CS5238 Combinatorial methods in bioinformatics 2004/2005 Semester 1

Lecture 10: Motif Finding

Lecturer: Wing-Kin Sung *Scribe: Tan Yee Fan, Neo Shi Yong and Wang Gang*

10.1 Introduction

One of the big challenges in molecular biology is to understand the regulation of gene expression; which is the process in which a segment of DNA is being decoded to form a protein. From the DNA level to the protein level, we see that the protein coding genes, which occupy a small percentage of the genome, are being regulated in three levels. They are the *transcription control level*, the *post transcription control level* and the *post translation control level*. Transcription control determines when the protein transcription can start. Post transcription control determines if the transcription is successful or not (e.g. RNA silencing) and the types of RNA generated (e.g. splicing). And the control on protein level is covered in post translation control. All these above controls occur in the following two-step process:

1. *Transcription*: synthesis of a single-stranded RNA molecule using the DNA template (1 strand of DNA is transcribed).

2. *Translation*: conversion of a messenger RNA sequence into the amino acid sequence of a polypeptide (i.e., protein synthesis).

In this lecture, we will focus on transcription control, which is highlighted in Figures 10.1 and 10.2. We shall first understand the functions of the protein coding sequence and regulatory sequences. From our earlier knowledge, we know that every gene consists of a protein coding sequence, which might be contiguous or broken up, forming of a series of exons and introns. They normally begin with a START codon (`ATG`) and ends with a STOP codon (`TAA`, `TAG` or `TGA`). Apart from this, a gene must also have regulatory sequences associated with it. The use of regulatory sequences is listed as follows:

1. Controlling the time and phase information of the gene expressions:

   Different genes are expressed in different phases. For example, the yeast cell fission cycle is divided into four phases: G1 phase, S phase, G2 phase and M phase.

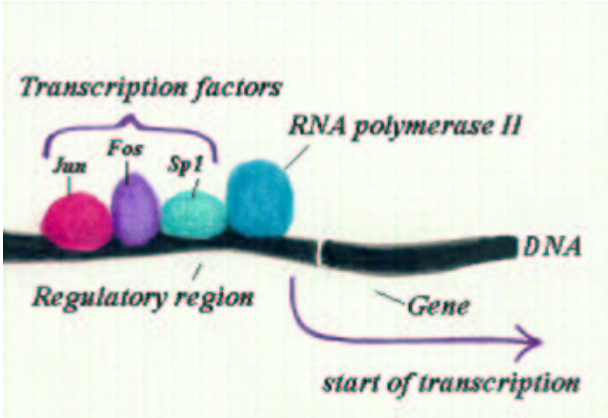


Figure 10.1: *Transcriptional Control I*

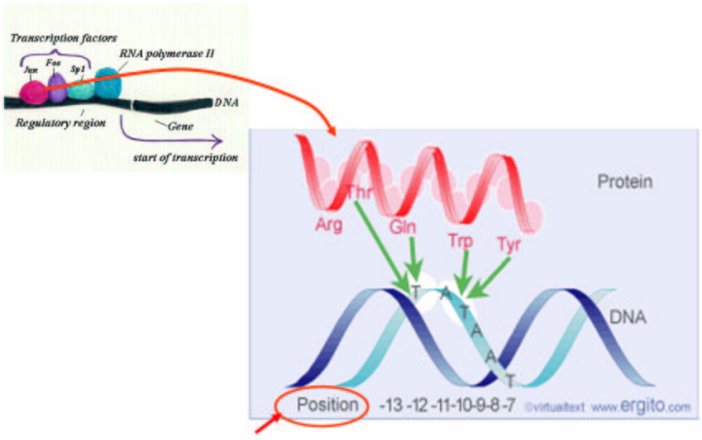


Figure 10.2: *Transcriptional Control II*

2. Controling the locality of the gene expressions. One example for such a case is tissue specific genes.

3. Controling the amount of gene expressions. For example, with enhancers, the gene expression is higher.

These regulatory sequences are stretches of DNA sequences which are not proteins sequences but binding sites for RNA polymerase and its accessory molecules. They also include a wide variety transcription factors. Together, the regulatory sequences with their bound proteins act as molecular switches that determine the activity state of the gene - e.g. *OFF* or *FULL-ON* or, more often, something in between. To start the transcription process for a particular gene, one or more transcription factors have to be bound to several specific regions, called binding

sites. These binding sites are located in the regulatory region of the gene and a single transcription factor can be bound to multiple binding sites. However, they must have similar length and DNA sequence pattern. We refer these binding sites as *motifs*.

Since the majority of the motifs are unknown to us, our task is to find such motifs. The discovery of motifs will allow the biologist to understand the varied and complex mechanism that regulated gene expression. Figure 10.3 shows examples of motifs, represented by the different shapes in the figure. Notice that the gene region is in red and promoter region is in yellow. For example, the transcription factor shown in the rectangle will interact with the *motifs* such as `tataaa`. (Note that binding sites are short, whose length may be up to 30 nucleotides.)

