A Tree Reconstruction Method That Is Economical in the Number of Pairwise Comparisons Used[1]

Jotun Hein
NIEHS

A fast method for reconstructing phylogenies from distance data is presented. The method is economical in the number of pairwise comparisons needed. It can be combined with a new phylogenetic alignment procedure to yield an algorithm that gives a complete history of a set of homologous sequences. The method is applicable to very large distance matrices. An auxiliary program was developed that simplifies large phylogenies without ignoring biologically essential features. A set of 213 globins from vertebrates, plants, and Vitreoscilla (a prokaryote) were analyzed using this method.

Introduction

The main aim of the tree construction method described in the present paper is to direct the alignment algorithm presented in the accompanying article (Hein 1989). The method is developed to be applied to sequence data, but the method can be applied to any objects that have both an evolutionary history that can be described by a tree and a measure of dissimilarity. Previous methods for construction of phylogenetic trees from distance data start by calculating the $n \times (n-1)/2$ entries in the distance matrix, where n is the number of sequences. Since each distance is obtained by a computationally expensive alignment in the present method, it is desirable to be more economical at this step. In the method presented here, as the number of sequences grows, only a decreasing fraction of the distance matrix is necessary for the calculation of the tree.

The overall order of the calculations is as follows:

1. The most informative distances for the tree construction process are calculated.
2. A distance tree is constructed for the sequences by adding sequences one by one to a growing tree.
3. Rearrangements are performed on the obtained tree to improve the overall fit of the tree to the distance data.
4. The resulting tree is used to guide the tree alignment algorithm such that a parsimony tree, a tree with ancestral sequence assigned to internal nodes and substitutions and insertions-deletions (indels) to the edges, is obtained that has the same topology as the distance tree. Thus, a complete history is obtained and the ancestral sequences are determined. The new tree might have branch lengths very different from those of the previous distance tree. It will also have an arbitrary root.
5. The criterion for the quality of the history of the sequences is parsimony, i.e.,

[1] Key words: tree additivity, phylogeny, phylogeny simplification, globins.

Address for correspondence and reprints: Dr. Jotun Hein, HIEHS, Research Triangle Park, North Carolina 27709.

Mol. Biol. Evol. 6(6):669–684. 1989.

0737-4038/89/0606-0007$02.00

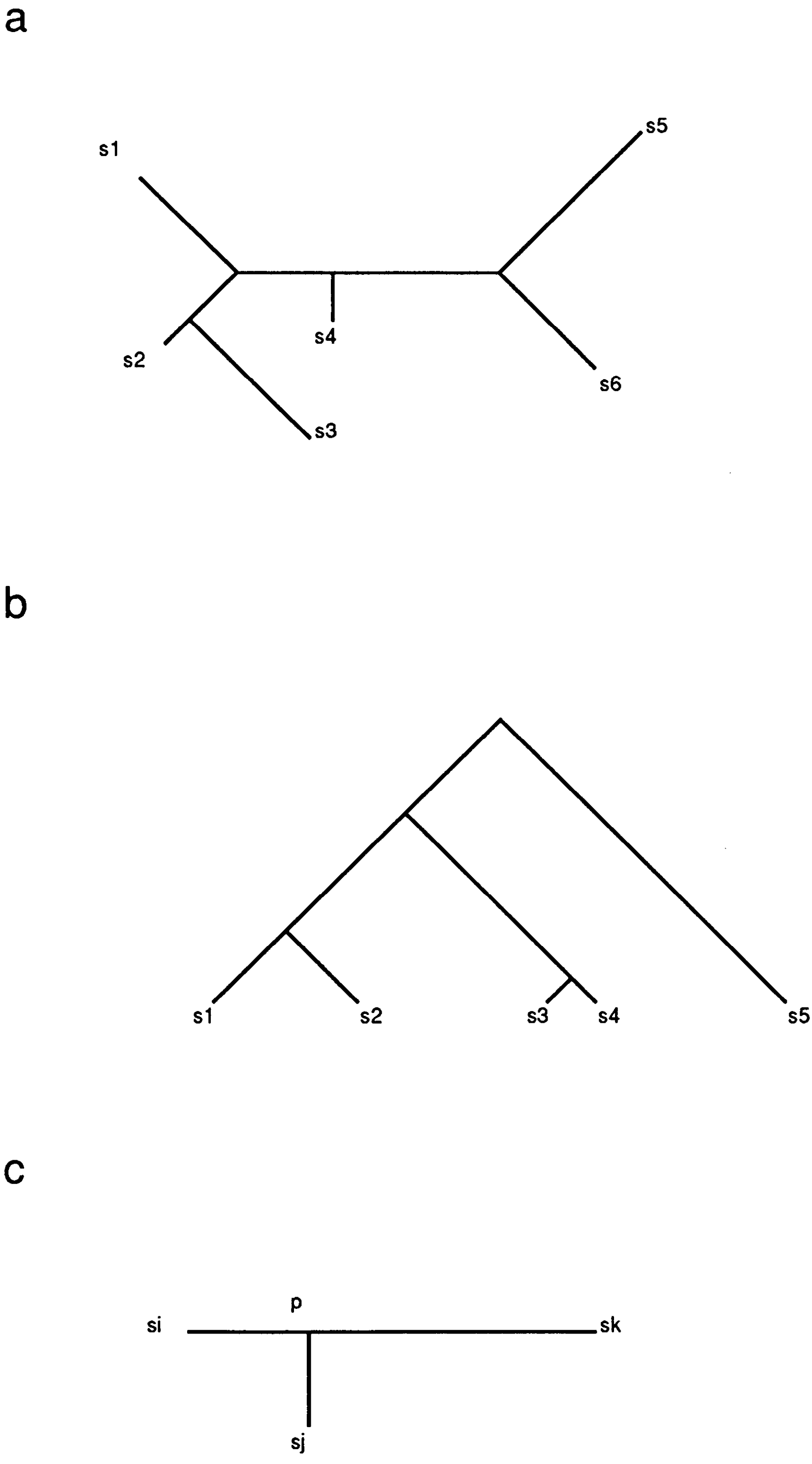


FIG. 1.—Tree additivity–ultrametricity. a, Six objects connected by a tree. If the metric equals the path length connecting two objects, then this will be tree additive. b, Rooted tree with a perfect clock. This will lead to an ultrametric. c, Tree leading to a tree-additive metric. This shows that the tree is easily reconstructed from the pairwise distances between three objects.

