Discriminative Learning of Composite Transcriptional Regulatory Modules

by

Tomer Naveh

Submitted to the School of Computer Science and Engineering in partial fulfillment of the requirements for the degree of

Master of Science

at the

HEBREW UNIVERSITY OF JERUSALEM
JERUSALEM, ISRAEL

August 2004

This work was carried out under the supervision of Prof. Nir Friedman

Abstract

A central goal of molecular biology is to uncover transcription regulation mechanisms that govern gene expression. Transcription factors play an important role in those mechanisms, as they affect the transcription rates of genes. Often, such regulatory circuits involve not only one transcription factor but rather several factors that act in concert to modulate the transcription of genes. The recent advances in high-throughput assays, such as microarray experiments and Chromatin Immunoprecipitation, allow us to infer groups of genes that are co-expressed or co-regulated. The challenge is to use this wealth of information to gain insights about transcriptional regulation. In this dissertation, we present a procedure for locating *regulatory complexes* in promoter regions. A regulatory complex represents the binding sites of a pair of transcription factors that act in cooperation. Our procedure takes a *discriminative* approach, searching for regulatory complexes that are overabundant in the promoter regions of the target group of co-expressed genes and are infrequent in the control group of genes outside the target group. By doing this, we filter out phenomena that are shared among both groups, ideally leaving us with the core motifs. We demonstrate the applicability of our method for finding regulatory complexes in a genome-wide analysis of the yeast genome.

Contents

Chapter 1

Introduction

1.1 DNA, Genes and Proteins

Biological organisms encode genetic information in the DNA and transfer copies of it from one generation to the next. The genetic information is stored in one or more DNA molecules, called *chromosomes*. A DNA molecule consists of two strands, each of which is a sequence of four different nucleic acids, or *nucleotides*: Adenine (A), Cytosine (C), Guanine (G) and Thymine (T). Both strands complement each other: if A occurs at some position in one of the strands, then T will occur in the corresponding position on the other strand, and similarly, C will be matched by G. DNA molecules are typically very long; the Human genome, for example, contains approximately 3×10^9 *basepairs* (A-T or C-G nucleotide pairs), organized in 23 chromosome pairs. The DNA encodes the information that is the basis for the operation of an organism. It is important to note, that the same DNA sequences are found in all cells of an organism, in all tissues, and under all conditions (unless damaged). So how do the different tissues exist, and how do different cellular processes take place, if the genetic information is the same?

Loosely speaking, the DNA serves as “operating instructions”. It encodes the “set of tools” that may be used. However, it does not directly determine which of those tools are used at any given point. A computer science allegory may be that the code of our operation is there, but the decision which code instructions will be executed at any given time is a dynamic process. It does not depend solely on the code, but also on environmental conditions and events.

So what exactly are those “tools”? Along the DNA strands, there are short fragments of distinguished subsequences called *genes*. Genes are typically hundreds or thousands of nucleotides long. The nucleotide sequences encode the sequences of the corresponding *proteins*. Proteins are the proletarian of the body, serving as the building blocks of cellular processes. Such processes include cell replication, signaling, metabolism, production of other proteins and more.

1.2 Protein Synthesis

The central dogma of molecular biology divides the process by which a protein is constructed from the corresponding gene into two parts. In the first stage, called *transcription*, a messenger RNA (or *mRNA*) is created by the following process. A complex called *RNA polymerase* binds to a certain location in the DNA known as the *core promoter*, near the *transcription initiation site*. This site is located very close to the DNA sequence of the target gene. The RNA polymerase unwinds the DNA's double helix, forming a local gap between the two DNA strands. It then initiates the formation of the complementary RNA sequence, one nucleotide after the other, on one of the DNA strands. This process of transcription ends when a termination signal is encountered. This signal is a distinguished short fragment occurring on the DNA strand. The mRNA molecule then undergoes several transformations. In some organisms, including Human, a process called *splicing* takes place. In this process, certain parts of the mRNA called *introns*, are removed.

At the second stage of the central dogma, called *translation*, the *mRNA* is translated by ribosomes to the corresponding protein. Ribosomes attach to the mRNA to create the protein sequence by matching each nucleotide triplet with the appropriate amino acid according to the genetic code. This translation is conducted until a *stop signal* is encountered. the ribosomes then disengage from the mRNA and the newly created protein is released. The translation process is conducted repeatedly until the *mRNA* is degraded (for example, by enzymes that affect the stability of the mRNA molecule). The overall process is demonstrated in figure 1.1.

1.3 Transcription Factors

When measuring gene expression levels in different tissues or under various conditions, one observes that those levels vary significantly between different genes, or for the same genes in different measurements. Comparisons between measurements reveals complex patterns that are not trivial to infer, suggesting the exis-

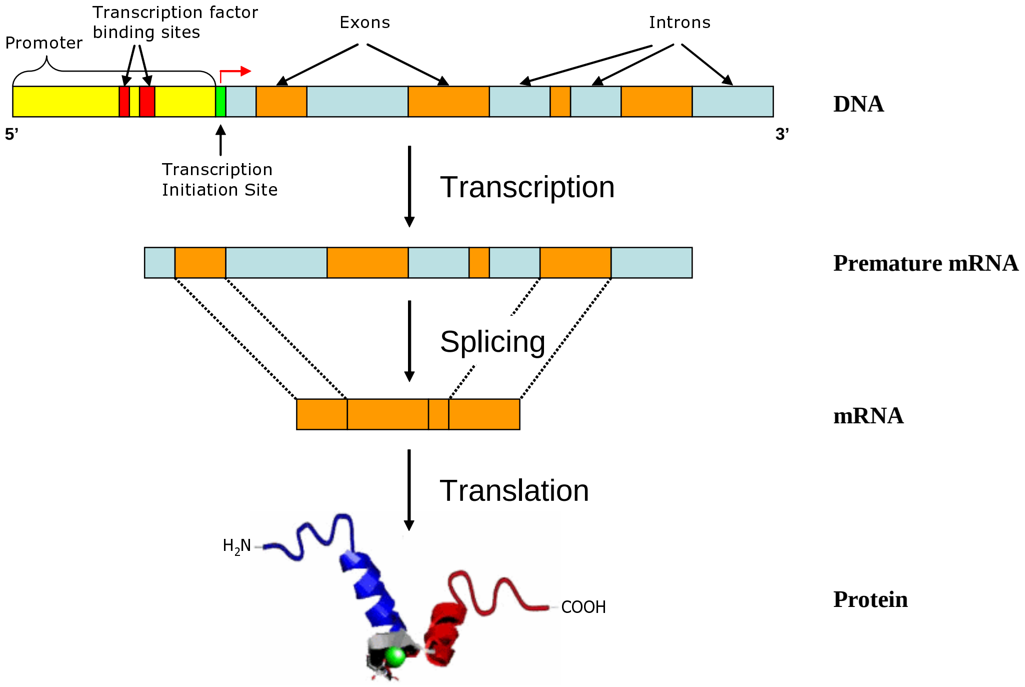


Figure 1.1: **From DNA to protein**. The gene is transcribed by an RNA polymerase, then the introns are cut out. Finally the mRNA is translated into the corresponding protein.

tence of regulatory mechanisms monitoring expression levels of genes. The existence of those mechanisms is also intuitively appealing, since different proteins serve a variety of functions under different cell conditions. There are times when certain proteins are required in high levels whereas others may not be required at all, or may even be harmful. Thus, a mechanism that regulates the expression levels of different genes is needed.

Much of the regulation of expression levels is done at the transcription stage, by regulating the transcription rate of genes. Transcriptional regulation is achieved by *transcription factors*. Transcription factors are proteins that regulate the *transcription rate* of genes (the rate in which a specific gene is transcribed by the RNA polymerase). They bind to the regulatory sequence of a gene, usually located upstream of the gene in an area called the *proximal promoter region*. This binding sets the ground for the binding of the RNA polymerase machinery, by forming a complex that helps it to bind. Transcription will only take place in the presence of the appropriate transcription factors, and thus the amount of those factors currently present and active in the cell directly affects the transcription rate of the target gene. Transcription factors operating in this manner are called *activators*. This mechanism can also work in a negative way. Certain transcription factors (*inhibitors* or *repressors*) may bind in a way that prevents the RNA polymerase or other required factors to bind to the DNA, thus suppressing the transcription of the target gene. The context in which transcription factors operate is illustrated in figure 1.2.

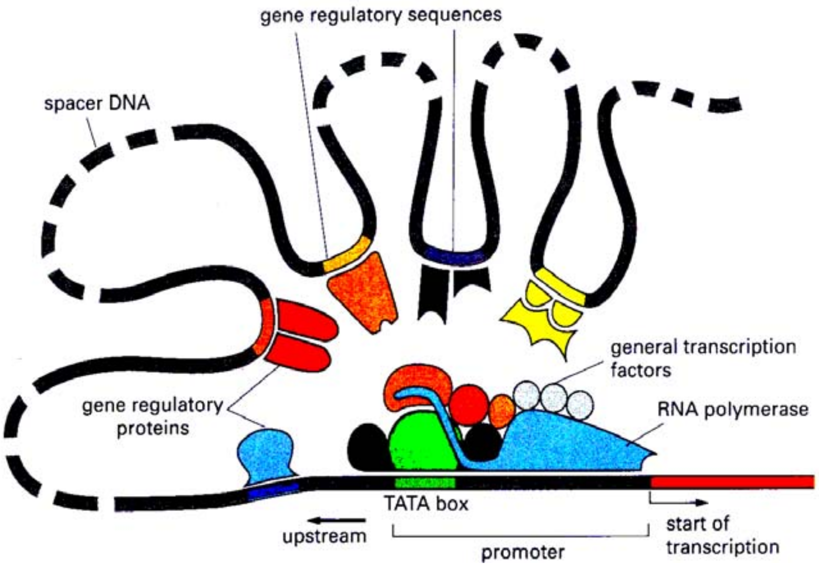


Figure 1.2: Transcription factors bind to the regulatory region upstream of the gene to be transcribed. Following that, the RNA polymerase binds to the core promoter and transcription may begin.

Transcription regulation mechanisms are often much more complex than simply the binding of one factor to increase or decrease the transcription rate of a gene. Complexity arises in two aspects. First, cellular processes are dynamic, so we may consider transcription as a process along the time axis. A certain transcription factor may regulate another gene, which is in turn transcribed into another transcription factor, regulating a different gene and so on. In some cases, known as *feedback loops*, one or more transcription factors may indirectly regulate their own transcription rates in future stages. Second, we frequently observe a group of transcription factors interacting to regulate the expression of a gene.

For example, the binding of a regulatory complex of two interacting factors may be required for the RNA polymerase to bind to the core promoter. We call such pairs *cooperating factors*. In many biological systems a single transcription factor has many cooperating factors. Hence the operation of this transcription factor has conditional effects that depend on which of its cooperating factors are present. The complexity and building blocks of regulatory networks encoded in the regulatory regions of genes is not yet fully understood. However, it is apparent that the regulatory mechanisms that govern gene expression patterns involve context specific cooperation between transcription factors, and that this kind of cooperation requires multiple regulatory elements occurring in proximity [1, 33]. Such *combinatorial regulation* allows a small number of transcription factors to encode a large number of regulatory states. For example, the yeast factor Swi6 is part of both SBF and MBF transcriptional complexes, each acting on a different group of target genes [13]. The complexity of transcription factors cooperation in yeast cell cycle is demonstrated in figure 1.3. The combinatorial nature of gene regulation may be even more complex in higher organisms. Berman *et al.* [6] studied the regulation of the *Drosophila* even-skipped gene (*eve*), and showed it to be regulated by heterogeneous clusters of more than 13 binding sites (of five different transcription factors) within a windows of 700bp (basepairs). In humans, TGFβ is known to have a key role in development and carcinogenesis. The cell's response to TGFβ is mediated via the SMAD-interacting proteins, whose cooperation with different transcription factors allow the versatility and diversification of the TGFβ response [10].

