

Lecture 9: May 20, 2004

*Lecturer: Ron Shamir**Scribe: Eran Balan and Maayan Goldstein¹*

9.1 Promoter Analysis

9.1.1 Introduction to Promoter Analysis

As we studied in our first lecture, each cell contains a copy of the whole genome. But we have many tissues that are constructed of different cells, that are responsible for various tasks. Thus, each cell utilizes only a subset of its genes. Most genes are highly regulated their expression is limited to specific tissues, developmental stages, physiological condition. What we would like to find out is how the expression of genes is regulated.

Regulation of genes can be done in different stages of the gene expression. The process of gene expression is regulated at multiple points including chromatin modifications (during the process of DNA packaging), transcription control (our focus here), splicing, transport and translation control. Biological regulation have more to it than just gene expression regulation, for example, protein interactions and post-translational modification are extremely important in many processes that we are not dealing with here. The most common way of regulation is called transcriptional regulation, which will be the main issue discussed in the lecture. It is done during the transcription phase, when the DNA is transcribed into preRNA.

9.1.2 Regulation of Transcription

The regulation of the transcription of a gene is mainly encoded in the DNA in a region called *promoter*. Each promoter contains several short DNA subsequences, called *binding sites (BSs)* that are specifically bound by regulatory proteins called *transcription factors (TFs)* (see Figure 9.1). Transcription factors typically combines to form "transcriptional switches" that encode complex logical functionality to control gene expression given a multitude of biological stimuli.

Transcription factors are proteins that bind to DNA region near the gene (the promoter region) and regulate its transcription. They attach to the DNA at specific binding sites. Transcription factors work in combinations forming complex logical schemes.

¹Lecture Notes from 20.05.04.

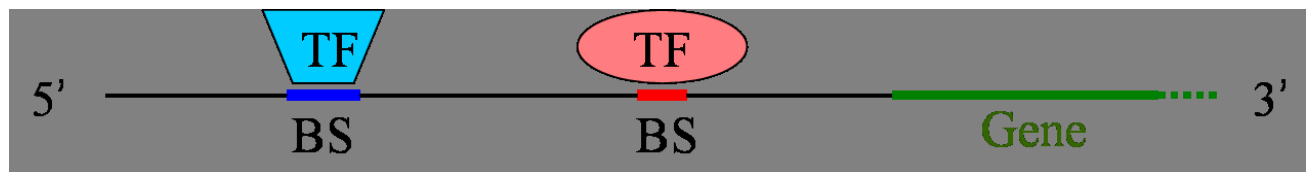


Figure 9.1: Binding sites.

An example of transcriptional switch is shown in figure 9.2. The regulatory role of the E2F transcription factor is facilitated via its sequence specific binding site. However, binding can be suppressed if a second regulatory protein called Rb is binding E2F. Moreover, Rb effect on E2F binding can be blocked by a third protein, called E7, and only in the presence of E7, transcription can take place. Figure 9.3 gives us a 3D picture of what is happening during the transcription factors attachment.

9.1.3 Cis and Trans Regulation

DNA sequence that acts to change the expression of the gene adjacent to it is *cis-acting*. A *trans-acting* element acts to change the expression of the gene at a distance. Promoter elements are cis acting. Sequence controlling the expression of the TF itself is trans acting. This lecture will focus on analysis of cis-acting regulatory elements.

9.1.4 Regulation of Transcription

By binding to a genes promoter, TFs can either induce or repress the recruitment of the transcription machinery. The conditions in which a gene is transcribed are determined by the specific combination of BSs in its promoter. A good example of this process is shown in figure 9.4 where a number of binding sites present, while the Tata protein is attached to one of them.

One of the ways to study promoter analysis is by analyzing the expression levels of RNA. The assumption is that genes that have similar expression levels, have similar transcriptional regulation control and common binding sites. Thus, we can use the knowledge we have on genom sequences in humans (or other species) in order to find promotor regions. In order to find binding sites in those regions we could use the various methods of dealing with DNA chips.

9.1.5 Promoter Region

The first thing we would like to define is how to find the *promoter region* in the DNA sequence. The 5-end region of a gene is very likely to overlap with the genes promoter region.

