

A Biclustering Approach to Combinatorial Transcription Control

Venkataraghavan Srinivasan

Thesis submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Master of Science

in

Computer Science

T. M. Murali, Chair

Brett M. Tyler

Liqing Zhang

Reinhard Laubenbacher

July 06, 2005

Blacksburg, Virginia

Keywords: Combinatorial transcription control, Promoter analysis, Biclustering,

Random sampling

Copyright 2005, Venkataraghavan Srinivasan

A Biclustering Approach to Combinatorial Transcription Control

Venkataraghavan Srinivasan

(ABSTRACT)

Combinatorial control of transcription is a well established phenomenon in the cell. Multiple transcription factors often bind to the same transcriptional control region of a gene and interact with each other to control the expression of the gene. It is thus necessary to consider the joint conservation of sequence pairs in order to identify combinations of binding sites to which the transcription factors bind. Conventional motif finding algorithms fail to address this issue. We propose a novel biclustering algorithm based on random sampling to identify candidate binding site combinations. We establish bounds on the various parameters to the algorithm and study the conditions under which the algorithm is guaranteed to identify candidate binding sites. We analyzed a yeast cell cycle gene expression data set using our algorithm and recovered certain novel combinations of binding sites, besides those already reported in the literature.

Acknowledgments

I would sincerely like to thank my advisors T. M. Murali and Brett Tyler who have been the most patient and helpful guides one can ask for. Working on this thesis has been a great learning experience for me and I am grateful to them for giving me the opportunity.

I would also like to express my sincere gratitude to Dr.Alok Bhattacharya of Jawaharlal Nehru University, New Delhi, India for mentoring my undergraduate thesis, from which I learnt a lot. Some of the concepts that I learnt while working on my undergraduate thesis have also found a place in this thesis.

I would also like to thank my fellow graduate students Jonathan Myers and Greg Grothaus for letting me modify and use their hypergeometric library.

Contents

1	Introduction	1
1.1	Regulatory Elements and Transcription Control	1
1.2	Combinatorial Transcription Control	2
1.2.1	Co-operative Binding	3
1.2.2	Synergistic Activation	3
1.3	Computational Approaches to Combinatorial Control	4
1.3.1	Synergistic Motif Combinations	4
1.3.2	Combining ChIP and Microarrays	5
1.3.3	Combining Phylogeny and Microarrays	6
1.4	Biclustering	7
1.4.1	Greedy Approach	9

1.4.2	Iterative Approach	10
1.4.3	Divide and Conquer Approach	10
1.4.4	Graph Theoretic Approach	11
1.4.5	Random Sampling Approach	12
1.5	Contributions of this Thesis	12
2	Statistical Background	14
2.1	Sampling	14
2.2	Hypothesis Testing	15
2.3	Hypergeometric Distribution	16
3	Algorithm	18
3.1	Underlying Principles	18
3.1.1	Over-representation and Motif Finding	19
3.1.2	Co-occurrence and Positional Proximity of Binding sites	19
3.2	Algorithm	20
3.2.1	Sampling	21
3.2.2	Significance Calculation and Correction for Multiple Testing	22

3.2.3	Extension of Motif Pairs	24
4	Simulations	26
4.1	Exploring the Parameter Space	29
4.1.1	Sample Size	30
4.1.2	Number of Samples Required till Success	31
4.1.3	Foreground and Background Frequencies	33
5	Biological Data Analysis	34
5.1	Algorithm Parameters	35
5.2	Significant Combinations	36
5.2.1	Potential Interactions between HSF1 and MBF	37
5.2.2	Co-occurrence of Potential Sites for Mbp1 and SWI4	37
6	Conclusions and Future Work	38

List of Figures

1.1	Clustering and biclustering of gene expression matrix	8
3.1	Motif pair extension	25
4.1	Variation in number of genes n_r and number of samples required s_r with background frequency and r	31
4.2	Effect of foreground and background frequencies on s_r	32
5.1	Biclusters	36

List of Tables

4.1	Effect of sample size r on s_r at foreground frequency = 0.7 and $\alpha = 0.05$.	31
4.2	Effects of background and foreground frequencies on s_r at $r = 20$ and $\alpha = 0.05$	33

Chapter 1

Introduction

In this chapter we introduce the notion of combinatorial transcription control. We also introduce a computational technique called biclustering, that has been used successfully to simultaneously cluster genes and conditions from DNA microarray expression data.

1.1 Regulatory Elements and Transcription Control

A transcription factor is a protein that regulates the expression of a gene. This regulation is achieved via the binding of the transcription factor to a set of nucleotides, called the *binding sites*, in the promoter region of the gene that the transcription factor regulates. Obtaining knowledge of these target sites and the mechanisms of

binding are thus fundamental tasks in understanding the transcription of genes.

Different computational approaches are in common use to detect these target sites. Most approaches analyze the promoters of co-regulated genes for presence of conserved binding sites. These co-regulated genes could either be those obtained from the microarray analysis or could be genes with similar cellular functions. These approaches however are difficult when the binding sites are highly divergent; which is the case more often than not.

1.2 Combinatorial Transcription Control

Comparative genome analysis has revealed a constancy in genetic content even among organisms in different phylogenetic domains. A natural question that arises now is how complex eukaryotes have achieved such diversity in phenotypes in spite of having more or less the same genetic content as other lower eukaryotes. A commonly accepted view is that this diversity is achieved by co-operativity and interaction between transcription factors.^{LT03} In order to understand this idea, it is first necessary to look at the mechanisms by which a transcription factor can bind to its binding site.

1.2.1 Co-operative Binding

Binding of a transcription factor to its target can be rendered favorable by the presence of another transcription factor due to protein-protein interactions between the two transcription factors. This phenomenon is known as co-operative binding. It has also been found that a transcription factor complex does not necessarily have to bind directly by itself to its target site. For example, the transcription factor Ndd1 does not directly bind to its target (GTAAACA) directly but it binds through transcription factor Fkh1 or Fkh2.^{KSEA00}

1.2.2 Synergistic Activation

In higher eukaryotes, synergism among transcription factors in activating transcription can contribute to the regulation of gene expression. In response to internal or external signals, different transcriptional activators can contribute in different synergistic ways to stabilize the assembly of the transcription initiation complex. Thus multiple transcription factors can interact in two distinct ways to modulate transcription in a non-linear manner; co-operative binding and synergistic activation.

1.3 Computational Approaches to Combinatorial Control

Recently, some computational algorithms have been developed to address combinatorial control. We discuss some of these in this section.

1.3.1 Synergistic Motif Combinations

Pilpel et. al. were among the first to develop a computational algorithm to study combinatorial control.^{PSC01} They established a database of known and putative regulatory motifs and used ScanACE^{RHEC98} to identify genes that contain each motif in their promoters. For each motif combination present in the genes, they calculated an expression coherence score, which was the measure of overall similarity of expression profiles of all genes containing the motif or combination. They defined synergistic motif pairs to be those pairs such that, the genes containing either of the motifs alone had lower expression coherence score than the genes containing both the motifs in their promoter region. Pilpel et. al analyzed certain stages of the yeast cell cycle and were able to generate motif synergy maps which represent interactions between transcription factors of the motifs. An interesting outcome of this work was establishing the fact that certain transcription factors may be acting as global facilitator proteins. This approach however does not take into account the spacing between the motifs of

synergistic combinations.

1.3.2 Combining ChIP and Microarrays

Zhang et. al proposed an approach that integrates chromatin immunoprecipitation (ChIP) data with microarray expression data and combinatorial transcription factor-motif analysis.^{KHB⁺04} ChIP data provides strong *in vivo* evidence of binding of a transcription factor to a motif. Zhang et. al. first identify over-represented binding sites for each transcription factor. For all combinations of these binding sites, they then identify genes that contain the combination and also have the most coherent expression. They assign over-represented transcription factor combinations to the respective over-represented motif combinations. The final output is combinations of transcription factor and motifs corresponding to a functional gene set. They were able to reconstruct a new transcriptional regulation model for the yeast cell cycle by identifying combinations of transcription factors and motifs that are specific to each of the cell cycle phases. An interesting outcome of this work was identification of some fundamental characteristics of combinatorial control, at least in the context of yeast cell cycle, viz.