Transcription is only one stage of the gene expression process. This process is subject to regulation by other players as well: *alternative splicing* is a mechanism by which the mRNA may be spliced in more than one way to form the final sequence that is translated into proteins. It enables insertion and removal of different functional cassettes from the protein sequence, and by that modifying its function. The choice of splicing therefore determines the resulting protein, and hence affects the expression of the gene. Gene expression levels may also be regulated by modifications in translation rates of mRNAs, i.e. the number of times an mRNA is translated into a protein in the ribosome within a given period, or in mRNA stability, affecting the timeframe in which an mRNA molecule can be translated. Post-translational mechanisms regulate gene expression by modifying the protein. A common example is that a protein may become active only when a phosphate group is attached to it. The addition of the phosphate group is done by certain enzymes (kinases), whose quantity in the cell thus affects the level of

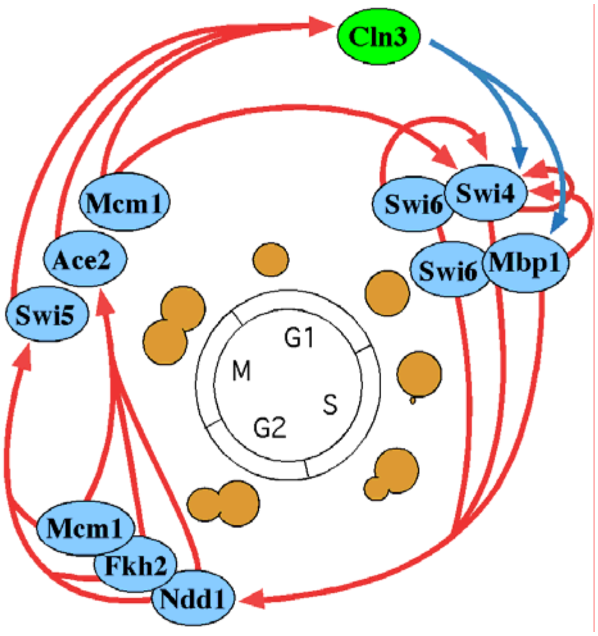


Figure 1.3: **Rich control of regulation in yeast cell cycle**. The figure (from Simon *et al.* [26]) shows interactions between dominant transcription factors in yeast cell cycle. As can be seen, along the different stages of the cell cycle, groups of factors work in concert to regulate the transcription of others in future stages. There are several such regulatory complexes that regulate one or more factors, which in turn regulate other factors and so on. The complexity of cross-talk between transcription factors is apparent even when only a handful of factors are considered. It is important to note that the actual relationships are much more complicated. Many more transcription factors are active during the different cell-cycle phases, and even more genes which are not necessarily transcription factors are regulated by these factors

active protein. In other cases, such mechanisms may cut of a part of the protein, rendering it inactive.

Transcriptional regulation is a dominant part of the different mechanisms that govern gene expression levels, and its basic principals are relatively known. Thus, this dissertation focuses on this aspect of gene expression regulation.

1.4 Binding Sites

Most transcription factors bind a specific sequence of nucleotides in the regulatory regions of genes. Those subsequences, called *cis*-regulatory elements or *binding sites*, are typically short and range between 6 to 20 bps. Binding sites are generally quite conserved. When examining the subsequences to which a certain factor would bind we may find, for example, that binding occurs only if the nucleotides in the binding site are TGACTCA (the consensus sequence for the binding site of the yeast transcription factor Gcn4). Such conserved sites are quite rare, however. In other cases, if the subsequence is similar enough to some consensus sequence, it would suffice for binding to occur. It is often observed that not all positions are equally important. Meaning that some positions do not affect the binding affinity, or affect it much weaker than other positions. For instance, it is possible that 4 specific positions out of 10 in the binding site can accommodate any nucleotide, while the other 6 must be conserved. Another common situation is that in certain positions, not all nucleotides may occur, but rather a subset of $\{$A,C,G,T$\}$, possibly with a different effect on the binding affinity (e.g. A strongly increases the affinity, C increases it only moderately, and G and T decrease the binding affinity substantially and thus are not observed in binding sites). Characterizing binding sites for transcription factors is a central goal on the path for understanding regulatory mechanisms in the cell, as it may help us to get a global view of the transcription factors involved in the regulation of each gene. We may use this information to gain knowledge about various functions of different genes. Genes that are involved in the same regulatory mechanisms, either as regulators or targets, are potentially taking part in the same biological processes. Achieving this *in silico* pinpoints promising candidates for *in vitro* verification.

Chapter 2

Algorithmic Approaches

2.1 Representing Binding Sites

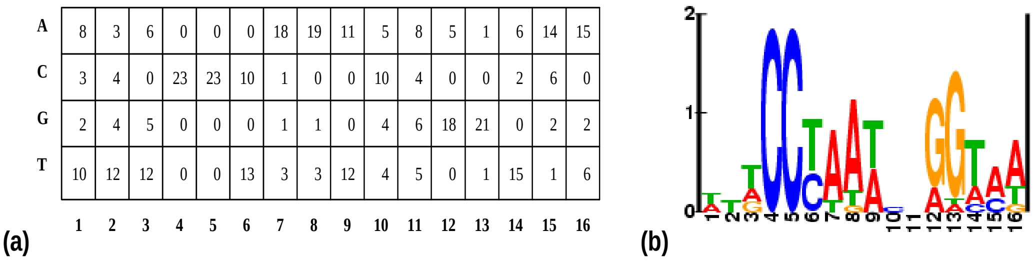
When one confronts the question of how to characterize binding sites and *in silico* one must first consider the question of how to effectively represent those binding sites mathematically. The binding sites of a transcription factor are DNA subsequences to which the transcription factor has high binding affinity. As previously discussed, those subsequences are often similar to each other as they must be recognized by the transcription factor, and thus must be composed of nucleotides that together create high binding affinity to for the specific factor. Thus, the set of putative binding sites for a transcription factor can be characterized by a *motif*, so that subsequences adhering to this motif are more likely to serve as binding sites. This choice of motif representation is subject to a tradeoff between the simplicity of the representation and its expressiveness. A simple representation may be easier to deal with computationally, and more intuitively appealing. However, too simple a representation may not well capture the characteristics of the subsequences to which the factor may bind.

On one side of the complexity scale for motif representations, we find the simplest representation of a *consensus sequence*. Under this representation, we align the known binding sites of a factor and take the most frequent nucleotide in every position to construct the motif. Although this representation of contiguous consensus sequences is straightforward, and easier to work with when trying to discover motifs or binding sites, it clearly does not capture well the biological richness of binding sites. For example, if half of the sites have A in one position and others have T in that position, this model discards half the sites in this position. It also does not capture differences in specificity between positions, such as positions that do not affect binding.

On the other extreme, one may represent a motif by the list of known or conjectured binding sites for the transcription factor. This approach encompasses all known sequence information about the motif, however it does not capture the essence of the binding sites but merely lists their different instances, and thus cannot be used to discover novel binding sites, or to discover motifs when an initial list of binding sites is unknown.

In between there are several other popular representations. One is to use consensus sequences over the IUPAC alphabet (an alphabet of 15 symbols, one per subset of nucleotides). This allows for better characterization of positions that do not have one dominant nucleotide. An extension of consensus sequences introduces the notion of *wildcards*. Wildcards are positions in which we allow all nucleotides to occur. For example, we can represent a motif by AC*ACG*T, so that positions 3 and 7 are ignored when examining a putative binding site. This approach models the fact that certain positions in binding sites do not influence the binding affinity, while maintaining a relatively simple motif model.

The above models are all using discrete spaces for motifs. Moving into continuous spaces, one may wish to construct a probabilistic model by assigning a distribution over the nucleotides $\{$A,C,G,T$\}$ to each position in the motif. Such a model is called a *profile* of the motif. It is usually represented as a matrix where the i^{th} column corresponds to the distribution of the nucleotides in the i^{th} position of the binding site. This model captures much of the biological characteristics of binding sites described above, as its only assumption is that positions in the binding sites are independent. Another common probabilistic representation for motifs is that of a *position specific scoring matrix* (PSSM, sometimes referred to as *position weight matrix* or PWM). This representation is tightly related to the profile of the motif, but it also takes into account a background distribution of nucleotides (e.g. the distribution of nucleotides across all promoters of the organism under study) by keeping a weight for each nucleotide in each motif position, which is the log of the ratio between the probability of a nucleotide in a motif position and the background probability for this nucleotide. An example of a profile can be seen in Figure 2.1.



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
A	8	3	6	0	0	0	18	19	11	5	8	5	1	6	14	15
C	3	4	0	23	23	10	1	0	0	10	4	0	0	2	6	0
G	2	4	5	0	0	0	1	1	0	4	6	18	21	0	2	2
T	10	12	12	0	0	13	3	3	12	4	5	0	1	15	1	6

Figure 2.1: (a) The counts of different nucleotides in each position of the Mcm1 binding site, based on known binding sites for this factor from the TRANSFAC database [34]. (b) Graphical representation of the profile data as a sequence logo. Each position represents a nucleotide distribution of one position in the binding site. The height of each nucleotide is proportional to its frequency in that position, and the total height of each position is proportional to its information content. This way we demonstrate which nucleotides occur with higher probabilities, but also the information content at the specific position, which illustrates how well this position is conserved.

2.2 Discovering Motifs

The "*motif finding problem*" can be informally stated as follows: *Given a set of genes that are conjectured to be co-regulated by a common transcription factor, find the best motif that is common to the regulatory regions of those genes, to which the transcription factor is likely to bind*.

A variety of methods have been developed for tackling these problems. They all follow a common scheme: First, one must identify a set of co-regulated genes. This can be achieved, for example, by taking a group of genes that share a common pattern over a series of expression measurements e.g. [21, 23, 28, 31], or genes that are known to be involved in the same biological process [14, 19] or that are homologous to known co-regulated genes in related species [18, 19].

Second, one must extract the regulatory regions of those genes. This task is not at all trivial. When dealing with organisms such as *Saccharomyces cerevisae* (the Baker's yeast) one usually considers the *proximal promoter*, i.e. the area located in proximity to the transcription initiation site, as the regulatory region. One may take out of the intergenic regions, say, 1000 bps upstream of each gene to construct the group of input promoters. The motivation is that by doing this, one gets much of the regulatory regions without introducing too much noise, but this by no means is an accurate extraction of the relevant regions. On higher organisms such as Human, Mouse and Fly, this task is much harder, as the approach described above will not perform well. On such organisms complex tools for promoter prediction, based on sequence and structural signals, must be used.

Finally, one applies a motif finding algorithm on the promoter regions, extracting motifs that are likely to characterize the binding sites of the transcription factor under study. As a post-processing application, the resulting motif can be used for genome-wide mining of additional putative binding sites of the transcription factor.

Motif finding algorithms focus on the last stage of the above scheme, where the regulatory sequences and putative annotation of co-regulated genes are given, and the objective is to find a common motif. They differ by the choice of motif representation, the choice of algorithm for searching the optimal motif subject to the chosen representation, and the choice of scoring function used to evaluate motifs. Next we survey established methods in this field.

2.2.1 Discrete Representation of Binding Sites

Several methods use the representation of conserved sub-sequences, allowing for multiple choices at different positions or for binding sites to contain up to a constant number of mismatches [7, 17, 27, 32]. This choice of relatively limited representation allows for exhaustive search in the space of motifs, and so guarantees that the optimal motif [1] is reported. Motifs are scored by their statistical overabundance in the promoter regions of a target group of genes, sometimes with relation to a statistical background model for the distribution of subsequences in the genome under study, e.g. a k^{th} order Markov model. The search is conducted either by systematically considering all patterns in the search space (or those that are present in the input sequences, in a data-driven approach) [7, 27, 32] or by more efficient methods such as *suffix trees* [17].

Pevzner and Sze [20] use a similar representation but construct search algorithms that rely on combinatorial characterization of the motif finding problem. They also formulate the motif finding problem as a combinatorial problem. This gave rise to several algorithms utilizing this framework to demonstrate their performance. Buhler and Tompa [8] do not conduct an exhaustive search, but rather use a randomized technique called *random projections*. This technique uses the

[1] Optimality is guaranteed for the scoring function used. Different algorithms may thus result in different optimal motifs.

representation of consensus sequences with wildcards, where the mask of wildcards is chosen at random. With some probability the choice of non-wildcard positions will match the more informative positions of the motif, and the motif will be detected by overabundance. This process is repeated many times to increase the probability of motif detection. They also show that the probability of motif detection and running time of the algorithm can be computed when the motif finding problem is formulated as by Pevzner and Sze [20]. The idea of testing many masks of wildcards also corresponds to our biological knowledge about the structure of binding sites. We know that different positions have different specificity, and thus would like to focus on those that are more informative, however we do not know those positions a-priori.