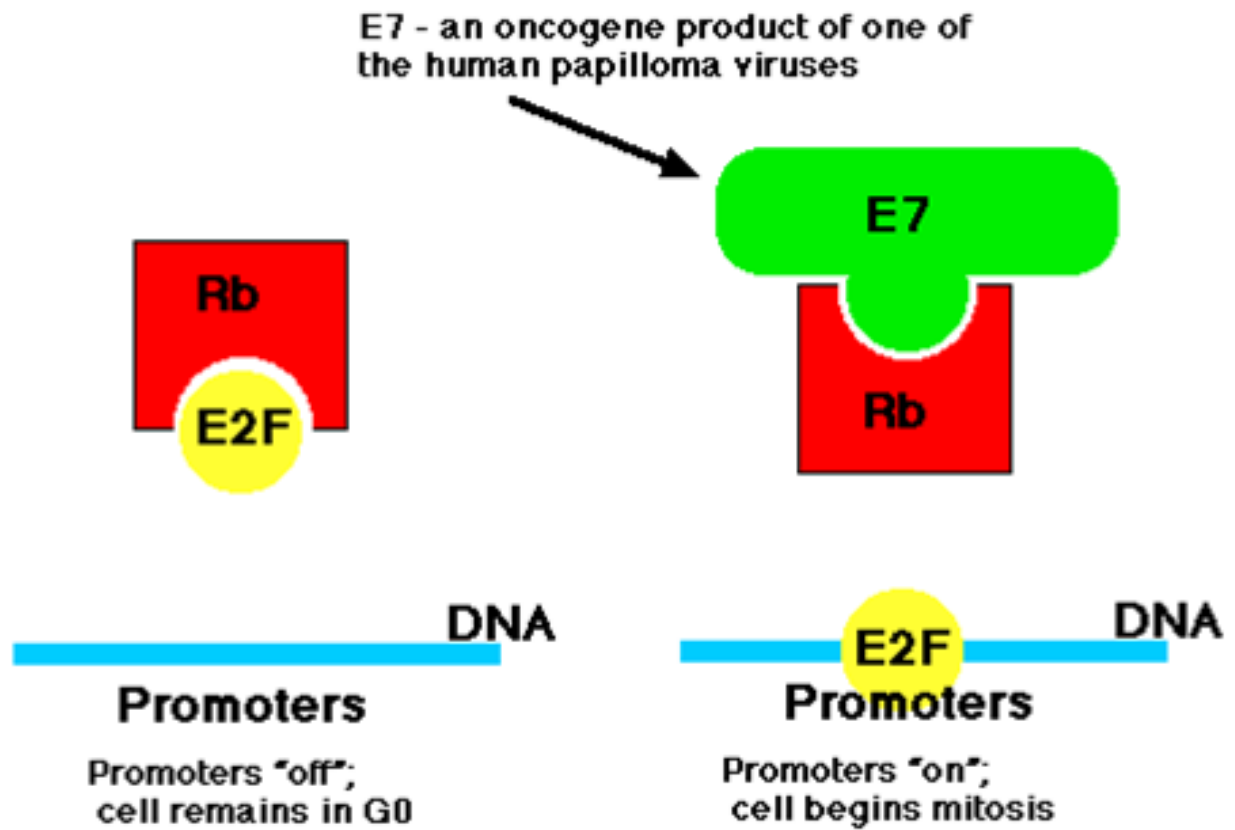


Figure 9.2: Source: [6] Regulation with E2F transcription factor.

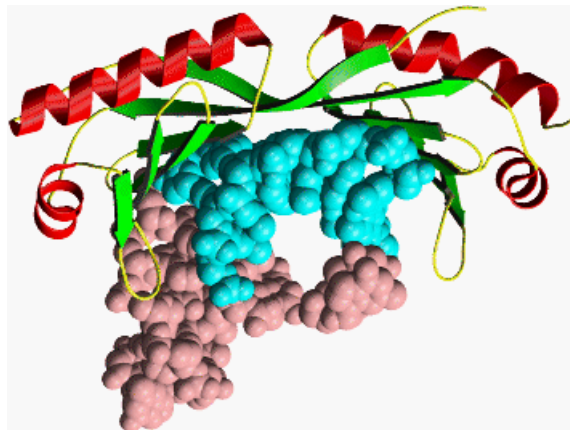


Figure 9.3: Source: [6] 3D Regulation Structure.

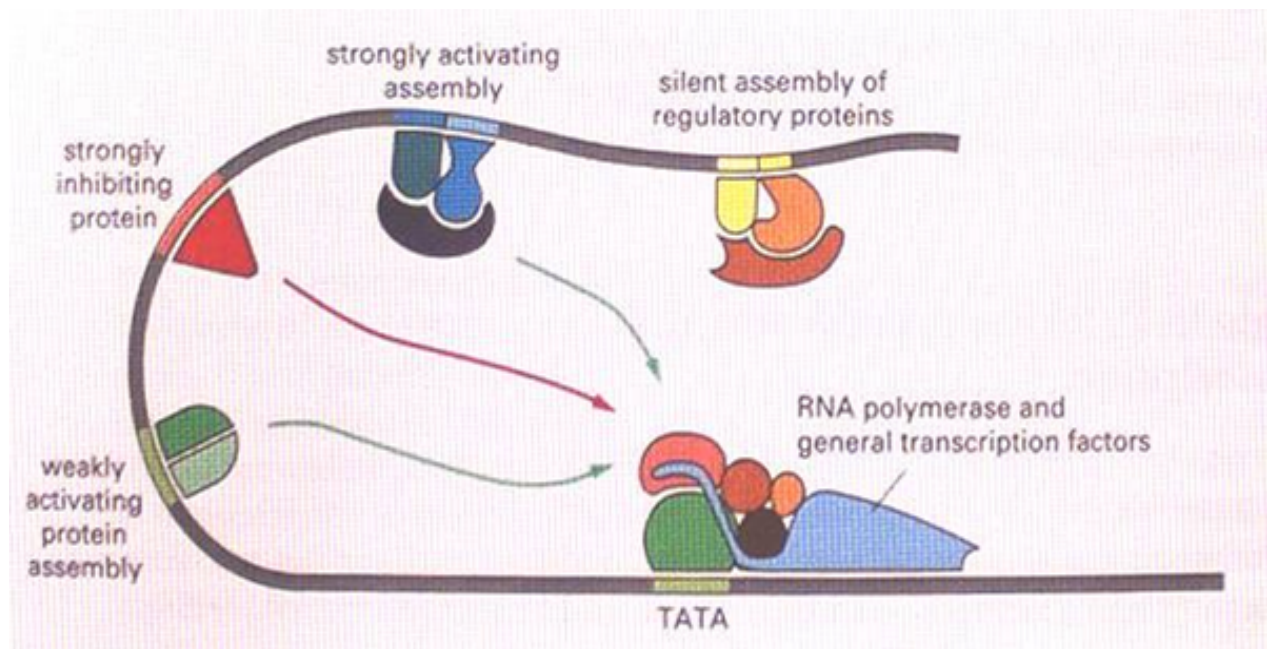


Figure 9.4: Regulation of Transcription.

Promoters are stretches of DNA sequences, generally located upstream of and overlapping the transcription start site (TSS) of genes. The promoter region is the main regulatory region for the expression of a gene. Thus, we will deal with Upstream Transcription Start Site (TSS), meaning that promoter region appears before the transcribed area.

While looking for binding sites we would like to consider two options: too short TSS will miss many real BSs (false negatives) while too long TSS will have lots of wrong hits (false positives). Usually, the length of TSS is species dependent (e.g., yeast 600bp, thousands in human), while the common practice is to use 500-2000bp. Also, experience show us that we should analyze both strands of the DNA. We would also like to mask-out repetitive sequences. Most of these sequences infiltrated the DNA during the evolution process and are not significant for the transcription process.

9.1.6 Models for finding Binding Sites

We shall consider a number of models: exact string model, string mismatches model, degenerate string model and, finally, position weight matrix.

Exact String model The *Exact String model* will try to find an exact sequence in the DNA sequence as shown in the example:

Example:

BS = TACACC , TACGGC

CAATGCAGGATACACCGATCGGTA

GGAGTACGGCAAGTCCCCATGTGA

AGGCTGGACCAGACTCTACACCTA

Figure 9.5: Exact String.

String Mismatches model The *String Mismatches model* will try to find an almost the exact sequence and will tolerate a mistake in one of the positions as shown in the example:

Example:

BS = TACACC + 1 mismatch

CAATGCAGGATTACACCGATCGGTA

GGAGTACAGCAAGTCCCCATGTGA

AGGCTGGACCAGACTCTACACCTA

Figure 9.6: String Mismatch.

