MODELING COOPERATIVE GENE REGULATION USING FAST ORTHOGONAL SEARCH

by

IAN MINZ

A thesis submitted to the
Department of Electrical and Computer Engineering
in conformity with the requirements for
the degree of Master of Science (Engineering)

Queen's University
Kingston, Ontario, Canada

August 2008

Copyright © Ian Minz, 2008

Abstract

A number of computational methods have suggested means by which gene transcription – the process through which RNA is created from DNA – is activated, but there are factors at work that no model has been able to fully explain. In eukaryotes, gene regulation is quite complex, so models have primarily focused on a relatively simple species, *Saccharomyces cerevisiae* (budding yeast). Because of the inherent complexity in higher species, and even in yeast, a method of identifying transcription factor (TF) binding motifs (specific, short DNA sequences) must be efficient and thorough in its analysis.

This thesis shows that a method using the Fast Orthogonal Search (FOS) algorithm to uncover binding motifs as well as cooperatively binding groups of motifs can explain variations in gene expression profiles, which reflect the level at which DNA is transcribed into RNA for a number of genes. The algorithm is very fast, exploring a motif list and constructing a final model within seconds to a few minutes. It produces model terms that are consistent with known motifs, while also revealing new motifs and interactions, and it causes impressive reductions in variance with relatively few model terms over the cell-cycle.

Co-authorship

Portions of this thesis were co-written along with Michael Korenberg for use in a presently unpublished paper.

Acknowledgements

I would like to thank my supervisor Michael Korenberg for leading me into this project and especially for his guidance and motivation along the way.

I would like to thank Debopriya Das from the Lawrence Berkeley National Laboratory, who kindly provided data sets and made helpful comments on a paper written based on the work in this thesis.

Contents

Abstract	ii
Co-authorship	iii
Acknowledgements	iv
Contents	v
List of Figures	vii
List of Tables	viii
Abbreviations	ix
Chapter 1: Introduction	1
1.1 Overview of the Field of Computational Biology	1
1.2 Statement of Purpose	2
1.3 Outline of Thesis	4
Chapter 2: Background and Literature Review	5
2.1 Biology Background	5
2.1.1 DNA	5
2.1.2 Transcription and Translation	7
2.1.3 Basics of Gene Regulation	9
2.1.4 Transcription Factors and Gene Regulation	11
2.1.5 Cell Cycle	12
2.2 Microarrays and DNA Sequences	13
2.3 Previous Methods	14
Chapter 3: Nonlinear System Identification	17
3.1 Overview of Fast Orthogonal Search	17
3.3 Stopping Criterion	23
Chapter 4: Methods	25
4.1 General	25
4.2 Data Sets	27
4.3 Applying Fast Orthogonal Search	29
4.4 Input Motif Lists	32

4.5 Considering Groups of Motifs.....	34
4.6 Reduction In Variance	36
4.7 Use of Splines	37
Chapter 5: Implementation and Results	38
5.1 General	38
5.2 Accuracy of Resulting Models.....	39
5.3 Significant Motifs and Groups of Motifs	43
5.4 Considering Location of Potential Binding Sites	50
5.5 Running Time	53
5.6 Cell-Cycle.....	55
5.7 Noise Levels and the Validity of the Results	62
Chapter 6: Conclusions.....	64
6.1 Summary	64
6.2 Future Work.....	66
Bibliography	69

List of Figures

Fig. #	Title	Page
Figure 2.1	Structure of DNA	6
Figure 2.2	Flowchart of Transcription/Translation	8
Figure 2.3	Illustration of Transcription Factor Binding	11
Figure 2.4	Cell Cycle Timeline	12
Figure 4.1	Flowchart of Data Processing	26
Figure 4.2	Sample of a Gene Expression Profile	28
Figure 4.3	Sample Yeast Promoter Sequence	29
Figure 4.4	Sample Word Count Profile	30
Figure 5.1	Important Motifs (5-mers, 6-mers, 7-mers)	47
Figure 5.2	Important Motifs (9bp skeletons)	48
Figure 5.3	Important Motifs (10bp skeletons)	49
Figure 5.4	Cell-cycle Plot of <code>acgcgt</code> Activity	58
Figure 5.5	Cell-cycle Plot of <code>aaacaa</code> Activity	59
Figure 5.6	Cell-cycle Plot of <code>ctaagc</code> Activity	59
Figure 5.7	Cell-cycle Plot of <code>ncctcgn</code> Activity	60
Figure 5.8	Cell-cycle Plot of <code>agggg/ctcga</code> Activity	60
Figure 5.9	Cell-cycle Plot of <code>gtaat/gtaat</code> Activity	61
Figure 5.10	Cell-cycle Plot of Expansion Coefficient	62

List of Tables

Table #	Title	Page
Table 5.1	Summary of Modeling Results	42
Table 5.2	Comparison of Different Location Functions	53
Table 5.3	Running Times	55

Abbreviations

BP	Base pair
CV	Cross-validation
DNA	Deoxyribonucleic Acid
FOS	Fast Orthogonal Search
<i>k</i> -mer	indicates <i>k</i> nucleotides (subunits) in a molecular chain
LS	Least Squares
MARS	Multivariate Adaptive Regression Splines
MCB	MLul Cell-cycle Box
mRNA	messenger Ribonucleic Acid
MSE	Mean-Squared Error
PWM	Position Weight Matrix
RIV	Reduction In Variance
RNA	Ribonucleic Acid
TF	Transcription Factor

Chapter 1: Introduction

1.1 Overview of the Field of Computational Biology

With the emergence of new technologies for acquiring biological data, the rate at which we are learning about genetic pathways has greatly increased. The invention of the microarray for gathering gene expression data has provided an abundant source of information previously unavailable to researchers. With the full sequencing of entire genomes, we are getting closer to understanding exactly how variations in genetic code determine the vast differences amongst all living things. The ability to easily gather genetic information has generated vast publicly-available libraries of gene sequences and experimental data. The massive amounts of data have created a need for computational methods that can extract meaning and propose answers to how DNA and proteins interact.

In part, with the goal of analyzing these large amounts of data, the field of computational biology has developed to create tools to aid biologists in their understanding of genetic mechanisms at the molecular level. Computational biology

generally refers to the application of computer science, mathematics and statistics techniques to the solution of biological problems. One such problem is the determination of binding sites in DNA.

1.2 Statement of Purpose

Gene regulation is not a fully understood process, and this has spurred recent important work [e.g. DAS04,06,07a]. The way in which genes are switched on and off within a cell is not accurately known. Research has found that proteins called transcription factors (TFs) bind to DNA in order to promote or repress that gene's transcription rate, the rate at which the gene's RNA is created from its DNA. Certain short sequences in the DNA called motifs have been discovered as binding sites to which these TFs attach. However, gene expression levels have not been accurately predicted solely as a function of these discovered motifs. Computational methods have been developed to more rigorously analyze the possible ways in which these motifs cause gene expression levels to vary. However, most previous methods have either been inaccurate predictors of the variations in gene expression data, or show other weaknesses, such as computational complexity. The following work was done under the hypothesis that a method able to identify motifs that are highly probable TF-binding sites, while improving upon other methods in running time and/or in accuracy of

describing the variations in gene expression, could be created by using the Fast Orthogonal Search (FOS) algorithm.

While the following work was done using yeast cell-cycle data for developing models, the method is highly applicable in a number of more relevant areas. The Multivariate Adaptive Regression Splines (MARS) method, for instance, has already been extended to identify transcription factor binding sites in humans and very recently has been used in finding motifs specific to hepatocellular carcinomas [DAS06]. Disease prevention and treatment can only be developed with increased understanding of the diseases' pathways. Methods that can bring to light the causes of disease are of great importance to the medical community. Yeast is an excellent organism on which to test a method, due to its relative simplicity. Like humans, yeast is a eukaryote (its cells contain its genetic material inside a nucleus), and thus has enough in common, genetically, for it to be considered a worthy species for scientific study by many researchers [DAS04; PIL01; BUS01].

Three main measures of this method's performance were deemed important in its design: the accuracy of the models, their simplicity, and the running time. The objective was to create concise prediction models with high levels of correlation to the expression levels, while lowering the amount of time required by previous methods. Models able to accurately predict expression levels with very few terms are obviously far more desirable for several reasons. Firstly, results from other studies have shown that fairly concise models could explain variations in expression level well [DAS04].

Secondly, a large number of terms in a model could mean a small contribution by each, so that those motifs' contributions would likely be statistically insignificant. By the same token, a large contribution by a small number of motifs or groups of motifs would mean each is more likely to be predictive.

1.3 Outline of Thesis

This thesis is comprised of six chapters. This first chapter was used to introduce the purpose and scope of this work and to give a brief overview of the work described in the subsequent chapters. In the second chapter, some closely-related previous work in gene expression modeling is summarized. A brief section on the basics of gene expression itself is included to allow the reader to understand some fundamentals of genetics to better appreciate the application of this work. The third chapter explains the nonlinear system identification algorithm, Fast Orthogonal Search (FOS), which is used for modeling gene expression. The fourth chapter is devoted to explaining the methodology used in creating models. The data sets and the way in which FOS was implemented are explained. The fifth chapter contains the results of modeling the gene expression data. This includes motifs of significance, quantitative measures of the different models produced, as well as other quantitative measures of the method's ability to model genetic data. The sixth chapter discusses the relevance of the work and mentions potential extensions for future related research.

Chapter 2: Background and Literature Review

2.1 Biology Background

2.1.1 DNA

Deoxyribonucleic Acid (DNA) is the molecule found in cells of living organisms that contains the information required to build all functional elements of cells. DNA exists in long strands and consists of a molecular chain of nucleotides, each composed of a sugar, a phosphate, and a nitrogen base. The base can be only one of four types: adenine (A), cytosine (C), guanine (G) or thymine (T). The DNA sequence (i.e. the sequence of nucleotides forming the DNA strand) varies from one organism to the next, though many similarities are found among all of life. The DNA sequence of humans is roughly 3 billion bases long. Lower species' genomes (the total of an organism's genetic material) can be much smaller, though genome size is not a measure of complexity, as is evident in the massive genome of the single-celled amoeba. The structure of DNA can be seen in Figure 2.1. Strands pair together in such a way that every nucleotide in one

strand pairs to its complement. The nucleotide bases A and T pair together, while C and G pair with each other. The double helix structure occurs as the two complementary strands of DNA wrap around each other, being bound by the complementary nucleotides in their sequence. Every organism thus contains DNA material with one code, but in two complementary forms.

Figure 2.1: Illustration of a short segment of DNA showing the binding of two strands of DNA. The two complementary strands wrap around one another and are bound by the hydrogen bonds between nucleotide base pairs A-T and C-G.

