Analysis of Gene Expression Data **Spring Semester, 2005**

Lecture 7: April 7, 2005

Lecturer: R.Shamir and C.Linhart *Scribe: A.Mosseri, E.Hirsh and Z.Bronstein*[1]

7.1 Promoter Analysis

7.1.1 Introduction to Promoter Analysis

As we studied in our first lecture, each cell contains a copy of the whole gene. 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 is 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 would not be dealt 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 transcripted into precursone RNA.

7.1.2 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 7.1). Transcription factors typically combine to form "transcritpional switches" that encode complex logical functionality to control gene expression given a multitude of biological stimuli. The TF can either encourage or suppress the transcription process. Only when the "right" TF are located, a transcription process can be done. Transcription factors are proteins that bind to DNA region near the gene (the promoter region) at binding sites and regulate its transcription. They attach to the DNA at specific binding sites. Transcription factors work in combinations forming complex logical schemes. An example

[1]Based on the scribe of Eran Balan and Maayan Goldstein, May 2004

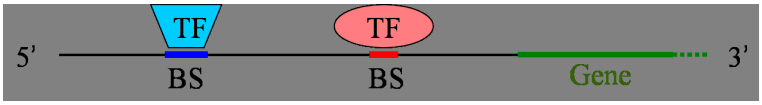

Figure 7.1: The location marked by BS are Binding sites where the transcription factors (TFs) can bind.

of transcriptional switch is shown in figure 7.2. 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 7.3 gives us a 3D picture of what is happening during the transcription factors attachment.

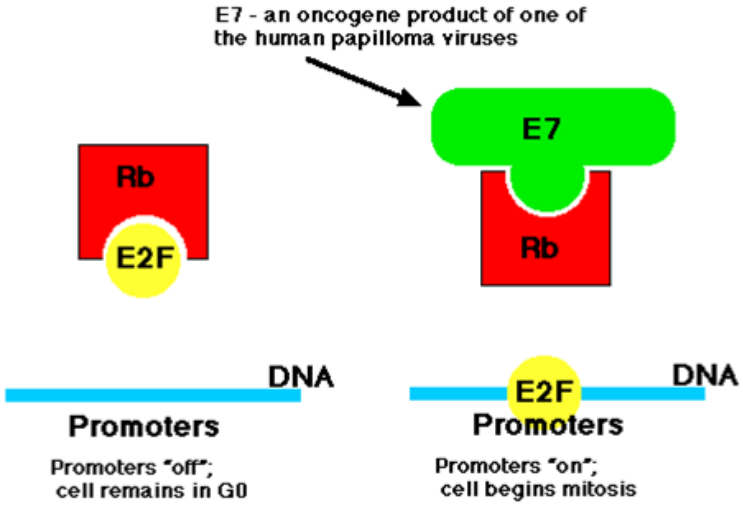

Figure 7.2: Source: [9]. Regulation with E2F transcription factor.

7.1.3 Cis and Trans Regulation

DNA sequence that acts to change the expression of the gene adjacent to it are *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.

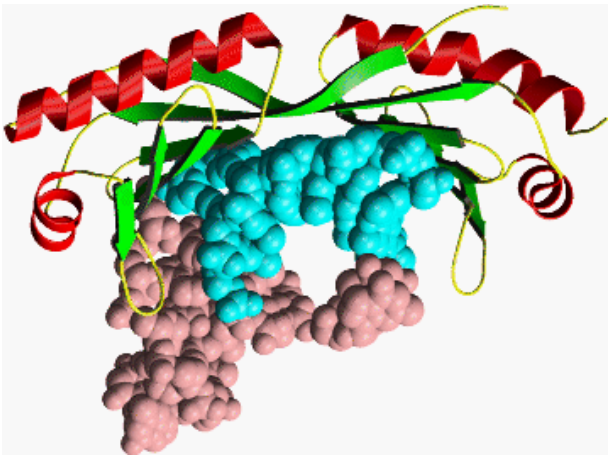

Figure 7.3: Source: [9]. 3D Regulation Structure.

