PROBABILISTIC MOTIF SEARCHING

Thesis for the M.S.c. Degree

Submitted to the Scientific Council of
The Weizmann Institute of Science
Rehovot 76100, Israel

By

Libi Hertzberg

Carried Out Under the Supervision of
Professor Eytan Domany
November 3, 2003

Acknowledgements

I would like to express my gratitude to Prof. Eytan Domany, whose expertise, patience and support were crucial to the success of this work. His profound understanding guided me safely through my first encounter with scientific research. I greatly enjoyed my year working in the lab due to the combination of a friendly surrounding and close scientific collaboration. Eytan's personality, his authority and sense of humor give the lab. its special atmosphere.

A special thanks goes to Or Zuk, for many insightful ideas during our collaboration, and for his friendship.

Dr. Gaddy Getz contributed greatly through his ingenuity, creativity, knowledge and enthusiasm to share them.

Prof. Ido Kanter's original thought was a constant source of inspiration.

Another special thanks goes to my good friends and colleagues Michal Mashiach, Lena Gorelick and Tamar Barzuza for their encouragement and many shared cups of coffee.

To all my friends at the lab., many thanks for the friendly and stimulating environment.

Finally, I would like to express my deepest gratitude for the constant support, understanding and love that I received from Yossi Mossel, my beloved boyfriend.

ABSTRACT

A central issue in molecular biology is understanding the regulatory mechanisms that control gene expression. The availability of whole genome sequences opens the way for computational methods to search for the key elements in transcription regulation. These include methods for discovering the binding sites of DNA-binding proteins, such as transcription factors. A common representation of transcription factor binding sites is a *position specific score matrix* (PSSM). We developed a probabilistic approach for searching putative binding sites. Given a promoter sequence and a PSSM we scan the promoter and find the position with the maximal score. Then we calculate the probability to get such a maximal score or higher on a random promoter. This is the p-value of the putative binding site. In this way we searched for putative binding sites in the upstream sequences of Saccharomyces Cerevisiae, where some binding sites are known (according to the Saccharomyces Cerevisiae Promoters Database, SCPD, Zhu and Zhang [20]). For each gene we found its statistically significant putative binding sites. We measured false negatives rate and false positives rate of the putative binding sites we found, by a comparison to the known binding sites. Then we compared our results to MatInspector's results. MatInspector (Quandt et al. 14) is a software that looks for putative binding sites in DNA sequences according to PSSMs. Our results were significantly better. In contrast with us, MatInspector doesn't calculate the exact statistical significance of its results.

Contents

Chapter 1

Introduction

1.1 Biological Background

1.1.1 Overview

In all living organisms, the genetic material is DNA. The DNA is replicated each time a cell divides, so that each daughter cell receives a copy of the DNA. The DNA contains the genes of an organism, which are used as templates for manufacturing RNA. The RNA can then be used as instructions for synthesizing proteins.

1.1.2 Proteins

A protein is a linear polymer of amino acids linked together by peptide bonds. Proteins make up much of our bodies. Some form the structural parts of our cells, while others catalyze biochemical reactions. The structure of a protein is mainly determined by its sequence of amino acids, as well as by its environment, association with other proteins, and chemical modifications.

1.1.3 DNA

DNA is a polymer of nucleotides. Each nucleotide contains one of four bases: adenine (A), guanine (G), thymine (T), and cytosine (C) (see figure 1.1 (A)). These bases are the ”code” of the DNA. The structure of DNA is of a double helix (see figure 1.2 (B)). Each helix is a chain of nucleotides held together by phospho-diester bonds. The two helices are held together by hydrogen bonds. Each base pair consists of one purine base (A or G), and one pyrimidine base (C or T), paired according the following rule: G $\equiv$ C,

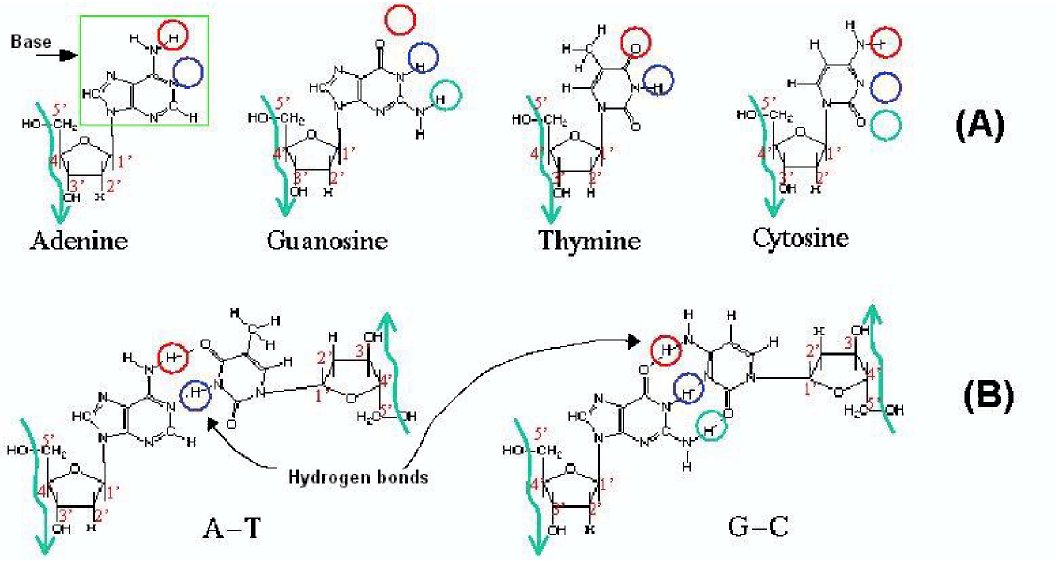


Figure 1.1: (**A**) The structure of the four nucleotides bases. Each nucleotide consists of a phosphate group, a sugar group, and one of the four bases adenine, cytosine, guanine, and thymine. (**B**) A is paired with T by 2 hydrogen bonds, G is paired with C by 3 hydrogen bonds. The molecules noted by numbers ($1' - 5'$) are carbons.

A = T (each '-' symbolizes a hydrogen bond) (see figure 1.1 (B)). The DNA molecule is directional, due to the asymmetrical structure of the sugars which constitute the skeleton of the molecule. Each sugar is connected to the strand upstream (i.e. preceding it in the chain) in its fifth carbon and to the strand downstream (i.e. following it in the chain) in its third carbon. Therefore, in biological jargon, the DNA strand goes from $5'$ (read five prime) to $3'$ (read three prime). The directions of the two complementary DNA strands are reversed to one another. Knowing the sequence of one strand allows one to infer the sequence of the other using the reverse complement, see figure 1.2 (A). When the structure of DNA was solved by Watson and Crick in 1953, it suggested an obvious mechanism for DNA replication (which turned out to be true): one strand is used as a template to synthesize the other strand.

In the cell, DNA is packaged into chromosomes with associated chromosomal proteins. The total genetic information stored in the chromosomes of an organism is said to constitute its genome. With few exceptions, every cell of a Eukaryotic multi-cellular organism contains a complete set of the genome, while the difference in functionality of cells from different tissues is the consequence of the variable expression of the corresponding genes.

The amount of DNA varies between different organisms. In humans, the genome is divided into 46 chromosomes: 22 pairs of homologous chromosomes plus the two sex

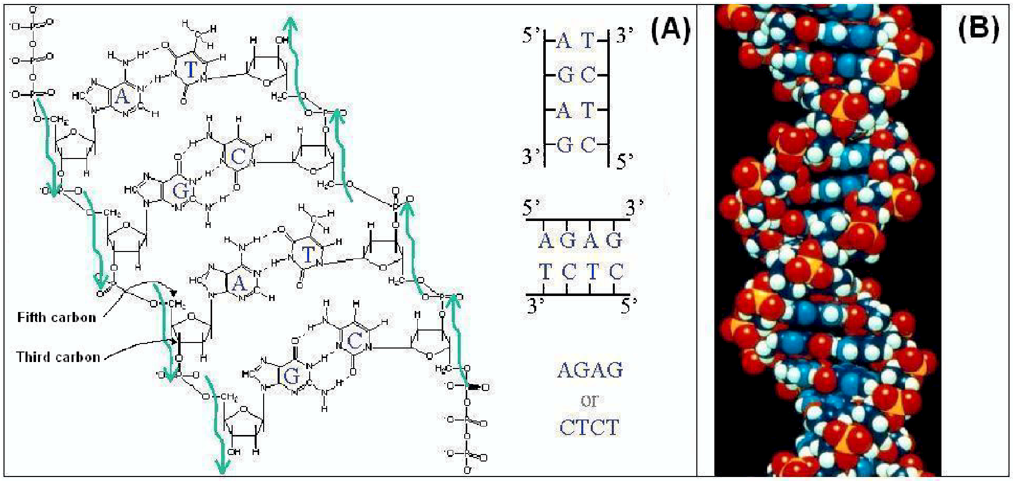


Figure 1.2: (**A**) The DNA molecule is directional, due to the asymmetrical structure of the sugars which constitute the skeleton of the molecule. Each sugar is connected to the strand upstream in its fifth carbon and to the strand downstream in its third carbon. Knowing the sequence of one strand allows one to infer the sequence of the other using the reverse complement. For example, the reverse complement of AGAG is CTCT. (**B**) In three dimensions, the two strands of DNA are wound around each other in a double helix structure.

chromosomes (XX or YY). A gene is a region of DNA that controls a discrete hereditary characteristic, usually corresponding to a single mRNA molecule carrying the information for constructing a protein. Pairs of homologous chromosomes are nearly identical to each other - they contain the same genes, but the sequence of the genes may be slightly different because one homolog comes from the mother and one comes from the father.

1.1.4 mRNA, Transcription and Translation

The expression of the genetic information stored in DNA involves the translation of a linear sequence of nucleotides into a linear sequence of amino acids in proteins. The flow is: DNA $\overset{transcription}{\rightarrow}$ mRNA $\overset{translation}{\rightarrow}$ Protein (see figure 1.3).

A segment of DNA is first copied into a complementary strand of mRNA (messenger RNA). RNA is similar in structure to DNA. It also has the bases A,G and C, but instead of T it uses the base U (uracil). The process of creating mRNA out of a DNA template is called transcription. It is catalyzed by the enzyme RNA polymerase (see figure 1.4) .

Near most of the genes there is a special pattern in the DNA called promoter, located upstream of the transcription start site. The promoter region informs the RNA polymerase where to begin the transcription.

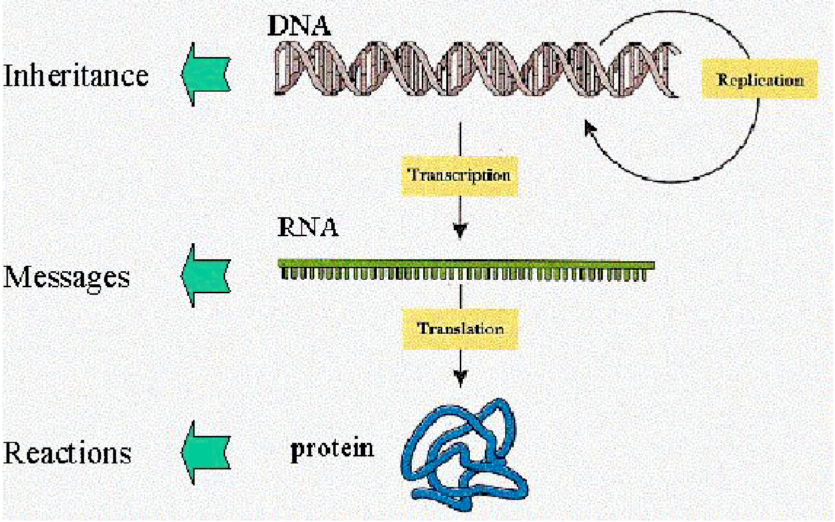


Figure 1.3: This is the central dogma: the DNA is transcribed into an RNA molecule, and the RNA is translated into a protein.

Transcription Factors

The RNA polymerase binds to the promotor with the assistance of transcription factors that recognize specific sequences on the promoter and bind to them (they are also called regulator proteins). Transcription factors are proteins which recognize specific DNA sequences by ”feeling” the chemical properties of the bases (see figure 1.5 (B)). By binding to the promoter region the transcription factors regulate gene expression. They can initiate transcription by recruiting the RNA polymerase and enabling it to start the process. Some of the transcription factors are repressors, i.e. they bind to the promoter sequence in a way that prevents the RNA polymerase from starting transcription. The transcription factors bind to specific short ($5-30$ bps) DNA sequences (motifs). Each transcription factor has its own specific motifs to which it can bind (see figure 1.5 (A)).

A common representation of transcription factor binding sites is a position weight matrix (PWM). A PWM is built out of all the known motifs to which the transcription factor binds, and it counts the number of appearances of every nucleotide in every position of the motif (see table 1.1).

Thus, the motivation of searching for transcription factors motifs in DNA sequences is clear - it can reveal the mechanism of gene regulation.

