CSE527 Lecture Notes: Motifs

Paul Gauthier
gauthier@cs.washington.edu

November 5, 2003

Class Monday 11/10

Zizhen Yao talk on "Functional Prediction in E. Coli Based on Heterogenous Data".

Review from Last Lecture

The $TATA$ box is found in most organisms. We covered the *weight matrix* motif model which assumes independent selection of each character in the motif and the background models.

Weight Matrix Example

Consider the 8 sequences of 3 characters. ATG is the start codon for mRNA transcription, so this is a realistic (but not real) example of a motif (GTG also occurs sometimes as the start codon).

A	T	G
A	T	G
A	T	G
A	T	G
A	T	G
G	T	G
G	T	G
T	T	G

Which produces the following profile:

	1	2	3
A	0.625	0	0
C	0	0	0
G	0.250	0	1
T	0.125	1	0

Assuming a background model where each character has an equal probability of 1/4, we obtain the following Log Likelihood Ratios: Notice that the log likelihood ratio table has $-\infty$ wherever the profile matrix had a 0.

$\log_2 \frac{f_{x_i,i}}{f_{x_i}}, f_{x_i} = \frac{1}{4}$

	1	2	3
A	1.32	$-\infty$	$-\infty$
C	$-\infty$	$-\infty$	$-\infty$
G	0	$-\infty$	2
T	-1	2	$-\infty$

Nonuniform Background

E. Coli has DNA consisting of approximately equal amounts of A, C, G, T. But M. jannaschii has approximately 68% A, T and 32% G, C. Consider the same motif profile as before, but with a background model with $f_A = f_T = \frac{3}{8}$ and $f_C = f_G = \frac{1}{8}$.

	1	2	3
A	0.737	$-\infty$	$-\infty$
C	$-\infty$	$-\infty$	$-\infty$
G	1	$-\infty$	3
T	-1.58	1.42	$-\infty$

The table indicates, for example, that a G in position 3 is $2^3 = 8$ times more likely to be part of the motif than the background model.

Because G is now less likely in the background model we see that G is positions 1 and 3 have increased their log likelihood ratios compared to the uniform background table.

Conversely, T is now more likely in the background model, so a T in position 2 has a reduced log likelihood ratio (down from 2 in the previous table).

How "Informative" is a WMM?

Recall the formula for Relative Entropy

$H(P||Q) = \sum_x P(x) \log_2 \frac{P(x)}{Q(x)}$

If x are fixed length sequences of characters, then $P(x)$ is the probability of x occurring according to the motif WMM, and $Q(x)$ is the probability of x occurring according to the background model.

The term $L_x = \log_2 \frac{P(x)}{Q(x)}$ is just the Log Likelihood Ratio for the sequence x. So Relative Entropy can be viewed as:

$H(P||Q) = \sum_x P(x) L_x$

That is, the expected Log Likelihood Ratio of a character string chosen from the motif's distribution.

For the WMM model, we can show:

$H(P||Q) = \sum_{i=1}^{n} H(P_i||Q_i)$

Where P_i, Q_i are distributions of the i^{th} position. This follows from the assumption of independence.

Recall our example WMM:

	1	2	3
A	0.625	0	0
C	0	0	0
G	0.250	0	1
T	0.125	1	0

Which produces the following relative entropy scores:

Uniform	1	2	3	Total Relative Entropy $H(P\|\|Q) = \sum_{i=1}^{3} H(P_i\|\|Q_i)$
A	1.32	$-\infty$	$-\infty$	
C	$-\infty$	$-\infty$	$-\infty$	
G	0	$-\infty$	2	
T	-1	2	$-\infty$	
Relative Entropy, $H(P_i\|\|Q_i)$	0.701	2	2	4.701
Nonuniform	1	2	3	
A	0.737	$-\infty$	$-\infty$	
C	$-\infty$	$-\infty$	$-\infty$	
G	1	$-\infty$	3	
T	-1.58	1.42	$-\infty$	
Relative Entropy, $H(P_i\|\|Q_i)$	0.512	1.42	3	4.932

It is worth noting that Relative Entropy is always non-negative, so adding additional characters to your model will result in higher scores. Care must be taken when comparing relative entropy scores of sequences of differing lengths to adjust for this bias towards longer sequences.