7.1.4 Regulation of Transcription

As explained before, by binding to a genes promoter, TFs can either encourage or suppress 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 7.4 where a number of binding sites present, while the RNA polymerase protein is attached to the TATA binding site of them. One of the ways to study

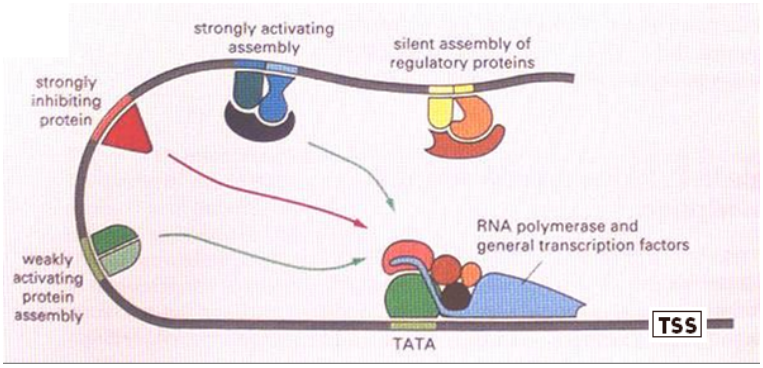

Figure 7.4: Regulation of Transcription. There are several TFs bound near the TSS location, they make it possible for the RNA polymerase to locate itself on the relevant gene sequence.

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 thus share a common binding site (due to the fact that similar transcriptional regulation cause

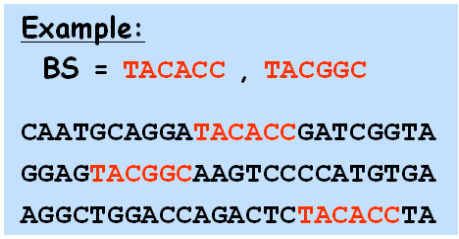

Figure 7.5: Exact string.

similar expression levels). Thus, we can use the knowledge we have on genome 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.

7.1.5 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. 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 transcripted area. Usually, the location of the TSS is known by the knowledge of the exact location of the mRNA sequence.

While looking for binding sites we would like to consider two problems: The promotor region we are looking at might be too short and we will miss many real BSs (false negatives). If the region is too long we will have lots of wrong hits (false positives). Usually, the length of promotor region 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.

7.1.6 Models for finding Binding Sites

We shell consider a number of models: exact string model, string mismatches model, degenerate string model and, finally, position weight matrix (PWM).

Exact String model The *Exact String model* will try to find an exact sequence in the DNA sequence (see Figure 7.5)

String Mismatches model The *String Mismatches model* will try to find an almost exact sequence and will tolerate a mistake in one of the positions (see figure 7.6).

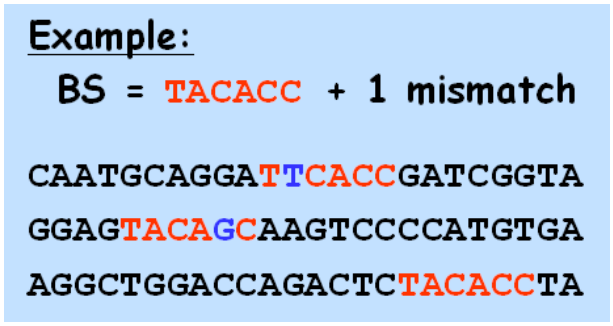

Figure 7.6: Exact mismatch.

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 (see figure 7.7).

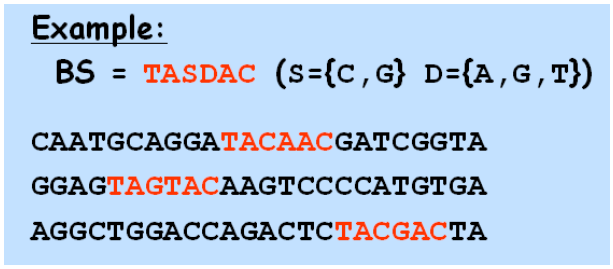

Figure 7.7: Degenerate String.