2.2.2 Probabilistic Representations of Binding Sites

The major drawback of discrete representations is that their ability to capture different specificities on different motif positions is very limited. We can overcome this limitation by representing binding sites using a probabilistic model, which can accurately account for different specificities. Probabilistic approaches for representing motifs include the aforementioned profile representation, where for each position we keep the frequencies of each of the four nucleotides, and the PSSM representation, where for each nucleotide in each position we keep a weight which is the log of the ratio between the probability of the nucleotide in this position of the binding site and the background frequency. Under these representations, we cannot explicitly enumerate the search space. Thus, all methods resort to using heuristics that do not guarantee reporting the best motif. However, using such a model allows us to come up with motifs that better describe the underlying biological signal, compensating for the lack of guarantee for optimality. Therefore those methods are often used in real life biological sequence analysis.

Methods utilizing the PSSM representation differ by the optimization algorithm used to learn the motifs parameters, i.e. the PSSM weights, and the scoring function that is being optimized. Those methods include *MEME* [2] that uses the EM algorithm to optimize a *log-likelihood* score, *AlignACE* [23] that uses “Gibbs sampling”-like algorithm to optimize a *maximum a-posteriori probability (MAP)* score, and *CONSENSUS* [12, 30] that uses an information-theoretic driven scoring function. It builds a motif iteratively, first from an alignment of two putative binding sites then extending it greedily while trying to maintain a motif characterization that has high information content in each position. For a comprehensive discussion of binding sites representations and motif discovery methods the reader is referred to a survey by Stormo[29]

Of all the methods described above, many have been used to show that one can learn motifs that indeed match experimentally verified motifs, or that genes containing those motifs adhere to common biological characteristics that are not taken into account during the motif finding process, such as sharing common expression patterns. Hence motif finding algorithms may indeed provide us with biological knowledge otherwise difficult to acquire on a genomic scale, *in silico*.

Chapter 3

Discriminative Learning of Transcriptional Regulatory Elements

3.1 Algorithm Overview

Most of the current approaches search for binding sites of a single transcription factor that accounts for the co-regulation of the target gene set. This focus on single binding sites can be misleading. As previously discussed, groups of transcription factors often act in concert. This combinatorial nature of biological regulatory systems suggests that we need to devise computational tools for identifying such groups. Several recent approaches attempt to use knowledge of transcription factor binding sites to dissect combinatorial regulation. Pilpel *et al.* [21] developed methods that use gene expression patterns to evaluate combinations of transcription factors and identify pairs that work in concert. More recently, several approaches were suggested for discovering regions with multiple transcription factor binding sites, known as *cis-regulatory modules* (CRMs) [6, 11, 13, 25] Most of these recent works focus primarily on finding interactions between known transcription factor binding motifs, and do not allow for discovery of novel motifs within their framework.

In this dissertation we aim to discover *transcriptional regulatory complexes*, in cases where the binding sites of each member of the regulatory complex are not known in advance. We use the general *discriminative* approach that was recently put forth for the single motif model [3, 24, **?**]. In this case, we aim to find a transcriptional regulatory complex that best *discriminates* between a putative set of co-regulated promoters and a control group of genes. Such a discrimination focuses on motif combinations that make the distinction between these two groups, and not just on overabundance in the positive examples. This allows us to avoid genomic phenomena that are not specific to the group of genes we are interested in (e.g. low-complexity regions). Our approach is based on an explicit probabilistic model of both positive (regulated by the complex) and negative promoters (not regulated by the complex). Learning in this model is posed as optimizing its parameters with respect to the conditional likelihood of the promoter labels given their sequences. We develop a scheme for learning this model that uses a systematic search to find a rough initial guess for the motifs in the regulatory complex, and then refines these by performing an optimization of the model parameters with respect to the discriminative likelihood function (see Figure 3.1). As we show, this approach discovers motifs that are involved in the regulatory complex, and performs better than the strawman approaches that learn each motif in the regulatory complex separately.

We also describe an extension of our method that takes into consideration that a single transcription factor might be involved in several regulatory complexes. We can exploit this combinatorial nature of regulation mechanisms to learn a better model of both the regulatory complexes and their components. To illustrate this concept, suppose we have a set of genes that are regulated by a complex of transcription factors A and B, and another set that is regulated by a complex of A and C. Since A appears in both complexes, we can improve the model of A's binding preferences by performing *concurrent learning* of its binding site model from both target sets. In doing this, however, we need to take into account the binding sites of B and C. We describe a procedure that performs such concurrent learning of several factors given the target genes of regulatory complexes that involve different combinations of these factors. As we show, this approach performs better than learning each regulatory complex separately, while taking the same amount of time.

The rest of this chapter describes the algorithm in detail.

3.2 Discriminative Learning

One drawback of motif finding algorithms such as MEME [2] and AlignACE [23] is that they search for motifs that are overabundant within the regulatory regions of a group of genes. This approach might fail as those promoter regions often contain

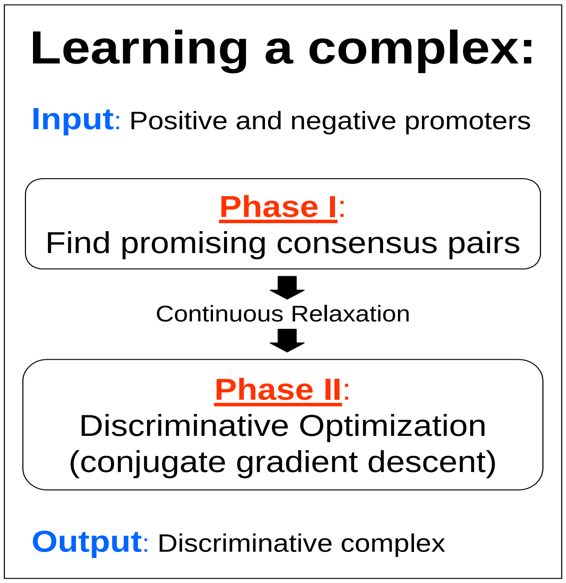

Learning a complex:
Input: Positive and negative promoters
Phase I:
Find promising consensus pairs
Continuous Relaxation
Phase II:
Discriminative Optimization
(conjugate gradient descent)
Output: Discriminative complex

Figure 3.1: **Algorithm overview:** The goal of our algorithm is to find a transcriptional regulatory complex that discriminates best between a putative set of co-regulated promoters and a control group of genes. In the first phase, we perform an exhaustive search for spatially adjacent consensus patterns that are enriched in the positive promoters compared to the control promoters. The best pair of patterns is transformed into an initial set of model parameters. In the second phase, optimize these parameters with respect to a discriminative likelihood function using conjugate gradient ascent.

inherent genomic phenomena that are not specific to the genes under study. We suggest to overcome this obstacle by taking a *discriminative* approach. In this approach we do not inspect only a group of promoter regions of co-regulated genes, but also take into account a control group of genes, with which we can control for phenomena that are not specific to the co-regulated group. Thus, the input to the algorithm is a set of genes that are considered "positive" (i.e., a cluster of co-regulated genes), and a "negative" set of control genes. We start by a systematic examination of all pairs of k-mers (short subsequences of k nucleotides) that appear in close proximity within promoter regions in our input, say up to 50bp apart, and search for pairs whose pattern of close-occurrences is significantly specific to the promoter regions of our input set of positive genes. We then use the most significant pairs as initial *seeds* for a procedure that learns a complex in which a

PSSM is used to represent each of the binding sites. Using the PSSM model in the second stage we capture a rich characterization of the binding sites, leading to better discrimination between “positive” and “negative” promoters.

A natural post-processing application of regulatory complexes that were learned discriminatively is to use them to predict exact locations of binding sites on the promoters as well as assigning posterior probabilities that a complex indeed occurs in a promoter sequence. This allows us to suggest putative genes that are regulated by the same complex, that were not part of the initial group of positive sequences. Although those applications are not specific to complexes learned discriminatively, they are very well suited for this task as their discriminative nature, that captures positive sequences while at the same time rejects the sequences of a control group, allows us to achieve a relatively low rate of false positive predictions.

3.3 Framework for Learning *cis*-Regulatory Complexes

We now describe the discriminative likelihood score and how to learn it. This is an extension of the score for a single discriminative motif described by Segal *et al.* [24].

3.3.1 Sequence Model

We start by describing a probabilistic model of promoter sequences. For this discussion we assume that our regulatory complex consists of two transcription factors, although this model can be easily extended to more elaborate complexes. We assume that each example in the input consists of a promoter sequence $\mathbf{S} = \langle S_1, \ldots, S_n \rangle$ and a label R. The label is either ’$+$’ if the promoter is regulated by the complex, or ’$-$’ if it is not.

In the case that $R =$ ’+’, we assume that the promoter contains binding sites of each of the complex components. Moreover, these sites are in close proximity to each other. The nucleotides that appear at these sites should match the motifs of their respective factor. We describe each motif using a PSSM and denote by $\psi_{k,j}(C)$ the probability distribution over different characters at the j^{th} position in the k^{th} PSSM. If we know these distributions, we can assign a probability to nucleotides in the binding sites of the promoters. All nucleotides that appear in the

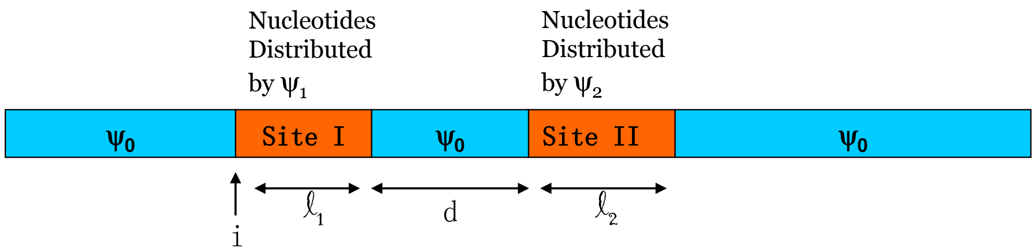


Figure 3.2: **Model of a regulated sequence:** A regulated sequence is composed of two binding sites distributed according to the two PSSMs of the complex (ψ_1 and ψ_2), while the rest of the sequence's nucleotides are distributed according to some background distribution ψ_0. We denote the start position of the first PSSM by i and the distance between the end of the first PSSM and the beginning of the second (the spacer) by d. The lengths of the first and second PSSMs are denoted by l_1 and l_2 respectively.

spaces between the binding sites are assumed to be drawn from the background distribution ψ_0. Thus, if the position of the first component is i, and the distance to the second component is d, then

$$P(\mathbf{S} \mid R=+,i,d,\mathbf{\Theta}) = \left(\prod_{\ell}\psi_0[S_\ell]\right)\left(\prod_{j_1=1}^{l_1}\frac{\psi_{1,j_1}[S_{i+j_1-1}]}{\psi_0[S_{i+j_1-1}]}\right)\left(\prod_{j_2=1}^{l_2}\frac{\psi_{2,j_2}[S_{i+l_1+d+j_2}]}{\psi_0[S_{i+l_1+d+j_2}]}\right)$$

Where l_1 and l_2 are the lengths of the PSSMs of the complex, and $\mathbf{\Theta}$ denotes the set of parameters. This description assumes we know the positions of the binding sites in the promoter. When they are unknown, we enumerate over their possible values and get

$$P(\mathbf{S} \mid R=+,\mathbf{\Theta}) = \sum_i \sum_{d\in\mathbf{D}} P_{site}(i)P_{dist}(d \mid i)P(\mathbf{S} \mid R=+,i,d,\mathbf{\Theta})$$

where $\mathbf{D}$ is the possible range of distances we consider, which can include negative distances (allowing for the binding site of the second PSSM to occur before the binding site of the first PSSM), and $P_{site}(i)$ and $P_{dist}(d)$ are prior distributions over the position of the first site and the distance between the two sites. In the rest of the manuscript, we assume that these two later distributions are uniform. An illustration of the regulated sequence model is presented in figure 3.2.

We now examine the case $R=-$. In this case, we assume that the promoter does not contain the complex binding sites. This can happen if either the promoter contains no binding sites of the complex components, or if it contains only one of the them. Again, we need to sum over these possible scenarios,

$$P(\mathbf{S} \mid R = -, \mathbf{\Theta}) = \prod_{\ell} \psi_0[S_\ell] \left[p_0 + p_1 \sum_i \prod_j \frac{\psi_{1,j}[S_{i+j-1}]}{\psi_0[S_{i+j-1}]} + p_2 \sum_i \prod_j \frac{\psi_{2,j}[S_{i+j-1}]}{\psi_0[S_{i+j-1}]} \right] \tag{3.1}$$

where p_0, p_1, p_2 are the probabilities of each of the scenarios.

