The Neighbor-joining Method: A New Method for Reconstructing Phylogenetic Trees[1]

Naruya Saitou[2] *and Masatoshi Nei*

Center for Demographic and Population Genetics, The University of Texas Health Science Center at Houston

A new method called the neighbor-joining method is proposed for reconstructing phylogenetic trees from evolutionary distance data. The principle of this method is to find pairs of operational taxonomic units (OTUs [=neighbors]) that minimize the total branch length at each stage of clustering of OTUs starting with a starlike tree. The branch lengths as well as the topology of a parsimonious tree can quickly be obtained by using this method. Using computer simulation, we studied the efficiency of this method in obtaining the correct unrooted tree in comparison with that of five other tree-making methods: the unweighted pair group method of analysis, Farris's method, Sattath and Tversky's method, Li's method, and Tateno et al.'s modified Farris method. The new, neighbor-joining method and Sattath and Tversky's method are shown to be generally better than the other methods.

Introduction

In the construction of phylogenetic trees, the principle of minimum evolution or maximum parsimony is often used. The standard algorithm of the tree-making methods based on this principle is to examine all possible topologies (branching patterns) or a certain number of topologies that are likely to be close to the true tree and to choose one that shows the smallest amount of total evolutionary change as the final tree. This method is quite time consuming, and, when the number of operational taxonomic units (OTUs) is large, only a small proportion of all possible topologies is examined. However, there are methods in which the process of searching for the minimum evolution tree is built into the algorithm, so that a unique final topology is obtained automatically. Some examples are the distance Wagner (DW) method (Farris 1972), modified Farris (MF) methods (Tateno et al. 1982; Faith 1985), and the neighborliness methods of Sattath and Tversky (ST method; 1977) and Fitch (1981). These methods are not guaranteed to produce the minimum-evolution tree, but their efficiency in obtaining the correct tree is often better than that of the standard maximum-parsimony algorithm (Saitou and Nei 1986). In the following we would like to present a new method (the neighbor-joining [NJ] method) that produces a unique final tree under the principle of minimum evolution. This method also does not necessarily produce the minimum-evolution tree, but computer simulations have shown that it

1. Key words: phylogenetic tree, neighbor-joining method, minimum-evolution tree, parsimonious tree.

2. Current address: Department of Anthropology, University of Tokyo, Tokyo 113, Japan.

Address for corespondence and reprints: Dr. Masatoshi Nei, Center for Demographic and Population Genetics, Graduate School of Biomedical Sciences, The University of Texas Health Science Center at Houston, P.O. Box 20334, Houston, Texas 77225.

Mol. Biol. Evol. 4(4):406–425. 1987.

0737-4038/87/0404-0007$02.00

is quite efficient in obtaining the correct tree topology. It is applicable to any type of evolutionary distance data.

Algorithm

The algorithm of the NJ method is similar to that of the ST method, whose objective is to construct the topology of a tree. Unlike this method, however, the NJ method provides not only the topology but also the branch lengths of the final tree.

Before discussing the algorithm of the present method, let us first define the term "neighbors." A pair of neighbors is a pair of OTUs connected through a single interior node in an unrooted, bifurcating tree. Thus, OTUs 1 and 2 in figure 1 are a pair of neighbors because they are connected through one interior node, A. There are two other pairs of neighbors in this tree (viz., [5, 6] and [7, 8]). The number of pairs of neighbors in a tree depends on the tree topology. For a tree with N (≥ 4) OTUs, the minimum number is always two, whereas the maximum number is $N/2$ when N is an even number and $(N-1)/2$ when N is an odd number.

If we combine OTUs 1 and 2 in figure 1, this combined OTU (1-2) and OTU 3 become a new pair of neighbors. It is possible to define the topology of a tree by successively joining pairs of neighbors and producing new pairs of neighbors. For example, the topology of the tree in figure 1 can be described by the following pairs of neighbors: [1, 2], [5, 6], [7, 8], [1-2, 3], and [1-2-3, 4]. Note that there is another pair of neighbors, [5-6, 7-8], that is complementary to [1-2-3, 4] in defining the topology. In general, $N-2$ pairs of neighbors can be produced from a bifurcating tree of N OTUs. By finding these pairs of neighbors successively, we can obtain the tree topology.

Our method of constructing a tree starts with a starlike tree, as given in figure 2(a), which is produced under the assumption that there is no clustering of OTUs. In

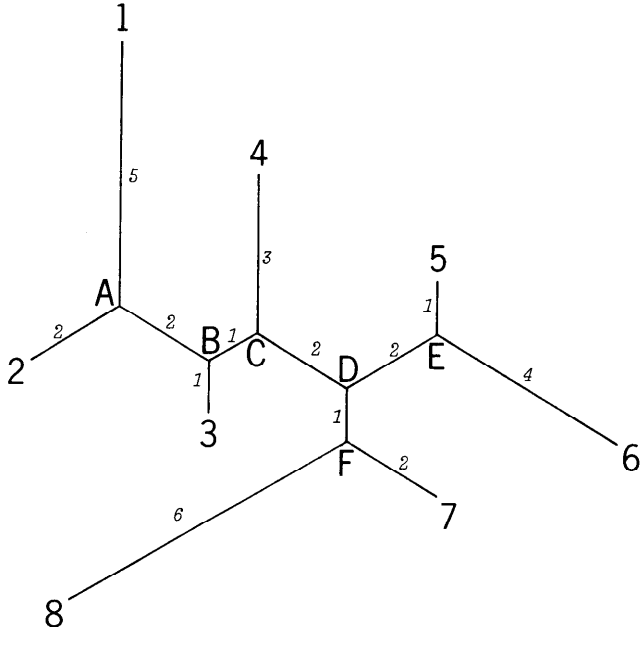

FIG. 1.—An unrooted tree of eight OTUs, 1–8. A–F are interior nodes, and italic numbers are branch lengths.

practice, some pairs of OTUs are more closely related to each other than other pairs are. Consider a tree that is of the form given in figure 2(b). In this tree there is only one interior branch, XY, which connects the paired OTUs (1 and 2) and the others (3, 4, . . . , N) that are connected by a single node, Y. Any pair of OTUs can take the positions of 1 and 2 in the tree, and there are $N(N-1)/2$ ways of choosing them. Among these possible pairs of OTUs, we choose the one that gives the smallest sum of branch lengths. This pair of OTUs is then regarded as a single OTU, and the next pair of OTUs that gives the smallest sum of branch lengths is again chosen. This procedure is continued until all $N-3$ interior branches are found.