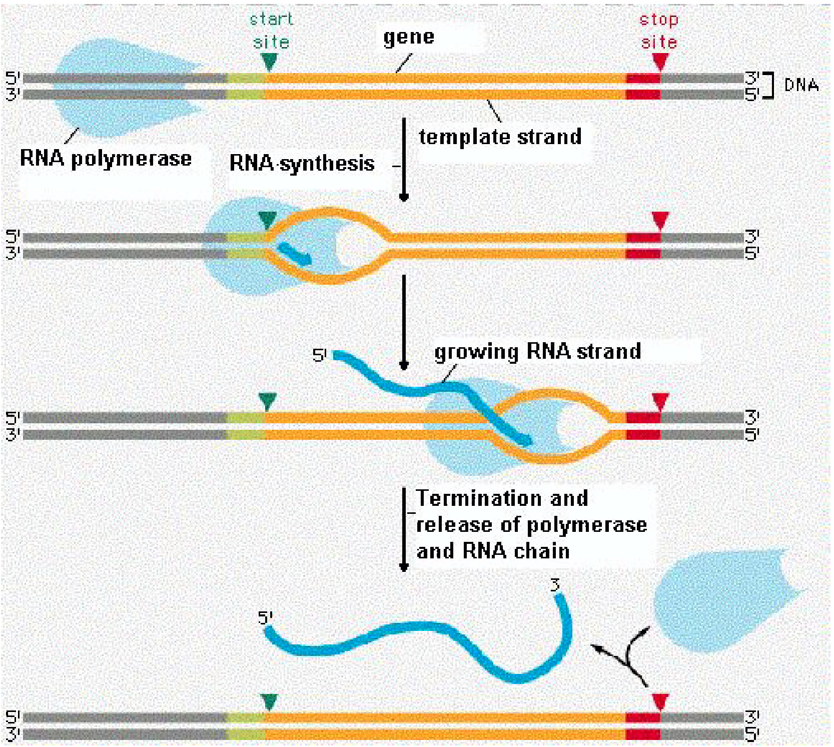


Figure 1.4: Transcription: The RNA polymerase binds to the DNA. Then the DNA double helix is opened, and the RNA polymerase starts to transcribe the template strand from the start site. An RNA chain is growing until the termination in the stop site. Then the RNA polymerase and the new RNA chain are released.

PWM

	Position					
	1	**2**	**3**	**4**	**5**	**6**
A	14	0	0	15	0	5
C	0	0	22	1	12	6
G	8	0	0	4	0	4
T	0	22	0	2	10	6

Table 1.1: An example of a PWM created by 22 sequences. For instance, in position 1 the letter A appears in 14 of the 22 known binding sites, G appears in 8, and C and T don't appear in position 1 in any of the known binding sites.

Translation

The mRNA molecule is then translated into a protein. To do this, the cell machinery uses the sequence of the mRNA to decide the sequence of amino acids of the protein. Starting with the first AUG of the RNA, the cell reads off triplets of bases which specify which amino acid to add to the growing chain of amino acids. Each triplet is called a codon, and the code that translates each codon to an amino acid is called the genetic code. The genetic code is identical in almost all organisms.

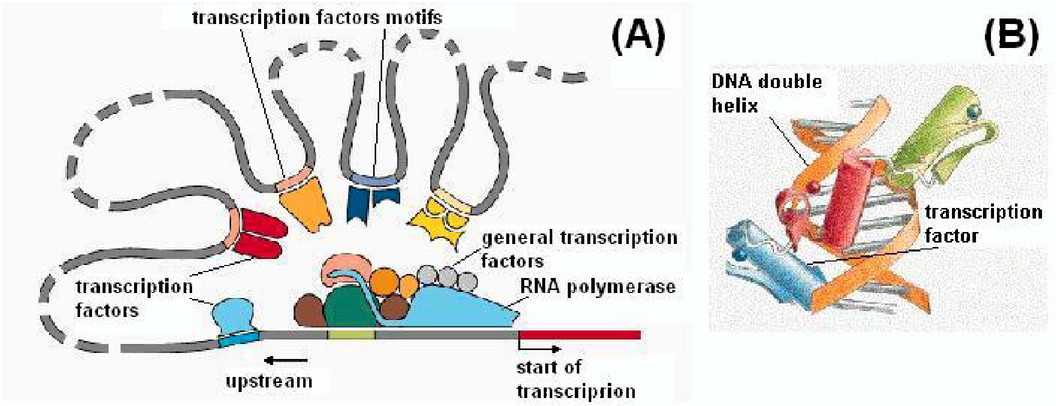


Figure 1.5: (**A**) Transcription factors are bound to the promoter upstream to transcription start site. Each transcription factor binds to its specific motif on the promoter. The transcription factors recruits the RNA polymerase, which then binds to the promoter next to the transcription start site. (**B**) A transcription factor recognizes a motif on the DNA by "feeling" the chemical properties of the bases.

1.2 Recent Works

Many aspects of transcription regulation involve transcription factors. As was described in 1.1.4, these factors modulate the expression of genes by binding to specific positions in genes' promoters. Identifying "binding sites", the locations to which these factors bind, remains a difficult problem in molecular biology. A central reason for this difficulty is that a single transcription factor might bind to regions which vary greatly in their sequence. Although the binding sites for a particular transcription factor share some common pattern, the pattern is not specific, and thus finding it is a difficult task.

One way to deal with this problem is through biological experiments, which are often costly and time consuming. The recent availability of complete genomic sequences (including intergenic regions) motivates attempts to understand the regulatory mechanisms through computational analysis. Algorithms and tools for searching regulatory elements can be divided into two major classes: 1) methods that search for known transcription factor binding motifs. 2) methods that try to detect new consensus patterns within a set of DNA sequences. In the following sections some of the recent works of the two classes are presented.

1.2.1 Searching for Known Transcription Factor Motifs

As was explained in section 1.1.4, table 1.1, binding sites of transcription factors are commonly represented by a PWM (position weight matrix). Many of the methods that search for known transcription factor binding sites use PWMs to model the binding sites. We describe here MatInspector (Quandt et al. [14]) and PRIMA (Elkon et al. [6]); both search for known transcription factor binding sites represented by PWMs. Another example for a recent method which searches for known transcription factor binding sites is Toucan (Aerts et al. [1]).

MatInspector

MatInspector is a commercially available software that looks for putative binding sites in DNA sequences according to PWMs. It uses a large database of PWMs, and given a DNA sequence, it searches for their appearances in the sequence (or sequences). The user can also define PWMs with which the search will be done.

From every PWM a position specific score matrix (PSSM) is built. The calculation of the PSSM takes into account the information content in every position of the PWM (see Quandt et al. [14] for full details). For example, the PSSM calculated from the PWM in table 1.1 is shown in table 1.2.

PSSM

	Position					
	1	**2**	**3**	**4**	**5**	**6**
A	0.124	0	0	0.095	0	0.011
C	0	0	0.331	0.006	0.103	0.013
G	0.071	0	0	0.025	0	0.009
T	0	0.331	0	0.012	0.086	0.013

Table 1.2: A PSSM calculated from the PWM in table 1.1. All the scores are positive and the maximal overall score for a sequence is 1. Here the sequence with maximal score is 'ATCACC': score(ATCACC) = $(0.124 + 0.331 + 0.331 + 0.124 + 0.103 + 0.013)$=1. Positions with high information content can contribute more to the overall score. For example, in position 2 there is high information content - the letter T appears in all 22 binding sites. Thus, the letter T in position 2 gets the maximal score, 0.331. In position 6, however, there is low information content - all the letters appear in position 6 in the 22 known binding sites. Thus all the letters get a low score in this position $(0.009 - 0.013)$.

In addition to the PSSM, a score is given to the core region of the PWM. The core region is defined as the four consecutive positions with the highest information content.

Given a DNA sequence, MatInspector scans it and computes the score for every PWM (and for its core region) in every position along the DNA. The output is a list of matches of every PWM. A match is a position in the sequence in which the score of the PWM is higher than a given threshold, and also the score of the core region is higher than a given (different) threshold. The user can decide on these thresholds. MatInspector recommends to use its optimized threshold values which are computed by MatInspector ”in a way that a minimum number of matches is found in non-regulatory test sequences”. The details of this computation are not published.
A comparison between MatInspector and our method is described in 3.2.

PRIMA

PRIMA (PRomoter Integration in Microarray Analysis) is a program for searching for common regulators in a set of genes. It finds transcription factors whose binding sites are significantly overrepresented in a given set of promoters. PRIMA requires two data collections: (1) Human promoters. PRIMA uses a set of putative promoters for 12981 human genes (which is called the ’13K set’). (2) Models for binding sites recognized by transcription factors. PRIMA uses PWMs for modelling binding sites recognized by transcription factors.

PRIMA gets as input two sets of genes: a target set which is suspected to have common regulators, and a background set (e.g., the 13K set), and for each PWM P it performs the following steps: (1) Compute a similarity threshold T(P). Subsequences in the scanned promoters with similarity scores above this threshold are considered as matches of P (i.e., putative binding sites of the transcription factor modelled by the PWM). (2) Scan the promoters of the target and the background sets for identification of matches of P. (3) Employ a statistical test to examine whether hits of P are significantly over-represented in the target set with respect to the background set.

The full details of the algorithm and the relevant computational analysis are described in Elkon et al. [6]. A comparison between PRIMA and our method is described in 3.6.1.

1.2.2 Searching for New Consensus Patterns

We describe here AlignACE (Hughes et al. [8]) which uses a Gibbs sampling technique to detect overrepresented motifs in a set of DNA sequences. Thijs et al. [18] also uses a Gibbs sampling technique for this purpose, while MEME (Bailey and Elkan [2]) employs an EM technique.

AlignACE

AlignACE (Aligns Nucleic Acid Conserved Elements) is a program which finds sequence elements conserved in a set of DNA sequences. It uses a Gibbs sampling strategy which is similar to that described by Lawrence et al. [10].

We review here a simplified version of the Gibbs sampling strategy, as described by Lawrence et al. [10]:
Input: A group of promoter sequences and an estimation of the motif length, k.
Initiation: Choose several random positions on the promoter sequences to create initial motif alignment.
Iterations: (1) Score all positions in the promoter sequences according to the current motif alignment. (2) Pick a position with a probability proportional to its score. (3) Add the position to the alignment if it increases the alignment score.

The score used in Lawrence et al. [10] and also in AlignACE to measure the quality of the alignment is the MAP (maximum a priori log likelihood) score. A detailed development of the formula is given by Liu et al. [11]. A useful approximation is given by the formula $NlogR$, where N is the number of aligned sites and R is the degree of overrepresentation of the motif in the input sequence.

The algorithm of AlignACE differs from that of Lawrence et al. [10] in the following ways: (1) the motif model was changed so that the base frequencies of the genome model are taken into acount. (2) Both strands of the input sequence are simultaneously considered at each step of the algorithm. (3) Simultaneous multiple motif searching was replaced by an approach in which single motifs were found and iteratively masked.

Another program described in Hughes et al. [8] is ScanACE. ScanACE searches a genome for close matches to a motif found by AlignACE. ScanACE can be set to return all genomic sites scoring better than a cutoff based on the mean and standard deviation of the scores of the aligned sites, or it can return a given number of best sites.

For full details and application to genes in the Saccharomyces Cerevisiae Genome, see Hughes et al. [8].

1.3 Our Approach

We present here a method that searches for binding sites of known transcription factors, using PWMs to model their binding sequences. We calculate PSSMs out of the PWMs (according to log-likelihood ratio, see 2.1). Given a DNA sequence D and a PSSM M, we scan the sequence and find the maximal score on it according to M. Suppose that the value of this maximal score is τ. We then need to decide wether τ is high enough to enable the transcription factor to bind to this position, i.e. we need to decide on a threshold score value, $\overline{\tau}$, such that a position with a score $\tau > \overline{\tau}$ is a 'match', and a position with a score $\tau < \overline{\tau}$ is not. All methods that use PSSMs have to deal with this question. In Elkon et al. [6] this threshold value is determined in a way that the number of matches in a big background promoter set (of 12,981 promoters of length 1200 bp) is approximately 10% of the whole set. In MatInspector it is chosen in a way that the estimated mean of the number of matches in a random sequence of length 1000 is not too high.

We improve this by calculating the probability to get such a maximal score τ, or higher, in a random sequence of the same length (of the given sequence D). In some cases we calculate the exact probability, and in others we get a tight upper bound on it. This is an improvement in two aspects: 1) we get a reliable p-value. 2) we get more information on the score than just if it is higher or lower than the threshold score. This is important, since the biological reality of the binding of a transcription factor has more than two possibilities (bound and not bound). The transcription factor can bind to the binding site with a certain affinity. Thus, a higher score (and a lower p-value) might indicate a higher affinity of binding of the transcription factor to this promoter.

Chapter 2

Methods

2.1 Preliminaries

Let Σ be the alphabet of the four nucleotides of which DNA sequences are composed, i.e. $\Sigma \equiv \{A, C, G, T\}$, and S the size of Σ, $S = |\Sigma| = 4$. Let $p_1, \ldots, p_4$ be the frequencies of the four nucleotides. We use here a random model for DNA sequences with independent nucleotides (not necessarily uniform). We denote the random model by B.

