6.047 / 6.878 Computational Biology: Genomes, Networks, Evolution
Fall 2008

1 Probabilistic Pattern Finding

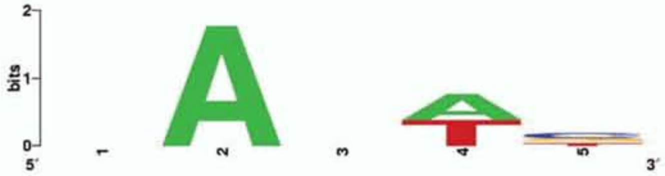
Many interesting motifs in biology are *degenerate*. We want to be able to identify and appropriately represent these motifs.

1.1 Representing Probabilistic Patterns

One of the most common and simplest ways to represent these motifs is using a position frequency matrix (PFM). In a PFM each position is modeled as a distribution over the possible characters.

	1	2	3	4	5
A	0.4	1.0	0.3	0.5	0.0
T	0.3	0.0	0.2	0.5	0.3
G	0.2	0.0	0.3	0.0	0.3
C	0.1	0.0	0.2	0.0	0.4

To visualize and compare it more easily, we can also display a PFM as a logo:



2 Expectation Maximization for Motif Finding

For EM motif finding we assume we are only given the size of the motif and we need to find it in a set of sequences. We also need a background probability so that we can evaluate when a motif matches a sequence.

First, we will begin with some definitions:

- Z_{ij}: the probability that the motif in sequence i starts at position j.
- S_{ij}: sequence i's jth character.
- M_{ij}: the motif's probability of having character i in position j. (the PFM). Let $M_{i,0}$ denote the background probabilities.

The basic EM approach is as follows:

```
set initial values for M (randomly)

repeat until M stops changing:
    compute Z from n (E-Step)
    compute M from Z (M-Step)

return M, Z
```

2.1 The E-Step

First, consider how to compute the probability of a sequence given the motif and where the motif occurs in the sequence:

$$P(S_i|Z_{ij}=1,M)=\prod_{k=1}^{j-1}M_{S_{ik},0}\prod_{k=j}^{j+W-1}M_{S_{ik},k-j+1}\prod_{k=j+W}^{L}M_{S_{ik},0}$$

where $\prod_{k=1}^{j-1}M_{S_{ik},0}\prod_{k=j+W}^{L}M_{S_{ik},0}$ gives the probability of the characters not part of the motif and $\prod_{k=j}^{j+W-1}M_{S_{ik},k-j+1}$ gives the probability of the characters that are. We see an example of this in the lecture notes.

Now, we can compute Z_{ij} if we have M:

$$Z_{ij}=\frac{P(S_i|Z_{ij}=1,M)P(Z_{ij}=1)}{\sum_{k=1}^{L-W+1}P(S_i|Z_{ik}=1,M)P(Z_{ik}=1)}$$

which follows from Bayes' rule on $P(Z_{ij}=1|S_i,M)$. If we assume that all positions are equally likely to contain a motif (which is sometimes the case), we have:

$$Z_{ij}=\frac{P(S_i|Z_{ij}=1,M))}{\sum_{k=1}^{L-W+1}P(S_i|Z_{ik}=1,M)}$$

Otherwise, we would have to provide the Z_{ij} priors.

2.2 The M-Step

Now we want to compute M (the position weight matrix) using Z (the starting location of each motif). Rather than just align the most likely positions in each sequence (which would be similar to the Viterbi algorithm), we want to let each position contribute the to PFM with weight proportional to the probability of the motif starting there. This is analogous to the Baum-Welch algorithm for HMM training. Taking the best position is often not wise because the best position may not be representative of all the sequences that match well.

Thus, we have

$$M_{c,k}=\frac{n_{c,k}+d_{c,k}}{\sum_b(n_{b,k}+d_{b,k})}$$

where $d_{c,k}$ is essentially a prior to make sure none of the $M_{c,k}$s are 0 (this is a parameter) and $n_{c,k}$ is a probabilistic count of the number of times character c is found in position k of the motif (we also perform this computation for position 0, i.e. the background).

2.3 ZOOPS Model

Often times we cannot really assume that each sequence has a match to the motif (e.g. co-expressed genes, which are easily to identify, may not always be co-regulated). Consequently, the EM algorithm has been extended with a Zero or One Occurrence Per Sequence (ZOOPS) model. We do this by incorporating the single variable λ which indicates the prior probability of any position matching the motif. Then we have:

$$Z_{ij}=\frac{P(S_i|Z_{ij}=1,M))\lambda}{(1-(L-W+1)\lambda)P(S_i|Z_{i0}=1,M)+\lambda\sum_{k=1}^{L-W+1}P(S_i|Z_{ik}=1,M)}$$

where $Z_{i0}=1$ indicates the sequence contains no motif. The update rule for M does not change; λ is updated by taking an average of Z_{ij}.

3 Gibbs Sampling

Gibbs sampling may be thought of as a stochastic analog of EM. It is a Markov Chain Monte Carlo algorithm. Like other Monte Carlo algorithms, Gibbs sampling will occasionally try a less than optimal alignment in order to ensure it looks around the parameter space better. This makes Gibbs sampling less likely to fall into a local maxima.

Now, the basic algorithm is

```
let $a_i$ be a random position in each sequence

repeat until convergence
    pick a sequence $S_i$
    estimate $p$ using all alignments except $S_i$'s
    sample a new position value for $a_i$

return a,p
```

Some notes: you can add pseudo counts in order to prevent a sequence from having 0 probability. Also, to compute how well a motif matches a sequence you should compute the probability of the motif generating the sequence divided by the probability of the background computing the sequence.

Ultimately, Gibbs sampling is less dependent on the initial parameters and is more versatile (heuristics are easily added), but more dependent on all sequences having the motif and less systematic about its search.

3.1 A Better Background

Repeat DNA can easily be confused as a motif. To deal with this, a background model that is more complex than what we have talked about can be used. But there are obvious computational ramifications of this (longer running times).