Pseudo-Counts

Are the $-\infty$ entries in the log likelihood ratio tables a problem? If you are *certain* that a given residue *never* occurs in a particular position of the motif, then yes $-\infty$ is reasonable. It will ensure that any candidate motif with that residue in that position will have an infinitely negative score.

In most cases, 0 counts and $-\infty$ log likelihood ratios are a result of taking a small sample – a larger sample would probably have uncovered a motif with that residue.

The typical fix is to add a small constant (0.5, 1, 2) to all the observed counts to produce a *pseudo-count*. Note that if you have many observations, the pseudo count adjustments will have a relatively small affect on the frequently observed data items.

This approach has some justification from a Bayesian viewpoint. You have a prior belief that there is almost no chance that a given motif will *never* have a certain base in a certain position. The small constant added to obtain pseudo-counts reflects that prior belief. A small sample resulting in an observed count of 0 can only influence that prior belief to a small degree.

Question: Would anyone ever use negative log likelihood ratio cutoffs during a motif scan? Perhaps, for example if the scan was to select a collection of (possibly weak) candidates which were to then undergo further analysis. For example, you want to get all possible $TATA$-box candidates and then subject the downstream sequence to some analysis to detect genes.

Questions

Given aligned instances of motifs, how do you build a model? Use frequency counts to build your WMM, as above, optionally with pseudo counts.

Given a model, how do you find (probable) instances of the motif? Scan the candidate sequence, scoring each substring with your model to obtain log likelihood ratios. Higher scoring substrings are likely to be motifs.

Given unaligned strings thought to contain a motif, how do you find it? A good example is locating upstream promoter regions in a collection of sequences believed to be co-regulated from a micro-array clustering experiment. That's the next topic.

Motif Discovery: Three Approaches

There are manjy approaches, but I'll only talk about the following three:

1. Greedy Search
2. Expectation Maximization
3. Gibbs Sampler

Finding a site of maximum relative entropy in a set of unaligned sequences is NP-Hard (Akutsu).

Motif discovery can also be used to find motifs in proteins.

Greedy Algorithm (Hertz & Stormo)

Input: Sequences $s_1, s_2, ..., s_k$, motif length l, a *breadth* value d and background model.

Algorithm:

1. Create a singleton set with each length l subsequence of each of $s_1, s_2, ..., s_k$.
2. For each set retained, add each possible length l subsequence not already present.
3. Compute relative entropy of each set and return d best.
4. Repeat until each set has k strings.

This greedy algorithm has all the usual problems with getting stuck in local minima, etc.

Expectation Maximization, MEME (Bailey & Elkan)

Input: Sequences $s_1, s_2, ..., s_k$, motif length l and background model. Again, we assume 1 instance per sequence, although variants to handle more than 1 are possible.

Observed data: the sequences s_i
Parameters: the WMM Θ
Hidden data: Where is the motif? $Y_{ij} = \begin{cases} 1 & \text{if it starts at position } j \text{ in sequence } i \\ 0 & \text{otherwise} \end{cases}$

If you were given the WMM, it would be easy to scan and locate the motif locations. Similarly, if you were given the motif locations (an alignment of the sequences) it would be easy to compute the WMM. As usual, the EM algorithm alternates between these two.

Expectation Step

Given a WMM Θ and the input sequences s_i, compute expected value of the hidden variables Y_{ik}.

$$\begin{aligned} \hat{Y}_{ik} &= E(Y_{ik}|s_i, \Theta) \\ &= P(Y_{ik} = 1|s_i, \Theta) && (E = 1 \times P(1) + 0 \times P(0)) \\ &= P(s_i|Y_{ik} = 1, \Theta)\frac{P(Y_{ik}=1|\Theta)}{P(s_i|\Theta)} && \text{(Bayes' rule)} \\ &= cP(s_i|Y_{ik} = 1, \Theta) \\ &= c'\Pi_{j=1}^{l} P(s_{i,k+j-1}|\Theta) \end{aligned}$$

We can replace the fraction $\frac{P(Y_{ik}=1|\Theta)}{P(s_i|\Theta)}$ with a constant c because we assume the motif is equally likely anywhere in the sequence. We fix c' so that $\sum_k \hat{Y}_{ik} = 1$.