The sum of the branch lengths is computed as follows: Let us define D_{ij} and L_{ab} as the distance between OTUs i and j and the branch length between nodes a and b, respectively. The sum of the branch lengths for the tree of figure 2(a) is then given by

$$S_O = \sum_{i=1}^{N} L_{iX} = \frac{1}{N-1} \sum_{i<j} D_{ij}, \quad (1)$$

since each branch is counted $N-1$ times when all distances are added. On the other hand, the branch length between nodes X and Y (L_{XY}) in the tree of figure 2(b) is given by

$$L_{XY} = \frac{1}{2(N-2)} \left[\sum_{k=3}^{N} (D_{1k} + D_{2k}) - (N-2)(L_{1X} + L_{2X}) - 2 \sum_{i=3}^{N} L_{iY} \right]. \quad (2)$$

The first term within the brackets of equation (2) is the sum of all distances that include L_{XY}, and the other two terms are for excluding irrelevant branch lengths. If we eliminate the interior branch (XY) from figure 2(b), two starlike topologies (one for OTUs 1 and 2 and the other for the remaining $N-2$ OTUs) appear. Thus, $L_{1X} + L_{2X}$ and $\sum_{i=3}^{N} L_{iY}$ can be obtained by applying equation (1):

$$L_{1X} + L_{2X} = D_{12}, \quad (3a)$$

$$\sum_{i=3}^{N} L_{iY} = \frac{1}{N-3} \sum_{3 \le i < j} D_{ij}. \quad (3b)$$

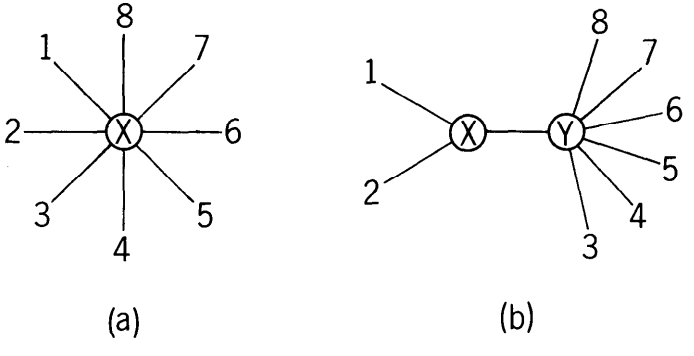

FIG. 2.—(a), A starlike tree with no hierarchical structure; and (b), a tree in which OTUs 1 and 2 are clustered.

Adding these branch lengths, we find that the sum (S_{12}) of all branch lengths of the tree in figure 2(b) becomes

$$S_{12} = L_{XY} + (L_{1X} + L_{2X}) + \sum_{i=3}^{N} L_{iY}$$
$$= \frac{1}{2(N-2)} \sum_{k=3}^{N} (D_{1k} + D_{2k}) + \frac{1}{2} D_{12} + \frac{1}{N-2} \sum_{3 \le i < j} D_{ij}. \quad (4)$$

It can be shown that equation (4) is the sum of the least-squares estimates of branch lengths (see Appendix A).

In general, we do not know which pairs of OTUs are true neighbors. Therefore, the sum of branch lengths (S_{ij}) is computed for all pairs of OTUs, and the pair that shows the smallest value of S_{ij} is chosen (inferred) as a pair of neighbors. In practice, even this pair may not be a pair of true neighbors; but, for a purely additive tree with no backward and parallel substitutions, this method is known to choose pairs of true neighbors (see the following section—Criterion for Minimum-Evolution Tree—for detail). At any rate, if S_{12} is found to be smallest among all S_{ij} values, OTUs 1 and 2 are designated as a pair of neighbors, and these are joined to make a combined OTU (1-2). The distance between this combined OTU and another OTU j is given by

$$D_{(1\text{-}2)j} = (D_{1j} + D_{2j})/2 \qquad (3 \le j \le N). \quad (5)$$

Thus, the number of OTUs is reduced by one, and, for the new distance matrix, the above procedure is again applied to find the next pair of neighbors. This cycle is repeated until the number of OTUs becomes three, where there is only one unrooted tree.

The branch lengths of a tree can be estimated by using Fitch and Margoliash's (1967) method. Suppose that OTUs 1 and 2 are the first pair to be joined in the tree of figure 1. L_{1X} and L_{2X} are then estimated by

$$L_{1X} = (D_{12} + D_{1Z} - D_{2Z})/2, \quad (6a)$$

$$L_{2X} = (D_{12} + D_{2Z} - D_{1Z})/2, \quad (6b)$$

where $D_{1Z} = (\sum_{i=3}^{N} D_{1i})/(N-2)$ and $D_{2Z} = (\sum_{i=3}^{N} D_{2i})/(N-2)$. Here, Z represents a group of OTUs including all but 1 and 2, and D_{1Z} and D_{2Z} are the distances between 1 and Z and 2 and Z, respectively (see Nei 1987, pp. 298–302, for an elementary exposition of this method). L_{1X} and L_{2X} are the least-squares estimates for the tree of figure 2(b) (see Appendix A), and they are estimates of L_{1A} and L_{2A}, respectively, in figure 1. Once L_{1A} and L_{2A} are estimated, OTUs 1 and 2 are combined as a single OTU (1-2), and the next neighbors are searched for. Suppose that (1-2) and 3 are the next neighbors to be joined, as in figure 1. Branch lengths $L_{(1\text{-}2)B}$ and L_{3B} are obtained by applying equations (6a) and (6b). Furthermore, L_{AB} is estimated by $L_{(1\text{-}2)B} - (D_{12})/2$. The above procedure is applied repeatedly until all branch lengths are estimated. If a tree is purely additive, this method gives the correct branch lengths for all branches (see Appendix B).

The principle of the NJ method can be extended to character-state data such as nucleotide or amino acid differences. In this case, one can use the total number of

Table 1
Distance Matrix for the Tree in Figure 1

	OTU						
OTU	1	2	3	4	5	6	7
2 . .	7						
3 . .	8	5					
4 . .	11	8	5				
5 . .	13	10	7	8			
6 . .	16	13	10	11	5		
7 . .	13	10	7	8	6	9	
8 . .	17	14	11	12	10	13	8

substitutions in place of the sum of branch lengths (S_{ij}), though the actual procedure is a little more complicated than that given above (Saitou 1986, pp. 90–98). However, since the algorithm turns out to be very similar to that of Hartigan (1973), we shall not present it here. Note also that most character-state data can be converted into distance data so that the above simpler algorithm applies.

An example: consider the distance matrix given in table 1. The distance D_{ij} in this matrix is obtained by adding all relevant branch lengths between OTUs i and j in figure 1 under the assumption that there is no backward and parallel substitution. The result of application of the NJ method is presented in table 2 and figure 3. In the

Table 2
S_{ij} Matrices for Two Cycles of the NJ Method for the Data in Table 1

A. Cycle 1: Neighbors = [1, 2]

	OTU						
OTU	1	2	3	4	5	6	7
2 . .	36.67						
3 . .	38.33	38.33					
4 . .	39.00	39.00	38.67				
5 . .	40.33	40.33	40.00	39.67			
6 . .	40.33	40.33	40.00	39.67	37.00		
7 . .	40.17	40.17	39.83	39.50	38.83	38.83	
8 . .	40.17	40.17	39.83	39.50	38.83	38.83	37.67

B. Cycle 2: Neighbors = [5, 6]

	OTU					
OTU	1-2	3	4	5	6	7
3 . .	31.50					
4 . .	32.30	32.30				
5 . .	33.90	33.90	33.70			
6 . .	33.90	33.90	33.70	31.30		
7 . .	33.70	33.70	33.50	33.10	33.10	
8 . .	33.70	33.70	33.50	33.10	33.10	31.90