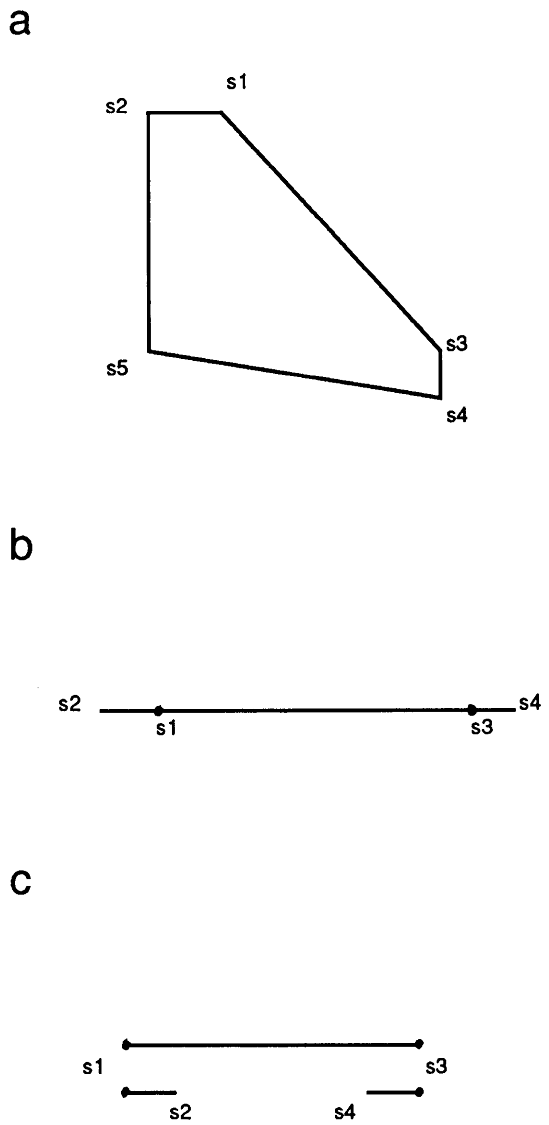


FIG. 2.—Metric inequalities. a, Five points in the plane. The edges are the pairwise distances known. d(s2,s4) is not known but must be less than the upper bound (panel b) and more than the lower bound (panel c). These inequalities can be visualized geometrically. Let the edges be inflexible rods, and let the nodes be perfect hinges. To get the best upper inequality between si and sj, take the graph in the nodes si and sj and pull them apart; the resulting distance between si and sj will be the upper bound. To find the best lower bound, pull si and sj together.

the fewer events the better. Again, rearrangements are performed on the tree to improve it, in an effort to make it more parsimonious.

Terminology and Background

The distance between two sequences is traditionally measured by the smallest weight of a series of weighted operations leading from the first sequence to the second sequence. This distance will almost always be smaller than the weight of the true

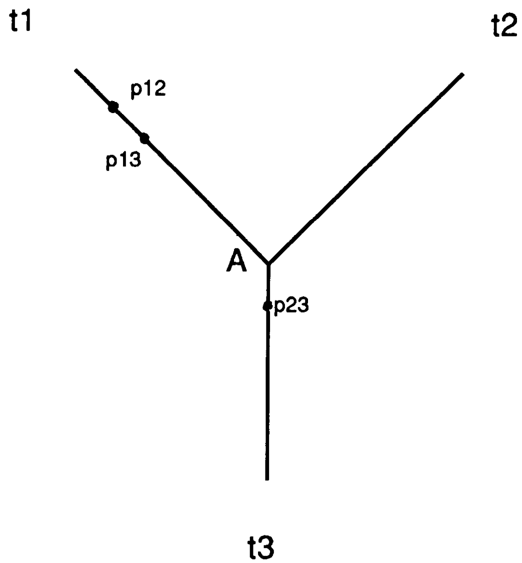


FIG. 3.—Investigation around the current internal node, A, to discover where the new sequence should be positioned. t1, t2, and t3 are the three subtrees radiating from A. The investigation will determine whether the new sequences should be attached to one of the three edges incident to A and, if not, in which direction (which subtree) the investigation should continue.

amount of evolution in the history leading to the present sequences from the most recent common ancestor. When the sequences are closely related, the discrepancy is probably small. As sequences become more distantly related, the minimal distance between them may be seriously underestimated. To compensate for this, a series of correction formulas has been developed. The original Jukes and Cantor correction (1969) formula for DNA, along with the analogous PAM (Dayhoff et al. 1972) for protein, were used in the present study. Most correction methods underestimate the real distance and are associated with a very large uncertainty for remotely related sequences.

A distance function on a set of objects connected by a tree is said to be *tree additive* (Buneman 1971) if a tree exists with weighted edges (lengths) such that the distance between two objects always equals the length of the simple path in the tree connecting the objects (fig. 1a). The genetic distance on homologous sequences cannot be expected to be perfectively tree additive, but if the sequences are closely related, the distances should be at least approximately tree additive. For sake of illustration, assume, until further notice, that the distances are perfectively tree additive. Given three sequences s_i, s_j, and s_k, the exact dimensions of the tree can be determined in terms of the three possible distances between them. Let p be the internal node in the tree. The tree has been fully described if the distances to p from all three sequences are determined. They are easily found (see fig. 1c):

$$\begin{aligned} d(s_i,p) &= [d(s_i,s_j)+d(s_i,s_k)-d(s_j,s_k)]/2\,; \\ d(s_j,p) &= [d(s_j,s_i)+d(s_j,s_k)-d(s_i,s_k)]/2\,; \qquad (1) \\ d(s_k,p) &= [d(s_k,s_j)+d(s_k,s_i)-d(s_j,s_i)]/2\,. \end{aligned}$$