Let R be a DNA sequence of length L ($R = r_1 \ldots r_L$, $r_i \in \Sigma$). Denote by F a given $S \times L$ position weight matrix (PWM) with $F_{r,i}$ denoting the number of appearances of nucleotide r in position i. We take the log likelihood ratio to be the score of R:

$$score(R) = log(\frac{\prod_{i=1}^{L} P(x_i = r_i/F)}{\prod_{i=1}^{L} P(x_i = r_i/B)}) = \sum_{i=1}^{L} log(\frac{P(x_i = r_i/F)}{P(x_i = r_i/B)}) \tag{2.1}$$

where

$$P(x_i = j/F) = \frac{F_{j,i}}{F_{1,i} + F_{2,i} + F_{3,i} + F_{4,i}}, \quad P(x_i = j/B) = p_j$$

We get that

$$score(R) = \sum_{i=1}^{L} M_{r_i,i}$$

Where $M_{j,i} = log(\frac{P(x_i=j/F)}{P(x_i=j/B)})$. M will be denoted as the position specific score matrix (PSSM). Our aim is to determine the distribution of scores of R on random promoter regions of length N.

2.2 A sketch of the Algorithm

Let $D = (d_1, \ldots d_N)$ be a promoter region of length N. We scan D and calculate $score(R)$ for $R = (d_i \ldots d_{i+L-1})$, $i = 1, \ldots, N - L + 1$. Let τ be the maximal score we found. We calculate the probability to get a maximal score T higher than τ, $P(T \geq \tau)$ in a random sequence of length N.

We divide this calculation into two parts:

1. Find the set of all sequences of length L, $A = A(\tau) \subset \Sigma^L$, which have a score higher than τ, $A = \{R = (r_1, \ldots, r_L), r_i \in \Sigma | score(R) \geq \tau\}$.
2. Let H be the number of occurrences of sequences from A in a DNA sequence of length N. We want to calculate the probability $P(H > 0)$ to observe at least one sequence from the set A in a random sequence of length N; clearly, $P(H > 0) = 1 - P(H = 0)$.

$P(H > 0)$ is the p-value produced by our algorithm, the probability of finding a higher score in a random sequence.

2.3 Finding A

Let K be the size of A, $K = |A|$. We first estimate K by a method taken from Staden [17] (see 2.3.2). This estimation gives very tight lower and upper bounds of K. If K is not too large, we enumerate all sequences in A by a branch and bound algorithm in time $O(K)$ (see 2.3.1). However, sometimes K is huge, and then enumerating K sequences will take too much time. In this case we use an upper bound on K.

2.3.1 Branch and Bound algorithm

Here we describe the Branch and Bound algorithm used to enumerate A, the set of all targets with score above the threshold. For this we need a new definition. Let

$R = (r_1, .., r_L)$ be a sequence. We define $R' = \bigcup_{i=1}^{L}(r_1, .., r_{i-1}, r'_i, r_{i+1}, .., r_L)$, where r'_i is the next best nucleotide to r_i in the $i - th$ position. Coping with ties inside columns of M is done by enumerating them in ascending indexing order. For this we need a more delicate definition given by :

$$r'_i = \begin{cases} r_i & ,M_{r_i,i} = \min\{M_{j,i} | j \in \Sigma\}, r_i = \max\{j | M_{r_i,i} = M_{j,i}\} \\ \arg\max_{j\in\Sigma,}\{M_{j,i} | \{M_{j,i} < M_{r_i,i}\} \cup & \\ \{M_{j,i} = M_{r_i,i}, j > r_i\}\} & \text{, otherwise.} \end{cases} \tag{2.2}$$

Notice that if there are multiple maxima, we always assume that arg max takes the smallest index among them. For a set of sequences B, we define $B' = \bigcup_{R\in B} R'$. We define also $B(\tau) = \bigcup_{R\in B, score(R)\geq\tau} R$ (the set of sequences with score above τ). The algorithm is as follows :

1. Start with the consensus sequence $A_0 = R^*$, by taking at each column the best nucleotide. $R^* = (r_1^*, \ldots, r_L^*)$ where r_i^* is defined by : $r_i^* = \arg\max_{j\in\Sigma} M_{j,i}$.
2. For i from 1 to $3L$ do : $A_i = A_{i-1} \bigcup A'_{i-1}(\tau)$
3. Output $A \equiv A_{3L}$

Note that the fact that we perform $3L$ steps guarantees that every one of the 4^L sequences might be reached ($3L$ is the maximal possible "distance" between the consensus sequence and the worst scoring sequence if we allow only operations of the kind $r_i \rightarrow r'_i$, and clearly this is the maximal such "distance" between any two sequences).

Actually, we usually do not have to do the for loop until the end. If at some step of the for loop we did not gain any new sequence (that is $A'_{i-1}(\tau) \subset A_{i-1}$), we can stop immediately.

Generating $A_{i-1}(\tau)'$ in step 2, is done by expanding each sequence $R \in A_{i-1}$ to at most L sequences which form R', and then checking for each sequence in R' whether its score is no less than τ, to get $R'(\tau)$. Since we enumerate only the K sequences with score above the threshold τ, and each one can be expanded to at most L sequences, the overall time complexity is $O(KL)$. Note that at each iteration of the for loop in step 2, we need to do "unique", since some sequences might appear more than once. This can be done, however, in linear time, using hash tables.

2.3.2 Calculating bounds on K

This method is taken from Staden [17]. It assumes that the random model of the DNA is composed of independent letters (not necessarily uniformly distributed). Given a score matrix, $M_{4\times L}$, every sequence R of length L ($R = r_1 \ldots r_L$, $r_i \in \{1, 2, 3, 4\}$) gets a score:

$$score(R) = \sum_{i=1}^{L} M_{r_i,i}$$

We would like to look at the distribution of $score(R)$ over all sequences of length L. It can be described as a sum of L independent random variables, $x_1, \ldots, x_L$, where

$$x_i = \begin{cases} M_{1,i}, & p_1 \\ M_{2,i}, & p_2 \\ M_{3,i}, & p_3 \\ M_{4,i}, & p_4 \end{cases} \tag{2.3}$$

where $p_1, \ldots, p_4$ are the given probabilities of the symbols in Σ.
The probability to get a score higher than τ in a specific position is the probability to see there at least one target from A, which is:

$$P(A) = \sum_{j=1}^{K} \prod_{n=1}^{L} P(d_{i+n-1} = r_n^j) \tag{2.4}$$

where $A = \{R^1, \ldots, R^K\}, R^i = (r_1^i, \ldots, r_L^i)$. Using (2.3) it can be calculated by:

$$P(A) = P(\sum_{i=1}^{L} x_i > \tau) \tag{2.5}$$

Notice that if we change x_i's definition by keeping the $M_{i,j}$'s and setting $p_1 = \ldots = p_4 = \frac{1}{4}$ in Eq. (2.3) (even if the true random model is not uniform), then K, the number of sequences of length L with a score higher than τ equals to:

$$K = P(\sum_{i=1}^{L} x_i > \tau) \times 4^L \tag{2.6}$$

Assume the score matrix contains only natural values. Then the random variables $x_1, \ldots, x_L$ can have only natural values. We define for each x_i its probability generating function: $G_i(x) = \sum_{j=1}^{4} p_j x^{M_{j,i}}$. The coefficient of x^N gives the probability that x_i has exactly the value N. Now, since we assume $x_1, \ldots, x_L$ are independent, the generating function of the probability distribution of $x_1 + \ldots + x_L$ is $G_{x_1+\ldots+x_L}(x) = G_1(x)G_2(x)\ldots G_L(x)$. By multiplying the generating functions we get the probability of every possible score (we need to calculate the coefficients of the multiplication).

Define for a polynomial $g(x) = \sum_{i=1}^{deg(g)} g_i x^i$ its *sparseness*, or the number of its non-zero coefficients to be $SP(g) = \sum_{i=1}^{deg(g)} 1_{g_i \neq 0}$. Then the computation of the coefficients of the multiplication of two polynomials $g(x), h(x)$ requires $O(SP(g) \cdot SP(h))$ operations. Denote $T = max\{\sum_{i=1}^{L} M_{r_i,i} | r_i \in \{A, C, G, T\}, i = 1, \ldots, L\}$. Then the complexity of the multiplication is $O(4(L-1)T)$: we have $L-1$ multiplications of a polynomial with degree less than T, with a polynomial with at most 4 non-zero entries.

However, our score matrix M doesn't necessarily contain only natural values. We would like to approximate K, the number of targets with a score higher than τ. We can have an upper bound and a lower bound for K in the following way:
To get an upper bound for K we can get a natural score matrix M' by $M'_{i,j} = \lceil M_{i,j} \cdot T \rceil$, where T is some large natural number. We use M' instead M in (2.3), and we calculate (exactly) the probability $p = P(x_1 + \ldots + x_L \geq \lfloor \tau \cdot T \rfloor)$ by the method of generating functions. If we use $p_1 = \ldots = p_4 = \frac{1}{4}$ then $K' = 4^L \cdot p$ is an upper bound for K: for every target $R = (r_1, \ldots, r_L)$ s.t. $score(R) = \sum_{i=1}^{L} M_{r_i,i} \geq \tau$, $\sum_{i=1}^{L} M'_{r_i,i} \geq \lfloor \tau \cdot T \rfloor$. The lower bound can be achieved in a similar way.

2.3.3 Dealing with both strands

Since DNA is double stranded, we should look for binding sites on both strands. We do it by simply scanning both strands and recording the best match to our $PSSM$. In order to give a p-value accounting for both strands, we simply assume we scan only the positive ($5'$) strand, but the set of 'legal' sequences is extending by taking $A \cup A^{RC}$, where A^{RC} is defined as the set of all reverse-complements of sequences from A. In case we enumerate A, we simply add right after step 3 in 2.3.1, :

4. $A = A \cup A^{RC}$

Note that here we also have to preform 'unique' since both a sequence and its reverse complement could get a score above τ. (This occurs often for palindromic motifs). If we only estimate $P(A)$ or K, we can simply multiply the bounds we got in 2.3.2 by two. This suffices since clearly, $|A \cup A^{RC}| \leq |A| + |A^{RC}| = 2K$, and $P(A \cup A^{RC}) \leq P(A) + P(A^{RC}) \leq 2P(A)$. (Remember that A consists the K sequences with highest probabilities.)

2.4 Calculating $P(H > 0)$

2.4.1 A Naive Approximation

There are $N-L+1$ positions in which the target sequence can appear. At each position, the probability of appearance is $P(A)$.

A usually more accurate estimation (though not a bound) comes from assuming that all the positions are independent, which gives the geometric distribution approximation

$$P(H = 0) \approx (1 - P(A))^{N-L+1} \tag{2.7}$$

As is shown in Robin and Daudin [16], the quality of this approximation varies, and depends on the overlapping structure of the sequence.

2.4.2 Exact Computation

One Target

We start with the simplest case, where $K = 1$, i.e. there is one target sequence, $R = (r_1, \ldots, r_L)$, with a score higher than τ. Denote by I_i the event of the target pattern appearing in the sequence at position i. that is :

$$I_i = \{D_i = r_1, ..., D_{i+L-1} = r_L\}$$

and denote by B_i the complementary event, that is $B_i = {I_i}^c$.

Now let T_i be the event of finding the pattern R at position i, and not finding it in $l < i$ (e.g. R was found at position i "for the first time"). That is , $T_i = (\bigcap_{j=1}^{i-1} B_j) \bigcap I_i$, and let $t_i = P(T_i)$. We define

$$\epsilon_i = \epsilon_i(R) = \begin{cases} 1 & R \text{ overlaps itself at distance } i \ (r_1 = r_{1+i}, ..., r_L = r_{L-i}) \\ 0 & \text{otherwise.} \end{cases} \tag{2.8}$$

With the above definitions, we can write :

$$P(H=0) = P(\bigcap_{i=1}^{N-L+1} B_i) = 1 - P(\bigcup_{i=1}^{N-L+1} T_i) \overset{T_i \bigcap T_j = \emptyset \text{ for } i \neq j}{=}$$

$$1 - \sum_{i=1}^{N-L+1} P(T_i) = 1 - \sum_{i=1}^{N-L+1} t_i \tag{2.9}$$

Thus, computing t_i for each i will help us get the desired result. To compute t_i we can write : (Here we assume $i \geq L$. The initial L values will be dealt with separately).

$$t_i = P(T_i) = P((\bigcap_{j=1}^{i-1} B_j) \bigcap I_i) = P(I_i) \cdot P(\bigcap_{j=1}^{i-1} B_j / I_i) =$$

$$P(A_i) \cdot (1 - P(\bigcup_{j=1}^{i-1} T_j / I_i)) = P(I_i) - P(I_i, \bigcup_{j=1}^{i-1} T_j) \overset{T_i \bigcap T_j = \emptyset \text{ for } i \neq j}{=}$$

$$P(I_i) - \Sigma_{j=1}^{i-L} P(I_i, T_j) - \Sigma_{j=i-L+1}^{i-1} P(I_i, T_j) \overset{I_i, T_j \text{ are independent for } j \leq i-L}{=}$$

$$P(I_i) \cdot (1 - \sum_{j=1}^{i-L} t_j) - \sum_{j=1}^{L-1} t_{i-L+j} \cdot P(I_i / T_{i-L+j}) = \tag{2.10}$$

$$\prod_{n=1}^{L} p_{r_n} \cdot (1 - \sum_{j=1}^{i-L} t_j) - \sum_{j=1}^{L-1} t_{i-j} \cdot \epsilon_j \prod_{n=L-j+1}^{L} p_{r_n} \tag{2.11}$$

This recurrence formula was deduced before in Robin and Daudin [16].