The genome can be divided up into smaller functional blocks of code, called genes, each of which contains the information to create a specific protein. Consensus on the exact number of genes in the human genome has still not been reached, though

many estimate that there are approximately 25,000. The yeast genome contains just over 6,000. Genes vary in length and in structure. A gene is often subdivided into numerous blocks of coding and non-coding sequences, called exons and introns, respectively. The number of each is quite variable from one gene to the next. All genes also contain a portion of non-coding DNA (DNA that is not copied into RNA) called the promoter region upstream (ahead, in sequence) of the first coding block. This region contains certain regulatory sequences that control the rate at which genes are transcribed.

2.1.2 Transcription and Translation

Transcription refers to the process during which RNA (Ribonucleic Acid) is created from DNA. Through a complex series of steps that take place at the molecular level, genes' DNA fragments are replicated. The double-stranded DNA breaks apart and one of the strands (sometimes called the "anti-sense strand") is used as a template for RNA polymerization (RNA creation). RNA is what is used to create proteins and differs from DNA in the backbone sugar of each nucleotide: nucleotides in DNA contain the deoxyribose sugar, while nucleotides in RNA contain ribose. RNA also appears as a long strand of nucleotides, but is built from a slightly different alphabet (adenine, cytosine, guanine and uracil).

Figure 2.2 shows the basic steps taken by the cell to produce proteins from DNA. A gene's promoter region determines when and how the gene's DNA will be transcribed into pre-mRNA, the earliest stage of messenger RNA. Messenger RNA (mRNA) is an RNA molecule encoding for a protein, and it carries the code from the nucleus (where transcription occurs) to the ribosomes (location of protein synthesis, i.e. translation). In the transcription step, the entire gene, excluding this promoter region, is copied. After the transcription process has created a pre-mRNA strand, these pre-RNA strands are spliced to remove non-coding portions called introns, resulting in a mature mRNA strand. It has been shown that the introns contain regulatory sequences that control the splicing process, analogously to the regulation of the transcription step by the promoter region [DAS07b]. The mature mRNA sequence is then translated into a protein via the genetic code. While RNA can be a final functional product, it is often the proteins that serve functional purposes within cells. While different definitions for "gene expression" exist, it will always be used in this thesis to refer to the level at which a gene's pre-mRNA has been transcribed from its DNA.

Figure 2.2: Flowchart showing the basic steps taken to create a protein from DNA for a given gene.

Proteins are strings of amino acids that each correspond directly to a group of three consecutive nucleotides called a codon. There are 20 known amino acids, and the groups of nucleotides that code for each is known. Since there are $4^3=64$ possible codons (sequences of three nucleotides), it is clear that a number of distinct sequences correspond to the same amino acid. This is sometimes called a degenerate code. Proteins have many functions in the cell, including acting as transcription factors.

2.1.3 Basics of Gene Regulation

Each gene is made up of strings of coding DNA and non-coding DNA. The former is the information that is transcribed into mRNA, which codes for proteins. The function of a particular protein created from the coding DNA depends on its unique molecular configuration – its 3-dimensional structure. Proteins are important in regulating thousands of biological processes, and occur in different concentrations in cells depending on the function of the cell. The way in which genes correspond to proteins has long been understood. Leaving out the non-coding part of the genes, the DNA sequence of a gene is basically a stencil that can be used to create identical copies of a specific protein. Many entire genomes have been fully sequenced, and many proteins and the genes that code for them have been identified and studied.

What is not as well understood is the function of the non-coding portions of genes. At one point, it was believed that large portions of DNA were “junk DNA”, serving

no purpose. However, it has been discovered that there are regulator sequences incorporated into the non-coding portions of genes that are important for controlling the rate at which the coding portions are transcribed into mRNA. Figure 2.3 illustrates the function of these regulator sequences. A number of short regulator sequences, called motifs, have been identified and experimentally tested. These sequences have been found to attract specific proteins called transcription factors (TFs), that bind to the motifs and act as catalysts that speed up (or slow down) the transcription process of the gene in which the motif to which it is bound is found. It has been noted that these motifs are found directly ahead, in sequence, of the beginning of the coding portion of genes, and these regions are mostly less than 800 nucleotides long [KEL02]. Many agree that the vast majority of regulatory elements (motifs) are found within the 600 nucleotides immediately upstream of the transcription start site [DAS04,07a; BUS01; SPE98]. This region is referred to as the promoter region of a gene, due to its role in promoting (and repressing) gene expression. Researchers have been able to pinpoint these regions by determining which regions are transcribed to mRNA. As the genomes of many species have been fully sequenced, the regions just upstream of the coding portions of DNA have been identified as promoters. The promoter regions of yeast, for example, have been identified, and their sequences are publicly available for use.

Figure 2.3: Illustration of gene transcription. Transcription factors binding to target motifs in the promoter region also bind to RNA polymerase, an enzyme (protein catalyzing biological reactions) required for transcription of DNA to RNA. This binding signals for RNA polymerase to bind to the transcription start site (first nucleotide of the coding sequence) and the transcription process initiates, with RNA corresponding to the coding sequence being created. A gene lacking the motifs in its promoter region will not be transcribed due to RNA polymerase not being recruited.

2.1.4 Transcription Factors and Gene Regulation

Transcription factors help to regulate gene expression by binding to a DNA sequence and either recruiting RNA polymerase (the enzyme causing transcription to take place), thus acting as an activator of the transcription process, or by blocking access to RNA polymerase, thus acting as a repressor. TFs generally contain several functional groups within each molecule. A portion called the DNA-binding domain is formed to attach to specific DNA sequences. Only sequences that closely (or exactly) fit this domain will bind to the TF. Binding generally occurs via a combination of hydrogen bonds and Van der Waals forces between the DNA and the TF and forces between each base and the TF can vary in strength from strong to non-existent [ROM05]. The other important portions of TFs are the trans-activating domain and the signal-sensing

domain, which are important for binding to and sensing other proteins that aid in regulating gene expression [LAT97].

2.1.5 Cell Cycle

Transcription must be highly regulated in order for a cell to remain viable. A lack of a certain element could prevent a necessary process from occurring, leading to abnormalities or cell death. At the molecular level, the cell functions using a highly ordered chain of events, whereby each step is crucial to the proper carrying out of the next step. The eukaryotic cell cycle consists of four phases: M (mitosis), G₁ (preparation for S phase and growth), S (synthesis, DNA replication), G₂ (preparation for mitosis) [KEL02]. Gene regulation maintains the integrity of the cell by controlling the timing of all cell cycle-related events.

Figure 2.4: The phases of the cell cycle can be seen here. Mitosis (M), Interphase I (G₁), Synthesis (S) and Interphase II (G₂) are the four phases through which a cell passes.

2.2 Microarrays and DNA Sequences

Microarray technology has enabled us to monitor gene expression activity in a very efficient way, allowing us to simultaneously gather a quantitative measure of the levels at which thousands of genes transcribe. Short sequences (often 25-60 nucleotides long) called probes are attached to the substrate in order to attract a target sequence. By printing probes to which specific mRNA strands will bind onto a chemical matrix, one can directly measure the levels of DNA transcription. Targets are marked with a fluorophore tag, so that the level of binding on the chip can be read using stimulation by light of a specific wavelength to obtain a quantitative measure of the concentration of the target genes' mRNA. Microarray technology is very new, with the first uses only reported in 1995 [SCH95].

DNA sequencing refers to the determination of the order of the nucleotides on a DNA strand. Only in the last decade has the full sequencing of the human genome been completed, and while a portion of our genes have been identified, a great deal of information remains hidden in our genome. Knowing the sequence of a DNA strand is not enough to know its function. Although the sequences of a genome might be known, understanding the function of the code requires further analysis. Researchers have sequenced a number of other genomes, including yeast. Computational tools have been used to analyze these sequences, mapping out genes within the overall sequence as well as designating more specific regions within genes. An example of such a designation is the identification of the promoter regions in yeast.

2.3 Previous Methods

A number of methods have been developed and applied to analyzing gene regulation. A traditional approach was to cluster genes that have similar gene expression profiles over a number of experimental conditions and then decide which motifs are over-represented among these genes [KAT04; KEL02; SPE98; CON03; EIS98]. These methods have been important in identifying several motifs that have been experimentally tested. While these are good starting methods to finding potentially active motif sequences, clustering is known to introduce a loss in information into models since genes within a cluster may lack motifs shared by other genes [BUS01]. Many genes do not cluster tightly enough to allow for the identification of their regulatory elements with reasonable accuracy [DAS07a]. Classifying genes into clusters also can prevent the discovery of motif pairs or groups that work together in regulating gene expression. Another source of noise comes from the fact that certain genes in a cluster might be affected directly by TFs, while other genes in that cluster might show similar expression profiles due to secondary effects [DAS07a]. Finally, a clustering method is dependent on data from multiple experimental conditions, which are not always available. It is often desirable to analyze only one experiment, such as a single time point in a cell-cycle experiment.

A number of methods have used linear regression to determine motifs whose number of appearances in a particular gene's promoter region correlates with a rise or fall in that gene's expression level [BUS01; CON03]. Several ways of going about finding

motifs have been implemented. An extension on earlier methods was the idea of cooperative regulation of transcription, whereby pairs of transcription factors would bind to two motifs appearing in a single gene's promoter region in order to induce transcription to take place [DAS04,07a; BUS01; KEL02]. A method that largely motivated this research was developed by Das and Zhang [DAS04,07a] called MARSMotif, where MARS stands for Multivariate Adaptive Regression Splines [FRI91]. This method emphasizes identifying the synergy of multiple transcription factors. Transcription regulation is a complex process and is inherently cooperative, and MARSMotif was developed to be able to capture this cooperativity and identify pairs or larger cooperating groups of motifs. It is clear that in higher eukaryotes, such as humans, cooperativity is crucial [DAS04,06,07a]. In yeast, a very simple eukaryote, transcription is known to be partially regulated by single transcription factors.

The MARS-based method starts with a set of candidate motifs and builds models by multiplying the functions of motif counts (i.e. the number of appearances of a motif in the promoter region of a gene) by spline basis functions. These splines, which are shaped like piecewise linear functions (i.e. zero below a threshold and linear above the threshold, or vice versa), are used to only consider motifs if their score is above (or below) a certain threshold so as to take into account the possibility that several instances of a motif are required in order for the transcription rate to be affected. MARSMotif found these basis functions to be important in modeling a "switch-like" behavior in transcription regulation. MARSMotif added motifs and pairs of motifs or three-way interactions to the model sequentially in order to best describe the

expression data. Because enormous amounts of computing power are required to analyze each possible pair of motifs, Das used a multi-step method in order to narrow down the number of candidates given for MARS to analyze. For the same data set used here, MARSMotif was able to create models with a percent reduction in variance (%RIV) ranging from 13.9-32.9% [DAS04,07a], when the average was taken over 77 experiments, with some %RIVs reaching ~50%, a clear improvement over all previous methods (approximately 1.5 to 3.5 times the RIV achieved by REDUCE [BUS01], according to Das et. al. [DAS04,07a]). More details on the RIV statistic are given in section 4.7. While a number of methods have focused on modeling regulation of the genes classified as cell-cycle-related, a number of methods have attempted to model genome-wide regulation. Beer and Tavazoie used Bayesian networks to predict mRNA expression patterns of 2,587 genes in *S. cerevisiae* [BEE04]. They found that their method could correctly predict expression patterns for 73% of genes. Yuan et. al. reexamined their work and found a small increase, being able to predict 79% of genes expression profiles [YUA07]. Both studies did show that there are regulators that only apply to a subset of genes and that regulation is not uniformly controlled across entire genomes. Focusing solely on cell-cycle-related genes, the resulting models should include motifs specific to cell-cycle regulation. It is known that groups of genes are co-expressed, and genes that code for products required at the same time in the cell will likely contain shared motifs so that they will be bound by the same TFs [BEE04].