Position Weight Matrix model (PWM) 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 (see figure 7.8). 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.

Figure 7.8: PWM string model. The matrix defines the probabilities of each base at different location in the binding site sequence. As you can see there are a couple of examples for specific sequences and their probability according to the table.

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. In order to have a complete probabilistic picture of the data we are handling, we should also define a model for the background sequences (sequences between binding sites). In order to determine if a sequence is a binding site or not, we have to calculate the ratio between the probabilities of the sequence under the binding site probability model and that of the background model.

7.2 Technology

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

7.2.1 Identifying regulatory elements

In this method we use a DNA fragment containing potential regulatory sequences, such as the region upstream from a regulated gene, that are cloned next to a reporter gene encoding an easily assayed protein (see figure 7.9). The construct is put into cells and regulation is monitored by the activity of the reporter gene. We'll move different parts of the promoter, run the test again and see the effect on the expression level of the reporter gene. The process in which the expression levels are being compared is depicted in Figure 7.10 We will conduct this method on various parts of the promotor region.

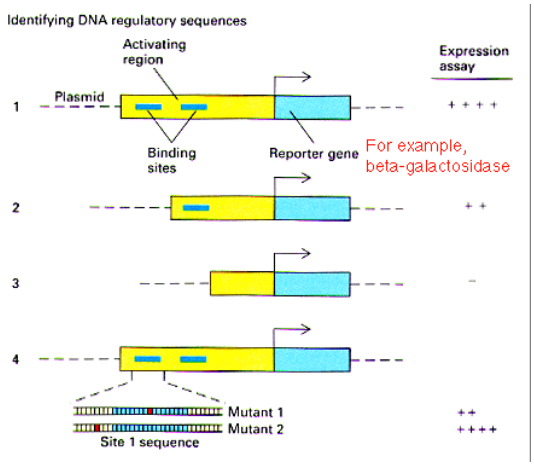

Figure 7.9: Source: [7]. Identifying regulatory elements. We are trying to find the influencing binding sites by removing different parts of the promoter region sequence (light color) and checking its effect of the expression level of the reporter gene (dark color).

7.2.2 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 [14]). A problem with this method is that creating the correct antibodies is not always simple.

The complex ChIP process 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 (see figure 7.11).

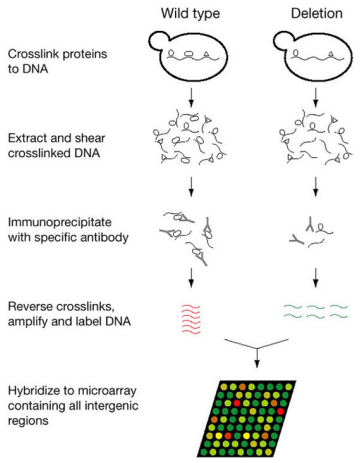

Figure 7.10: Source: [3] Strategy for analyzing genome-wide proteinDNA interactions. The reference probe can either consist of DNA generated in parallel from a strain bearing a deletion of the gene encoding the protein of interest (as depicted), or of non-fractioned genomic DNA amplified and labelled 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.

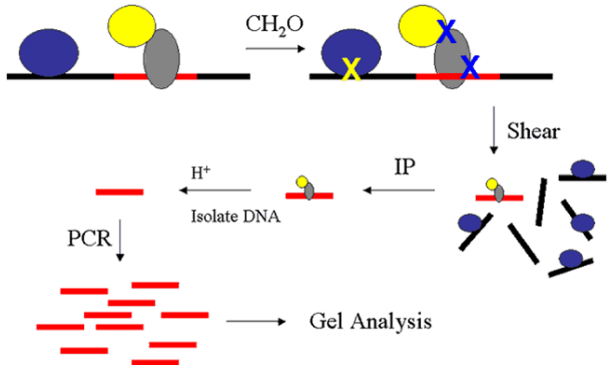

Figure 7.11: Source: [8] Chromatin Immunoprecipitation. Here we can see the complex process that was explained in section 7.2.2.

7.2.3 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 immunorecipitation (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 [2]).