3.3.2 Discriminative Likelihood

We are given labeled samples $D = \{(R^m, \mathbf{S}^m) : m = 1, \ldots, M\}$, where $\mathbf{S}^m = \langle S_1^m, \ldots, S_{n_m}^m \rangle$ is the m'th promoter sequence, and R^m is its label. In a discriminative setting, we aim to learn a model so to maximize the correct predictions of promoter labels. Formally, we aim to maximize the *log-likelihood* of the labels

$$\ell(\mathbf{\Theta} : D) = \sum_m \log P(R^m \mid \mathbf{S}^m, \mathbf{\Theta})$$

The question is how to relate this conditional probability to the sequence models we described above. Consider the term $P(R = + \mid \mathbf{S}, \mathbf{\Theta})$ for a particular sequence-label pair. Using Bayes rule and simple algebraic manipulations, we can write

$$\begin{aligned} P(R = + \mid S, \mathbf{\Theta}) &= \frac{P(R = +, S \mid \mathbf{\Theta})}{P(S \mid \mathbf{\Theta})} = \frac{1}{1 + \frac{P(R=-,S \mid \mathbf{\Theta})}{P(R=+,S \mid \mathbf{\Theta})}} \\ &= \frac{1}{1 + \frac{P(R=- \mid \mathbf{\Theta})}{P(R=+ \mid \mathbf{\Theta})} \frac{P(S \mid R=-,\mathbf{\Theta})}{P(S \mid R=+,\mathbf{\Theta})}} \\ &= \frac{1}{1 + e^{-\log \frac{P(R=+ \mid \mathbf{\Theta})}{P(R=- \mid \mathbf{\Theta})} - \log \frac{P(S \mid R=+,\mathbf{\Theta})}{P(S \mid R=-,\mathbf{\Theta})}}} \end{aligned}$$

And so we get

$$P(R = + \mid \mathbf{S}, \mathbf{\Theta}) = \mathit{logistic} \left(\log \frac{P(R = + \mid \mathbf{\Theta})}{P(R = - \mid \mathbf{\Theta})} + \log \frac{P(\mathbf{S} \mid R = +, \mathbf{\Theta})}{P(\mathbf{S} \mid R = -, \mathbf{\Theta})} \right), \tag{3.2}$$

where $\mathit{logistic}(x) = \frac{1}{1+e^{-x}}$. We see that the likelihood ratio $\frac{P(\mathbf{S} \mid R=+,\mathbf{\Theta})}{P(\mathbf{S} \mid R=-,\mathbf{\Theta})}$ determines the probability of the label. Thus, if the example is positive, we want to maximize likelihood ratio. And conversely, if the example is negative, we want to minimize it.

3.3.3 Optimizing the Score

Our aim is to learn the model parameters that maximize the log-likelihood score. The parameters of the described model are $P(R = + \mid \mathbf{\Theta}), P(R = - \mid \mathbf{\Theta}), \psi_{k,j}$, and ψ_0. However, note that in evaluating $P(R \mid \mathbf{S}, \mathbf{\Theta})$ in Eq. 3.2, the term $\prod_\ell \psi_0[S_\ell]$ cancels out. And thus, ψ_0 appears only in terms of the form $\frac{\psi_{k,j}[S]}{\psi_0[S]}$. We can exploit this by defining $w_{k,j}[S] = \log \frac{\psi_{ij}[S]}{\psi_0[S]}$. Similarly, we define another parameter $v = \log \frac{P(R=+|\mathbf{\Theta})}{P(R=-|\mathbf{\Theta})}$, that represents the log ratio between the priors of the two regulation models. The new parameter set is of smaller dimension. Moreover, the range of the new parameters is the whole real line. And so, we do not need to consider positivity constraints during optimization. We optimize the log-likelihood function by using a *conjugate gradient ascent* procedure. At each iteration, this procedure calculates the gradient of the log-likelihood function with respect to the parameters vector. It then updates the parameters in a direction that takes into account this gradient and previous steps, so that the step directions are conjugate. These iterations are continued until the procedure converges to a local optimum (Conjugate step directions allow for faster convergence). A detailed description of conjugate gradient methods can be found in [22]. The computationally intensive steps of this procedure are the computation of the likelihood function and its gradient (which is computed analytically). The computation of both likelihood and gradient requires time linear in the promoter length times the maximal distance allowed. A detailed derivation of the gradient is presented in appendix A.

3.4 Preliminary Seed Searching

The conjugate gradient ascent might be trapped in local maxima. Moreover, the maxima it converges to strongly depends on the initialization point (see figure 3.3).

Thus, it is crucial to start the optimization stage from a point that is in the general vicinity of the global maxima. For this we perform an initialization phase, in which we conduct a systematic pattern search that screens for good starting points (or *seeds*) for the optimization procedure. We search in the relatively simple space of patterns. The emphasis is on searching the whole space so to make sure we did not miss a potential pattern. However, patterns can be represented in several different ways. Choosing a more expressive representation may lead to unveiling

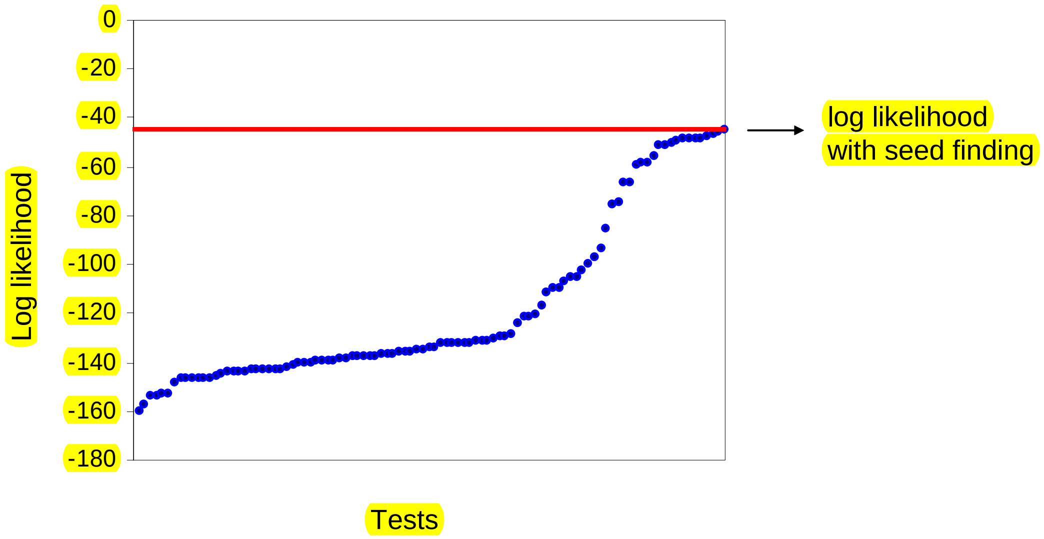


Figure 3.3: **Effect of preliminary seed finding:** 100 runs of the gradient ascent method started from different random points were conducted. The final log likelihood for each of those runs is presented (the x-axis is sorted ascendingly by the log-likelihood). The bar at the top of the figure is positioned at the log likelihood that was achieved when seed finding was used to initialize the gradient ascent process. As can be seen, the vast majority of random starting point runs are considerably worse than the run that was initialized with a preliminary seed finding procedure, while only a handful are of comparable performance.

better starting points, but this comes at a cost of larger search space, that we may not be able to explicitly search over. In typical real life data we search for two patterns each of size 5-7bp, which may be shorter than the real size of the binding sites but is not too expensive computationally. We thus have a search space of roughly $10^6 - 10^8$ pattern pairs even in the simplest representation of contiguous consensus patterns. Thus, we strike to find a balance between the expressiveness of the patterns we search for here and the their number. The drawback of searching for contiguous consensus patterns, is that they might miss binding sites that are not perfectly conserved. A regulatory complex that is present, with variations, in all of the promoters of a group of genes, might generate several patterns, each appearing only in a fraction of these promoters. Thus, each pattern that matches some of the occurrences of the complex will receive a non-significant score and our procedure will miss it. We therefore use an alternative approach, aiming at discovering patterns with wildcard positions, following the previously discussed *random projections* approach of Buhler and Tompa [8]. We randomly choose two masks with k mismatches, and scan the promoters for patterns with respect to those masks. We repeat this process many times with different random choices of masks to increase the probability of finding a complex if it exists in the data. Unlike Buhler and Tompa, we use the discriminative hyper-geometric score of Barash *et al.* [3] to evaluate patterns, rather than just finding overabundant ones. This score is appealing in our context as it takes into account not only the number of occurrences of a pattern in a target group of promoters, bust also the number of occurrences in a background control group. Thus it is well suited for seeding the continuous discriminative learning phase, together composing a fully discriminative framework. To get a starting point for the parameter optimization phase we take the most promising pair of motifs, occurring within some maximal distance (e.g. 200bp) and expand it into a probabilistic model by smoothing the distribution implied by the seeds, e.g. if a seed contains A in some position, we convert it to a distribution over all four nucleotides in which A is assigned a probability of 0.7 while the other three nucleotides are assigned a probability of 0.1. Such smoothing starts the search in the continuous space from a point the is close to the original seeds, but also allows it to explore other options as it does not assign zero or near-zero probability to subsequences that do not completely adhere to the seeds.

3.5 Pinpointing Binding Sites

When binding sites are characterized by patterns, it is straightforward to decide whether a given subsequence is a putative binding site or not. One may test for an exact match between the pattern and the subsequence, or allow a number of mismatches between them, based on the number of false positives that one is willing to accept. Providing a similar tool for assessing the probability that a pair of subsequences matches a learned regulatory complex is crucial for post-processing applications. Given a learned complex, we may use it for whole-genome scan to find novel binding sites for a complex, or to gain a level of confidence that certain positions along a promoter region indeed match the binding sites of the TFs characterized by a complex. What we strive for is a measure that given a complex and a pair of subsequences, denoted $x = x_1, ..., x_{l_1}$ and $y = y_1, ..., y_{l_2}$ (where l_1 and l_2 are the lengths of the two PSSMs in a complex), assigns a probability that x matches the first PSSM and y matches the second PSSM of the complex. Due to the nature of the PSSM model, we can easily calculate a score that is the sum of weights for x and y by the corresponding positions in the PSSMs. This score will be higher when x and y better match the complex.

Optimally we would have wanted to assign a p-value to each pair x, y of subsequences. This can be done by calculating the tail of the distribution for the corresponding sum of PSSM weights, and thus get the probability of seeing a pair of subsequences with such a score or higher by chance. This way we will be able to mark pairs that are unlikely to be observed randomly and are therefore likely to be occurrences of the complex. However, for a typical PSSM of 15 positions there will be as much as 4^{15} different scores and thus explicitly enumerating all possible subsequences of 15 nucleotides to calculate the p-value would be very time consuming, and impossible when complexes of two PSSMs are considered. A simple approximation to this distribution arises when observing that the score is a sum of typically $30 - 40$ random variables, and therefore is likely to be close to a a normal (gaussian) distribution, with the mean and variance defined by the weights of the PSSM. This is fairly easy to compute, however it does not always give satisfactory results.

We can compute a better approximation following the method of Bejerano [5] for exact p-value computation. Bejerano uses a branch and bound algorithm to compute the exact p-value, traversing along the paths of sequences that may exceed the score for which p-value is computed. This method becomes too expensive when dealing with scores of pairs of PSSMs. Nevertheless, using approximations along the branch and bound process we are able to achieve p-value computation that is both fast and indistinguishable from the exact p-value for practical purposes. To do that, if the score is the sum of weights of k positions, we iteratively construct a grid of scores and their corresponding approximate p-values, where the i^{th} grid is calculated using the $i - 1^{th}$ grid and the weights of the i^{th} position. The algorithm for pre-computing the grid is described in figure 3.4.

The ability to compute a p-value provides us with a way to assess the probability that a promoter is regulated by a given complex. We can do this in two ways, from which we can choose according to the problem we are confronting.

Best p-value. A biologically appealing approach is to say that a complex occurs in a promoter if the best p-value over all valid positionings of the complex on the promoter is lower than a threshold. The p-value must be subject to a Bonferroni correction, multiplying it by the number of valid positionings on the promoter which is roughly the product of its length and the maximal distance allowed between the two PSSMs of a complex.

Probability of a promoter to be regulated by a complex. Another approach is to calculate the posterior probability that a sequence S is regulated by a complex $\mathbf{\Theta}$ as defined above: $P(R = + \mid S, \mathbf{\Theta})$

The first approach (best p-value) is targeted at finding the actual binding sites of a complex, while the second (posterior probabilities) may be more appropriate for separating between regulated and non-regulated promoters. In the presence of a very strong signal (good match between the complex and a pair of putative binding sites) both measures will indicate that the promoter is regulated by the complex. However, the posterior probabilities approach, which may seem to be better justified theoretically, may consider several ”weak” occurrences of the complex as an evidence that the promoter is regulated, whereas the best p-value approach requires more of a clear-cut proof that certain positions in the sequence match the probabilistic model learned, and is therefore more convincing when the underlying biological processes are considered.