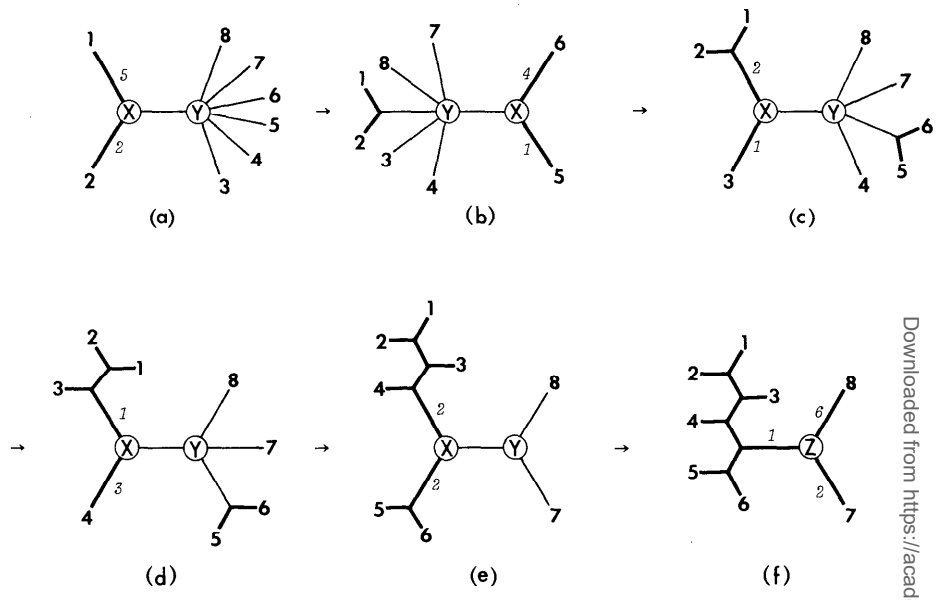

FIG. 3.—Application of the neighbor-joining method to the distance matrix of table 1. Italic numbers are branch lengths, and branches with thicker lines indicate that their lengths have been determined.

search for the first pair of neighbors (cycle 1), OTUs 1 and 2 are chosen because S_{12} is smallest among 28 S_{ij}'s (see table 2). S_{12} (= 36.67) is smaller than the sum (S_O = 39.28) of branch lengths of the starting starlike topology, but, interestingly, some S_{ij}'s are larger than S_O. D_{1Z} and D_{2Z} in equations (6a) and (6b) become 13 and 10, respectively. Thus, the branch lengths L_{1A} and L_{2A} are obtained to be $(7 + 13 - 10)/2 = 5$ and $(7 + 10 - 13)/2 = 2$, respectively, which are identical with those of the true tree in figure 1 (fig. 3[a]). OTUs 1 and 2 are then combined, and the average distances ($D_{(1\text{-}2)j}$; $j = 3, \ldots, 8$) are computed by equation (5). In the next step (cycle 2 in table 2), OTUs 5 and 6 are found to be a pair of neighbors, and L_{5E} and L_{6E} are estimated to be 1 and 4, respectively, which are again identical with those of the true tree (fig. 3[b]). In cycle 3, OTUs (1-2) and 3 are chosen as a pair of neighbors, and the branch lengths for L_{3B} and $L_{(1\text{-}2)B}$ become 1 and 5.5, respectively. Thus, the branch length L_{AB} is estimated to be $5.5 - 7/2 = 2$. These are again the correct values (fig. 3[c]). In cycle 4, [1-2-3, 4] is identified as a pair of neighbors (fig. 3[d]), and in cycle 5 [1-2-3-4, 5-6] is chosen. The choice of the latter pair of neighbors automatically leads to the identification of the final pair of neighbors [7, 8]. The S_{ij} for [1-2-3-4, 5-6] is identical with that for [7, 8]. The topology of the reconstructed tree is therefore given by figure 3(e), which is identical with that of figure 1. The branch lengths L_{7F} ($=L_{7Z} = 2$) and L_{8F} ($=L_{8Z} = 6$) are obtained by using equations (6a) and (6b), whereas L_{DF} becomes $L_{(1\text{-}2\text{-}3\text{-}4\text{-}5\text{-}6)Z} - D_{(1\text{-}2\text{-}3\text{-}4)(5\text{-}6)}/2 = 2$ (fig. 3[f]). It is thus clear that all branch lengths as well as the topology are correctly reconstructed in the present case.

Criterion for the Minimum-Evolution Tree

In this section, we first show that the algorithm developed above produces the correct tree for a purely additive tree. We shall then discuss a criterion for the minimum-evolution tree.

Consider a tree for N (≥ 4) OTUs and assume that OTUs 1 and 2 are a pair of true neighbors. For an additive tree, we obviously have the following inequalities.

$$D_{12}+D_{ij}<D_{1i}+D_{2j}, \qquad D_{12}+D_{ij}<D_{1j}+D_{2i}, \tag{7}$$

where i and j are any OTUs ($3 \leq i < j \leq N$). Under this condition, it can be shown that S_{12} is smaller than S_{1j} or S_{2j} ($3 \leq j \leq N$). To show this, let us consider the pairing of OTUs 1 and 3 as an example. The total length of the tree with this pairing can be written as

$$S_{13}=\frac{1}{N-2}\sum_{i<j}D_{ij}-\frac{1}{2(N-2)}\sum_{\substack{k=2\\k\neq 3}}^{N}(D_{1k}+D_{3k})+\frac{N-4}{2(N-2)}D_{13}. \tag{8a}$$

In a similar manner,

$$S_{12}=\frac{1}{N-2}\sum_{i<j}D_{ij}-\frac{1}{2(N-2)}\sum_{k=3}^{N}(D_{1k}+D_{2k})+\frac{N-4}{2(N-2)}D_{12}. \tag{8b}$$

Hence,

$$\begin{aligned} S_{13}-S_{12} &= \frac{N-3}{2(N-2)}(D_{13}-D_{12})+\frac{1}{2(N-2)}\sum_{k=4}^{N}(D_{2k}-D_{3k}) \\ &= \frac{1}{2(N-2)}\sum_{k=4}^{N}[(D_{13}+D_{2k})-(D_{12}+D_{3k})]. \end{aligned} \tag{9}$$

If we note the inequalities in formula (7), $D_{12} + D_{3k} < D_{13} + D_{2k}$ ($4 \leq k \leq N$). Therefore, $S_{13} > S_{12}$. The same inequality also holds for any other pairs involving OTUs 1 and 2: $S_{1j} > S_{12}$ and $S_{2j} > S_{12}$ ($3 \leq j \leq N$). Furthermore, in our algorithm we search for a pair of OTUs that shows the smallest S_{ij}. Therefore, if OTUs 1 and 2 are such a pair, S_{12} must be smallest among all S_{ij}'s. However, this is not what we need in our algorithm. Our algorithm requires that if S_{12} is smallest among all S_{ij}'s, OTUs 1 and 2 are neighbors. Proof of this theorem is somewhat complicated, but it can be done (see Appendix C). Therefore, our algorithm produces the correct unrooted tree for a purely additive tree.

Of course, actual data usually involve backward and parallel substitutions, so that there is no guarantee that the correct topology is obtained by the NJ method. However, computer simulations, which will be discussed below, have shown that, compared with other methods, the NJ method is efficient in obtaining the correct topology.