Taking (2.11) and subtracting two consecutive values yields

$$t_i - t_{i-1} = -\prod_{n=1}^{L} p_{r_n} t_{i-L} - \sum_{j=1}^{L-1} t_{i-j} \epsilon_j \prod_{n=L-j+1}^{L} p_{r_n} + \sum_{j=1}^{L-1} t_{i-1-j} \epsilon_j \prod_{n=L-j+1}^{L} p_{r_n} =$$

$$t_{i-L}(\epsilon_{L-1}\prod_{n=2}^{L}p_{r_n}-\prod_{n=1}^{L}p_{r_n})-t_{i-1}\epsilon_1 p_{r_L}+\sum_{j=2}^{L-1}t_{i-j}(\epsilon_{j-1}\prod_{n=L-j+2}^{L}p_{r_n}-\epsilon_j\prod_{n=L-j+1}^{L}p_{r_n})$$

which can be written as :

$$\sum_{j=0}^{L}(\epsilon_j p_{r_{L-j+1}}-\epsilon_{j-1})(\prod_{n=L-j+2}^{L}p_{r_n})\cdot t_{i-j}=0 \tag{2.12}$$

where we have extended the definition of ϵ to include :

$$\epsilon_0=\epsilon_L=1,\epsilon_{-1}=0$$

This is a standard linear recursion formulation with distance L. It has an explicit solution in terms of the roots of the characteristic polynomial (see Eriksson [7]). In order to simplify the solution for $P(H>0)$ we define

$$v_i=1-\sum_{j=1}^{i}t_j \tag{2.13}$$

From 2.11 we get

$$v_{i-1}-v_i=\prod_{n=1}^{L}p_{r_n}v_{i-L}-\sum_{j=1}^{L-1}(v_{i-j-1}-v_{i-j})\cdot\epsilon_j\prod_{n=L-j+1}^{L}p_{r_n}. \tag{2.14}$$

Which gives after simplification

$$\sum_{j=0}^{L}(\epsilon_j p_{r_{L-j+1}}-\epsilon_{j-1})(\prod_{n=L-j+2}^{L}p_{r_n})\cdot v_{i-j}=0 \tag{2.15}$$

Which is identical to 2.12! Thus, the difference in the values of t_k and v_k is only due to the different initial conditions. The initial conditions are given by $(i=1,\ldots,L)$:

$$t_i=\prod_{n=1}^{L}p_{r_n}-\sum_{j=1}^{i-1}\epsilon_j(\prod_{n=L-j+1}^{L}p_{r_n})t_{i-j}$$

$$(v_0=1),v_i=v_{i-1}-t_i, \tag{2.16}$$

The characteristic polynomial is

$$\rho(x) = \sum_{j=0}^{L} C_j x^j \tag{2.17}$$

Where the coefficients C_j are given by

$$C_{L-j} = (\epsilon_{j-1} - \epsilon_j p_{r_{L-j+1}})(\prod_{n=L-j+2}^{L} p_{r_n})\,\forall j = 0,..,L$$

Let $\lambda_1,..,\lambda_r$ be the roots of $\rho(x)$ with multiplicities $m_1,..,m_r$, respectively. The solution of 2.15 is : (for the t_i's the solution is similar)

$$P(H=0) = v_{N-L+1} = \sum_{j=1}^{r} w_j(N-L+1)\lambda_j^{N-L+1},$$

where the w_j's are polynomials of degree $m_j - 1$,

$$w_j(i) = \sum_{k=0}^{m_j-1} C_k^{(j)} i^k$$

The coefficients $C_k^{(j)}$ of the w_j's are determined by the L initial conditions, $v_1,\ldots,v_L$. If the L roots are distinct, the solution is a linear combination of exponents, given by :

$$P(H=0) = v_{N-L+1} = \sum_{j=1}^{L} w_j \lambda_j^{N-L+1} \tag{2.18}$$

The constants w_j are determined by the linear system :

$$v_i = \sum_{j=1}^{L} \lambda_j^i w_j, \;\; i = 1,\ldots,L$$

Or, in matrix form :

$$\vec{w} = \Lambda^{-1}\vec{v}$$

where $\vec{w} = (w_1,\ldots,w_L)$, $\vec{v} = (v_1,\ldots,v_L)$, and $\Lambda(i,k) = \lambda_i^k, \forall i,k = 1,\ldots,L$. Λ is invertible since it is the product of a van der Monde matrix ($\lambda_i \neq 0\,\forall i = 1,\ldots,L$) and an invertible diagonal matrix.

Multiple Targets

Now we move to the more general case, where $K > 1$. Suppose K is small enough, and we are given the set A of all sequences with score higher than τ, $A = \{R^1, \ldots, R^K\}$. Recall that H counts the number of appearances of any of them in a DNA sequence. In a similar way to the case of $K = 1$, we define T_i to be the event of finding any of the K targets at position i, and not finding any of them at $l < i$ (e.g. a target was found at i "for the first time"). Let $t_i = P(T_i)$; $P(H > 0) = \sum_{i=1}^{N-L+1} t_i$. The t_i's are calculated recursively; define I_i to be the event of finding any of the K targets at position i (not necessarily for the first time). In addition, for every $j = 1, .., K$, define $I_i(j)$ to be the event of finding R^j at index i, and $T_i(j) = I_i(j) \bigcap T_i$. Let also $t_i(j) = P(T_i(j))$. Clearly, from the above definitions, it follows that :

$$T_i = \bigcup_{j=1}^{K} T_i(j) \ , \ t_i = \sum_{j=1}^{K} t_i(j) \tag{2.19}$$

The solution will depend on the overlap patterns of pairs of our K sequences. Define for each pair of sequences R^j and R^m the overlap matrix

$$\epsilon_i(m, j) = \epsilon_i(R^m, R^j) = \begin{cases} 1 & R^j \text{ overlaps } R^m \text{ at distance } i \ (r_1^j = r_{1+i}^m, \ldots, r_{L-i}^j = r_L^m) \\ 0 & \text{otherwise.} \end{cases} \tag{2.20}$$

Then a recursion similar to 2.10 holds (Blom and Thorburn [4]):

$$t_i(j) = P(I_i(j))(1 - \sum_{n=1}^{i-L} t_n) - \sum_{n=1}^{L-1} \sum_{m=1}^{K} t_{i-n}(m) P(I_i(j)/T_{i-n}(m)) \tag{2.21}$$

Define L vectors of size K, $\vec{Q}_1, \ldots, \vec{Q}_L$ by

$$\vec{Q}_j(m) = \prod_{n=1}^{j} P(x = r_n^m), \ m = 1, \ldots, K, \ n = 1, \ldots, L \tag{2.22}$$

Then $P(I_i(j)/T_{i-n}(m)) = \epsilon(m, j) \cdot \vec{Q}_n(j)$, and Eq. (2.21) can be written in matrix form:

$$\vec{t_i} = diag(\vec{Q}_L)(\vec{1}_K - \sum_{n=1}^{i-L} 1_{K\times K}\vec{t_n}) - \sum_{n=1}^{L-1} diag(\vec{Q}_n)\epsilon_n^T \vec{t}_{i-n} \qquad (2.23)$$

where ϵ_n is the matrix of overlaps at distance n, $\epsilon_n^T(i,j) = \epsilon_n(j,i)$. $\vec{t_i}$ is the vector of the $t_i(j)$'s, $\vec{1_K}$ and $1_{K\times K}$ are a vector and a matrix of all ones, respectively, and $diag(\vec{Q})$ is a diagonal matrix of size $K \times K$ with the elements of $\vec{Q}$ on the diagonal.

The initial conditions are $(i = 1, \dots, L)$:

$$\vec{t_i} = \vec{Q}_L - \sum_{n=1}^{i-1} diag(\vec{Q}_n)\epsilon_n^T \vec{t}_{i-n}$$

The first sum on the right hand side of Eq. (2.23) can be easily computed, since all the rows of $1_{K\times K}$ are identical. Therefore, the major computational effort when advancing one step in the recursion, is in calculating the second sum. This can be done efficiently if we compute the matrices $\epsilon_1, \dots, \epsilon_{L-1}$ once, in advance. The exact performance depends on the structure of the ϵ_n's, but it is bounded by $O(LK)$: suppose ϵ_n^T has m different rows, $\vec{c}_1, \dots, \vec{c}_m$. Every row specifies for all targets if their last $L-n$ letters are identical to a specific sequence of length $L-n$ (unique to this row) which is the beginning of one of the targets. If $\vec{c}_j(i) = 1$ it means that target number i ends with the unique sequence of length $L-n$ of row $\vec{c}_j$. Then we get that for every $k \neq j$, $\vec{c}_k(i) = 0$. Let n_j, $j = 1, \dots, m$ be a count of the number of non-zero entries in each row. Then $\sum_{j=1}^m n_j \leq K$. It means that each of the ϵ_n's matrices can be stored in space $O(K)$. We need to multiply $\vec{c}_j$ and $\vec{t}_{i-n}$. The multiplication takes $O(n_j)$, and the multiplication of all different rows takes $O(\sum_{j=1}^m n_j) = O(K)$. The result is then multiplied by $diag(\vec{Q}_n)$, which also takes $O(K)$. Since we have $L-1$ different matrices the computation of one recursion step takes $O(LK)$, and thus the computation of the whole recursion takes $O(NLK)$.

To calculate the ϵ_n's in the beginning, we need to compare two lists of size K. The first contains the first n letters of all the sequences, and the second contains the last n letters of them. This can be done efficiently using the 'meet in the middle' approach ([5]), often used in cryptology. In this method one of the lists is sorted, and then every element in the second list is compared to the sorted list by a binary search. The time complexity of the beginning step reduces to $O(LKlog(K))$.

The total complexity of the algorithm is $O(LK(N + log(K)))$.

2.4.3 An upper bound for $P(H > 0)$

Suppose K is large, and we are not given A, but a (tight) upper bound on its probability: $\rho \geq P(I_i)$. Then to get an upper bound on $P(H > 0)$ we set $\epsilon \equiv 0$ (i.e. no overlaps between all the K sequences). In section 2.4.4 we give a proof that this yields an upper bound.

Substituting $\epsilon \equiv 0$ in Eq. (2.21) gives :

$$t_i = \rho(1 - \sum_{n=1}^{i-L} t_n) \tag{2.24}$$

With initial conditions :

$$t_1 = \ldots = t_L = \rho \tag{2.25}$$

This can be solved using the characteristic polynomial :

$$P(x) = x^L - x^{L-1} + \rho \tag{2.26}$$

which usually has only simple roots. In this case the solution is

$$P(H = 0) = \sum_{j=1}^{L} u_j g_j^{N-L+1} \tag{2.27}$$

Where $g_1, .., g_L$ are the roots of the characteristic polynomial, and $u_1, .., u_L$ are determined by the initial conditions. If (2.26) has non simple roots, than there is a slight change in the solution (see section 2.4.2 for details on solving the recurrence relations). So we have obtained easily an upper bound for $P(H > 0) = \sum_{i=1}^{N-L+1} t_i$. Time complexity: Finding numerical approximations to the roots of the polynomial depends only on L. The calculation of $P(H = 0)$ then takes $O(LlogN)$.

In section 2.4.5 it is shown that in the cases of interest, where $P(H > 0)$ is small, there is a group of sequences $A \subset \Sigma^L$ for which $\epsilon \equiv 0$, and thus we expect the bound to be tight.

2.4.4 $\epsilon \equiv 0$

Intuitive Explanation

Let $A \subset \Sigma^L$ be a general set of size K, $A = \{R^1, \dots, R^K\}$, and assume there exists a non overlapping set $A' \subset \Sigma^L$ of size K, $A' = \{R^{1'}, \dots, R^{K'}\}$ such that $P(x = r_n^{m'}) = P(x = r_n^m)$, for $m = 1, \dots, K$, $n = 1, \dots, L$. We want to show that:

$$P'(H > 0) \geq P(H > 0) \tag{2.28}$$

where $P'(H > 0)$ is the probability to have at least one appearance of a sequence from A' in a random DNA sequence of length N, and $P(H > 0)$ is the same for A.

Notice that the mean number of matches of targets from A in a random sequence of length N is equal to that of targets from A' (in section 3.3 there is a description of the calculation of the mean number of matches). It means that if we look at all possible sequences of length N, the overall number of matches of targets from A will be equal to that of targets from A'. Now, suppose we concatenate all possible sequences of length N. Since the targets from A have self overlaps they will tend to appear in groups more than the targets from A', which don't have self overlaps. The targets from A' will be spread in a more uniform manner on the long sequence. Thus, if we cut randomly a sequence of length N from this long sequence, the probability to have at least one appearance of a target from A' will be higher than the probability to see at least one appearance of a target from A.

Proof

Let $\vec{v}_i(m)$ be defined by $\vec{v}_i(m) = \frac{1}{K} - \sum_{j=1}^{i} t_j(m)$, so $v_i = \sum_{m=1}^{K} \vec{v}_i(m)$ holds (see 2.4.2 for the definition of v_i) .