Computation of approximate p-values table

We are given a PSSM $pssm$ of n positions, and a background distribution P_{bg}.
We iteratively compute $pval\_table[i,s]$ - a lookup table for approximate p-values of scores s by the first i positions of $pssm$. The values s for which we compute the entries in $pval\_table$ are equally spread, in intervals of $precision$ (a pre-defined parameter).
To get the approximate p-value of an arbitrary score s we take the closest table entries below and above s, denoted s_1 and s_2, and compute it by a linear approximation:

```
$\underline{approx\_pval(i,s)}$
return $(pval\_table[i,s_2] - pval\_table[i,s_1])\frac{s-s_1}{s_2-s_1} + pval\_table[i,s_1]$
```

We compute $pval\_table[i,s]$ iteratively for all i's between $k+1$ and n, where k is some small constant, for which we can compute $pval\_table[k,s]$ quickly by enumerating over all possible k-mers and scoring them according to $pssm$.

```
$\underline{compute\_pval\_table(i)}$
$min\_score$ = the minimal score for the first $i$ positions of $pssm$
$max\_score$ = the maximal score for the first $i$ positions of $pssm$
for $s = min\_score;\ s \leq max\_score;\ s += precision$ do
  $pval\_table[i,s] = 0$
  // We sum over all possible ways to get score $s$ with $i$ positions,
  // using the approximation for $i-1$ positions and the $i^{th}$ position of $pssm$.
  for all $b \in \{A,C,G,T\}$ do
    $pval\_table[i,s] += P_{bg}[b] * approx\_pval(i-1, s-pssm[i,b])$
  end for
end for
```

After pre-computing $pval\_table[n,s]$ we can get an approximate p-value for an arbitrary score s immediately by calling $approx\_pval(n,s)$, saving the exponential time required for the exact computation.

Figure 3.4: Computation of approximate p-values table

Chapter 4

Results

4.1 Synthetic Data

To evaluate our approach we begin by applying it to synthetic data, which aims to maintain high degree of faithfulness to real life (noisy) datasets. Figure 4.1a shows the reconstruction of a regulatory complex composed of two yeast transcription factors from the TRANSFAC database [34]. Figure 4.1b shows how the learned complex was used to classify a similar unseen test set.

4.2 Methionine metabolism *cis*-Regulatory complex

van Helden *et al.* [32] studied a group of 11 yeast genes repressed by methionine. We applied our algorithm, taking those 11 genes as the positive group and the rest (~ 6000) yeast promoters as the negative group. Despite the small number of positive samples, the algorithm reported a regulatory complex with two PSSMs whose consensi match the motifs reported by van Helden *et al.* The reconstructed complex is shown in Figure 4.2. The learned complex was used to scan all yeast promoters for more putative occurrences. With a stringent p-value of 10^{-4}, only the original 11 genes as well as three novel putative targets were found to contain a significant hit. The three additional genes are HOM6 (YJR139C), MET17 (YLR303W), and ICY2 (YPL250C). The first two are known to be involved in methionine metabolism [9]. The third does not have a known function or biological process associated, and it may be conjectured as related to methionine metabolism according to our results. The results are summarized in 4.1.

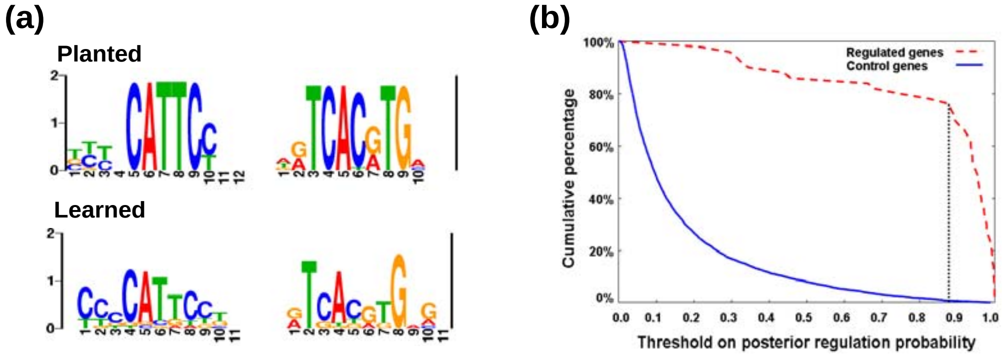


Figure 4.1: **Analysis of synthetic data** (a) A dataset of 5120 synthetic sequences of length 500bp was constructed stochastically using a 3^{rd}-order Markov background model and TRANSFAC's [34] `F$ABAA_01` and `F$CBF1_B` motifs (Top PSSMs). The dataset consists of 50 *true positive*, 20 *false positive*, 50 *false negative* and 5000 *true negative* sequences. The bottom complex was learned, correctly reconstructing both motifs. (b) Evaluation of learned complex on 5050 unlabeled test sequences. Using an *a-posteriori* probability of 0.88 as a threshold, 77% of the regulated sequences are found, along with about 1% of false positives.

4.3 Genome-wide Yeast Location Analysis

To evaluate our method on real life data we applied it to genome-wide *Chromatin Immunoprecipitation* (ChIP) location analyses [16, 26], which specify the target genes of 106 yeast transcription factors at various conditions. Such assays provide a way of inferring a group of putative co-regulated genes, by taking the intersection of two transcription factors' target genes (using a p-value of 0.01 as significance cutoff). For each such positive group, we also constructed the corresponding negative groups that consists of the rest of the promoter sequences. We then applied our algorithm to characterize what best differentiates the promoter sequences of the presumably co-regulated group, from those of the control group. For the evaluation of our method, we considered only pairs of factors under the same treatment (e.g. YPD), whose intersection size was > 20. This resulted in 143 groups. To demonstrate the utility of learning a regulatory complex, we compared our results with those obtained by a procedure that learns two separate PSSMs on the same datasets, using a single-PSSM version of our method [24]. After the first single PSSM was learned, its occurrences in all promoters were masked, and a second PSSM was learned. We compared both methods in terms of their sen-

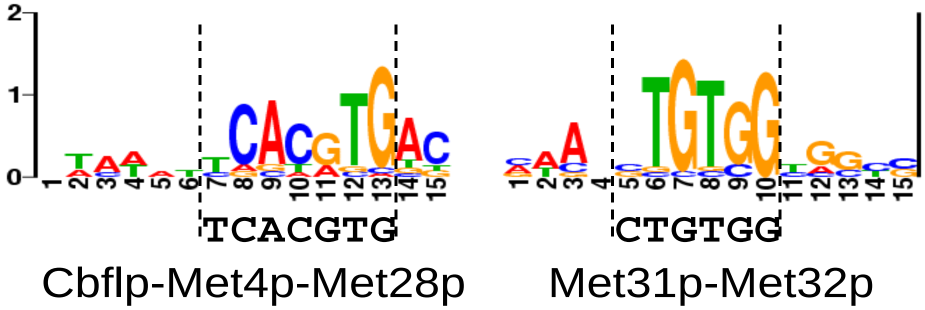


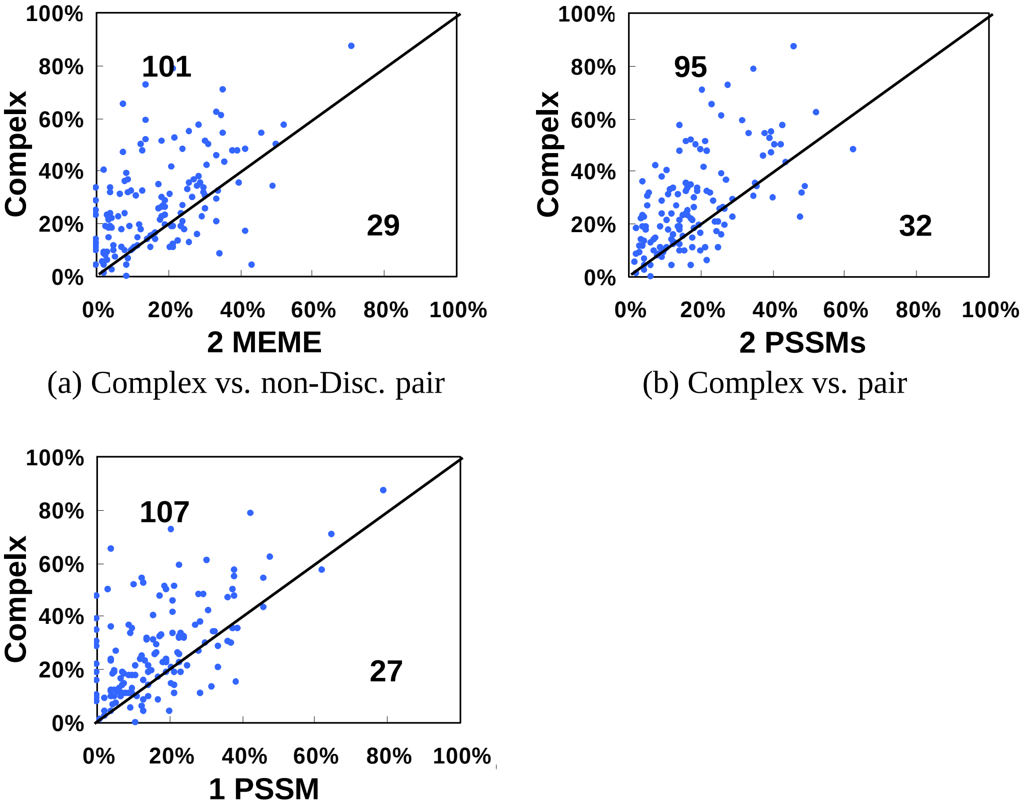
Figure 4.2: **Methionine metabolism *cis*-Regulatory Complex.** PSSMs were learned from 11 target genes from van Helden [32], and match the known consensi reported for Cbflp-Met4p-Met28p and Met31p-Met32p binding sites.

Table 4.1: p-values of genome-wide scan of yeast promoters with MET family regulatory complex (threshold of 10^{-4}).

Name	P-Value
MET14	1.11e-06
SAM2	2.19e-06
MUP3	3.69e-06
MET2	4.69e-06
MET3	5.18e-06
MET17	6.20e-06
MET1	6.20e-06
MET30	7.85e-06
MET6	9.03e-06
MET19	1.07e-05
MET25	1.42e-05
SAM1	2.31e-05
YPL250C	4.05e-05
HOM6	7.52e-05

sitivity and specificity on held-out test data, using a 5-fold cross validation test (following the protocol of [4]). The maximal distance allowed between PSSMs in a complex was set to 200bp. In each run, we measured the *true positive ratio*, when allowing for 1% *false positive rate*. In addition, we report the performance of the first PSSM learned, to illustrate the change in discrimination quality when learning a complex vs. a single PSSM. Finally, we added the best two PSSMs learned by MEME [2], using a 3^{rd}-order Markov model of yeast promoters to represent the background distribution of sequences. This was done to compare our results with the ones of a non-discriminative approach.

The analysis of all 143 pairs clearly shows that the learned complexes significantly outperform all other methods (on held-out test data). The comparison of true positives rates between those methods for a fixed cutoff of 1% false positives is shown in Figure 4.3. A detailed table of results is presented in appendix B.



(a) Complex vs. non-Disc. pair

(b) Complex vs. pair

(c) Complex vs. Single

Figure 4.3: **Comparison of the specificities of** 143 ***ChIP*** **datasets**. Each figure shows a comparison of two methods, where the coordinates of every point correspond to the true positives rate when allowing for 1% of false positives. The accuracy figures are evaluated using five-fold cross validation. (a) A comparison of our method (y-axis) vs. learning two PSSMs using MEME. Our discriminative procedure performs better on 101 of the datasets, equally as good on seven, and worse in 29 of the others. (b) and (c) show a comparison of learning a complex vs. learning two single discriminate PSSMs or learning one discriminate PSSM.

Chapter 5

Concurrent Learning of Several Factors

5.1 The Algorithm

Until now we focused on learning a specific regulatory complex for a particular set of target genes. In general, when analyzing large genomic datasets, we have several target sets, each regulated by a different combination of transcription factors. We now describe an algorithm that is aimed at exploiting combinatorial effects in learning complexes that involve several overlapping transcription factors. The idea is to assume that each transcription factor uses the same binding specificities, even when cooperating with different factors. This allows us to unite its binding site parameters into one single set.