Figure 10.3: *Transcriptional control III*

10.2 Motif

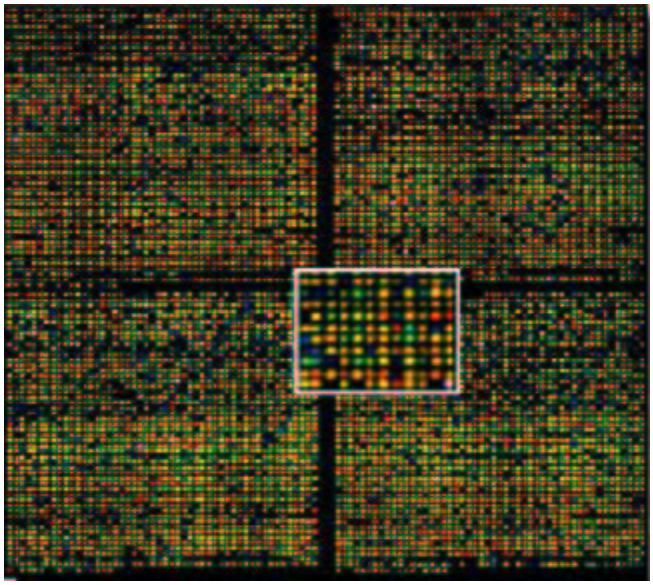
A motif is defined to be a short segment that occurs frequently in a DNA sequence, but it is not required to be an exact copy. This property of motif makes motif mining very difficult. In fact, motif finding problem is proven to be NP-Complete.

The steps of finding motifs are as follows:

- First, find a set of promoters which contain the same motif. Here, we will discuss two methods: *co-expressed genes method* and *chromatin immunoprecipitation data method*.
- Next, evaluate the motifs by experiment.
- Finally, look for the motif by using some computational methods.

10.2.1 Finding co-expressed genes through microarray

Co-expressed genes are genes that will be expressed together. They are likely to be regulated by the same transcription factors. Co-expressed genes can be identified through clustering of microarray data, which is shown in Figure 10.4.



http://www.sri.com/pharmdisc/cancer_biology/laderoute.html

Figure 10.4: *Microarray*

Microarray is an array which can have up to tens of thousands of single-stranded DNA attached. It is based on hybridization of a single-stranded DNA, labeled with a fluorescent tag to a complementary molecule attached to the chip. Microarray is used to detect the presence or the absence, of a particular type of DNA molecule in the test tube. We could use a microarray to find co-expressed genes. Based on the microarray, we can find genes that are up-regulated and down-regulated together (co-expressed genes). These genes are expected to be regulated by the same transciption factor.

10.2.2 Chromatin immunoprecipitation experiment

Chromatin immunoprecipitation, or ChIP, refers to a procedure to determine whether a given protein binds to a specific DNA sequence in vivo. Figures 10.5 and 10.6 show the process to detect the interaction between protein and DNA. The first step in a ChIP experiment is to break open cells and shear the DNA into small fragments. Then it binds antibodies specific to the DNA-binding protein

and isolates the complex by precipitation. Finally, it reverses the cross-linking to release the DNA and digest the proteins. The sequences extracted by ChIP experiments are expected to be bound by the targeted proteins. In other words, these sequences are expected to contain the binding sites.

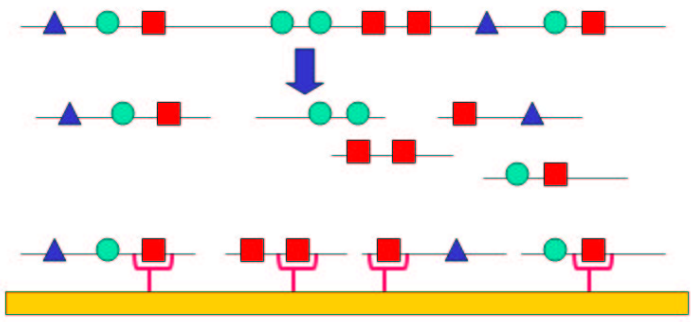


Figure 10.5: *Chromatin immunoprecipitation experiment I*

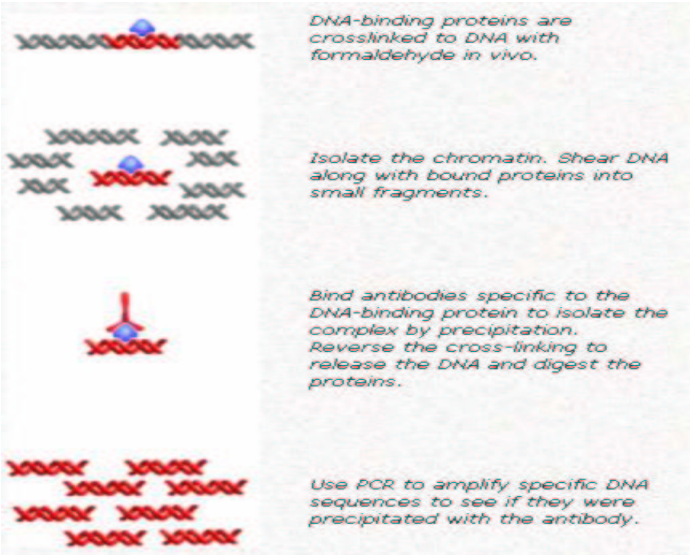


Figure 10.6: *Chromatin immunoprecipitation experiment II*

10.2.3 Evaluate the motif by experiment

Biological experiments to verify regulatory sites are tedious and time-consuming. One approach taken is to mutate different combinations of nucleotides until its functionality changes. However, this is very complex and time-consuming. Hence a series of computational methods have been proposed by computer scientists to reduce the time required for motif discovery.