In constructing the topology of a tree, Sattath and Tversky (1977) and Fitch (1981) used the inequalities in formula (7). Their method is to count the number (neighborliness) of cases satisfying formula (7) for each pair of OTUs and choose the pair showing the largest number as neighbors. Since Sattath and Tversky's (1977) algorithm uses equation (5) for making the new distance matrix, their method is expected to give a result similar to ours. Fitch (1981) uses interior-distance matrices for constructing the topology, so that his algorithm is different from ours. Nevertheless, these three methods as well as some other tree-making methods require the same

condition for obtaining the correct topology for the case of four OTUs, as shown below.

Let us consider the tree of four OTUs given in figure 4. Saitou and Nei (1986) showed that the condition for obtaining the correct unrooted tree for four OTUs is the same for the DW method (Farris 1972), the MF methods (Tateno et al. 1982; Faith 1985), and the transformed-distance method (Farris 1977; Klotz and Blanken 1981; Li 1981). It is given by

$$D_{12}+D_{34}<D_{13}+D_{24}, \qquad D_{12}+D_{34}<D_{14}+D_{23}. \tag{10}$$

The same condition is required for the NJ method. When $N = 4$, equation (4) reduces to

$$S_{12}=(D_{13}+D_{14}+D_{23}+D_{24})/4+(D_{12}+D_{34})/2. \tag{11a}$$

We also have $S_{34} = S_{12}$, and

$$S_{13}=S_{24}=(D_{12}+D_{14}+D_{23}+D_{34})/4+(D_{13}+D_{24})/2, \tag{11b}$$

$$S_{14}=S_{23}=(D_{12}+D_{13}+D_{24}+D_{34})/4+(D_{14}+D_{23})/2. \tag{11c}$$

In figure 4, [1, 2] and [3, 4] are pairs of neighbors. Thus $S_{12} < S_{13}$ and $S_{12} < S_{14}$ is the necessary condition to obtain the correct topology. From this condition and equations (11), we obtain formula (10). The inequalities in formula (10) are also the condition required for neighborliness methods (Sattath and Tversky 1977; Fitch 1981) to produce the correct topology. The condition posited by formula (10) is similar to "the four-point condition" (Buneman 1971) or "relaxed additivity" (Fitch 1981).

The condition posited by formula (10) may be used as a criterion for the minimum-evolution tree (minimality test). If this condition holds for any group of four OTUs of a reconstructed tree, the tree is likely to be the minimum-length tree (Fitch 1981). Furthermore, this condition can be extended to test each interior branch of a tree. Let us consider the interior branch CD of the tree in figure 1 as an example. If this branch really exists, the following inequalities should be satisfied.

$$\begin{aligned} D_{(1\text{-}2\text{-}3)4}+D_{(5\text{-}6)(7\text{-}8)}&<D_{(1\text{-}2\text{-}3)(5\text{-}6)}+D_{4(7\text{-}8)}, \\ D_{(1\text{-}2\text{-}3)4}+D_{(5\text{-}6)(7\text{-}8)}&<D_{(1\text{-}2\text{-}3)(7\text{-}8)}+D_{4(5\text{-}6)}, \end{aligned} \tag{12}$$

where $D_{(1\text{-}2\text{-}3)4} = (D_{14} + D_{24} + D_{34})/3$, $D_{(5\text{-}6)(7\text{-}8)} = (D_{57} + D_{67} + D_{58} + D_{68})/4$, and so on. Numerical computation shows that this is indeed the case. If we conduct a similar test for the five remaining interior branches of the tree in figure 1, the existence of all branches is justified.

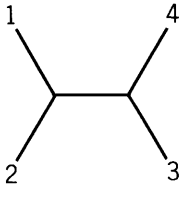

FIG. 4.—An unrooted tree for four OTUs

An example: applying his neighborliness method to Case's (1978) data on immunological distance, Fitch (1981) constructed a phylogenetic tree of nine frog (*Rana*) species. If we use the NJ method, a slightly different tree is obtained (fig. 5); that is, while the closest species to the *R. aurora-R. boylii* group is *R. cascadae* in Fitch's tree, it is *R. muscosa* in our tree. The latter topology is also obtained by the ST method. We can apply the minimality test in formula (10) to see which topology is more reasonable. The test can be done if we consider the four OTU groups, i.e., the *aurora* and *boylii* group, *muscosa, cascadae,* and the remaining five species. Application of the test supports the topology presented in figure 5 rather than Fitch's. Comparison of the sum of branch lengths between the two topologies also supports the topology in figure 5. (This particular comparison was conducted under the condition that all branch lengths are nonnegative and that each estimated [patristic] distance is greater than or equal to the corresponding observed distance, because Fitch's tree was constructed under this condition.) We also note that the branch lengths estimated by the NJ method are close to those estimated by a linear programming method (see Fitch 1981).

Efficiency of the NJ Method in Recovering the Correct Topology

Since the exact evolutionary pathways of extant organisms are usually unknown, it is not suitable to use real data for examining the efficiency of a tree-making method. Therefore, we employed a computer simulation, comparing reconstructed trees with their model trees. In this study we compared the efficiency of the NJ method with that of five other methods: UPGMA (Sokal and Sneath 1963), the DW method, the ST method, Li's (LI; 1981) method, and the MF method. The LI method is a transformed distance method (see Nei [1987, pp. 302–305] for the explanation of the transformed distance method), and the MF method is a modification of Farris's (1972) method. All these methods produce a unique parsimonious tree from distance data. We considered both cases of constant and varying (expected) rates of nucleotide substitution.

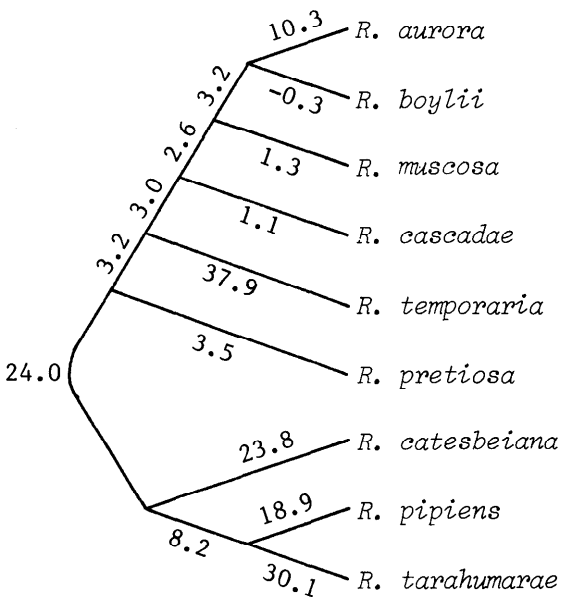

FIG. 5.—Tree obtained by the NJ method from immunological distance data of Case (1978)

Constant Rate of Nucleotide Substitution

To examine the effect of topological differences, we considered two different model trees (trees [A] and [B] of fig. 6), both of which consist of eight OTUs. Model tree (A) has two neighboring pairs ([1, 2] and [7, 8]), whereas model tree (B) has four ([1, 2], [3, 4], [5, 6], and [7, 8]). To make the effect of branch lengths comparable for the two model trees, we assumed that the interior branch length (a) is the same for both trees. We also tried to make the average ($\bar{D}$) of all pairwise distances (D_{ij}'s) nearly the same for the two trees. Hence, we set $c = b + 3a$ or $c \approx b + 3a$, where a, b, and c are the expected branch lengths (expected numbers of nucleotide substitutions per site) given in figure 6. In a computer simulation conducted with the same topology as that of model tree (A), Tateno et al. (1982) set $a = b$. In the present study, we set $a \ll b$ in model tree (A) so that the differences between different D_{ij}'s were relatively smaller. This makes it more difficult to reconstruct the correct tree than in the case of Tateno et al.'s simulation.

