Lecture 8: June 14, 2007

Lecturer: R.Shamir and C.Linhart *Scribe: O.Ish-Shalom, G.Tannenbaum*[1]

8.1 Promoter analysis

8.1.1 Control of gene expression

Nearly all the cells of a multicellular organism contain the same genome. In the course of embryonic development, a fertilized egg cell gives rise to many cell types that differ dramatically in both structure and function. The differences between a mammalian neuron and a lymphocyte, for example, are so extreme that it is difficult to imagine that the two cells contain the same DNA. Cell differentiation is achieved mainly by control of gene expression. When a cell expresses a gene it means the relevant portion of DNA is first transcribed into RNA, and then translated into protein. By controlling which genes are transcribed, the cell can control which proteins to synthesize. Of course, this description is somewhat simplified, and the pathway from DNA to protein contains multiple steps.

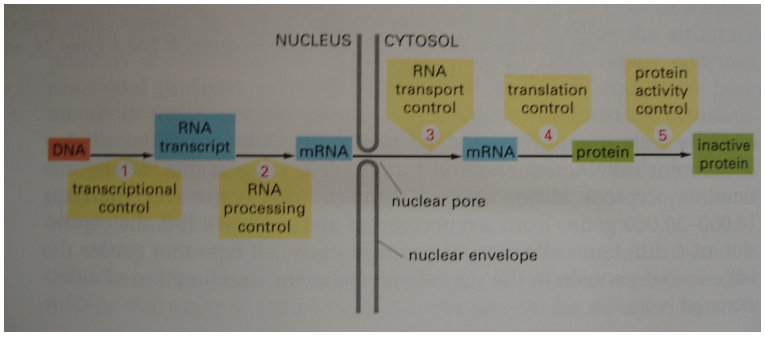

Figure 8.1: Examples of regulation at each of the steps above are known. Still, for most genes, the main site of regulation is step 1: transcription of DNA into RNA.

In principal, a cell can control all steps that appear in Figure 8.1: (1) controlling when and how often a gene is transcribed, (2) controlling how the primary RNA transcript is spliced or otherwise processed, (3) regulating the transport of RNA from the nucleus to the cytosol, (4) selecting which mRNAs are translated by ribosomes, or (5) selectively activating or inactivating proteins after they have been made. For most genes, however, the control of transcription is paramount.

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

8.1.2 Regulation of transcription

As mentioned above, transcription is the process in which template DNA is used to create a matching RNA molecule. The enzyme that carries out transcription is called *RNA polymerase* (Figure 8.2). However, RNA polymerase can not initiate transcription on its own. It requires two special sets of proteins (not shown in the figure): *basic transcription proteins* and *transcription factors*.

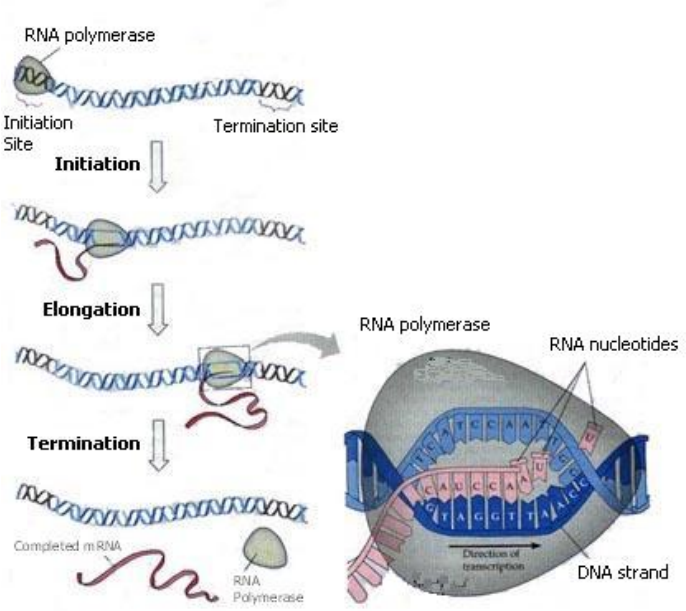

Figure 8.2: The different stages of transcription are shown above. First, the RNA polymerase binds tightly to the DNA stretch just upstream from the gene. Then, after initiation is completed, RNA polymerase starts synthesizing the RNA molecule of the gene. Signals in the DNA sequence mark the end of the gene, and the polymerase literally falls off the DNA. The newly made RNA molecule is released and translated into protein. Regulation of transcription is exerted when the process is initiated

Basic transcription proteins

As mentioned in section 8.1.2, RNA polymerase can not initiate transcription on its on. It needs the help of *basic transcription proteins*. These proteins bind to the DNA stretch just upstream the start of the gene (to be henceforth called the *promoter region*). They start accumulate at a specific DNA segment called the *TATA box*, and form an elaborate assembly that performs the following tasks: (1) position the RNA polymerase at the start of the gene, (2) help it bind firmly to the DNA, (3) aid in pulling apart the two strands of DNA and (4) allow the RNA polymerase to detach, and start transcription. An important characteristic of

these proteins is that they are not gene-specific. In fact, probably the same proteins assemble before most genes. We will henceforth use the term *transcription machinery* to describe the combined effect of RNA polymerase and the basic transcription proteins (Figure 8.3).

