1 Introduction

So far, we have dealt with biological problems in which we must: make a comparison between two pieces of data, e.g. sequence alignment; search for a sequence in a database, e.g. BLAST; classify unknown objects given known objects, e.g. supervised learning; and classify unknown objects without known objects, e.g. unsupervised learning. Today we turn to the problem of how to find common motifs - that is, commonly occurring sequences and their degenerate cases - in co-regulated genes without knowing what these motifs look like before we start our search.

2 Motivation

Unlike the islands containing perhaps hundreds of bases that we sought in previous lectures, the motifs that we seek in motif discovery are only 6-8 bases long. Though small by comparison, these motifs serve as the binding sites for RNA and proteins that will initiate transcription. If we revert to our somewhat unwelcome dogma of DNA→RNA→Protein→Trait, then we can make the connection between these binding sites and the expression of genes.

Two questions of immediate importance now surface:
1) Can there really be \(4^8 = 65,536\) regulatory proteins, 1 for each possible motif sequence?
2) But they’re really so short - can we even find them and separate them from background noise?

To answer question (1), consider that we have only about 20,000 genes. Thus, we’re not likely to see \(4^8\) different motifs. Question (2), as we will see, will require us to use a bit of probability to make a hypothesis about how likely it is that some motif that we find did not arise by chance in an intergenic region.

What might complicate matters is that these motifs need not be adjacent or even too near the genes they serve. Because of the folded nature of DNA, it is entirely possible for a motif to be 1 or 2 genes away from its gene and to still be close enough in three dimensional space for a protein to initiate transcription. When our algorithms operate on a string of DNA bases, they are of course not aware of this spatial relationship, so we must beware of that consideration, as well.
3 Possible Methods of Finding Motifs

We could:

1) perform a local alignment across many co-regulated genes and explore the alignments with a very high score.

2) train a HMM on the gene, and then use the generative model to test against unrelated genes.

3) reduce the search space by applying expert knowledge, e.g. that motifs tend to be palindromes or GC-rich.

4) find conserved genes.

Once we have a list of what we think might be motifs, we might make the assumption that the least common one across the whole genome has the highest chance of being a real motif because it does not arise by chance. In other words, there must be some selective pressure to have that particular sequence only in specific places. Professor Kellis dealt with this in his Ph.D. thesis from 2003, which is available at:


4 Combinatorial vs. Probabilistic Methods

With the exception of exhaustive search, all of the combinatorial methods - such as narrowing our search space by looking for palindromes - are heuristics, so they are not guaranteed to find the optimal solution. Probabilistic methods, of course, are based in chance, so they are not guaranteed to find the optimal solution, either, and it is unknown whether or not you can solve the motif discovery problem in polynomial time.

5 Combinatorial Methods

5.1 Exhaustive Search

To find the provably "best" motif, we could simply test every possible motif that we see. However, this is exponential in time, so it is impractical even for short sequences.

5.2 Greedy Motif Clustering

For a better running time, we can perform Greedy Motif Clustering. For every k-mer \( W \) in the sequence \( S \), we can perform an alignment against every other k-mer \( W' \) in \( S \) (without gaps) and keep the \( n \) k-mer pairs that have the highest scoring alignments. We could iterate through this process again, but instead of aligning every k-mer \( W \) to every other k-mer \( W' \), we could align every \( W \) with the alignments from the previous iteration.
5.3 Motif Refinement

If we begin by assuming that we know approximately what a motif looks like - say, we know three of the bases and their positions - then we could search for the possible motifs that meet the assumed criterion. The Motif Refinement method does exactly that, but for each $W$ in $S$. What we will find using this method is a motif that has a small Hamming Distance to other $W'$.

6 Probabilistic Methods

6.1 Expectation Maximization

Suppose we have several aligned sequences, and we assume that there is a motif common to each of them. We can easily calculate the profile matrix by examining the distribution of bases at each position (see lecture slides, page 5, number 4, “Starting postitions $<=$ Motif matrix”). We do not yet know the motif’s true starting position, but we can start at the most likely position as given by the expected value of the profile matrix. As might be expected, since we are dealing with expectation, we average over all starting positions and weight each position by its probability of being the true starting position. We can then estimate what the motif is, and update our expected starting position. If we iterate, we will find the most likely starting position as given by the profile matrix and our assumption - namely, that there is a motif present.

6.2 Gibbs Sampling

Unlike Expectation Maximization, Gibbs Sampling will find the probabilities $p_i$ that a motif begins at some randomly chosen positions $a_i$ and will remember the highest one. We pick a sequence $X_i$ and estimate $p$ according to the positions $a$. This is known as the update step. Next, we sample a new motif at a new position $a_j$ for sequence $X_j$. If we iterate, our $p$ values will eventually converge. Problem set 3, problem number 3 examines a simplified example of Gibbs Sampling to illustrate the convergence.

The mathematics behind Gibbs Sampling are not so advanced, and certainly we have seen these principles of probability estimation before. One prominent disadvantage that it faces is that the points to sample from are chosen randomly, and therefore we could simply choose places where there are no motifs, yet the algorithm would dutifully report the probabilities it found. More samples could be taken, but in general, this is not a systematic way to examine data, so it does have a flaw even in the intuition about its capabilities.

For a demonstration of Gibbs Sampling in action online, see:

http://bayesweb.wadsworth.org/cgi-bin/gibbs.11.pl?data_type=DNA
6.3 Improving Probabilistic Motif Discovery

As stated previously, the combinatorial methods of motif discovery are heuristics, but heuristics are not limited to the combinatorial methods. We commonly represent, e.g., our transition probabilities as first-order probabilities. (In this instance, a first-order probability is one in which the probabilities for reaching a particular next base are dependent only upon the base that we are currently at. A second-order base would depend upon the current one and the one preceding it.) In practice, this might not be an accurate description of the world, or at least not as accurate as a higher-order probability model might be. Up to third-order probability models have been applied to Expectation Maximization and Gibbs Sampling.

7 The Choice of Algorithm for Real Data

Several studies have been made to compare the results attained by the various motif discovery algorithms to each other and to the real motifs found by other means. A paper by Tompa, et. al., shows that any particular algorithm finds only some of the known motifs, and thus it is advisable to use several methods for analysis. The paper is available here:

http://www.nature.com/nbt/journal/v23/n1/full/nbt1053.html