The scheme of the computer simulation used is as follows: The ancestral sequence of a given number of nucleotides was generated by using pseudorandom numbers, and this sequence was assumed to evolve according to the predetermined branching pattern of the model tree. Random nucleotide substitutions were introduced in each branch of the tree following a Poisson distribution with the mean equal to the expected branch length. Although the expected rate of nucleotide substitution was the same for all lineages, the actual number of substitutions varied considerably with lineage because of stochastic errors. After the nucleotide sequences for eight OTUs were produced, nucleotide differences were counted for all pairs of sequences, and the evolutionary distance (Jukes and Cantor 1969) was computed for each pair of OTUs. With the six tree-making methods mentioned above, tree topologies were determined from data either on the proportion of different nucleotides between the two sequences compared (p) or on the Jukes-Cantor distance (d). Note that p is a metric, whereas d is not. The entire process of simulation was repeated 100 times.

Two measures are used to quantify the efficiency of a tree-making method in recovering the topology of the model tree. One is the proportion (P_c) of correct trees (topologies) obtained. The other is the average distortion index (Tateno et al. 1982) based on Robinson and Foulds' (1981) metric on tree comparison. The distortion index (d_T) is twice the number of branch interchanges required for a reconstructed tree to be converted to the true tree. Here, we consider only unrooted trees.

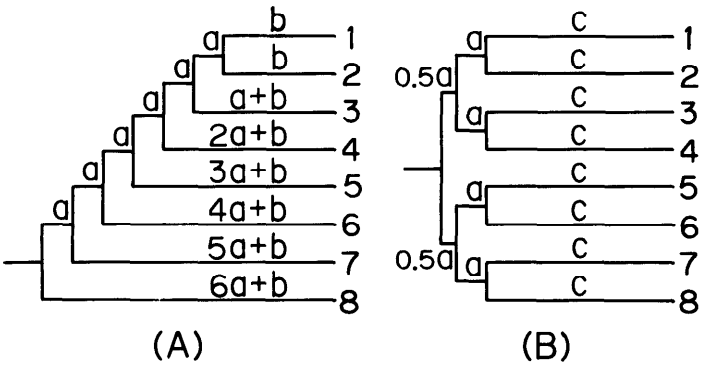

FIG. 6.—Model trees (A) and (B) under the assumption of constant rate of nucleotide substitution

Table 3
P_c and d_T (in parentheses) for Six Tree-making Methods for the Case of $a = 0.01$, $b = 0.04$, and $c = 0.07$

	Model Tree A[a]			Model Tree B[a]		
Method	300	600	900	300	600	900
UPGMA:						
p[b]	14 (3.18)	36 (1.72)	58 (0.98)	14 (4.54)	36 (2.74)	51 (1.68)
d[c]	15 (3.18)	34 (1.74)	56 (1.04)	13 (4.56)	35 (2.70)	52 (1.60)
MF:						
p	39 (1.76)	73 (0.58)	95 (0.10)	24 (2.86)	51 (1.30)	67 (0.26)
d	38 (1.92)	72 (0.62)	95 (0.10)	19 (2.94)	48 (1.42)	64 (0.86)
DW:						
p	42 (1.70)	75 (0.54)	96 (0.08)	26 (2.36)	55 (1.12)	79 (0.48)
d	37 (1.74)	74 (0.58)	95 (0.10)	28 (2.36)	58 (1.06)	79 (0.46)
LI:						
p	41 (1.58)	71 (0.70)	94 (0.12)	40 (2.04)	70 (0.78)	90 (0.22)
d	36 (1.84)	66 (0.82)	89 (0.24)	39 (2.10)	70 (0.78)	90 (0.26)
ST:						
p	48 (1.26)	75 (0.54)	97 (0.06)	45 (1.66)	75 (0.62)	91 (0.22)
d	44 (1.48)	70 (0.62)	96 (0.08)	43 (1.62)	74 (0.64)	91 (0.22)
NJ:						
p	48 (1.36)	76 (0.54)	97 (0.06)	46 (1.64)	76 (0.60)	91 (0.20)
d	41 (1.60)	70 (0.62)	96 (0.08)	45 (1.62)	75 (0.60)	91 (0.20)

[a] As shown in fig. 6.
[b] Trees reconstructed from data on the proportion of different nucleotides between the sequences compared.
[c] Trees reconstructed from the Jukes-Cantor distance.

Table 3 shows the results for the case of $a = 0.01$, $b = 0.04$, and $c = 0.07$, where the $\bar{D}$ for all OTUs is 0.16 for both model trees. It is clear that in all tree-making methods P_c increases as the number of nucleotides used (n) increases, whereas d_T decreases. This is of course due to the fact that the sampling error of the distance between a pair of OTUs decreases as n increases. The P_c and d_T values obtained by using p and d are nearly the same, though p tends to show a better performance in recovering the correct topology, particularly for model tree (A).

In the case of model tree (A) UPGMA shows the poorest performance in terms of both criterion P_c and criterion d_T. Even when 900 nucleotides are used, the proportion of correct trees obtained is ~57%. The other five tree-making methods show a much better performance than UPGMA, and when 900 nucleotides are used, P_c is ~95%. Interestingly, all of them show a similar performance for all n's examined. In the case of model tree (B), UPGMA again shows a poorer performance than any other method. In this case, however, all the five methods do not show the same performance. Rather, the NJ and the ST methods are better than the LI method, which is in turn better than the DW and MF methods.

The results for the case of $a = 0.02$, $b = 0.13$, and $c = 0.19$ are presented in table 4. The $\bar{D}$ for this case is 0.42 for model tree (A) and 0.43 for model tree (B). For model tree (A), UPGMA shows an improved performance compared with the case in table 3. However, all other methods show a small value of P_c and a larger value of d_T than those in table 3. This is apparently due to the fact that there are more backward

Table 4
P_c and d_T (in parentheses) for Six Tree-making Methods for the Case of $a = 0.02$, $b = 0.13$, and $c = 0.19$

	Model Tree A[a]			Model Tree B[a]		
Method	300	600	900	300	600	900
UPGMA:						
p	15 (3.24)	50 (1.32)	62 (0.82)	11 (4.62)	28 (2.94)	54 (1.48)
d	15 (3.28)	49 (1.34)	61 (0.84)	13 (4.50)	30 (2.90)	57 (1.44)
MF:						
p	34 (2.38)	65 (0.82)	79 (0.44)	10 (4.00)	25 (2.22)	43 (1.48)
d	30 (2.70)	62 (1.02)	76 (0.54)	9 (4.12)	22 (2.28)	43 (1.48)
DW:						
p	27 (2.40)	66 (0.96)	77 (0.54)	17 (3.54)	39 (1.92)	54 (1.10)
d	27 (2.52)	62 (1.02)	70 (0.70)	18 (3.54)	36 (1.98)	53 (1.16)
LI:						
p	23 (2.60)	44 (1.34)	67 (0.80)	25 (3.54)	50 (1.52)	81 (0.52)
d	20 (2.82)	33 (1.78)	55 (1.12)	20 (3.70)	49 (1.54)	81 (0.50)
ST:						
p	35 (2.06)	67 (0.74)	82 (0.38)	34 (2.40)	60 (1.08)	82 (0.38)
d	26 (2.42)	61 (0.96)	78 (0.48)	31 (2.50)	58 (1.16)	83 (0.36)
NJ:						
p	36 (2.14)	64 (0.88)	83 (0.34)	34 (2.32)	63 (0.96)	82 (0.36)
d	26 (2.38)	58 (1.08)	78 (0.48)	33 (2.56)	61 (1.04)	83 (0.34)