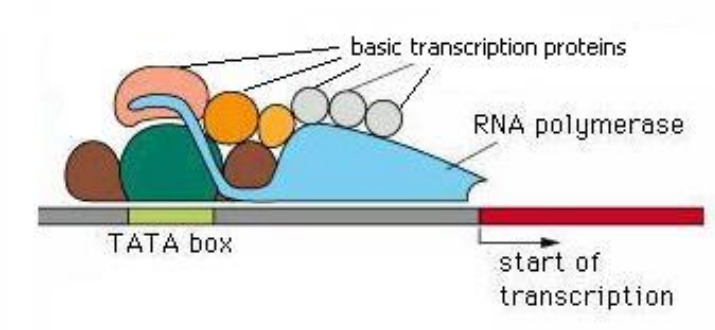

Figure 8.3: RNA polymerase together with basic transcription proteins form the transcription machinery. As mentioned above, RNA polymerase can not initiate transcription on its own.

Transcription Factors (*TFs*)

Although the transcription machinery as a whole can initiate transcription *in vitro*, it fails to do so inside the cell. Indeed, nearly all human genes will fail to initiate transcription without dedicated transcription factors. TFs can bind DNA in specific binding bites (*BSs*), and promote the recruitment of the transcription machinery. These TFs are called *activators*, because, as the name implies, they help activate the transcription machinery. Another group of TFs is *repressors*. As one can guess, when they bind DNA in specific BSs, they repress the recruitment of the transcription machinery. The different TFs work together as a "committee" to control the expression of a gene. The effects of multiple TFs is combined to determine the rate of transcription initiation (Figure 8.4).

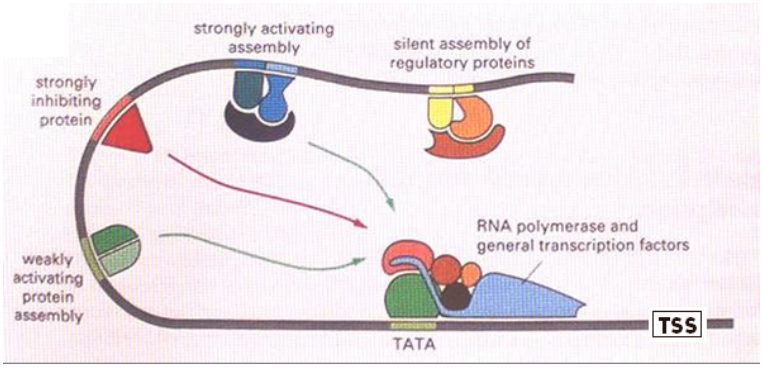

Figure 8.4: Multiple TFs bind to specific DNA BSs. They work together, combining their effects to determine the overall rate of transcription initiation.

In 1979 it was first discovered that TFs can be bound thousands of nucleotide pairs away from the gene and still regulate its expression. Moreover, influencing TFs were also found downstream from some genes. What model can account for this "action at a distance"? It seems that in most cases, the simplest model applies: The DNA loops out and allow the TF to directly influence events at the start of the gene (Figure 8.5).

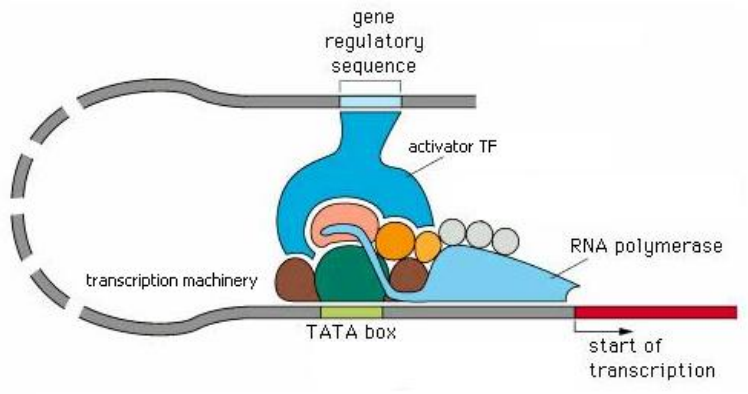

Figure 8.5: Transcription machinery bound to activator TF. Some TFs are thousands of nucleotide pairs away and manage to interact with the transcription machinery by DNA loops. The broken stretch of DNA signifies that the length varies.

8.1.3 Promoter analysis - problem definition

Promoter analysis deals with the following problems:
Find new BSs for some known TF.
Find new BSs for unknown TFs (called *motifs*).
Find combinations of TFs that regulate genes together.

Promoter analysis - preliminaries and assumptions

- Ignore repetitive DNA sequences

  Consider some repetitive sequence that appears millions of times throghout the genome. The probability that this sequence is an active BS is slim. Therefore, repetitive sequences are usually masked prior to promoter analysis.

- Consider both strands of DNA.

  Though only one strand contains the coding sequence of the gene, both strands are used for DNA-protein bindings.

- A good downstream bound for the promoter region exists.

  The promoter region is defined as the DNA stretch upstream from the gene. The Transcription Start Site (*TSS*) is therefore the downstream bound for the promoter region. Experimentally verified TSS are therefore essential for a proper analysis.

- A heuristic upstream bound for the promoter region is used.

  The only upstream bound that naturally comes to mind is the end of the next gene upstream. This upper bound can indeed be used in yeasts. The intergenic regions there are relatively small (hundreds of nucleotides). In humans however, the distance between consecutive genes can be huge (our genes constitute only about 2 percent of the entire genome).

