly used methods to measure the dissimilarity of two data objects. It is defined as

$$Euclidean(O_i, O_j) = \sqrt{\sum_{d=1}^p (o_{id} - o_{jd})^2} \quad (3.1)$$

Where O_i and O_j are two data objects that we want to measure their distance.

We can use vectors $\vec{O}_i = \{o_{ij} \mid 1 \leq j \leq p\}$ ($i=1, 2$) to represent the data objects, where the o_{ij} means the j -th element in i -th data object and p is the total number of elements in a data object.

Pearson's correlation coefficient is another popular similarity measure. It is defined as

$$pearson(O_i, O_j) = \frac{\sum_{d=1}^p (o_{id} - \mu_{oi})(o_{jd} - \mu_{oj})}{\sqrt{\sum_{d=1}^p (o_{id} - \mu_{oi})^2} \sqrt{\sum_{d=1}^p (o_{jd} - \mu_{oj})^2}} \quad (3.2)$$

Where μ_{oi} and μ_{oj} are the means of $\vec{O}_i$ and $\vec{O}_j$ respectively.

Most of the clustering algorithms use either *Euclidean* distance or *Pearson's*

correlation coefficient to measure dissimilarity/similarity between two objects.

There are a couple of clustering methods available for dealing with gene expression profile data generated from DNA microarray data. Among them *K-Means* and *Hierarchical Clustering* are two of the most frequently used methods.

3.1.1 K-Means

K-Means is one of the simplest unsupervised learning algorithms that solve clustering problems. The procedure follows a simple and easy way to classify a given data set through a certain number of clusters (assume k clusters) fixed in advance. The main idea is to define k centroids, one for each cluster. These centroids should be placed in a proper way because of different location causes different result. So, a good choice is to place them as much as possible far away from each other. The next step is to take each point belonging to a given data set and associate it to the nearest centroid. When no point is pending, the first step is completed and an early group is done. At this point we need to re-calculate k new centroids according to the clusters resulting from the previous step. After we have these k new centroids, we reassign the points to new groups the same as first step. A loop has been generated. As a result of this loop, k centroids do not change their location step by step until no more changes are done. In other words centroids do not move any more.

This algorithm aims at minimizing an *objective function*, in this case a squared error function. The objective function

$$J = \sum_{j=1}^k \sum_{i=1}^n \|x_i^{(j)} - c_j\|^2, \quad (3.3)$$

where $\|x_i^{(j)} - c_j\|^2$ is a chosen distance measure between a data point $x_i^{(j)}$ and the cluster center c_j , is an indicator of the distance of the n data points from their respective cluster centers.

3.1.2 Hierarchical clustering

In contrast to partition-based clustering, which attempts to directly decompose the data set into a set of disjoint clusters, hierarchical clustering generates a hierarchical series of nested clusters which can be graphically represented by a tree, called *dendrogram*. The branches of a dendrogram not only record the formation of the clusters but also indicate the similarity between the clusters. By cutting the dendrogram at some level, we can obtain a specified number of clusters. By reordering the objects such that the branches of the corresponding dendrogram do not cross, the data set can be arranged with similar objects placed together.

Hierarchical clustering algorithms can be further divided into *agglomerative* approaches and *divisive* approaches based on how the hierarchical dendrogram is formed. Agglomerative algorithms (bottom-up approach) initially regard each data object as an individual cluster, and at each step, merge the closest pair of clusters until all the groups are merged into one cluster. Divisive algorithms (top-down

approach) starts with one cluster containing all the data objects, and at each step split a cluster until only singleton clusters of individual objects remain.

Eisen et al.[25] applied an agglomerative algorithm called UPGMA (Unweighted Pair Group Method with Arithmetic Mean) and adopted a method to graphically represent the clustered data set. In this method, each cell of the gene expression matrix is colored on the basis of the measured fluorescence ratio, and the rows of the matrix are re-ordered based on the hierarchical dendrogram structure and a consistent node-ordering rule. After clustering, the original gene expression matrix is represented by a colored table (a *cluster image*) where large contiguous patches of color represent groups of genes that share similar expression patterns over multiple conditions.

3.2 Microarray gene expression profiling and DNA binding motif finding

Gene expression profile data derived from DNA microarray has opened new opportunities to us in the identification of co-expressed genes. Clustering analysis of gene expression profiles could be used to detect and group those co-expressed genes which have similar expression patterns under various conditions. We can postulate that these co-expressed genes are co-regulated by common regulatory proteins. A major challenge to the computational biologist is to define novel

regulatory elements (motifs) in such sets of co-expressed (putatively co-regulated) genes. Co-regulated genes are known to share some similarities in their regulatory mechanism at the transcriptional level. So they might have consensus DNA motifs in their promoter region, and the motifs could be recognized by the same regulatory proteins. It is well established that DNA motifs located upstream of the transcriptional start site are important in facilitating the binding of regulatory proteins that control the transcription of genes[26, 27]. Genes have similar function are usually controlled by same transcription factor, therefore they may have common motif in their upstream sequence. So the information about DNA binding motifs in the upstream regions of co-regulated genes is useful for the investigation of transcription control of these genes.

In addition to providing interim data about co-expressed genes for DNA motif finding, DNA microarray can also be used to develop techniques to find binding sites motifs directly. For example, ChIP-chip becomes a widely used method to find binding motifs in gene promoter regions. The first “ChIP” stands for chromatin immuno-precipitation, and the second “chip” refers to microarray (DNA chips). In chapter 4 we briefly review the current methods of computational motif finding, and we also provide a short introduction of ChIP-chip experiment. However, the identification of transcription factor binding sites with experimental methods is a time consuming, labor intensive and expensive work, so it is valuable to develop novel computational motif finding method. In chapter 5 we proposed a

method of finding transcriptional factor binding motifs for co-regulated genes using promoter region sequences of multiple closely related species.

Chapter 4 Finding DNA Binding Motifs

4.1 Transcriptional Factors and DNA Binding motifs

Understanding the mechanisms that regulate the gene expression in eukaryotes is a major challenge in modern molecular biology. To a large extent, gene regulation is the control of transcription, and it is accomplished by a number of regulatory proteins called transcription factors binding to specific sites on DNA. These DNA binding sites contain about 6-25 base pairs and are usually located in the upstream regions of functionally related genes. These sites are important in facilitating the binding of regulatory proteins that control the transcription of these genes. Many transcription factors are able to bind to specific sets of short conserved sequences, whose consensus are called DNA binding motifs. Recognition of transcriptional factor binding motifs in the upstream regions of co-expressed (co-regulated) genes is crucial for elucidating gene regulatory networks.

4.2 Representation of DNA Motifs

There are two common ways to represent motifs: consensus sequence and position specific scoring matrix (PSSM).

4.2.1 Consensus sequence

The consensus sequence describes a set of oligonucleotides with a single oligonucleotide in which each position is the most frequent nucleotide among that set of oligonucleotides. However, sometimes some positions do not show a definite preference to a certain nucleotide, but admit more than one base or even any base. Thus, instead of using a single nucleotide in one position to describe the consensus sequence, another strategy is adopted which use ambiguous symbols in each position to incorporate the different alternatives in the description of sequence. These ambiguous symbols are called IUPAC code (Table 4-1). This method allows a maximum number of substitutions for each position and it is used to describe binding site consensus in transcriptional factor databases such as TRANSFAC[28] and SCPD[29]. For example, in consensus DNA sequence ARNV, the first base is always A; the second base can be A or G; the third base can be any nucleotide; and the fourth base can be A, C or G.

The shortcoming of using consensus sequence is that it does not provide the relative nucleotide frequency at each base position. Sequence logo (Figure 4-1) is an alternative method of graphically representing a motif. It uses a graph to represent the nucleotide frequency at each position, in which the size of the symbol is related to the frequency of a nucleotide occurs at a certain position.

code	Description
A	Adenine
C	Cytosine
G	Guanine
T	Thymine
U	Uracil
R	Purine (A or G)
Y	Pyrimidine (C, T, or U)
M	C or A
K	T, U, or G
W	T, U, or A
S	C or G
B	C, T, U, or G (not A)
D	A, T, U, or G (not C)
H	A, T, U, or C (not G)
V	A, C, or G (not T, not U)
N	Any base (A, C, G, T, or U)

Table 4-1 IUPAC code

Figure 4-1 A sequence logo

Sequence logo gives a graphic representation of a motif. It displays the nucleotide frequency at each position, in which the size of the symbol is related to the frequency of a nucleotide occurs at the position.

4.2.2 PSSM

Position Specific Scoring Matrix (PSSM), also called position specific weight matrix (PSWM) or Position Weight Matrix (PWM) is a more flexible representation of motifs. PSSM can be expressed in different ways. In the simplest way, a PSSM is obtained by aligning the TFBS (transcriptional factor binding sites) and computing the nucleotide frequency at each position of the alignment [30].