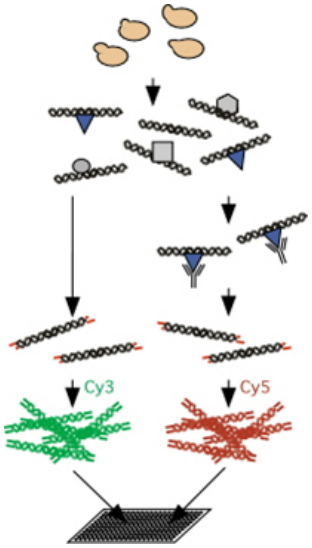

Figure 7.12: Source: [12]. Location analysis.

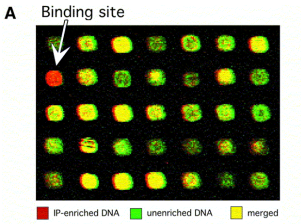

Figure 7.13: Source: [12]. Close-up of a scanned image of a microarray containing DNA fragments representing 6361 intergenic regions of the yeast genome. The arrow points to a spot where the red intensity is over-represented, identifying a region bound in vivo by the protein under investigation.

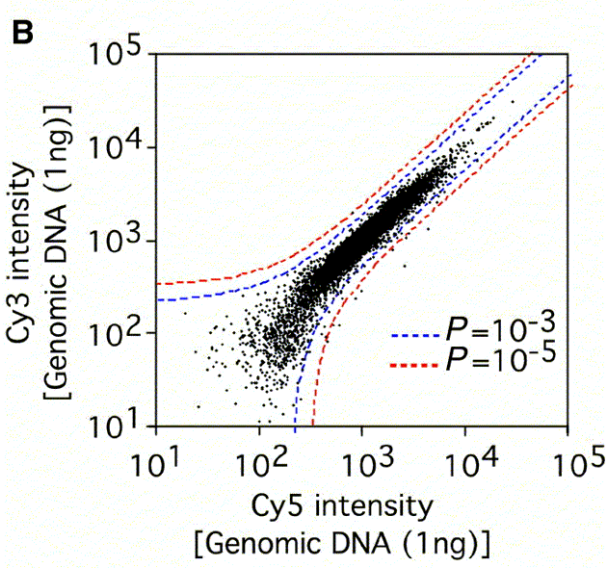

Figure 7.14: Source: [12]. Analysis of Cy3 and Cy5 labeled DNA amplified from 1ng of yeast genomic DNA using a single-array error model. The error model cutoffs for P values equal to 10^{-3} and 10^{-5} are displayed.

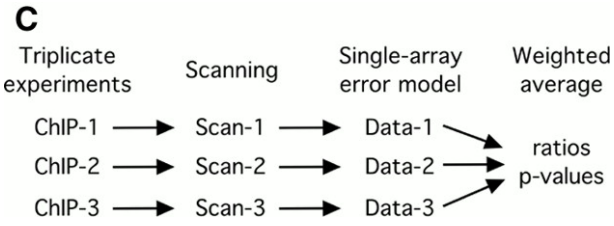

Figure 7.15: Source: [12]. Experimental design. For each factor, three independent experiments were performed and each of the three samples were analyzed individually using a single-array error model. The average binding ratio and associated P value from the triplicate experiments were calculated using a weighted average analysis method.

7.3 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.
- 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.

7.3.1 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 (PWM) for binding sites (BSs) *of known TFs*, PRIMA identifies TFs whose BSs are significantly over-represented in a given set of co-expressed genes' promoters (taking into consideration multiple BS's per promoter).
The algorithm is integrated into the Expander software (see [10]).

The algorithm: 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 of the human genome) and PWMs of known TFs (taken out of large internet TF databases). Output: 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 background (henceforth 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)

Computing a threshold for the PWD's: 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 computes as follows: First, we compute 2nd-order Markov-Model of BG seqs. Using the MM model, we generate random sequences (for e.g., 1,000 seqs of length 1,000 bp). Then, we set threshold so that PWM has f hits in the random sequences (e.g., f=100). This method of determining the PMW's parameters ensures a pre-defined false-positives rate, but has no guarantee on false-negatives rate. Estimating false-negatives (positives) rate requires good positive (negative) training-sets.