- waiting-activating systems
- join-phase combinations
- joint-process combinations

A waiting-activating system waits for some signal in a repressed state and then activates transcription. Several transcription factors bind to their targets in a repressed state and cause transcription when the signal occurs. In join-phase combinations, some genes are bound by a transcription factor that work in a particular stage of cell cycle and then by another transcription factor that binds in a later stage of cell cycle; the two regulators may either work independently or cooperatively. In joint-process combinations, different combinations of transcription factors allow genes to respond to different signals.

1.3.3 Combining Phylogeny and Microarrays

Eisen et. al. first identify conserved sequence motifs between four different species of *Saccharomyces* based on CLUSTALW^{THG94} alignments.^{CMK+03} They then tested for motif combinations that are found more frequently than at random. They also incorporated spatial relationships between motifs by retaining only those motif combinations that displayed closer physical spacing than expected by chance. They then identified gene subsets containing these motif combinations that also have similar expression profiles.

1.4 Biclustering

DNA microarrays are a powerful tool to study the expression levels of thousands of genes simultaneously. Genes that have similar expression profiles are more likely to be co-regulated as well. Expression levels of genes obtained from the microarray experiment are arranged in the form of a matrix, where the rows correspond to the genes and the columns correspond to experimental conditions (samples). A large number of clustering algorithms exist to identify such co-expressed genes^{ESBB98, TSM+99}. These algorithms either group the genes into categories based on the expressions under multiple conditions or group the conditions into categories based on the expression of the genes (see Figure 1.3^{TSS04}).

However, clustering algorithms have certain inherent drawbacks. A group of genes could be co-expressed only in certain conditions but may behave independently otherwise. This phenomenon is understandable because the transcription factor responsible for the expression of the gene is active only under certain conditions. Clustering algorithms however assign a gene to a cluster which spans all the columns (conditions; see figure 1.3). Clustering algorithms also partition genes into mutually exclusive clusters which means that a particular gene can only belong to one cluster. Such clusters can not capture the possibility that a gene can participate in multiple cellular pathways.

Biclustering overcomes the disadvantages associated with clustering algorithms. A

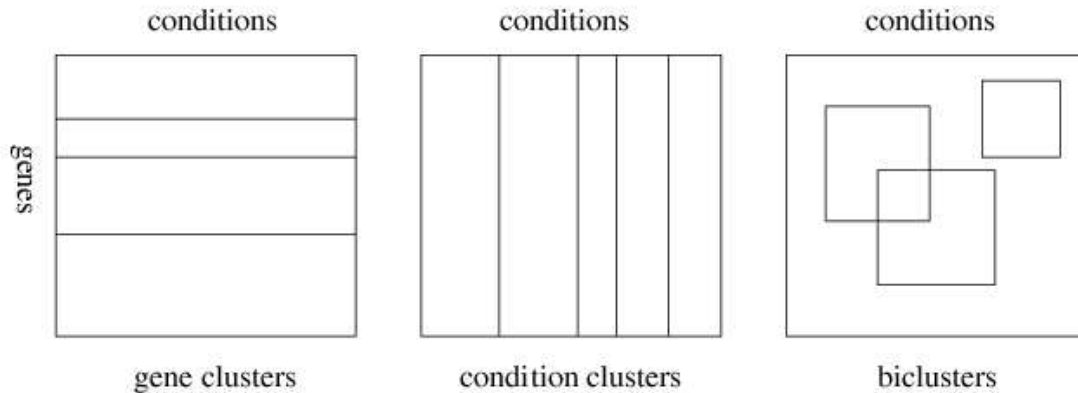


Figure 1.1: Clustering and biclustering of gene expression matrix. A cluster is represented by horizontal and vertical strips. Thus a gene cluster must contain all conditions and a condition cluster must contain all the genes. A bicluster is represented by rectangular box. A bicluster thus can contain a subset of genes and a subset of samples. Also, note the overlapping rectangles. This means that a gene may belong to multiple biclusters.

bicluster is a subset of genes and a subset of samples (see Figure 1.1). Thus a bicluster is a local model whereas a cluster is a global model i.e. whereas each gene in a cluster is defined using *all* the conditions, each gene in a bicluster is defined using only a *subset* of conditions (see Figure 1.3^{TSS04}). It overcomes the problems associated with conventional clustering algorithms by finding similarity based on a subset of attributes and by allowing overlapped grouping. Not only does it discover the grouping (for example, a set of co-expressed genes), but the context (for example, the set of conditions under which the genes are co-expressed) as well. It is thus more robust to noise and missing data.

Computing biclusters in a gene expression matrix is computationally a very hard problem. A brute force approach would be $O(2^m 2^n)$ in time complexity, where m

and n respectively are the number of rows and columns in the gene expression matrix. Such a brute force approach however is not feasible with microarray data as they typically contain thousands of rows. Different algorithms have been developed to find biclusters in gene expression data.^{CC00,MK03,TSS02} The biclustering algorithms essentially differ in what they define as a bicluster and the way they evaluate the significance of a bicluster. In this section we discuss different types biclustering algorithms based on the approach they use. Note that some of the algorithms may belong to multiple categories and hence the division is not strict. Madeira et. al. also present an exhaustive survey^{MO04} based along these lines.

1.4.1 Greedy Approach

Cheng and Church were the first to introduce the idea of biclustering.^{CC00} They used it to find biclusters in yeast and human microarray data. They define a bicluster to be a submatrix which has more or less constant expression levels for each row. A bicluster is deemed significant if the score (average mean squared residue) of the bicluster is below a threshold. The algorithm starts with the entire gene expression matrix and removes rows and columns in a greedy fashion such that the variance of the resulting submatrix is lower than the previous one. The algorithm terminates when it is no longer possible to remove any more rows and columns without raising the variance of the resulting submatrix. If the score of the submatrix is below the threshold, it is then reported as a significant bicluster. To find multiple biclusters, the expression values

corresponding to this submatrix are randomized and the procedure is repeated. This continues iteratively till a user defined number of biclusters is found.

1.4.2 Iterative Approach

Getz et. al. proposed an iterative approach called Coupled Two-Way Clustering, that transforms a hierarchical clustering algorithm into a biclustering algorithm.^{GLD00} They start with the gene expression matrix and apply the clustering algorithm to the matrix twice, first along the rows to find the gene clusters and then along the column to find the condition clusters. The algorithm only takes stable gene and condition clusters into further consideration. For every combination of stable gene and condition cluster, the algorithm then iteratively keeps applying the clustering algorithm (again twice, once each along row and column) till it detects stable clusters. The success of this algorithm is thus largely contingent upon the underlying clustering algorithm chosen. Getz et. al. used a hierarchical clustering algorithm called super-paramagnetic clustering^{BWD96} and analyzed a variety of clinical data sets.^{GLDZ00}

1.4.3 Divide and Conquer Approach

Hartigan proposed a divide and conquer approach called direct clustering.^{Har72} Hartigan begins with the entire gene expression matrix in one block and in each step, divides the block into two blocks such that the reduction in within-block variance is

the largest. This splitting continues recursively till either a defined number of biclusters is found or till the within block variance reaches a certain threshold. Though this algorithm is very fast, it is likely to miss good biclusters since they may be split before they can even be identified.^{MO04}

1.4.4 Graph Theoretic Approach

Shamir et. al. propose a biclustering algorithm called SAMBA.^{TSKS04} Gene expression data is modeled as a bipartite graph. A bipartite graph is a graph whose nodes can be partitioned into two disjoint sets such that no two nodes within the same set are adjacent. In this case, the graph is partitioned into two sets, one which consists of genes and the other consisting of conditions. The edges between the genes and conditions are assigned scores based on a probabilistic model such that subgraphs with largest weight correspond to significant biclusters. The problem thus is reduced to finding dense subgraphs in a bipartite graph. This problem is known to be *NP* complete. Shamir et. al. proposed a heuristic to find an approximate polynomial time solution. They used the algorithm to assign genes with unknown functions into functional categories.