As input, we assume we are given K training datasets (with positive and negative labels) for the different complexes. We also assume we know which transcription factor participates in each regulatory complex. As above, we can define the log-likelihood function of each training set as a function of the specificity of the transcription factors it is composed of. Now, however, we constrain each of the k complexes to share the parameters of their common motifs and try to optimize at once all K log-likelihood functions with k parameter sets. Thus if we have k training datasets, each regulated by two out of m binding sites, we learn the parameters of m PSSMs instead of the $2 \times k$ that we learn when considering each dataset separately. This leads to a dramatic decrease in the number of free parameters. For example, if we have 6 datasets that are regulated by pairs out of a set of 4 factors, we learn 4 PSSMs instead of 12. An illustration of this setup

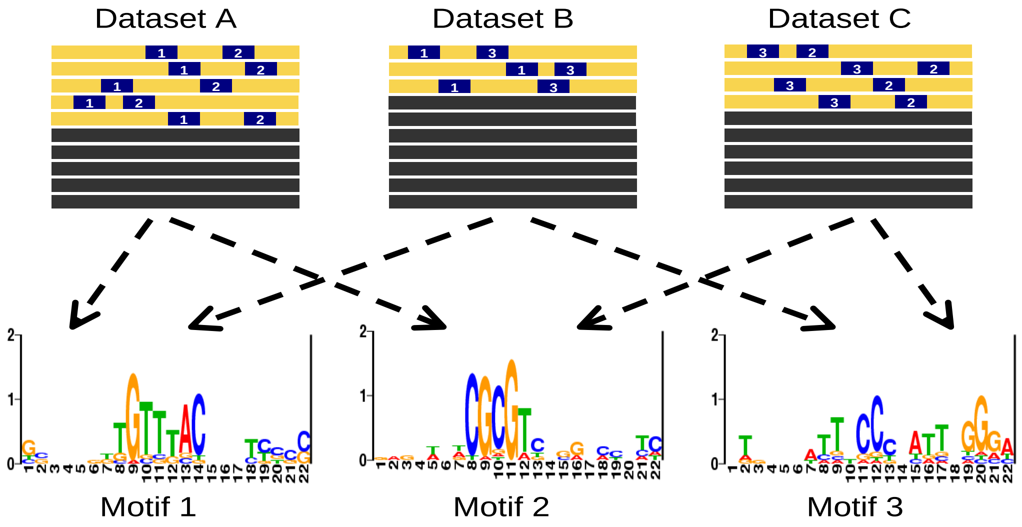


Figure 5.1: **Concurrent learning of regulatory complexes from several datasets**: Each dataset is composed of a co-regulated group of genes, and a background group. Instead of learning a pair on each dataset, we unify the different appearances of each transcription factor in one PSSM model, and optimize them on all relevant datasets.

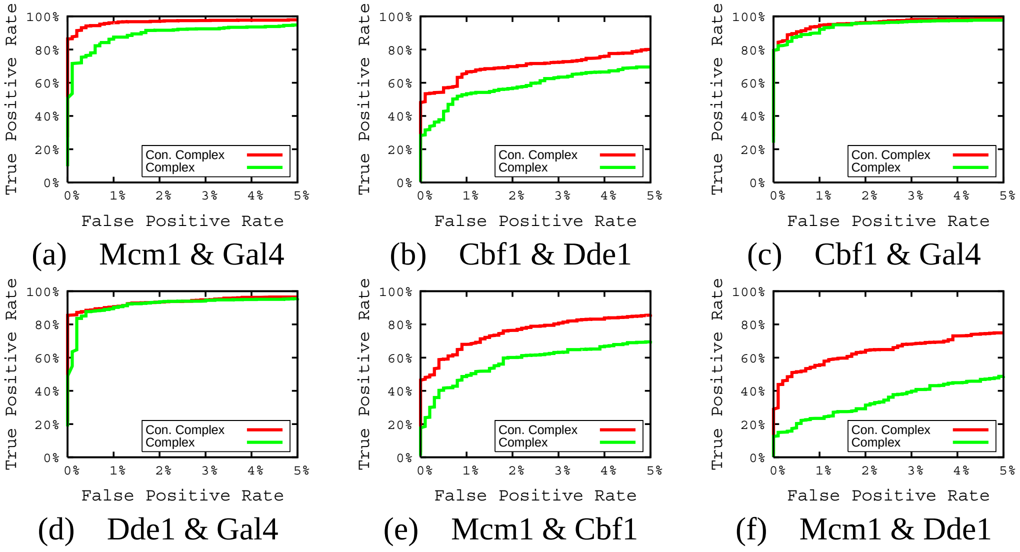
is presented in Figure 5.1. Formally, we define a new log-likelihood function for the combined dataset to be the sum of the K original log-likelihood function, and optimize it.

The intuition is that such a *concurrent training* integrates different evidence about each transcription factor. In addition, this learning results in a dramatic reduction in the number of parameters needed to optimize, while keeping the computational cost fixed. As we demonstrate, this approach leads to learning complexes that outperform the complexes learned for each pair of factors separately.

Seed Finding. An important aspect of learning more than two factors concurrently is that we can no longer enumerate over all possible seeds, as the number of different seeds for k factors, each of size l is 4^{kl}. For real-life values of l this becomes infeasible even for $k = 3$. For example, for $l = 7$ and $k = 3$ the search space is of size $4^{21} \approx 4 \times 10^{12}$ which is too large. We therefore conduct an iterative heuristic search, were in each iteration we fix $k - 1$ of the patterns and search for the best k^{th} pattern. This procedure continues until convergence, and the resulting k patterns are used to initialize the learning procedure by the same method used for initializing a complex of a pair of PSSMs.

5.2 Evaluation

To evaluate this method we generated synthetic training data by taking sequences sampled from a 3^{rd}-order Markov model trained on yeast promoters and planting motifs sampled from PSSMs taken from the TRANSFAC database [34]. As the ground set of motifs we took four PSSMs of yeast motifs: Mcm1, Cbf1, Dde1 and Gal4. We created six datasets, one per pair of motifs. Each dataset consists of 300 background sequences and 50 positive sequences in which the corresponding motif pair was planted. We then learned a set of four PSSMs, and tested its performance on test sequences (other negative and positive sequences that were not part of the training set). Our strawman was the basic procedure that learned one complex of two PSSMs per each of the six training datasets. We compared the ability of those complexes to distinguish between positive and negative sequences with this of the corresponding pair taken from the set of four PSSMs learned concurrently. The results of this evaluation are shown in Figure 5.2.



(a) Mcm1 & Gal4 (b) Cbf1 & Dde1 (c) Cbf1 & Gal4

(d) Dde1 & Gal4 (e) Mcm1 & Cbf1 (f) Mcm1 & Dde1

Figure 5.2: **Concurrent learning**. ROC curves for six datasets, comparing the performance of the concurrently learned complexes to complexes learned separately on each dataset. To allow for a fair evaluation of the learning procedure, both methods were initialized with the same seeds.

Chapter 6

Discussion

This dissertation presents a method for learning combinatorial patterns of *cis*-regulatory complexes. We have shown how such complexes can be learned from raw sequence data, and how they can be used for genome-wide scanning for novel binding sites. We have also shown that learning such complexes directly outperforms learning each of the factors separately in terms of the ability to discriminate between regulated and control sequences. The motivation behind our method is discriminative. By taking this approach we can learn regulatory complexes that are better suited for the task of separating between regulated and control sequences. We demonstrate the strength of the discriminative approach by comparing to MEME, a well-established method that is based on a generative algorithm that focuses on positive sequences only. It is important to bear in mind, however, that a motif learned discriminatively is not necessarily the best description of the underlying binding sites. It does not aim to provide us with the distribution of positions withing the binding sites, but with what differentiates those sites from the rest of the promoter sequences of the organism. The discriminative procedure is extended by the concurrent learning approach to combine information from multiple training sets. This allows us to learn better motifs for transcription factors that take part in different complexes.

There are several ways to extend the methods described here. Clearly, application of discriminative models is crucial for dealing with more complex genomic structure, as in higher-order organisms. An interesting future direction for extending our method is to learn parameters that model a non-uniform distribution of spacer length between the two factors of a regulatory complex. As the distance between factors in a complex is in some cases conserved, incorporating this information into the learning procedure may lead to better characterization of regulatory complexes. Another path that would be interesting to explore, is to add information of conserved positions in promoter regions, e.g. comparative genomics information, to strengthen the signal of the true binding sites. Incorporating comparative genomics information into motif finding algorithms has yielded promising results lately [15], and we expect it to have a significant effect in the context of composite regulatory modules as well. Another source of information that may be used to strengthen the signal of binding sites is that of dependencies between different positions. Our approach, as the vast majority of other works in this field, has built upon a probabilistic model that assumes that different positions within binding sites are independent. A recent study by Barash *et al.* [4] demonstrates that incorporating such dependencies into the learning framework often leads to learning motifs that outperform those that do not take such dependencies into account. Finally, it would be interesting to use complexes learned on sequences of an organism to construct global maps of interactions between transcription factors for this organism. Such maps can be used to gain insights about groups of transcription factors that take part in the same biological process or that carry related functions, and can also be used to infer groups of datasets that may serve as candidates for the concurrent learning procedure.

Appendix A

Calculating the Likelihood's Gradient

For using the gradient ascent algorithm to maximize the discriminative log-likelihood scoring function, we need to calculate the its gradient, i.e. the derivative w.r.t. each of the parameters. Recall that the log-likelihood is

$$\ell(\mathbf{\Theta} : D) = \sum_m \log P(R^m \mid \mathbf{S}^m, \mathbf{\Theta}),$$

and thus we can compute the derivative for each sequence separately.

We have shown in section 3.3 that

$$P(R = + \mid S, \mathbf{\Theta}) = \mathit{logistic}\left(\log \frac{P(R = + \mid \mathbf{\Theta})}{P(R = - \mid \mathbf{\Theta})} + \log \frac{P(S \mid R = +, \mathbf{\Theta})}{P(S \mid R = -, \mathbf{\Theta})}\right)$$

and that

$$P(S \mid R = +, \mathbf{\Theta}) = \left(\prod_l \psi_0[S_l]\right) \frac{1}{n - l_1 - l_2} \sum_i \sum_{d \in D} \frac{1}{|D|} \left(\prod_{j=1}^{l_1} \frac{\psi_{1j}[S_{i+j-1}]}{\psi_0[S_{i+j-1}]}\right) \left(\prod_{j=1}^{l_2} \frac{\psi_{2j}[S_{i+d+j}]}{\psi_0[S_{i+d+j}]}\right)$$

and

$$P(S \mid R = -, \mathbf{\Theta}) = \left(\prod_l \psi_0[S_l]\right) \left[p_0 + \frac{p_1}{n - l_1} \sum_i \prod_{j=1}^{l_1} \frac{\psi_{1j}[S_{i+j-1}]}{\psi_0[S_{i+j-1}]} + \frac{p_2}{n - l_2} \sum_i \prod_{j=1}^{l_2} \frac{\psi_{2j}[S_{i+j-1}]}{\psi_0[S_{i+j-1}]}\right].$$

The parameters are defined by $v = \log \frac{P(R=1|\mathbf{\Theta})}{P(R=-1|\mathbf{\Theta})}$ and $w_{ij}[c] = \log \frac{\psi_{ij}[c]}{\psi_0[c]}$. Thus, we can write the posterior probability that a sequence is positive as

$$P(R = + \mid S, \mathbf{\Theta}) \quad = \quad \mathit{logistic}\left(\log \frac{P(R = + \mid \mathbf{\Theta})}{P(R = - \mid \mathbf{\Theta})} + \log \frac{P(S \mid R = +, \mathbf{\Theta})}{P(S \mid R = -, \mathbf{\Theta})}\right)$$

$$
\begin{aligned}
&= \mathit{logistic}[v + \log \sum_i \sum_{d\in D} \frac{1}{|D|} \left(\prod_{j=1}^{l_1} e^{w_{1j}[S_{i+j-1}]} \right) \left(\prod_{j=1}^{l_2} e^{w_{2j}[S_{i+d+j}]} \right) \\
&\quad - \log(n - l_1 - l_2) \\
&\quad - \log \left(p_0 + \frac{p_1}{n-l_1} \sum_i \prod_{j=1}^{l_1} e^{w_{1j}[S_{i+j-1}]} + \frac{p_2}{n-l_2} \sum_i \prod_{j=1}^{l_2} e^{w_{2j}[S_{i+j-1}]} \right)] \\
&= \mathit{logistic}[v + F(S) - \log(n - l_1 - l_2)],
\end{aligned}
$$

where

$$
\begin{aligned}
F(S) &= \log \sum_i \sum_{d\in D} \frac{1}{|D|} \left(\prod_{j=1}^{l_1} e^{w_{1j}[S_{i+j-1}]} \right) \left(\prod_{j=1}^{l_2} e^{w_{2j}[S_{i+d+j}]} \right) \\
&\quad - \log \left(p_0 + \frac{p_1}{n-l_1} \sum_i \prod_{j=1}^{l_1} e^{w_{1j}[S_{i+j-1}]} + \frac{p_2}{n-l_2} \sum_i \prod_{j=1}^{l_2} e^{w_{2j}[S_{i+j-1}]} \right)
\end{aligned}
$$