  Today, it is common practice to set the upstream bound for promoters at 500-2000 nucleotides upstream from the TSS. If too short promoter regions are used, we might miss real BSs. On the other hand, if we use very long promoter regions, the portion of the real BSs will decrease and the rate of false hits will increase.

8.2 Biological approaches to promoter analysis

This section presents some of the common methods used for promoter analysis.

8.2.1 Finding new motifs

Using reporter genes

This method looks for *BSs* of unknown *TFs*. It uses a gene that encodes an easily detectable protein (*reporter gene*). DNA engineering is used to place our promoter just upstream to the reporter gene. The construct is put into cells, and the rate of transcription is easily monitored by the levels of protein produced. We can perform multiple tests, using different promoter subsequences each time (see figure 8.6).

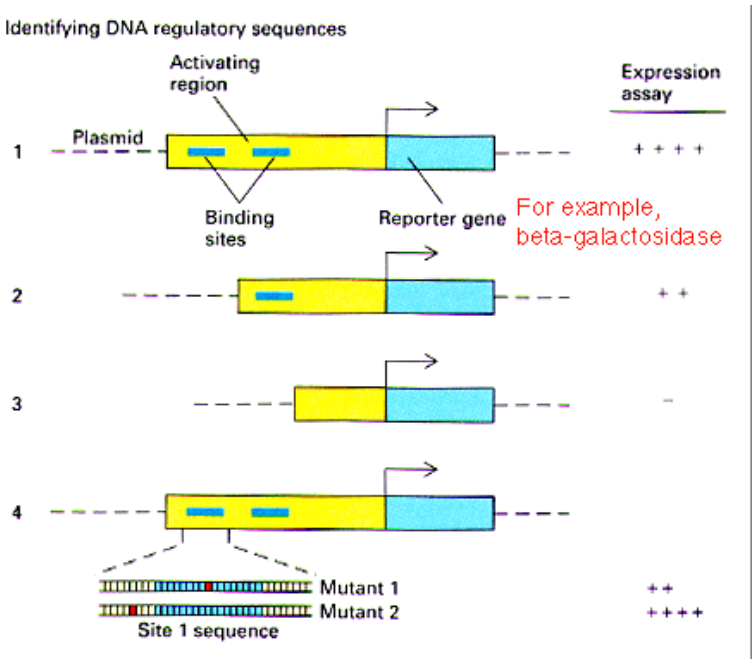

Figure 8.6: Source: [11]. Finding new motifs. New BSs of known TFs can be found by using reporter genes. The rate of transcription is easily monitored by measuring the protein levels produced. By performing multiple tests with different promoter subsequences, we can deduce which subsequences affect the rate of transcription. In other words: we can find where the BSs are.

8.2.2 ChIP - Finding BSs of known TFs

ChIP (Chromatin Immunoprecipitation) is a procedure that identifies BSs bound by known TFs *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 allows us to purify the BSs bound by known TFs at the time of cross-linking. The purified DNA can be amplified by PCR, and sequence information can be obtained by gel electrophoresis. (see Figure 8.7 and also [15]). Note that making antibodies against arbitrary TFs is not always easy.

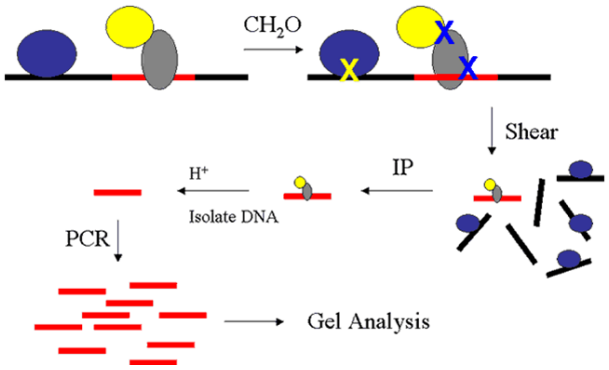

Figure 8.7: Source: [12] Performing ChIP analysis involves the following stages: (1) expose the cell to a given a set of conditions, (2) freeze its chemical stage, (3) lyse the cell and create fragmented DNA, (4) add the antibodies that bind to the TFs, (5) isolate antibody bound DNA-protein complexes, (6) remove TFs and antibodies from DNA, (7) perform PCR and (8) obtain BSs sequences by gel electrophoresis.

8.2.3 ChIP on Chip - Identifying BSs of known TFs throughout the whole genome

ChIP on (DNA) Chip basically performs all the steps of the standard ChIP procedure (see 8.2.2). However, since we are dealing with vast amounts of DNA, sequencing with gel electrophoresis is simply not feasible. Instead, we use dedicated intergenic DNA chips. Intergenic DNA chips that span the entire non-repetitive part of the human genome is already in commercial use (see Figure 8.8 and [2]).

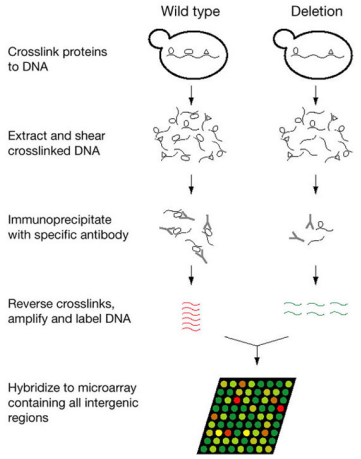

Figure 8.8: Source: [6] BSs for a specific TF can be found across the whole genome. The process is like the ChIP procedure described above, with one difference: instead of using gel electrophoresis, we use dedicated intergenic DNA chips. Once the purified DNA fragments are received, we use denaturation to open DNA into single strands. The single stranded DNA is then poured over the surface of the DNA chip. The reference set consists of DNA fragments generated under the very same conditions in a strain bearing a deletion of the gene that encodes the TF protein.