1.4.5 Random Sampling Approach

Murali and Kasif proposed an algorithm to find biclusters in gene expression data. Their output was what they called an *xMotif* or a conserved gene expression motif. An *xMotif* is a bicluster such that the expression of each gene in the *xMotif* is simultaneously conserved across every sample in the *xMotif*. They define gene states to be a range of expression values and that there are a fixed number of such gene states. Murali and Kasif aimed to find the largest *xMotif* in the data set i.e. the one that had the greatest number of genes. They repeatedly select random subsets of samples, and for each random subset, they find more genes that are in the same state in the random subset and add such genes to the subset. They next add any samples that “fit” the *xMotif*. Murali and Kasif applied this algorithm to clinical data sets.^{GST+99}

1.5 Contributions of this Thesis

Combinatorial control of transcription is an important phenomenon in eukaryotic genomes.^{Car98,LT03,CMK+03,KHB+04,PSC01} The strong possibility that multiple transcription factors bind to targets which are in proximity to each other may be exploited to identify jointly conserved binding sequences. Motif finding algorithms seek to identify binding sites independently of each other completely overlooking this phenomenon. We propose a novel algorithm to detect combinations of transcription factor binding sites that seeks to exploit co-conservation of binding sites. Our algorithm uses

an $xMotif^{MK03}$ -like random sampling based biclustering approach to efficiently prune the search space for candidate binding site combinations. Our algorithm needs no *a priori* knowledge about the binding sites and does not require any additional sources of data. Our algorithm also combines the the strengths of the other computational approaches^{KHB⁺04, PSC01, CMK⁺03} in that it addresses simultaneous conservation, efficient pruning of search space and identifying unequal sized binding sites combinations.

Chapter 2

Statistical Background

Our algorithm uses ideas from inferential statistics. The aim of this chapter provide the necessary statistical background required to understand the rest of this thesis. In this chapter, we discuss the notions of random sampling, hypothesis testing and multiple hypothesis correction.

2.1 Sampling

A sample is a unit of data representative of the entire population under consideration. When the population is large, it is not feasible to do a census i.e. it is not possible to visit every member of the population. It is thus desired to pick a small subset of the population in order to model the entire population. In other words, a sample should

reflect the typical behavior of the entire population.

Sampling falls broadly into two categories: sampling with replacement and sampling without replacement.

Sampling with replacement is the method in which, as soon as an element of the population is selected to be included in the sample, it is returned back to the population.

Sampling without replacement is the method in which approach an element which is selected for inclusion in the sample is not returned back to the population pool.

2.2 Hypothesis Testing

A hypothesis is a claim that is put forward, with or without a proof, to be used in an argument. An example of a hypothesis is the claim that a new drug is effective in treating a disease, the argument here being the effectiveness of the new drug in treating the disease.

A null hypothesis is the negative of the claim that is being put forward. Our aim is to prove or disprove the null hypothesis. Results of hypothesis testing are expressed in terms of the null hypothesis i.e. accepting or rejecting the null hypothesis in favor of an alternate hypothesis, because it is usually easier to model the null hypothesis. In other words the aim of hypothesis testing is to find sufficient evidence for or against the null hypothesis.

In order to perform hypothesis testing, we need to calculate a test statistic from our sample data. The method for calculating the test statistic depends on the probability model chosen for the population. We compare the test statistic to a chosen threshold in order to decide whether to accept the null hypothesis or not.

2.3 Hypergeometric Distribution

The hypergeometric distribution used in several hypothesis testing procedures in our algorithm. Consider a population whose elements can be partitioned into two mutually exclusive classes. We then select a sample uniformly at random and without replacement from the population. This sample will contain elements either exclusively from either of the two classes or elements from both the classes. We wish to calculate the significance of the occurrence this configuration of elements in our sample. We formulate the null and the alternate hypothesis accordingly.

It is easier to visualize hypergeometric distribution in the context of the so-called “urn problem”.^{Wil68} Let us say that we have an urn that contains n balls, of which p are red in color and $n - p$ are blue in color. We draw a sample of x balls uniformly at random and without replacement from this population. Now let us say that we need to compute the probability that our sample has at least r red balls. We wish to evaluate the significance of this event i.e. the probability this event occurred by random chance. Thus our null hypothesis H_0 is that our sample contains at least r

red balls and that this event occurred at random, and the alternate hypothesis H_1 is that this event is not random.

We calculate the probability of occurrence of this event is as follows: The total number of ways of selecting i red balls in a sample, out of p such balls is $\binom{p}{i}$. The total number of ways of selecting $x - i$ blue balls in a sample, out of $n - p$ balls = $\binom{n-p}{x-i}$. Thus the probability of occurrence of this event is given by:

$$P(\text{number of red balls in the sample} \geq r) = \sum_{i=r}^{\min(p,x)} \frac{\binom{p}{i} \binom{n-p}{x-i}}{\binom{n}{x}}$$

We would accept H_0 to be true if and only if the value of the hypergeometric test statistic calculated above is greater than a specified threshold.

Chapter 3

Algorithm

In this chapter, we first briefly describe the underlying principles of our algorithm. Although most of these concepts have been dealt with exhaustively in Chapter 1 of this thesis, we nevertheless summarize the central ideas here for the sake of continuity. We then go on to describe the algorithm in detail.

3.1 Underlying Principles

In this section we discuss concepts which are central to our algorithm.

3.1.1 Over-representation and Motif Finding

A set of genes is said to be *co-regulated* when the genes in the set are bound to by a common transcription factor which potentially controls the expression of the genes simultaneously. A transcription factor binds to a location in the promoter region of a gene, known as a *binding site* or a regulatory element. Thus if a set of genes is co-regulated, it is expected that they share a common binding site, to which the transcription factor would bind. In other words, given a set of genes that are co-regulated, we would expect the binding sites to occur more frequently in the set of genes than expected at random. This phenomenon is known as *over-representation* and has been successfully applied in identifying regulatory elements.^{vHACV98,ST02,LMS96} This observation implies that given a set of genes, some of which are co-regulated, in principle it is possible to separate the co-regulated genes from the rest by simply looking for the presence for a common (and over-represented) shared sequence. The goal of our algorithm therefore is to detect over-represented motifs in a given set of genes.

3.1.2 Co-occurrence and Positional Proximity of Binding sites

Binding sites are frequently known to occur in small groups as opposed to individually in a promoter region. This co-occurrence could be because the binding of a transcription factor to a binding site to recruit RNA Polymerase may not be energetically favorable but a transcription factor interacting with another may result in a stable

three-dimensional conformation such that it now can recruit the RNA Polymerase machinery.^{Car98} It is also known that some groups of binding sites occur in close proximity to each other. Again, the proximity might be to facilitate the interaction of transcription factors. Our algorithm takes these factors into account.

3.2 Algorithm

We propose a novel algorithm to identify motif combinations which are potential binding sites in a given set of potentially co-regulated genes. Our algorithm uses a random sampling algorithm similar to the xMotif^{MK03} algorithm discussed in section 1.4.5. The algorithm takes as input the following parameters: a set of genes some of which are believed to be co-regulated (called *foreground set* henceforth), the entire genome of the organism (called *background set* henceforth), size of the random sample r , a p -value threshold α and number of iterations n to perform. The algorithm produces as output biclusters (which are sets of potential binding site pairs and the genes they regulate) sorted in increasing order of their p -values (statistical significance).

The algorithm begins by selecting r genes from the foreground set uniformly at random and without replacement. It then enumerates all pairs of motifs, each motif of the pair being of length four, present in the promoter regions of these r genes. It then calculates the over-representation of each motif pair based on a hypergeometric distribution. All the motif pairs with hypergeometric p -value less than α (after correc-

tion for multiple testing using the False Discovery Rate^{BY95}) are said to be significant. The algorithm then extends all significant motif pairs into longer pairs (see Figure 3.1) and calculates the significance of the extended set of motif pairs. The algorithm continues iteratively till significant motif pairs continue to be found and terminates thereafter.