Computing the enrichment score Suppose each promoter has 0 or 1 hits. Then, define

- B is the number of BG promoters.
- T is the number of target-set promoters.
- b is the number of hits in BG promoters.
- t is the number of hits in target-set promoters.

Hence the probability for t hits in the target-set equals to

$$P(t) = \left(\begin{array}{c} b \\ t \end{array} \right) \left(\begin{array}{c} B-b \\ T-t \end{array} \right) \Big/ \left(\begin{array}{c} B \\ T \end{array} \right)$$

The probability for at least t hits is

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

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 together could encourage the transcription. It increases the possibility of getting a hit. We will take into account up to 3 hits per promoter. Let:
B,T = # of promoters in BG, target-set.
b1 , b2 , b3 = # of BG promoters with 1,2,3 hits.
t = total # of hits in target-set.

Thus the probability for at least t hits (Hyper-Genomic score) is:

$$\frac{\sum_{i+2j+3k\geq t} \binom{b_1}{i}\binom{b_2}{j}\binom{b_3}{k}\binom{B-b_1-b_2-b_3}{T-i-j-k}}{\binom{B}{T}}$$

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!)
Thus the probability for co-occurrence of at least t_{12} is

$$\frac{\sum_{i\geq t_{12}} \binom{t_1}{i}\binom{T-t_1}{t_2-i}}{\binom{T}{t_2}}$$

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.

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.

7.3.2 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 that identifies and visualizes 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

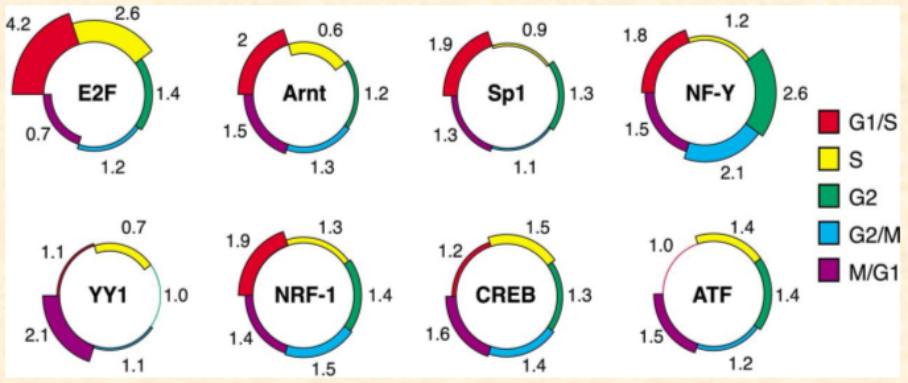

Figure 7.16: Source: [11]. Representation of TF PWMs in the cell cycle phase clusters. The eight circles correspond to the PWMs that were highly enriched in promoters of cell cycle-regulated genes. Each circle is divided into 5 zones, corresponding to the phase clusters. The number adjacent to the zone 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.

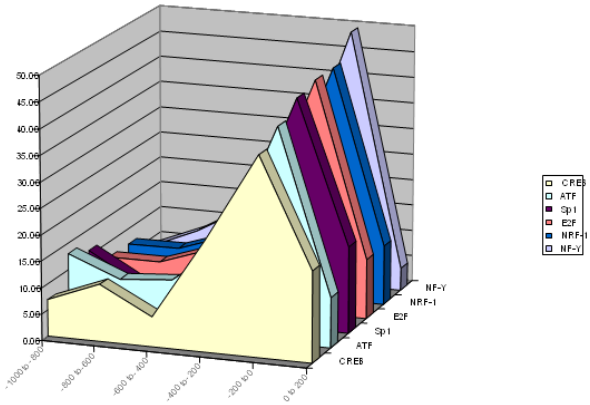

Figure 7.17: Source: [11]. Distribution of locations of TFs putative binding sites found in 568 cell cycleregulated promoters. Promoters were divided into six intervals, 200 bp each. For each of the PWMs, the number of times its computationally identified binding sites appeared in each interval was counted (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).