8.2.4 ChIP-DSL (DNA Selection and Ligation)

We start the process by identifying in advance a sequence of 40 nucleotides, in all human promoters that is unique as possible. We synthesize, for each human promoter, the two 20mer oligonucleotide sequences. The set of all pairs of 20mer oligonucleotide sequences will be henceforth called the *oligo pool*. The ChIP-DSL can be described as follows: (figure 8.9)
(1) Perform the standard ChIP procedure.
(2) Before the PCR step, mix DNA with oligo pool.
(3) Wait for interactions to take place, wash unbound oligonucleotide sequences. Ligate matching 20mer pairs.
(4) Perform PCR at the **oligonucleotide sequences** that are attached to DNA.

ChIP-DSL has several advantages over its predecessor ChIP on chip. First, it handles better de-cross linking: if the DNA is not released from the TF cross linkage, it will not

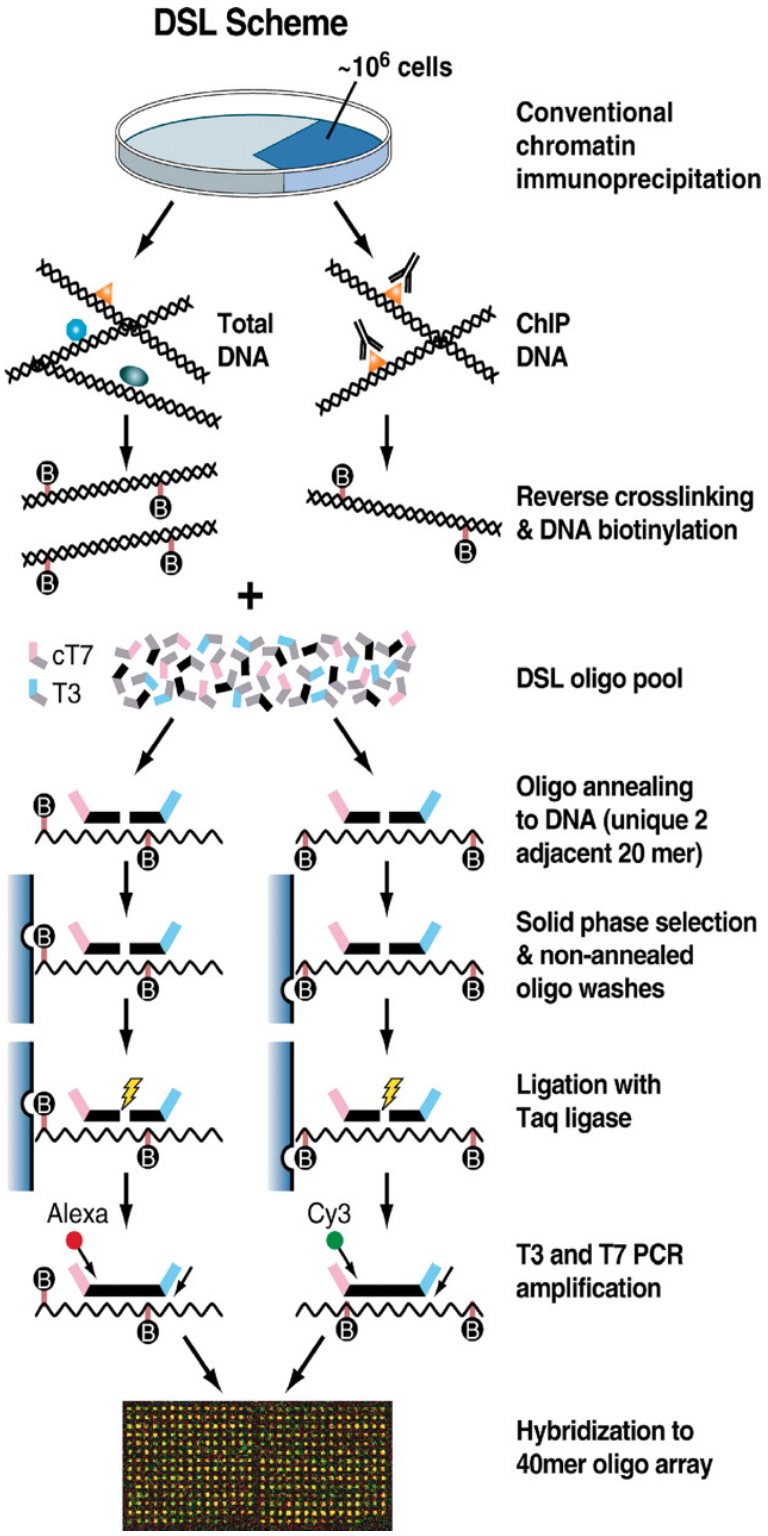

Figure 8.9: Source: [7] A schematic description of ChIP-DSL.

be amplified efficiently during PCR. On the other hand, in ChIP-DSL, the oligonucleotide ligation is less affected. Second, since the oligo pool contains only distinct sequences, it does not contain repetitive sequences, avoiding potential interference. Third, since the PCR process is performed on the oligonucleotide sequences, the amplicons all have the same size, and amplification is likely to be less biased.

