Reconstruction of ancestral DNA sequences

Zhentao Li

August 24, 2005

We study the reconstruction of ancestral DNA sequences from those of contemporary species. Given a set of species with known DNA sequences at the leaves of a given phylogenetic tree and a multiple alignment of the sequences, we must find the sequences for their ancestors that would minimize the total number of insertions and deletions along the branches of the tree. We also consider the model where there is affine gap penalty. For each internal node of the tree, we want to determine at which positions of the alignment has a nucleotide and which has a gap. Determining which of the four nucleotides (A, C, G or T) is most likely can be done in polynomial time using Felsenstein's algorithm. We study the model where, if two leaves have nucleotides at any given position, the path between them in the tree must also have nucleotides at that position.

We prove that our model is equivalent to the one with the following restrictions: (i) At a given position, if two species have a nucleotide, their least common ancestor also has a nucleotide. (ii) At a given position, if both an ancestor and a descendant of a species have a nucleotide then that species also has a nucleotide.

Under the model where only the second restriction applies, we prove that the problem is NP-complete.

It has previously been shown that this problem can be encoded as a integer linear programming problem, whose optimal solution can usually be accurately approximated by solving its fractional relaxation. Nonetheless, the size of the problem solved (eventually, whole genomes) requires us to break the problem into independant subproblems.

Under the model where both restrictions apply, we give a polynomial time algorithm which allows us to break the problem into smaller ones. The optimal solutions can be merged into an optimal solution to the original problem. We also give another rule for breaking the problem into problems of sizes as small as we wish and show that each such break can increase the cost of the solution by at most three.

Using these two sets of rules, we can solve optimally or near-optimally problems with 20 sequences of 10^6 nucleotides.