Note that $(\log \mathit{logistic}(x))' = 1 - \mathit{logistic}(x)$. Thus,

$$
\frac{\partial \log P(R = + \mid S, \mathbf{\Theta})}{\partial v} = 1 - P(R = + \mid S, \mathbf{\Theta}) = P(R = - \mid S, \mathbf{\Theta}).
$$

Similarly,

$$
\begin{aligned}
\frac{\partial \log P(R = + \mid S, \mathbf{\Theta})}{\partial w_{ij}[c]} &= \frac{\partial \log P(R = + \mid S, \mathbf{\Theta})}{\partial F[S]} \frac{\partial F[S]}{\partial w_{ij}[c]} \\
&= (1 - P(R = + \mid S, \mathbf{\Theta})) \frac{\partial F[S]}{\partial w_{ij}[c]} \\
&= P(R = - \mid S, \mathbf{\Theta}) \frac{\partial F[S]}{\partial w_{ij}[c]}.
\end{aligned}
$$

Denote

$$
G(S) = \sum_i \sum_{d\in D} \frac{1}{|D|} \left(\prod_{j=1}^{l_1} e^{w_{1j}[S_{i+j-1}]} \right) \left(\prod_{j=1}^{l_2} e^{w_{2j}[S_{i+d+j}]} \right),
$$

and

$$
H(S) = p_0 + \frac{p_1}{n-l_1} \sum_i \prod_{j=1}^{l_1} e^{w_{1j}[S_{i+j-1}]} + \frac{p_2}{n-l_2} \sum_i \prod_{j=1}^{l_2} e^{w_{2j}[S_{i+j-1}]}.
$$

Then, $F(S) = \log G(S) - \log H(S)$, and hence

$$\frac{\partial F[S]}{\partial w_{ij}[c]} = \frac{1}{G(S)}\frac{\partial G(S)}{\partial w_{ij}[c]} - \frac{1}{H(S)}\frac{\partial H(S)}{\partial w_{ij}[c]}.$$

The derivatives of G w.r.t $w_{ij}[c]$ are

$$\frac{\partial G(S)}{\partial w_{1j}[c]} = \sum_{i:S_{i+j-1}=c} \sum_{d\in D} \frac{1}{|D|} \left(\prod_{j=1}^{l_1} e^{w_{1j}[S_{i+j-1}]}\right) \left(\prod_{j=1}^{l_2} e^{w_{2j}[S_{i+d+j}]}\right)$$

and

$$\frac{\partial G(S)}{\partial w_{2j}[c]} = \sum_{i} \sum_{d\in D:S_{i+d+j}=c} \frac{1}{|D|} \left(\prod_{j=1}^{l_1} e^{w_{1j}[S_{i+j-1}]}\right) \left(\prod_{j=1}^{l_2} e^{w_{2j}[S_{i+d+j}]}\right),$$

and the derivatives of H are

$$\frac{\partial H(S)}{\partial w_{1j}[c]} = \frac{p_1}{n-l_1} \sum_{i:S_{i+j-1}=c} \prod_{j=1}^{l_1} e^{w_{1j}[S_{i+j-1}]}$$

and

$$\frac{\partial G(S)}{\partial w_{2j}[c]} = \frac{p_2}{n-l_2} \sum_{i:S_{i+j}=c} \prod_{j=1}^{l_2} e^{w_{2j}[S_{i+j}]}.$$

To calculate the derivatives of $logP(R = - \mid S, \mathbf{\Theta})$ we use the fact that

$$P(R = - \mid S, \mathbf{\Theta}) = 1 - P(R = + \mid S, \mathbf{\Theta}).$$

Appendix B

Genome-wide Yeast Location Analysis - Methods Comparison

Table B.1: Performance of various methods on all 143 datasets created from the location experiments of Lee *et al.* [16] by the procedures described in section 4.3. For each pair of factors (under the treatments of Lee *et al.*) we compare the performance of the following methods: Learning a complex, Learning two PSSMs separately, learning one PSSM, and learning two PSSMs using MEME. For each set of parameters learning we compare its true positives rate on unseen data (using 5-fold cross validation) when allowing for 1% of false positives. As can be seen, learning a discriminative complex performs better than other methods.

TF1(treatment)	TF2(treatment)	#Regulated	Complex (%TP)	2 PSSMs (%TP)	1 PSSM (%TP)	MEME (%TP)
ABF1(YPD)	CBF1(YPD)	29	**72**	27	20	13
ABF1(YPD)	REB1(YPD)	24	**41**	20	20	20
ABF1(YPD)	SWI6(YPD)	22	**59**	31	22	13
ACE2(YPD)	FKH2(YPD)	25	**36**	4	4	8
ACE2(YPD)	MBP1(YPD)	31	12	6	6	**25**
ACE2(YPD)	NDD1(YPD)	32	**50**	18	3	31
ACE2(YPD)	SKN7(YPD)	35	8	8	17	**34**
ACE2(YPD)	SWI4(YPD)	31	**35**	16	9	25
ACE2(YPD)	SWI5(YPD)	42	11	14	4	**21**
ARG80(YPD)	ARG81(YPD)	21	**19**	14	0	9
ARG80(YPD)	RTG3(YPD)	25	**24**	16	4	4
ASH1(14hr_But)	MSS11(14hr_But)	41	9	7	4	**9**

Continued on next page

TF1(treatment)	TF2(treatment)	#Regulated	Complex (%TP)	2 PSSMs (%TP)	1 PSSM (%TP)	MEME (%TP)
ASH1(14hr_But)	PHD1(14hr_But)	54	5	1	**9**	1
ASH1(14hr_But)	RLM1(14hr_But)	30	16	**20**	6	16
ASH1(14hr_But)	SOK2(14hr_But)	98	1	2	1	**2**
ASH1(YPD)	CIN5(YPD)	23	**60**	26	30	34
ASH1(YPD)	NRG1(YPD)	24	**54**	33	12	45
ASH1(YPD)	PHD1(YPD)	27	18	**18**	7	3
ASH1(YPD)	SOK2(YPD)	21	**61**	52	47	33
ASH1(YPD)	SWI4(YPD)	28	21	10	**25**	3
ASH1(YPD)	YAP6(YPD)	23	**52**	39	13	21
CIN5(YPD)	CUP9(YPD)	27	29	18	**37**	18
CIN5(YPD)	HSF1(YPD)	21	19	14	14	**23**
CIN5(YPD)	INO4(YPD)	21	23	9	19	**23**
CIN5(YPD)	NRG1(YPD)	44	**18**	2	4	4
CIN5(YPD)	PHD1(YPD)	45	**31**	11	15	6
CIN5(YPD)	RAP1(YPD)	28	**21**	14	14	17
CIN5(YPD)	SKN7(YPD)	26	19	**26**	15	19
CIN5(YPD)	SOK2(YPD)	37	**27**	13	5	18
CIN5(YPD)	SWI4(YPD)	28	**14**	3	7	0
CIN5(YPD)	YAP1(YPD)	25	**24**	12	12	8
CIN5(YPD)	YAP6(YPD)	80	11	8	8	**15**
CUP9(Cu2)	MAC1(Cu2)	62	32	19	24	**33**
CUP9(YPD)	NRG1(YPD)	36	30	5	**36**	11
CUP9(YPD)	ROX1(YPD)	31	12	**12**	9	0
CUP9(YPD)	SOK2(YPD)	28	10	25	**28**	10
CUP9(YPD)	YAP6(YPD)	40	**17**	5	10	12
DIG1(14hr_But)	STE12(14hr_But)	125	**11**	8	9	0
DIG1(90min_But)	STE12(90min_But)	79	**11**	3	3	5
DIG1(Alpha)	STE12(Alpha)	71	9	**15**	11	0
DIG1(YPD)	STE12(YPD)	46	4	**4**	2	2
FHL1(YPD)	GAL4(YPD)	30	30	**40**	30	26
FHL1(YPD)	GAT3(YPD)	75	48	**62**	29	41
FHL1(YPD)	PDR1(YPD)	39	43	43	**46**	35
FHL1(YPD)	RAP1(YPD)	91	**47**	39	36	7
FHL1(YPD)	RGM1(YPD)	44	22	**47**	22	29
FHL1(YPD)	SFP1(YPD)	34	**70**	20	64	35
FHL1(YPD)	YAP5(YPD)	61	34	49	32	**49**
FKH1(YPD)	FKH2(YPD)	46	**32**	21	17	13

Continued on next page

TF1(treatment)	TF2(treatment)	#Regulated	Complex (%TP)	2 PSSMs (%TP)	1 PSSM (%TP)	MEME (%TP)
FKH2(YPD)	MBP1(YPD)	41	19	**19**	4	12
FKH2(YPD)	MCM1(YPD)	37	10	21	21	**21**
FKH2(YPD)	NDD1(YPD)	73	10	17	8	**20**
FKH2(YPD)	RAP1(YPD)	30	**33**	16	23	30
FKH2(YPD)	SKN7(YPD)	31	**32**	16	22	9
FKH2(YPD)	SWI4(YPD)	47	0	6	**10**	8
FKH2(YPD)	SWI6(YPD)	33	**9**	3	0	3
FZF1(YPD)	GCR2(YPD)	28	**17**	14	10	7
FZF1(YPD)	SRD1(YPD)	27	**22**	3	0	3
FZF1(YPD)	STP1(YPD)	21	**28**	23	0	0
GAL4(Cu2)	HAA1(Cu2)	25	**48**	20	28	24
GAL4(YPD)	GAT3(YPD)	33	**51**	21	21	18
GAL4(YPD)	PDR1(YPD)	23	**47**	21	17	39
GAL4(YPD)	RAP1(YPD)	24	**45**	37	20	33
GAL4(YPD)	RGM1(YPD)	24	20	25	33	**33**
GAL4(YPD)	YAP5(YPD)	26	**42**	7	30	30
GAT3(YPD)	PDR1(YPD)	53	33	**35**	32	28
GAT3(YPD)	RAP1(YPD)	66	**78**	34	42	21
GAT3(YPD)	RGM1(YPD)	79	**32**	11	17	25
GAT3(YPD)	SFP1(YPD)	24	**87**	45	79	70
GAT3(YPD)	SMP1(YPD)	21	**47**	14	38	38
GAT3(YPD)	YAP5(YPD)	93	25	**26**	22	17
GCR2(YPD)	NRG1(YPD)	21	19	14	19	**23**
GCR2(YPD)	SRD1(YPD)	32	6	**21**	12	3
GRF10Pho2(Pi-)	PHO4(Pi-)	28	10	**10**	7	7
HAP4(YPD)	PDR1(YPD)	21	57	42	**61**	52
HIR1(YPD)	RCS1(YPD)	24	12	**12**	4	0
HSF1(YPD)	RAP1(YPD)	25	32	**48**	24	4
HSF1(YPD)	SWI4(YPD)	29	20	24	20	**24**
IME4(YPD)	NRG1(YPD)	21	**57**	14	38	28
IME4(YPD)	PDR1(YPD)	21	28	9	**33**	19
INO2(YPD)	INO4(YPD)	46	8	2	**13**	2
MBP1(YPD)	MCM1(YPD)	22	**36**	27	9	9
MBP1(YPD)	NDD1(YPD)	42	19	4	7	**21**
MBP1(YPD)	SKN7(YPD)	34	11	2	5	**11**
MBP1(YPD)	STB1(YPD)	26	**65**	23	3	7
MBP1(YPD)	SWI4(YPD)	79	**18**	8	5	2