Maximization Step

Given parameter Θ^t at the t^{th} iteration, find a new Θ which maximizes the expected value of the full data likelihood. For simplicity, we assume all s_i are of the same length, and let $n = |s_i| - l + 1$.

$$\begin{aligned}
Q(\Theta|\Theta^t) &= E_{Y\sim\Theta^t}[\log P(s, Y|\Theta)] \\
&= E_{Y\sim\Theta^t}[\log \textstyle\prod_{i=1}^{k} P(s_i, Y_i|\Theta)] \\
&= E_{Y\sim\Theta^t}[\log \textstyle\prod_{i=1}^{k} (P(s_i|Y_i, \Theta) \cdot P(Y_i|\Theta))] \\
&= E_{Y\sim\Theta^t}[\log \textstyle\prod_{i=1}^{k} (P(s_i|Y_i, \Theta)/n)] \\
&= E_{Y\sim\Theta^t}[\log \textstyle\prod_{i=1}^{k} ((\prod_{j=1}^{n} P(s_i|Y_{ij} = 1, \Theta)^{Y_{ij}})/n)] \\
&= E_{Y\sim\Theta^t}[\textstyle\sum_{i=1}^{k} (\sum_{j=1}^{n} Y_{ij} \log P(s_i|Y_{ij} = 1, \Theta)))] - k \log n \\
&= \textstyle\sum_{i=1}^{k} \sum_{j=1}^{n} E_{Y\sim\Theta^t}[Y_{ij}] \log(P(s_i|Y_{ij} = 1, \Theta)) - k \log n \\
&= \textstyle\sum_{i=1}^{k} \sum_{j=1}^{n} \widehat{Y}_{ij} \log(P(s_i|Y_{ij} = 1, \Theta)) - k \log n
\end{aligned}$$

The second line above follows from the assumption that the k strings s_i are independent. The third line follows from the definition of conditional probability, and the fourth from the assumption that all motif positions are equally likely *a priori*. The fifth line is perhaps the most subtle. Each Y_i is completely determined by the unique j for which $Y_{ij} = 1$, and since all other Y_{ij} are zero, the product over j reduces to the desired single term $P(s_i|Y_{ij} = 1, \Theta)$ for the unique j such that $Y_{ij} = 1$. The advantage of this is that distributing the log in the next line brings the Y_{ij} out of the exponents, and we end up with a sum of easily computed terms, weighted by the expected values of the Y_{ij}, exactly what we computed in the E-step. (This trick of using indicator variables in exponents is common in EM algorithms.)

Finally, we're left with the problem of finding the Θ that maximizes this expression. We leave it as an exercise to show that it is maximized by the weight matrix model Θ obtained by "counting" frequencies in the alignment, where counts are $\hat{Y}_{ij}$. Essentially, we build the model by doing frequency counts as usual, but weighting them by the $E(Y_{ij})$ (the expectation that the motif starts at each location).

In a sense, this is again a greedy algorithm, and definitely still has the possibility of converging on a local but not global maximum; hopefully, the probalilistic weighting over all possible motif positions reduces this danger.

Initialization

1. Buy a supercomputer (ie: the San Diego Super Computer center).
2. Try *every* motif-length substring from the input sequences as the initial Θ (WMM) weighted to 80% (evenly distribute the other 20% across the other bases).
3. Run a few iterations of E-M from each starting Θ.
4. Run the best few to completion.

Question: What if the sequences contain repeated subsequences other than the one which is being sought? That is very common, since DNA contains a lot of repetitive data (like the ALU repeat, for example). A program called *RepeatMasker* is commonly used to preprocess sequences to remove all commonly known repeats.

Gibbs Sampler (Next Lecture)

Given an initial alignment of k l-length sequences (one from each of the k input sequences):

1. Throw out one of them at random, say the one from sequence s.
2. Rescan sequence s (using the remaining $k - 1$ aligned sequences to build a WMM) and compute the likelihood of that each l-length substring was generated by the motif.

3. Randomly select an l-length substring from s to replace the evicted string. Each substring's chance of being selected is weighted by its probability of being generated by the motif (as determined by the WMM).

4. Repeat till convergence.