The result is a $4 \times n$ matrix, where n is the length of the motif, and the sum of each column of the matrix equals to 1. In this way, the preference of each position to different alternative nucleotides can be expressed by nucleotide probability in the column. A PSSM score is defined as $\prod_{j=1}^N m_{i(j),j}$, where j represents position in the aligned oligonucleotides, $i(j)$ is the symbol at position j in the aligned oligonucleotides, and $m_{i,j}$ is the score in row i , column j of the matrix. In other words, a PSSM score is the product of position-specific scores for each symbol in the aligned oligonucleotides. Most commonly, each column defines log-likelihoods for each of the different symbols, thus PSSM score is defined as

the column. A PSSM score is defined as $\prod_{j=1}^N m_{i(j),j}$, where j represents position in the aligned oligonucleotides, $i(j)$ is the symbol at position j in the aligned oligonucleotides, and $m_{i,j}$ is the score in row i , column j of the matrix. In other words, a PSSM score is the product of position-specific scores for each symbol in the aligned oligonucleotides. Most commonly, each column defines log-likelihoods for each of the different symbols, thus PSSM score is defined as

the aligned oligonucleotides. Most commonly, each column defines log-likelihoods for each of the different symbols, thus PSSM score is defined as

$$\sum_{j=1}^N m_{i(j),j}.$$

Instead of using log-likelihood values in the PSSM, as described in the previous paragraph, several methods uses log-odds scores in the PSSMs. An element in a PSSM is then calculated as $m_{i,j} = \log(p_{i,j} / b_i)$, where $p_{i,j}$ is the probability of observing symbol i at position j of the motif, and b_i is the probability of observing

the symbol i in a background model. The PSSM score then corresponds to the log-odds of the substring being generated by the motif versus being generated by the background, in a generative model of the sequence.

4.3 Experimental method for identifying transcription factor binding sites

A lot of known TFBSs have been found by traditional experimental methods. Microarray based readout of chromatin immuno-precipitation assays (ChIP-chip), is currently the most widely used method for identifying transcription factor binding sites in vivo[31, 32]. In a ChIP-chip experiment a known transcription regulator is tagged with an antibody epitope, and the tagged regulator is expressed in a suitable system where it binds to DNA, either directly or via other proteins. The complex is then chemically crosslinked, the DNA is fragmented, and the protein-DNA complex is isolated by immunoprecipitation. The genomic position of the DNA fragment is then identified by a microarray experiment. This gives the location of binding sites for this specific regulator[33]. It has already been used to find many transcription factors in the yeast *Saccharomyces cerevisiae*[34-36] and in mammalian cells[37-39]. Since identification of transcription factor binding sites with experimental method is a time consuming and expensive work, more and more computational methods are developed recently, such as *AlignACE*[40], *MEME*[41], *Consensus*[42], *MotifSampler*[43], *YMF*[44], *GLAM*[45], etc.

4.4 Computational methods of finding DNA binding motifs

Many algorithms to identify transcription factor binding sites in non-coding regions of DNA have been developed since the computational approaches became available to biology. Recently, applying computational methods to this area has become more and more popular with the development of microarray technique. With the power of gene clustering data from microarrays, now we can investigate transcriptional regulatory networks in the context of the whole genome, and finding of transcriptional factor binding sites in the regulatory regions of co-expressed genes is a crucial step of in this process.

The main challenge of this problem always has been the development of effective algorithm that can treat all the intrinsic complexities associated with the nature of binding sites. The difficulty mainly arises from the nature of the binding site. The base sequence of most binding sites is often called a regulatory motif, and as we can easily infer from this name, it is a tiny, highly variable sequence. What makes it much worse to identify is that they usually have gaps in them.

Now the DNA motif finding with computational methods is a popular field in modern biology, there are many different methods that have been published. Several computational methods have been introduced in the literature[46-48]. These methods can be generally divided into three groups depending on their principle. The first one is to find conserved sequences cross close relation species. This is called phylogenetic footprinting. The second approach makes use of an exhaustive calculation to evaluate the frequency of occurrence of all possible

sequences of length n (n -mers). A background sequence distribution (of n -mers found in a set of non-coregulated genes) is used as a basis for comparison and any n -mers that are more overly abundant than expected are saved for further analysis as putative motifs[49-52]. The third approach specifies the n -mer as a probability matrix and utilizes an iterative algorithm, typically represented as an expectation maximization[53-57] or Gibbs sampling procedure[58-62].

4.4.1 Expectation Maximization

Expectation Maximization (EM) algorithm was proposed by Lawrence and Reilly in 1990[55]. There is an assumption that the binding sites for a regulatory protein have a fixed length l . This assumption is a critical limitation for this model. This model regards a motif as a $4 \times l$ matrix of probability values, each entry giving the probability of the nucleotides A, C, G, T occurring in that column position. This is called a probabilistic model of a motif. Then, motifs are compared with background sequence, where nucleotides are chosen independently from a common distribution. In a typical data set, each observed upstream sequence is assumed to have a single instance of the motif, but its exact location is unknown. EM algorithm use this missing data and it iteratively maximizes the expected *log likelihood* over the conditional distribution of the missing data, given the observed data and current estimates of parameters θ .

MEME is one of the most popular programs for finding motifs. It uses the MM

algorithm, which is an extension of the expectation maximization algorithm.

MM is basically the same as EM algorithm except for that it does not require that each sequence in the dataset contains one occurrence of the motif.

4.4.2 Gibbs Sampling Algorithm

In 1993, Lawrence et al. published a Bayesian version of the motif model which applies Gibbs sampling to estimate parameters and motif locations[59]. Gibbs sampling can stochastically find out motifs from a set of sequences which are believed to contain the motif. Gibbs sampling approach is essentially a special numerical approximation method, Markov Chain Monte Carlo (MCMC), which enables one to draw samples of high-dimensional random variables in an iterative fashion.

The *Gibbs sampling* uses a randomized approach to iteratively improve a motif. Initially, a motif, represented as a PSSM, is generated by selecting one random length l segment (substring) from each of the m sequences involved. Then the Gibbs sampling will repeatedly perform a *predictive update step* and a *sampling step* to iteratively improve the motif.

When using the Gibbs sampling, different runs will give different results. Normally, the motif is chosen from the best out of a number of runs.

Also, there are various variants of the standard Gibbs sampling approach. These

include the *Gibbs motif sampler*, *AlignACE* and *BioProspector*[63].

4.4.3 Limitation of EM and Gibbs sampling

The major drawback of EM based algorithms is that it cannot escape from a local optimum. Other limitations of EM based motif finding tools are: 1). the motif length l must be given fixed beforehand as input; 2). the expect number of different motifs found also need to be pre-defined as the input.

By contrast, Gibbs sampling algorithms combine gradient search steps with random jumps in the search space, so they can spend an unpredictable number of iterations before converging. Gibbs Sampling algorithm is one of the pattern sampling methods like EM, which is always probabilistic. The pattern sampling method is usually very fast, but it only delivers approximate solutions.

4.4.4 Phylogenetic footprinting

Phylogenetic footprinting is a technique that uses functional/non-functional sequence dichotomy to identify regulatory elements. Functional sequences tend to evolve much slower than non-functional sequences, as they are subject to selective pressure. It is this difference in mutation rates that phylogenetic footprinting exploits. To identify regulatory elements associated with a given gene, one considers a set of orthologous non-coding sequences from a group of related species. If these sequences contain unusually well conserved regions, it is a reasonable conjecture that these regions have some regulatory function. Many

papers using this method have been published[64-70].

Chapter 5 Finding TF binding motifs by combining sequence overrepresentation and cross-species conservation

One major challenge of modern biology is to understand how gene expression is regulated. An important step toward this understanding is to reliably identify the transcriptional factor binding sites (TFBSs) of co-regulated genes. Novel computational methods for finding TFBSs have long been sought owing to the arduous lab work. The current prevailing TFBS finding methods yield a large number of false positive predictions due to the short, variable nature of transcriptional binding sites (TFBSs). To reduce the false positive rate of motif finding we proposed a method here that combines sequence over-representation and cross-species sequence conservation to detect TFBSs in upstream regions of a given set of co-regulated genes.

5.1 Material

In this study, we considered gene promoter regions of four yeast species, namely *S. cerevisiae*, *S. mikatae*, *S. bayanus* and *S. paradoxus*. The last three are believed to be separated from *S. cerevisiae* by an estimated 5-20 million years of evolution, and they were found to have sufficient sequence similarity to *S. cerevisiae* to allow

orthologous regions to be aligned reliably[71].

We obtained the information about gene regulation network of *S. cerevisiae* from the database SCPD (The Promoter Database of *Saccharomyces cerevisiae*)[29], which provided us sets of co-regulated genes for different TFs. The sequences of genes and their upstream regions for each of the four yeast species were downloaded from <http://www.broad.mit.edu/>.

With the increment of the number of species, the sensitivity of the algorithm to find the motif will decrease, but the correspondence to known motif will enhance. It has been recommended that three or four species should be used [70].

The genes known to be co-regulated by specific TFs such as *STE12* and *GAL4* were used to evaluate our method. We let *PS* (positive set) denote the collection of *S. cerevisiae* genes co-regulated by a common TF, in the meanwhile we built another collection *NS* (negative set) which consists all the genes from yeast genome except those in *PS*. For each gene in both *PS* and *NS* we extracted the upstream promoter region sequences for all the four species, and aligned them using multiple sequence alignment program *ClustalW*.

We also applied our proposed method to *Drosophila* genomes. We got gene promoter regions of four *Drosophila* species, namely *D. melanogaster*, *D. yakuba*, *D. erecta* and *D. ananassae*, from the database FlyBase (<http://www.flybase.org/>),

and we got the information about gene regulation networks of *D. melanogaster* from DBSD (The *Drosophila* Binding Site Database). The known binding sites of the transcription factors were obtained from the TRANSFAC.