Chapter 3: Nonlinear System Identification

3.1 Overview of Fast Orthogonal Search

Fast Orthogonal Search (FOS) [KOR87,89] is a system identification algorithm that operates by searching through a list of pre-designated candidate functions and iteratively adding the term that lowers the mean-squared error (MSE) of the model by the greatest amount. The algorithm has been shown to have success in modeling nonlinear systems of various kinds. FOS has been applied to a wide array of applications, including Raman spectroscopy [KOR97], analysis of insulin rhythms [SHA97] and estimating speed of AC induction motors [MCG06], to name a few.

The models developed by FOS are of the form:

$$y(n) = \sum_{m=0}^M a_m p_m(n) + e(n) \quad (3.1.1)$$

where $p_0(n) = 1$ and for $m > 0$ the $p_m(n)$ are the non-orthogonal candidate functions which were selected to be included in the model, and the a_m are the associated coefficients which best fit the output (i.e., are least-square (LS) estimates). Finally, $y(n)$ is the output of the system being modeled, $e(n)$ is the model error, and M is the number of (non-constant) terms included in the model and is not held fixed. Since a_0 is a least-square estimate and $p_0(n) = 1$, it follows that the average value of $e(n)$ over all points n used to identify the model is zero. Because a constant term is first added to the model prior to adding any other terms, the model can alternatively be written as:

$$y(n) = a_0 + \sum_{m=1}^M a_m p_m(n) + e(n) \quad (3.1.2)$$

It is important to note that, as stated above, the candidate functions, $p_m(n)$, are not assumed to be orthogonal functions.

3.2 Selecting Model Terms

Fast Orthogonal Search [KOR87,89] is related to an algorithm by Desrochers [DES81] for selecting terms for a nonlinear model. However, amongst various differences, the computational cost and memory storage requirements of the latter algorithm are proportional to the square of the number of candidate functions whereas in FOS they depend linearly on the number.

Using fast orthogonal search, we can efficiently build up, implicitly, an economical series representation of the form

$$y(n) = \sum_{m=0}^M g_m w_m(n) + e(n) \quad (3.2.1)$$

where the $w_m(n)$ are mutually orthogonal over the data record, and the g_m are the orthogonal expansion coefficients achieving a least-squares fit.

Let the over-bar below denote the time-average from $n = 0$ to $n = N - 1$, where N is the length of the time series. Then finding a_m to minimize the MSE

$$e_1 = \overline{\left(y(n) - \sum_{m=0}^M a_m p_m(n) \right)^2} \quad (3.2.2)$$

is equivalent to finding g_m to minimize

$$e_2 = \overline{\left(y(n) - \sum_{m=0}^M g_m w_m(n) \right)^2}, \quad (3.2.3)$$

i.e., $e_1 = e_2$ when both have been minimized. The minimum occurs when

$$g_l = \frac{\overline{y(n)w_l(n)}}{\overline{w_l^2(n)}}, \text{ for } l=0,\dots,M, \quad (3.2.4)$$

due to the mutual orthogonality of the $w_m(n)$. This mutual orthogonality also implies that the resulting MSE $e_1 = e_2$ equals

$$\overline{y^2(n)} - \sum_{m=0}^M g_m^2 \overline{w_m^2(n)}. \quad (3.2.5)$$

This shows that the reduction in mean-squared error achievable by selecting any given candidate is readily obtained from the norm of the corresponding orthogonal function and the orthogonal expansion coefficient. A similar observation has been made by Desrochers [DES81]. Suppose we have a model of the form of (3.1.1), consisting of terms $p_0(n), \dots, p_{M-1}(n)$, and we add one further term, $a_M p_M(n)$. In terms of the orthogonalized model, this corresponds to the addition of the term $g_M w_M(n)$, and the associated reduction in mean-squared error is given by

$$Q(M) = g_M^2 \overline{w_M^2(n)} \quad (3.2.6)$$

Therefore, to increase the model size by one further term, from $M - 1$ non-constant terms to M , $p_M(n)$ should be chosen in the following manner. For each candidate function for $p_M(n)$, find the corresponding value of $Q(M)$ according to (3.2.12), then choose the candidate for which $Q(M)$ is greatest, as this will result in the greatest reduction in MSE. Repeating this process to iteratively add terms until no term can reduce the MSE by more than a preset threshold, accurate models of systems can be created efficiently.

The creation of the orthogonal functions $w_m(n)$ is expensive in computing time and memory, but is avoided by FOS. The algorithm uses a slightly modified Cholesky decomposition whereby the coefficients a_m can be directly extracted without the need

to explicitly calculate the orthogonal functions. This leads to increased efficiency in comparison to a typical Gram-Schmidt orthogonalization.

The orthogonal expansion coefficients in (3.2.1) are given by [KOR89]

$$g_m = \frac{C(m)}{D(m,m)}, \text{ for } m=0,\dots,M \quad (3.2.7)$$

where

$$D(0,0) = 1 \quad (3.2.8)$$

$$D(m,0) = \overline{p_m(n)}, \text{ for } m=1,\dots,M \quad (3.2.9)$$

$$D(m,r) = \overline{p_m(n)p_r(n)} - \sum_{i=0}^{r-1} \alpha_{ri} D(m,i), \text{ for } m=1,\dots,M; r=0,\dots,m-1 \quad (3.2.10)$$

$$\alpha_{mr} = \frac{D(m,r)}{D(r,r)}, \text{ for } m=1,\dots,M; r=0,\dots,m-1 \quad (3.2.11)$$

$$C(0) = \overline{y(n)} \quad (3.2.12)$$

$$C(m) = \overline{y(n)p_m(n)} - \sum_{r=0}^{m-1} \alpha_{mr} C(r), \text{ for } m=1,\dots,M \quad (3.2.13)$$

The α_{mr} and $D(m,r)$ can be calculated by the following pseudocode, which achieves a Cholesky factorization:

$$D(0,0) = 1$$

FOR $m = 1$ TO M

 Calculate $D(m,0)$ from (3.2.9)

FOR $m = 1$ TO M

FOR $r = 0$ TO $m - 1$

Calculate α_{mr} from (3.2.11)

Calculate $D(m, r + 1)$ from (3.2.10)

Once the g_m and α_{mr} are known, the coefficients a_m in (3.1.1) can be obtained by [KOR87,88,89]:

$$a_m = \sum_{i=m}^M g_i v_i \quad (3.2.14)$$

where

$$v_m = 1 \quad (3.2.15)$$

$$v_i = - \sum_{r=m}^{i-1} \alpha_{ir} v_r; \quad i = m + 1, \dots, M \quad (3.2.16)$$

Note that the reduction in MSE by adding the M -th non-constant term, $p_M(n)$, is equivalently

$$Q(M) = g_M^2 D(M, M). \quad (3.2.17)$$

In selecting $p_M(n)$, we only need to carry out the above pseudocode for $m = M$, which allows us to avoid repeating calculations done for previous values of m . Once $p_M(n)$ has been chosen, the abbreviated pseudocode can be repeated to recalculate $C(M)$ and g_M . This allows the α_{mr} , $D(M, M)$, $C(M)$ and g_M to be properly set prior to searching for

$p_{M+1}(n)$ [KOR89]. This implementation reduces memory storage requirements, an important consideration in searching for regulatory motifs and cooperating groups of motifs, but more efficient code is available [ADE00] with increased memory storage.

3.3 Stopping Criterion

FOS will continue to add terms so long as the addition of some term can further reduce the MSE of the model by more than a preset amount. This threshold is designed to assure that models are built using only terms that have a significant correlation to the output of the system, keeping models concise. Maintaining a higher threshold also reduces the amount of noise fitting that takes place in the modeling. Any term unable to adequately improve the accuracy of the model will not be included in the model. The threshold is of the form [KOR91]:

$$\Delta_{min} = \frac{K}{N} \cdot \left[\overline{y^2(n)} - \sum_{m=0}^{M-1} g^2(m) D(m, m) \right] \quad (3.3.1)$$

Here $y(n)$ is the output of the system being modeled, N is the length of the time series, $M - 1$ is the number of (non-constant) terms already in the model and the over-bar again signifies taking the average. The $g(m)$ and $D(m, m)$ are calculated as described in [KOR89]. K is a constant chosen to satisfy a particular confidence interval. For example, to achieve about a 95% confidence interval, K would be chosen to equal 4, as seen in

inequality #25 in [KOR91]. Choosing a value of 10.9 would yield a confidence interval of about 99.9% [KOR97].

Chapter 4: Methods

4.1 General

A program that accepts (1) an array of gene expression data, (2) genes' promoter region sequences and (3) a list of candidate motifs and outputs a list of potential TF binding motifs was created using MATLAB. A flowchart giving a high-level overview of how these data are processed is shown in Figure 4.1. In contrast with previous methods that have not exploited an orthogonal approach [DAS04,06,07a; BUS01; CHI03; YU06], this method simply uses FOS to correlate the word counts present in the promoter regions of genes to the logarithm of the ratio of those genes' expression levels (the ratio between the test sample and a control). The ratio of the expression level to a control is taken for experimental accuracy. It has been found that taking the log of the ratio leads to better results [DAS04].

Figure 4.1: Flowchart showing an overview of the data processing. A word counter program counts the number of appearances of every motif from the motif list in each gene's promoter region to create a list of M word count profiles (715 genes long, each). This list of word count profiles is treated as the list of candidate functions by Fast Orthogonal Search, which first pre-screens the motifs then builds a model from a reduced set of candidate functions plus cross-products thereof to approximate the given gene expression profile (715 genes long). The model is built and a concise set of significant motifs and their weighting factors (as chosen by FOS) are returned. When tenfold cross-validation was used (Chapter 5), models were built based on 90% of the genes rather than all of them.

The program was mainly designed to analyze one time point at a time. At different time points, gene expression becomes regulated by different transcription factors, and while results are, in general, from the 14 minute time point, the program can analyze any number of time points, by creating separate models that correlate to the expression levels at the corresponding times.