8.2.5 Protein binding microarrays

The PBM method starts by synthetically generating all possible DNA sequence variants of a given length k (all *k-mers*) on a single universal microarray. This is done by converting high-density single-stranded oligonucleotide arrays to double-stranded DNA arrays (Figure 8.10). These microarrays are used for comprehensively determining the binding specificities over a full range of affinities for different TFs of different structural classes from different orgarnisms (Figure 8.11). The unbiased coverage of all k-mers permits high-throughput interrogation of binding site preferences. (see [4])

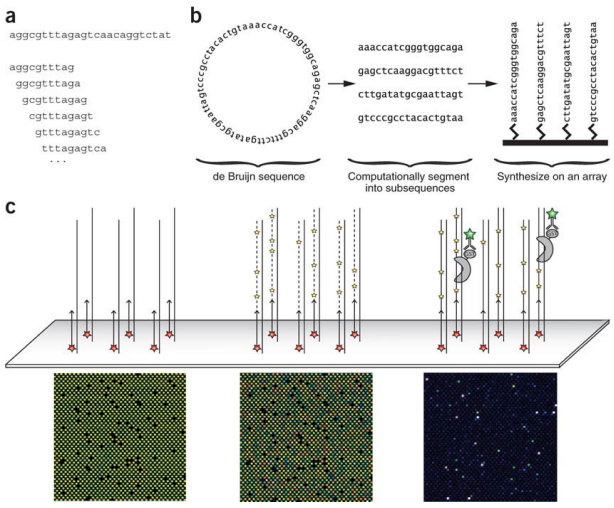

Figure 8.10: Source: [4]. (a) Overlapping k-mers. Each sequence on the microarray contains several distinct, overlapping k-mer binding sites. Here, k = 10. (b) Example of a de Bruijn sequence of order 3. A de Bruijn sequence of order 3 contains all 64 3-mer variants exactly once. The de Bruijn sequence is partitioned into subsequences that overlap by two bases, preserving all 3-mers in the sequence. These subsequences then become the spots on the microarray. (c) Universal PBM containing all possible 10-mer binding sites, bound by the S. cerevisiae TF Cbf1 expressed with a glutathione S-transferase (GST) epitope tag. Above is a schematic showing the three main stages of each experiment: primer annealing, primer extension, and protein binding. Beneath are zoom-in images of each stage for the same microarray, scanned at different wavelengths: Cy5-labeled universal primer, Cy3-labeled dUTP and Alexa488-conjugated alpha-GST antibody. Fluorescence intensities are shown in false color, with blue indicating low signal intensity, green indicating moderate signal intensity, yellow indicating high signal intensity, and white indicating saturated signal intensity. The variability observed in the Cy3-dUTP signal is due to differences in the nucleotide composition of each feature. The blank spots are single-stranded negative control probes that do not contain the universal primer sequence.

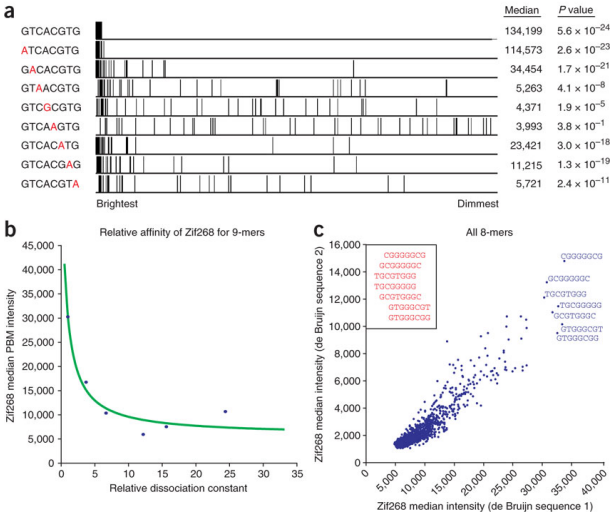

Figure 8.11: Source: [4]. (a) Enrichment of different Cbf1 binding site variants. All spots are ranked in descending order by their normalized signal intensities, and spots containing a match to each specified 8-mer are marked. For each 8-mer the median intensity over all such spots is shown (in fluorescence units), as is the p-value for enrichment as calculated by the Wilcoxon-Mann-Whitney test. (b) Correspondence between signal intensity and binding affinity. The median intensities for six 9-mer binding site variants for the mouse TF Zif268 are plotted against their relative dissociation constants as measured by a quantitative binding (QuMFRA). (c) Correspondence between separate PBM experiments performed on microarrays constructed with independent de Bruijn sequences. The median intensity for spots containing a match to each 8-mer is shown for each experiment. As evident here, the PBM data are consistent not only for the k-mers with highest affinity but also for the k-mers with moderate and low affinity.