5.2 Method

The method proposed here requests gene upstream sequences from multiple closely related ortholog species. Usually we are interested in motif finding for only one of the species, namely the principal species, whereas the sequences from other species are helpful for the reduction of false positives. For a given TF we collect two sets of genes, namely positive set (*PS*) and negative set (*NS*). *PS* consists of the genes co-regulated by this TF, whereas *NS* is a set of genes randomly collected from the genome of the principal species.

5.2.1 Finding overrepresented sequence patterns

To find overrepresented sequences, we only need to consider the principal species. We first searched the upstream region of genes in *NS* for each possible sequence of length M ($6 \leq M \leq 17$) under following constraints: the first three nucleotides in the left flank and the last three nucleotides in the right flank were the core elements and fixed, between the two core elements there might be M_0 nucleotides ($M_0=0, 1, 2, \dots, 11$) and each of them could be any of the nucleotides A, C, T and G. So within the L bp upstream region of a gene there are $L - M + 1$ possible locations that can be occupied by a sequence of length M . We called the fraction

$$p = \frac{c_n}{c} \quad (5.1)$$

background probability of the given sequence, where c_n is the number of occurrences of the given sequence in the upstream regions of genes in NS , and c is the total number of possible locations for a sequence of M bp in the upstream regions of genes in NS . In the same way we can obtain the number of the sequence occurrences K and the total number N of possible locations that could be occupied by a M bp sequence in upstream regions of genes in PS . Using binomial distribution[40, 72-74] we can calculate the probability of the considered sequence pattern occurring more than K times in the upstream regions of the genes in PS as following:

$$P = \sum_{k=K}^N \frac{N!}{(N-k)!k!} p^k (1-p)^{N-k}, \quad (5.2)$$

where p is the background probability estimated by (5.1). We choose sequences with p -values less than a threshold p^* (usually in magnitude of 10^{-6} after Bonferroni adjustment) for further analysis. If two sequences have an overlap of more than 80% of the entire length of one of them, then we removed the one with larger P -value from our collection of overrepresented sequences. Both DNA strands were considered when we calculated the number occurrences of a given sequence in the upstream region. For example, when we count the occurrence of sequence *TGAAAC*, we also count the occurrence of sequence *GTTTCA*. If the given sequence is not a palindrome, we add the two numbers together and regard it as the total occurrences. If the sequence is a palindrome, we just use the count in one strand as the total occurrence.

Bonferroni Adjustment

To perform multiple statistical significance tests for determining overrepresented sequences, we use the Bonferroni adjustment to make it more "difficult" for any single test to be statistically significant. The Bonferroni adjustment usually is accomplished by dividing the significant level (usually set to 0.05, 0.01, etc.) by the number of tests being performed. In our case, the threshold of P-value we set as 0.05 divided by the number of all possible sequences in the form of $NNNnn...nnNNN$, where NNN stands for three fixed nucleotides and $nn...nn$ stands for unfixed number (from 0 to 11) of nucleotides that each of them could be any of the A, C, T and G.

5.2.2 Relative entropy and Conservation criteria

Let α be the background nucleotide distribution, and β be the nucleotide distribution at a given position in a multiple sequence alignment. For the two probability distributions α and β , the relative entropy (also known as Kullback-Leibler ‘distance’) is defined by:[75]

$$H(\beta \parallel \alpha) = \sum_i \beta_i \log \frac{\beta_i}{\alpha_i}, \quad (5.3)$$

We can prove that relative entropy is always a non-negative value. It reflects the extent of the deviation of actual nucleotide distribution from background

distribution at a given site in the alignment. The larger the value, the greater the deviation between the actual distribution and the background distribution at that site

Given an overrepresented sequence O obtained as described above, we search for O in the alignment of upstream sequences from four species for each gene in PS and NS , respectively. If we find an occurrence in the alignment for any gene in PS , we extract the corresponding sequence block from the alignment and put the four sequences that form this block to a sequence set S_{OP} . Similarly we also build a sequence set S_{ON} for genes in NS .

We further align all the sequences in either S_{OP} or S_{ON} . These two alignments are used to evaluate the degree of conservation of O across closely related species. We define the average relative entropy (ARE_P) of S_{OP} as

$$ARE_P = \frac{\sum_{i=1}^M EP_i}{M}, \quad (5.4)$$

where EP_i is the relative entropy at the position i of the alignment of the sequences in S_{OP} , and M is the length of O . If O is not found in the alignment of upstream sequences for any gene in NS , then we deposit O to the collection of candidate motifs for further consideration. Otherwise we could also calculate the average relative entropy ARE_N for the sequences in non-empty set S_{ON} . We define a Z-score as

$$Z = \frac{ARE_P - ARE_N}{\sqrt{\frac{s_N^2}{M}}}, \quad (5.5)$$

where s_N is the standard deviation of the relative entropies at different positions of the multiple upstream sequence (across multiple species) alignments of all genes in NS .

Binding motifs tend to be conserve in the orthologous species (see Figure 5-2 and Figure5-3), so we removed the sequences that are overrepresented but not conserved from our collection of candidate sequences. We set $Z=2$ as our threshold of Z-score, an overrepresented sequence O with $Z > 2$ will be kept as candidate sequences for further considerations.

a

```

Scer_YMR232W      C-TTGTCGCGAAGTTAATCTGAAACAATATATGTTACCTAC-----TGAAACAGCGCATGT
Spar_16391         C-TTGTCGCGAAGTATATCTGAAACATAAATGTTACATAT-----TGAACCGGCGCATGT
Smik_17300         C-TTGTCGCGATGCTAACCTGAAACATACTTATTACATGCCATGCTATGCCGGCCTAGGT
Sbay_19697         AGTTGTCGCGATGCGCATCTGAAACAATACATGCTACACGG-----TTAACTGGTACATGC
                  ***** * * ***** * *** * * *

```

b

```

Scer_YML047C      CATACTGAAACGAATGATCTGAAACAATAAAAAAAG----ATATTTGTATATAAATTAT
Spar_17501         CATAATGAAACAAATGATCTGAAACAATGAAAAAAATTCTGTATCTGGATATATATAAT
Smik_16240         TGTAATGAAAAATGAAGATCTGAAACAATGGTAAAAAT---ACAGCCGTATATAAATAA-
Sbay_17947         CATAAAAAAGGCAAAATGTTTAAAAACGCTGGAAAAAA-----GATGCTCATGTATATAAT
                  ** ** * * * * * * * * * *

```

c

```

Scer_YMR232W      C-TTGTCGCGAAGTTAATCTGAAACATATATGTTACCTAC-----TGAAACAGCGCATGT
Spar_16391         C-TTGTCGCGAAGTATATCTGAAACATAAATGTTACATAT-----TGAACCGGCGCATGT
Smik_17300         C-TTGTCGCGATGCTAACCTGAAACATACTTATTACATGCCATGCTATGCCGGCCTAGGT
Sbay_19697         AGTTGTCGCGATGCGCATCTGAAACATACATGCTACACGG-----TTAACTGCTACATGC
                  ***** * * ***** * *** * * *

```

Figure 5-1 Conservation of motif sequence among close related species

The known DNA binding motif *TGAAACA* of transcriptional factor *Ste12* for *S. cerevisiae* genes YMR232W(a), YML047C(b) and YMR232W(c), which are commonly regulated by *Ste12*. The motif is conserve across the four closely related yeast species, namely *S. cerevisiae*, *S. mikatae*, *S. bayanus* and *S. paradoxus*. The subfigures (a), (b) and (c) are three segments of the alignment, in which the motif incidences are highlighted using boxes.

Figure 5-2 minus logarithm of p -value versus average relative entropy

(for sequence segments with top 100 lowest over-representation p -values in the upstream of *Ace2*-regulated *S. cerevisiae* genes) The arrow points to the known binding motif GCTGGT of *Ace2* in *S. cerevisiae*. We can see that the average relative entropy of this known binding motif is greater than that of most other sequences.

Figure 5-3 minus logarithm of p -value versus Z -value

(for sequence segments with top 100 lowest over-representation p -values in the upstream of *Ace2*-regulated *S. cerevisiae* genes) The arrow points to the known binding motif GCTGGT of *Ace2* in *S. cerevisiae*. We can see that the Z -value of this known binding motif is greater than that of most other sequences.

5.2.3 Building a profile for a candidate motif

We searched for a given candidate motif in the alignments generated from upstream sequences of each genes in *PS*. If an instance was found in any of the species, we extracted the corresponding sequence block from the alignment for further consideration. We used e_B to denote the average of the relative entropies of M different positions of an alignment block of length M . For each block obtained above we set $h_p = \mu_p + 2(\sigma_p / \sqrt{M})$ as our cutoff value for block selection, where μ_p and σ_p are the population mean and population standard deviation of the entropy, respectively, for a randomly chosen position in the alignment of the four upstream sequences of a randomly selected gene in *PS*. The value μ_p and σ_p could be estimated respectively by the average and standard deviation of entropies at all positions of the alignments of upstream sequences for genes in *PS*. For a given candidate motif, we used the blocks with e_B greater than h_p to compute a profile for representing the potential motif. For example, we searched for a candidate motif *GTTTCA* in the alignments of upstream sequences of genes in *PS*. If we found it in any species in the alignment we extracted the corresponding block, calculated e_B and compared it with h_p to decide whether we would keep this block for further consideration. For the selected blocks we calculated the base frequencies at each position and created thereafter the profile to represent the candidate motif. Both strands were considered when we constructed the profile.