10.3 Representing motifs

There are two common ways of representing motifs: by *consensus sequence* and by *position weight matrix*.

For both methods, we shall use the following set of candidate binding sites as illustration:

```
TATGAT
TATAAA
TATAAT
TAATAA
TATAAT
TATAAA
TATTAT
GATAAA
GATACT
TACGAA
GATACT
TAAGAT
TATAAA
```

The matrix below shows the frequency of occurrence of `A`, `C`, `G` and `T` in the different positions:

	1	2	3	4	5	6
A	0	12	2	8	11	6
C	0	0	1	0	2	0
G	3	0	0	3	0	0
T	10	0	10	2	0	7

10.3.1 Consensus sequence

Here, the consensus sequence has been calculated using the following rules: if the majority of symbols are a particular letter, then use that letter; if equal numbers of different residues are present, then show all the residues in the consensus.

For the above example, we can find that `T` is the majority symbol in location 1. Thus the 1st position of the consensus sequence is `T`. We can do it in same way in position 2, 3, 4 and 5. In position 6, we find that the number of symbol `A` and symbol `T` is nearly the same. So we obtain a consensus sequence `TATAA[AT]`, which implies that the consensus sequence can be either `TATAAA` or `TATAAT`.

10.3.2 Positional weight matrix

The positional weight matrix (PWM) can be generated from the matrix above by normalizing every column so that the sum of each column equals one:

	1	2	3	4	5	6
`A`	0	1	0.15	0.62	0.85	0.46
`C`	0	0	0.08	0	0.15	0
`G`	0.23	0	0	0.23	0	0
`T`	0.77	0	0.77	0.15	0	0.54

Effectively, the PWM gives the probability of each nucleotide occurring at each position of the motif sequence.

10.4 Methods of finding motifs

There are two main methods for finding motifs in a set of sequences. The first is to scan for known motifs in the biology literature. The second way is to use statistical or combinatorial methods to find motifs.

10.4.1 Scanning for known motifs

The biology literature has some known transcription factor binding sites, for example, the TRANSFAC database.

Given the set of input sequences, we can scan the known experimental transcription factor binding sites in the input sequences. However, the database is not exhaustive.

A list of known binding sites is shown in Table 10.1.

10.4.2 The TRANSFAC database

The TRANSFAC database [WCF01] is a collection of known transcription factors as well as their DNA binding sites and profiles, and is bundled together with a number of useful progam routines for identifying potential transcription factor binding sites or for localizing individual components in the regulatory network of a cell. It is accessible at the following URL: `http://www.gene-regulation.de/`

SITE_ID	FACTOR(S)	SITE	SYSTEM	SEQUENCE
S00655	BPV-E2	(E2)-EIIaE1	MAMM	GACGTAGTTTTCGCGC
S00906	(GR)	(GR)-Mo-MuLV	MAMM	AGAACAGATG
S01822	(Myogenin)	(Myogenin) CS	MAMM	TTGCACCTGTTNNTT
S00610	(SRF)	(SRF)-actin	MAMM/AMPHI	AAGATGCGGATATTGGCGAT
S00857	(Sp1)	(Sp1)-MT-I.1	MAMM	GGGGGCGG
S00859	(Sp1)	(Sp1)-MT-I.2	MAMM	TGCACTCCGCCC
S01187	(Sp1)	(Sp1)-TK.1	MAMM	CCCCGCCC
S01188	(Sp1)	(Sp1)-TK.2	MAMM	GGGGCGGCGCGG
S01026	(Sp1)	(Sp1)-U2snR.1	MAMM	GGGCGG
S01027	(Sp1)	(Sp1)-U2snR.2	MAMM	ACGCCC
S01028	(Sp1)	(Sp1)-U2snR.3	MAMM	GGGCGG
S00783	(TFIID/TBF)	(TFIID/TBF)-RS	MAMM	TATAAA
S00739	(TFIID/TBP)	(TFIID/TBP)-H2B1	ECHINO	TATAAATAG
S01171	unknown	16S/32S rRNA.1	PROK	TTTATATG
S01765	unknown	21-boxAl	PROK	GGCTCTTTA
S01763	unknown	21-boxAr	PROK	TGCTCTTTA
S01206	TT factor	28S RNA termination CS	MAMM	AGGTCGACCAGWWNTCCG
S01172	unknown	30S rRNA.IF3	PROK	AGGT
S01927	unknown	3MC-inducible-GST-Ya site	MAMM	CGTCAGGCATGTTGCGTGCA
S00845	60k-protein	60k-protein RS1	PLANT	GAATTTAATTAA

Table 10.1: *Some known binding sites*

Information in TRANSFAC is presented in six flat files. The largest files are SITE and FACTOR, which contain information on transcription factor binding sites in eukaryptoc genes and transcription factors respectively. The other four files are GENE, CELL, CLASS and MATRIX, with MATRIX having particular importance, since it represents DNA binding profiles for individual or groups of transcription factors.