4.2 Data Sets

For this study, the yeast cell-cycle was used for modeling gene regulation. Microarray-gathered gene expression data for 715 cell-cycle related genes of *Saccharomyces cerevisiae*, budding yeast, in 77 experiments were used in the following modeling. These data, along with the sequences of the promoter regions of the same genes, was generously given by Dr. Debopriya Das of the Lawrence Berkeley National Laboratory, and are the same data to which several methods have been previously applied [CHO98; SPE98; BUS01; PIL01; DAS04,07a]. The genes used here are those that have been shown to be cell-cycle-related by Spellman et. al. [SPE98]. In each experiment, corresponding to a specific time point in the cell-cycle, the logarithm of the ratio of each gene's expression level to the expression level of a control experiment was tabulated. A sample of expression levels are plotted in Figure 4.2. The logarithm of the ratio of the expression levels at one time point to a control level are shown across 700 genes.

Figure 4.2: A sample of gene expression data (taken from the 14 minute time point of the alpha-arrest experiment by Cho et. al. [CHO98]. The log ratio of the expression levels for 700 genes to a control are shown here.

It should be noted that in each experiment, the expression levels of a number of genes (5-20 in most experiments) were missing. This, however, should not have affected the results, since a large enough sample of genes was still available for study at each time point. As only one time point was analyzed at a time, the only downside to not having expression data for a specific gene was to slightly reduce the pool of genes. Among the 18 time points from the alpha-arrest cell-cycle experiments, data for no less than 659 genes were available. At the 14 minute time point, the point most used for modeling, data were available for 699 genes.

As noted, promoter sequences of these 715 genes were also obtained. The sequences were limited to a maximum of 600 nucleotides per gene, as distances greater

than this far upstream of the transcription start site were assumed to have no effect on regulation. This length was also taken by a number of other researchers [DAS04,07a; BUS01; SPE98]. Figure 4.3 shows a sample promoter sequence.

```
accgcacactttcatcccactccctccctcctctctgtacactacctcattataaaattgaaacg
atcccaatctcaccgcacttcccataaagccttcacggtatacttttttttctcatctttttttt
ctctctctctctctctctcttttgcgtatctttttcgtcctgtttctttcttacttaagtatctt
tttataatatattaagagagccaaaattatttaaccaaaccaagaatcttgtattctataacgaat
ccttaattcttcttcttcttcttatttaattgccccaaaacagaatttaatttcccgaagaaaaaaa
ttgacttaaaaaatcataggaaaggatgaatgatttacggtacggcgacacggtatcatcaaacg
caagaaagcataccagattgcaaaa
```

Figure 4.3: A sample of a promoter region sequence. The sequence above corresponds to the DNA sequence of the promoter region of the YAL001C gene in yeast, one of the 715 genes used for this study. These 415 nucleotides are found directly upstream (ahead in sequence) of the first coding section of the YAL001C gene.

4.3 Applying Fast Orthogonal Search

The Fast Orthogonal Search algorithm was employed to create concise models based on candidate motifs and motif groups that exhibit word count profiles with strong correlation to gene expression levels. For an example of a word count profile, see Figure 4.4. Unlike previously reported methods, which use various types of linear regression or regression trees [PHU04], FOS implicitly orthogonalizes the terms prior to adding them to the model, ensuring the minimal amount of redundancy in the model and very fast operation [KOR89]. With FOS, a large set of candidates can be quickly explored for the best choice without carrying out a full linear regression for each candidate. Moreover,

the implicit orthogonalization of terms added to the model avoids the need to recalculate previously computed quantities for existing terms. Terms are added to the model sequentially until no candidate is able to improve the MSE beyond a certain threshold. This threshold is based on a standard correlation test requiring that a potential model addition be highly correlated with the unexplained residual at that point, and helps to avoid adding terms that are merely fitting noise [KOR91]. Once the (non-orthogonal) terms have been selected, least-square estimates of their coefficients are given directly as a byproduct of the algorithm [KOR89].

Figure 4.4: A sample of a word count profile. The profile above corresponds to the number of appearances of the short sequence `acgcgt` in the promoter region of a gene, for genes 1 through 634. For this motif, many genes' promoters contained 0-2 copies. Less often were 3 or 4 appearances found in a given gene, and no gene in this subset contained 5 or more appearances of this motif. This word count profile was used as a candidate function as input to FOS.

The set of candidate functions was in part made up of basis functions of the form $c_m(n)$, the count of appearances of the motif m in the gene n , for n in the range $1, \dots, N$. In addition to these $c_m(n)$, functions of pairs of motifs and groups of three motifs were created and added to the set of candidate functions. Section 4.5 outlines the details of how functions of interacting motifs were created. The models developed by FOS are of the form of (3.1.1), i.e.:

$$E(n) = \sum_{m=0}^M a_m p_m(n) + e(n) \quad (4.3.1)$$

where

$$p_0(n) = 1 \quad (4.3.2)$$

and for $m > 0$ the $p_m(n)$ are the non-orthogonal candidate functions (motif count profiles and functions of interacting motifs, defined in section 4.5) which were selected to be included in the model, the a_m are the associated coefficients which best fit the output (i.e., are least-square estimates), $E(n)$ is the log expression ratio of gene n , $e(n)$ is the model error, and M is the number of (non-constant) terms included in the model and is not held fixed.

In summary, to model the gene expression level by motif appearances in the promoter regions of the genes, the gene expression data were treated as a time series, while profiles of the raw word counts of each motif in each gene, and cross-products thereof, were treated as the candidate functions.

4.4 Input Motif Lists

Several motif lists, falling into two categories, were used as lists of candidate motifs from which to build models. Initially, it was desired that all motifs of lengths 5bp (5 base pairs, i.e. 5 nucleotides) to 10bp be searched. However, due to the massive numbers of possible candidate motifs (approximately 250,000 possible 10bp-long motifs were found in the promoter regions of the 715 genes used), lists of degenerate (less rigidly defined by a specific sequence, i.e. including some variable bases) “skeleton” motifs were created. By only designating a certain number of “prongs” (representing fixed bases at specific positions) in each motif list, the number of different motifs in these longer lists was decreased to allow for much more computationally feasible searching. The skeleton lists will hereafter be referred to by names of the form NbpPp (“N base pairs P prongs”), where N is the number of base pairs and P is the number of prongs. An example of a skeleton motif is the following: `actnntcngn`. This motif is 10 base pairs in length and each of the 4 spaces, denoted by “n”, can be any of the four bases (“a”, “c”, “g” or “t”). Due to its length and number of required prongs (six), it would be in the 10bp6p (“10 base pairs 6 prongs”) list.

Three different lists of skeletons were created for each of the 10mers and 9mers. The number of prongs used in each list was 5, 6, or 7. The six skeleton lists created are (i) 10bp7p, (ii) 10bp6p, (iii) 10bp5p, (iv) 9bp7p, (v) 9bp6p, and (vi) 9bp5p. These lists ended up with approximately 4000 skeleton motifs each. While this method of grouping motifs together to create one skeleton representing a family of motifs does result in

information loss, since FOS will then not search each possible 10bp motif for activity, it allows for the possibility that transcription factors are able to bind to any of a family of motifs. Some details about the binding of TFs to DNA are still unknown, and the possibility that spaces in the motifs are necessary must not be overlooked. Methods have analyzed and found potentially functional degenerate motifs [DAS04,06; KEL02; ROM05]. Methods have analyzed oligomers with fixed spacing [CON03]; however, no previous methods have used lists composed of skeleton motifs with irregularly placed prongs. It has been noted that human TFs are likely to bind to much more degenerate motifs than in yeast [DAS06], and so it will become more important to search these types of lists in analyzing human gene expression data.

These lists were created starting with a complete list of all 10bp (or 9bp) possible motifs found in the promoter regions of the set of genes. All motifs were compared to the first motif in this list, and a histogram showing the number of matches at each base position was generated. For 10-mers and 7-pronged skeletons, three positions are allowed to be arbitrary. The three positions at which the least motifs shared a base with the first motif were then allowed to be any of the four bases and a new skeleton motif was built with prongs (i.e., fixed bases) matching the first motif at the other positions. All motifs fitting this skeleton were removed from the list, and the skeleton was added to the list of skeleton motifs. This process was repeated until all motifs in the 10-mer list were represented in the skeleton list. This process was analogously repeated for the five other lists of various forms of skeleton motifs.

In addition to the skeleton motifs, lists of every occurring 5-mer, 6-mer and 7-mer were searched for possibly active binding motifs. While most other methods have inputted lists of motifs found to be significant by previous methods, no assumptions are made here and the program requires no a priori information.

4.5 Considering Groups of Motifs

To analyze pairs of motifs, a function of the two motifs needs to be created. This function can then be treated as a single candidate function to be considered by FOS. There are many ways to treat pairs of motifs. One way to do so is to take, for each gene, the minimum number of appearances between the two motifs in that gene. This assures that the count is not increased unless there are binding sites at which both TFs can simultaneously bind in equal ratio. Another method is to take, for each gene, the product of the two motif counts for that gene. Using this method, if no copies of the first motif are present, the function will be zero for that gene regardless of the number of appearances of the other motif; however, should there be one or more copies of the first motif, increases in the second motif will result in increases in the function.

There are obvious differences in using the two methods. Should expression levels be controlled by a constant ratio of binding TFs, the first method will more accurately capture the synergy. However, the fact that multiple binding sites exist for a single TF should also be incorporated into the model. It should be noted that an increase

in appearances of one motif will still increase the binding probability, and thus will more strongly affect the expression level. For example, if one instance of some motif m_1 is found in gene n , i.e. $c_{m_1}(n)=1$, and two instances of some motif m_2 are found, i.e. $c_{m_2}(n)=2$, the function $c_{m_1,m_2}(n)=c_{m_1}(n)c_{m_2}(n)$ will be equal to 2. If the count of m_2 , $c_{m_2}(n)$, were to change to 4, then $c_{m_1,m_2}(n)$ would become 4, even though the number of pairs of motifs would be 1 in both cases. This method also introduces a 2nd order characteristic to the function of pairs of motifs. In addition to these two methods several variations are of interest as well. By squaring the function in the first method, a third method can be formed which takes on properties of both the first and second methods. All three methods were applied to the 6-mers and the second method was deemed to have better models based on the RIV. For all results, unless otherwise stated, the method of creating functions of multiple motifs should be assumed to be this second method, i.e.

$$c_{m_1,\dots,m_k}(n) = \prod_{i=1}^k c_{m_i}(n), \quad (4.5.1)$$

where $c_{m_i}(n)$ are the word counts of motif m_i in the promoter region of gene n , for $n = 1, \dots, N$, and k is the number of interacting motifs considered in a synergistic group.