5.2.4 Building PSSM

The profile obtained represents the candidate motif derived from all the ortholog species. Usually we are only interested in the motif finding for one species which was named as the principal species in our analysis, and it is necessary to build a species specific *PSSM* (Position Specific Score Matrix) for the candidate motif[76]. For genes in *PS* from the principal species, we search their upstream sequences using the above motif profile, and all the significant hits found are included in building the species specific *PSSM*. The profile search is performed as following. For each M -bp segment of upstream sequences of genes in *PS*, we calculate a score

$$Sc = \prod_{i=1}^M q_i, \quad (5.6)$$

where q_i is the probability of observing the i -th nucleotide of the segment, which is defined by the position specific nucleotide distribution in the profile of the initial candidate motif. To determine the criteria for a segment to be considered as a significant match, we first calculate scores for all the possible M -bp segments of the upstream sequences (for principal species only) of genes in *NS*, and rank these scores in the descending order. We use the 0.001-quantile of these ranked scores, denoted as Sc^* , as the threshold value to determine whether a match is significant in the profile search. For example, if there are 1000 genes in the *NS* and the length of each promoter region is L -bp, then there are totally $1000*(L-M+1)$ possible segments, so we have $1000*(L-M+1)$ scores. We sort the scores in the descending order and set the n -th value as the cut-off score Sc^* with $n = L - M + 1$. We calculate Sc for each possible segment in the upstream regions (principal species only) of

the genes in PS . If $Sc \geq Sc^*$, we deposit the segment into set I , which is the set of incidences of the candidate motif under consideration.

5.2.5 Optimal motif length

Let k be the number of sequence segments in set I . In order to determine the optimal length of the potential motif, we extend 0 to 5 bp in both flanks of each M -bp segment in I in terms of its original mother sequence in the upstream of genes in PS . So we have totally 36 possible combinations (left flank extended by $M_L = 0, 1, 2, 3, 4$ or 5 bp; right flank extended by $M_R = 0, 1, 2, 3, 4$ or 5 bp). For each possible combination (M_L, M_R) we put the newly added flanks into a block with k sequences and $M_L + M_R$ columns. We calculate the average relative entropies of all the 36 blocks and choose the combination (M_{L^*}, M_{R^*}) which delivers the maximum average relative entropy e_{B^*} for further consideration. In the meanwhile, we randomly generate 1000 sequence blocks, each with k sequences and $M_{L^*} + M_{R^*}$ columns, in terms of the background nucleotide distribution α . We calculate the mean e_{rand} and the standard deviation s_{rand} of the average relative entropies of these 1000 blocks. If e_{B^*} is greater than $e_{rand} + 2s_{rand}$, then we accept the extension (M_{L^*}, M_{R^*}) and set the final motif length at $M + M_{L^*} + M_{R^*}$, otherwise we still keep the original motif length M . The extended sequences (M_{L^*} bp in left flank and M_{R^*} bp in right flank) of the segments in I form a new sequence set I_e , which is the set of the incidences of the extended motif. Using all the sequences in I_e we build the *PSSM* for a general

representation of the final motif.

5.3 Results

We used 35 well studied yeast transcriptional factors (*Ste12*, *Gal4*, *MET31*, *Mbp1*, *Leu3*, *Pho4*, *Cbf*, etc.) to evaluate the proposed method. The criteria for selecting these TFs are: 1). their true DNA binding motifs are known; 2). the orthologous genes are available in all four yeast species, and the upstream sequences of these genes are available. For each TF we built a positive set (*PS*), which consists all the genes that are known to be co-regulated by the this TF (see Table 5-1), in the meanwhile we also built a negative set (*NS*), which consists all the genes from yeast genome except those in *PS*. The *NS* was used to evaluate the background information, as well as to serve as a control in our motif finding process. For each gene in both *PS* and *NS* we extract its upstream promoter region sequences from genomes of four different yeast species, namely *S. cerevisiae*, *S. mikatae*, *S. bayanus* and *S. paradoxus*. We took *S. cerevisiae* as the principal specie in our study.

We wrote a PERL script to implement our method for finding binding motifs of a given TF (see Table 5-1 for a subset of TFs considered in this study) in the upstream sequences of its co-regulated genes. We found the true motifs, which are identical or similar to the known binding motif, for 25 TFs among the total 35 used in the study. In table 5-2 we listed the known DNA binding motif of each TF and

the motif found using our method that is identical or most similar to the known motif. Three different threshold settings were used when we searched for binding motifs for these 35 TFs. We firstly applied the over-representation p-value threshold p^* after *Bonferroni* Adjustment as described above to the motif searching. If none motif is reported, then we change the p-value threshold to 0.01, if there is still no motif reported, then we changed the default ARE_p threshold to 0.8 instead of the default value 1.0. We compared our method with four different popular motif finding tools: *AlignACE*, *MEME*, *PHYME* and *Footprinter*. *AlignACE* and *MEME* are fed with single-genome sequences, whereas *PHYME* and *Footprinter* use multi-genome sequences as inputs. To compare our method with *AlignACE* and *MEME*, we used the upstream sequences of *S. cerevisiae* genes in the *PS* of each TF as the input of *MEME* and *AlignACE*, respectively. All the parameters were set to default when we used *AlignACE* in finding motifs. To apply *MEME* to the motif finding, we need to specify some parameters. We set the minimum length of the potential motif as 6, which is the same as our method, and we set the number of motifs expected to be found at the same as the number of predicted motifs found using our method, and other parameters set as default. The comparisons are listed in Table 5-3 and Table 5-4 respectively. We can see that the number of predicted motifs found for each TF by our method is in general much less than that found by *AlignACE* (Table 5-3) or *MEME* (Table 5-4). Table 5-3 and Table 5-4 showed that our method is more efficient in finding the true motifs comparing to *AlignACE* and *MEME*, in the sense that it returned less number of potential motifs, and the ranks

of the true motifs are also generally higher than that in the output of *AlignACE* or *MEME*. For example, there were 11 potential motifs found by *AlignACE* for *Ste12*, and the known motif of *Ste12* ranked 2nd in the output; however, using our method only one motif was found, and it was the known motif. The results for other TFs showed the same tendency. *AlignACE* and *MEME* could only find true binding motifs for 14 and 12 TFs, respectively, out of the 25 TFs whose true binding motifs were detected using our method.

Unlike single genome based motif finding methods *AlignACE* and *MEME*, our methods uses sequence information from multi-genome sequences, so it is more reasonable to compare it with *PHYME* and *Footprinter*, which are two popular multiple-genome based methods. For a given TF, *Footprinter* usually predicts over 1000 candidate motifs, and the overwhelming number of predictions makes it impractical. The reason of too many predicted motifs from *Footprinter* may be that it is very difficult to have an optimal or sub-optimal parameter setting for this method. So we focus here on the comparison of our method with *PHYME*. To apply *PHYME* to the motif finding, we set the motif length limit as 17, which is the maximum length among all known binding motifs of 35 TFs. For each regulon the number of motifs predicted was set as 10 and the motifs were searched on both strands. The results of this comparison are listed in Table 5-5. Using our method we can find known binding motifs for 25 out of the 35 TFs. Using *PHYME* we can find known motifs for 17 out of the 25 TFs whose binding sites can be found by our method. In addition, *PHYME* can find known motifs for 6 out of the 10 TFs

whose known motifs can not be found by our method. From Table 5-5 we can see that the two methods nearly have the same ability in motif finding, however the rank of the known motif found by our method is generally higher than that found by *PHYME*. Table 5-6 gives the rank frequencies of 35 known TF binding motifs found using different methods. We can see that 11 out of the 35 known binding motifs ranked 1 with our method, which is the highest among the four methods under comparison. In this sense the sensitivity of our method is higher than the others.

Table 5-7 gives a list of TFs whose binding motifs could be found using our method, but could not be found by *MEME*, *AlignACE* and *PHYME*. Our method could not find known binding motifs for 10 TFs out of the 35 (Table 5-8). Out of these 10 TFs, *AlignACE* and *MEME* can find known binding motifs for 5 and 3 TFs, respectively. With *PHYME*, we can find known binding motifs for 6 TFs out of these 10 (Table 5-5).

To further test our method, we also applied it to the TFBS finding of two *Drosophila* transcription factors, namely *dl* and *ttk*. The result (Table 5-9) shows that our method is able to find the known binding sites of these two transcription factors.