10.4.3 Statistical and combinatorial approaches

If we want to discover new binding sites, we cannot use the previous method of scanning the known motifs from the databases. Hence, we need to use either the statistical or the combinatorial approach.

Some statistical approaches include the Gibbs sampler and expectation maximization.

Some combinatorial approaches include SP-STAR, random projection and WINNOWER.

These will be described in more detail in the subsequent sections.

10.5 Planted motif problem

In this section, we first define the *planted motif problem*. All the statistical and combinatorial approaches aim to solve this problem.

Next, we describe the *exhaustive pattern driven algorithm*, which is a brute-force algorithm that solves the planted motif problem in exponential time.

10.5.1 Definition

The planted motif problem is defined as follows:

Input:

- A set S of m sequences, each of length n.
- Two integers l and d, with $d < l < n$.

Output:

- A pattern M, such that every sequence in S contains a length l substring which can be transformed to M after at most d substitutions.

Such a pattern M is known as a (l, d)-*motif* of the set S of sequences.
As an example, we consider the following four sequences:

```
TAGTACTAGGTCGGACTCGCGTCTTGCCGC
CAAGGTCCGGCTCTCATATTCAACGGTTCG
TACGCGCCAAAGGCGGGGCTCGCATCCGGC
ACTCTGTGACGTCTCAGGTCGGGCTCTCAA
```

Then a $(15, 2)$-motif for the above sequences is `AGGTCGGGCTCGCAT`.

In 2000, Pevzner and Sze issued the *challenge problem*, stated as follows:

> Find a signal in a sample of sequences, each 600 nucleotides long and each containing an unknown signal (pattern) of length 15 with 4 mismatches. [PS00]

This highlights the difficulty in the motif finding problem, for example, the planted $(15, 4)$-motif problem in 20 sequences. However, in 2001, Buhler and Tompa claimed that for the planted motif problem with the same parameters (20 sequences, each 600 nucleotides long), finding $(14, 4)$-, $(16, 5)$- and $(18, 6)$-motifs are considerably more difficult [BT01].

10.5.2 Exhaustive pattern driven algorithm

Let $S = \{S_1, S_2, \ldots, S_m\}$ be a set of sequences.
For a length l pattern M, define $\delta(S_i, M)$ to be the minimum number of substitutions between S_i and M.
Note that $\delta(S_i, M)$ can be computed in $O(nd)$ time.
Define $score(M) = \sum_{i=1}^{m} \delta(S_i, M)$.

As an example, suppose we have

$$S_i = \texttt{TACGCGCCAAAGGCGGGGCTCGCATCCGGC}$$

and $M = \texttt{AGGTCGGGCTCGCAT}$. Then $\delta(S_i, M) = 2$.

As another example, suppose $S = \{S_1, S_2, S_3, S_4\}$, where

$$\begin{aligned} S_1 &= \texttt{TAGTACTAGGTCGGACTCGCGTCTTGCCGC} \\ S_2 &= \texttt{CAAGGTCCGGCTCTCATATTCAACGGTTCG} \\ S_3 &= \texttt{TACGCGCCAAAGGCGGGGCTCGCATCCGGC} \\ S_4 &= \texttt{CCTCTGTGACGTCTCAGGTCGGGCTCTCAA} \end{aligned}$$

and $M = \texttt{AGGTCGGGCTCGCAT}$. Then we have $\delta(S_1, M) = 2$, $\delta(S_2, M) = 2$, $\delta(S_3, M) = 2$ and $\delta(S_4, M) = 2$. Thus, $score(M) = 2 + 2 + 2 + 2 = 8$.

In the *exhaustive pattern driven algorithm* proposed by Waterman, the objective is to find the best motif M which minimizes $score(M)$. The brute-force algorithm is as follows:

1. Set $M_{opt} = \texttt{AA}\ldots\texttt{A}$.
2. For every length l pattern M from $\texttt{AA}\ldots\texttt{A}$ to $\texttt{TT}\ldots\texttt{T}$:
   (a) For $i = 1$ to m:
      i. Compute $\delta(S_i, M)$.
      ii. If $\delta(S_i, M) > d$, then try the next M.
   (b) Compute $score(M)$.
   (c) If $score(M) < score(M_{opt})$, then set $M_{opt} = M$.
3. Return M_{opt}.

The time and space analysis follows:

- There are 4^l different patterns for M.
- For each item M, we need to compute $\delta(S_i, M)$ for m sequences. Since computation of each $\delta(S_i, M)$ takes $O(nd)$ time, the time taken to compute $\delta(S_i, M)$ for m sequences is $O(mnd)$.
- In total, the time complexity is $O(mnd4^l)$.
- The space complexity is $O(mn)$, for storing the m sequences.