The above method is repeated n times and significant biclusters from each iteration are reported sorted by their hypergeometric p -values.

These steps and the rationale behind them are explained in detail in the next few sections.

3.2.1 Sampling

The very first step in our algorithm is to select a random sample of size r from the foreground set. We hope to be able to include “some” members of the set of co-regulated genes from the foreground set in the random sample. As explained earlier, if the given set of genes is co-regulated, we would expect them to share common binding sites. Thus if we are able to pick a sufficient number of the potentially co-regulated genes in our samples, we would in principle be able to detect the over-represented binding sites by using statistical tests of significance. By sampling multiple times, we increase the probability of being able to include members from the set of co-regulated genes in our sample.

3.2.2 Significance Calculation and Correction for Multiple Testing

Once we have selected a sample, we enumerate all pairs of motifs, each motif of the pair being of length four, present in the genes in our sample. The rationale behind starting with small motif pairs (and then extending them; see section 3.2.3 on extension) as opposed to starting with longer motif pairs, is that binding sites seldom occur as perfect matches in all the genes. They may share certain common regions which match perfectly, interspersed with regions which do not match exactly (referred to as redundancy, henceforth in this thesis). We thus would like to identify these regions of perfect matches as our starting point. If we select larger pairs of motifs (say each motif of size equal to six base pairs greater), we are likely miss many regions that are perfect matches. If we start with smaller motif pairs (say of size 3) they are not likely to be over-represented. We found that starting with pairs of motifs of size four worked best and was also computationally more feasible than starting with longer oligomer pairs.

As an option, we can also take the spatial distribution of the motif pairs into account; since binding sites tend to occur in close proximity to each other. In this case, we divide the positions upstream of the promoter into different bins and consider only the motif pairs that occur in these bins. These bins are 0–100, 100–300, 300–500 and 500–800 base pairs upstream of the promoter region.

For evaluating the significance of a motif pair in our sample i.e. the probability that the motif pair occurs more frequently in the sample than expected at random, we use the hypergeometric test. Let us say that the motif pair occurs in x of the r genes in our sample and in y out of b genes in the background set. We calculate the hypergeometric test statistic X for the occurrence of this event as:

$$P(X \geq x) = \sum_{i=x}^r \frac{\binom{r}{i} \binom{b}{y}}{\binom{b+r}{i+y}}$$

Similarly, we calculate the hypergeometric p -values for each motif pair that occurs in our sample. Since we have calculated the p -values for occurrence of each event independently, we need to correct for the possibility that we may have some false positives. We use the False Discovery Rate (FDR) correction test^{BY95} to address this issue. In order to apply the FDR test, we first sort the p -values of all possible motif pairs, from the lowest to the highest. Let $p_1 \leq p_2 \leq \dots \leq p_m$ be the ordered hypergeometric p -values for m motif pairs present in our sample. We define the testing procedure as follows:^{BY95}

Let k be the largest i such that $p_i \leq \left(\frac{i\alpha}{m}\right)$ where α is a user defined threshold, then reject all motif pairs whose p values are larger than p_k .

The FDR correction is less stringent than the Bonferroni.^{Sim86} It allows a controlled but greater number of false positives while keeping the number of false negatives down. The Bonferroni test on the other hand keeps the number of false positives down but also has a substantially higher rate of false negatives.

3.2.3 Extension of Motif Pairs

Rejecting the non-significant motif pairs prunes the search space drastically. These significant motif pairs are now candidates for being parts of larger binding sites. In order to detect longer regions of perfect matches from them, we now extend (see Figure 3.1) the significant motifs into longer motif pairs that are also statistically significant. For each statistically significant motif pair M , we first try to extend the constituent motifs one base pair to right and one base pair to the left and thus form a motif pair such that each constituent motif of the pair is of length six. For example the motif pair (GTCT,TATA) shown in the first sequence Figure 3.1 extends into (AGTCTT,GTATAT). We similarly extend the motif pair M in other sequences as well. Thus the set of genes that contained the motif pair M splits into smaller groups; each group containing a different extended version of the motif pair M (of length 6). Once we have extended all the motif pairs in this manner, we calculate the significance of the new set of candidate set of motif pairs. If the set of significant motif pairs is not empty, we continue this process of extension. If however at some point the set does become empty, then instead of doing an extension on both sides, we extend the motif pairs only to the right and if this fails, we choose to extend the motif pairs to the left.

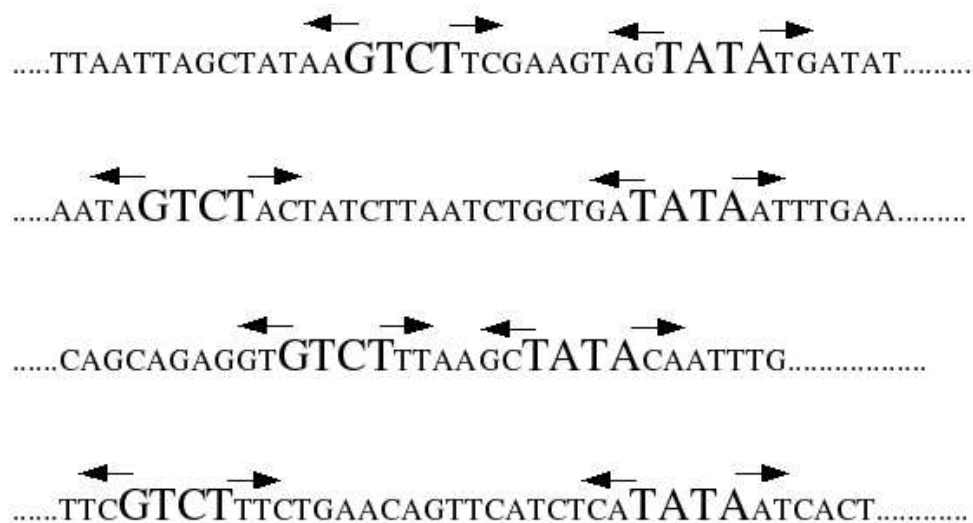


Figure 3.1: The motif pair being considered is GTCT and TATA in the promoter regions of 4 genes. This motif pair would be extended to left and right to AGTCTT and GTATAT. Similar extension is performed for all the other motif pairs as well. If none of the 6-mer pairs pass the significance test, we extend the motif pairs to right. For example this motif pair would get extended into GTCTT and TATAT. If none of these motif pairs pass the significance test, we would extend them to the left. If this fails as well, we choose to extend only one of the motifs keeping the other as it is. Finally, if even this fails, we discard the sample and select a new random sample from the foreground set.

Chapter 4

Simulations

Since our algorithm involves repeated sampling, we would like to know how many times we need to sample to have a “reasonable” chance of detecting over-represented binding sites. In other words, we want to know whether the number of samples required become prohibitively large to be performed within the constraints of the computational resources commonly available. Another issue to be addressed is the variability in the proportions of potentially co-regulated genes (in a given set of genes which forms the foreground set for the algorithm). We would like to study the performance of the algorithm for different proportions of co-regulated genes present in the foreground set.

In this section we address these issues. Given a motif pair, we adopt the following terminology:

n_b : Total number of genes in the genome.

n_f : Total number of genes in the foreground set.

n_c : number of genes in the foreground set that contain the motif pair.

n_{pb} : Number of genes in the background set that contain the motif pair

r : Number of genes in the sample.

α : cutoff value

p_s : Success probability

We frequently use the term *foreground frequency* in this thesis to refer to $\frac{n_c}{n_f}$, the fraction of the foreground genes that are co-regulated (i.e. contain the motif pair).

Similarly, *background frequency* refers to $\frac{n_{pb}}{n_b}$, the fraction of genes in the background set that contain the motif pair.

Let us say that we picked a sample of size r . Let X be the number of co-regulated genes in the sample. Let n_r be the number of genes from n_c that would need to be in this sample so that the hypergeometric p -value for this event is less than α . This would allow us to reject the null hypothesis that the motif pair was over-represented by chance.