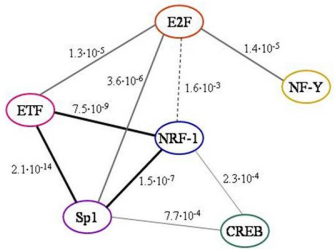

Figure 7.18: Source: [11]. Pairs of PWMs that co-occur significantly in promoters of genes regulated in a cell cycle manner. It was examined whether the PWMs can be organized into regulatory modules. For each possible pair formed by these PWMs, we tested whether the prevalence of cell cycle-regulated 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.

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 [6]).

Goal: Discover modules which are 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 the limited knowledge of TFs, the usage of PWMs to model BSs, the fact that TFs order and number of hits within the module is not taken into account.

Definitions:

- 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.

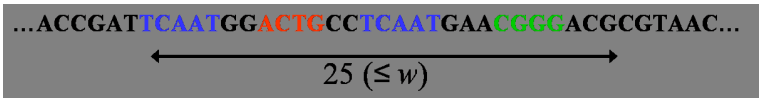

Figure 7.19: Example: Instance of a (r=3,w=30)-module.

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

The algorithm:

- Find enriched PWMs (p-value 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-value less than 0.05.
- Filter similar modules (more than 75% overlapping instances).

If we look closely at the third step of the algorithm, we 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.

Possible simplification: 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

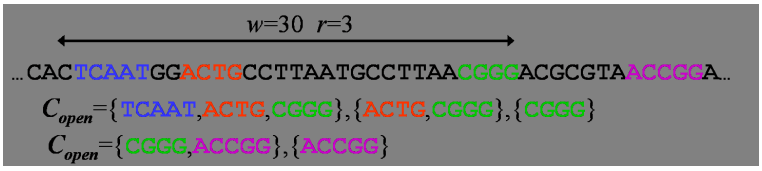

Figure 7.20: Instance of a (r=3,w=30)-module and possible instances of Copen.

The details of the algorithm are shown in Figure 7.21 (see [5]).
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 7.21: Source: [13] 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.

7.3.3 Motif Finding Tools

Definitions Motif(l,d) as a 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 and ACCAGAGAG.

Goal: 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

The main idea of the algorithm is to 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 motif bucket. Thus buckets with large count are likely to correspond to a motif.

The algorithm: (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.

An example for the algorithm is seen in Figure 7.22.

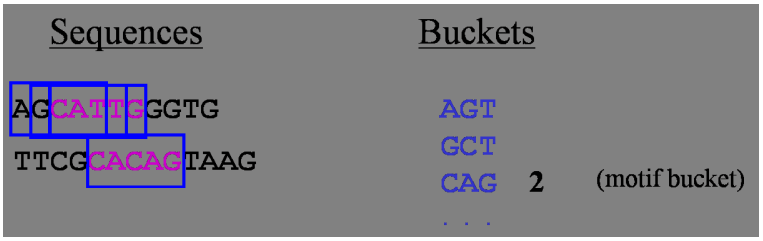

Figure 7.22: An example of random projection, with l=5, d=1, k=3, motif M="CATAG" and projection function h($x_1x_2x_3x_4x_5$)=$x_1x_2x_5$, The motif bucket is CAG. In the example, we can use any base for x_3 and x_4 and we look at all the sub-sequences that fall into the same bucket. And we find x_3 and x_4 according to the most frequent sub-sequences.

Analysis: Choosing k and s is very important, 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.

Known good values for k and s are: k = l - d - 1 (to keep average bucket size small) $s = 2t(L - l + 1)/4^k$ where L is the average promoter length.