TF	Number of co-regulated genes	Genes regulated by the TF
Ste12	9	YBR083, YCL055W, YFL027C, YJL170C, YLR452C, YML047C, YMR232W, YNL279W, YPL156C
Gal4	10	GAL2, GAL3, GAL1/10, GAL7, MTH1, FUR4, PCL10, GAL80, PGM2, GCY1
MET31	8	YEL015W, YEL016C, STR3, MET16, NUT2, SSN8, YJL060W, YEL072W
Mbp1	18	YEL018W, MMS21, YCK2, MCD1, MCM2, RPS9A, MOT1, OPY2, CLB5, YER071C, VTC1, YJL045W, MSH6, YNR009W, HXT10, YER087C-A, TOF1, YNL274C
Leu3	10	YDR279W, LEU1, OAC1, YOR271C, YDL228C, YHR209W, YHR207C, BAT1, ILV2, RRP6
Cbf1	16	YAL026C, YBR089C-A, YBR225W, YDR438W, YIL074C, YIL126W, YIL127C, YJL167W, YJL168C, YJL209W, YJR010W, YKL191W, YKL192C, YNL094W, YNL095C, YNL282W
Ace2	1	YLR286C
Gcn4	6	YBL103C, YDL170W, YKL015W, YLR451W, YML099C, YNL103W
Abf1	15	YAL038W, YBR248C, YCR012W, YFL038C, YFR031C, YGL234W, YGR059W, YHR174W, YIL160C, YJL166W, YKL112W, YLR203C, YLR204W, YOR116C, YPR110C
Hap1	4	YEL039C, YJR048W, YML054C, YOR065W
Ino4	6	YDR050C, YER026C, YGR157W, YHR123W, YMR084W, YNR016C
Mcb	6	YDL102W, YDL164C, YJL194W, YMR199W, YNL102W, YOR074C
Mse	1	YGR059W
Nbf	1	YJL153C
Pdr3	2	YBL005W, YGR281W
Pho4	2	YDR481C, YGR233C
Put3	1	YHR037W
Rap1	8	YFL014W, YFR031C, YGL123W, YKL062W, YLR399C, YNL216W, YOL082W, YPR102C
Swi5	2	YDL227C, YNL327W
Uasino	1	YJL153C
Uasrad	2	YCR066W, YGL058W
Adr1	2	YDR256C, YMR303C
Mig1	7	YBR019C, YBR020W, YDR009W, YDR146C, YIL162W, YKL109W, YPL248C
T4c	2	YJL106W, YJL153C

		Our Method
Uasphr	14	YBR114W, YDL200C, YDR217C, YEL037C, YER095W, YER142C, YGL058W, YIL066C, YJL026W, YJR035W, YJR052W, YML032C, YNL250W, YPL022W
Ap-1	1	YGR209C
Bas2	2	YCL030C, YGL234W
Csre	2	YER065C, YNL117W
Mac1	11	YDR058C, YDR075W, YER145C, YER146W, YGR136W, YJR049C, YJR050W, YNL250W, YNL251C, YPR110C, YPR111W
Gcr1	2	YAL038W, YGR215W
Mcm1	17	YAL040C, YBR160W, YBR202W, YDR146C, YDR403W, YER111C, YFL026W, YGL008C, YGR108W, YJL159W, YJL194W, YKL178C, YKL209C, YKR066C, YNL277W, YPR113W, YPR119W
Reb1	12	YCR012W, YDL164C, YDR007W, YDR050C, YDR146C, YER086W, YFL039C, YGL026C, YNL216W, YOL004W, YOL006C, YPL231W
Rox1	2	YDR044W, YPR065W
Scb	2	YDL227C, YMR199W
Sff	3	YDR146C, YGR108W, YPR119W

Table 5-1 Transcription factors and the genes being regulated

TF	Genes in PS	Known motif	Motif Found	P-value	Z-value	ARE _p
Ste12 ^{a)}	9	TGAAACA	TGAAACA	5.6e-12	3.68	1.00
Gal4 ^{a)}	10	CGGNNNNNNNNNNNCCG	CGGNNNNNNNNNNNCCG	3.7e-12	2.43	1.22
Mbp1 ^{a)}	18	ACGCGTNA	ACGCGT	3.0e-7	3.37	1.35
Leu3 ^{a)}	10	CCGGNNCCGG	CGGNNCCGG	7.0e-13	3.15	1.27
Cbf1 ^{a)}	16	RTCACRTG	CACGTG	7.7e-13	2.98	1.19
MET31 ^{a)}	8	CTGTGGC	TGTGGC	6.7e-7	3.39	1.06
Abf1 ^{b)}	15	TCRNNNNNNNACG	TCANNNNNNNACG	1.3e-3	3.73	1.26
Ace2 ^{b)}	1	GCTGGT	TGCTGGT	1.4e-3	6.07	1.55
Gcn4 ^{b)}	6	TGANTN	ATGACT	8.7e-4	4.45	1.10
Hap1 ^{b)}	4	CGGNNNTANCCG	TGCCGNNNNNNNCCG	2.3e-4	6.09	1.64
Ino4 ^{b)}	6	CATGTGAAAT	CATGTT	2.9e-4	5.60	1.31
Mcb ^{b)}	6	WCGCGW	CGCNTCG	4.1e-4	4.66	1.36
Mse ^{b)}	1	CRCAA AW	GACNCAA	8.3e-3	4.05	1.19
Nbf ^{b)}	1	ATGYGRAWW	CATGTG	5.9e-3	5.85	1.36
Pdr3 ^{b)}	2	TCCGYGGA	TCCNNGGA	4.3e-4	2.88	1.03
Pho4 ^{b)}	2	CACGTK	GCGCGT	1.8e-3	3.55	1.20
Put3 ^{b)}	1	CGGNNNNNNNNNNNCCG	TCGNNNNNNNNNNNCCG	2.6e-4	4.65	1.51
Rap1 ^{b)}	8	RMACCCA	GTCNNNNNCCCAT	8.8e-3	3.16	1.01
Swi5 ^{b)}	2	KGCTGR	TGCTGG	6.5e-4	4.45	1.19
Uasino ^{b)}	1	ATCTGA AWW	CATGTG	5.9e-3	5.83	1.36
Uasrad ^{b)}	2	WTTTCCCGS	TCCNGCT	1.1e-3	4.42	1.24
Adr1 ^{c)}	2	TCTCC	CTCCNNNNNTCC	1.6e-3	2.18	0.88
Mig1 ^{c)}	7	CCCCRNWWWW	ACCCCA	7.2e-3	2.18	0.82
Uasphr ^{c)}	14	CTTCCT	TCTNNNNNNNNNTCCT	2.2e-3	2.38	0.93
T4c ^{c)}	2	TTTTCTYCG	TTTTCNNNNNTCC	1.2e-3	2.69	0.96

Table 5-2 Known motifs that can be found by our method

Comparison of known motifs and the motifs found (using our method) that are the most similar to the known motifs. Different parameter threshold settings are used in motif finding. a) P-value in a magnitude of 10^{-6} (after Bonferroni correction), $ARE_p=1.0$, and Z-value =2.0; b) P-value = 0.01 (without Bonferroni correction),

$ARE_p=1.0$, and Z-value =2.0; c) P-value = 0.01 (without Bonferroni correction),

$ARE_p=0.8$, and Z-value =2.0.

Our Method				
TF	AlignACE		Our Method	
	The number of motifs found	The rank of the known motif	The number of motifs found	The rank of the known motif
Ste12 ^{a)}	11	2	1	1
Gal4 ^{a)}	9	2	1	1
Leu3 ^{a)}	22	3	1	1
Mbp1 ^{a)}	20	7	2	1
Cbfl ^{a)}	29	1	4	2
Met31 ^{a)}	20	15	1	1
Abfl ^{b)}	10	4	5	3
Ace2 ^{b)}	7	Not found	4	2
Gcn4 ^{b)}	11	4	39	6
Hap1 ^{b)}	6	Not found	23	1
Ino4 ^{b)}	13	Not found	22	4
Mcb ^{b)}	18	2	11	1
Mse ^{b)}	6	Not found	5	5
Nbf ^{b)}	6	1	16	15
Pdr3 ^{b)}	12	Not found	10	2
Pho4 ^{b)}	8	Not found	9	1
Put3 ^{b)}	2	Not found	4	1
Rap1 ^{b)}	13	6	14	11
Swi5 ^{b)}	9	Not found	3	1
Uasino ^{b)}	4	Not found	14	13
Uasrad ^{b)}	4	Not found	20	3
Adr1 ^{c)}	4	2	14	2
Mig1 ^{c)}	30	2	32	30
Uasphr ^{c)}	14	Not found	47	13
T4c ^{c)}	9	1	28	6

Table 5-3 Comparison of our method with *AlignACE*

Comparison of our method with *AlignACE*: For each TF, we listed the numbers of predicted motifs found using the two methods, as well as the rank of the known motif among the predicted motifs found using the two methods, respectively. In our motif finding method, three different parameter threshold settings a), b) and c) are

used as given in Table 5-2.

				Our Method
TF	Meme		Our Method	
	The number of motifs found	The rank of the known motif	The number of motifs found	The rank of the known motif
Ste12 ^{a)}	1	Not found	1	1
Gal4 ^{a)}	1	1	1	1
Leu3 ^{a)}	1	1	1	1
Mbp1 ^{a)}	2	1	2	1
Cbf1 ^{a)}	4	1	4	2
Met31 ^{a)}	1	Not found	1	1
Abf1 ^{b)}	5	Not found	5	3
Ace2 ^{b)}	4	Not found	4	2
Gcn4 ^{b)}	39	Not found	39	6
Hap1 ^{b)}	9	Not found	23	1
Ino4 ^{b)}	22	10	22	4
Mcb ^{b)}	11	1	11	1
Mse ^{b)}	5	Not found	5	5
Nbf ^{b)}	16	3	16	15
Pdr3 ^{b)}	10	1	10	1
Pho4 ^{b)}	9	4	9	1
Put3 ^{b)}	4	Not found	4	1
Rap1 ^{b)}	14	Not found	14	11
Swi5 ^{b)}	3	Not found	3	1
Uasino ^{b)}	14	Not found	14	13
Uasrad ^{b)}	20	1	20	3
Adr1 ^{c)}	10	Not found	14	2
Mig1 ^{c)}	32	2	32	30
Uasphr ^{c)}	47	Not found	47	13
T4c ^{c)}	28	23	28	6

Table 5-4 Comparison of our method with *MEME*

Comparison of our method with *MEME*: *MEME* requests for a pre-determined number of predicted motifs as its input, so when we compared our method with *MEME*, we let the number of predicted motifs fed to *MEME* be the same as the number of predicted motifs found using our method. For each TF, we also listed the

rank of the known motif among the predicted motifs found using the two methods, respectively. In our motif finding method, three different parameter threshold settings a), b) and c) are used as given in Table 5-2.