The probability that the motif pair came from the background set can be modeled as hypergeometric distribution as follows :

$$P(X \geq n_r | n_b, n_{pb}) = \sum_{i=n_r}^{\min(r, n_c)} \frac{\binom{n_b-r}{n_{pb}-i} \binom{r}{i}}{\binom{n_b}{n_{pb}}} < \alpha$$

We use the above equation for computing the value of n_r .

For our simulations, we can calculate the probability of obtaining $X \geq n_r$ under the alternative hypothesis that the foreground frequency is greater than background frequency. This is calculated as follows:

$$P(X \geq n_r | n_f, n_c) = \sum_{i=n_r}^{\min(r, n_c)} \frac{\binom{n_c}{i} \binom{n_f-n_c}{r-i}}{\binom{n_f}{r}}$$

Also, a motif pair can occur in a gene in the sample either by chance (from the background) or because the gene belongs to the true set of co-regulated genes (from the foreground). Thus the probability of failure i.e. our sample does *not* include the n_r of the n_c co-regulated genes (in a single trial) is:

$$p_f = (1 - P(X \geq n_r | n_b, n_{pb})) (1 - P(X \geq n_r | n_f, n_c))$$

Let us say that we need to sample at least s_r times to have a given success rate of p_s i.e. the probability that at least one sample is able to pick up at least n_r of n_c genes. In other words, if we sample s_r times, we would have a certainty of p_s that we could to detect the over-represented binding site.

$$p_f^{s_r} = (1 - p_s)$$

or

$$s_r = \frac{\log(1-p_s)}{\log(p_f)}$$

However, if we sample s_r times, we also need to correct for multiple hypothesis testing. Applying the Bonferroni correction for multiple testing, the correct threshold for calling a sample significant would be $\frac{\alpha}{s_r}$. To obtain an improved estimate of s_r , we now recompute it for the new significance threshold keeping the other parameters the same. We iterate this process till s_r becomes constant or n_r becomes larger than n_c . If n_r is larger than n_c we conclude that for the given values of input parameters, the algorithm would not find the over-represented set of binding sites. Note that although we use False Discovery Rate (FDR) correction in our algorithm, in our simulations, we use the Bonferroni correction for computational efficiency. Since the Bonferroni correction is more conservative than the FDR, the values of s_r and n_r obtained by applying the Bonferroni correction are in fact the upper bounds on the values we would have obtained had we used the False Discovery Rate correction.

4.1 Exploring the Parameter Space

In this section we study the effects of the different parameters to the algorithm viz. foreground and background frequencies ($\frac{n_c}{n_f}$ and $\frac{n_{pb}}{n_b}$ respectively), α and r , on s_r and n_r . We also study the conditions under which running the algorithm is infeasible.

4.1.1 Sample Size

We used the simulation to study the effect of sample size r on s_r . We observed that one would need to sample less number of times when r is increased.

Table 4.1 shows the effect of sample size on s_r at foreground frequency of 0.7 and $\alpha = 0.05$. Certain values have been denoted by a '-'. This means that for these cases n_r is greater than r to cross the significance threshold α , which is not possible.

To illustrate what each row in the table means, consider row 1 of the table for $r = 30$. This set of data means that if the motif pair occurs in 70% of the foreground genes and at most 60% of the background genes, we would need to sample 518 times for the algorithm to be able to detect it with a confidence of p_s . Note that if the motif pair occurs in greater than 60% of the background genes, the algorithm would not be able to detect it as significant at this threshold. If the motif pair occurs in less than 60% of the background genes, the algorithm would succeed in detecting it in fewer trials.

Note that the value of foreground frequency is chosen just for illustration. Our simulations show that the conclusion is still valid for different values of the foreground frequency.

As further illustration, we also study the parameter n_r at different values of background frequencies and r . This is shown in Figure 4.1.

Table 4.1: Effect of sample size r on s_r at foreground frequency = 0.7 and $\alpha = 0.05$. Here, $n_f = 100$, $n_b = 1000$ and $p_s = 0.95$

$r = 10$		$r = 30$	
Max. Background Frequency	s_r	Max. Background Frequency	s_r
0.60	-	0.60	518
0.55	-	0.55	26
0.50	-	0.50	5
0.45	83	0.45	3
0.40	17	0.40	2
0.35	6	0.35	1
0.30	3	0.30	1

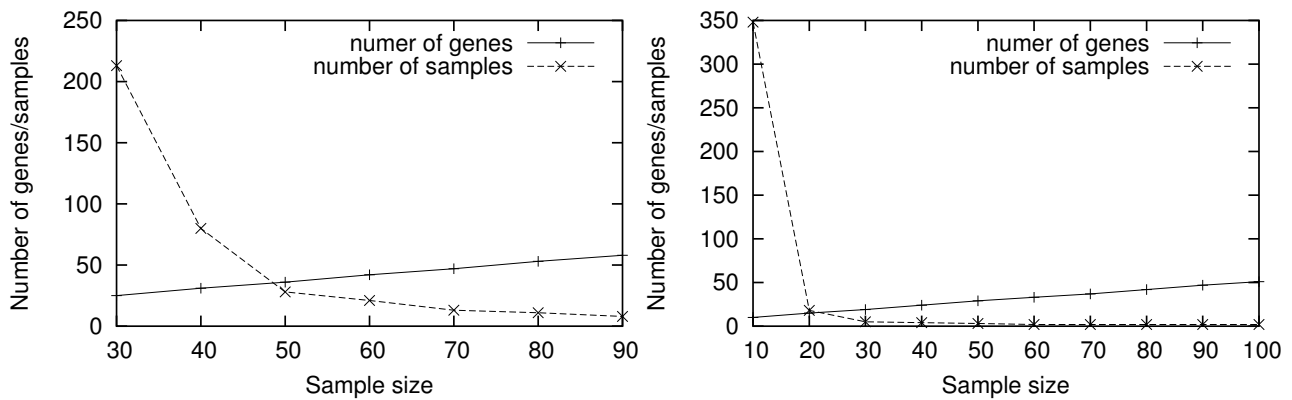


Figure 4.1: Variation in number of genes n_r and number of samples required s_r at foreground frequency of 0.6 and background frequencies of 0.5 and 0.4 respectively. In all cases, $n_f = 100$, $n_b = 1000$, $p_s = 0.95$ and $\alpha = 0.05$

4.1.2 Number of Samples Required till Success

We studied the variations in the values for the parameter s_r at different foreground and background frequencies. We noticed that s_r becomes exceedingly large when the foreground frequency is low and the background frequency is close to the foreground frequency (but less) as can be seen from Table 4.2.