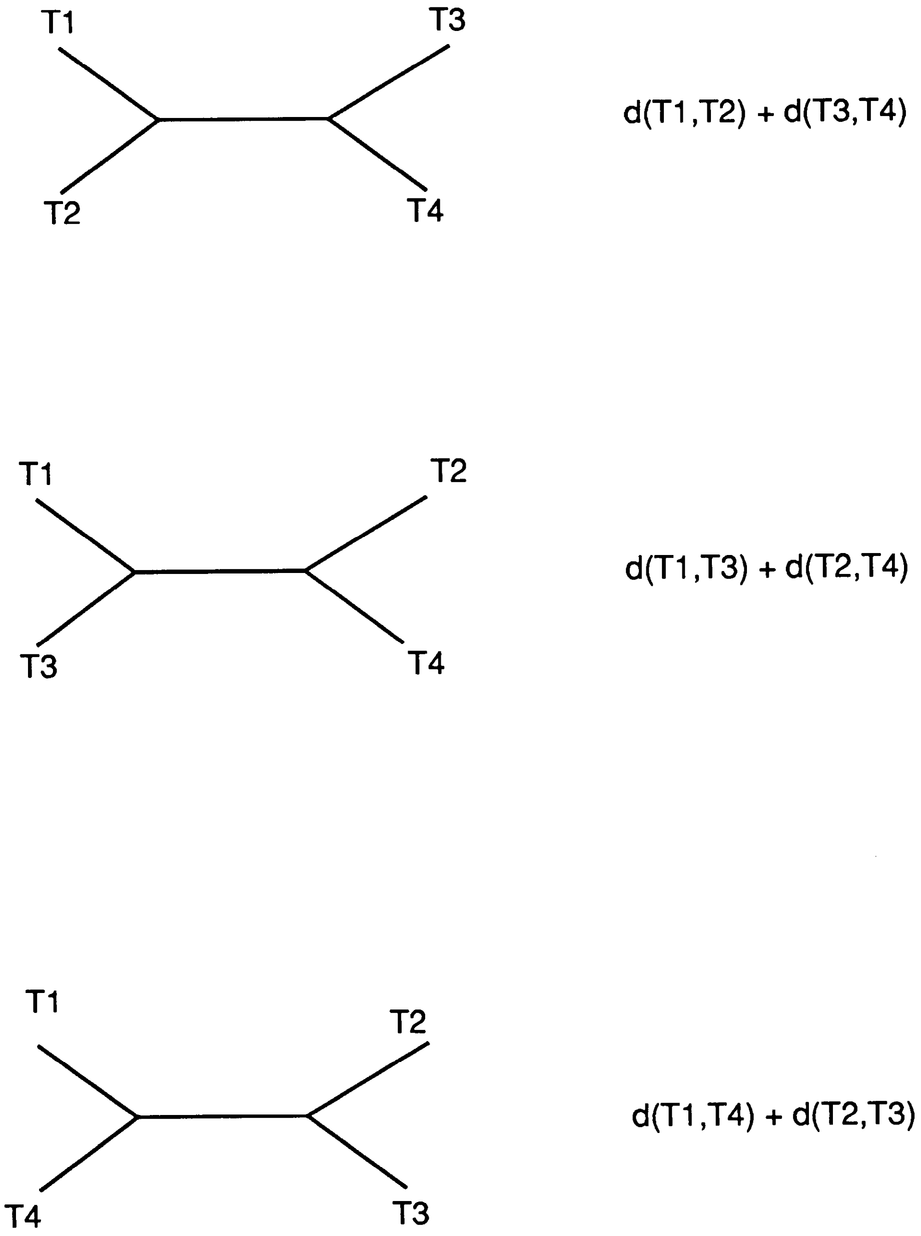
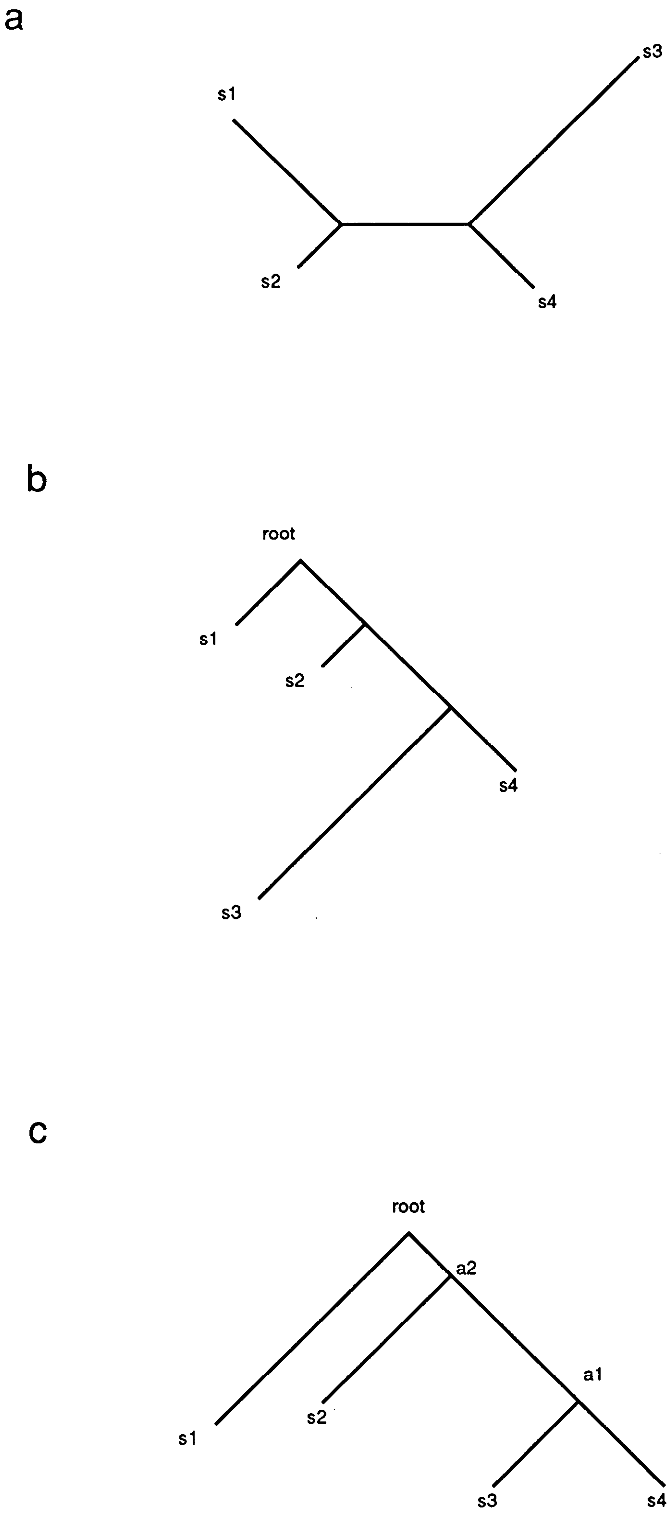


FIG. 4.—Three possible configurations of the four subtrees around an internal edge and the associated quantities that measure the "weight" of the configuration.