Degenerate String model The *Degenerate String model*, also known as *consensus model* will try to find a sequence, but allows various bases to be placed in specific positions of the sequence. In the example, positions 3,4 of the sequence could be represented by two or three bases. This gives us 6 possible string to search for.

Example:

BS = TASDAC (S={C, G} D={A, G, T})

CAATGCAGGATACAACGATCGGTA

GGAGTAGTACAAGTCCCCATGTGA

AGGCTGGACCAGACTCTACGACTA

Figure 9.7: Degenerate String.

Position Weight Matrix model The *Position Weight Matrix model*, also known as *Position Specific Scoring Matrix model* will create a matrix, where each column represents

a position and each row represents a base and the value in the cell is the probability of the base to appear in the specified position. When scanning the target, we compute the total probability, while we assume that appearances of each base at any position are statistically independent. As shown in the example, we compute various scores and choose those with the higher scores (above predefined threshold) - higher probability.

A	0.1	0.8	0	0.7	0.2	0
C	0	0.1	0.5	0.1	0.4	0.6
G	0	0	0.5	0.1	0.4	0.1
T	0.9	0.1	0	0.1	0	0.3

ATGCAGGATACACCGATCGGTA	0.0605
GGAGTAGAGCAAGTCCCGTGA	0.0605
AAGACTCTACAATTATGGCGT	0.0151

Figure 9.8: Degenerate String.

There are also more complex models such as *PWM with spacers*, *Markov model* (dependency between adjacent columns of PWM), *Hybrid models*, e.g., mixture of two PWMs and more.

9.2 Technology

In this section we will present some of the common technologies used for promoter analysis.

9.2.1 Identifying regulatory elements

In this method we use a DNA fragment containing potential regulatory sequences (light color), such as the region upstream from a regulated gene, is cloned next to a reporter gene (dark color) encoding an easily assayed protein (for example, beta-galactosidase), see figure 9.9. The construct is put into cells and regulation is monitored by the activity of the reporter gene.

Identifying DNA regulatory sequences

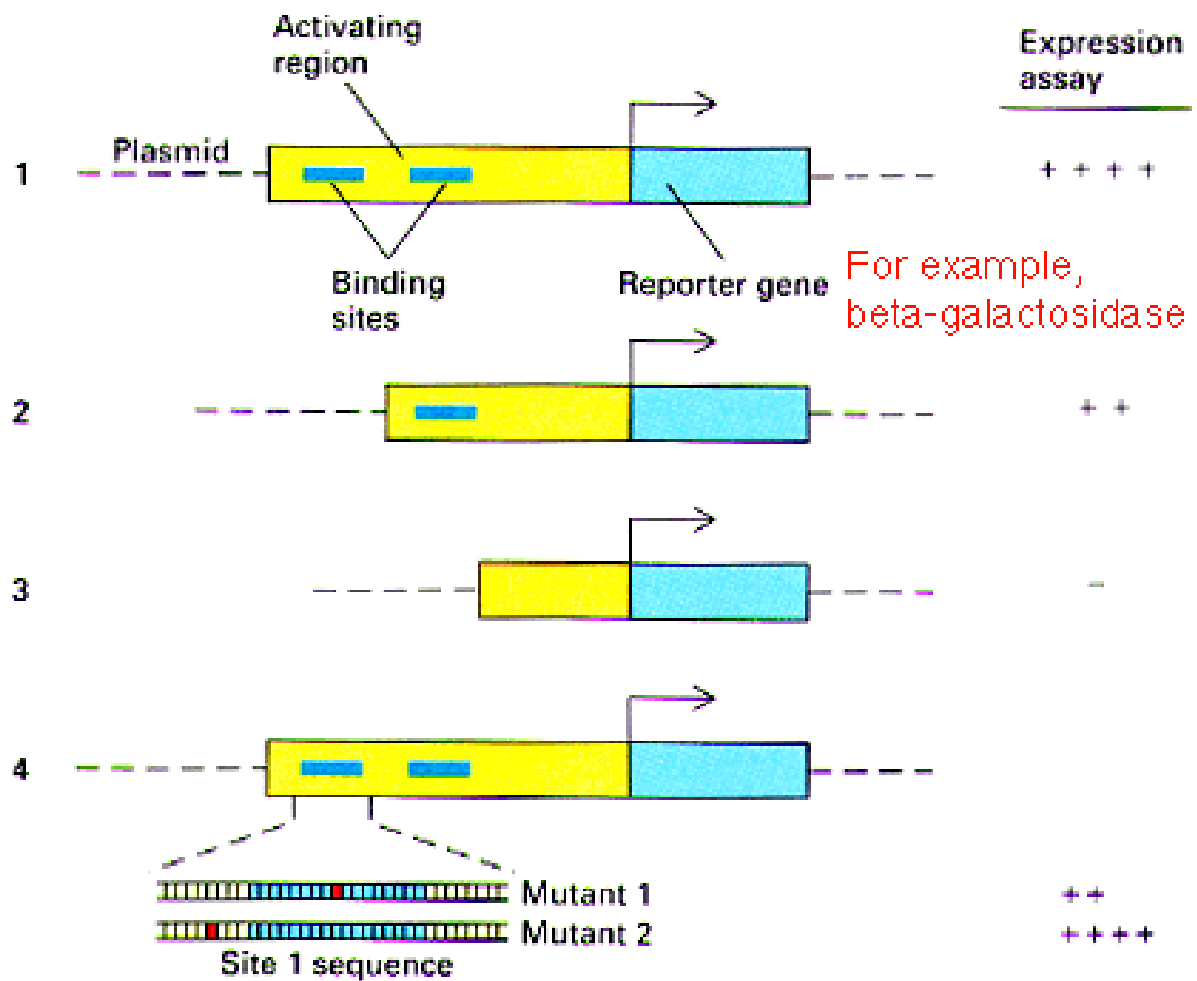


Figure 9.9: Source: [4]Identifying regulatory elements.

9.2.2 Chromatin Immunoprecipitation (ChIP)

A procedure that identifies DNA elements occupied by DNA regulatory proteins in vivo under a given set of conditions. Briefly, proteins are covalently cross-linked to DNA in living cells, the cells are lysed, and DNA is fragmented via sonication. Antibodies to the binding protein can then be used to immunoprecipitate the protein-DNA complex. This technique provides a method of purifying the regulatory regions of the DNA bound to the protein at the time of cross-linking. The purified DNA can be amplified and sequence information can be obtained. (see [11])

Figure 9.10 shows the complex ChIP process. It includes the following stages: freezing the current chemical stage, shearing the desired proteins and then to replicate these part of the DNA by the PCR process.