Now

$$P'(H > 0) = 1 - v'_{N-L+1} = 1 - \sum_{m=1}^{K} \vec{v'}_{N-L+1}(m)$$

and

$$P(H > 0) = 1 - v_{N-L+1} = 1 - \sum_{m=1}^{K} \vec{v}_{N-L+1}(m)$$

It is enough to show that $\forall N \ \forall m = 1, \ldots, K, \ \vec{v'}_{N-L+1}(m) \leq \vec{v}_{N-L+1}(m)$, which we will write as:

$$\vec{v'}_{N-L=1} \leq \vec{v}_{N-L=1}$$

Define $p_{max} = max\{p_1, \ldots, p_4\}$. We will show by induction:

$$\vec{v}_i - \vec{v'}_i \geq p_{max}(\vec{v}_{i-1} - \vec{v'}_{i-1}), \forall i > 1$$

(For $i = 1$, $\vec{v}_1 = \vec{v'}_1$ and the claim is obvious).
Using Eq. (2.23) we get

$$\vec{v}_{i-1} - \vec{v}_i = \vec{Q}_L(v_{i-L}) - \sum_{n=1}^{L-1} diag(\vec{Q}_n)\epsilon_n^T(\vec{v}_{i-n-1} - \vec{v}_{i-n}) \tag{2.29}$$

This gives the recursion:

$$\sum_{j=0}^{L}(diag(\vec{Q}_j)\epsilon_j^T - diag(\vec{Q}_{j-1})\epsilon_{j-1}^T)\vec{v}_{i-j} = \vec{0} \tag{2.30}$$

where $\epsilon_{-1}, \epsilon_0, \epsilon_L$ are all matrices of size $K \times K$, ϵ_{-1} is the zero matrix, ϵ_0 is the identity matrix, and ϵ_L is a matrix of all ones.

Induction Basis

The initial conditions are : $(i = 1, \ldots, L)$

$$\vec{v}_i = \frac{1}{K}\vec{1}_K - \sum_{j=1}^{i}\vec{t}_j, \quad \vec{v'}_i = (\frac{1}{K} \cdot \vec{1}_K - i\vec{Q}_L) \tag{2.31}$$

Using (2.31), we need to show :

$$i\vec{Q}_L - \sum_{j=1}^{i}\vec{t}_j \geq p_{max}((i-1)\vec{Q}_L - \sum_{j=1}^{i-1}\vec{t}_j)$$

Or :

$$((1-p_{max})(i-1)\vec{Q}_L + \vec{Q}_L) \geq (1-p_{max})\sum_{j=1}^{i-1}\vec{t}_j + \vec{t}_i$$

The latter inequality results immediately from the noting that

$$\vec{t}_i \leq \vec{Q}_L, \forall i \in \mathbb{N}, m = 1, \ldots, K$$

Induction Step

For $i > L$, using (2.30) for $\vec{v'}_i$ and $\vec{v}_i$ and subtracting gives after simplification :

$$\vec{v}_i - \vec{v'}_i = \vec{v}_{i-1} - \vec{v'}_{i-1} - diag(\vec{Q}_L)(\vec{v}_{i-L} - \vec{v'}_{i-L}) + \sum_{j=1}^{L-1} \epsilon_j^T diag(\vec{Q}_j)(\vec{v}_{i-1-j} - \vec{v}_{i-j})$$

Now assume, by induction :

$$\vec{v}_m - \vec{v'}_m \geq p_{max}(\vec{v}_{m-1} - \vec{v'}_{m-1}),\ \forall m < i$$

Then

$$\vec{v}_i - \vec{v'}_i \geq \vec{v}_{i-1} - \vec{v'}_{i-1} - diag(\vec{Q}_L)(\vec{v}_{i-L} - \vec{v'}_{i-L}) \geq$$
$$(\vec{v}_{i-1} - \vec{v'}_{i-1})(1 - p_{max}) \overset{(*)}{\geq} p_{max}(\vec{v}_{i-1} - \vec{v'}_{i-1}) \tag{2.32}$$

(*) Notice that (2.32) holds only for $p_{max} \leq \frac{1}{2}$. However, the frequencies of DNA letters are usually measured on both strands, which implies %A=%T, %C=%G. Moreover, according to Chargaff's second parity rule (Karkas et al. [9]), the latter equalities hold approximately also for long single strands of DNA. Although there are known deviations from this rule, it is very unlikely that one of the nucleotides composes most the promoter regions of a given genome. Thus in practice we can assume $p_{max} \leq \frac{1}{2}$, and using (2.32) we get that $\vec{v}_i \geq \vec{v'}_i, \forall i \in \mathbb{N}$.

2.4.5 The Maximum Size of non overlapping sets

Let $A \subset \Sigma^L$, $|\Sigma| = S$, be a subset of sequences. We say that A is a non overlapping set if $\epsilon_i(j,l) = 0, \forall i = 1..L, j,l = 1..|A|$. Let $\Gamma \subset \Sigma^L$ be the set of all non overlapping sequences. Then $K_0(S,L)$ is defined by

$$K_0(S,L) = max\{|A|.A \subset \Gamma, \text{A is a non overlapping set}\} \tag{2.33}$$

For $K \leq K_0$, there is a non overlapping set of size K. Therefore we expect that in this case our bound will be tight. For $K > K_0$, however, there is no such set, and the computation of the bound is meaningless so we cannot expect it to be a tight bound. Thus, in order to know when our bound is tight, we need to know $K_0(S,L)$. For small values of S and L, $K_0(S,L)$ can be found by exhaustive search. For the general case, we have bounds on the value of K_0.

A Lower Bound for K_0

We show here that

$$\frac{S^{L+1}}{S^{2\sqrt{L}}} \leq K_0(S,L)$$

Let m be an integer, $0 < m \leq \lfloor \frac{L}{2} \rfloor$. Define the set A_m as

$$A_m = \{R | r_1, \ldots, r_m = 1, r_{m+1} \neq 1, r_L \neq 1,$$
$$(r_{m+1+i}, .., r_{2m+i}) \neq \underbrace{(1, .., 1)}_{m}, \forall i = 1, .., L-2m-1\} \tag{2.34}$$

A_m is a non overlapping set. Thus $K_0 \geq \max_{m=0}^{\lfloor \frac{L}{2} \rfloor} |A_m|$

For a given m, let Y_n^i be the number of sequences of length n over Σ which end with exactly i $'1'$'s and don't contain m contiguous $'1'$'s, $i = 0, \ldots, m-1$. Then

We can write the following recursion relations :

$$Y_n^i = Y_{n-1}^{i-1}, \text{ for } i = 1, \ldots, m-1$$

$$Y_n^0 = (S-1)(\sum_{i=0}^{m-1} Y_{n-1}^i)$$

Then

$$|A_m| = (S-1)^2 \sum_{i=0}^{m-1} Y_{L-m-2}^i$$

Notice that for $i = 1, \ldots, m-1$, $Y_n^i = Y_{n-1}^{i-1} = \ldots = Y_{n-i}^0$

Then

$$Y_n^0 = (S-1) \sum_{i=1}^{m} Y_{n-i}^0$$

and

$$|A_m| = (S-1)^2 \sum_{i=1}^{m} Y_{L-m-1-i}^0 = (S-1) Y_{L-m-1}^0 \tag{2.35}$$

Define $Y_n := Y_{n-m+1}^0$, then we get $|A_m| = (S-1)Y_{L-2}$, where Y_n is given by the following recursion relation :

$$Y_n = (S-1)\sum_{i=1}^{m} Y_{n-i}$$

With the initialization:

$$Y_1 = Y_2 = \ldots Y_{m-1} = 0, \;\; Y_m = S-1$$

The characteristic polynomial of this recursion relation is

$$Q(x) = x^m + (S-1)\sum_{i=0}^{m-1} x^i$$

and thus, as before, Y_n is given by :

$$Y_n = \sum_{i=1}^{m} u_i g_i^n$$

Where the constants u_i's are determined by the initial conditions. Substituting in 2.35 gives

$$|A_m| = (S-1)\sum_{i=1}^{m} u_i g_i^{L-2}$$

$|A_m| \geq (S-1)^2 S^{\frac{(m-1)(L-m-2)}{m}}$, since the number of sequences of length $L-m-2$ which don't contain m contiguous $'0'$ is at least $S^{\frac{(m-1)(L-m-2)}{m}}$. This is true because we can divide the sequence of length $L-m-2$ into subsequent sequences of length m, and in each of them put any of the S letters in the first $m-1$ indices, and something different than $'1'$ in the $m'th$ index. Take $m = \sqrt{L}$ to get :

$$K_0(S,L) \geq |A_{\sqrt{L}}| \geq (S-1)^2 S^{\frac{(\sqrt{L}-1)(L-\sqrt{L}-2)}{\sqrt{L}}} \geq S^{L-2\sqrt{L}+1+\frac{2}{\sqrt{L}}} \geq \frac{S^{L+1}}{S^{2\sqrt{L}}}$$

An upper bound for K_0

Suppose $K \leq K_0$. Let A be a non overlapping set, $|A| = K$. We use here the uniform random DNA model. Clearly, $\forall k > 0$, $0 \leq t_k \leq 1$. Recall that the characteristic polynomial for the non-overlapping case is :

$$P(x) = x^L - x^{L-1} + K \cdot S^{-L} \tag{2.36}$$

We will use the following lemma : (proof in the following page)

$$\text{If } P(x) = x^L - x^{L-1} + K \cdot S^{-L} \text{ has no root } r \in [0,1], \text{then } \exists k > 0 \text{ such that } t_k < 0. \tag{2.37}$$

Looking at $P(x)$, we see that $P(0) = P(1) = K \cdot S^{-L} > 0$.
Derivation and comparing to zero gives :

$$P'(x) = L \cdot x^{L-1} + (L-1) \cdot x^{L-2} = 0 \tag{2.38}$$

Which gives : $x = 0$ or $x = 1 - \frac{1}{L}$, the latter being a local minima of P.
Thus, $\exists r, 0 < r < 1, P(r) = 0 \iff P(1 - \frac{1}{L}) \leq 0$.
This gives :

$$(1 - \frac{1}{L})^L - (1 - \frac{1}{L})^{L-1} + KS^{-L} \leq 0 \tag{2.39}$$

$$K \leq \frac{(1 - \frac{1}{L})^{L-1} S^L}{L} \tag{2.40}$$

Thus, from lemma 2.37, we conclude that

$$K_0(S, L) \leq \frac{(1 - \frac{1}{L})^{L-1} S^L}{L} \sim \frac{S^L}{eL} \tag{2.41}$$

The Tightness of the Bounds for K_0

We showed that
$$\frac{S^{L+1}}{S^{2\sqrt{L}}} \leq K_0(S, L) \leq \frac{(1 - \frac{1}{L})^{L-1} S^L}{L} \sim \frac{S^L}{eL}$$

The two bounds are asymptotically tight in the sense that :

$$\frac{log(S^{L+1-2\sqrt{L}})}{log(\frac{S^L}{eL})} \underset{L \to \infty}{\longrightarrow} 1$$

The tightness of the bounds can be viewed in Fig. 2.1 which is for $S = 4$.
It is clear that we are interested in the case where $P(H > 0)$ is very small (If not, the binding site will not be statistically significant). In this case $K \ll S^L$. We conclude

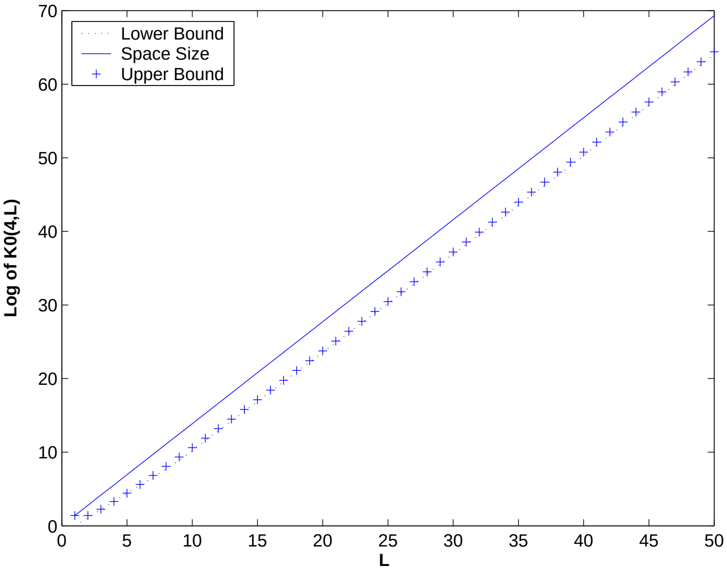


Figure 2.1: $K_0(S, L)$ for $S = 4$ and $1 \leq L \leq 50$: Space size (contiguous line), lower bound (dotted line), and upper bound ('+' line)

that in the cases of interest, the bound will be tight. This is because for $K < \frac{s^{L+1}}{S^2\sqrt{L}}$ we know there is a non overlapping set of size K, and thus we expect the bound to be tight.