These equations can be used to make an algorithm that reconstructs the whole tree by successively adding a sequence to a growing tree and each time determining its position by suitable comparisons to sequences in the tree and by using equation (1) (Waterman et al. 1977). All trees will be assumed to describe duplications of sequences, which implies that internal nodes will have exactly three incident edges.

To be more specific: Assume that a tree of size $k - 1$, T_{k-1}, has been constructed and that now sequence s_k is to be added. Pick out two reference sequences, s_1 and s_2,



a
s1
s2
s3
s4
b
root
s1
s2
s4
s3
c
root
a2
a1
s1
s2
s3
s4

that are tips in T_{k-1}. If the distances to these to s_k are calculated, equation (1) can be used to determine exactly where on the path between s_1 and s_2 the branch leading to s_k should be attached and also how long it must be. If this point does not coincide with the point of attachment of a subtree to the path, then the position of s_k in the tree has been determined. If it coincides with a subtree, then s_k must be located in that subtree somewhere, and an additional reference sequence s_3 is chosen at some tip in that subtree. It is the conditions that (a) internal nodes have degree three and (b) edge lengths must be strictly positive that prohibits s_k from being positioned at the same node as the subtree. One reference point is already known, since the distance to the root of the subtree is known. Equation (1) is then used to determine where, on the path between the root and s_3, s_k should be attached. If the point coincides with a subtree, the comparisons continue; otherwise s_k has been correctly positioned.

The above principle can easily be used to make an $o[n \log(n)]$ worst-case algorithm to reconstruct trees from distance data that are perfectively tree additive (Hein, in press). When two reference sequences are chosen, it is possible to determine in which, if any, of the subtrees radiating from the path between them the new s_k should be positioned. This limits the possible area of attachment considerably. One simple rule that would do this is the following: Find the node such that the smallest number of tips of the three subtrees radiating from it is as large as possible. The size of the largest of the subtrees incident to this point cannot be larger than half the size of the total tree. If reference sequences are chosen so that this point is on the path between them, the possible area of attachment for s_k would have been at least halved. This would be true for each cycle, and the attachment point of s_k would be determined in fewer than $o[\log(k)]$ steps, leading to an overall performance of $o[n \log(n)]$.

If the sequences evolve at perfectly constant rates, the metric obtained from the corresponding tree will be an *ultrametric* (fig. 1b). The tree relating three sequences can then be found by taking the closest pair making them siblings, and the duplication that gave rise to them will be half their distance from both of them. Ultrametricity is a more restrictive concept than tree additivity. Both unequal rates of evolution and convergent/parallel events can make the distance function on sequences nonultrametric. Nonetheless, there are situations where it is more sensible to assume ultrametricity than tree additivity in the algorithm. If the three sequences were from *Escherichia coli,* chimpanzee, and man, for example, it should be possible to determine the lengths of the branches leading to man and chimpanzee by using *E. coli* as reference sequence. In reality, differences in the distances from *E. coli* to man and chimpanzee will more likely be due to convergence than to differences in evolutionary rates at the branches leading to man and chimpanzee. The most sensible thing to do is to assume equality of rates and to make the branches leading to man and to chimpanzee equally long—half the distance between man and chimpanzee.

It is also useful to assume approximate ultrametricity when trying to make a qualified guess about the location of the most ancestral point (the root) in a tree. When the root has been determined, all events will acquire a direction. For instance,

FIG. 5.—Distance into parsimony. a, Unrooted tree. b, The same tree with an arbitrary root on it. c, Tree resulting from using panel b to make a parsimony tree, which changed the length of the branches but not the relationship between the sequences. In addition, all internal nodes shown in panel c are associated with proposed ancestral sequences.

it becomes possible to investigate whether certain amino acids are more variable than others.

Method

In this section the details of the tree construction method, except the alignment procedure, are presented.

1. Calculation of Distance Matrix Entries

The set of sequences, S, and the distance function, $d(,)$ on S constitutes a metric space:

$$\text{a) } d(s_1,s_2) = 0 \Leftrightarrow s_1 = s_2 \,;$$

$$\text{b) } d(s_1,s_2) = d(s_2,s_1) \,;$$

$$\text{c) } d(s_1,s) + d(s,s_2) \geq d(s_1,s_2) \,.$$

By using inequality (c) and already known distances, it is possible to get information about a distance without actually calculating it. To envision this, consider figure 2. The nodes in the graph are sequences, and the edges represent calculated lengths. If two nodes are not connected by an edge, it is because the distance between them has not been calculated.

Then, because of the metric property of the sequences, the distance $d(s_2,s_4)$ is bounded above and below by $d(s_1,s_3) - d(s_1,s_2) - d(s_3,s_4) \leq d(s_2,s_4) \leq d(s_1,s_3) + d(s_1,s_2) + d(s_3,s_4)$, and, because $d(s_1,s_2)$ and $d(s_3,s_4)$ are very small, $d(s_2,s_4)$ is, for all practical purposes, known, without having been calculated.

Generally, in a metric space with n objects and only some distances known, a set of similar inequalities can be derived. This can be represented by an undirected graph with n nodes. Two nodes are connected if their distances have been calculated. The length of the connecting edge is their distance.

The least upper bound on a distance $d(s_i,s_j)$ is min{length of path connecting i and j}, where the minimum is taken over all paths connecting i and j.

The greatest lower bound for $d(s_i,s_j)$ will be max {longest segment on path minus length of the rest of the path}, where the maximum again is taken over the paths connecting i and j. If $d(s_i,s_j)$ has been calculated, the upper and lower bounds will coincide.

If the difference between the upper and lower bounds is small, $d(s_i,s_j)$ will not be calculated. Small in this case denotes a user-specified parameter. In the present version of the program the lower inequalities are not always the best possible. They are calculated using the same path that gave the best upper inequality. This is reasonable in most situations, since such paths typically have one very long segment.