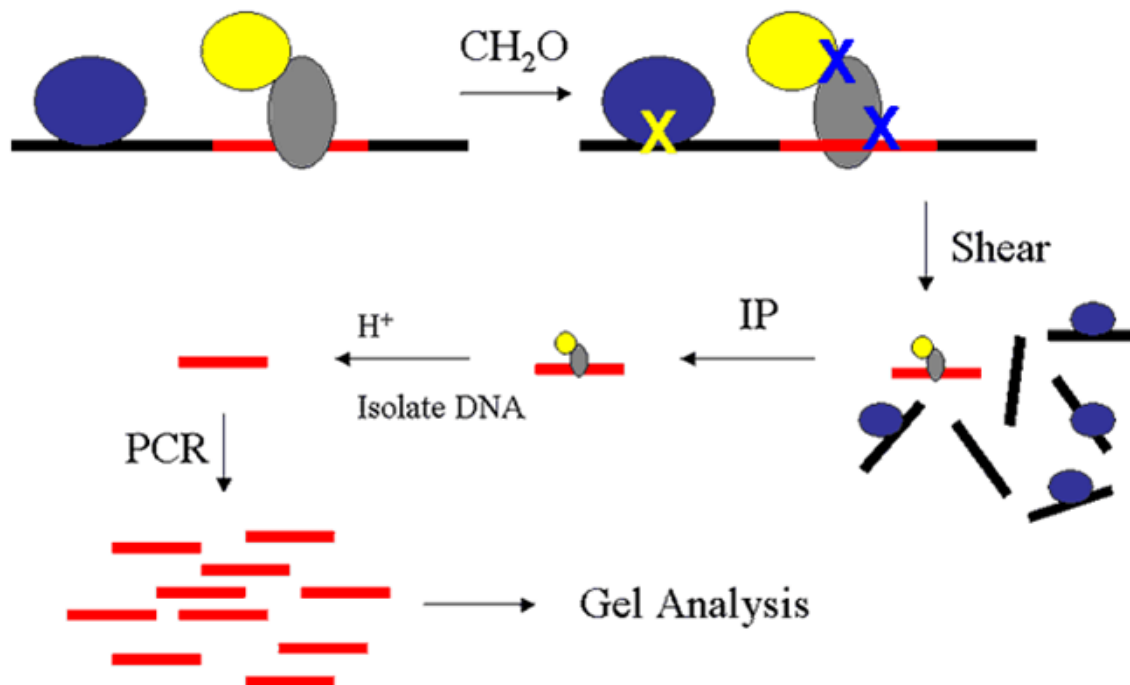


Figure 9.10: Source: [5]Chromatin Immunoprecipitation.

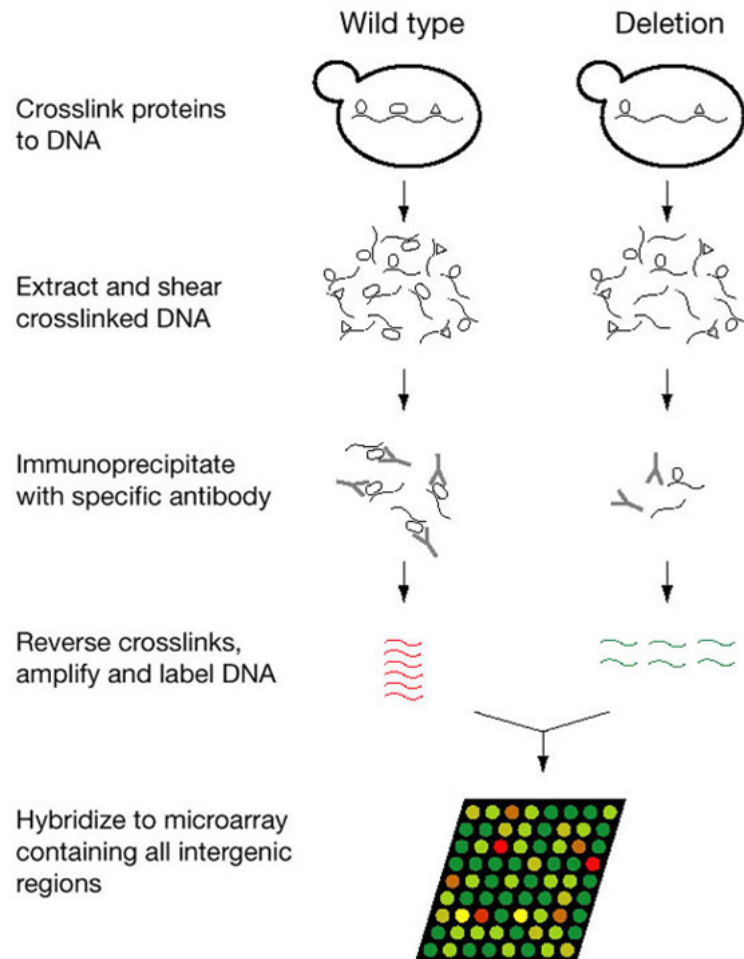


Figure 9.11: Source: [2]Strategy for analyzing genome-wide protein-DNA interactions. The reference probe can either consist of DNA generated in parallel from a strain bearing a deletion of the gene encoding the protein of interest (as depicted), or of unfractionated genomic DNA amplified and labelled in the same manner. Alternatively, an epitope-tagged version of the protein of interest can be immunoprecipitated with an antibody directed against the epitope. The DNA microarray includes all of the intergenic regions or promoters from the genome. The Cy5/Cy3 fluorescence ratio for each locus reflects its enrichment by immunoprecipitation (IP) and therefore, in general, its relative occupancy by the cognate protein.

9.2.3 Location analysis

The genome-wide location analysis method allows protein-DNA interactions to be monitored across the entire yeast genome. The method combines a modified chromatin immunoprecipitation (ChIP) procedure, which has been previously used to study protein-DNA interactions at a small number of specific DNA sites, with DNA microarray analysis (see [1])

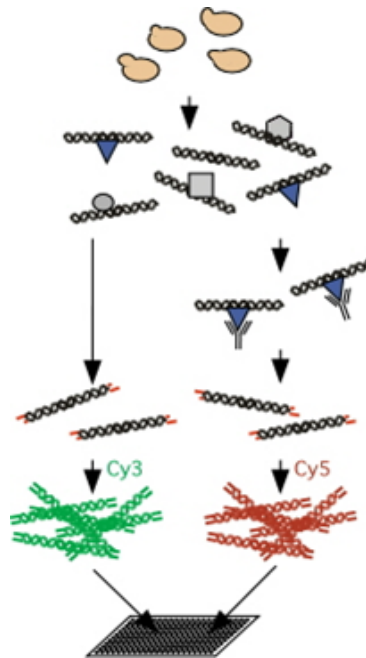


Figure 9.12: Source: [9]Location analysis.