Unfortunately, the exhaustive pattern driven algorithm is only of theoretical interest because the time complexity is exhorbitantly large.

10.6 Statistical approaches

This section describes two statistical approaches for solving the planted motif problem, namely *Gibbs sampler* and *MEME*.

10.6.1 Gibbs sampler

The *Gibbs sampler* [LAB93] uses a randomized approach to iteratively improve a motif. Initially, a motif, represented as a PWM, is generated by selecting one random length l segment (substring) from each of the m sequences involved. Then the Gibbs sampler will repeatedly perform a *predictive update step* and a *sampling step* to iteratively improve the motif.

Predictive update step. In the predictive update step, one of the selected length l segment, chosen either at random or by some predetermined order, is removed from the selection.

Sampling step. Suppose that the removed segment is from sequence S_i. In the sampling step, we add a length l segment from S_i to the list of selected segments, which we will describe in more detail.

Firstly, from the remaining $m-1$ selected segments, we compute the PWMs for the motif model θ (which is used to represent the motif) as well as the background model θ_0 (which is used to represent the background distribution of the sequences). The PWM for θ_0 will be computed based on the nucleotides in the sequences that are outside the selected segments, and every column of the PWM for θ_0 will be the same.

Let $S_{i,j}$ be the length l segment at position j in S_i. We define W_θ to be the PWM for the model θ and $W_\theta(j)$ to be the value of the nucleotide at position j in W_θ. Also, we define

$$P(S_{i,j}|\theta) = \prod_{k=1}^{l} W_\theta(S_{i,j}[k], k),$$

and $P(S_{i,j}|\theta_0)$ is defined analogously.

For each segment $S_{i,j}$, we compute a weight $A_{i,j}$, defined to be

$$A_{i,j} = \frac{P(S_{i,j}|\theta)}{P(S_{i,j}|\theta_0)}.$$

The weight $A_{i,j}$, after normalization, gives the probability of choosing segment $S_{i,j}$ for addition into the set of selected segments. In other words, the probability of choosing $S_{i,j}$ is $\frac{A_{i,j}}{\sum_j A_{i,j}}$.

When using the Gibbs sampler, different runs will give different results. Normally, the motif is chosen from the best out of a number of runs.

Also, there are various variants of the standard Gibbs sampling approach. These include the Gibbs motif sampler, AlignACE and BioProspector.

10.6.2 MEME

MEME [BE94] is one of the most popular software for finding motifs. It is implemented using the MM algorithm, which is an extension of the expectation maximization technique for finite mixture models.

The idea of the MM algorithm is to find an initial motif, and then iteratively executes the expectation and maximization steps to improve the motif until it cannot be improved beyond a certain threshold or a certain number of iterations has been reached.

In the MM algorithm, we have a motif model θ that is used to represent the motif, as well as a background model θ_0 that is used to represent the distribution of the sequences involved. Both the motif model and the background model can be represented using PWMs, as in the following examples:

θ	1	2	3	4	5
A	0.2	0.8	0.1	0.7	0.8
C	0	0.1	0.2	0	0.1
G	0.1	0.1	0.1	0.2	0.1
T	0.7	0	0.6	0.1	0

θ_0	1	2	3	4	5
A	0.25	0.25	0.25	0.25	0.25
C	0.25	0.25	0.25	0.25	0.25
G	0.25	0.25	0.25	0.25	0.25
T	0.25	0.25	0.25	0.25	0.25

Note that all the columns in the PWM of θ_0 are the same.

Expectation. The expectation step computes the probability of finding the actual motif at every position in every sequence.

For every sequence S_i, a sliding window of length l is used. For every length l contiguous segment (substring) $S_{i,j}$ in the sequence, we compute $P(S_{i,j}|\theta)$ as well as $P(S_{i,j}|\theta_0)$ from the PWMs. If we define W_θ to be the PWM for the model θ and $W_\theta(i,j)$ to be the entry for nucleotide i at position j in W_θ, then

$$P(S_{i,j}|\theta) = \prod_{k=1}^{l} W_\theta(S_{i,j}[k], k).$$

Then we compute the likelihood ratio $p_{i,j}$ for $S_{i,j}$:

$$p_{i,j} = \frac{P(S_{i,j}|\theta)}{P(S_{i,j}|\theta_0)}.$$

As an example, suppose we have the sequence $S_1 =$ `TGATATAACGATC` and the PWMs for θ and θ_0 as above. This sequence has the follwing possible positions for motifs:

```
TGATATAACGATC
TGATA          p1,1
 GATAT         p1,2
  ATATA        p1,3
   TATAA       p1,4
    ATAAC      p1,5
     TAACG     p1,6
      AACGA    p1,7
       ACGAT   p1,8
        CGATC  p1,9
```

For the first segment `TGATA`, we compute $p_{1,1}$ using the PWM as follows:

θ	1	2	3	4	5
A	0.2	0.8	**0.1**	0.7	**0.8**
C	0	0.1	0.2	0	0.1
G	0.1	**0.1**	0.1	0.2	0.1
T	**0.7**	0	0.6	**0.1**	0

$$p_{1,1} = \frac{0.7 \times 0.1 \times 0.1 \times 0.1 \times 0.8}{0.25 \times 0.25 \times 0.25 \times 0.25 \times 0.25}.$$

Maximization. In the maximization step, we refine the PWM of the motif given the probabilities for every position and every sequence.