2. Construction of Initial Distance Tree

The initial tree is constructed by a sequential algorithm that adds sequences, one by one, until a distance tree has been constructed for all sequences. Assume that a tree with $k - 1$ tips, T_{k-1}, has been constructed and that now sequence s_k is to be added. s_k is chosen so that $d(s_k,T_{k-1})$ is as small as possible, where $d(s_k,T_{k-1})$ is

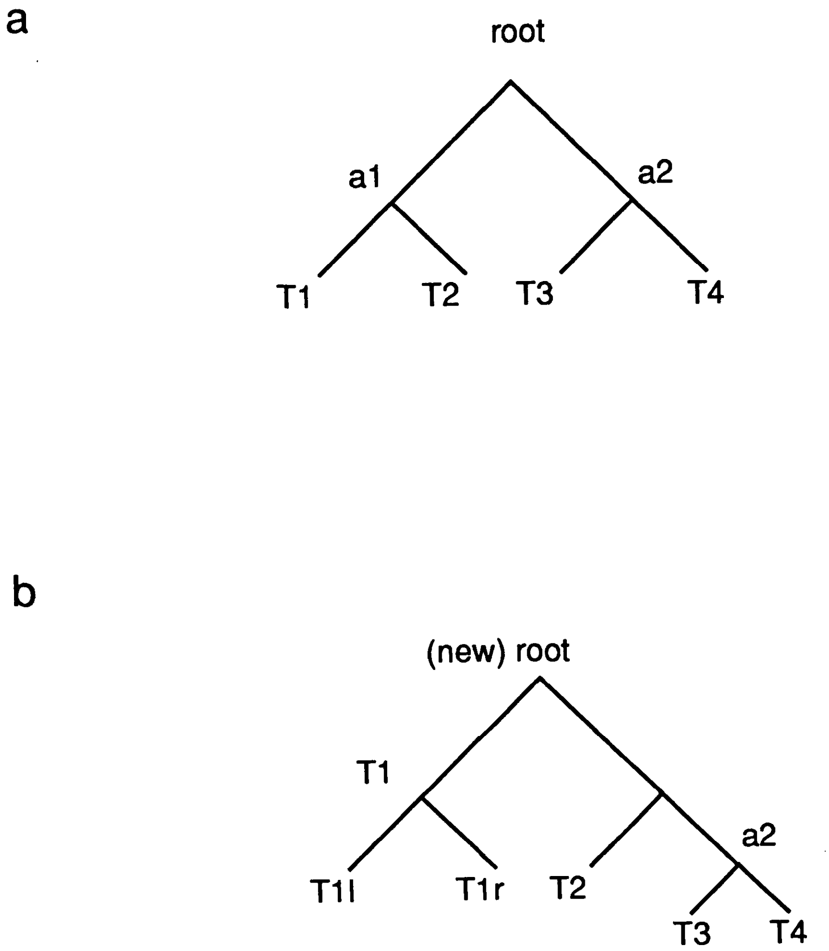
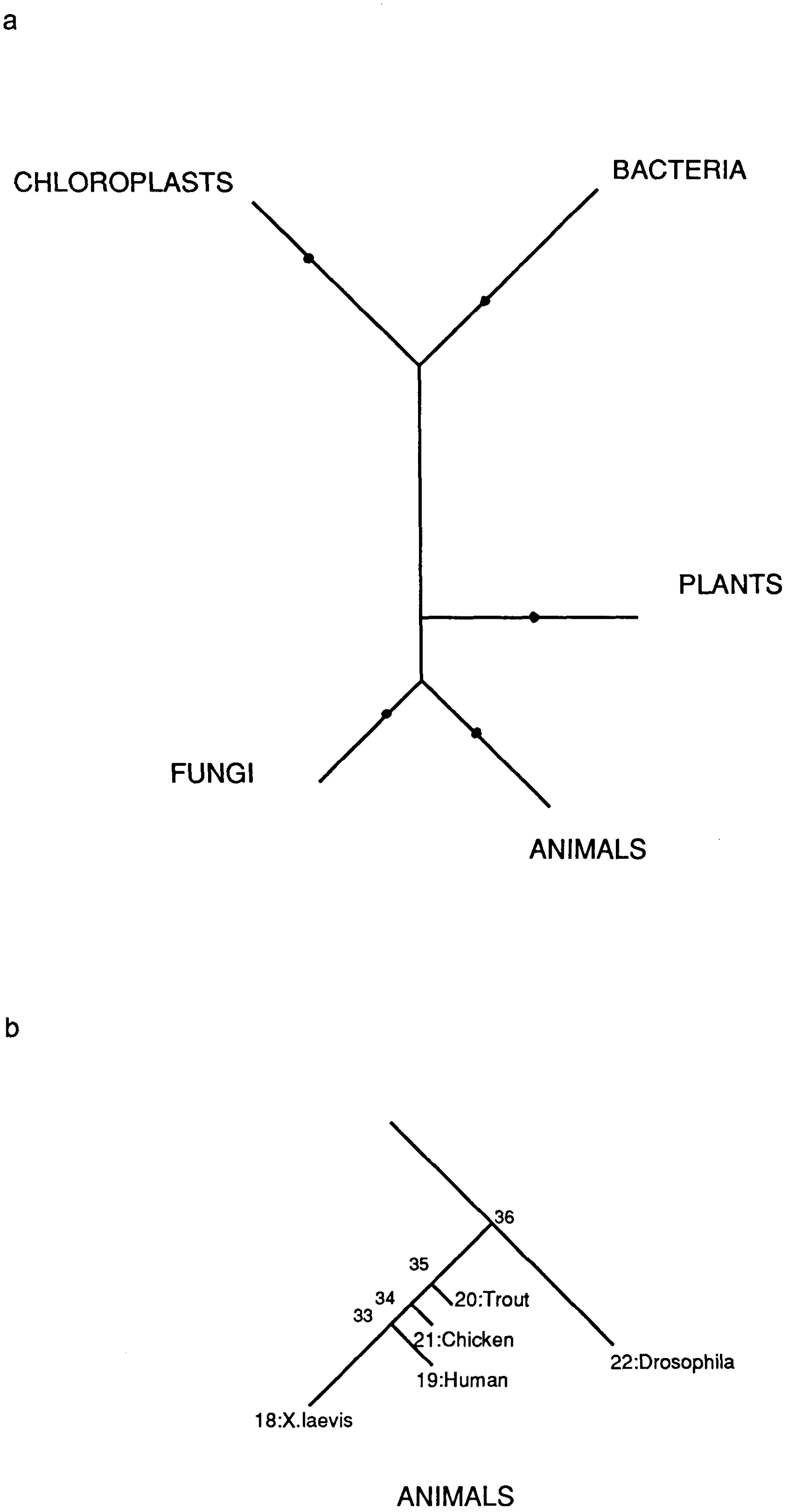