Continued on next page

TF1(treatment)	TF2(treatment)	#Regulated	Complex (%TP)	2 PSSMs (%TP)	1 PSSM (%TP)	MEME (%TP)
MBP1(YPD)	SWI5(YPD)	26	**23**	15	3	19
MBP1(YPD)	SWI6(YPD)	84	**40**	10	15	2
MCM1(Alpha)	STE12(Alpha)	26	50	42	19	**50**
MCM1(YPD)	NDD1(YPD)	41	17	24	17	**41**
MCM1(YPD)	STE12(YPD)	23	**39**	26	0	8
MCM1(YPD)	SWI4(YPD)	24	29	29	16	**33**
MCM1(YPD)	SWI6(YPD)	27	**14**	7	7	3
MSN2(Acid)	MSN4(Acid)	50	10	**10**	4	8
MSN2(H2O2)	MSN4(H2O2)	90	8	**10**	2	2
MSN2(H2O2)	YAP1(H2O2)	34	14	17	**20**	11
MSN4(H2O2)	YAP1(H2O2)	50	4	12	**20**	0
MSS11(14hr_But)	SOK2(14hr_But)	29	10	10	0	**10**
MSS11(YPD)	SIG1(YPD)	30	10	**20**	6	10
NDD1(YPD)	SKN7(YPD)	45	17	11	8	**24**
NDD1(YPD)	SWI4(YPD)	48	4	6	4	**8**
NDD1(YPD)	SWI5(YPD)	28	14	7	**21**	14
NDD1(YPD)	SWI6(YPD)	24	**33**	12	20	4
NRG1(YPD)	PHD1(YPD)	30	**23**	16	13	3
NRG1(YPD)	ROX1(YPD)	31	16	**25**	12	16
NRG1(YPD)	SKN7(YPD)	29	**31**	13	13	20
NRG1(YPD)	SOK2(YPD)	42	26	**26**	16	19
NRG1(YPD)	SWI4(YPD)	22	**36**	27	27	27
NRG1(YPD)	YAP6(YPD)	49	14	12	14	**16**
PDR1(YPD)	RAP1(YPD)	37	**54**	37	45	35
PDR1(YPD)	RGM1(YPD)	43	**51**	16	18	30
PDR1(YPD)	SFP1(YPD)	29	**51**	17	10	13
PDR1(YPD)	SMP1(YPD)	50	**32**	6	14	30
PDR1(YPD)	SWI5(YPD)	23	4	17	13	**43**
PDR1(YPD)	YAP5(YPD)	63	25	25	15	**30**
PHD1(14hr_But)	SOK2(14hr_But)	86	2	4	2	**4**
PHD1(YPD)	ROX1(YPD)	23	**34**	17	0	17
PHD1(YPD)	SKN7(YPD)	32	15	12	0	**28**
PHD1(YPD)	SOK2(YPD)	31	22	**29**	19	6
PHD1(YPD)	SWI4(YPD)	42	9	14	**14**	4
PHD1(YPD)	YAP6(YPD)	37	8	**8**	0	2
RAP1(YPD)	RGM1(YPD)	48	35	35	37	**39**
RAP1(YPD)	SMP1(YPD)	21	**38**	9	28	28

Continued on next page

TF1(treatment)	TF2(treatment)	#Regulated	Complex	2 PSSMs	1 PSSM	MEME
			(%TP)	(%TP)	(%TP)	(%TP)
RAP1(YPD)	YAP5(YPD)	58	**55**	39	37	25
RGM1(YPD)	YAP5(YPD)	75	26	5	**28**	24
RLM1(14hr_But)	SOK2(14hr_But)	26	**23**	3	19	0
ROX1(YPD)	SOK2(YPD)	26	15	15	**38**	15
ROX1(YPD)	SWI4(YPD)	21	**33**	19	9	0
ROX1(YPD)	YAP6(YPD)	44	6	4	4	**9**
RTG1(YPD)	RTG3(YPD)	23	**47**	21	0	13
SKN7(YPD)	SWI4(YPD)	54	7	**9**	5	5
SKN7(YPD)	SWI5(YPD)	27	**25**	18	22	18
SKN7(YPD)	YAP6(YPD)	22	13	4	**31**	22
SKO1(YPD)	SOK2(YPD)	24	**25**	16	12	0
SMP1(YPD)	YAP5(YPD)	31	35	16	**38**	29
SOK2(14hr_But)	STE12(14hr_But)	22	**31**	22	22	9
SOK2(YPD)	YAP6(YPD)	43	18	13	**23**	20
STB1(YPD)	SWI4(YPD)	32	**50**	40	37	12
SWI4(YPD)	SWI5(YPD)	23	30	**34**	0	30
SWI4(YPD)	SWI6(YPD)	74	18	4	**21**	4
SWI4(YPD)	YAP6(YPD)	23	**21**	17	0	4
SWI5(YPD)	YAP5(YPD)	22	**22**	4	18	18

Bibliography

[1] MI Arnone and EH Davidson. The hardwiring of development: organization and function of genomic regulatory systems. *Development*, 124(10):1851–1864, 1997.

[2] T.L. Bailey and Elkan C. Fitting a mixture model by expectation maximization to discover motifs in biopolymers. In *Proc. Int. Conf. Intell. Syst. Mol. Biol.*, volume 2, pages 28–36. 1994.

[3] Y. Barash, G. Bejerano, and N. Friedman. A simple hyper-geometric approach for discovering putative transcription factor binding sites. In O. Gascuel and B. M. E. Moret, editors, *Algorithms in Bioinformatics: Proc. First International Workshop*, number 2149 in LNCS, pages 278–293. 2001.

[4] Y. Barash, G. Elidan, N. Friedman, and T. Kaplan. Modeling dependencies in Protein-DNA binding sites. *Proc. Seventh Inter. Conf. Res. in Comp. Mol. Bio. (RECOMB)*, 2003.

[5] J. Bejerano. Efficient exact p-value computation and applications to biosequence analysis. In *Proc. Seventh Inter. Conf. Res. in Comp. Mol. Bio. (RECOMB)*. 2003.

[6] BP Berman, Y Nibu, BD Pfeiffer, P Tomancak, SE Celniker, M Levine, GM Rubin, and MB Eisen. Exploiting transcription factor binding site clustering to identify *cis*-regulatory modules involved in pattern formation in the Drosophila genome. *Proc. Natl. Acad. Sci.*, 99:757–762, 2002.

[7] A. Brazma, I. Jonassen, J. Vilo, and E. Ukkonen. Predicting gene regulatory elements in silico on a genomic scale. *Genome Res.*, 8:1202–15, 1998.

[8] J. Buhler and M. Tompa. Finding motifs using random projections. In *RECOMB'01*. 2001.

[9] J. M. Cherry, C. Ball, K. Dolinski, S. Dwight, M. Harris, J. C. Matese, G. Sherlock, G. Binkley, H. Jin, S. Weng, and D. Botstein. Saccharomyces genome database. http://genome-www.stanford.edu/Saccharomyces/, 2001.

[10] R. Derynck and Y. E. Zhang. Smad-dependent and smad-independent pathways in tgf-beta family signalling. *Nature*, 425(6958):577–84, 2003.

[11] Martin C. Frith, John L. Spouge, Ulla Hansen, and Zhiping Weng. Statistical significance of clusters of motifs represented by position specific scoring matrices in nucleotide sequences. *Nucl. Acids. Res.*, 30(14):3214–3224, 2002.

[12] GZ Hertz, GW III Hartzell, and GD Stormo. Identification of consensus patterns in unaligned dna sequences known to be funtionally related. *Comput. Appl. Biosci.*, 6:81–92, 1990.

[13] C. E. Horak, N. M. Luscombe, J. Qian, P. Bertone, S. Piccirrillo, M. Gerstein, and M. Snyder. Complex transcriptional circuitry at the G1/S transition in *saccharomyces cerevisiae*. *Genes Dev.*, 16(23):3017–3033, 2002.

[14] J. D. Hughes, P. E. Estep, S. Tavazoie, and G. M. Church. Computational identification of *cis*-regulatory elements associated with groups of functional related genes in *saccharomyces cerevisiae*. *J. Molecular Biology*, 296:1205–1214, 2000.

[15] M. Kellis, N. Patterson, M. Endrizzi, B. Birren, and E. S. Lander. Sequencing and comparison of yeast species to identify genes and regulatory elements. *Nature*, 423(6937):241–54, 2003.

[16] TI Lee, NJ Rinaldi, F Robert, DT Odom, Z Bar-Joseph, GK Gerber, NM Hannett, CT Harbison, CM Thompson, I Simon, J Zeitlinger, EG Jennings, HL Murray, DB Gordon, B Ren, JJ Wyrick, JB Tagne, TL Volkert, E Fraenkel, DK Gifford, and RA Young. Transcriptional regulatory networks in *saccharomyces cerevisiae*. *Science*, 298:799–804, 2002.

[17] L Marsan and MF Sagot. Algorithms for extracting structured motifs using a suffix tree with an application to promoter and regulatory site consensus identification. *J. Comp. Bio.*, 7:345–62, 2000.

[18] Lee Ann McCue, William Thompson, C. Steven Carmack, Michael P. Ryan, Jun S. Liu, Victoria Derbyshire, and Charles E. Lawrence. Phylogenetic

footprinting of transcription factor binding sites in proteobacterial genomes. *Nucl. Acids. Res.*, 29(3):774–782, 2001.

[19] AM McGuire, JD Hughes, and GM Church. Conservation of DNA regulatory motifs and discovery of new motifs in microbial genomes. *Genome Res.*, 10(6):744–757, 2000.

[20] P.A. Pevzner and S.H. Sze. Combinatorial approaches to finding subtle signals in dna sequences. In *Proc. Int. Conf. Intell. Syst. Mol. Biol.*, volume 8, pages 269–78. 2000.

[21] Y. Pilpel, P. Sudarsanam, and G.M. Church. Identifying regulatory networks by combinatorial analysis of promoter elements. *Nature Genetics*, 29:153–9, 2001.

[22] William H. Price. *Numerical Recipes in C*. Cambridge University Press, Cambridge, 1992.

[23] F.P. Roth, P.W. Hughes, J.D. Estep, and G.M. Church. Finding DNA regulatory motifs within unaligned noncoding sequences clustered by whole-genome mRNA quantitation. *Nat. Biotechnol.*, 16:939–945, 1998.

[24] E. Segal, Y. Barash, I. Simon, N. Friedman, and D. Koller. From promoter sequence to expression: A probabilistic framework. In *RECOMB'02*. 2002.

[25] R. Sharan, I. Ovcharenko, A. Ben-Hur, and R. M. Karp. Creme: a framework for identifying cis-regulatory modules in human-mouse conserved segments. *Bioinformatics*, 19(Suppl 1):I283–I291., 2003.

[26] I. Simon, J. Barnett, N. Hannett, C.T. Harbison, N.J. Rinaldi, T.L. Volkert, J.J. Wyrick, J. Zeitlinger, D.K. Gifford, T.S. Jaakkola, and R.A. Young. Serial regulation of transcriptional regulators in the yeast cell cycle. *Cell*, 106:697–708, 2001.

[27] S. Sinha and M. Tompa. A statistical method for finding transcription factor binding sites. In *Proc. Int. Conf. Intell. Syst. Mol. Biol.*, volume 8, pages 344–54. 2000.

[28] P. T. Spellman, G. Sherlock, M. Q. Zhang, V. R. Iyer, K. Anders, M. B. Eisen, P. O. Brown, D. Botstein, and B. Futcher. Comprehensive identification of cell cycle-regulated genes of the yeast saccharomyces cerevisiae by microarray hybridization. *Mol. Biol. Cell*, 9(12):3273–97, 1998.

[29] G Stormo. DNA binding sites: representation and discovery. *Bioinformatics*, 16:16–23, 2000.

[30] GD Stormo and GW III Hartzell. Identifying protein-binding sited from unaligned dna fragments. *PNAS*, 86:1183–1187, 1989.

[31] S. Tavazoie, J. D. Hughes, M. J. Campbell, R. J. Cho, and G. M. Church. Systematic determination of genetic network architecture. *Nat Genet*, 22(3):281–5, 1999.

[32] J. van Helden, B. Andre, 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):827–42, 1998.

[33] R.J. White. *Gene Transcripton: Mechanisms and Control*. Blackwell, 2001.

[34] E. Wingender, X. Chen, Fricke E., R. Geffers, R. Hehl, I. Liebich, M. Krull, V. Matys, H. Michael, R. Ohnhauser, M. Pruss, F. Schacherer, S. Thiele, and S. Urbach. The TRANSFAC system on gene expression regulation. *Nuc. Acids Res.*, 29:281–283, 2001.