Bioinformatics 2005 – Paper 9 Answer 9 (PL)

(*a*) Present the aim of phylogeny algorithms.

(*i*) Describe the main differences between Parsimony, Distance and Likelihood-based algorithms [5 marks]

The method of maximum likelihood (ML) is one of the standard tools of statistics. The likelihood is the probability of data given the model parameters (tree topology and branch lengths). In molecular phylogenetics, the ML tree is the one that renders the observed sequences the most plausible, given the chosen model of sequence evolution. It permits the inference of phylogenetic trees using complex evolutionary models including the ability to estimate model parameters and so make inferences simultaneously about the patterns and processes of evolution and provides the means for comparing competing trees and models of evolution (such as Jukes-Cantor, Kimura 2 parameters, HKY85). ML phylogenetic inference suffers from the fact that each possible tree topology should be assessed individually, and, when examining large numbers of sequences, the number of possible tree topologies is huge. Heuristic searches for the ML tree are widely used, but give no guarantee of finding the optimal tree. The likelihood framework permits estimation of parameter values and their standard errors from the observed data, with no need for any a priori knowledge. Comparisons of two competing models are also possible, using likelihood ratio tests. Bayesian approaches in phylogeny stem from ML inference. Distance methods use the same models of evolution as ML to estimate the evolutionary distance between each pair of sequences from the set under analysis, and then to fit a phylogenetic tree to those distances. The distances will usually be ML estimates for each pair (considered independently of the other sequences), but the set of all pairwise distances will not be compatible with any tree and a best-fitting phylogeny is derived using non-ML methods. Disadvantages of distance methods include the inevitable loss of evolutionary information when a sequence alignment is converted to pairwise distances, and the inability to deal with models containing parameters for which the values are not known a priori. Maximum parsimony selects the tree or trees that require the fewest evolutionary changes. If the number of changes per sequence position is relatively small, then maximum parsimony approximates ML and its estimates of tree topology will be similar to those of ML estimation. When the true tree has short internal branches and long terminal branches, a phenomenon can occur whereby the long branches appear to attract one another and can be erroneously inferred to be too closely related. In addition, parsimony lacks an explicit model of evolution.

(*ii*) Describe the input and the output of a distance-based algorithm [5 marks]

The basic input is a sequence alignment. Using a model of evolution (the same as for ML) and a star tree, a distance matrix is obtained. This is a symmetric $n \times n$ matrix M, whose element M_{ij} is the distance between objects (DNA bases, amino acids) i and j. This matrix satisfies the triangle inequality, that is $M_{ij} \leq M_{ik} + M_{kj}$ for all i,j, and k. Using the distance matrix and additional parameters (sequence lengths, transition/transversion ratio) a unique edge-weighted tree (unrooted) is obtained. Each leaf corresponds to one object and such that distances measured on the tree between leaves i and j correspond exactly to the value of M_{ij}.

(*iii*) Discuss the complexity of the Neighbor-Joining algorithm? [5 marks]

The Neighbour Joining method is a method for constructing phylogenetic trees, and computing the lengths of the branches of this tree. It is a greedy algorithm that attempts to minimize the sum of all branch-lengths on the constructed tree. Conceptually, it starts out with a star-formed tree where each leaf corresponds to a species, and iteratively picks two nodes adjacent to the root and joins them, by inserting a new node between the root and the two selected nodes. The original algorithm from Saitou and Nei (1987) had $O(n^5)$ complexity, whereas, choosing an alternative parameter, Studier and Keppler (1988) found a complexity of $O(n^3)$. Details: When joining nodes, the method selects the pair of nodes i, j that minimizes the branch-length sum of the resulting new tree. This pair can be found by minimizing the expression $Q_{ij} = (r-2)d_{ij} - (R_i + R_j)$, where d_{ij} is the distance between node i and j (assumed symmetric, i.e. $d_{ij} = d_{ji}$), R_k is the row sum over row k of the distance matrix: $R_k = \sum i d_{ik}$ (where i ranges over all nodes adjacent to the root node), and r is the remaining number of nodes adjacent to the root. When nodes i and j are joined, they are replaced with a new node, A, with distance to the remaining nodes given by $d_{Ak} = (d_{ik} + d_{jk} - d_{ij})/2$. The method performs a search for $minQ_{ij}$, using time $O(r^2)$, and joins i and j , using time $O(r)$ to update d. This search and join is continued until only three nodes are adjacent to the root (i.e. for n-3 joins), giving a total time complexity of $O(n^3)$, and a space complexity of $O(n^2)$.

(*b*) Describe with one example the Needleman-Wunsch algorithm [5 marks]

The Needleman-Wunsch Algorithm: Score F = (#matches) $\times$ m $-$ (#mismatches) $\times$ s(#gaps) $\times$ d. The optimal (maximal) score between two sequences is the maximal score of all alignments of these sequences. The additive form of the score allows us to perform dynamic programming to compute optimal score efficiently

1) Initialization.

F(0, 0) = 0;
F(0, j)= - j x d;
F(i, 0) = - i x d.

2) Main Iteration: filling-in partial alignments

For each i = 1..M

For each j = 1..N

$$\text{F(i, j)} = \max \{ \begin{array}{ll} \text{F(i-1,j)-d} & \text{[case 1]} \\ \text{F(i, j-1)-d} & \text{[case 2]} \\ \text{F(i-1, j-1)} + \text{s}(x_i, y_j) & \text{[case 3]} \end{array}$$

Ptr(i,j)	UP	if [case 1]
Ptr(i,j)	LEFT	if [case 2]
Ptr(i,j)	DIAG	if [case 3]

3) Termination.

F(M, N) is the optimal score, and Ptr(M, N) can trace back optimal alignment

Example:

seq1: GCGTAGTAC seq2: TAGTGAAAC

the transition scoring matrix

	A	C	G	T
A	4	-2	1	-2
C	-2	4	-2	1
G	1	-2	4	-2
T	-2	1	-2	4

and the linear gap penalty -3.

		C	G	C	A	T	C	A	T	G
	0	-3	-6	-9	-12	-15	-18	-21	-24	-27
A	-3	-2	-2	-5	–5	-8	-11	-14	-17	-20
T	-6	-2	-4	-1	-4	-1	-4	-7	-10	-13
C	-9	-2	-4	0	-3	-3	3	0	-3	-6
A	-12	-5	-3	-1	4	1	0	-7	4	1
C	-15	-8	-4	3	1	5	5	4	8	5
T	-18	-11	-7	0	1	5	6	3	8	6
C	-21	-14	-10	-3	-2	2	9	6	5	6
G	-24	-17	-10	-6	-2	-1	6	10	7	9
G	-27	-20	-13	-9	-5	-4	3	7	8	11

(a1) The global alignment is

AGCA-TCATG

CTCACTCG-G

(a2) The score is 11.