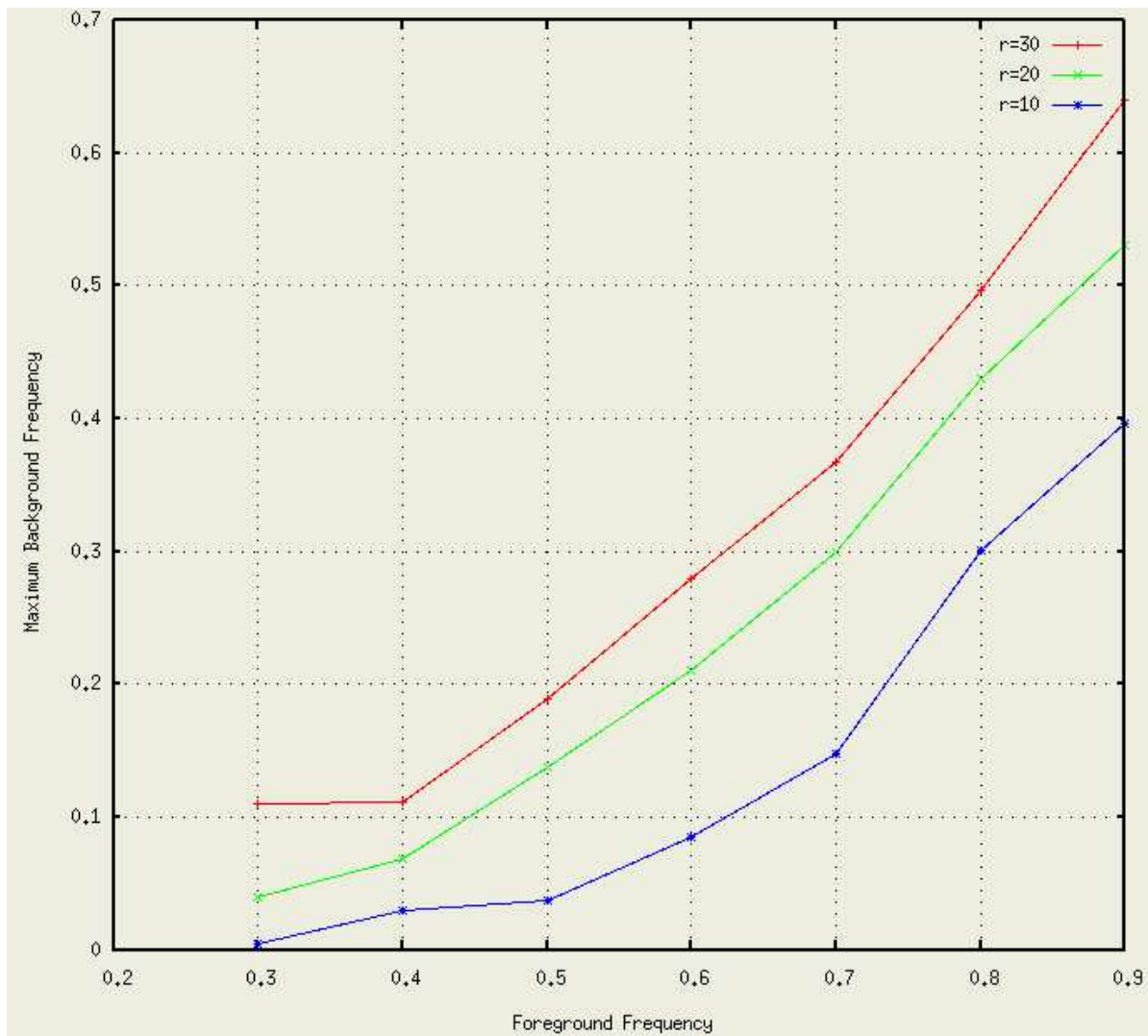


Figure 4.2: Foreground frequency vs maximum background frequency such that $s_r = 1$. Here, $n_f = 100$, $n_b = 1000$, $p_s = 0.95$ and $\alpha = 0.05$

Table 4.2: Effects of background and foreground frequencies on s_r at $r = 20$ and $\alpha = 0.05$. Here, $n_f = 100$, $n_b = 1000$ and $p_s = 0.95$

Max. Background Frequency	Foreground Frequency			
	0.75	0.5	0.25	0.1
0.60	508	-	-	-
0.50	6	-	-	-
0.40	2	4669	-	-
0.30	1	17	-	-
0.20	1	3	15531	-
0.15	1	1	109	-
0.10	1	1	11	-
0.09	1	1	6	-
0.08	1	1	4	24596
0.07	1	1	4	5258
0.06	1	1	4	1273
0.05	1	1	4	107
0.04	1	1	1	38
0.03	1	1	1	38
0.02	1	1	1	15
0.01	1	1	1	7

4.1.3 Foreground and Background Frequencies

In this section we study the effects of the differences in foreground and background frequencies on the number of samples required s_r . This is best illustrated by figure 4.1. We calculated s_r at different values of foreground and background frequencies. We found that the foreground frequency must be higher than background frequency for us to be able to detect the combinations.

Chapter 5

Biological Data Analysis

We analyzed the gene expression data for the yeast cell cycle produced by Spellman et. al.^{SSZ+98} They created a catalog of genes (and the binding sites for transcription factors for these genes) that are known to be active during the different phases of the yeast cell cycle. They list CACGAAA,^{SSZ+98,Nas85,AH89,LJJ91,MASS91} ACGCGT,^{SSZ+98,KMN+93} TTACCNAATTNGGTAA^{SSZ+98,AZV97} (where N=A,G,C or T), GTMAACAA (where M=A or C),^{SSZ+98,ASW+95} ACCAGC,^{SSZ+98,DVS96} ACCAGC^{SSZ+98,KBSN96} as the binding sites of the transcription factors that regulate the entire 4 phases of the yeast cell cycle.

They assigned about 800 yeast genes to the 4 distinct phases of the yeast cell cycle. Note that the assignment was not mutually exclusive; some genes were in fact assigned to more than one phase. An interesting aspect of the yeast cell cycle is that the genes involved in the different phases are also sometimes regulated by the cell cycle

itself. The genes participate in processes such as DNA synthesis, budding, cytokinesis.^{SSZ⁺98} The cell cycle thus is a self-regulating program.

We analyzed the stage *G1* of the yeast cell cycle. Spellman et. al identified 205 genes that participate in this phase. The transcription factor complexes MBF (composed of Mbp1 and Swi6) , SBF (composed of Swi6 and Swi4) are known to be active in this phase.