TF	Known Motif	Found by our method	Rank	Found by <i>PHYME</i>	Rank
Ste12 ^{a)}	TGAAACA	TGAAACA	1	TGAAACA	3
Gal4 ^{a)}	CGGNNNNNNNNNNNCCG	CGGNNNNNNNNNNNCCG	1	CCGAATAGTCTGCCCCG	8
Mbp1 ^{a)}	ACGCGTNA	ACGCGT	1	ACGCGTCA	3
Leu3 ^{a)}	CCGGNNCCGG	CGGNNCCGG	1	CCGGTACCGG	3
Cbfl ^{a)}	RTCACRTG	CACGTG	2	GTCACGTG	2
MET31 ^{a)}	CTGTGGC	TGTGGC	1	Not found	
Abf1 ^{b)}	TCRNNNNNNNACG	TCANNNNNNACG	3	Not found	
Ace2 ^{b)}	GCTGGT	TGCTGGT	2	Not found	
Gcn4 ^{b)}	TGANTN	ATGACT	6	TGAGTC	6
Hap1 ^{b)}	CGGNNNTANCCG	TGCCGNNNNNNNCCG	1	Not found	
Ino4 ^{b)}	CATGTGAAAT	CATGTT	4	Not found	
Mcb ^{b)}	WCGCGW	CGCNTCG	1	ACGCGT	1
Mse ^{b)}	CRCAAAW	GACNCAA	5	CACAAAA	3
Nbf ^{b)}	ATGYGRAWW	CATGTG	15	ATGTGAAAT	1
Pdr3 ^{b)}	TCCGYGGA	TCCNNGGA	1	TCCGCGGA	2
Pho4 ^{b)}	CACGTK	GCGCGT	1	Not found	
Put3 ^{b)}	CGGNNNNNNNNNNNCCG	TCGNNNNNNNNNNNCCG	1	Not found	
Rap1 ^{b)}	RMACCCA	GTCNNNNNCCCAT	11	AAACCGA	4
Swi5 ^{b)}	KGCTGR	TGCTGG	1	TGCTGAATG	1
Uasino ^{b)}	ATCTGAAWW	CATGTG	13	Not found	
Uasrad ^{b)}	WTTTCCCGS	TCCNGCT	3	TTTCCAC	4
Adr1 ^{c)}	TCTCC	CTCCNNNNNTCC	2	ACTCC	4
Mig1 ^{c)}	CCCCRNNWWWW	ACCCA	30	CCCCGCCCC	4
Uasphr ^{c)}	CTTCCT	TCTNNNNNNNNNNNTCCT	13	GCTTTCTTT	8
T4c ^{c)}	TTTTCTYCG	TTTTCNNNNNTCC	6	TTTTTCTTTT	1
Ap-1	TTANTAA	Not found		Not found	
Bas2	TAATRA,TAANTAA	Not found		TAATAG	8
Csre	YCGGAYRRRAWGG	Not found		Not found	
Mac1	GAGCAAA	Not found		GAGAAAA	3
Gcr1	CWTCC	Not found		Not found	
Mcm1	CCNNNWWRGG	Not found		CCGTTTTGGG	5
Reb1	YYACCCG	Not found		CTACCCG	5
Rox1	YYNATTGTTY	Not found		Not found	
Scb	CNCGAAA	Not found		CACGAAA	1
Sff	GTMAACAA	Not found		GTTAACAA	6

Table 5-5 Comparison of our method with *PHYME*

For each TF, we listed the numbers of predicted motifs found using the two methods, as well as the rank of the known motif among the predicted motifs found using the

two methods, respectively. In our motif finding method, three different parameter threshold settings a), b) and c) are used as given in Table 5-2.

Method	R1	R2	R3	R4	R5	R6-R10	>R10	Not Found
OM	11	3	2	1	1	2	5	10
MEME	8	1	2	1	0	2	1	20
AlighACE	4	6	2	2	1	2	2	16
Phyme	5	2	5	4	2	5	0	12

Table 5-6 Rank frequencies of the 35 known TF binding motifs found with different methods.

OM stands for our method. Rn ($n=1,2, \dots$) gives the count of the known motifs ranked n in the output of a motif finding method. From this table we can see that the known motifs found by our method tend to have higher ranks than other methods.

TF	Genes	In	Found by our method	Known Motif
	PS			
Ace2 ^{b)}	1		GCTGGT	TGCTGGT
Hap1 ^{b)}	4		TGCCGNNNNNNNCGG	CGGNNNTANCGG
Put3 ^{b)}	1		TCGNNNNNNNNNNCG	CGGNNNNNNNNNNCCG
Uasino ^{b)}	1		CATGTG	ATCTGAAWW

Table 5-7 The TFs whose known binding sites only can be found by our method

The TFs whose known binding motifs were found using our method, but can not be found by *MEME*, *AlignACE* and *PHYME*. In our motif finding method, three different parameter threshold settings a), b) and c) are used as given in Table 5-2.

	Known	AlignACE	rank	meme	rank
Ap-1	TTANTAA	Not find		TTAGTAA	3/10
Bas2	TAATRA,TAANTAA	Not find		Not find	
Csre	YCGGAYRRRAWGG	Not find		GTCCGGAC	8/10
Mac1	GAGCAAA	GGAAGCAAA	17/33	Not find	
Ger1	CWTCC	ATTGTTTTCC	5/5	Not find	
Mcm1	CCNNNWRGG	TTACNNNTAGGAAA	2/11	TTCCCTAATTAGGAAA	1/10
Reb1	YYACCCG	TTACCCGCACGGC	3/8	Not find	
Rox1	YYNATTGTTY	Not find		Not find	
Scb	CNCGAAA	AAGCCACGAAAA	1/13	Not find	
Sff	GTMAACAA	Not find		Not find	

Table 5-8 TFs whose binding site can not be found by our method

MEME and *AlignACE* results for the TFs whose known binding site can not be found using our method. We set the expected number of motifs predicted by *MEME* as 10.

Transcription Factors	Known Motifs (from TRANSFAC)	Motifs found by our method
dl	HGRGAAAANCV	GAGAANNNNNGAG
ttk	GGTCCTGC	GGTCCT

Table 5-9 Binding motifs of two *Drosophila* transcription factors

To further test our method, we applied it to finding binding motifs for two *Drosophila* transcription factors *dl* and *ttk*. The result suggested that our method can also be used to other genomes if the requested data are available.

5.4 Discussion

Transcriptional factors and the sites within the DNA to which they bind are two of the most important functional elements in eukaryotic genomes. A thorough study of the interactions of TFs and their DNA binding sites is important for mapping the regulatory pathways and understanding the potential function of the genes regulated by the TFs and the newly identified DNA-binding sites[77]. In the past decade clustering of gene expression profiles obtained from large scale DNA microarray experiments has been successfully used in identifying co-expressed genes[23, 78], and we believed that these co-expressed genes may share common regulators that bind to their upstream regions. Finding TF binding motifs of these potentially co-regulated genes becomes critical for understanding the interaction of the genes and their regulators[74, 79-81]. So far the binding specificities are well characterized for only a small number of TFs[82]. TFBS are usually quite short (around 6-25 bp) and degenerate, which leads to the difficulties in finding them reliably using current motif finding tools. Even though the *ab initio* motif finding tools have been used successfully in many cases, their performances are far from satisfying. The major drawback of these tools is that they produce many false positive predictions. Under default parameter setting they yield usually tens or hundreds of putative motifs, and it is difficult to judge which candidate motif out of them is functional[82]. Phylogenetic footprinting method has been suggested recently[65, 69, 83, 84], by which the inter-species comparative sequence information is used for helping to signal the presence of TF binding sites that

might not have been predicted using sequences from a single genome. For example, binding sites found in human sequence that are also found in orthologous mouse or other mammalian sequences are far more likely to be functional than those found only in human[85]. We refer to these short orthologous sequences that are conserved over 6 bp or more as phylogenetic footprint.

The method proposed here considers both overrepresentation and cross-species conservation of potential binding motifs. We used binomial test to determine the statistically overrepresented candidate sequences, and the average relative entropy of the aligned sequence block was used to measure the cross-species conservation of these candidates. The relative entropy is a popular measure of the degree of conservation at a site in a DNA or protein sequence alignment[75], and it is useful for finding unusual patterns in biological sequences. The input data of our method are the upstream sequences of two groups of genes, namely co-regulated genes of a TF (*PS*) and control genes (*NS*) randomly selected from the genome of the principal species under study, as well as the orthologous sequences from other species which are closely related to the principal species. Usually the co-regulated genes are collected through wet lab experiments, or predicted through gene expression profile analysis using microarray data. The upstream sequences of genes in *PS* and *NS* could be extracted from the genome of the principal species, and the corresponding upstream sequences from other species could be obtained by doing BLAST[86] or by downloading from the related databases that are publicly

available.