4.6 Reduction In Variance

Because the field of gene expression analysis favors reporting reduction in variance (RIV) over mean squared error (MSE), the reduction in variance (RIV) was used as a measure of the created models' abilities to explain variations in the gene expression data, and was calculated by the following equation:

$$\%RIV = \left(1 - \frac{\overline{[E(n) - \hat{E}(n) - \overline{[E(n) - \hat{E}(n)]}]^2}}{\overline{[E(n) - \overline{E(n)}]^2}} \right) \times 100\% \quad (4.6.1)$$

where

$$\hat{E}(n) = \sum_{m=0}^M a_m p_m(n) \quad (4.6.2)$$

and where $E(n)$ is the log expression ratio of gene n at the single time point studied, and the over-bar signifies taking the average over all the genes used to fit the model. Note that in the numerator, inside the mean-square term, the mean

$$\overline{\left[E(n) - \sum_{m=0}^M a_m p_m(n) \right]} = \overline{e(n)} = 0 \quad (4.6.3)$$

over the genes used to identify the model. However, the %RIV will also be used to evaluate the identified model over novel gene expression data, in which cases the over-

bar will denote the average over the new data where the mean in question is not necessarily zero.

4.7 Use of Splines

While the MARS methods used splines as basis functions in order to more specifically choose terms, having functions based on word counts already introduces a switch-like characteristic that the splines are meant to emphasize. When splines were incorporated into the FOS program, only very minimal improvements were found in the models, at the expense of greater complexity and longer running time. Typically, increased reductions in variance on the order of 0-1.5% were achieved when splines were included, while the same binding motifs were found. Because of the fact that each candidate motif needs to be considered using a number of different spline knots (the threshold at which the spline becomes linear), the candidate function list is greatly increased. In this study, it was deemed more valuable to allow a greater variety of candidate motif sequences in lieu of simply increasing the number of candidates by multiplying them by various spline functions.

Chapter 5: Implementation and Results

5.1 General

To apply FOS as the search algorithm, lists of word count profiles from candidate motifs were given to FOS for inspection along with a set of gene expression data from the yeast cell-cycle. Upon completion of the program's execution, FOS outputs a concise list of motifs and groups of motifs that it predicts to have a high probability of acting as binding targets to TFs, along with the reduction in variance (RIV) of the model built by FOS. The majority of the models were created using the expression levels of the alpha-arrest experiment at the 14 minute time point, which takes place during the M/G1 boundary of the cell-cycle [BUS01]. This time point was used in order to compare results to those found by previous methods. The following chapter outlines the results of the modeling of gene regulation using FOS.

5.2 Accuracy of Resulting Models

For each motif list, tenfold cross-validation was used to determine regulatory motifs that repeatedly are found by FOS. Tenfold cross-validation refers to the genes being divided into groups of modeling genes and testing genes ten different ways. For each set of conditions (motif list, order of interactions allowed, pre-screening [pre-ranking] on or off), models were built on 90% of the genes and tested for their fit on the other 10%. This was repeated so as to build a model holding out each of the sets of 10% of the points, making ten models per set of conditions. Following the building of each model, FOS would return a set of motifs and motifs groups used to build that model. The program kept track of how many times each motif or motif group was returned throughout the building of the ten models. A histogram of motif appearances in the models was generated, and those appearing above a certain number of times were considered significant. The number of motif appearances out of 10 is called its CV score. Generally, known motifs appeared 8-10 out of 10 models, so a threshold of 7 appearances was used in order to maintain a high level of rigor in considering motifs to be significant. Those motifs found only 1-3 times are quite possibly false findings, which are found due to their counts' similarities to the noise in the gene expression levels. Those with 7-10 appearances, however, are highly unlikely to be accidentally found.

In general, very concise models were built by FOS. Even with low numbers of terms, the models' reductions in variance (RIVs) ranged from 6.6-32.8%. Most methods have reported lower RIVs and this range is only equaled by MARSMotif, which reported

a very similar range of 13.9-32.9% [DAS04,07a] when the average was taken over 77 experiments, with some %RIVs reaching ~50%. Thus, although there is a difference in the way the accuracy was computed, FOS was able to create accurate models with very few terms. Often, 3-5 motifs or groups (i.e., cross-products) of motifs were able to accurately explain the variation in the genes' expression levels. A number of runs were pre-screened, while a number were not. The pre-screening process involves running FOS on the entire list and pre-ranking the top individual motifs. Then FOS is re-run with only functions of these top individual motifs and their interactions as candidates. This process limits the number of higher order interactions by only considering the possibility of cooperation between motifs that have a high significance level on their own. The number of top terms considered by FOS was generally taken to be 10 (not counting their cross-products), but several trials were executed with 5 or 15 terms. When pre-screening was not used, all possible single motifs were searched, though the prescreening process was still done to limit the number of higher order interactions.

Table 5.1 shows the RIVs associated with the models built from various lists. Reduction in variance is reported in six columns per motif list. The model RIV refers to the RIV calculated based on the 90% of genes used for determining the model, while the test RIV refers to the RIV calculated based on the 10% of genes withheld during modeling. FOS was run once when allowing only single motifs (i.e., no interactions) to be searched (1st and 4th columns for modeling and testing RIV, respectively). FOS was run again with 2nd order interactions of motifs being considered in addition to the single motifs (2nd and 5th columns), and then once more with 3rd order interactions being

added onto the 1st and 2nd order interactions (3rd and 6th columns). The best results appear in **bold**.

The models that achieved a very high RIV but did not perform well when tested on the points that were held out show that FOS very accurately modeled the training expression data, but did not capture a model that explains gene expression in general. In other words, these models have included terms that have only fit the noise in the data. A more accurate measure of the models' accuracy is their ability to explain variations in novel genes' expression levels (i.e. the testing genes – those genes not used to build the model). A high RIV on the testing genes implies that a model has found terms which explain all genes' expression profiles. Strong evidence of the value of the pre-screening process appears from the largely negative %RIV on the testing genes achieved when bypassing the pre-screening process. The majority of the runs that were not pre-screened resulted in models performing poorly on the testing genes. The 5-mers and 6-mers achieved better results than most, but their performance is still enhanced by pre-screening.

Looking now at only pre-screened runs, the 6-mer and 7-mer motifs and the 6-pronged 9bp skeleton motifs had the largest test RIV on average, showing the best ability to explain novel gene expression data, and may be an indication of the number of bases that are typically used in binding to a motif. The 5-mers demonstrate good performance on the testing genes when no interactions or only up to 2nd order interactions are allowed. The %RIV drops off when 3rd order interactions are included, a

sign of noise-fitting terms being added. This drop-off is in agreement with the literature which states that 3rd order interactions tend to be rare in yeast [DAS04].

Table 5.1: Summary of Modeling Results

Pre Screened?	# Of Top Terms	Motif List	%RIV on Modeling Genes			%RIV on Testing Genes		
			1 st	2 nd	3 rd	1 st	2 nd	3 rd
Yes	10	10bp7p	11.77	12.50	12.50	3.36	2.69	2.69
Yes	15	10bp6p	13.72	19.33	20.08	3.72	5.33	6.24
Yes	5	10bp6p	10.70	11.25	11.21	6.27	2.14	2.09
Yes	15	10bp5p	11.68	17.94	19.15	2.61	4.66	1.15
Yes	10	10bp5p	10.94	14.16	14.17	2.35	5.53	4.58
Yes	10	9bp7p	10.37	11.22	11.22	0.83	0.90	0.90
Yes	10	9bp6p	19.11	20.23	20.27	8.61	9.76	9.48
Yes	10	9bp5p	6.58	8.69	11.55	4.30	2.37	0.88
Yes	10	7-mers	24.79	24.79	24.79	14.32	14.32	14.32
Yes	15	6-mers	20.54	28.19	28.75	15.35	13.01	12.25
Yes	10	6-mers	20.05	21.00	21.77	16.27	14.87	15.08
Yes	10	5-mers	17.85	21.52	23.10	13.77	15.18	5.87
Yes	15	5-mers	18.16	24.94	28.84	12.52	8.61	-0.59
No	10	10bp7p	15.96	17.03		-10.65	-9.40	
No	15	10bp6p	16.59	20.23		-4.47	0.59	
No	10	10bp6p	16.59	18.09		-4.47	-2.44	
No	5	10bp6p	16.59	17.00		-4.47	-5.49	
No	15	10bp5p	14.91	20.47		-7.27	0.45	
No	10	10bp5p	14.91	17.66		-7.27	1.25	
No	10	9bp7p	13.41	16.08		-20.89	-18.27	
No	10	9bp6p	21.41	21.90		-2.31	3.16	
No	10	9bp5p	21.51	22.11		-2.53	-3.15	
No	10	7-mers	32.75	32.75		-17.07	-17.07	
No	10	6-mers	22.75	22.58		12.00	12.55	
No	10	5-mers	18.18	22.35		9.64	13.24	
No	15	5-mers	18.18	25.29		9.64	9.00	

5.3 Significant Motifs and Groups of Motifs

FOS returned a number of motifs through analysis of the various motif lists. A histogram of the cross-validation scores of these motifs for the 14-minute time point (at the M/G1 boundary) is shown in Figures 5.1 (5-, 6-, and 7-mers), 5.2 (9bp skeleton motif lists) and 5.3 (10bp skeleton motif lists). In all cases here, pre-screening was used to select the top 10 terms, which, together with cross-products up to 3rd order, were then searched by FOS to find the final models. One motif that is consistently found by FOS is the well-known MCB (MLul Cell-cycle Box) element, a motif that has been found by a number of methods [DAS04,07a; KEL02; BUS01; BAN03; PIL01] and has been studied and experimentally tested. There are several forms of the element whose appearances predict expression levels very accurately. Tests were done to evaluate individual motifs' contributions to %RIV and some of the results are shown in Section 5.6. An effective form of the MCB element, in terms of reduction in variance, was found to be the 5-mer `acgcg` (the form with the largest individual contribution was the 6-mer version `acgcgt`). While another form of the motif, `cgcgt`, occasionally appears in models, `acgcg` is found every time among the models built from the 5-mer list. Under all conditions, cross-validation (CV) shows this motif to appear 10 out of 10 times in these experiments. Another 5-mer that appears consistently is `aaaca`, known as the STE12 element, another experimentally tested motif that has been found by other methods. Two motifs not previously reported are the `gtaat` and `tgttc` motifs, which FOS finds to appear quite regularly. Both consistently appear 8 to 9 times out of 10 through CV

experiments, which hints that these motifs have a high probability of being functional. Among the 5-mers, FOS finds three pairs of motifs as well as two third order sets that consistently appear in the models. One of these pairs, *agggg/ctcga*, is found by FOS every run. CV results show it to appear 10 times under most conditions. This pair of motifs is highly significant, and it is almost certain that the cooperativity between those two motifs is strongly influential on gene expression at the particular time point of the alpha-arrest experiment used. Both elements in the pair have been previously found; the first is the known STRE element, and the second corresponds to a portion of the motif *cctcgac*, which has been reported in the literature [BUS01]. While both have been found, their synergistic functionality has not been previously mentioned. The *gtaat/gtaat* squared term is found 9 times through CV. Slightly less significant is the *tgttc/tgttc* squared term, which appears 8 times under the set of conditions considered here. Two 3rd order groups are found with regularity. While not appearing quite as often as some of the single motifs, there is reason to believe that the *cgcgt/tgttc/gcccg* and *gtaat/gtaat/gtaat* groups, which appear 8 and 7 times through CV, respectively, might be functional in cooperative regulation.