5.1 Algorithm Parameters

The 205 genes from the stage *G1* of the yeast cell cycle form our foreground set and the other genes from the yeast genome form the background set. Spellman et. al. also analyzed microarray expression data for the yeast cell cycle. They performed hierarchical clustering^{ESBB98} on gene expression data corresponding to each phase of the cell cycle and found that stage *G1* had clusters which contained about 80 genes in them. We use this observation as the starting point of our algorithm. We used simulations (from chapter 4) to decide how many iterations to perform. Even by conservative estimates on the proportion of background genes that would have the binding site combinations at random we were required to sample just once. Note however, that this does not mean that the algorithm requires *a priori* information about the number of times to sample (r). r is in fact a user defined parameter. We ran the algorithm with $\alpha = 0.01$ and $r = 40$.

```

2.61707e-12      AAGAAG:ACGCGT      YAR008W YBL035C YBR007C YCR065W YDL101C YDR279W YGR189C YHR110W YKL089W YLR032W YLR313C YNL102W YPR075C YPR175W
1.45794e-11      TAAATT:ACGCGT      YBL035C YBR007C YCR065W YDL164C YDR279W YHR110W YKL089W YKR091W YLR032W YLR313C YNL165W YPR075C YPR175W
1.9997e-11       AAATTT:ACGCGT      YBL035C YBR007C YDL164C YDR279W YEL047C YFR027W YHR110W YKL089W YLR032W YLR313C YLR383W YNL102W YNL165W YPR075C YPR175W
2.02941e-11      ATTTTT:ACGCGT      YAR008W YBL035C YCR065W YDL164C YDR279W YEL047C YFR027W YGR189C YHR110W YKL089W YKR091W YLR313C YLR383W YNL102W YNL165W YPR175W
2.02941e-11      GAAAAA:ACGCGT      YBL035C YBR007C YCR065W YDL101C YDR279W YDR309C YEL047C YFR027W YGR189C YKL089W YKR091W YLR032W YLR313C YNL102W YPR075C YPR175W
6.92164e-11      AAAGAA:ACGCGT      YAR008W YBL035C YBR007C YCR065W YDL101C YDL164C YDR279W YDR309C YEL047C YHR110W YKL089W YLR032W YLR313C YNL102W YPR175W
8.06924e-11      AAAAAG:ACGCGT      YAR008W YBL035C YBR007C YCR065W YDR309C YEL047C YFR027W YGR189C YKR091W YLR032W YLR313C YLR383W YNL102W YNL165W YPR075C
8.30497e-11      AGAAAT:ACGCGT      YAR008W YBL035C YDL101C YDL164C YDR279W YDR309C YHR110W YKL089W YLR032W YLR313C YLR383W YNL102W YPR075C
4.65702e-10      AAAAGA:ACGCGT      YAR008W YBR007C YCR065W YDL101C YDR279W YDR309C YEL047C YFR027W YGR189C YLR032W YNL102W YNL165W YPR075C YPR175W
9.64771e-10      YBL035C YDL101C YDL164C YDR279W YDR309C YEL047C YHR110W YKL089W YLR032W YLR313C YLR383W YNL102W YPR075C
1.13848e-09      GAAAAA:ACGCGT      YBL035C YBR007C YCR065W YDL101C YEL047C YFR027W YGR189C YKR091W YLR032W YLR313C YNL102W YPR075C
1.27529e-09      AAAAAA:ACGCGT      YAR008W YBL035C YBR007C YCR065W YDL101C YDR279W YDR309C YEL047C YFR027W YKL089W YLR032W YLR313C YNL102W YPR075C
1.45521e-09      AAAAGT:ACGCGT      YAR008W YBL035C YCR065W YDL101C YDR279W YHR110W YKL089W YLR032W YLR313C YLR383W YNL102W YPR175W
2.15079e-09      ATTCAA:ACGCGT      YBL035C YCR065W YDR279W YDR309C YEL047C YFR027W YGR189C YHR110W YLR383W YNL102W YPR075C
2.18536e-09      TTTTTC:ACGCGT      YBL035C YCR065W YDL164C YDR279W YEL047C YFR027W YGR189C YHR110W YKL089W YLR032W YLR383W YNL102W YNL165W YPR175W
2.38763e-09      GAAGAA:ACGCGT      YBL035C YBR007C YCR065W YDL164C YDR279W YFR027W YGR189C YHR110W YKL089W YLR032W YLR313C YPR175W
1.26094e-08      CTCTCT:ACGCGT      YAR008W YDL164C YDR279W YDR309C YEL047C YFR027W YKL089W YKR091W YLR032W YNL165W YPR175W
1.58662e-08      GCAGGT:ACGCGT      YAR008W YDL101C YGR189C YHR110W YKL089W YLR383W YNL165W
1.67425e-08      ATAAAT:ACGCGT      YBL035C YCR065W YDL101C YDL164C YDR279W YHR110W YKL089W YKR091W YLR032W YLR313C YPR075C
1.69812e-08      AAAACG:ACGCGT      YDL101C YDR309C YEL047C YFR027W YHR110W YLR313C YLR383W YNL102W YPR075C YPR175W
1.69812e-08      TAAAGA:ACGCGT      YAR008W YBL035C YDR279W YDR309C YEL047C YKL089W YKR091W YLR032W YLR383W YPR075C
2.48316e-08      CAAAAA:ACGCGT      YAR008W YBL035C YBR007C YCR065W YDR309C YFR027W YKL089W YLR032W YLR313C YLR383W YNL102W YPR075C
2.58994e-08      TCTCTT:ACGCGT      YAR008W YDL164C YDR279W YEL047C YHR110W YKR091W YLR313C YLR383W YNL102W YNL165W YPR175W
2.80279e-08      TTTTCT:ACGCGT      YAR008W YBR007C YDL164C YDR279W YDR309C YEL047C YFR027W YHR110W YKL089W YLR032W YLR383W YNL102W YPR175W
3.36742e-08      TTTCAT:ACGCGT      YAR008W YBL035C YDL164C YEL047C YGR189C YHR110W YKR091W YLR032W YLR383W YNL165W YPR175W
4.05447e-08      TCAAAA:ACGCGT      YAR008W YCR065W YDL101C YDR279W YEL047C YFR027W YKL089W YLR032W YLR313C YLR383W YNL102W YPR075C
4.1324e-08        TCAAAA:ACGCGT      YBL035C YCR065W YDR279W YDR309C YFR027W YHR110W YKL089W YLR032W YLR383W YNL102W YPR075C

```

Figure 5.1: Sample output produced by our algorithm. First column denotes the p -value of the biclusters. The second column denotes the candidate binding site pairs (delimited by a “:”). The third column is the yeast genes (ORFs) these binding sites may regulate.

5.2 Significant Combinations

The output of this run is available online at

<http://bioinformatics.cs.vt.edu/~svenkat/spellman/g1-0.01.txt>.

A sample output is shown in Figure 5.1.

We found that the motif ACGCGT was significantly over-represented in combination with other motifs. ACGCGT is the binding site of the complex MBF (comprised of transcription factors Swi6p and Mbp1p) which is known to be active during the $G1$ phase of the cell cycle.^{SSZ+98} This observation was also in agreement with predictions from earlier computational approaches.^{KHB+04}

5.2.1 Potential Interactions between HSF1 and MBF

We found co-occurrence of potential binding sites for HSF1 transcription factor and the MBF complex. The HSF1 binding site is consists of AGAAN and its inverted repeat NTTCT (it is in fact AGAANNTTCT).^{FXL94,WCF+01} The most over-represented combination in our result was those of AAGAAG and ACGCGT (with p -value of 2.61707×10^{-12}). Note that AAGAAG contains one half of the consensus binding site of the transcription factor HSF1. We also detected several significantly over-represented combinations of the form NTTCTN-ACGCGT which correspond to the other half of the HSF1. For example we detect the CTTCTT , ACGCGT combination with p -value of 1.26094×10^{-8} and another combination of TTTCTT and CGCGT with p -value of 2.80279×10^{-8} . The co-occurrence of these binding sites suggests that the transcription factors may interact.

5.2.2 Co-occurrence of Potential Sites for Mbp1 and SWI4

The SWI4 consensus binding site is CRCGAAA.^{IHS+01} We recovered the binding site CGAAAAAA (the partial binding site of SWI4) which occurred in combination with Mbp1 binding site AACGCGTC, with a p -value 2.1578×10^{-5} . This suggest that there is a possibility that the transcription factors SWI4 and Mbp1 interact. This result is also in close agreement with those found by Zhang et. al.^{KHB+04}

Chapter 6

Conclusions and Future Work

Identification of binding sites is a problem that is still far from solved. Binding sites for a transcription factor may differ considerably between different genes. The approach presented here tries to leverage the notion of co-occurrence of binding sites. The approach identified some of the known binding sites in a particular phase of yeast cell cycle besides also identifying a novel binding site combination. Our biclustering technique efficiently prunes the search space for candidate binding sites and thus is scalable and can be applied to larger genomes.

Our algorithm, like many motif finding algorithms^{vHACV98,ST02} is based on the over-representation of transcription factor binding sites. However we note that the complete binding site for a transcription factor need not be over-represented. Parts of the binding site may be perfectly conserved and may be over-represented. Even though

our algorithm may succeed in recovering these perfectly conserved sequences, the algorithm currently has no procedure for recovering entire binding sites from these shorter sequences. An invaluable improvement to the algorithm would be a method to recover complete binding site sequences. It would also be worthwhile to study ways to incorporate additional biological data into the algorithm. The algorithm thus far has been tested only mainly on the yeast cell cycle data (besides simulated data sets). It would be interesting to run the algorithm on sequences from other species.

Bibliography

- [AH89] B. J. Andrews and I. Herskowitz. Identification of a DNA binding factor involved in cell-cycle control of the yeast HO gene. *Cell*, 57(1):21–29, April 1989.
- [ASW⁺95] H. Althoefer, A. Schleiffer, K. Wassmann, A. Nordheim, and G. Ammerer. Mcm1 is required to coordinate G2-specific transcription in *Saccharomyces cerevisiae*. *Mol Cell Biol*, 15(11):5917–5928, November 1995.
- [AZV97] T. B. Acton, H. Zhong, and A. K. Vershon. DNA-binding specificity of mcm1: operator mutations that alter DNA-bending and transcriptional activities by a MADS box protein. *Mol Cell Biol*, 17(4):1881–1889, April 1997.
- [BWD96] M. Blatt, S. Wiseman, and E. Domany. Superparamagnetic clustering of data. *Physical Review Letters*, 76(18):3251–3254, April 1996.
- [BY95] Yoichi Hocheberg and Yoichi Benjamini. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal*

Statistical Society, 57, 1995.