9.3 Computational approaches to promoter analysis

In this section, we will present various techniques to find binding sites in groups of promoters. We can divide the promoter analysis computational problem into three strategies:

- Given groups of co-regulated genes and known binding sites models (PWMs) find enriched Cis elements in the groups, for instance, using PRIMA algorithm.
- Given a set of binding site models (PWMs) find CRM (cis-regulatory-modules) which are sets of binding sites that tends to cluster together, for instance, using CREME algorithm.

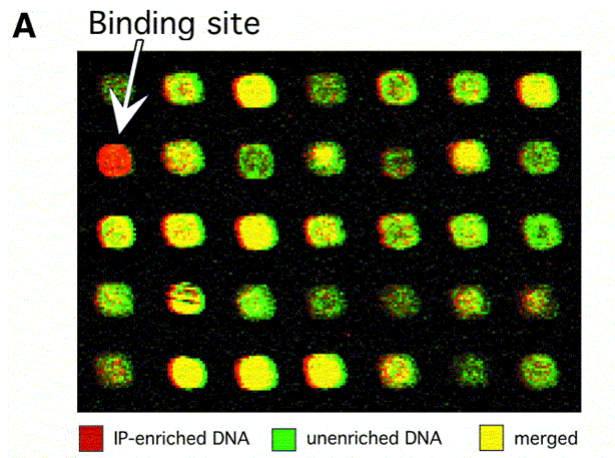


Figure 9.13: Source: [9]Close-up of a scanned image of a microarray containing DNA fragments representing 6361 intergenic regions of the yeast genome. The arrow points to a spot where the red intensity is over-represented, identifying a region bound in vivo by the protein under investigation.

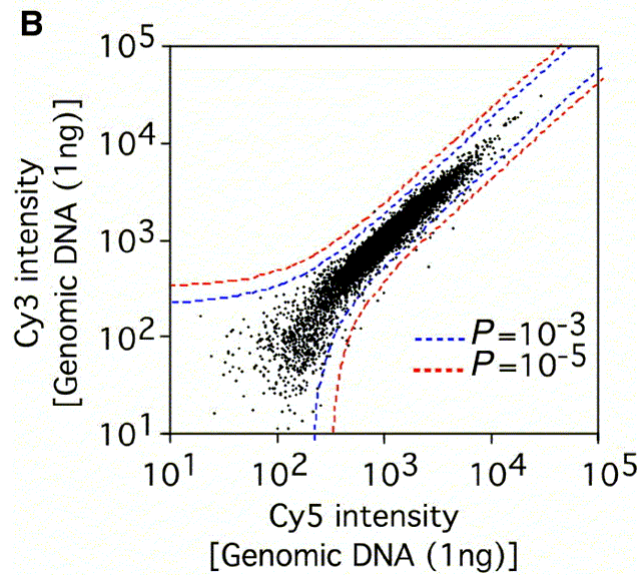


Figure 9.14: Source: [9]Analysis of Cy3- and Cy5-labeled DNA amplified from 1ng of yeast genomic DNA using a single-array error model (8). The error model cutoffs for P values equal to 10^{-3} and 10^{-5} are displayed.

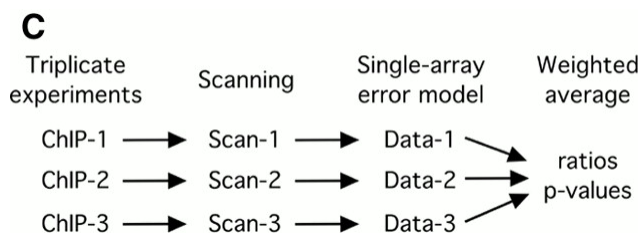


Figure 9.15: Source: [9]Experimental design. For each factor, three independent experiments were performed and each of the three samples were analyzed individually using a single-array error model. The average binding ratio and associated P value from the triplicate experiments were calculated using a weighted average analysis method.

- Given a set of co-regulated genes (from gene expression clustering) or putative targets of a TF (from chip-ChIP) build motif models that are enriched in the sets. We will show two algorithms to solve this problem: Random Projections and Meme algorithms.

9.3.1 PRIMA

PRIMA (PRomoter Integration in Microarray Analysis) is a program for finding transcription factors (TFs) whose binding sites are enriched in a given set of promoters. PRIMA is typically used for the analysis of large-scale gene expression data. Microarray ('DNA chip') measurements point to alterations in gene expression levels under varying biological conditions, but they do not, however, directly reveal the transcriptional networks that underlie the observed transcriptional modulations. PRIMA is aimed at the identification of TFs that take part in these networks. The basic biological assumption is that genes that are co-expressed over multiple biological conditions are regulated by common TFs, and therefore are expected to share common regulatory elements in their promoters. By utilizing human genomic sequences and models for binding sites (BSs) of known TFs, PRIMA identifies TFs whose BSs are significantly over-represented in a given set of promoters. (see [7])

The algorithm: receives as its input: a target set (e.g., a list of co-expressed genes found in a microarray experiment) and a background set (e.g., the 13K set) and PWMs of known TFs. Its output is: p-values of enriched TFs

For each PWM:

- Compute a threshold score for declaring hits of the PWM (hit = subsequence that is similar to the PWM = hypothetical BS)
- Scan BG and target-set promoters for hits.

- Compute enrichment score to decide whether the number of hits in the target-set is significantly higher than expected by chance, given the distribution of hits in the BG. (Synergism test: Find co-occurring pairs of TFs)

In order to identify putative binding sites, or hits, of a TF, a threshold $T(P)$ for the similarity score of the TFs PWM P is determined. Subsequences with a similarity score above $T(P)$ are regarded as hits of P . The threshold for each PWM is computed as follows:

- Compute 2nd-order Markov-Model of BG seqs.
- Generate random seqs using MM (e.g., 1,000 seqs of length 1,000 bp)
- Set threshold s.t. PWM has f hits in the random sequences (e.g., $f=100$)

This ensures a pre-defined false-positives rate, but no guarantee on false-negatives rate. Estimating false-negatives (positives) rate requires good positive (negative) training-sets. The enrichment score is computed as follows: Suppose each promoter has 0 or 1 hits.

