Lecture 10

Finding Instances of Unknown Sites

February 8, 2000
Notes: Dylan Chivian

In order to find instances of unknown sites, we would like to be able to solve the relative entropy site selection problem (Section 9.3) exactly and efficiently. Unfortunately, Theorem 9.1 shows that the relative entropy site selection problem is NP-complete, so we are unlikely to find an algorithm that will compute an optimal solution efficiently. However, if we relax the optimality constraint, it may be possible to develop algorithms that compute "good" solutions efficiently. Because of the problem abstraction required to model the biological problem mathematically, the mathematically optimal solution need not necessarily be the most biologically significant. Lower scoring solutions are potentially the "correct" answer in their biological context. Therefore, giving up on the mathematical optimality of solutions to the relative entropy site selection problem seems the right compromise.

As an example of a typical application of finding instances of unknown sites, consider the genes involved in digestion in yeast. It is likely that many of these genes have some transcription factors in common, and therefore similarities in their promoter regions. Applying the site selection problem to the 1Kb DNA sequences upstream of known digestion genes may well yield some of these transcription factor binding sites. As another example, we could use the site selection problem to find common motifs in a protein family.

As defined, the relative entropy site selection problem limits its solution to contain exactly one site per input sequence, which may not be realistic in all applications. In some applications, there may be zero or many such sites in some of the input sequences. The algorithms discussed below are described in terms of the single site assumption, but can be modified to handle the general case as well.

But in the context of this general case, this is a good point at which to consider the effects on relative entropy of increasing either the number of sites or the length of each site. Increasing the number of sites will not increase the relative entropy, which is a function only of the fraction $P(s)$ of sites containing each residue s, and not the absolute number of such sites. For instance, a perfectly conserved position has $P(s) = 1$, regardless of whether it is present in all 10 sites or all 100 sites. This aspect of relative entropy is both a strength and a weakness. The strength is that it measures the degree of conservation, but the weakness is that we would like the measure to increase with more instances of a conserved residue.

However, increasing the length n of each site *does* increase the relative entropy, as it is additive and always nonnegative (Theorem 8.8). If comparing relative entropies of different length sites is important, one may normalize by dividing by the length n of the site or, alternatively, subtracting the expected relative entropy from each position.

10.1. Greedy Algorithm

Hertz and Stormo [2] described an efficient algorithm for the relative entropy site selection problem that uses a "greedy" approach. Greedy algorithms pick the locally best choice at each step, without concern for the impact on future choices. In most applications, the greedy method will result in solutions that are far from optimal, for some input instances. However, it does work efficiently, and may produce good solutions on many of its input instances.

Hertz and Stormo's algorithm for the relative entropy site selection problem proceeds as follows. The user specifies the length n of sites. The user also specifies a maximum number d of profiles to retain at each step. Profiles with lower relative entropy scores than the top d will be discarded; this is precisely the greedy aspect of the algorithm.

INPUT: sequences $s_1, s_2, \ldots, s_k$, and n, d, and the background distribution.

ALGORITHM:

1. Create a singleton set (i.e., only one member) for each possible length n substring of each of the k input sequences.
2. For each set S retained so far, add each possible length n substring from an input sequence s_i not yet represented in S. Compute the profile and relative entropy with respect to the background for each new set. Retain the d sets with the highest relative entropy.
3. Repeat step 2 until each set has k members.

A small example from Hertz and Stormo [2] is shown in Figure 10.1. From this example it is clear that pruning the number of sets to d is crucial, in order to avoid the exponentially many possible sets. The greedy nature of this pruning biases the selection from the remaining input sequences. High scoring profiles chosen from the first few sequences may not be well represented in the remaining sequences, whereas medium scoring profiles may be well represented in most of the k sequences, and thus would have yielded superior scores.

Note that one may modify the algorithm to circumvent the assumption of a single site per sequence, by permitting multiple substrings to be chosen from the same sequence. In this case, a different stopping condition is needed.