The refinement is done as follows: for each position, and for each nucleotide, the entry in the PWM for θ is the of the probabilities for the segments having that nucleotide. The entries of the PWM are then normalized such that the sum of the entries for each column is 1.

More formally, let

$$q_{i,j}(x,k) = \begin{cases} p_{i,j} & \text{if } S_{i,j}[k] = x, \\ 0 & \text{otherwise.} \end{cases}$$

We then update W_θ as follows:

$$W_\theta(x,k) = \frac{\sum_{i,j} q_{i,j}(x,k)}{\sum_{i,j} p_{i,j}}.$$

Using the same example as above, the entry for nucleotide `T` at position 1 of the refined PWM for θ will be

$$\frac{p_{1,1} + p_{1,4} + p_{1,6}}{p_{1,1} + p_{1,2} + p_{1,3} + p_{1,4} + p_{1,5} + p_{1,6} + p_{1,7} + p_{1,8} + p_{1,9}}.$$

10.7 Combinatorial approaches

We will describe three combinatorial approaches *SP-STAR*, *WINNOWER* and *random projection* in detail which prove to give better performance than some of the most popular signal finding algorithms mention earlier, such as Gibbs Sampler and MEME, when applied to simulated samples with uniform background distribution.

10.7.1 SP-STAR

SP-STAR [PS00] is proposed by Pevzner and Sze. First, it chooses a suitable scoring function to access the goodness of a motif. Then, for each l-mer appearing in the sample, find its best instance in each sequence and collect these instances together to form an initial motif. Employ a local improvement heuristic to improve each initial motif.

Definitions

- Consider a set of length-l sequences $w_1, w_2, \ldots, w_m$. Define

$$SPscore(w_1, w_2, \ldots, w_m) = \sum_{i,j} \delta(w_i, w_j),$$

  where $\delta(x, y)$ is the hamming distance between x and y.

Algorithm

1. Let W be the set of length-l words in all m sequences S_1, S_2, ..., S_m.
2. For any length-l word $w \in W$:
   (a) Find the best match w_1, w_2, ..., w_m in each of the m sequences.
   (b) Get the consensus word, by counting the most frequently appearing word in each column. Let w be the consensus word.
   (c) Repeat steps (a) and (b) until $SPscore(w_1, w_2, \ldots, w_m)$ cannot be further improved.

Example

- Given a string v, locate the string w_i which is closest to v in each sequence (Figure 10.7).
- Use the most frequent letter in each position to define a new majority string (Figure 10.8).
- Repeat until there are no more changes in the string v.

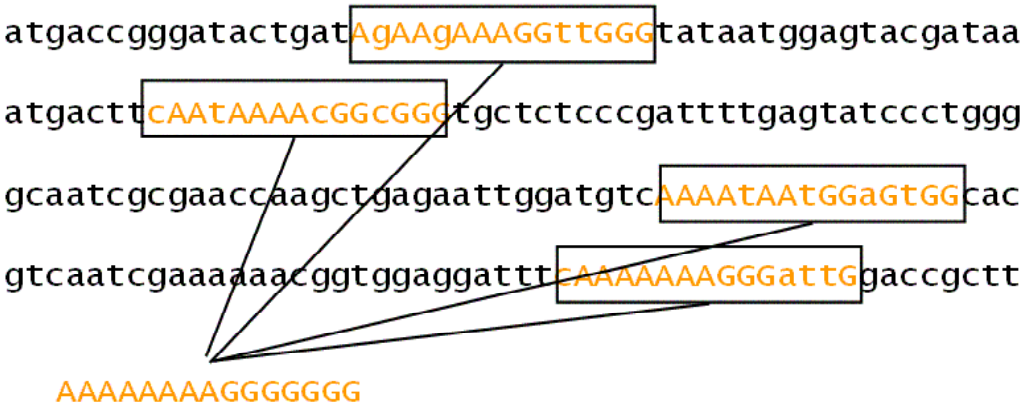


Figure 10.7: *Mapping of the most similar strings*

	AGAAGAAAGGTTGGG
	CAATAAAACGGCGGG
	AAAATAATGGAGTGG
	CAAAAAAAGGGATTG
Majority string	AAAAAAAAGGGAGGG

Figure 10.8: *Getting consensus from strings*

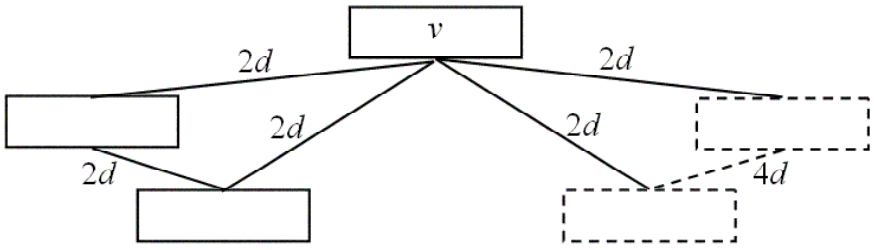


Figure 10.9: *Illustration of distance between strings*

The main reason why SP score is chosen over STAR is that it is able to give a better initial estimate of the goodness of a string. For subtle signals, typical distance between signal instances is less than or equal to $2d$, while typical distance between two random best instances from v can be as large as $4d$, where d is the number of mismatches, as shown in Figure 10.9.