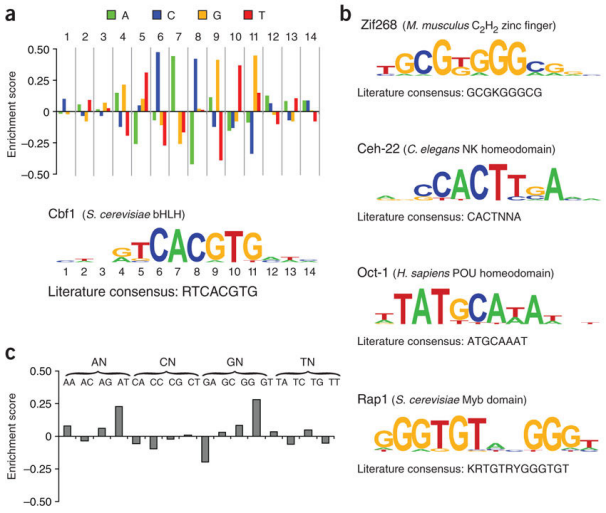

Figure 8.12: Source: [4]. (a) Method of constructing PWMs and sequence logos, using Cbf1 as an example. First, all 8-mers containing three gapped positions or fewer are evaluated using an enrichment score, and the highest scoring 8-mer (in this case GTCACGTG) is used as a seed for constructing the motif. Second, at each position within this 8-mer seed, all four possible nucleotides are compared by inspecting the ranks of the probes matching each of the four variants. This analysis produces a score between 0.5 and 0.5 for each variant at each position. Third, positions outside the 8-mer seed are inspected by dropping the least informative position within the seed and repeating the preceding analysis at every additional position that yields an 8-mer with at most three gaps (ensuring that the positions inspected outside of the 8-mer seed are based on a roughly equal number of samples to those within the 8-mer seed). This analysis produces the bar graph shown. (b) Logos for four additional TFs constructed using this method. For each, the organism and structural class are given. Consensus sequences in a and b were obtained from the literature for Cbf1, Zif268, Ceh-22, Oct-1 and Rap1 (standard IUPAC abbreviations are used (K = T,G; R = A,G; Y = C,T; N = A,C,G,T). (c) Extension of the method for motif construction described in a to the case of dinucleotide variants and applied to the first two positions in the Cbf1 motif. Here, all 16 variants of the form NNCACGTG were obtained, and the enrichment score of each was computed.

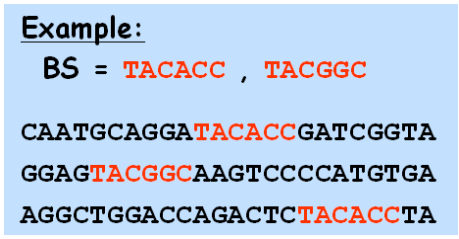

Figure 8.13: Exact string: only exact matches of the target string are considered hits.

8.3 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. (promoter elements that control the gene adjacent to them are called cis-acting elements).

- 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 (e.g. 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.

8.3.1 String matching models in finding BSs

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

Exact string model Using the *exact string model* is trying to find an exact sequence in the DNA sequence (see Figure 8.13)

String mismatches model Using the *string mismatches model* is trying to find an almost exact sequence while tolerating mistakes in some of the positions (see Figure 8.14).

Degenerate string model When using the *degenerate string model*, also known as *consensus model*, one tries 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 strings to search for (see Figure 8.15).

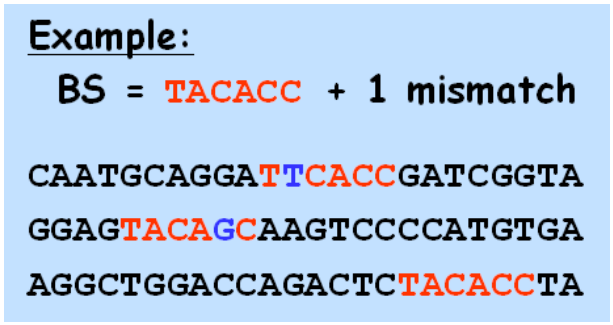

Figure 8.14: Exact mismatch: some mismatches are tolerated when searching for the target sequence.

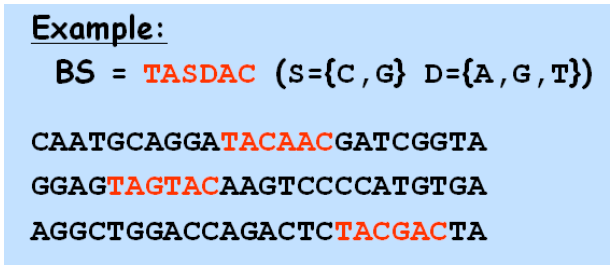

Figure 8.15: Degenerate String: searching for matches of the original string, where S can be replaced by C or G, and D by A, G or T.

Position Weight Matrix model (PWM) When using the *position weight matrix model*, also known as *position specific scoring matrix model*, one creates 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 8.16). 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 a predefined threshold).

Figure 8.16: 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 model and that of the background model.

8.3.2 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 [3]).

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 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 computed as follows: First a 2nd-order Markov-Model of background sequences is computed. Using the MM model, random sequences are generated (for e.g., 1,000 seqs of length 1,000 bp). Then, a threshold is set so that the PWM has given amount of hits, f, 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.

The probablity for getting t hits when selecting T promoters at random out of B promoters, according to hypergeomtric distribution, 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. 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-Geomtric distribution) 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 cell-cycle regulated 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 BS 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.

Alternative enrichment scores: The Hypergeomtric enrichment score used by PRIMA is not the only possible one to use.

- Pros: It's model independent and gives accurate results.