Hertz and Stormo applied their technique to find CRP binding sites (see Section 7.1) with some success. With 18 genes containing 24 known CRP binding sites, their best solution contained 19 correct sites, plus 3 more that overlap correct sites.

10.2. Gibbs Sampler

Lawrence *et al.* [4] developed a different approach to the relative entropy site selection problem based on "Gibbs sampling". The idea behind this technique is to *start* with a complete set of k substrings (candidate sites), from which we iteratively remove one at random, and then add a new one at random with probability proportional to its score, hopefully resulting in an improved score. In the following description, we again make the assumption that we choose one site per input sequence, but this method also can be extended to permit any number of sites per sequence.

INPUT: sequences $s_1, s_2, \ldots, s_k$, n, and the background distribution.

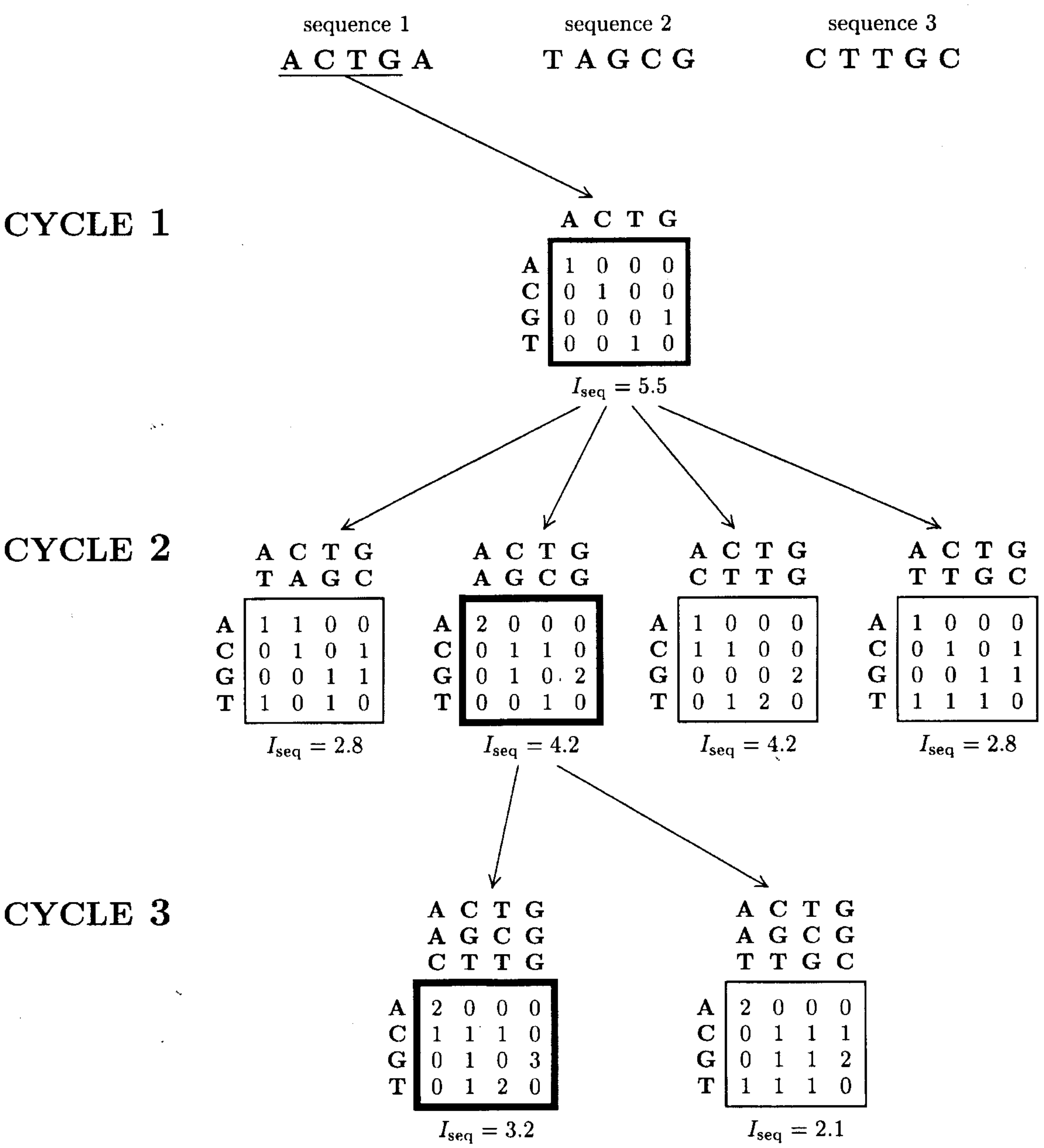

Figure 10.1: Example of Hertz and Stormo's greedy algorithm. I_{seq} denotes the relative entropy.

ALGORITHM: Initialize set T to contain substrings $t_1, t_2, \ldots, t_k$, where t_i is a substring of s_i chosen randomly and uniformly. Now perform a series of iterations, each of which consists of the following steps:

1. Choose i randomly and uniformly from $\{1, 2, \ldots, k\}$ and remove t_i from T.
2. For every j in $\{1, 2, \ldots, |s_i| - n + 1\}$:
   (a) Let t_{ij} be the length n substring of s_i that starts at position j.

   (b) Compute D_j, the relative entropy of $T \cup \{t_{ij}\}$ with respect to the background.

   (c) Let $P_j = D_j / \sum_h D_h$.

3. Randomly choose t_i to be t_{ij} with probability P_j, and add t_i to T.

We iterate until a stopping condition is met, either a fixed number of iterations or relative stability of the scores, and return the best solution set T seen in all iterations. The hope with this approach is that the random choices help to avoid some of the local optima of greedy algorithms.