10.7.2 WINNOWER

WINNOWER [PS00] is a graph-theoretic approach which represents a motif as a large clique (complete subgraph) and attempt to solve the clique problem efficiently by filtering.

Problem Reduction

Given a set of sequences $S = \{S_1, S_2, \ldots, S_m\}$ and suppose we are looking for a (l, d)-motif. We construct a graph G as follows:

- Every vertex in G corresponds to a length-l word in S.
- Consider two words x and y appear in two different sequences in S. x and y are connected by an edge if their hamming distance is at most $2d$.

Note that G is a m-partite graph. An example is shown in Figure 10.10.

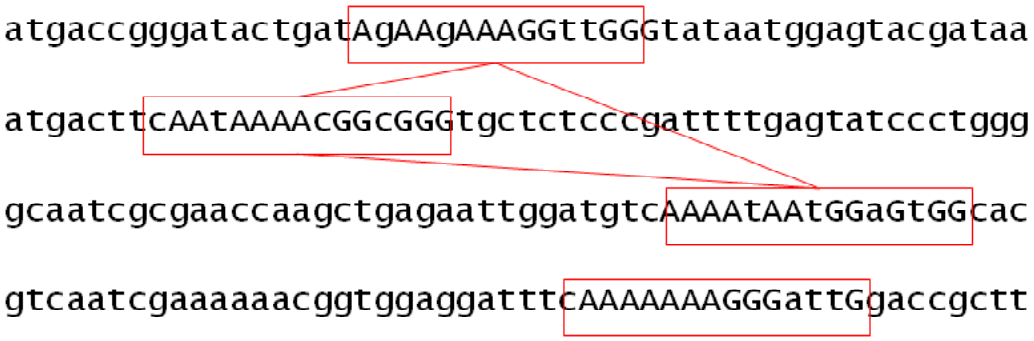


Figure 10.10: *3-clique*

Notice that the problem of finding a (l, d)-motif corresponds to finding a clique of size m, where m is the number of strings. Thus, the problem of finding motifs is reduced to the problem of finding large cliques. However, finding cliques is a NP-complete problem. Therefore WINNOWER proposes a method to filter edges which definitely do not belong to any large cliques.

Definitions

- A vertex v is a *neighbor* of a clique C if it is connected to every vertex in this clique (i.e., $C \cup \{v\}$ is a clique).
- A clique is called *extendable* if it has at least one neighbor in every part of the multipartite graph G.
- An edge is called *spurious* if it does not belong to any extendable clique of size k.

WINNOWER is an iterative algorithm that converges to a collection of extendable cliques by filtering out spurious edges.

Algorithm

1. Construct a graph G by:
    - Consider two words x and y appear in two different sequences in S. x and y are connected by an edge if their hamming distance is at most $2d$.
2. Filtering of spurious edges:
    (a) Filtering weak vertices:
        - Vertices that are not supported by a neighbor in every part of G are filtered out.
    (b) Filtering weak edges:
        - Unsupported edges are removed.
    (c) Filtering weak triangles.
    (d) If the computation allow, filter more.

Example

- An example is illustrated in Figure 10.11 and Figure 10.12.

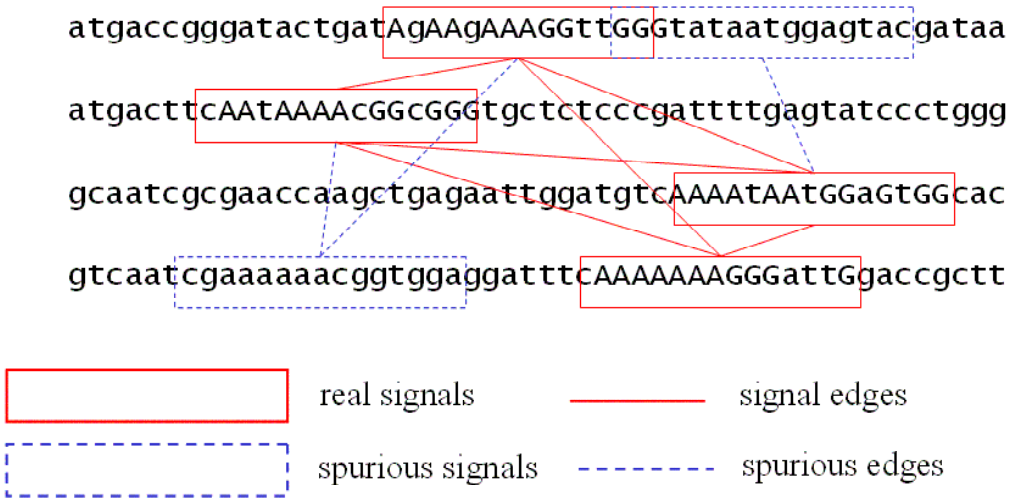


Figure 10.11: *Picture showing cliques in a string*

Figure 10.12: *After filtering*

10.7.3 Random projection

The *random projection* [BT01] algorithm chooses a projection by selecting k out of l positions at random, then each length-l string is *hashed* into buckets based on these k positions.