The 6-mers show highly significant motifs as well, with a great deal of corroboration with the 5-mers. Three motifs, *acgcgt*, *aaacaa* and *ctaagc* are found in every model built. The first of these is the most well-known form of the MCB element. The second motif is the STE12 element. It does not seem that the last motif appears in the literature, but there is agreement with other motif findings (skeletons) by FOS. The motif *gttccg* is also consistently found, with a CV score of 9. A small number of pairs

of motifs are found, but no higher-order combinations of 6-mers are found often enough to be worthy of note.

Analysis of the 7-mers shows two motifs of the MCB family that are found by FOS without fail. The `aacgcgt` and `gacgcgt` elements both appear in every model created with the 7-mer list and cross-validation shows the two to appear 10 times on every run. Two other motifs are also found to be in every model: `cctcgac` and `tgccaag`. Another two single 7-mers are found to be highly significant: the motifs `tgagaac` and `tttgaac`. The striking similarities in the last three mentioned motifs indicate that there is likely a degenerate motif to which an important TF can bind. The two similar motifs `tgccaag` and `tgagaac` could be grouped together as `tgmgaas` (where `m` can be either `a` or `c`, and `s` can be either `c` or `g`), while `tgagaac` and `tttgaac` could be variations of the degenerate motif `tnngaac`. No significant pairs of motifs or groups of third order are found among the 7-mers, likely because of the fact that requiring seven straight bases to be held is a fairly rigid constraint.

Turning to the results of the modeling based on the skeleton motifs, let us begin with the least degenerate list, the 9bp7p list. The motif `ngacnaagc` is repeatedly found, with CV scores of either 9 or 10. This motif is a near fit to `tgccaag`, and especially the 6-mer `ctaagc`, both found by FOS. Among this list, FOS also finds one pair of skeleton motifs, `tntacgcnt/atcnactan`, which is quite significant. The first of the two closely fits the 6-mer version of the MCB element, `acgcgt`. Two other individual skeletons are found to repeatedly appear as well (`ggnggnccc` and

acnttggn). Among the 9bp6p list, three notable individual motifs come up 10 times each: acgntcnn, fitting the MCB shape (acgcgt), the skeleton nngntgggt, and ncctcgncn, the latter closely fitting the cctcgac element also found among the 7-mers. Among the 9bp5p list, the STE12 motif aaaca is found in the naaacannn skeleton.

Based on the high level of corroboration between the motifs found by FOS and those found by previous methods, the results empirically demonstrate a convincing ability to model the control of gene expression. In particular, FOS finds some of the most well-known motifs such as the MCB and STE12 elements consistently, so it is clear that FOS is able to correctly determine motifs that have controlling effects on gene expression. A high level of corroboration also exists between the different lists searched by FOS, with many bases being shared among the high-ranking motifs found in different lists. With confidence that motifs found by FOS are correctly discovered, the fact that FOS also finds unknown motifs and groups of motifs with high levels of consistency suggests that it may be able to predict other motifs that could be further analyzed for functionality.

Figure 5.1 – Cross-validation scores of some of the most commonly found 5-, 6-, and 7-mer motifs and groups of motifs by FOS are shown for the 14-minute time point (at the M/G1 boundary). Motifs are separated by lists. Known motifs are marked, while novel findings are not.

- ¹ Known motif (STE12 element)
- [^] Known motif (MCB element)
- ^{*} Known motif (other element)

Figure 5.2 – Cross-validation scores of some of the most commonly found 9bp skeleton motifs and groups of motifs by FOS are shown for the 14-minute time point (at the M/G1 boundary). Motifs are separated by lists. Known motifs are marked, while novel findings are not.

¹ Known motif (STE12 element)

[^] Known motif (MCB element)

Figure 5.3 – Cross-validation scores of some of the most commonly found 10bp skeleton motifs and groups of motifs by FOS are shown for the 14-minute time point (at the M/G1 boundary). Motifs are separated by lists. Known motifs are marked, while novel findings are not.

^ Known motif (MCB element)

5.4 Considering Location of Potential Binding Sites

A great deal of the literature takes into account the distance of potential binding sites from the transcription start point, in the form of position weight matrices (PWMs) [DAS04,07a,KELES]. Because this factor is so often included in the analysis of gene regulation, it seemed worthwhile to consider the effects of discounting the value of motif appearances at a greater distance from this start point. Recall that the promoter sequences were limited to a maximum of 600 nucleotides per gene. Several functions were created in order to compare different ways of discounting the value of word counts as the words' distance from the transcription initiation point increases. Keles et. al. [KEL02] created PWMs where further sites were given less weight, based on an assumption that binding probability decreased with distance from the transcription start site. They implemented PWMs based on various probability density functions. In the work done here, it was decided that PWMs be implemented maintaining the form:

$$L_a(m) = \left(1 - \frac{d_a(m)}{600}\right)^K \quad (5.4.1)$$

where $d_a(m)$ is the distance of appearance a of motif m from the transcription start point (an integer between 1 and 600), and where K is a constant integer used to control how strongly to discount the value of distant potential binding sites. Note that the form of this location function is quite arbitrary, and that many functions could be created to weight positions of potential binding sites. The K -value, for instance, could be fractional, and does not necessarily need to be an integer. The main goal is to discount further sites

in some way, and using integer values is a way of roughly gauging the effect of various degrees of discounting.

Using (5.4.1) to calculate $L_a(m)$, the candidate function $p_m(n)$ representing motif m in gene n would be:

$$p_m(n) = \sum_{a=1}^A L_a(m) \quad (5.4.2)$$

where A is the total number of appearances of motif m in the promoter region of gene n . Note that in all other sections of this thesis, the models were created using $K = 0$, a flat function in which all potential TF-binding sites are considered equally valuable to the overall control process.

A comparison of location functions was done by using different K -values. This comparison was done using only the list of possible 5mer motifs and the results are summarized in Table 5.2. Clearly, the ability of FOS to create models with high RIVs is reduced by decreasing the valuation of more remote potential binding sites. However, use of a K -value of 2 leads to improvement in the *test* RIV for higher order models over use of a K -value of zero. This might be indicative of the fact that at least some binding in third order interactions of TFs is more likely to occur nearer to the transcription start site. While single motifs seem to be more properly modeled by equal treatment of all possible binding sites in the promoter region, in considering cooperatively binding TFs, closer binding is preferred. In a signaling process, one TF binding far from the

transcription start site might recruit another TF to bind somewhere nearer to the site, which might in turn recruit RNA polymerase to the start site to initiate transcription of the gene.

It should be noted, however, that the drop-off in the RIV on testing genes when allowing 3rd order interactions and using a flat ($K=0$) function is in agreement with the MARS-based models. Das et. al. claim not to find many significant 3rd order groups of motifs for yeast. Additionally, the motifs and groups of motifs returned by FOS when location was considered were inconsistent from one model to the next, when proximity to the transcription start site was highly valued. The MCB and STE12 motifs showed up consistently, due to their high individual contributions to the gene expression levels, but for an increasing discounting of more remote potential binding sites, the number of consistently-found motifs dropped off. The models using $K>0$ whose RIVs are shown in Table 5.2 were likely to have a significant amount of noise-fitting motif groups. These results point to the conclusion that, at least in yeast, all locations within the promoter region are roughly equivalent, in terms of allowing binding TFs to have similar control of gene expression.

Table 5.2: Comparison of Different Location Functions (based on 5mer models)

K-value	Pre Screened?	%RIV on Modeling Genes			%RIV on Testing Genes		
		1 st	2 nd	3 rd	1 st	2 nd	3 rd
0	Yes	17.85	21.52	23.10	13.77	15.18	5.87
	No	18.18	22.35		9.64	13.24	
1	Yes	15.03	17.62	20.34	10.27	10.14	9.68
	No	16.21	18.22		7.38	8.05	
2	Yes	13.37	14.96	16.85	10.43	12.83	13.40
	No	13.79	15.54		9.17	11.54	
3	Yes	12.19	13.23	15.01	9.88	10.99	11.97
	No	12.63	14.04		8.41	9.30	
4	Yes	11.07	13.64	13.83	9.45	12.26	10.41
	No	11.51	14.00		8.07	11.29	

5.5 Running Time

One of the advantages of employing FOS as the method of model-building is its speed. Speed is determined by several algorithm and data-related factors: the length of the time series, the MSE threshold below which no terms are added, and the number of candidate functions. Because 715 cell-cycle genes were used, the time series length was constant for all runs. The threshold, Δ_{min} , that had to be exceeded to add the M -th non-constant term was chosen to be:

$$\Delta_{min} = \frac{10.9}{N} \cdot \left[\overline{E^2(n)} - \sum_{m=0}^{M-1} g^2(m) D(m, m) \right] \quad (5.5.1)$$

as explained in section 3.3, where the $E(n)$ are the log expression ratios of the genes n used to identify the model, N is the number of genes, $M - 1$ is the number of (non-

constant) terms already in the model and the over-bar signifies taking the average. The $g(m)$ and $D(m, m)$ are calculated as described in section 3.2 and [KOR89]. The $g(m)$ are the coefficients of the orthogonal functions that are implicitly created from the non-orthogonal $p_m(n)$, and the orthogonality means that existing $g(m)$ do not have to be recalculated when new terms are added to the model. The $D(m, m)$ are equal to the mean-square of the corresponding orthogonal functions implicitly created.

The only factor that was variable was the number of candidate functions, which corresponds to the total of the number of motifs and motif interactions searched. When the pre-ranking function is turned on, only the top 1st order motifs and their cross-products were included as candidates. For the majority of runs, the top 10 motifs from the pre-ranking were used. While other cutoffs were tried, cross-validation showed that allowing more than the top 10 introduced higher levels of noise fitting, while less than 10 sometimes excluded motifs that are necessary for cooperative binding. Creating cross-product terms using the top 10 motifs would then add 55 and 220 distinct 2nd and 3rd order terms, respectively. Pre-screening motifs would involve running FOS on the entire set of individual motifs in a given list, meaning anywhere from 1024 (5-mers) to 16384 (7-mers) motifs would be searched.

Table 5.3 shows the measured running times for a selected set of runs. All runs were executed on a notebook with a 2.0 GHz Pentium Centrino processor with 2GB of RAM running Windows XP. Clearly, the length of the pre-screening process is highly dependent on the length of the candidate motif list. However, once only the most

significant candidates are found, the addition of a relatively small number of candidate pairs and groups of motifs does not greatly affect the overall running time. Another item to note is the linear relationship between the prescreening time and the number of motifs. Only fairly concise lists of motifs (≤ 16384) were searched (in ≤ 81 seconds) in the pre-screening, but more lengthy lists could be used without running time blowing up. Memory constraints, however, do become a factor as the number of candidate functions increase, and thus more powerful computers become desirable.