Three parameters are considered in our method, they are 1). P -value, which is used to evaluate the over-representation of a candidate sequence, 2) average relative entropy ARE_p of S_{Op} , which gives the degree of conservation of a candidate motif, 3) Z -value, which is used to assess the statistical significance of the conservation. In order to have a balanced consideration of the sensitivity and the specificity, and to deal with different situations, we applied three different parameter threshold settings to screen candidate motifs, and they are: a) P -value in a magnitude of 10^{-6} (after Bonferroni correction), $ARE_p=1.0$, and Z -value $=2.0$; b) P -value $= 0.01$ (without Bonferroni correction), $ARE_p=1.0$, and Z -value $=2.0$; c) P -value $= 0.01$ (without Bonferroni correction), $ARE_p=0.8$, and Z -value $=2.0$. Theoretically we can find most of the known motifs as long as the criteria for overrepresentation and conservation are set loose enough, however, the loose criteria may result in numerous putative motifs that are actually false positives. Considering the high cost of verifying a predicted motif through lab experiment, we used firstly the strict criteria for candidate sequence screening, so parameter setting a) was set as default in our method. Using this strict parameter threshold setting we may miss the true TF binding motifs (see table 5-3 and table 5-4), especially those without very high-level statistical significance of overrepresentation, and the method may not be able to return any predictions. So we loosed the criteria to setting b) or setting c) in actual motif finding process, if there is no hit using the default

thresholds. Setting b) has a moderate criterion for overrepresentation, so it allows more candidate motif to pass the screening. With setting c) we loose the criterion of the degree of conservation, since there do exist some known TF binding motifs with ARE_p less than 1.0 (See table 5-2).

The method proposed by us is not a replacement of single-genome tools such as *MEME* and *AlignACE*. The major limitation of our method is its strong prerequisite. Multiple closely related species and the upstream sequences of each co-regulated gene for all species under study are requested in our method. In many cases these prerequisites may not be met, and the method is therefore not generally applicable. Another problem is how to choose the appropriate species to evaluate the cross-species conservation. In principle, the species selected in the study should be close enough so that the conservation of motif sequences could be detected in a multiple alignment, in the meanwhile their evolutionary distances should not be too close, so that the signals could be distinguished from the noises[87]. The number of species used in this method is also a factor which needs to be considered. We recommended three or four species be used in the study, since too many species may bring up strong noise and reduce the detection power of the method.

Even though we used yeast genomes to assess our method in this study, it could also be applied to other organisms if suitable related species are available and the upstream sequences of co-regulated genes could be obtained for the multiple

species.

5.5 Conclusion

Comparing with the motif finding software such as *MEME*, *AlignACE* and *PHYME* our method has some advantages: 1) both overrepresentation and conservation criteria are used in screening candidate motifs, so the method gives relatively less predicted motifs for a group of co-regulated genes (Table 5-3, Table 5-4 and Table 5-5), hence it is helpful for reducing false positive predictions; 2). The rank of a true motif in the output of our method is in general higher than that of other methods (see both Table 5-3, Table 5-4 and Table 5-5), and this is of practical importance, since we usually focus only on putative binding motifs with high ranks despite the large number of predicted motifs; 3) Unlike the most common motif finding tools, our method requests no prior inputs such as the length of the motifs or the number of predictions.

5.6 Implement of the method

We use a PERL script to implement the method described here. The script and example input data are available for downloading as a Winzip package under URL:

<http://www.ntu.edu.sg/sbs/jmli/motif/index.html>.

References

1. Smith, L. and A. Greenfield, *DNA microarrays and development*. Hum Mol Genet, 2003.
12 Spec No 1: p. R1-8.
2. Brown, P.O. and D. Botstein, *Exploring the new world of the genome with DNA microarrays*. Nat Genet, 1999. **21**(1 Suppl): p. 33-7.
3. Bassett Jr, D.E., M.B. Eisen, and M.S. Boguski, *Gene expression informatics--it's all in your mine*. Nat Genet, 1999. **21**(1 Suppl): p. 51-55.
4. Burgess, J.K., *Gene expression studies using microarrays*. Clin Exp Pharmacol Physiol, 2001. **28**(4): p. 321-8.
5. Ivakhno, S.S. and O.I. Korneliuk, *Microarrays: technologies overview and data analysis*. Ukr Biokhim Zh, 2004. **76**(2): p. 5-19.
6. Auburn, R.P., et al., *Robotic spotting of cDNA and oligonucleotide microarrays*. Trends Biotechnol, 2005. **23**(7): p. 374-9.
7. Hegde, P., et al., *A concise guide to cDNA microarray analysis*. Biotechniques, 2000. **29**(3): p. 548-50, 552-4, 556 passim.
8. Stoughton, R.B., *Applications of DNA microarrays in biology*. Annu Rev Biochem, 2005. **74**: p. 53-82.
9. Debouck, C. and P.N. Goodfellow, *DNA microarrays in drug discovery and development*. Nat Genet, 1999. **21**(1 Suppl): p. 48-50.
10. Cole, K.A., D.B. Krizman, and M.R. Emmert-Buck, *The genetics of cancer--a 3D model*. Nat Genet, 1999. **21**(1 Suppl): p. 38-41.
11. Hardiman, G., *Microarray platforms--comparisons and contrasts*. Pharmacogenomics,

2004. **5**(5): p. 487-502.
12. Lipshutz, R.J., et al., *High density synthetic oligonucleotide arrays*. Nat Genet, 1999. **21**(1 Suppl): p. 20-4.
13. Cheung, V.G., et al., *Making and reading microarrays*. Nat Genet, 1999. **21**(1 Suppl): p. 15-9.
14. Duggan, D.J., et al., *Expression profiling using cDNA microarrays*. Nat Genet, 1999. **21**(1 Suppl): p. 10-14.
15. Coe, B. and C. Antler, *Spot your genes-An overview of the microarray*. The Science Creative Quarterly, 2006.
16. yee Hwa Yang, M.J.B., Sandrine Dudoit and Terence P. Speed, *comparison of methods for image analysis on cDNA microarray data*. 2000.
17. Qin, L., et al., *Spot detection and image segmentation in DNA microarray data*. Appl Bioinformatics, 2005. **4**(1): p. 1-11.
18. Lumonics, G., *QuantaArray Analysis Software, Operator's Manual*. 1999.
19. Eisen, M.B., *ScanAlyze*. 1999.
20. Beare, R. and M. Buckley, *Spot: cDNA Microarray Image Analysis Users Guide*. 2004.
21. Axon Instruments Inc, *GenePix Pro 4.0 User's Guide and Tutorial*. 2001.
22. Wang, H., et al., *Direct interaction of Arabidopsis cryptochromes with COP1 in light control development*. Science, 2001. **294**(5540): p. 154-8.
23. Ma, L., et al., *Light control of Arabidopsis development entails coordinated regulation of genome expression and cellular pathways*. Plant Cell, 2001. **13**(12): p. 2589-607.
24. Nunez-Garcia, J., V. Mersinias, and K.H. Cho, *The statistical distribution of the intensity*

- of pixels within spots of DNA microarrays: What is the appropriate single-value representative?* Applied Bioinformatics, 2003. **2**(4): p. 229-239.
25. Eisen, M.B., et al., *Cluster analysis and display of genome-wide expression patterns*. Proc Natl Acad Sci U S A, 1998. **95**(25): p. 14863-8.
 26. Palin, K., et al., *Correlating gene promoters and expression in gene disruption experiments*. Bioinformatics, 2002. **18 Suppl 2**: p. S172-80.
 27. Clarke, N.D. and J.A. Granek, *Rank order metrics for quantifying the association of sequence features with gene regulation*. Bioinformatics, 2003. **19**(2): p. 212-8.
 28. Matys, V., et al., *TRANSFAC: transcriptional regulation, from patterns to profiles*. Nucleic Acids Res, 2003. **31**(1): p. 374-8.
 29. Zhu, J. and M.Q. Zhang, *SCPD: a promoter database of the yeast Saccharomyces cerevisiae*. Bioinformatics, 1999. **15**(7-8): p. 607-11.
 30. Stormo, G.D., *DNA binding sites: representation and discovery*. Bioinformatics, 2000. **16**(1): p. 16-23.
 31. Ren, B., et al., *Genome-wide location and function of DNA binding proteins*. Science, 2000. **290**(5500): p. 2306-9.
 32. Wyrick, J.J. and R.A. Young, *Deciphering gene expression regulatory networks*. Curr Opin Genet Dev, 2002. **12**(2): p. 130-6.
 33. Sandve, G.K. and F. Drablos, *A survey of motif discovery methods in an integrated framework*. Biol Direct, 2006. **1**: p. 11.
 34. Iyer, V.R., et al., *Genomic binding sites of the yeast cell-cycle transcription factors SBF and MBF*. Nature, 2001. **409**(6819): p. 533-8.