Lemma

Here we prove that if $P(x) = x^L - x^{L-1} + K \cdot S^{-L}$ has no root $r \in [0, 1]$, then $\exists k > 0$ such that $t_k < 0$. By derivation of P, we see that all the roots of P are simple. (The only multiple root possible is $x = 1 - \frac{1}{L} \in [0, 1]$ which is obtained for $K = \frac{(1-\frac{1}{L})^{L-1}S^L}{L}$). It can also be easily seen that P has no pure imaginary roots.
First, we will show that if $z = re^{i\theta}$ is a root of the characteristic polynomial P, then the only other root with the same absolute value r is its adjoint $\overline{z} = re^{-i\theta}$
Assume z is a root, and , suppose there is some other root $z' = re^{i\eta}$. Then , substituting both into P we get

$$r^L e^{i\theta L} + r^{L-1} e^{i\theta(L-1)} = r^L e^{i\eta L} + r^{L-1} e^{i\eta(L-1)} \tag{2.42}$$

$$re^{i\theta L} + e^{i\theta(L-1)} = re^{i\eta L} + e^{i\eta(L-1)}$$

Which gives two equations (for real and imaginary parts)

$$r(\cos(\theta L) - \cos(\eta L)) = \cos(\eta(L-1)) - \cos(\theta(L-1))$$

$$r(\sin(\theta L) - \sin(\eta L)) = \sin(\eta(L-1)) - \sin(\theta(L-1))$$

Multiplying the two equations gives :

$$(\cos(\theta L) - \cos(\eta L))(\sin(\eta(L-1)) - \sin(\theta(L-1))) = (\sin(\theta L) - \sin(\eta L))(\cos(\eta(L-1)) - \cos(\theta(L-1)))$$

Using simple trigonometric identities ($sina - sinb = 2sin\frac{a-b}{2}cos\frac{a+b}{2}, \ \ cosa - cosb = -2sin\frac{a-b}{2}sin\frac{a+b}{2}$) we get :

$$-2\sin\frac{L(\theta+\eta)}{2}\sin\frac{L(\theta-\eta)}{2}2\cos\frac{(L-1)(\eta+\theta)}{2}\sin\frac{(L-1)(\eta-\theta)}{2} =$$
$$2\cos\frac{L(\theta+\eta)}{2}\sin\frac{L(\theta-\eta)}{2}(-2)\sin\frac{(L-1)(\eta+\theta)}{2}\sin\frac{(L-1)(\eta-\theta)}{2}$$

Consider first the case where, either $\sin\frac{L(\theta-\eta)}{2} = 0$, or $\sin\frac{(L-1)(\theta-\eta)}{2} = 0$. This gives $\eta = \theta - \frac{2\Pi k}{L}$, or $\eta = \theta - \frac{2\Pi k}{L-1}$.
Assign $\eta = \theta - \frac{2\Pi k}{L}$ to equation 2.42 gives:

$$re^{i\theta L} + e^{i\theta(L-1)} = re^{i\theta - \frac{2\Pi k}{L}L} + e^{i\theta - \frac{2\Pi k}{L}(L-1)}$$

$$e^{i\theta L}[r + e^{-i\theta} - re^{-i2\Pi k} - e^{i(\frac{2\Pi k}{L} - (\theta + 2\Pi k))}] = 0 \Rightarrow e^{-i\theta} = e^{i(\frac{2\Pi k}{L} - (\theta))} \Rightarrow$$

$$\Rightarrow e^{i\frac{2\Pi k}{L}} = 1 \Rightarrow \frac{2\Pi k}{L} = 2\Pi k', \ \ \text{for some } k' \in N$$

In a similar way, assigning $\eta = \theta - \frac{2\Pi k}{L-1}$ to 2.42 gives $k = (L-1)k'$ for some $k' \in N$. Both cases give in fact $\eta = \theta$, or $z' = z$.
If the latter two cases are not satisfied, we get :

$$\sin\frac{L(\theta+\eta)}{2}\cos\frac{(L-1)(\eta+\theta)}{2} - \cos\frac{L(\theta+\eta)}{2}\sin\frac{(L-1)(\eta+\theta)}{2} = 0$$

Which Gives ($sin(a \pm b) = sina \cdot cosb \pm cosa \cdot sinb$)

$$\sin(\frac{L(\theta+\eta)}{2} - \frac{(L-1)(\eta+\theta)}{2}) = 0$$

$$\sin\frac{\theta+\eta}{2} = 0$$

$$\eta = 2\Pi k - \theta$$

Which implies $z' = \overline{z}$.
So we have proven that for a given root of P, $z = re^{i\theta}$, the only other root with the same absolute value is $\overline{z} = re^{-i\theta}$.

Using the fact that P has L distinct roots, we can order them, according to their absolute value, $z_1, z_2, \ldots, z_L$. Suppose z_1 is negative (and real). Recall that : $t_k = \sum_{i=0}^{L} u_i z_i^k$, for some constants u_i's. Then we get

$$\lim_{k\to\infty} \frac{t_k}{u_1 z_1^k} = 1$$

This implies that for every k_0, there is some $k > k_0$ such that $t_k < 0$.

If z_1 is not real and negative, and since we assume no real positive roots, the two roots with the largest absolute value are $z_1 = re^{i\theta}$ and $z_2 = \overline{z_1} = re^{-i\theta}$, and get that

$$\lim_{k\to\infty} \frac{t_k}{u_1 z_1^k + u_2 z_2^k} = 1 \tag{2.43}$$

But we have

$$\widehat{t_k} := u_1 z_1^k + u_2 z_2^k = r^k(u_1 e^{i\theta k} + u_2 e^{-i\theta k}) =$$

$$r^k(u_1 \cos(\theta k) + u_2 \cos(-\theta k) + i(u_1 \sin(\theta k) + u_2 \sin(-\theta k))) =$$

$$r^k((u_1 + u_2)\cos(\theta k) + i(u_1 - u_2)\sin(\theta k))$$

Which gives :

$$\frac{Im(\widehat{t_k})}{Re(\widehat{t_k})} = \frac{u_1 - u_2}{u_1 + u_2}\tan(\theta k)$$

Recall that $\theta \neq 0$. (We assume no real positive roots) and $\theta \neq \pm\frac{\Pi}{2}$ also, since P does not have pure imaginary roots. It is clear that there is some $\epsilon > 0$ such that for every k_0 there is some $k > k_0$ such that $\tan(\theta k) > \epsilon$. Thus, for every k_0, there is some $k > k_0$ such that $\frac{Im(\widehat{t_k})}{Re(\widehat{t_k})} > \epsilon\frac{u_1-u_2}{u_1+u_2}$. But, t_k is real, so for 2.43 to hold we must have

$$\frac{Im(\widehat{t_k})}{Re(\widehat{t_k})} \xrightarrow[k\to\infty]{} 0 \Rightarrow u_1 = u_2$$

This implies $\widehat{t_k} = 2r^k u_1 \cos(\theta k)$
Therefore, there is some k_0 such that for every $k > k_0$ we have : $sgn(t_k) = sgn(\widehat{t_k}) = sgn(2r^k u_1 \cos(\theta k))$
But for every k_0, there is some $k > k_0$ such that $\cos(\theta k) < 0)$, which gives, $t_k < 0$.

Chapter 3

Results and Discussion

Here we compare the results of our algorithm with MatInspector (Quandt et al. 14), a commercially available software that looks for putative binding sites in DNA sequences according to PSSMs. The comparison is based on the Saccharomyces Cerevisiae Promoters Database (SCPD, Zhu and Zhang [20]), which contains a list of known transcription factors' binding sites, and their location in the Saccharomyces Cerevisiae genome. In section 3.1 we describe the data we use and how we measure the performance of our algorithm on it. In section 3.2 we describe how we compare our results to MatInspector's results. The comparison is done with different interpretations of the output of MatInspector software.

3.1 Saccharomyces Cerevisiae Genome

We used the Saccharomyces Cerevisiae Promoters Database (SCPD, Zhu and Zhang [20]), which contains a list of known transcription factors' binding sites, and their location in the Saccharomyces Cerevisiae genome. It contains the PWMs of 24 transcription factors. We extracted the upstream sequences of all genes that are known to be bound by at least one of these 24 transcription factors. There are 135 such genes. We used the upstream regions given in SCPD, all of them end right before the translation start codon (ATG), and their length is between 500 to 800 bases.

We computed the p-values (for $H > 0$) for each of the 24 transcription factors on each of the 135 genes (we use here the upper bounds of the p-values, see section 2.4.3). To deal with the problem of multiple comparisons we used the method of Benjamini and

Hochberg [3], which controls the false discovery rate (FDR method). This method ensures that the mean of the false discovery rate is lower than the parameter $0 < Q < 1$. We applied it to each of the genes (to the 24 p-values of the transcription factors), to get the statistically significant putative binding sites for every gene (see figure 3.1).

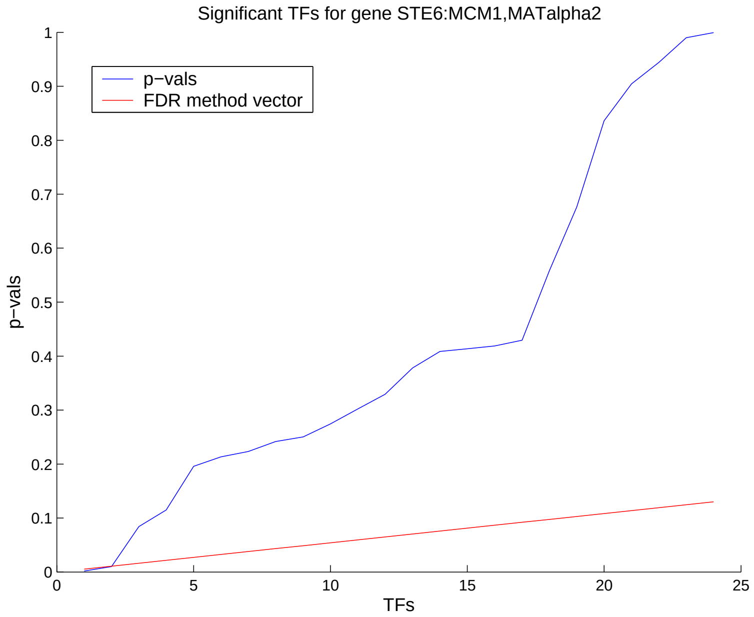


Figure 3.1: Applying the FDR method on the p-values of the 24 transcription factors on gene STE6. Here we choose $Q = 0.13$, which outputs two 'real' transcription factors. The two statistically significant transcription factors are MCM1, MATalpha2; these are exactly the transcription factors which are known to bind STE6 (SCPD).

For a specific gene, a transcription factor is said to be "positive" at a certain value of Q if it was found to be statistically significant by the FDR method using this value of Q. It is said to be "true positive" (TP) if it is positive, and it is also known to bind this gene (by SCPD). It is "false positive" (FP) if it is positive, but it isn't known to bound this gene. In a similar way we define "true negatives" (TN) and "false negatives" (FN). We present the results in a ROC (Receiver Operating Characteristic) curve. This curve describes the relative amount of correct predictions as a function of the relative amount of incorrect ones, or the tradeoff between "true positives" and "false positives", averaged on all 135 genes. Clearly, a low Q value ensures low number of false positives, but at the cost of low number of true positives. Hence we used a range of values of the FDR method parameter, $0.01 < Q < 0.45$, to select an optimal working point. The results are shown in Fig. 3.2.

3.2 Comparison with MatInspector

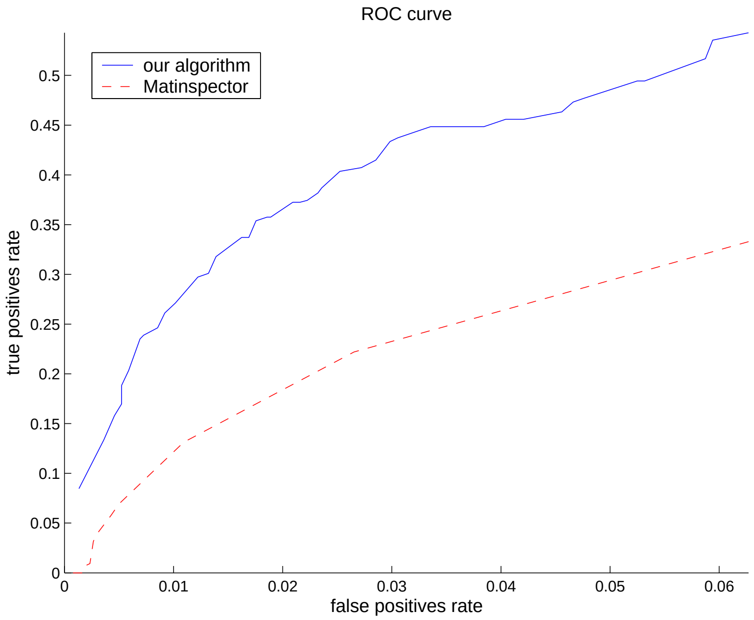


Figure 3.2: ROC curves showing ability to identify binding sites in promoters of genes from SCPD. The x-axis shows the *false positives rate* FP/(FP+TN), the y-axis shows the *true positives rate*, TP/(TP+FN). The results shown are average values on all 135 genes. The solid curve shows our results in a range of values of the FDR method parameter, $Q \in [0.01, 0.45]$. The dashed curve shows MatInspector results in a range of thresholds on the number of matches $(1-150)$.