Let: B = # of BG promoters

T = # of target-set promoters

b = # of hits in BG promoters

t = # of hits in target-set promoters

Then: Prob. for t hits in target-set:

$$P(t) = \binom{b}{t} \binom{B-b}{T-t} / \binom{B}{T}$$

Prob. for at least t hits:

$$p - value = \sum_{i=t}^{\min\{b,T\}} P(i)$$

Now, we would like to take into account more than 1 hit per promoter. The reason for this is that sometimes there is a number of BSs that are supposed to encourage the transcription. It increases the possibility of getting a hit.

So, we will take into account up to 3 hits per promoter.

Let: B, T = # of promoters in BG, target-set

b_1, b_2, b_3 = # of BG promoters with 1,2,3 hits

t = total # of hits in target-set

Then: Prob. for at least t hits: (HG score)

$$p - value = \frac{\sum \binom{b_1}{i} \binom{b_2}{j} \binom{b_3}{k} \binom{B-b_1-b_2-b_3}{T-i-j-k}}{\binom{B}{T}}, i + 2j + 3k \geq t$$

Synergism score: Find pairs of TFs that tend to occur in the same promoters

Let: $T = \#$ of promoters in target-set

$t_1, t_2 = \#$ of promoters with 1+ hits of TF 1,2

$t_{12} = \#$ of promoters with 1+ hits of both TFs (w/o overlaps!)

Then: Prob. for co-occurrence of at least t_{12} :

$$\text{synergism} - p - \text{value} = \frac{\sum \binom{t_1}{i} \binom{T - t_1}{t_2 - i}}{\binom{T}{t_2}}, i \geq t_{12}$$

PRIMA results on HCC: Whitfield et al. (Whitfield et al. 2002) partitioned the cell cycle-regulated genes according to their expression periodicity patterns into five clusters corresponding to different phases of the cell cycle. When the promoter sequences of these clusters were scanned for enriched PWMs, two PWMs were enriched in a specific phase cluster, but not in the 568 set as a whole. The results of the experiment are presented in figures 16-18.

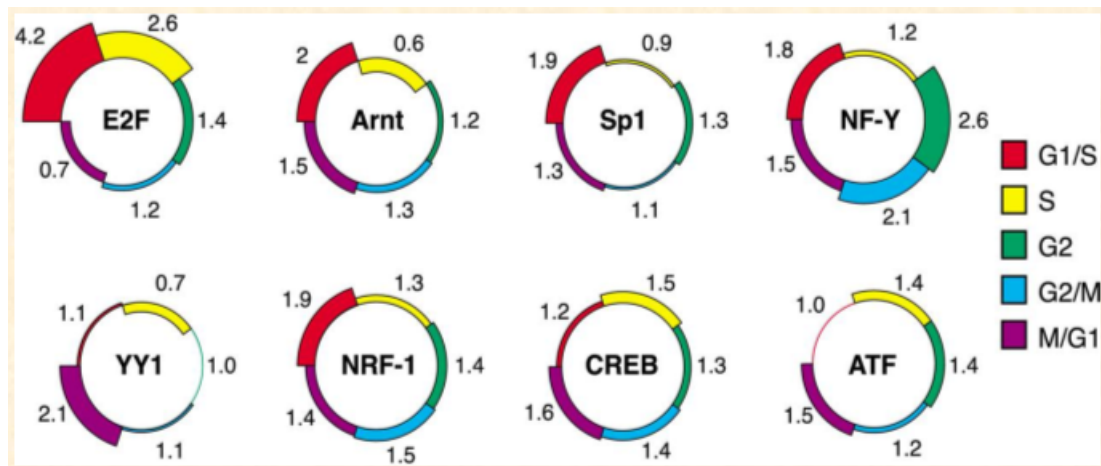


Figure 9.16: Source: [8]Representation of TF PWMs in the cell cycle phase clusters. The eight circles correspond to the PWMs that were highly enriched in promoters of cell cycle-regulated genes. Each circle is divided into 5 zones, corresponding to the phase clusters. The number adjacent to the zone represents the ratio of its prevalence in promoters contained in each of the cell cycle phase clusters to its prevalence in the set of 13K background promoters. Note that several TFs show a tendency towards specific cell cycle phases: e.g., over-representation of the E2F PWM in promoters of the G1/S and S clusters, and its under-representation in promoters of the M/G1 cluster.

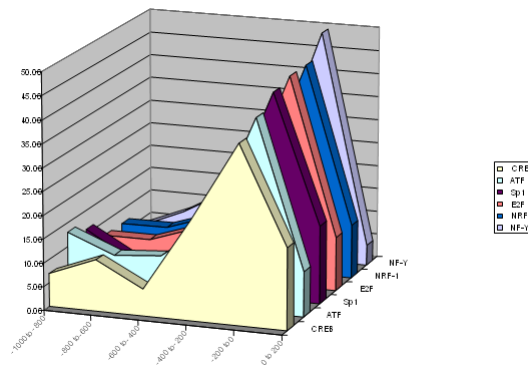


Figure 9.17: Source: [8] Distribution of locations of TFs putative binding sites found in 568 cell cycleregulated promoters. Promoters were divided into six intervals, 200 bp each. For each of the PWMs, the number of times its computationally identified binding sites appeared in each interval was counted (after accounting for the actual number of bps scanned in each interval. This number changes as the masked sequences are not uniformly distributed among the six intervals). Locations of NRF-1, CREB, NF-Y, Sp1, ATF and E2F binding sites tend to concentrate in the vicinity of the TSSs (chi-square test, p less than 0.01).

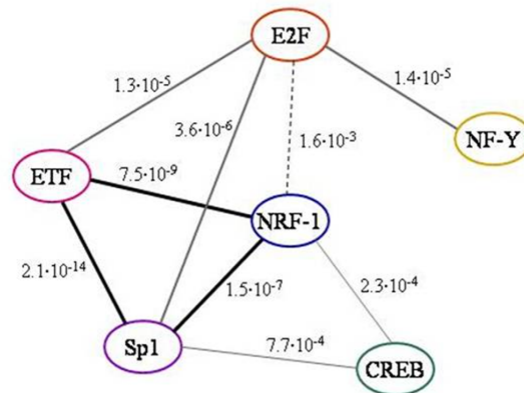


Figure 9.18: Source: [8] Pairs of PWMs that co-occur significantly in promoters of genes regulated in a cell cycle manner. It was examined whether the PWMs can be organized into regulatory modules. For each possible pair formed by these PWMs, we tested whether the prevalence of cell cycle-regulated promoters that contain hits for both PWMs is significantly higher than would be expected if the PWMs occurred independently. Eight significant pairs were identified, each connected by an edge. The corresponding p -value is indicated next to the edge. The edge connecting the E2F-NRF1 pair is dashed to indicate that its significance is borderline.