FIG. 6.—a, Root moving. Configuration of four subtrees (T1, T2, T3, and T4) around the internal edge (a1 and a2) with the arbitrary root on it. The root is moved to the edge (a1 and T1) leading to the tree shown in panel b. The old root and a1 disappear and are substituted by a new root and an ancestor to a2 and T2.

defined as the smallest $d(s_k,s_j)$, s_j is a tip in T_{k-1}, and $d(s_k,s_j)$ has already been calculated (fig. 3). Let s_1 be this closest known sequence in T_{k-1}, and let A be the node of T_{k-1} that is adjacent to s_1. Three subtrees (t_1, t_2, and t_3) radiate from A. The following quantities are now defined: $d(t_i,\mathrm{A})$ is the average of the known distances from the tips in t_i to A, and $d(t_i,s_k)$ is defined as the average distance form s_k to the tips in t_i, where this distance has already been calculated. The distance between two subtrees, $d(t_i,t_j)$, is defined in an analogous fashion. Since the distance matrix is incomplete, it is possible that $d(s_k,t_i)$ or $d(t_i,t_j)$ is undefined; in that case an arbitrary sequence is chosen in the subtree and the distance to it is calculated.

For each of the three possible pairs of (t_i,t_j), the following is done: In analogy with equation (1) the edge from s_k is joined to the path between t_i and t_j such that the distance from t_i to the joining point is

$$d(t_i,p_{ij}) = [d(t_i,s_k)+d(t_i,t_j)-d(t_j,s_k)]/2\ . \qquad (2)$$


a
CHLOROPLASTS
BACTERIA
PLANTS
FUNGI
ANIMALS
b
36
35
34
33
20:Trout
21:Chicken
19:Human
18:X.laevis
22:Drosophila
ANIMALS

This defines three vectors from A toward t_i to the attachment point (which can be negative). From these three vectors, three new vectors—v_1, v_2, and v_3—are calculated. For instance, v_1 is the average length in the direction of t_1 of p_{12} and p_{13} from A. The longest of these three vectors determines the positioning of s_k. Assume v_1 is the longest. If v_1 is longer than the edge leading to the root of t_l, then the root of t_l is chosen as a new A. If A is an internal node, the same procedure is repeated with the constraint that A has not previously been examined (this is to prevent an infinite loop). If t_1 is a tip, s_k is positioned. If v_1 is shorter than the edge it is pointing out of, then s_k is also positioned at the point between p_{12} and p_{13}. The length of the branch leading to s_k will be the average of the branch length according to (t_1,t_2) and (t_1,t_3).

The method acquires its robustness against non–tree additivity for two reasons. First, it uses many sequences as reference points in equation (2). Second, it employs three paths going through a node—instead of just one, as in the use of equation (1). This gives the algorithm alternative opportunities for positioning a new sequence correctly.

The intuitive idea of the algorithm is to let the s_k wander until it finds its correct position in the tree. It starts wandering at the sequence it resembles most, thus making the used distances as tree additive as possible. When the three paths intersecting a node are used, two of these paths will pull s_k in the direction of the correct edge in the tree.

3. Cycles of Nearest-Neighbor Interchanges

This distance tree obtained is then subjected to cycles of nearest-neighbor interchanges to improve the fit of the phylogeny to the known pairwise distances. Nearest-neighbor interchange was first used by Robinson (1971). Four subtrees radiate from each internal edge. There are three possible ways to relate these four subtrees. (fig. 4).

It can be shown that if the data were perfectively tree additive, than two of these quantities would be identical and the other quantity would be as small or smaller. The configuration associated with the smallest quantity would be the true configuration. This method was first used by Fitch (1981). All internal edges are visited a user-specified number of times, continuously improving the topology of the tree.

If one of the four subtrees represents a very distant outgroup, the criterion for the best configuration is switched from tree additivity to ultrametricity: the closest pair must be sisters. The reason is that a distant outgroup will have a tendency to root incorrectly the tree consisting of the other three subtrees; as a result, to determine the sister group, it is more reliable to use the assumption of approximate constancy of evolutionary rates.

4. From Distance to Parsimony

The distance tree so obtained is used to make a parsimony tree, by aligning sequence graphs in an order such that the parsimony tree has the same topology as the distance tree (fig. 5). First the tree is rooted arbitrarily (fig. 5b). Then, sequences are aligned and ancestors are reconstructed in the order prescribed by this rooted distance tree. The resulting parsimony tree will have the same branching order as the

FIG. 7.—a, Locations of the major groups—plants, animals, fungi, bacteria, and chloroplasts—on the complete tree. This is called the compressed tree and has five leaves. The node of degree two (flanks two edges) corresponds to the most recent common ancestor to the group that has been compressed into one tip. b, Subtree involving animals.