The probability for a motif instance to hash into its bucket is

$$\alpha = \frac{\binom{l-d}{k}}{\binom{l}{k}}$$

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) is

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

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

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

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

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

Refinement procedure: Definitions:

- S is a multi-set of l-mers that are hashed to a specific bucket.
- f_i is the BG distribution of base i
- A, W = 4 x l-matrices

The algorithm:

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

$$W_{i,j} \leftarrow \log_2(\frac{p_{i,j}}{f_i})$$

$$p_{i,j} = A_{i,j} \Big/ \sum_k A_{k,j}$$

- 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 (see [1]) 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 [4]) 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 [15])

The Algorithm (Mixture Model version)
The data we are starting with is a promotor DNA sequence. We should look at all overlapping sequential l-mers in the input and analyze the probability of the motif we are suggesting. We'll define the input data as follows: X = (X_1,...,X_n) : X_i is an input l-mer.
Let's assume the X_is were generated by a two-component mixture model - $\theta = (\theta_1, \theta_2)$:
Motif model = θ_1: $f_{i,b}$ = Probability of base b at position i in a motif, i = 1..L (an example of the PWM model can be seen in Figure 7.8).
BG model = θ_2: $f_{0,b}$ = Probability of base b at any position.
Mixing parameter: $\lambda = (\lambda_1\ ,\ \lambda_2)$ λ_j = Probability that model j is used (as noted in the definition above, the motif model is marked as 1 and the BG model is marked as 2). ($\lambda_1 + \lambda_2$=1)
After understanding the input data and the probability model, let's define the missing data format. We define a random variables set:

$$Z = (Z_1, ..., Z_n), Z_i = (Z_{i1}, Z_{i2})$$

$Z_{ij} = 1$ if X_i is from model j and 0 otherwise. Z_{ij} is an indicator to the fact that the i'th l-mer is was create by model j.

Next we define the likelihood of the data according to the probability model's parameters:

$$L(\theta, \lambda|X, Z) = P(X, Z|\theta, \lambda) = \prod_{i=1...n} P(X_i, Z_i|\theta, \lambda)$$

Usually, when searching for the maximum or minimum of a function, it is easier to look at the function's Log:

$$log(L) = \sum_{i=1...n} \sum_{j=1,2} Z_{ij} log(\lambda_j P(X_i|\theta_j))$$

After the problem model was defined, our goal now is to maximize the the expected log-likelihood by changing the θ's and λ's. As noted, the EM algorithm will be used.

Outline of EM algorithm

- Choose starting $\theta^{(0)}$, $\lambda^{(0)}$.
- Repeat until convergence of θ:
  - E-step: Re-estimate Z from θ, λ, X.
  - M-step: Re-estimate θ, λ from X, Z.
- Repeat all of the above for various $\theta^{(0)}$, $\lambda^{(0)}$ starting points.

E-step:
Let us compute the expectation of log L over Z:

$$logL|X,Z] = \sum_{i=1...n} \sum_{j=1,2} Z_{ij}^{(0)} log(\lambda_j P(X_i|\theta_j))$$

$$(Eq.7.1)$$

Where:

$$Z_{ij}^{(0)} = E[Z_{ij}] = P(Z_{ij} = 1|\theta^{(0)}, \lambda^{(0)}, X_i) = \frac{P(Z_{ij} = 1, X_i|\theta^{(0)}, \lambda^{(0)})}{P(X_i|\theta^{(0)}, \lambda^{(0)})} =$$

$$= \frac{P(Z_{ij} = 1, X_i|\theta^{(0)}, \lambda^{(0)})}{\sum_{k=1,2} P(Z_{ik} = 1, X_i|\theta^{(0)}, \lambda^{(0)})} =$$

$$= \frac{\lambda_j^{(0)} P(X_i|\theta_j^{(0)})}{\sum_{k=1,2} \lambda_k^{(0)} P(X_i|\theta_k^{(0)})}$$

After getting a simple expression of the expected log likelihood, we can find θ and λ that maximize it.

M-step:
Find θ and λ that maximize the expected log-likelihood (Eq. 7.1). To find λ it is sufficient to maximize $L_1 = \sum_{i=1...n} \sum_{j=1,2} Z_{ij}$ Log(λ_j), and get $\lambda_j^{(1)} = \sum_{i=1}^{n} \frac{Z_{ij}^{(0)}}{n}$, j=1,2. To find θ, we need to solve $\theta_j^{(1)} = \underset{\theta_j}{\text{argmax}} \sum_{i=1}^{n} Z_{ij}^{(0)} Log(P(X_i|\theta_j))$. As seen in Eq. 13-20 in [1], $\theta_j^{(1)}$ can be calculated easily.