NOTE.—Notations are as in table 3.
[a] As shown in fig. 6.

and parallel substitutions involved in this case. Nevertheless, UPGMA still shows a poorer performance than all other methods except LI, which is less efficient than UPGMA for the case of $n = 600$. The NJ, ST, DW, and MF methods give similar results, though the first two methods give slightly better results than the others for $n = 900$. We also note that p gives a better result than d for all methods but UPGMA, for which both p and d give essentially the same results. In the case of model tree (B), the P_c values for UPGMA are not necessarily higher than those in table 3, but they are higher than those for the MF method for the same case. The DW method also shows a rather poor performance, though it is slightly better than the UPGMA and MF methods. The NJ and ST methods again show the best performance, but their P_c values are slightly lower than those for the case of table 3. The LI method is quite good but not as good as the NJ and ST methods. Interestingly, p and d give similar results for all methods, unlike the case of model tree (A).

Table 5 shows the results for the case of $a = 0.03$, $b = 0.34$, $c = 0.42$, and $\bar{D} = 0.92$ for tree (A) and 0.91 for tree (B). Compared with the two previous cases, the frequency of backward and parallel substitutions is expected to be much higher because of the larger D_{ij} values used. Therefore, we used $n = 500$, 1,000, and 2,000 for this case. Yet, the P_c values are smaller than those for the two previous cases. The relative merits of different tree-making methods for the case of model tree (A) are more or less the same as those for the case of table 4, except that the LI method tends to show a poorer performance than UPGMA. When $n = 500$, the MF and DW methods show a slightly higher value of P_c than the ST and NJ methods, but for the other two n

Table 5
P_c and d_T (in parentheses) for Six Tree-making Methods for the Case of $a = 0.03$, $b = 0.34$, and $c = 0.42$

	Model Tree A[a]			Model Tree B[a]		
Method	500	1,000	2,000	500	1,000	2,000
UPGMA:						
p	9 (3.78)	27 (2.10)	62 (0.86)	10 (5.20)	18 (3.76)	54 (1.32)
d	9 (3.78)	27 (2.10)	62 (0.88)	11 (5.30)	18 (3.74)	55 (1.26)
MF:						
p	15 (4.02)	41 (1.82)	62 (0.92)	3 (5.68)	17 (3.64)	28 (2.40)
d	13 (4.42)	34 (2.14)	55 (1.14)	3 (5.72)	13 (3.80)	26 (2.48)
DW:						
p	16 (3.78)	46 (1.54)	63 (0.82)	4 (5.42)	18 (3.28)	41 (1.72)
d	15 (4.22)	40 (1.96)	58 (0.98)	5 (5.50)	18 (3.48)	35 (1.82)
LI:						
p	3 (4.26)	37 (2.00)	53 (1.18)	15 (4.48)	28 (2.98)	70 (0.90)
d	3 (4.84)	25 (2.60)	39 (1.66)	12 (4.72)	27 (3.06)	66 (1.02)
ST:						
p	10 (3.56)	44 (1.62)	68 (0.76)	13 (4.00)	36 (2.34)	74 (0.62)
d	6 (4.06)	40 (1.82)	56 (1.04)	10 (4.32)	34 (2.34)	71 (0.72)
NJ:						
p	11 (3.70)	44 (1.68)	67 (0.80)	13 (4.46)	34 (2.38)	75 (0.62)
d	5 (4.24)	38 (2.00)	57 (1.06)	14 (4.44)	32 (2.42)	73 (0.72)

Note.—Notations are as in table 3.
[a] As shown in fig. 6.

values they show more or less the same performance. Data on *p* again give a better result for the five methods (except for UPGMA) than do those on *d*. In the case of model tree (B), the MF method shows a poorer performance than UPGMA, which now gives results similar to the DW method. However, the P_c values for the LI, ST, and NJ methods are substantially higher than those for UPGMA and the DW methods.

Although the above computer simulations were done for a limited number of cases, the results obtained may be summarized as follows: (1) The efficiency of the NJ method in recovering the true unrooted tree is virtually the same as that of the ST method. (2) The NJ and ST methods perform well for both model tree (A) and model tree (B), whereas the DW and MF methods are good only for tree (A) and the LI method is good only for tree (B). For both model trees, UPGMA is rather poor in recovering the true unrooted tree. (3) In the case of model tree (A), data on *p* tend to give slightly better results than those on *d*, except for UPGMA. For model tree (B), however, both *p* and *d* give similar results.

Conclusion (3) above indicates that data on *p* are better than those on *d* for constructing a topology, particularly when the OTUs used form a topology similar to model tree (A). However, since *p* is not a linear function of nucleotide substitutions, it does not provide good estimates of branch lengths unless the *p* values are very small. It is therefore advised that once a topology is obtained by using data on *p*, branch lengths should be estimated by using data on *d*.

Tateno et al. (1982) and Sourdis and Krimbas (1987) conducted similar computer-simulation studies, comparing the efficiency of the UPGMA and the DW and MF methods as well as Fitch and Margoliash's (1967) method for model tree (A). Although

the parameter values used in their simulations are different from ours, their conclusions with respect to unrooted trees are more or less the same as ours.

Varying Rate of Nucleotide Substitution

When the rate of nucleotide substitution varies from evolutionary lineage to evolutionary lineage, the probability of obtaining the correct tree is expected to be lower than that for the case of rate constancy. To see the effect of this factor on P_c, we conducted another computer simulation.

In this simulation, we used the two model trees ([A] and [B]) given in figure 7. The topologies of trees (A) and (B) in fig. 7 are identical, respectively, with those of trees (A) and (B) in figure 6. The value given for each branch of these trees is the *expected* branch length (the *expected* number of nucleotide substitutions per site). The expected branch lengths for tree (A) in figure 7 were obtained under the assumption that b in figure 6(A) varies according to the gamma distribution with mean 0.04 and variance 0.08 (see Tateno et al. 1982 for the justification of this procedure). Similarly, the expected branch lengths for tree (B) in figure 7 were obtained under the assumption that c in figure 6(B) varies according to the gamma distribution with mean 0.07 and variance 0.14. The value of a and the expectation of $\bar{D}$ over all branches were 0.01 and 0.016, respectively. Therefore, the simulations for model trees (A) and (B) correspond, respectively, to those for trees (A) and (B) in table 3. Once the expected length of a particular branch was determined, the actual number of nucleotide substitutions for that branch was obtained by using the Poisson distribution. The eight nucleotide sequences thus obtained were used for the construction of phylogenetic trees. This process was repeated 100 times. In this simulation, only the case of 600 nucleotides was examined, and the trees were constructed by using the p values.