PRIMA future directions: Possible improvements to the algorithm could be in several aspects. First, choice of the region to scan within the promoters could be improved. Finding strand bias could improve normalization. In addition to that, more complex BSs models could be used. The enrichment score could also be improved (by using other scores), since as presented, it is problematic when promoters are of different lengths. Synergism can take into account distance between hits and we could find synergism of multiple transcription factors.

9.3.2 CREME - Cis-Regulatory Module Explorer

Abstract: Eukaryotic genes are often regulated by several transcription factors, whose binding sites are spatially clustered and form cis-regulatory modules. CREME is a web-server for identifying and visualizing cis-regulatory modules in the promoter regions of a given set of potentially co-regulated genes. CREME relies on a database of putative transcription factor binding sites (TFBS) that have been carefully annotated across the human genome using evolutionary conservation with the mouse and rat genomes. An efficient search algorithm is applied to this data set to identify combinations of transcription factors, whose binding sites tend to co-occur in close proximity within the promoter regions of the input gene set. These combinations are statistically evaluated, and significant combinations are reported and visualized. (see [3])

Goal: Discover modules = groups of TFs whose BSs are abundant and tend to co-occur in close proximity in promoters of co-expressed genes. The main characteristics of these modules are: limited knowledge of TFs, uses PWMs to model BSs, ignores order of TFs within the module, does not take into account multiple hits per TF.

Module = Set of PWMs

r = # of PWMs in the module

Instance of a module = A set of hits, at least one per PWM in the module, that occur in a short interval in a promoter

w = length of interval

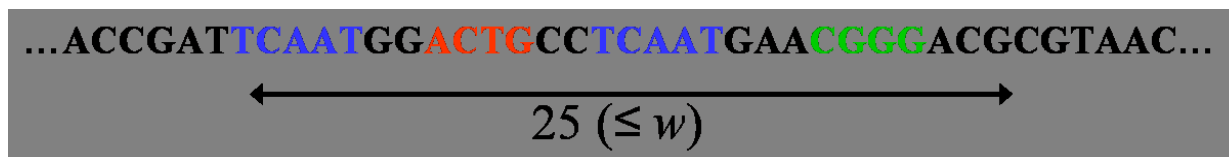


Figure 9.19: Example: Instance of a ($r=3, w=30$)- module.

The algorithm receives as its input: promoter sequences of BG and target sets PWMs of known TFs, module parameters (r, w). The output of the algorithm is p-values of enriched

modules.

The algorithm:

- Find enriched PWMs (p less than 0.01).
- Filter similar PWMs (more than 50% overlapping hits).
- Build a list of all (r,w)-modules that have instances in the target-set.
- Compute Monte-Carlo enrichment score of each module (given enrichment of PWMs), and pass those with p less than 0.05.
- Filter similar modules (more than 75% overlapping instances).

If we look closely at the third step of the algorithm, we shall see that if $n = \#$ of given PWMs then there are n^r possible modules. We'll check only those that actually have (one or more) instances in the target-set.

Simplification (not required): Search for modules with a consecutive instance = a promoter interval that contains 1+ hits for each PWM in the module, and no hits for other PWMs

Finding modules with a consecutive instance in a promoter sequence using a hashing algorithm.:

Let: M = list of all hits, ordered by position. We shall build a hash C of modules C_{open} = a hash of active modules and their starting positions

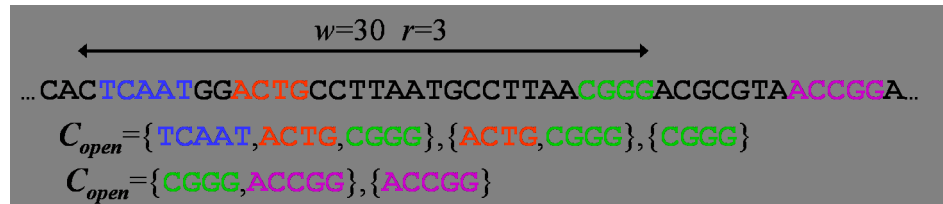


Figure 9.20: Example: Instance of a (r=3,w=30)- module and possible instances of Copen.

The details of the algorithm are shown in figure 9.21

The running time of the algorithm is $O(r|M|)$ since C_{open} contains at most r modules.

```

 $\mathcal{C} \leftarrow \emptyset$  # A hash of motif clusters whose keys are motif sets.
 $C_{open} \leftarrow \emptyset$  # A hash of active clusters and their starting positions.
For  $i = 1$  to  $|\mathcal{M}|$  do:
    Let  $h$  be the  $i$ -th hit in  $\mathcal{M}$  occurring at position  $pos(h)$ .
    For every  $(C, start) \in C_{open}$  do:
        If  $(pos(h) - start \geq w$  or  $h \notin C)$  then  $Insert(\mathcal{C}, C); Delete(C_{open}, C)$ .
        If  $(h \notin C$  and  $|C| < r)$  then  $Insert(C_{open}, (C \cup \{h\}, start))$ .
        If  $C = \{h\}$  then  $start \leftarrow pos(h)$ .
    If  $\{h\} \notin C_{open}$  then  $Insert(C_{open}, (\{h\}, pos(h)))$ .
For every  $C \in C_{open}$  do:  $Insert(\mathcal{C}, C)$  # Add remaining active clusters.
Output  $\mathcal{C}$ .

```

Figure 9.21: Source: [10] An algorithm for identifying all motif clusters with at least one consecutive instance in a given sequence. Procedures $Insert(H, e)$ and $Delete(H, e)$ insert/delete an element from a hash table H .

9.3.3 Motif Finding Tools

Motif(l, d) = string M of length l that appears in many of the given promoters, each occurrence contains (exactly) d mismatches. For example, the string "CATA" is a (4,1)-motif in: AGGCCTAGGTG , GTAAACATGAAG , ACCAGAGAG.

Our Goal is given a set of t promoters, and l, d , find the (l, d)-motif(s) that appear in at least t of the promoters.

Random Projection

Main idea of the algorithm is: choose a projection $h : 4^l \rightarrow 4^k$, hash each l -mer x in the input sequence to its bucket $h(x)$. $h(x)$ is constructed by choosing k (out of l) positions at random. Many instances of the motif are likely to fall into the same bucket = motif bucket, thus buckets with large count are likely to correspond to a motif.