Given a PWM and a DNA sequence, MatInspector outputs a list of all the matches of this PWM in the promoter. MatInspector computes a PSSM out of the PWM (not by taking log likelihood ratio). A match is a position in the sequence in which the score according to the PSSM is above a certain threshold. MatInspector recommends to use its optimized score threshold (a different score threshold for every PSSM), which is computed by MatInspector "in a way that a minimum number of matches is found in non-regulatory test sequences". However, it is not published how this threshold is computed for every PSSM. We use the same 24 PWMs as in our search for transcription factors binding sites (see section 3.1).

3.2.1 Our $P(H > 0)$ and FDR vs. MatInspector's Optimized Score Threshold and a Threshold on the Number of Matches

In order to decide which of the transcription factors has a number of matches which is statistically significant, we decide on a threshold on the number of matches. We used a range of threshold values (which play the role of Q in our method) to find the optimal working point. When a transcription factor has a number of matches which is higher than the threshold value (the threshold on the number of matches), we decide that it is statistically significant. We made this for the same 135 genes and 24 PWMs. A comparison between MatInspector results and our results is shown in Fig. 3.2.

It can be seen in Fig. 3.2 that our algorithm is significantly more precise in searching for putative binding sites in genes from SCPD than MatInspector (in the sense of achieving a better true positives vs. false positives tradeoff). We repeated our analysis using MatInspector's PSSMs (instead of log-likelihood ratio PSSMs) and our results were very similar. Thus we conclude that our higher performance is not due to differences in the PSSMs.

3.2.2 Using MatInspector's RE Values

MatInspector provides an additional parameter for every PWM. It is called ”RE value”, and it is an estimation of the mean number of matches of a PWM in a random sequence of length 1000. A match here is a score higher than 0.85 according to MatInspector's PSSMs (MatInspector's PSSMs have scores in the range $[0, 1]$). We evaluated MatInspector results using the RE values too. We ran the MatInspector software on the same 135 promoters and 24 PWMs. We used a threshold of 0.85 on the PWMs scores in order to use the RE parameter in a meaningful way. In order to decide which of the transcription factors has a number of matches which is statistically significant, we calculated the ratio between the RE value of each PWM (out of the 24) and the number of matches of this PWM (both are given by MatInspector). When this ratio is lower than a certain threshold, we decide that this PWM is statistically significant in the given DNA sequence (we used a range of threshold values).

We repeated this analysis using our computation of the mean of number of matches of every PWM (see section 3.2.3). We used the same PSSMs as MatInspector, and the match score thresholds were all 0.85. We used our estimation to the mean of the number of matches instead of the RE values given by MatInspector. As can be seen in Fig. 3.3, our estimation of the mean number of matches again improves the performance.

3.2.3 Using MatInspector's Optimized Score Thresholds with Our Calculation of the Mean Number of Matches

Finally we used MatInspector optimized score thresholds with our estimation of the number of matches of every PWM. For each PWM we calculated the mean number of matches, where a match is having a score higher than MatInspecor's optimized score threshold for the PSSM (MatInspector's PSSM) of this PWM. This was done because we saw that our estimation of the mean of the number of matches improves the performance in the case of a constant score threshold (0.85). In addition, it is obvious that taking a constant score threshold on all the PSSMs doesn't give the best results. Since different PWMs are defined with different stringency, each PWM should have its own score threshold. However, MatInspector software has a limitation on the definition of the score thresholds. The only possibility to choose different score thresholds for different PWMs is by choosing the optimized score thresholds, which are determined by an unpublished procedure of MatInspector (and there is a possibility to choose the optimized score threshold ± a constant, the same constant for all the matrices). The other possibility is to choose a constant score threshold for all PWMs (we did this with a score threshold of 0.85). Choosing the optimized score thresholds combined with using our estimation of the mean number of matches gave the highest performance of all MatInspector's runs. Still, our method shows the highest results in terms of true positives rate vs. false positives rate tradeoff. A comparison of all the results is shown in Fig. 3.3.

3.3 Our calculation of the Mean Number of Matches

Let

$$y_i = \begin{cases} 1 & \text{there is a match in position } i \ (score(D_i \ldots D_{i+L-1}) \geq \text{threshold, or } I_i \text{ occurred}). \\ 0 & \text{otherwise.} \end{cases} \tag{3.1}$$

for $i = 1, \ldots, N-L+1$. Then the mean of the number of matches,

$$E = E(H) = E(\sum_{i=1}^{N-L+1} y_i) = \sum_{i=1}^{N-L+1} E(y_i) = (N-L+1) \cdot E(y_1)$$

$$E(y_1) = Prob(y_1 = 1) = P(A)$$

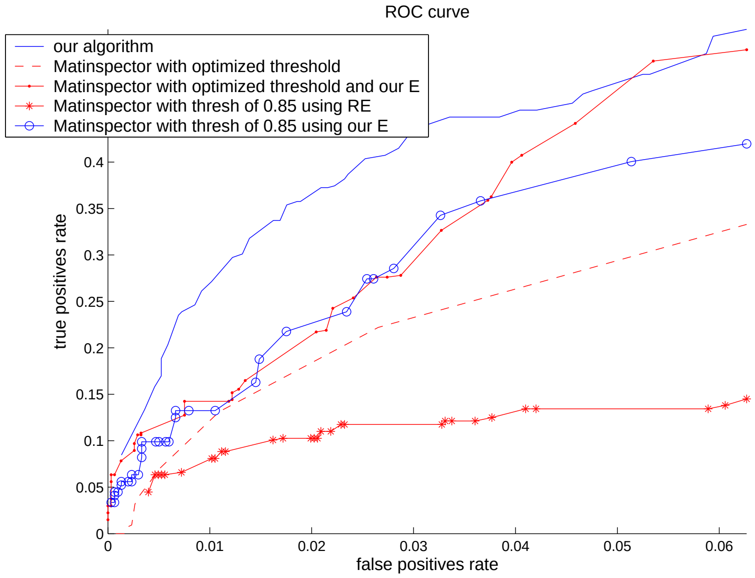


Figure 3.3: ROC curves showing ability to identify binding sites in promoters of genes from SCPD. The x-axis shows the *false positives rate* FP/(FP+TN), the y-axis shows the *true positives rate*, TP/(TP+FN). The results shown are average values on all 135 genes. The solid curve shows our results in a range of values of the FDR method parameter, $Q \in [0.01, 0.45]$ (same as in Fig. 3.2). The dashed curve shows MatInspector results using the optimized score thresholds suggested by MatInspector in a range of thresholds on the number of matches $(3 - 150)$. The solid curve with stars shows MatInspector results with a constant score threshold of 0.85 using the RE value. The range of values for the threshold of RE/(num matches) to be considered a match is $[0.0067, 0.06]$. The solid curve with circles shows MatInspector results with a constant score threshold of 0.85, using our estimation of E instead of MatInspector's RE value. The range of values for the threshold of E/(num matches) to be considered a match is $[0.0067, 0.11]$. The solid curve with points shows MatInspector results with the optimized score thresholds suggested by MatInspector, using our estimation of the mean number of matches. The range of values for the threshold of E/(num matches) to be considered a match is $[0.0067, 0.1]$.

Thus in order to get an accurate estimation of E we must compute $P(A)$ very accurately. In section 2.3.2 we describe how it is done.

3.4 Reasons for differences in performance

There is a central difference between our approach and MatInspector's. Given a promoter sequence and a PWM, we look at the best match of this PWM in the sequence, and ask what is the probability to find a match of such quality (score) in a random promoter. Intuitively we ask: 'does the TF bind the promoter region in at least one position, or not?'. MatInspector refers to the number of matches of the PWM in the sequence, where a match is defined as a position with a score higher than a certain threshold. MatInspector then uses an estimation of the mean number of matches. We believe that this difference is a main cause of our higher performance. However, it might be also due to MatInspector's choice of the optimal score thresholds (whose calculation is not published).

As was described in 3.2.2, MatInspector uses an estimation of the mean number of matches of a PWM in a random sequence, the RE value. However, the calculation of the RE value is not published. We present in 3.3 our calculation of the mean number of matches. As was shown in Fig. 3.3, using our calculation instead that of MatInspector improves significantly MatInspector's results. This suggests that we calculate the mean number of matches more accurately than MatInspector. This is another cause to MatInspector's lower performance.

3.5 Synthetic Data

We continue with evaluating our methods on synthetic data. It enables us to measure our performance on many (100) datasets, and to calculate the mean and standard deviation of our performance (which is measured as the tradeoff between true positives and false positives). It could be that the real dataset we used (SCPD) is exceptional for some reason, so in this way we get a more reliable picture of our results.

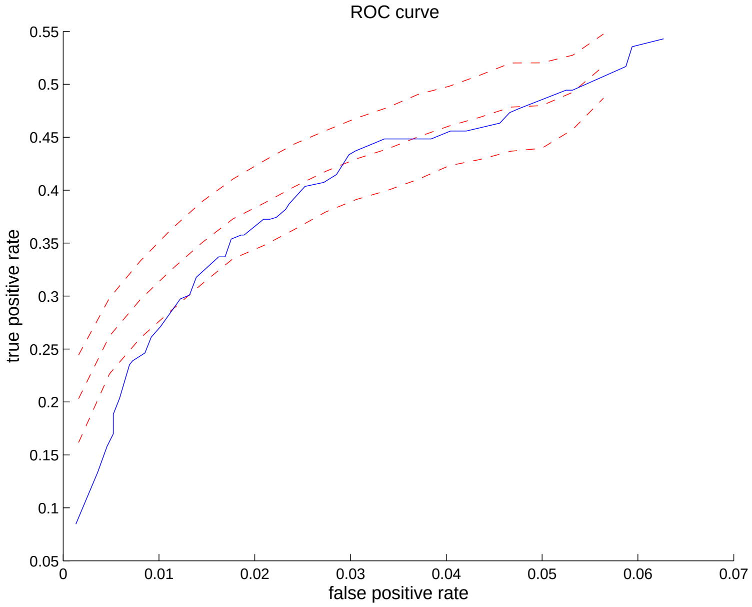


Figure 3.4: ROC curves showing ability to identify binding sites in promoters of genes from SCPD. The x-axis shows the *false positives rate* FP/(FP+TN), the y-axis shows the *true positives rate*, TP/(TP+FN). The results shown are average values on all 135 genes. The solid curve shows our results in a range of values of the FDR method parameter, $Q \in [0.01, 0.45]$. The dashed curves show the mean ± standard deviation of the 100 ROC curves of the synthetic datasets.

We built 100 datasets. Each dataset is similar in its structure to the real dataset we used (SCPD). All datasets consist of 135 promoters of length 500. The promoter sequences were sampled from a 3-order Markov model background distribution, trained on Saccharomyces Cerevisiae promoter regions. In the promoter regions we planted motifs which are known to be true binding sites of the 24 transcription factors from SCPD. Each motif was sampled uniformly from the list of known binding sites of a specific transcription factor. The number of motifs in every promoter is similar to that in the real dataset. We ran our method on these 100 datasets and compared the results to our results on the real dataset (Fig. 3.4).

It can be seen in Fig. 3.4 that our performance on the SCPD database is similar to our performance on synthetic datasets. It means that our high performance on the SCPD database is not incidental, but it reflects the power of our method. It also indicates that the SCPD database is 'normal', for example in the sense that there are not many binding sites which are not listed there (had there been, our performance on it would have been worse than that on the synthetic data).

3.6 Finding a common TF in a group of genes

Suppose we are given a set of promoters of genes, G, which are suspected to be coregulated, i.e. they might have a common transcription factor that regulates them. Let n be the size of G. Let F be a set of PSSMs of known transcription factors. Let m be the size of F. We describe here a method to search for a common transcription factor in this set of genes using our probabilistic approach.