Table 5.3: Running times using various motif lists

Motif List	# of Motifs	Running Time (seconds)		
		Prescreening	Modeling	Total
5mers (1^{st} order)	1024	4.9	<0.1	5.0
5mers ($1^{\text{st}} + 2^{\text{nd}}$ order)	1024	4.9	0.2	5.2
5mers ($1^{\text{st}}, 2^{\text{nd}} + 3^{\text{rd}}$ order)	1024	4.9	2.6	7.6
6mers (1^{st} order)	4096	18.4	<0.1	18.5
6mers ($1^{\text{st}} + 2^{\text{nd}}$ order)	4096	18.4	0.2	18.7
6mers ($1^{\text{st}}, 2^{\text{nd}} + 3^{\text{rd}}$ order)	4096	18.4	2.2	20.6
7mers (1^{st} order)	16384	80.8	<0.1	80.9
7mers ($1^{\text{st}} + 2^{\text{nd}}$ order)	16384	80.8	0.3	81.2
7mers ($1^{\text{st}}, 2^{\text{nd}} + 3^{\text{rd}}$ order)	16384	80.8	3.5	84.4
10bp7p skeletons ($1^{\text{st}} + 2^{\text{nd}}$ order)	4041	16.7	<0.1	16.8
9bp7p skeletons (1^{st} order)	3706	14.8	<0.1	14.9

5.6 Cell-Cycle

Over the course of the cell-cycle, genes' transcription levels change drastically, allowing for physiological changes to take place within the cell. Various TF-binding

motifs found by FOS at the 14 minute time point were used to show their individual contributions to the reduction in variance across the 715 genes over the course of the cell-cycle. This was done by giving FOS only one candidate function (the word count profile for the motif under study). For each time point from the cell-cycle data, FOS was allowed to create a one-term model, or choose to not use the term. The RIV achieved by each of these models was then plotted at the corresponding time point to show how the motif's appearance correlates to expression throughout the timeline of the cell cycle.

The reduction in variance of the well-known MCB (*acgcgt*) element was shown to have a sinusoidal-like plot across the cell-cycle in [DAS04,07a]. FOS also shows a sinusoidal-like plot, with no contribution from MCB at 4 time points across the two cell-cycles, which was also found by MARSMotif [DAS04,07a] and REDUCE [BUS01]. FOS clearly is able to capture the peak activity of MCB that occurs at the 21 minute time point, exactly agreeing with MARSMotif [DAS04,07a]. The plot of the *acgcgt* motif's contribution to the RIV across the two cell-cycles is shown in Figure 5.4. At certain points where the MCB element is known to be inactive, FOS finds the MCB contribution to be negligible and it is not chosen to be included in the model.

The contributions of several other motifs found by FOS are shown over the entire 120 minutes of the alpha-arrest experiment. The individual contributions of motifs *aaacaa*, *ctaagc*, and *ncctcgncn* are shown in Figures 5.5, 5.6 and 5.7, respectively. The contribution of two pairs of 5mer motifs is shown in Figures 5.8 (*agggg/ctcga*) and 5.9 (*gtaat/gtaat*). It is clear that all motifs and groups of

motifs shown in these plots are active for different lengths of time in the cell-cycle. Because all motifs were found by modeling the expression data at the 14 minute time point, all plots show activity in that early portion of the data. Several motifs activate at multiple times, while others, such as the *agggg/ctcga* are only active during one portion of the cell-cycle.

The expression data cover two cell cycles, and so it would seem natural that the plots would contain very similar %RIV profiles over the first hour and second hour, corresponding to the two full cycles. There are several possible explanations for why the modeling results in variations between the two cycles. Because the data are very noisy, the algorithm can make different estimates of the value of a motif or motif group at different time points, resulting in different RIVs. It is clear that in Figure 5.4, the MCB element lowers the RIV by very different amounts in the first and second cycles. This difference is also seen in [DAS04,07a] and is largely attributable to the noise in the data, as it is a well-known functional motif.

The other cell cycle plots show quite visible differences between the two cycles. The very different behaviors in the plots of Figures 5.6, 5.7, 5.8 and 5.9 might be attributable to overwhelming noise in the second cycle, but might also be evidence of false positive motif findings. The latter might be the case, for example, with the pair *agggg/ctcga*, which in Figure 5.8 shows activity in only cycle one. However, it is hard to discount any of these motifs entirely. The variation between the RIV in cycles one and two due to *ctaagc* (Figure 5.6) can be seen, essentially, as the removal of a peak at

around 110 minutes. However, the second peak in the first cycle only corresponds to a 2% reduction in variance. In comparison, the first peak due to the `acgcgt` motif (Figure 5.4) displays approximately a 10 percentage point drop in RIV from cycle one to two – a drop to half of the value found in the first cycle – and its functionality cannot be discounted.

Figure 5.4: Percent Reduction in Variance of the `acgcgt` (MCB) element over two cell-cycles

Figure 5.5: Percent Reduction in Variance of the aaacaa (STE12) element over two cell-cycles

Figure 5.6: Percent Reduction in Variance of the ctaagc element over two cell-cycles

Figure 5.7: Percent Reduction in Variance of the *ncctcgnon* element over two cell-cycles

Figure 5.8: Percent Reduction in Variance of the *agggg/ctcga* pair over two cell-cycles

Figure 5.9: Percent Reduction in Variance of the *gtaat/gtaat* pair over two cell-cycles

It should be noted that while all %RIVs are positive, the source of the reduction in variance comes from both activating and repressive behaviors by motifs. The expansion coefficients for the *acgcgt* motif found across the cell cycle are plotted in Figure 5.10. Here, it is shown that the motif likely binds to a TF with a strongly activating nature (positive a_m term – as defined in equation (3.1.1)), reaching peak activity at or near the 21 minute time point before changing modes to correlate with a repressing nature (negative a_m term). Another possibility is that, in genes with many appearances of the *acgcgt* motif, expression is largely dependent on high levels of a particular TF, and thus when that TF dips to a low level, these genes' expression is much more diminished compared with genes not dependent on this motif binding to the TF.

Figure 5.10: Expansion coefficient a_m , as defined in equation (3.1.1), of the `acgcgt` motif over two cell-cycles. At certain times, the expansion coefficient is found to be negative, corresponding to a possible repressive tendency of the motif `acgcgt` towards transcription. Simply put, the genes containing this motif are down-regulated around 55 minutes and again around 120 minutes. At times where the coefficient becomes positive, the transcription process is activated by the presence of the motif.

5.7 Noise Levels and the Validity of the Results

Noise levels are very significant in the process of modeling gene expression purely by word counts of potential regulatory elements. Expression is enhanced and repressed by the binding of transcription factors to the regulatory motifs found in the DNA upstream of coding regions; however, several factors other than motif appearances have an effect on the changes in transcription level. Therefore, it is impossible to account for all the variations in the expression data solely by the counts of motif appearances.

Das et. al. [DAS04,07a] estimate noise to account for approximately 50% of the variations in the data. Some noise is introduced by inaccuracies in the acquiring of the microarray data. A more important source of noise comes from the fact that concentrations in the TFs have great effects on the expression levels. Although a gene may have plenty of binding sites, should the corresponding TF concentration in the cell be low, binding will not occur and transcription will occur at a lower rate than would be predicted by the number of motif appearances. Additionally, higher order TF-TF interactions are required to activate TFs. While a particular TF might be present, unless it has been conformed properly, it might not bind to the target motif [YU06]. Therefore, depending on the physiological conditions, the activity of a TF and thus the expression levels of genes will be variable. While such variability accounts for noise, this method and alternative methods have still been able to uncover motifs that are probable or known binding sites.

Chapter 6: Conclusions

6.1 Summary

In recent years, the increase in the ability to gather genetic data has led to a vast library of knowledge. With many fully sequenced genomes and the availability of microarray data that can quite accurately measure transcription levels, a number of methods have been developed to analyze these data. In this thesis, FOS has been found to be an effective method of uncovering potential TF-binding motifs. It has been demonstrated that FOS compares favorably with previous methods and easily finds known functional motifs to which transcription factors can bind. Because of the efficiency of FOS, the discussed method can execute in very little time and build concise models with high degree of accuracy. The MARS-based methods [DAS04,07a] were able to predict several pairs of motifs (and even three-way interactions) that are likely to be functional in cooperatively binding to transcription factors. Because of Fast Orthogonal Search's efficiency, third order cross-products are easily searched and several groups that show high potential for functional cooperative behavior are found.

One strength of this method is that it is able to discover motifs and synergistic pairs and groups of motifs without introducing a great number of parameters, and avoided the addition of basis functions like splines [DAS04,07a]. While more complex models can approximate a system's output (gene expression) slightly more accurately over the training data, there can sometimes be a tendency for the added accuracy to be a side effect of noise fitting. All models were built by FOS with minimal parameters to reduce this problem. Cross-validation was used to eliminate motifs that appear infrequently, indicating that those motifs are more likely to be fitting noise rather than explaining real regulatory effects on expression levels.

The implicit orthogonalization carried out by FOS is a key characteristic to its success in a number of applications. In terms of modeling gene expression profiles with motif counts, the implicit orthogonalization ensures that once a term is added to the model, the next term added will be the one which best explains the output while taking into account the previous terms added. Other methods not using this type of orthogonalization will find terms that explain the data, but might not conveniently find the terms that explain the remaining portions of the data after adding terms. The fact that FOS does this makes it a powerful tool in quickly building accurate and concise models.

The fact that no a priori information is needed means that this method can be employed on a complete set of motifs, working from scratch without requiring a separate method to narrow down the possible candidates. Because of the efficiency of

the method, large numbers of candidates can be searched quickly. In view of the corroboration of some FOS-found motifs by those previously reported, it is likely that motifs frequently returned by FOS have a high probability of being active binding sites for transcription factors.

6.2 Future Work

As previously mentioned, there are a number of important applications for this type of method. The need to understand the expression regulators in genes is as important as understanding the functions of genes themselves. A wide variation in phenotypes appears from organisms with a large portion of similar genetic sequences, and the non-coding portions of genes hold the secret to the vast differences among species. Certain diseases arise from mutations in the non-coding portions of genes, which lead to the creation or removal of transcription factor binding sites, and being able to use a method that can relate a disease to a specific motif or group of motifs would be of great importance in identifying the source of the disease. Many studies have applied computational methods to analyzing a variety of sets of data in order to uncover information pertaining to transcriptional networks involved in disease proliferation, and FOS would likely be able to return important characteristics if applied to these other gene expression data sets.