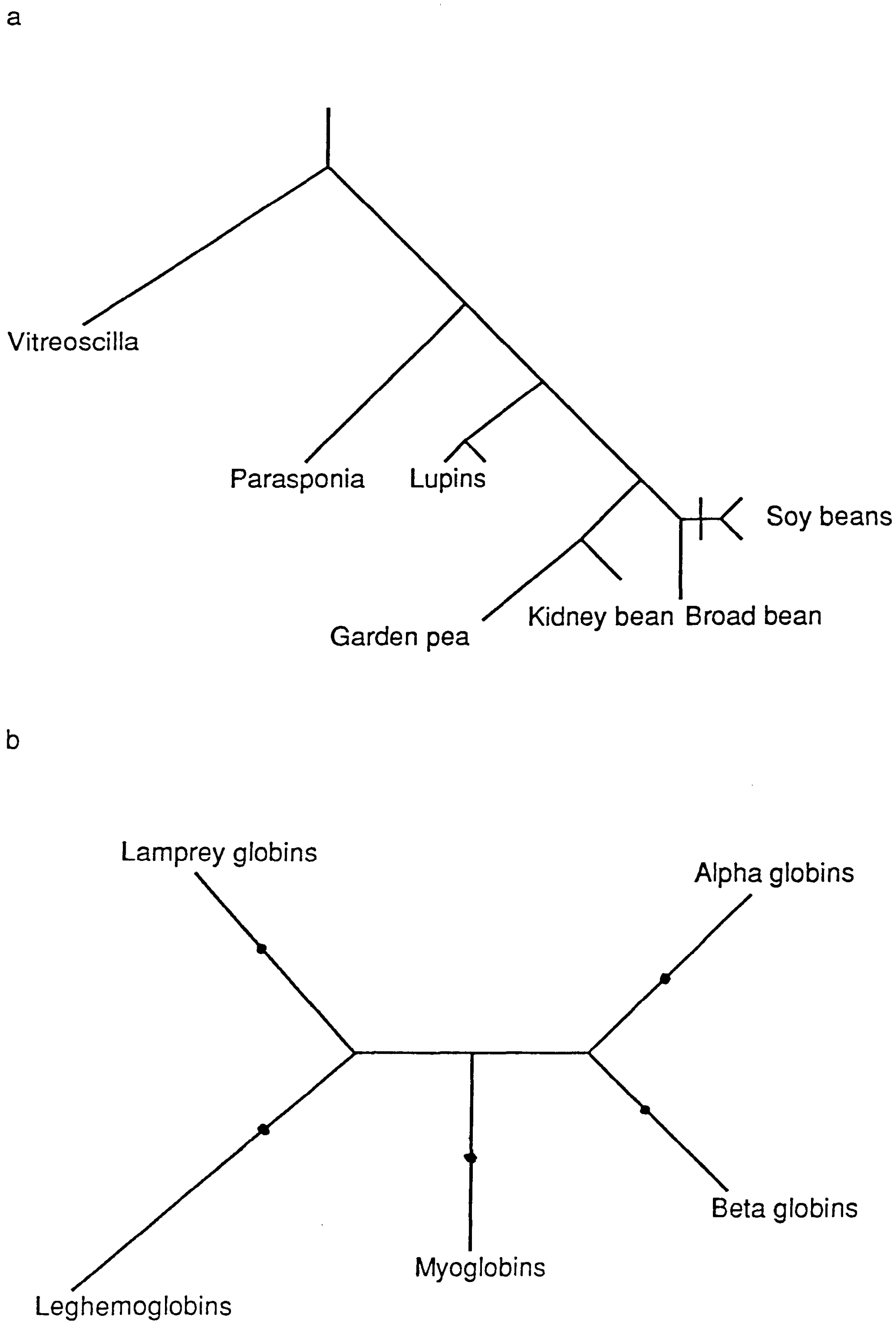


FIG. 8.—a, Marginal tree corresponding to the leghemoglobins, Vitreoscilla, and an outgroup. Lupin is represented by two sequences, and soybean is represented by four. The tree is in accordance with biological knowledge of the sequences. b, Location of the subtrees containing leghemoglobins (including Vitreoscilla) (11), alpha globins (77), beta globins (78), myoglobins (45), and lamprey globins (2). The total length of this phylogeny is 2,344. The nodes on the middle of the edges are the positions of the root of the subtree.

distance tree (fig. 5c), although branch lengths may differ. It will also have ancestral sequences assigned to internal nodes in the tree.

5. Nearest-Neighbor Interchange on the Parsimony Tree

The parsimony tree is subjected to cycles of nearest-neighbor interchanges. Each internal edge is visited in a cycle, nearest-neighbor interchanges are performed, and the most parsimonious tree is retained. These cycles of nearest-neighbor interchange at each internal edge are continued until either no improvement occurs in a complete cycle or some user-specified number of cycles has been completed. The method for aligning the sequences requires a root. When the three possible configurations of the four subtrees around one internal edge are investigated, the root is always on this edge; accordingly, when different edges are visited, the root must be moved. Figure 6 illustrates what happens if the root must be moved from its present position to the neighboring edge leading to T1. Ancestors in the new tree, not known in the previous tree, must be calculated. Here the ancestor of T2 and a2 is unknown—as is the ancestor of T1 and a1, which will be the new root. Thus it requires two realignments to move the root from an edge to a neighboring edge. When all internal edges are to be visited, it is often necessary to move the root over areas of the tree that have already been investigated, and the total number of realignments necessary to visit all internal edges is more than two per internal edge. The exact number depends on the nearest-neighbor interchanges performed. In addition, it costs six alignments to evaluate the two alternative configurations at each internal edge.

The nearest-neighbor interchanges on parsimony trees are computationally more expensive than those on distance trees, since alignments are costly. A computational speedup could be introduced here in cycle number two by not investigating those areas of the tree that have been stable in previous cycles.

Simplification of Large Phylogenies

A strength of the method presented here is its ability to analyze very large sets of data. Phylogenies of 200–300 sequences, however, are completely incomprehensible and will take days to analyze manually. The following describes a method that simplifies large phylogenies without losing essential information in the analysis.

When a large phylogeny is analyzed, there are two classes of important questions: first, there are questions involving the branching order within a specified subtree, and, second, there are questions about the relationship between specified subtrees. To simplify a large phylogeny in a relevant way according to the first kind of question is not difficult. One needs only to write out the subtree within which the problems are located.

The length from this middle node to the tip is the average length from a tip in the subtree to the root in the subtree that has been compressed to one tip. If the tree is rooted on the branch leading to the leghemoglobins, the history of the sequences is as follows: The first duplication created animal globins and the globins, including plants and Vitreoscilla, respectively. This corroborates the hypothesis that plants have acquired their globin from bacteria. Within the vertebrate globins the first split gave rise to the lamprey globins and to the lineage leading to alpha globins, beta globins, and myoglobin. This branching order emerged in several comparisons made by the author; it is in contradiction with results published by Goodman et al. (1988). The next split led to the appearance of myoglobin and to the ancestor to alpha globin and beta globin. The lengths of the branches leading to each group are split into two components as follows: (1) the length from the tip to the internal node above it represents the average length from the latest common ancestor to the present representatives, and (2) the length above this node is from the earliest common ancestor to (only) this group to the latest common ancestor.