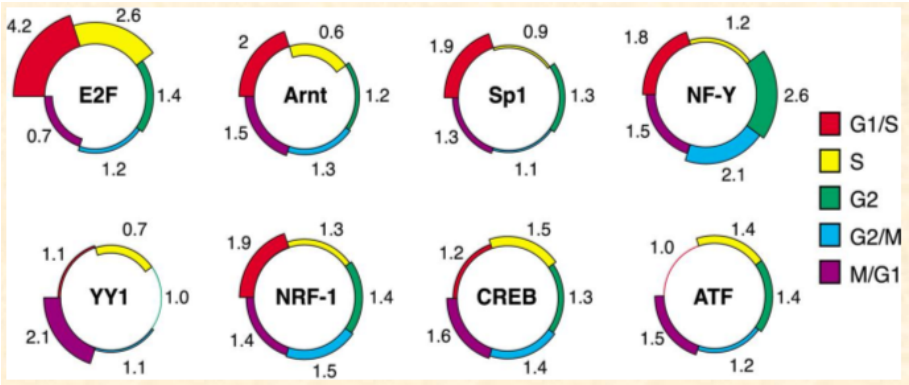

Figure 8.17: Source: [13]. Representation of TF PWMs in the cell cycle phase clusters. The eight circles correspond to the PWMs that were highly enriched in promoters of cell cycle-regulated genes. Each circle is divided into 5 zones, corresponding to the phase clusters. The number adjacent to the zone 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.

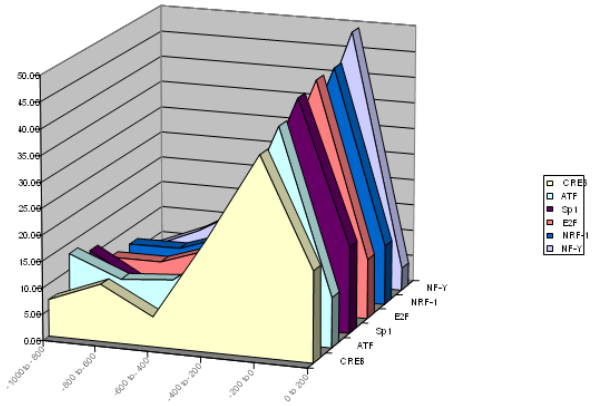

Figure 8.18: Source: [13]. Distribution of locations of TFs putative binding sites found in 568 cell cycleregulated promoters. Promoters were divided into six intervals, 200 bp each. For each of the PWMs, the number of times its computationally identified binding sites appeared in each interval was counted (after accounting for the actual number of base pairs 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, $p<0.01$).

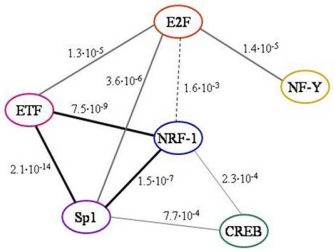

Figure 8.19: Source: [13]. Pairs of PWMs that co-occur significantly in promoters of genes regulated in a cell cycle manner. It was examined whether the PWMs can be organized into regulatory modules. For each possible pair formed by these PWMs, it was tested whether the prevalence of cell cycle-regulated 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.

- Cons: Assumes all promoters are equally likely to be in the target set, which is not accurate, for example because different promoters have different lengths and GC content.

Other possible enrichment scores can be model based (e.g. likelihood score), or use promoter bins.

Binning promoters:

- Bin promoters according to their length / GC-content (e.g., bin 1 contains promoters of length 1-100, bin 2 contains promoters of length 101-200...)
  $b(p)$=bin of promoter p

- Use background set to estimate the expectation $E(b)$ and variance $V(b)$ of the number of hits in a promoter in bin b

- By Central Limit Theorem:

$$t \sim N(\sum_{p \in t} E(b(p)), \sum_{p \in t} V(b(p)))$$

- And like the HG case, given t hits, the enrichment score is the tail of the distribution:

$$score = \int_{t}^{\infty} N()$$

8.3.3 CREME - Cis-Regulatory Module Explorer

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 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 [10]).

Definitions:

- Module = Set of PWMs.
- r is the number of PWMs in the module.
- Instance of a module is 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.

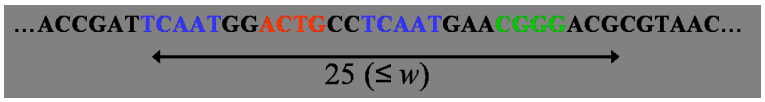

Figure 8.20: Example: Instance of a (r=3,w=30)-module.

The algorithm receives as its input promoter sequences of background, 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 < 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 < 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 is the number 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 more than one hit 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 be the list of all hits, ordered by position. We shall build a hash C of modules where C_{open} is a hash of active modules and their starting positions

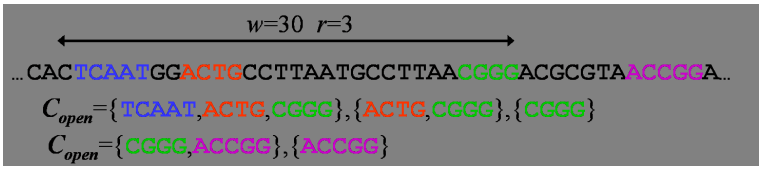

Figure 8.21: Instance of a (r=3,w=30)-module and possible instances of Copen.

The details of the algorithm are shown in Figure 8.22 (see [9]).