Homology

In order to optimize the motif search, one can use homology between species. Homology, which is the resemblance between two organisms' genome, will probably contain much of the shared binding sites.
According to the same reasoning, transcription factors that are conserved throughout evolution, will probably have significant biological function. That is, the more a TF is abundant in various species, the more important it is.
A more advanced technique will be to study homologies between species, under the same conditions. See figure 7.23.

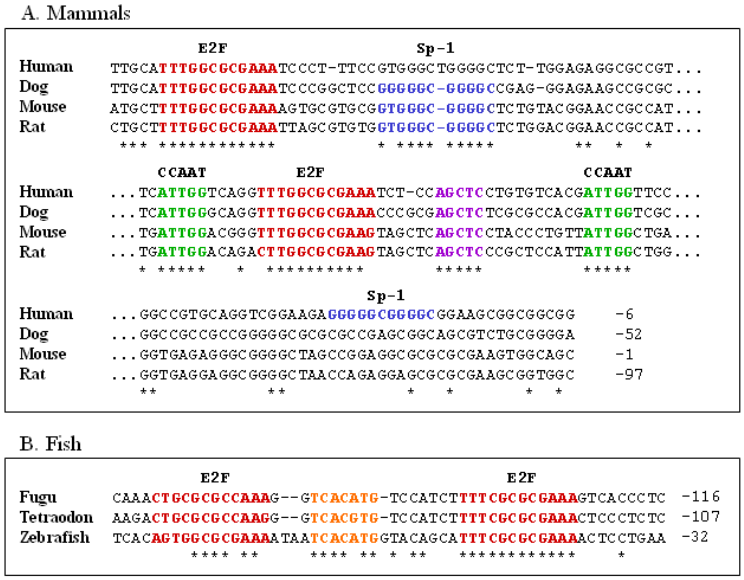

Figure 7.23: A homology between human, dog, mouse and rat promotor sequences, and the position of real Binding sites. Notice that the homology areas contains the known binding sites.

Bibliography

[1] T.L. Bailey and C. Elkan. Fitting a mixture model by expectation maximization to discover motifs in biopolymers. *ISMB*, pages 28–36, 1994.

[2] R. Bing and R. Francois et al. Genome-wide location and function of dna binding proteins. *Science*, pages 2306–2309, 2000.

[3] V. R. Iyer et al. Genomic binding sites of the yeast cell-cycle transcription factors sbf and mbf. *Nature*, pages 533–539, 2001.

[4] C. E. Lawrence and A. A. Reilly. An expectation maximization (em) algorithm for the identification and characterization of common sites in unaligned biopolymer sequences. *Proteins 7*, pages 41–51, 1990.

[5] R. Sharan, A. Ben-Hur, G. Loots, and I. Ovcharenko. Creme:cis-regulatory module explorer for the human genome. *Nucleic Acids Research*, pages 253–256, 2004.

[6] `http://creme.dcode.org/`.

[7] `http://oregonstate.edu/instruction/bb492/fignumbers/figL11-43.html/`.

[8] `http://proteomics.swmed.edu/~chiptochip.htm/`.

[9] `http://www.biochem.ucl.ac.uk/bsm/xtal/teach/trans/tata.html/`.

[10] `http://www.math.tau.ac.il/~rshamir/prima/PRIMA.htm/`.

[11] `http://www.math.tau.ac.il/~rshamir/prima/PRIMA.htm/`.

[12] `http://www.sciencemag.org/cgi/content/short/290/5500/2306?ck=nck/`.

[13] `http://www.technion.ac.il/~asa/Papers/mc.pdf/`.

[14] `http://www-users.med.cornell.edu/~jawagne//`.

[15] `www.cis.nctu.edu.tw/~is89048/fx.ppt/`.