Intuition

Some instances of a motif agree on a subset of positions. With appropriate k, these instances will be grouped together.

Consider the following three sequences:

ATAATTCGCT
ACACTTCTCT

ACAGTTATCT

The sequences are actually $(10, 2)$-motifs. Ideally, if we are able to choose positions 1, 3, 5, 6, 9 and 10, then we can actually hash the strings into the same bucket and conclude that the strings which are in the bucket to be candidate motifs.

Motif refinement

Since real motif instances are likely to be put together as a large group, we can consider only large groups. Within each group, the selected k positions are already fixed. Use information in the other $\binom{l}{k}$ positions as a starting point of an iterative motif finding algorithm such as the Gibbs sampler or MEME. These algorithms typically work much better when given a good starting point. For example, random projection provides such good starting points.

Parameter selection

It is very important to choose an appropriate projection size k. However, there are some conflicting goals:

- k has to be small so that a significant number of motif instances are grouped together under the projection.
- k cannot be so small that non-motif instances are grouped together as motif instances. The random projection algorithm can be run *multiple* times.
- The best motif from these runs is taken to be the answer.

In the paper, Buhler and Tompa suggested setting $k = l - d - 1$.

10.7.4 Other approaches

Besides the three algorithms mentioned earlier, there are also a number of other algorithms which have been developed to tackle the problem of motif finding. We will mention them briefly here.

Multiprofiler

Based on the (l, d)-motif model, use the neighborhood of a candidate pattern to find "spelling" errors which prevent it from being the correct motif. It is a heuristic which starts from strings in the sample.

Patternbranching

Use a branching heuristic based on the (l, d)-motif model, starting from strings that appear in the sample and changing one letter at a time to search for a correct motif.

Finding profiles instead of patterns

Represent a motif as a profile (where each position is represented as a probability distribution of letters from $\{\mathtt{A}, \mathtt{C}, \mathtt{G}, \mathtt{T}\}$) rather than a simple pattern.

There are other recent methods apart from those mentioned above. Two examples are [SLC04] and [E04].

10.7.5 General problems of combinatorial approaches

Most of the recent combinatorial approaches do not consider statistical significance of motifs very carefully. Although these new approaches confirm to have better performance over older approaches on simulated samples, it has not been shown that they have significant advantages on real biological samples.

In practice, motif finding algorithms have to take into account characteristics of real input samples. These include:

1. Motifs with unknown length.
2. Samples with biased nucleotide composition.
3. Corrupted samples (not every sequence contains a motif).
4. Regulatory sites can lie on either DNA strand.

References

[WCF01] Wingender E, Chen X, Fricke E, Geffers R, Hehl R, Liebich I, Krull M, Matys V, Michael H, Ohnhauser R, Pruss M, Schacherer F, Thiele S and Urbach S. The TRANSFAC system on gene expression regulation. *Nucleic Acids Research*, 29(1):281-283, 2001.

[LAB93] Lawrence C E, Altschul S F, Boguski M S, Liu J S, Neuwald A F and Wootton J C. Detecting subtle sequence signals: a Gibbs sampling strategy for multiple alignment. *Science*, 262, 208-214, 1993.

[BE94] Bailey T L and Elkan C. Fitting a mixture model by expectation maximization to discover motifs in biopolymers. *Proceedings of the Second International Conference on Intelligent Systems for Molecular Biology*, 28-36, 1994.

[PS00] Pevzner P A and Sze S H. Combinatorial approaches to finding subtle signals in DNA sequences. *Proceedings of the Eighth International Conference on Intelligent Systems for Molecular Biology (ISMB 2000)*, 269-278, 2000.

[BT01] Buhler J and Tompa M. Finding motifs using random projections. *Proceedings of the Fifth Annual International Conference on Research in Computational Molecular Biology, RECOMB-01*, 69-76, 2001.

[SLC04] Sze S H, Lu S and Chen J. Integrating sample-driven and pattern-driven approaches in motif finding. *Lecture Notes in Computer Science/Lecture Notes in Bioinformatics (WABI 2004)*, 438-449, 2004.

[E04] Eskin E. From profiles to patterns and back again: a branch and bound algorithm for finding near optimal motif profiles. *Proceedings of the Eighth Annual International Conference on Research on Computational Molecular Biology, RECOMB-04*, 115-124, 2004.