Table 1
Genetic Events/Weighting Table and Indels

A. Replacements/Weights

	Cys	Ser	Thr	Pro	Ala	Gly	Asn	Asp	Glu	Gln	His	Arg	Lys	Met	Iso	Leu	Val	Phe
Cys		2	4	4	4	3	4	5	6	5	4	4	6	4	4	4	4	3
Ser	17		1	2	1	1	1	2	3	3	3	3	4	4	4	2	3	3
Thr	6	162		2	1	2	2	3	3	3	4	3	2	3	3	4	2	4
Pro	1	20	14		1	2	4	3	3	3	3	3	4	4	4	3	2	3
Ala	24	265	165	95		1	3	2	2	3	4	4	3	3	4	4	1	3
Gly	6	126	33	17	236		3	2	2	4	5	3	4	5	4	4	2	4
Asn	6	85	50	6	26	29		1	2	3	2	3	2	5	4	5	4	4
Asp	3	35	11	6	62	40	89		1	2	3	4	3	4	5	5	3	5
Glu	1	11	13	15	90	26	34	207		2	4	3	2	4	5	5	2	4
Gln	1	15	22	11	23	10	10	26	58		2	3	2	4	5	4	4	5
His	4	34	10	12	21	10	69	9	10	78		2	3	4	4	3	5	4
Arg	3	15	4	2	5	4	5	1	3	5	23		1	4	4	4	4	5
Lys	3	46	48	6	58	15	47	23	51	63	26	99		4	4	4	3	5
Met	1	0	6	3	18	2	0	0	4	1	2	2	9		2	1	2	3
Iso	6	10	23	1	27	4	6	1	1	1	2	1	2	21		1	4	2
Leu	3	11	14	7	41	2	2	7	4	15	24	10	21	83	113		4	2
Val	14	16	51	13	134	19	2	8	20	1	3	2	8	34	173	74		2
Phe	3	9	4	2	7	1	0	0	3	0	9	0	1	11	23	100	20	
Tyr	3	1	2	2	5	1	4	1	4	0	20	6	1	1	3	11	2	62
Trp	0	0	0	0	2	1	0	0	3	1	0	0	0	1	1	7	5	16

B. Indels

	LENGTH			
	1	2	3	4
Nos.	39	11	4	4

NOTE.—The upper triangle of the matrix shows the weights assigned to different mutations. The lower part of the matrix shows the observed numbers of the possible replacem that the numbers and lengths of the observed indels are tabulated.

Day (1985) has devised a fast algorithm that is relevant for the second type of problem. It determines whether a cluster is present on one subtree of a large tree. This algorithm was modified, with some cost in speed, to find the smallest number of subtrees on which all members of a cluster—and no others—are located.

As an illustration, consider figure 5 in the accompanying article (Hein 1989) that describes the evolution of 22 5S RNAs. An example of a question of the first kind could be, Is the branching order within the animals in accordance with known paleontological data? This can be answered by investigating the subtree consisting only of animals. The result, called a subtree, is shown in figure 7b. An example of a question of the second kind could be, Are the groups of animals, plants, fungi, prokaryotes, and chloroplasts related as expected, and are the single groups on the large tree? Since the phylogeny is small, it is easily seen that these clusters are located as they should be on the large tree. What has been done is that predefined nonoverlapping clusters are used to define subtrees on the complete phylogeny. The result, called a *compressed tree,* is shown in figure 7c for the tree. A program, condense.c, was written that performed both types of operations on large phylogenies.

An Illustration

A total of 213 vertebrate and plant globins were taken from the NBRF data base and were analyzed by the method described above. The program took 126 min to run on a VAX 11/730. Results from this analysis are shown in figure 8 and table 1. Of the 22,578 entries in the distance matrix, 1,683 were calculated before the tree construction. In addition, 341 more were needed in the tree construction. This is <10% of all pairwise distances; the fraction decreases as the number of sequences grows. The alignment takes an additional 212, corresponding to the number of internal nodes. The program also includes an option of using a user-defined tree, in cases where this should be known; in such a case the program is considerably faster. In the above example it would simply have skipped the 2,224 comparisons needed to make the tree and would only have performed the 212 alignments to reconstruct the history of the sequences.

The total weight of the resulting history was 10,008. The gap penalty function used was $g_k = 10 + (3 \times k)$, and the metric employed for amino acids is shown in table 1. In table 1 are shown the replacements and the indels in the reconstructed history of the sequences. The parameter used to determine how much of the initial distance matrix should be calculated was 132.4. A strong bias toward conservative replacements was observed.

Summary

The presented method has several advantages. It is fast and can be applied to very large data sets. In contrast to some other phylogeny programs, it is fully automated and does not necessitate any human interaction. Since it can be applied to large data sets, without loss of clarity, it should enhance efficient use of existing data bases by researchers with new sequences.

LITERATURE CITED

BUNEMAN, P. 1971. The recovery of trees from measures of dissimilarity. Pp. 387–395 *in* F. R. HODSON, D. G. KENDALL, and P. TAUTU, eds. Mathematics in the archaeological and historical sciences. Edinburgh University Press, Edinburgh.

DAY, W. H. E. 1985. Optimal algorithms for comparing trees with labeled leaves. J. Class. **2:** 7–28.

DAYHOFF, M. O., R. V. ECK, and C. M. PARK. 1972. A model of evolutionary change in proteins. Pp. 89–99 *in* M. O. DAYHOFF, ed. Atlas of protein sequence and structure. National Biomedical Research Foundation, Washington, D.C.

FITCH, W. M. 1981. A non-sequential method for constructing trees and hierarchic classification. J. Mol. Evol. **18:**30–37.

GOODMAN, M., J. PEDWAYDON, J. CZELUSNIAK, T. SUZUKI, T. GOTOH, L. MOENS, F. SHISHIKURA, D. WATZ, and S. VINOGRADOV. 1988. An evolutionary tree for invertebrate globin sequences. J. Mol. Evol. **27:**236–249.

HEIN, J. J. 1989. A new method that simultaneously aligns and reconstructs ancestral sequences for any number of homologous sequences, when the phylogeny is given. Mol. Biol. Evol. **6:** 649–668.

———. An optimal algorithm that reconstructs phylogenies from tree-additive data. Bull. Math. Biol. (in press).

JUKES, T. H., and C. R. CANTOR. 1969. Evolution of protein molecules. Pp. 21–132 *in* H. N. MUNRO, ed. Mammalian protein metabolism. Academic Press, New York.

ROBINSON, D. F. 1971. Comparing labelled trees with valency three. J. Combinatorial Theor. **11:**105–119.

WATERMAN, M. S., T. F. SMITH, M. SINGH, and W. A. BEYER. 1977. Additive evolutionary trees. J. Theor. Biol. **64:**199–213.

WALTER M. FITCH, reviewing editor

Received April 13, 1989; revision received July 14, 1989

Accepted July 17, 1989