The algorithm : Run m iterations:

- Choose a random projection h .
- Scan promoters using h and fill buckets.
- For each bucket with count larger than s , try to recover motif using an iterative refinement procedure.

Figure 9.22 shows an example of random projection, where $l=5$, $d=1$, $k=3$, motif: $M=CATAG$, $h(x_1x_2x_3x_4x_5)=x_1x_2x_5$, The motif bucket is CAG. In the example, we can use any base for x_3 and x_4 and we look at all the sub-sequences that fall into the same bucket. And we find x_3 and x_4 according to the most frequent sub-sequences.

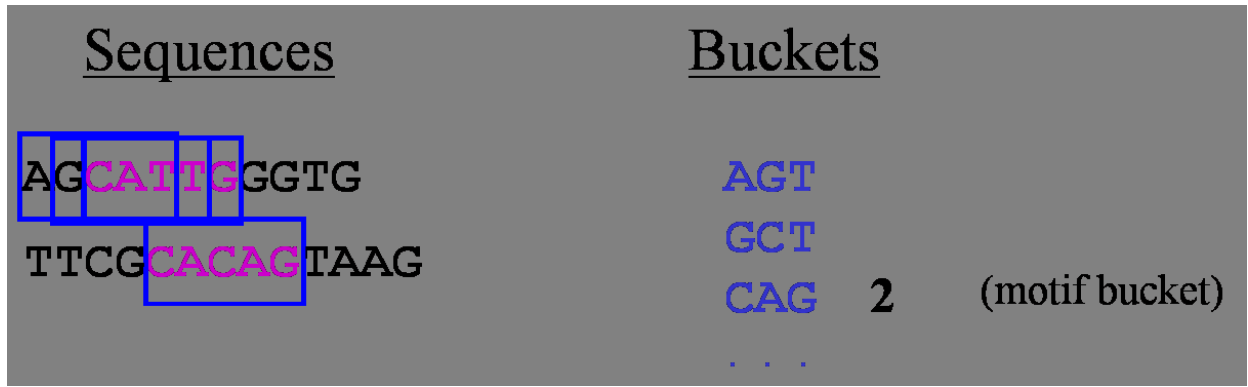


Figure 9.22: Random Projection

Analysis : Choosing of k and s can be very important, since for larger k values we get more buckets, but in every one of them there more true sub-sequence values. When k is small, we get less buckets, but in every one of them there are more false positives.

Good values for k and s are as follows: $k = l - d - 1$ (to keep average bucket size small)
 $s = 2t(L - l + 1)/4^k$

where L = average promoter length.

The probability that a motif instance hashes to the motif bucket is:

$$B(\alpha, s, t') = \sum \binom{t'}{i} \alpha^i (1 - \alpha)^{t'-i}, 0 \leq i < s$$

since $l-d$ known positions define a bucket.

The probability that fewer than s (out of t) motif instances hash to the motif bucket (in a single iteration):

$$B(\alpha, s, t') = \sum \binom{t'}{i} \alpha^i (1 - \alpha)^{t'-i}, 0 \leq i < s$$

The probability that s or more motif instances hash to the motif bucket in at least 1 (out of m) iteration:

$$1 - (B(\alpha, s, t'))^m$$

Thus, the number of iterations required to ensure a certain success rate, p is:

$$m = \lceil \log(1 - p) / \log(B(\alpha, s, t')) \rceil$$

Refinement procedure: Let: S = multiset of l -mers that hashed to current bucket

f_i = BG distribution of base i

$A, W = 4 \times l$ matrices

Initialize: $A_{i,j}$ (# l-mers in S with base i at pos j) + f_i

$$W_{i,j} \leftarrow \log_2\left(\frac{p_{i,j}}{f_i}\right) \text{ and } p_{i,j} = A_{i,j} / \sum_k A_{k,j}$$

Repeat until convergence:

- Reset A: $A_{i,j}, f_i$.
- Score all l-mers in promoters using W.
- Add to A each l-mer with positive score.
- Compute W' from A.
- if(entropy(W') < entropy(W)) \Rightarrow (W \leftarrow W')

Scan promoters using W, select best l-mer from each promoter (with positive score), and output their consensus.

MEME Algorithm

MEME uses the method of Bailey and Elkan to identify likely motifs within the input set of sequences. You may specify a range of motif widths to target, as well as the number of unique motifs to search for. MEME uses Bayesian probability to incorporate prior knowledge of the similarities among amino acids into its predictions of likely motifs. The resulting motifs are output as profiles. A profile is a log-odds matrix used to judge how well an unknown sequence segment matches the motif.

MEME is one of the most popular programs for motif finding. It uses the expectation-maximization (EM) approach: first obtain an initial motif (which may not be very good), then iteratively obtain a better motif with the following two steps:

Expectation: compute the statistical composition of the current motif and find the probability of finding the site at each position in each sequence.

Maximization: These probabilities are used to update the statistical composition. (see [12])

The Algorithm: Let: $z_{i,j}$ = prob. of BS at pos j in promoter i $p_{b,c}$ = prob. of base b at pos c in motif.

Main steps:

Choose starting p

Repeat until convergence of p:

Re-estimate z from p

Re-estimate p from z

Bibliography

- [1] Francois Robert et al. Bing Ren. Genome-wide location and function of dna binding proteins. *Science*, pages 2306–2309, 2000.
- [2] Vishwanath R. Iyer et al. Genomic binding sites of the yeast cell-cycle transcription factors sbf and mbf. *Nature*, pages 533–539, 2001.
- [3] <http://creme.dcode.org/>.
- [4] <http://oregonstate.edu/instruction/bb492/fignumbers/figL11-43.html/>.
- [5] <http://proteomics.swmed.edu/~chiptochip.htm/>.
- [6] <http://www.biochem.ucl.ac.uk/bsm/xtal/teach/trans/tata.html/>.
- [7] <http://www.math.tau.ac.il/~rshamir/prima/PRIMA.htm/>.
- [8] <http://www.math.tau.ac.il/~rshamir/prima/PRIMA.htm/>.
- [9] <http://www.sciencemag.org/cgi/content/short/290/5500/2306?ck=nck/>.
- [10] <http://www.technion.ac.il/~asa/Papers/mc.pdf/>.
- [11] <http://www-users.med.cornell.edu/~jawagne//>.
- [12] www.cis.nctu.edu.tw/~is89048/fx.ppt/.