$\mathcal{C} \leftarrow \emptyset$ # 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 8.22: Source: [14] 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.

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

8.3.4 Motif finding tools

Definitions Motif(l,d) is 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 8.23.

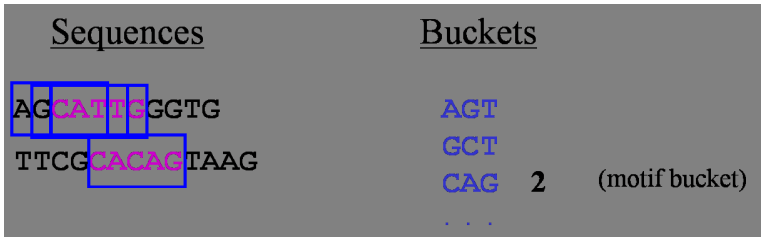

Figure 8.23: An example of random projection, with l=5, d=1, k=3, motif M="CATAG" and projection function h($x_1x_2x_3x_4x_5$)=$x_1x_2x_5$, The motif bucket is CAG. In the example, we can use any base for x_3 and x_4 and we look at all the sub-sequences that fall into the same bucket. And we find x_3 and x_4 according to the most frequent sub-sequences.

Analysis: Choosing proper 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

$$\alpha = \frac{\left(\begin{array}{c} l-d \\ k \end{array} \right)}{\left(\begin{array}{c} l \\ k \end{array} \right)}$$

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

$$B(\alpha, s, t') = \sum_{0 \leq i < s} \left(\begin{array}{c} t' \\ i \end{array} \right) \alpha^i (1-\alpha)^{t'-i}$$

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 principles 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 [8]) 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 [16])

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 8.16).
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 logarithm:

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

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:

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

$$(Eq.7.1)$$

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 $L_1 = \sum_{i=1\ldots n} \sum_{j=1,2} Z_{ij}$ 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

Homology is the similarity between different organisms due to shared ancestry. Homologous genes are most likely to have high sequence similarity. Considering homologous genes from different organisms is often useful; Inside the promoter region, the BSs are significantly more conserved. This can narrow the search for novel motifs, improving the rate of false hits and the overall performance (Figure 8.24).

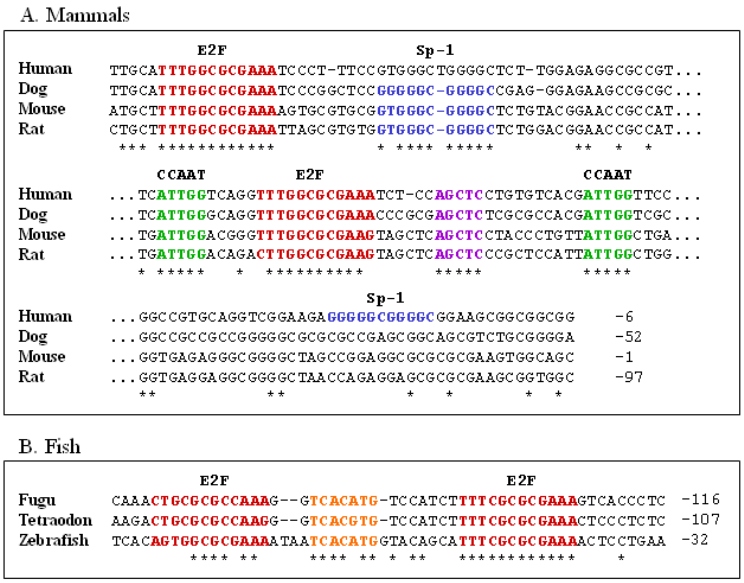

Figure 8.24: Promoter region of the gene MCM6 and its homologous genes. The BSs of several TFs are shown above. Note the remarkable similarity of BSs sequences and locations. Using homologous genes is often useful to discover BSs.

8.4 Promoter analysis - computational challenges

8.4.1 Promoter Analysis - Inherent problems

- Some BSs can bind more than one TF.
- Some TFs have multiple BSs in the same promoter.
- Some BSs are very far (5,000 nucleotide pairs or more) from the TSS.
- Most BSs are too short compared to the large promoter region. Random sequences are likely to be suspected as BSs.
- Promoters are very hard to model mathematically. Different structural signals exist (for example, G-C content), and it is hard to take everything into account.

8.4.2 Promoter analysis - Implementation problems

Data set is huge: about 500 billion consensus strings of length 10.

8.4.3 Promoter analysis - current status of motif discovery tools

- Extant tools perform reasonably well at finding known and novel motifs in organisms with short, simple promoters (e.g. yeast), and at identifying some of the known motifs

in complex species (e.g. TFs whose BSs are ussually close to the TSS), but often fail in other cases.

- Each tool is custom-built for a specific target score
- For a comparison of tools see [5]

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] 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 Research*, 13(5):773–780, 2003.

[4] Berger et al. Compact, universal DNA microarrays to comprehensively determine transcription-factor binding site specificities. *Nature Biotechnology*, pages 1429–1435, 2006.

[5] Tompa et al. Discovery of regulatory elements in vertebrates through comparative genomics. *Nature*, pages 1249–1256, 2005.

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

[7] Y. S. Kwon et al. Sensitive ChIP-DSL technology reveals an extensive estrogen receptor binding program on human gene promoters. *PNAS*, pages 4852–4857, 2007.

[8] 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.

[9] 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.

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

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

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

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

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

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

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