Note that the Gibbs sampler may discard a substring that yields a higher scoring profile than the one that replaces it, or may restore the substring that was discarded itself. Neither of these occurrences is particularly significant, since the sampling will tend toward higher scoring profiles due to the probabilistic weighting of the substitutions by relative entropy. The Gibbs sampler does retain some degree of greediness (which is desirable), so that there may be cases where a strong signal in only a few sequences incorrectly outweighs a weaker signal in all of the sequences.

Lawrence *et al.* applied their technique to find motifs in protein families. In particular, they successfully discovered a helix-turn-helix motif, as well as motifs in lipocalins and prenyltransferases.

10.3. Other Methods

Possible extensions to the Gibbs sampler technique of Section 10.2 include the following:

1. Weight which t_i to discard in step 1 (analogously to weighting which to add in step 3).

2. Use simulated annealing (see, for example, Johnson *et al.* [3]) where, as time progresses, the probability decreases that you make a substitution that worsens the relative entropy score, yielding a more stable set T.

Another technique that has been used to solve the site selection problem is "expectation maximization" (for example, in the MEME system [1]).

References

[1] T. L. Bailey and C. Elkan. Unsupervised learning of multiple motifs in biopolymers using expectation maximization. *Machine Learning*, 21(1-2):51–80, Oct. 1995.

[2] G. Z. Hertz and G. D. Stormo. Identifying DNA and protein patterns with statistically significant alignments of multiple sequences. *Bioinformatics*, 15(7/8):563–577, July/August 1999.

[3] D. S. Johnson, C. R. Aragon, L. A. McGeoch, and C. Schevon. Optimization by simulated annealing: an experimental evaluation; part I, graph partitioning. *Operations Research*, 37(6):865–892, Nov.–Dec. 1989.

[4] C. E. Lawrence, S. F. Altschul, M. S. Boguski, J. S. Liu, A. F. Neuwald, and J. C. Wootton. Detecting subtle sequence signals: a Gibbs sampling strategy for multiple alignment. *Science*, 262:208–214, 8 October 1993.