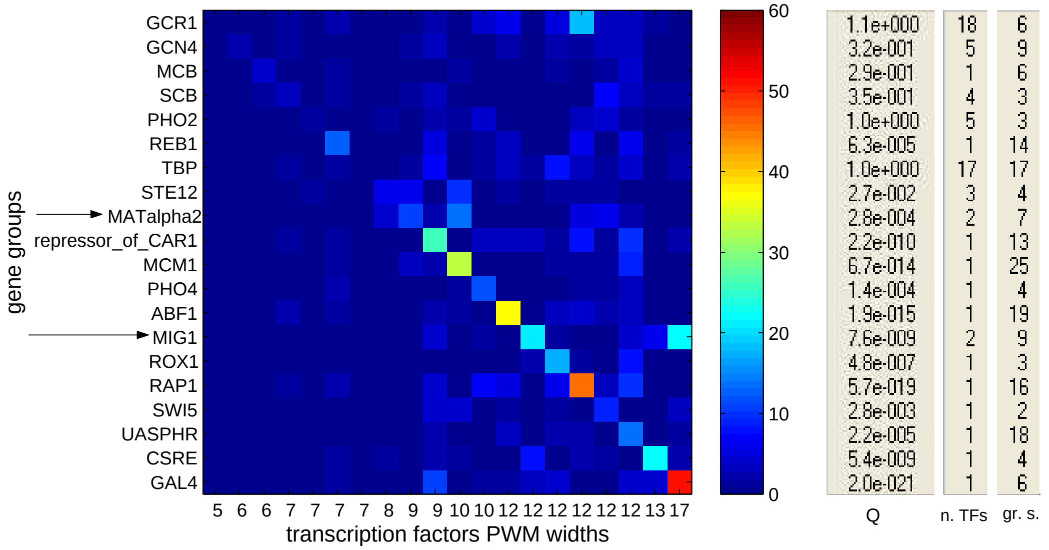
First we compute p-values for every pair of PSSM and gene. Then for every PSSM $i = 1, \ldots, m$ we have n p-values, $p_{i,1}, \ldots, p_{i,n}$. Our null hypothesis is :

$$\forall i \in \{1, \ldots, m\}. \forall j \in \{1, \ldots, n\}. p_{i,j} \sim \text{uniform}[0, 1]$$

Let $score(i) = \prod_{j=1}^{n} p_{i,j} = s_i$. This score represents the plausibility of the transcription factor to be a common regulator of the gene group. The lower the score, the more plausible it is. We calculate this score for every $M \in F$, $s_1, \ldots, s_m$. We now can compute the probability $p_i = P(score(i) \leq s_i)|$ null model) for every $M \in F$. The density function of the product of n uniform independent random variables on the interval $[0, 1]$, $x_1, \ldots, x_n$ is:

$$P_{x_1 \ldots x_n}(t) = \frac{(-1)^{n-1}}{(n-1)!}(ln(t))^{n-1}$$

In this way we get a p-value for every transcription factor for being a common regulator of the gene group. We use the FDR method on the set of m p-values to get the statistically significant transcription factors for the group of genes. If there are statistically significant transcription factors, they are suspected to be common regulators for this group of genes.



Q	n. TFs	gr. s.
1.1e+000	18	6
3.2e-001	5	9
2.9e-001	1	6
3.5e-001	4	3
1.0e+000	5	3
6.3e-005	1	14
1.0e+000	17	17
2.7e-002	3	4
2.8e-004	2	7
2.2e-010	1	13
6.7e-014	1	25
1.4e-004	1	4
1.9e-015	1	19
7.6e-009	2	9
4.8e-007	1	3
5.7e-019	1	16
2.8e-003	1	2
2.2e-005	1	18
5.4e-009	1	4
2.0e-021	1	6

Figure 3.5: Plot of $-log$(p-values), obtained for 20 transcription factors for 20 groups of genes. Each group of genes contains the known targets of one of the transcription factors. Each row corresponds to a gene group, labelled by the name of their common transcription factor. Each column corresponds to a transcription factor; the numbers under the columns are the widths of their PWMs. The columns are placed in an increasing order of the widths and the gene groups appear in the same order as the transcription factors. The Q - values that are listed give for each gene group the minimal value of the Q parameter needed in order to get the common transcription factor of this gene group as statistically significant (when using the FDR method). The numbers next to the Q - values, labelled 'n. TFs', shows the number of transcription factors that are reported as statistically significant by the FDR method when the minimal Q value is used for the corresponding gene group. The size of each gene group is shown in the column labelled by 'gr. s.'.

We applied this on groups of coregulated genes taken from SCPD. We took the group of all genes that a specific transcription factor is known to bind. There are 20 transcription factors in SCPD that have known PWM, which are also known to bind to more than one gene. Thus we work on 20 groups of coregulated genes. It can be seen in figure 3.5 that in most of the gene groups the transcription factor to which we assign the lowest p-value is their known common transcription factor (the values on the diagonal are usually the highest of all the values in a row). The gene groups for which this doesn't hold can be divided into two classes:

1. Gene groups whose known common transcription factor has a short PWM (of width between 5 and 8). When a PWM is short it can't get a very low p-value on a specific gene. For example, suppose the width of a PWM, M, is 6, the length of the gene's promoter is 500, and that the promoter contains a position with the maximal possible score for M. A rough estimation for the probability to get such a maximal score is (see 2.4.1):

   $$1 - (\frac{4^6 - 1}{4^6})^{500} = 0.11$$

   It means that the minimal p-value of this PWM on each gene is approximately 0.1, and we get that $p_{M,1}, \ldots, p_{M,n} \geq 0.11$. It can be seen in Fig. 3.5 that the left side of the square is darker from the right side (higher p-values). This is because the transcription factors are ordered in an increasing order of their PWM's widths, and the first 9 transcription factors have widths between 5 and 9, which result in higher p-values.

2. Gene groups that contain a sub-group of genes with a different additional common transcription factor. This is the case in two groups of the genes (by arrows in Fig. 3.5): the identified genes whose common regulator is MATalpha2 (all the genes but one are also regulated by the transcription factor MCM1), and the genes whose common regulator is MIG1 (a subgroup of the genes are regulated by the transcription factor GAL4).

This shows that the p-values we calculate for a transcription factor to be bound to different genes can be used to estimate the p-value for a transcription factor to be a common regulator of a group of genes. As was shown here, this also gives good results. This analysis can be also applied to clusters of co-expressed genes, resulting from microarray experiments, which are usually suspected to have common regulators.

3.6.1 Comparison with PRIMA

We compared our method with PRIMA, a program for searching for common transcription factors in a set of genes using PWMs. PRIMA uses a big background set of promoters, and compares the enrichment of the PWMs in the given set with that in the background set. Elkon et al. [6] present PRIMA's results for several clusters of genes. One of these clusters is a set of 103 human promoters corresponding to E2F target genes reported by Ren et al. [15]. PRIMA scanned this set with 107 PWMs from the TRANSFAC database. It found E2F to be significantly enriched, as well as three other transcription factors: NF-Y, CREB, and NRF. We scanned the same group of promoters with 197 PWMs from the TRANSFAC database (Wingender et al. [19]). We calculated a p-value for every transcription factor for being a common regulator of the gene group as was described in 3.6. The comparison between our results and PRIMA is presented in Table 3.1.

Transcription Factor	**PRIMA's analytical score**	**We found as statistically significant**	**Our p-value**	**Q parameter**
E2F	1.9×10^{-10}	+	8.5×10^{-6}	7.2×10^{-5}
NF-Y	1.7×10^{-14}	+	7.4×10^{-37}	4.8×10^{-35}
NRF	3.1×10^{-4}	+	7.2×10^{-5}	5.1×10^{-4}
CREB	2.5×10^{-5}	-	0.93	1.13

Table 3.1: Comparison between our results and PRIMA's, on a set of 103 human promoters corresponding to E2F target genes reported by Ren et al. [15]. PRIMA used 107 PWMs and we used 197 PWMs (both from the TRANSFAC database, Wingender et al. [19]). PRIMA found four significantly enriched PWMs. Three of them, E2F, NF-Y and NRF where found to be statistically significant also by our method. Indicated for each one are the analytical score given by PRIMA, our p-value for this transcription factor for being a common regulator, and the minimal value of the Q parameter needed in order to get this transcription factor as statistically significant by our method.

As listed in Table 3.1, we found the transcription factor E2F to be statistically significant for its target genes (with Q parameter of 7.2×10^{-5} and p-value of 8.5×10^{-6}). It means that for this set of genes our method finds the true common transcription factor. In addition, our method found the transcription factors NF-Y and NRF to be statistically significant, in accordance with PRIMA's results. This strengthen the reliability of both methods. Notice that PRIMA uses additional information, a big background promoter set, while we use the given promoter set alone. Our method didn't find the transcription factor CREB to be statistically significant. It might be because its PWM is short (its width is 8). As was described in 3.6, our method might miss transcription factors with short PWMs.

Chapter 4

Summary

In this thesis I have presented a method for searching for putative transcription factors' binding sites in promoter sequences, and estimate their statistical significance. Many of the new methodologies search for a common binding site in a group of genes (for example Elkon et al. [6], Hughes et al. [8]). Searching for a binding site in a single promoter sequence is more exposed to noise, raising the need for a subtle analysis. The accuracy of the p-values of our method enables us to perform a better search for binding sites in single promoters. Our ability to predict correctly putative binding sites is significantly higher than that of MatInspector, which also searches for binding sites in single promoters. In addition, as was shown in section 3.6, given a group of genes which are suspected to be coregulated, our method can be used to estimate the statistical significance of a transcription factor to be a common regulator of this group. Thus we can exploit the sequence information of several genes together.

This work can be extended in several directions. First, the methods we have developed can be extended to searching for multiple occurrences of a motif in a promoter sequence, with a calculation of a p-value for the number of occurrences found. Second, we have concentrated on searching for a motif of a single transcription factor. However, it is known that in many cases transcription factors cooperate with each other, and form combinatorial transcriptional regulation. Thus, an interesting direction is to search for motif combinations and calculate their statistical significance. Third, an important challenge is to integrate our method with additional data in order to improve the tradeoff between true positives and false positives . For example, it can be combined with gene expression data as was suggested in section 3.6, or gene homology data, (see, for example Loots et al. [12]). The binding sites found using our methods can be further investigated using several analysis techniques. For instance Pilpel et al. [13] use gene expression data to detect cooperation between pairs of transcription factors.

List of Figures

Bibliography

[1] S. Aerts, G. Thijs, B. Coessens, M. Staes, Y. Moreau, and B. De Moor. Toucan: deciphering the cis-regulatory logic of coregulated genes. *Nucl. Acids. Res.*, 31(6): 1753–1764, 2003.

[2] T. Bailey and C. Elkan. Unsupervised learning of multiple motifs in biopolymers using expectation maximization. *Machine Learning*, 21:51–80, 1995.

[3] Y. Benjamini and Y. Hochberg. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *J. R. Statist. Soc.*, 57(1):289–300, 1995.

[4] G. Blom and D. Thorburn. How many random digits are required until given sequences are obtained? *J. Appl. Prob.*, 19:518–531, 1982.

[5] W. Diffie and M. E. Hellman. Exhaustive cryptanalysis of the nbs data encryption standard. *IEEE Computer*, 10(6):74–84, June 1977.

[6] R. Elkon, C. Linhart, R. Sharan, R. Shamir, and Y. Shiloh. Genome-wide in silico identification of transcriptional regulators controlling the cell cycle in human cells. *Genome Res.*, 13:773–780, 2003.

[7] K. Eriksson. A summary of recursion solving techniques. http://www.math.kth.se/ bek/diskret/linrek.pdf, 1999.

[8] J. D. Hughes, P. W. Estep, S. Tavazoie, and G. M. Church. Computational identification of cis-regulatory elements associated with groups of functionally related genes in saccharomyces cerevisiae. *J. Mol. Biol.*, 296:1205–1214, 2000.

[9] J.D. Karkas, R. Rudner, and E. Chargaff. Separation of B. subtilis DNA into complementary strands II. Template functions and composition as determined by transcription with RNA polymerase. *Proc. Nat. Acad. Sci. U.S.A.*, 59:804–819, 1968.

[10] C.E. Lawrence, S.F. Altschul, M.S. Boguski, J.S. Liu, A.F. Neuwald, and J.C. Wootton. Detecting subtle sequence signals: a gibbs sampling strategy for multiple alignment. *Science*, 262:208–214, 1993.

[11] J.S. Liu, A.F. Neuwald, and C.E. Lawrence. Bayesian models for multiple local sequence alignment and gibbs sampling strategies. *J. Am. Stat. Assoc.*, 90:1156–1170, 1995.

[12] G.G. Loots, I. Ovcharenko, L. Pachter, I. Dubchak, and E.M. Rubin. rVista for comparative sequence-based discovery of functional transcription factor binding sites. *Genome Res.*, 12(5):832–839, May 2002.

[13] Y. Pilpel, P. Sudarsanam, and G.M. Church. Identifying regulatory networks by combinatorial analysis of promoter elements. *Nature Genet.*, 29(2):153–159, Oct 2001.

[14] K. Quandt, K. Frech, H. Karas, E. Wingender, and T. Werner. MatInd and MatInspector: new fast and versatile tools for detection of consensus matches in nucleotide sequence data. *Nucl. Acids. Res.*, 23:4878–4884, 1995.

[15] B. Ren, H. Cam, Y. Takahashi, T. Volkert, J. Terragni, R.A. Young, and B.D. Dynlacht. E2F intergrates cell cycle progression with dna repair, replication , and G_2/M checkpoints. *Genes and Dev.*, 16:245–256, 2002.

[16] S. Robin and J. J. Daudin. Exact distribution of word occurrences in a random sequence of letters. *J. Appl. Prob.*, 36:179–193, 1999.

[17] Rodger Staden. Methods for calculating the probabilities of finding patterns in sequences. *CABIOS*, 5(2), 1989.

[18] G. Thijs, K. Marchel, M. Lescot, S. Rombauts, B. De Moor, P. Rouze, and Y. Moreau. A gibbs sampling method to detect overrepresented motifs in the upstream regions of coexpressed genes. *J. Comp. Biol.*, 9(2):447–464, 2002.

[19] E. Wingender, X. Chen, E. Fricke, 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. *Nucl. Acids. Res.*, 29:281–283, 2001.

[20] J. Zhu and M. Q. Zhang. SCPD: A promoter database of yeast saccharomyces cerevisiae. *Bioinformatics*, 15:607–611, 1999.