CSE 527 Notes 15: AlignACE and Gibbs with Gaps

November 17, 2003

Transcribed by Michael F. Ringenburg

1 AlignACE

The Gibbs sampling method we discussed last time (Lawrence *et al.*) was designed for protein motif finding. Roth *et al.* designed a system called AlignACE for DNA motif finding. AlignACE is similar, but has a few differences:

- Paid attention to the background model in yeast (the test organism). The bases are 62% A-T, so a uniform background model would not work well. This paper and the previous paper assumed an independent background model, which is not true. There is correlation between the neighboring bases, because of chemistry, DNA repair mechanisms, etc...
- Both strands of the DNA were used (not an issue with proteins)
- Overlapping motifs were prohibited, because transcription factors can not bind to overlapping sites at the same time. This prevents some common repeats from being labeled as motifs.
- Multiple motifs were found by finding the strongest motif, and then masking it and searching for the next strongest motif.
- Used "MAP" scoring: Maximum A Postiori Probability.

2 Gibbs with Gaps

Rocke and Tompa adapted the Gibbs sampling techniques to find gapped motifs in non-coding regions of DNA (in "An Algorithm for Finding Novel Gapped Motifs in DNA Sequences", *RECOMB98: Proceedings of the Second Annual International Conference on Computational Molecular Biology*, New York, NY, March 1998, 228-233). Gapped motifs are important because:

- Similar transcription factors can bind to similar sites.
- The same transcription factor can bind to two similar sites.
- Some transcription factors (dimeric) bind to the beginning and end, and don't care as much about the middle
- The transcription factors or DNA may loop, bulge, or fold.

Why are gaps hard? Two problems:

1. How do we handle scoring? What is the penalty/reward for a gap?
2. How do we align the sequences in the presence of gaps? Using the Smith-Waterman algorithm, we can find the optimal alignment of a pair of strings (sequences) in $O(n^2)$ time using dynamic programming. But we need the alignment of k strings, which takes $O(n^k)$ time using Smith-Waterman. And we need to do this process in the inner loop of the Gibbs sampling algorithm. This gets too expensive to be reasonable.

For scoring, Rocke and Tompa used relative entropy and the familiar weight matrix model, with an extra row for gaps. Gaps were scored like background noise, minus a small, user-specified penalty (to discourage gaps).

To solve the alignment problem, Rocke and Tompa took advantage of the fact that Gibbs sampling replaces one string/sequence every iteration. Rather than computing a k-wise alignment, they computed the pairwise alignment between the new string and the previously computed alignment of the $(k-1)$ other strings.

They also made a couple modifications to the Gibbs sampling algorithm:

- Because there may be multiple copies of a motif, they throw out between 0 and 2 strings every iteration (based on the score), rather than throwing out 1 every iteration.
- They also picked the replacement greedily, rather than sampling based on the probabilities. This makes convergence to a local maximum more likely. To avoid local maximums, they used random restarts.

Their tests were run using *Haemophilus influenzae*, an organism with 1.8 megabases. After deleting coding regions, there are 350 kilobases. They then concatenated these regions (on both strands) separated by markers, ending up with about 700 kilobases.

The graphs in the slides compare the scores found on the test data with the scores found using random data. We want good separation between these two curves (indicating that real data tends to give higher scores than random data).

They made a couple modifications based on the observed results:

- **Rewindowing.** In some found motifs, the neighboring bases also matched well. To account for this, after convergence they chose a subset of the strings and adjusted the boundaries (via a greedy heuristic—exact optimization seems hard), then reran the algorithm with this as the starting point. The results showed a stronger separation between the curves.
- They found a strong motif that corresponded to known RNA coding genes (ribosomal and transfer RNAs). They removed these regions and reran the algorithm

These modifications emphasize the point that once you implement and run the algorithm, you are not done—you may need to adjust the algorithm or implementation based on the observed results.