The MARS-based method has already been applied to the human genome and its designers have noticed some differences in transcription factors between humans and yeast [DAS06]. The fact that human transcription factors are more degenerate and possibly longer suggests that more motifs will have to be included as candidates. The computational requirements for human analysis will be much higher, and the efficiency of FOS in comparison to other methods will make it a valuable alternative, especially as the motif lists expand.

There are reasons to believe that the implementation of FOS used here could yield important findings if extended to humans, but the results can still be improved by making some variations. Firstly, one option is to not force FOS to include a constant term [ADE00]. While the constant aids to remove a bias from the expression data, there is a chance that a term with a near-constant word count profile might not be chosen due to this term. Secondly, variations in the way cross-validation was used might also help to reduce the number of false positive motifs. Thirdly, by trying a broader selection of position weight matrices, the preferred locations of TF-binding motifs might become elucidated. Finally, a key difference to consider, in making the extension to analysis of higher order species, seems to be the length and degeneracy of the motifs. Lists of all possible degenerate motifs 9 and 10 nucleotides long will have to be screened by FOS, whereas only a small portion of all possible degenerate motifs were considered here.

The computational requirements of considering millions of candidate motifs will become difficult for a personal computer to handle. While running time only grows

linearly with $M \times C \times G$, where M is the number of terms added to the model, C is the number of candidate functions and G is the number of genes used, memory constraints become a problem when the candidate arrays get large. The candidate array includes C candidate motif word count functions, each G points (genes) long (i.e. size is $C \times G$ points). It was observed that FOS could find motifs using less than the 600-700 genes generally used. Cutting the number of genes under analysis could allow for more motifs to be searched at a time, for the same candidate array size. Only running FOS on a subset of candidate motifs at a time could eliminate the memory constraints. Each subset could be pre-screened and then a model could be created by once again running FOS on the top few from each subset (perhaps 10 from each).

Bibliography

- [ADE00] Adeney K. M. & Korenberg M. J. "Iterative fast orthogonal search for MDL-based training of generalized single-layer networks". *Neural Networks*, Vol. 13, pp. 787-799, 2000.
- [BAN03] Banerjee N. & Zhang M. Q. "Identifying cooperativity among transcription factors controlling the cell cycle in yeast". *Nucleic Acids Research*, Vol. 31, pp. 7024-7031, 2003.
- [BEE04] Beer M. A. & Tavazoie S. "Predicting gene expression from sequence". *Cell*, Vol. 117, pp. 185–198, 2004.
- [BUS01] Bussemaker H. J., Li H. & Siggia E. D. "Regulatory element detection using correlation with expression". *Nature Genetics*, Vol. 27, pp. 167-171, 2001.
- [CHI03] Chiang D. Y., Moses A. M., Kellis M., Lander E. S. & Eisen M. B. "Position specific variation in the rate of evolution in transcription factor binding sites". *Genome Biology*, Vol. 4, R43, 2003.
- [CHO98] Cho R. J., Campbell M. J., Winzeler E. A., Steinmetz L., Conway A., Wodicka L., Wolfsberg T. G., Gabrielian A. E., Landsman D., Lockhart D. J. & Davis R. W. "A genome-wide transcriptional analysis of the mitotic cell cycle". *Molecular Cell*, Vol. 2, pp. 65-73, 1998.
- [CON03] Conlon E. M., Liu X. S., Lieb J. D. & Liu J. S. "Integrating regulatory motif discovery and genome-wide expression analysis". *Proceedings of the National Academy of Sciences*, Vol. 100, pp. 3339-3344, 2003.
- [DAS04] Das D., Banerjee N. & Zhang M. Q. "Interacting Models of Cooperative Gene Regulation". *Proceedings of the National Academy of Sciences*, Vol. 101, pp. 16234-16239, 2004.

- [DAS06] Das D., Nahle Z. & Zhang M. Q. "Adaptively inferring human transcriptional subnetworks". *Molecular Systems Biology*, Vol. 2, 2006.0029, 2006.
- [DAS07a] Das D. & Zhang M. Q. "Predictive Models of Gene Regulation: Application of Regression Methods to Microarray Data", in "Microarray Data Analysis: Methods and Applications". Korenberg, M. J., ed., Totowa, N. J.: Humana Press, pp. 95-110, 2007.
- [DAS07b] Das D., Clark T. A., Schweitzer A., Yamamoto M., Marr H., Arribere J., Minovitsky S., Poliakov A., Dubchak I., Blume J. E. & Conboy J. G. "A correlation with exon expression approach to identify cis-regulatory elements for tissue-specific alternative splicing". *Nucleic Acids Research*, Vol. 35, pp. 4845-4857, 2007.
- [DES81] Desrochers A. A. "On an improved model reduction technique for nonlinear systems". *Automatica*, Vol. 17, pp. 407-409, 1981.
- [EIS98] Eisen M. B., Spellman P. T., Brown P. O. & Botstein D. "Cluster analysis and display of genome-wide expression patterns". *Proceedings of the National Academy of Sciences*, Vol. 95, pp. 14863–14868, 1998.
- [ELE05] Elemento O. & Tavazoie S. "Fast and systematic genome-wide discovery of conserved regulatory elements using a non-alignment based approach". *Genome Biology* 6:R18.1-27, 2005.
- [FRI91] Friedman J. H. "Multivariate adaptive regression splines". *Annals of Statistics*, Vol. 19, pp. 1-67, 1991.
- [KAT04] Kato M., Hata N., Banerjee N., Futcher B. & Zhang M. Q. "Identifying combinatorial regulation of transcription factors and binding motifs". *Genome Biology*, Vol. 5, pp. R56.1-13, 2004.
- [KEL02] Keles S., van der Laan M. & Eisen M. B. "Identification of regulatory elements using a feature selection method". *Bioinformatics*, Vol. 18, pp. 1167-1175, 2002.
- [KIM98] Kim S. H., Lee H. J., Huh J. M. & Chang B. C. "Spectral analysis of heart valve sound for detection of prosthetic heart valve diseases". *Yonsei Medical Journal*, Vol. 39, Issue 4, pp. 302-308, 1998.
- [KOR87] Korenberg M. J. "Fast orthogonal identification of nonlinear difference equation and function expansion models". *Proceedings of the 28th Midwest Symposium on Circuits and Systems*, pp. 270-276, 1987.

- [KOR88] Korenberg M. J., Bruder S. B. & McIlroy P. J. "Exact orthogonal kernel estimation from finite data records: extending Wiener's identification of nonlinear systems". *Annals of Biomedical Engineering*, Vol. 16, no. 2, pp. 201-2104, 1988.
- [KOR89] Korenberg M. J. "A robust orthogonal algorithm for system identification and time-series analysis". *Biological Cybernetics*, Vol. 60, pp. 267-276, 1989.
- [KOR91] Korenberg M. J. & Paarmann L.D. "Orthogonal approaches to time-series analysis and system identification". *IEEE SP Magazine* July, pp. 29-43, 1991.
- [KOR97] Korenberg M. J., Brenan C. J. & Hunter I. W. "Raman spectral estimation via fast orthogonal search". *The Analyst (Royal Society of Chemistry, Cambridge, U.K.)*, Vol. 122, pp. 879-882, 1997.
- [LAT97] Latchman D. S. "Transcription Factors: An Overview". *International Journal of Biochemistry Cell Biology*, Vol. 29, pp. 1305-1312, 1997.
- [MCG06] McGaughey D. R., Tarbouchi M., Nutt K. & Chikhani A. "Speed sensorless estimation of AC induction motors using the fast orthogonal search algorithm". *IEEE Transactions on energy conversion*, Vol. 21, Issue 1, pp. 112-120, 2006.
- [MIN08] Minz I. & Korenberg M. J. "Modeling Cooperative Gene Regulation Using Fast Orthogonal Search". Unpublished, 2008.
- [NGU06] Nguyen D. H. & D'haeseleer P. "Deciphering principles of transcription regulation in eukaryotic genomes". *Molecular Systems Biology*, Vol. 2, 2006.0012, 2006.
- [PHU04] Phuong T. M., Lee D. & Lee K. H. "Regression trees for regulatory element identification". *Bioinformatics*, Vol. 20, pp. 750-757, 2004.
- [PIE04] Pierce B. A. "Control of gene expression", in "Genetics: A Conceptual Approach". W H Freeman & Co, pp. 434-465, 2004.
- [PIL01] Pilpel Y., Sudarsanam P. & Church G. M. "Identifying regulatory networks by combinatorial analysis of promoter elements". *Nature Genetics*, Vol. 29, pp. 153-159, 2001.
- [PRI08] Pritzker M., Liu Y., Beer M. A. & Tavazoie S. "Whole-genome discovery of transcription factor binding sites by network-level conservation". *Cold Spring Harbor Laboratory Press*, Vol. 14, pp. 99-108, 2008.

- [ROM05] Romanenkov A. S., Ustyugov A. A., Zatsepin T. S., Nikulova A. A., Kolesnikov I. V., Metelev V. G., Oretskaya T. S. & Kubareva E. A. "Analysis of DNA-protein interactions in complexes of transcription factor NF- κ B with DNA". *Biochemistry (Moscow)*, Vol. 70, pp. 1212-1222, 2005.
- [SCH95] Schena M., Shalon D., Davis R. W. & Brown P. O. "Quantitative monitoring of gene expression patterns with a complementary DNA microarray". *Science*, Vol. 270, pp. 467-470, 1995.
- [SHA97] Shannahoff-Khalsa D. S., Kennedy B., Yates F. E. & Ziegler M. G. "Low-frequency ultradian insulin rhythms are coupled to cardiovascular, autonomic, and neuroendocrine rhythms". *American Journal of Physiology – Regulatory Integrative and Comparative Physiology*, Vol. 272, Issue 3, pp. R962-R968, 1997.
- [SPE98] Spellman P. T., Sherlock G., Zhang M. Q., Iyer V. R., Anders K., Eisen M. B., Brown P. O., Botstein D. & Futcher B. "Comprehensive identification of cell cycle-regulated genes of the yeast *Saccharomyces cerevisiae* by microarray hybridization". *Molecular Biology Cell*, Vol. 9, pp. 3273-3297, 1998.
- [TSA05] Tsai H., Lu H. H. & Li W. "Statistical methods for identifying yeast cell cycle transcription factors". *Proceedings of the National Academy of Sciences*, Vol. 102, pp. 13532-13537, 2005.
- [YU06] Yu X., Lin J., Masuda T., Esumi N., Zack D. J. & Qian J. "Genome-wide prediction and characterization of interactions between transcription factors in *Saccharomyces cerevisiae*". *Nucleic Acids Research*, Vol. 34, pp. 917-927, 2006.
- [YUA07] Yuan Y., Guo L., Shen L. & Liu J. S. "Predicting gene expression from sequence: a reexamination". *PLoS Computational Biology*, Vol. 3, pp. 2391-2397, 2007.