35. Simon, I., et al., *Serial regulation of transcriptional regulators in the yeast cell cycle*. Cell, 2001. **106**(6): p. 697-708.
36. Lee, T.I., et al., *Transcriptional regulatory networks in Saccharomyces cerevisiae*. Science, 2002. **298**(5594): p. 799-804.
37. Ren, B., et al., *E2F integrates cell cycle progression with DNA repair, replication, and G(2)/M checkpoints*. Genes Dev, 2002. **16**(2): p. 245-56.
38. Horak, C.E., et al., *GATA-1 binding sites mapped in the beta-globin locus by using mammalian chIp-chip analysis*. Proc Natl Acad Sci U S A, 2002. **99**(5): p. 2924-9.
39. Weinmann, A.S., et al., *Isolating human transcription factor targets by coupling chromatin immunoprecipitation and CpG island microarray analysis*. Genes Dev, 2002. **16**(2): p. 235-44.
40. Hughes, J.D., et al., *Computational identification of cis-regulatory elements associated with groups of functionally related genes in Saccharomyces cerevisiae*. J Mol Biol, 2000. **296**(5): p. 1205-14.
41. Bailey, T.L. and C. Elkan, *The value of prior knowledge in discovering motifs with MEME*. Proc Int Conf Intell Syst Mol Biol, 1995. **3**: p. 21-9.
42. Hertz, G.Z. and G.D. Stormo, *Identifying DNA and protein patterns with statistically significant alignments of multiple sequences*. Bioinformatics, 1999. **15**(7-8): p. 563-77.
43. Thijs, G., et al., *A higher-order background model improves the detection of promoter regulatory elements by Gibbs sampling*. Bioinformatics, 2001. **17**(12): p. 1113-22.
44. Sinha, S. and M. Tompa, *YMF: A program for discovery of novel transcription factor binding sites by statistical overrepresentation*. Nucleic Acids Res, 2003. **31**(13): p. 3586-8.

45. Frith, M.C., et al., *Finding functional sequence elements by multiple local alignment*.
Nucleic Acids Res, 2004. **32**(1): p. 189-200.
46. Bulyk, M.L., *Computational prediction of transcription-factor binding site locations*.
Genome Biol, 2003. **5**(1): p. 201.
47. Ohler, U. and H. Niemann, *Identification and analysis of eukaryotic promoters: recent computational approaches*. Trends Genet, 2001. **17**(2): p. 56-60.
48. Pavesi, G., G. Mauri, and G. Pesole, *In silico representation and discovery of transcription factor binding sites*. Brief Bioinform, 2004. **5**(3): p. 217-36.
49. Atteson, K., *Calculating the exact probability of language-like patterns in biomolecular sequences*. Proc Int Conf Intell Syst Mol Biol, 1998. **6**: p. 17-24.
50. Brazma, A., et al., *Predicting gene regulatory elements in silico on a genomic scale*.
Genome Res, 1998. **8**(11): p. 1202-15.
51. Jensen, L.J. and S. Knudsen, *Automatic discovery of regulatory patterns in promoter regions based on whole cell expression data and functional annotation*. Bioinformatics, 2000. **16**(4): p. 326-33.
52. Tompa, M., *An exact method for finding short motifs in sequences, with application to the ribosome binding site problem*. Proc Int Conf Intell Syst Mol Biol, 1999: p. 262-71.
53. Bailey, T.L. and C. Elkan, *Fitting a mixture model by expectation maximization to discover motifs in biopolymers*. Proc Int Conf Intell Syst Mol Biol, 1994. **2**: p. 28-36.
54. Cardon, L.R. and G.D. Stormo, *Expectation maximization algorithm for identifying protein-binding sites with variable lengths from unaligned DNA fragments*. J Mol Biol, 1992. **223**(1): p. 159-70.

55. Lawrence, C.E. and A.A. Reilly, *An expectation maximization (EM) algorithm for the identification and characterization of common sites in unaligned biopolymer sequences*. Proteins, 1990. **7**(1): p. 41-51.
56. Moses, A.M., D.Y. Chiang, and M.B. Eisen, *Phylogenetic motif detection by expectation-maximization on evolutionary mixtures*. Pac Symp Biocomput, 2004: p. 324-35.
57. Qi, Y., P. Ye, and J.S. Bader, *Genetic Interaction Motif Finding by expectation maximization--a novel statistical model for inferring gene modules from synthetic lethality*. BMC Bioinformatics, 2005. **6**: p. 288.
58. Favorov, A.V., et al., *A Gibbs sampler for identification of symmetrically structured, spaced DNA motifs with improved estimation of the signal length*. Bioinformatics, 2005. **21**(10): p. 2240-5.
59. Lawrence, C.E., et al., *Detecting subtle sequence signals: a Gibbs sampling strategy for multiple alignment*. Science, 1993. **262**(5131): p. 208-14.
60. Meireles, L.M. and T. Akutsu, *A Gibbs sampling approach to detection of tree motifs*. Genome Inform Ser Workshop Genome Inform, 2005. **16**(1): p. 34-43.
61. Siddharthan, R., E.D. Siggia, and E. van Nimwegen, *PhyloGibbs: a Gibbs sampling motif finder that incorporates phylogeny*. PLoS Comput Biol, 2005. **1**(7): p. e67.
62. Thijs, G., et al., *A Gibbs sampling method to detect overrepresented motifs in the upstream regions of coexpressed genes*. J Comput Biol, 2002. **9**(2): p. 447-64.
63. Liu, X., D.L. Brutlag, and J.S. Liu, *BioProspector: discovering conserved DNA motifs in upstream regulatory regions of co-expressed genes*. Pac Symp Biocomput, 2001: p.

- 127-38.
64. Cheremushkin, E. and A. Kel, *Whole genome human/mouse phylogenetic footprinting of potential transcription regulatory signals*. Pac Symp Biocomput, 2003: p. 291-302.
 65. Blanchette, M. and M. Tompa, *Discovery of regulatory elements by a computational method for phylogenetic footprinting*. Genome Res, 2002. **12**(5): p. 739-48.
 66. Yan, B., et al., *Computational prediction of conserved operons and phylogenetic footprinting of transcription regulatory elements in the metal-reducing bacterial family Geobacteraceae*. J Theor Biol, 2004. **230**(1): p. 133-44.
 67. Van Hellefont, R., et al., *A novel approach to identifying regulatory motifs in distantly related genomes*. Genome Biol, 2005. **6**(13): p. R113.
 68. Lenhard, B., et al., *Identification of conserved regulatory elements by comparative genome analysis*. J Biol, 2003. **2**(2): p. 13.
 69. McCue, L., et al., *Phylogenetic footprinting of transcription factor binding sites in proteobacterial genomes*. Nucleic Acids Res, 2001. **29**(3): p. 774-82.
 70. McCue, L.A., et al., *Factors influencing the identification of transcription factor binding sites by cross-species comparison*. Genome Res, 2002. **12**(10): p. 1523-32.
 71. Kellis, M., et al., *Sequencing and comparison of yeast species to identify genes and regulatory elements*. Nature, 2003. **423**(6937): p. 241-54.
 72. Zheng, J., J. Wu, and Z. Sun, *An approach to identify over-represented cis-elements in related sequences*. Nucleic Acids Res, 2003. **31**(7): p. 1995-2005.
 73. Cora, D., et al., *Computational identification of transcription factor binding sites by functional analysis of sets of genes sharing overrepresented upstream motifs*. BMC

- Bioinformatics, 2004. **5**: p. 57.
74. Caselle, M., F. Di Cunto, and P. Provero, *Correlating overrepresented upstream motifs to gene expression: a computational approach to regulatory element discovery in eukaryotes*. BMC Bioinformatics, 2002. **3**: p. 7.
75. Durbin, R., *Biological Sequence Analysis*. 1998.
76. Mount, D.W., *Bioinformatics: Sequence and Genome Analysis*. Cold Spring Harbor Laboratory Press, 2004.
77. Qiu, P., *Recent advances in computational promoter analysis in understanding the transcriptional regulatory network*. Biochem Biophys Res Commun, 2003. **309**(3): p. 495-501.
78. Fogel, G.B., et al., *Discovery of sequence motifs related to coexpression of genes using evolutionary computation*. Nucleic Acids Res, 2004. **32**(13): p. 3826-35.
79. Haverty, P.M., U. Hansen, and Z. Weng, *Computational inference of transcriptional regulatory networks from expression profiling and transcription factor binding site identification*. Nucleic Acids Res, 2004. **32**(1): p. 179-88.
80. Mao, L., et al., *Combining microarray and genomic data to predict DNA binding motifs*. Microbiology, 2005. **151**(Pt 10): p. 3197-213.
81. Conlon, E.M., et al., *Integrating regulatory motif discovery and genome-wide expression analysis*. Proc Natl Acad Sci U S A, 2003. **100**(6): p. 3339-44.
82. Wasserman, W.W. and A. Sandelin, *Applied bioinformatics for the identification of regulatory elements*. Nat Rev Genet, 2004. **5**(4): p. 276-87.
83. Sinha, S., M. Blanchette, and M. Tompa, *PhyME: a probabilistic algorithm for finding*

- motifs in sets of orthologous sequences*. BMC Bioinformatics, 2004. **5**: p. 170.
84. Blanchette, M., B. Schwikowski, and M. Tompa, *Algorithms for phylogenetic footprinting*. J Comput Biol, 2002. **9**(2): p. 211-23.
85. Prakash, A. and M. Tompa, *Discovery of regulatory elements in vertebrates through comparative genomics*. Nat Biotechnol, 2005. **23**(10): p. 1249-56.
86. Altschul, S.F., et al., *Basic local alignment search tool*. J Mol Biol, 1990. **215**(3): p. 403-10.
87. Doniger, S.W., J. Huh, and J.C. Fay, *Identification of functional transcription factor binding sites using closely related Saccharomyces species*. Genome Res, 2005. **15**(5): p. 701-9.