The results of this simulation are presented in table 6. One striking feature in this simulation is that the performance of UPGMA was very poor and that in none of the 100 replications was the correct tree obtained for both model tree (A) and model tree (B). This is in sharp contrast to the case of rate constancy (table 3), in which the P_c for UPGMA is 36% when $n = 600$. The effect of varying rate on the P_c value is less noticeable for the other tree-making methods. The P_c values for the LI method are somewhat lower than those for the case of constant rate (see tables 3 and 6). In the remaining four methods, the P_c values are virtually the same for both cases of constant

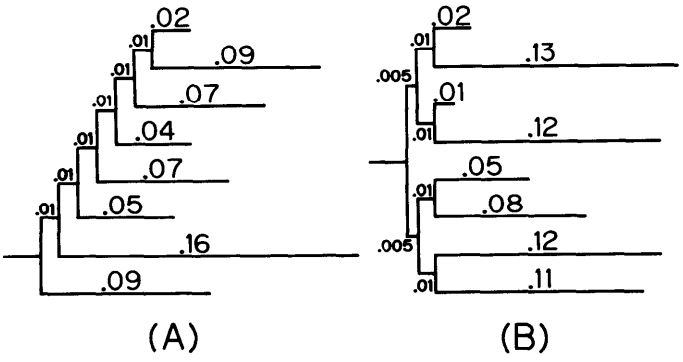

FIG. 7.—Model trees (A) and (B) under the assumption of varying rate of nucleotide substitution

Table 6
P_c and d_T (in parentheses) for Six Tree-making Methods for the Case of Varying Rate of Nucleotide Substitution

Method	Model Tree A[a]	Model Tree B[a]
UPGMA: p ...	0 (8.06)	0 (9.74)
MF: p	77 (0.50)	57 (1.46)
DW: p	69 (0.72)	59 (1.26)
LI: p	46 (1.30)	45 (1.68)
ST: p	77 (0.50)	69 (0.82)
NJ: p	75 (0.56)	72 (0.78)

NOTE.—Notations are as in table 3.
[a] As shown in fig. 7.

and varying rates of nucleotide substitution. Therefore, the conclusions obtained for the case of constant rate also apply to the case of varying rate as far as the NJ, ST, MF, and DW methods are concerned.

Discussion

Unlike the standard algorithm for minimum-evolution trees, the NJ method minimizes the sum of branch lengths at each stage of clustering of OTUs starting with a starlike tree. Therefore, the final tree produced may not be the minimum-evolution tree among all possible trees. However, it should be noted that the real minimum-evolution tree is not necessarily the true tree. Saitou and Nei (1986) have shown that the minimum-evolution or maximum-parsimony tree often has an erroneous topology and that the maximum-parsimony method of tree making is not always the best in recovering the true topology. It seems to us that the relative efficiencies of different tree-making methods should eventually be evaluated by computer simulation. Our computer simulation has shown that the NJ method is quite efficient compared with other tree-making methods that produce a single parsimonious tree.

We have shown that the estimates of branch lengths of the tree obtained by the NJ method are least-squares estimates determined at each stage of clustering of OTUs. This does not mean that these estimates are identical with those that are obtainable by the least-squares method for all branches of the final tree topology. Nevertheless, this property gives some assurance about the reliability of the estimates of branch lengths. Particularly when the number of OTUs is four or less, the branch lengths are exactly least-squares estimates, as is clear from equation (A4) below.

Our procedure of estimating branch lengths is essentially the same as that of Fitch and Margoliash (1967). Some estimates of branch lengths may therefore become negative. If one is reluctant to accept negative estimates, there are two ways to eliminate them. One is to impose the condition that all branches be positive and then to reestimate the branch lengths. The other is to assume that negative estimates are due to sampling error and that the real values are zero rather than negative. Under this assumption, one may simply convert all negative estimates to zero. The second method is justified if we note that the absolute values of negative estimates are usually very small.

A computer program for constructing a tree by using the NJ method is available from the authors on request.

Acknowledgments

We thank Clay Stephens for his comments and John Sourdis for his help in computer simulation. This study was supported by research grants from the National Institutes of Health and the National Science Foundation.

APPENDIX A

Least-Squares Estimation of the Branch Lengths

Let us consider the tree of figure 2(b). If we use matrix notation, the problem is to obtain the least-squares solution of the linear equation $\mathbf{Ax} = \mathbf{d}$, where $\mathbf{x}$ is a column vector of $N + 1$ branch lengths ($\mathbf{x}^t = [L_{1X}, L_{2X}, L_{3Y}, L_{4Y}, \ldots, L_{NY}, L_{XY}]$), $\mathbf{d}$ is a column vector of $N(N-1)/2$ pairwise distances ($\mathbf{d}^t = [D_{12}, D_{13}, D_{14}, D_{15}, \ldots D_{1N}, D_{23}, D_{24}, \ldots D_{2N}, \ldots D_{(N-1)N}]$), and $\mathbf{A}$ is an $[N(N-1)/2] \times (N+1)$ matrix. The element of the ith row and the jth column of matrix $\mathbf{A}$ is given by

$$a_{ij} = \begin{cases} 1 & \text{if the } i\text{th distance includes the } j\text{th branch} \\ 0 & \text{otherwise} \end{cases}.$$

An example of $\mathbf{A}$ for $N = 5$ is shown below:

$$\mathbf{A} = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \end{bmatrix}.$$

The least-squares solution of the equation $\mathbf{Ax} = \mathbf{d}$ is given by solving the equation $\mathbf{A}^t\mathbf{Ax} = \mathbf{A}^t\mathbf{d}$. It becomes $\mathbf{x}_L = \mathbf{B}^{-1}\mathbf{A}^t\mathbf{d}$, where $\mathbf{B} = \mathbf{A}^t\mathbf{A}$. The general expressions of symmetric matrices $\mathbf{B}$ and $\mathbf{B}^{-1}$ are

$$\mathbf{B} = \begin{bmatrix} N-1 & 1 & 1 & \cdots & 1 & N-2 \\ 1 & N-1 & 1 & \cdots & 1 & N-2 \\ 1 & 1 & N-1 & \cdots & 1 & 2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & 1 & 1 & \cdots & N-1 & 2 \\ N-2 & N-2 & 2 & \cdots & 2 & 2(N-2) \end{bmatrix}, \quad \text{(A1)}$$

$$\mathbf{B}^{-1} = \begin{bmatrix} a & b & 0 & 0 & 0 & \cdots & 0 & e \\ b & a & 0 & 0 & 0 & \cdots & 0 & e \\ 0 & 0 & c & d & d & \cdots & d & f \\ 0 & 0 & d & c & d & \cdots & d & f \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & d & d & d & \cdots & c & f \\ e & e & f & f & f & \cdots & f & g \end{bmatrix}, \quad \text{(A2)}$$

where $a = N/4(N-2)$, $b = (N-4)/4(N-2)$, $c = (2N^2 - 11N + 16)/2(N-2)^2(N-3)$, $d = -(N-4)/2(N-2)^2(N-3)$, $e = -1/4$, $f = -1/2(N-2)(N-3)$, and $g = (N-1)/4(N-3)$. Therefore, $\mathbf{x}_L$ becomes