- [Car98] M. Carey. The enhanceosome and transcriptional synergy. *Cell*, 92(1):5–8, January 1998.
- [CC00] Y. Cheng and G. M. Church. Biclustering of expression data. *Proc Int Conf Intell Syst Mol Biol*, 8:93–103, 2000.
- [CMK⁺03] D. Y. Chiang, A. M. Moses, M. Kellis, E. S. Lander, and M. B. Eisen. Phylogenetically and spatially conserved word pairs associated with gene-expression changes in yeasts. *Genome Biol*, 4(7), 2003.
- [DVS96] P. R. Dohrmann, W. P. Voth, and D. J. Stillman. Role of negative regulation in promoter specificity of the homologous transcriptional activators *ace2p* and *swi5p*. *Mol Cell Biol*, 16(4):1746–1758, April 1996.
- [ESBB98] M. B. Eisen, P. T. Spellman, P. O. Brown, and D. Botstein. Cluster analysis and display of genome-wide expression patterns. *Proc Natl Acad Sci U S A*, 95(25):14863–14868, December 1998.
- [FXL94] M. Fernandes, H. Xiao, and J. T. Lis. Fine structure analyses of the drosophila and saccharomyces heat shock factor–heat shock element interactions. *Nucleic Acids Research*, 22(2):167–173, 1994.

- [GLD00] G. Getz, E. Levine, and E. Domany. Coupled two-way clustering analysis of gene microarray data. *Proc Natl Acad Sci U S A*, 97(22):12079–12084, October 2000.
- [GLDZ00] G. Getz, E. Levine, E. Domany, and M. Q. Zhang. Super-paramagnetic clustering of yeast gene expression profiles. *Physica*, Nov 2000.
- [GST⁺99] T. R. Golub, D. K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. P. Mesirov, H. Coller, M. L. Loh, J. R. Downing, M. A. Caligiuri, C. D. Bloomfield, and E. S. Lander. Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science*, 286(5439):531–537, October 1999.
- [Har72] J.A. Hartigan. Direct clustering of data matrix. *Journal of the American Statistical Association*, pages 123–129, 1972.
- [IHS⁺01] V. R. Iyer, C. E. Horak, C. S. Scafe, D. Botstein, M. Snyder, and P. O. Brown. Genomic binding sites of the yeast cell-cycle transcription factors sbf and mbf. *Nature*, 409(6819):533–538, January 2001.
- [KBSN96] D. Knapp, L. Bhoite, D. J. Stillman, and K. Nasmyth. The transcription factor swi5 regulates expression of the cyclin kinase inhibitor p40sic1. *Mol Cell Biol*, 16(10):5701–5707, October 1996.
- [KHB⁺04] Mamoru Kato, Naoya Hata, Nilanjana Banerjee, Bruce Futcher, and Michael Q. Zhang. Identifying combinatorial regulation of transcrip-

- tion factors and binding motifs. *Genome Biology*, 5(8):R56–R56.11, June 2004.
- [KMN⁺93] C. Koch, T. Moll, M. Neuberg, H. Ahorn, and K. Nasmyth. A role for the transcription factors mbp1 and swi4 in progression from g1 to s phase. *Science*, 261(5128):1551–1557, September 1993.
- [KSEA00] M. Koranda, A. Schleiffer, L. Endler, and G. Ammerer. Forkhead-like transcription factors recruit ndd1 to the chromatin of g2/m-specific promoters. *Nature*, 406(6791):94–98, July 2000.
- [LJJ91] N. F. Lowndes, A. L. Johnson, and L. H. Johnston. Coordination of expression of DNA synthesis genes in budding yeast by a cell-cycle regulated trans factor. *Nature*, 350(6315):247–250, March 1991.
- [LMS96] M. Y. Leung, G. M. Marsh, and T. P. Speed. Over- and underrepresentation of short DNA words in herpesvirus genomes. *J Comput Biol*, 3(3):345–360, 1996.
- [LT03] M. Levine and R. Tjian. Transcription regulation and animal diversity. *Nature*, 424(6945):147–151, July 2003.
- [MASS91] E. M. Mcintosh, T. Atkinson, R. K. Storms, and M. Smith. Characterization of a short, cis-acting dna sequence which conveys cell cycle stage-dependent transcription in *saccharomyces cerevisiae*. *Mol Cell Biol*, 11(1):329–337, January 1991.

- [MK03] T. M. Murali and S. Kasif. Extracting conserved gene expression motifs from gene expression data. *Pac Symp Biocomput*, pages 77–88, 2003.
- [MO04] Sara C. Madeira and Arlindo L. Oliveira. Biclustering algorithms for biological data analysis: A survey. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 1:24–45, 2004.
- [Nas85] K. Nasmyth. At least 1400 base pairs of 5'-flanking dna is required for the correct expression of the HO gene in yeast. *Cell*, 42(1):213–223, August 1985.
- [PSC01] Y. Pilpel, P. Sudarsanam, and G. M. Church. Identifying regulatory networks by combinatorial analysis of promoter elements. *Nat Genet*, 29(2):153–159, October 2001.
- [RHEC98] F. P. Roth, J. D. Hughes, P. W. Estep, and G. M. Church. Finding DNA regulatory motifs within unaligned noncoding sequences clustered by whole-genome mRNA quantitation. *Nat Biotechnol*, 16(10):939–945, October 1998.
- [Sim86] R. J. Simes. An improved bonferroni procedure for multiple tests of significance. *Biometrika*, (73):751–754, 1986.
- [SSZ⁺98] P. T. Spellman, G. Sherlock, M. Q. Zhang, V. R. Iyer, K. Anders, M. B. Eisen, P. O. Brown, D. Botstein, and B. Futcher. Comprehensive identification of cell cycle-regulated genes of the yeast *saccharomyces cerevisiae*

- by microarray hybridization. *Mol Biol Cell*, 9(12):3273–3297, December 1998.
- [ST02] S. Sinha and M. Tompa. Discovery of novel transcription factor binding sites by statistical overrepresentation. *Nucleic Acids Res*, 30(24):5549–5560, December 2002.
- [THG94] J. D. Thompson, D. G. Higgins, and T. J. Gibson. CLUSTALW: Improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic Acids Res*, 22(22):4673–4680, November 1994.
- [TSKS04] A. Tanay, R. Sharan, M. Kupiec, and R. Shamir. Revealing modularity and organization in the yeast molecular network by integrated analysis of highly heterogeneous genomewide data. *Proc Natl Acad Sci U S A*, 101(9):2981–2986, March 2004.
- [TSM⁺99] Pablo Tamayo, Donna Slonim, Jill Mesirov, Qing Zhudagger, Sutsak Kitareewandagger, Ethan Dmitrovskydagger, Eric S. Lander, and Todd R. Golub. Interpreting patterns of gene expression with self-organizing maps: Methods and application to hematopoietic differentiation. *Proc Natl Acad Sci U S A*, 96(6):2907–2912, March 1999.
- [TSS02] Amos Tanay, Roded Sharan, and Ron Shamir. Discovering statistically significant biclusters in gene expression data. *Bioinformatics*, 18:S136–

S144, 2002.

- [TSS04] Amos Tanay, Roded Sharan, and Ron Shamir. Biclustering algorithms: A survey. May 2004.
- [vHACV98] J. van Helden, B. Andr, and J. Collado-Vides. Extracting regulatory sites from the upstream region of yeast genes by computational analysis of oligonucleotide frequencies. *J Mol Biol*, 281(5):827–842, September 1998.
- [WCF⁺01] E. Wingender, X. Chen, E. Fricke, R. Geffers, R. Hehl, I. Liebich, M. Krull, V. Matys, H. Michael, R. Ohnhuser, M. Prss, F. Schacherer, S. Thiele, and S. Urbach. The TRANSFAC system on gene expression regulation. *Nucleic Acids Res*, 29(1):281–283, January 2001.
- [Wil68] F. William. An introduction to probability theory and its applications, volume 1. *John Wiley & Sons*, 1968.

Vita

Venkataraman Srinivasan graduated first class with a Bachelor's degree in Computer Science from Delhi Institute of Technology, University of Delhi in May 2003. Since August 2003 he has been a Master's student in Computer Science Virginia Polytechnic Institute and State University. After graduating in August 2005, he returns to India to take up a full time position in the software industry.