$$L_{1X} = \frac{1}{2}D_{12} + \frac{1}{2(N-2)}(P-Q), \tag{A3a}$$

$$L_{2X} = \frac{1}{2}D_{12} + \frac{1}{2(N-2)}(Q-P), \tag{A3b}$$

$$L_{iY} = \frac{1}{N-2}U_i - \frac{1}{(N-2)^2}(P+Q) - \frac{N-4}{(N-2)^2(N-3)}V, \qquad (3 \le i \le N) \tag{A3c}$$

$$L_{XY} = \frac{1}{2(N-2)}(P+Q) - \frac{1}{2}D_{12} - \frac{1}{(N-2)(N-3)}V, \tag{A3d}$$

where $P = \sum_{j=3}^{N} D_{1j}$, $Q = \sum_{j=3}^{N} D_{2j}$, $U_i = \sum_{j\neq i}^{N} D_{ij}$ $(i \ge 3)$, and $V = \sum_{3 \le j < k} D_{jk}$. Note that equations (A3a) and (A3b) are equivalent to equations (6a) and (6b), respectively. Thus, the sum of branch lengths (S_{12}) for the topology in which OTUs 1 and 2 are clustered becomes

$$S_{12} = L_{1X} + L_{2X} + \sum_{i=3}^{N} L_{iY} + L_{XY} = \frac{1}{2(N-2)}(P+Q) + \frac{1}{2}D_{12} + \frac{1}{N-2}V. \tag{A4}$$

Equation (A4) is equivalent to equation (4).

APPENDIX B

Branch Lengths for a Purely Additive Tree

Let us consider the tree given in figure 1. If the tree is purely additive, $D_{12} = L_{1A} + L_{2A}$ and $D_{1j} - D_{2j} = L_{1A} - L_{2A}$ $(3 \le j \le N)$. Substituting these equations into equation (6a), we have

$$L_{1X} = \frac{1}{2}(L_{1A} + L_{2A}) + \frac{1}{2(N-2)}[(N-2)(L_{1A} - L_{2A})] = L_{1A}. \tag{A5}$$

The estimated branch length (L_{1X}) is identical with the true one (L_{1A}). The same thing can be proven for L_{2X}. Therefore, the node X is identical with the node A in the tree in figure 1.

If OTUs 1 and 2 are neighbors, they are combined into a single OTU, (1-2). Suppose that OTUs (1-2) and 3 are a new pair of neighbors. The estimates of branch lengths for AB and $3B$ can then be obtained correctly, as shown below. Since the tree is purely additive,

$$D_{(1\text{-}2)3} = (D_{13} + D_{23})/2 = [(L_{1A} + L_{A3}) + (L_{2A} + L_{A3})]/2 = D_{12}/2 + (D_{13} - L_{1A}) \tag{A6a}$$

and

$$D_{(1\text{-}2)j} - D_{3j} = D_{12}/2 + (D_{1j} - L_{1A}) - (L_{3B} + L_{Bj}) = D_{12}/2 + L_{AB} - L_{3B} \quad (j \ge 4). \tag{A6b}$$

Substituting these into equation (6a), we have

$$L_{(1\text{-}2)X} = \frac{1}{2}D_{(1\text{-}2)3} + \frac{1}{2(N-3)}\sum_{j=4}^{N}[D_{(1\text{-}2)j} - D_{3j}] = \frac{1}{4}D_{12} + \frac{1}{2}(D_{13} - L_{1A}) + \frac{1}{2(N-3)}[(N-3)(D_{12}/2 + L_{AB} - L_{3B})] = \frac{1}{2}D_{12} + L_{AB}. \quad \text{(A7)}$$

Since $L_{AX} = L_{(1\text{-}2)X} - D_{12}/2$, $L_{AX} = L_{AB}$. On the other hand, as before, it easily can be shown that $L_{3X} = L_{3B}$. Therefore, $X \equiv B$.

The above argument can be applied to any situation if the additivity of branch lengths is maintained.

APPENDIX C

The Smallest S_{ij} Gives the True Neighbors

In the following, we show that for a purely additive tree OTUs 1 and 2 are true neighbors when S_{12} is smallest among all S_{ij}'s. We first show this for the case of four OTUs and then use the principle of induction to prove that it is generally true.

Using the results presented in the Criterion for the Minimum-Evolution Tree section, we can state that the condition for S_{12} to be smallest among the six S_{ij}'s for four OTUs is

$$D_{12} + D_{34} < D_{13} + D_{24}$$

and (A8)

$$D_{12} + D_{34} < D_{14} + D_{23}.$$

Our task is to show that if S_{12} is smallest, OTUs 1 and 2 are true neighbors. In the case of four OTUs, OTUs 3 and 4 are also neighbors if OTUs 1 and 2 are neighbors (see fig. 4). We prove our assertion by showing that when S_{12} is smallest, only OTUs 1 and 2 (and OTUs 3 and 4) are neighbors. To prove this, we first assume that OTUs 1 and 3 (and 2 and 4) are neighbors. We then should have

$$D_{13} + D_{24} = (b_1 + b_3) + (b_2 + b_4),$$

$$D_{12} + D_{34} = (b_1 + b_2 + a) + (b_3 + b_4 + a),$$

from formula (7), in which b_i is the branch length between the ith OTU and its nearest interior node and a is the length between two interior nodes. Since $a > 0$, $(D_{13} + D_{24})$ should be smaller than $(D_{12} + D_{34})$. However, this contradicts formula (A8). Therefore, OTUs 1 and 3 cannot be neighbors. Similarly, it can be shown that OTUs 1 and 4 are not neighbors. Therefore, only OTUs 1 and 2 (and OTUs 3 and 4) are the neighbors.

For the cases of more than four OTUs, we use the induction principle. Assuming that OTUs 1 and 2 are true neighbors when S_{12} is smallest among all S_{ij}'s for the case of $N - 1$ OTUs, we prove that the same rule applies in the case of N OTUs.

Suppose that S_{12} is smallest among all S_{ij}'s when there are N OTUs. If we ignore the Nth OTU, OTUs 1 and 2 are, by assumption, neighbors for the remaining $N - 1$ OTUs. Therefore, there are three possible pairs of neighbors when the Nth OTU is added: OTUs 1 and 2, OTUs 1 and N, and OTUs 2 and N. From equation (9), we have

$$S_{1N} - S_{12} = \sum_{k=3}^{N-1}[(D_{1N} + D_{2k}) - (D_{12} + D_{Nk})]/[2(N-2)].$$

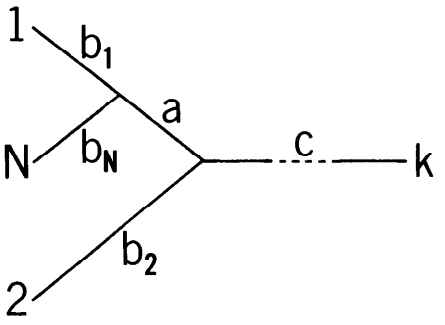

FIG. A1.—A possible relationship for four OTUs (1, 2, N, and k). a, b_1, b_2, and c are branch lengths

If OTUs 1 and N are neighbors, $D_{1N} = b_1 + b_N$, $D_{2k} = b_2 + c$, $D_{12} = b_1 + b_2 + a$, and $D_{Nk} = b_N + a + c$ (see fig. A1). Thus, $(D_{1N} + D_{2k}) - (D_{12} + D_{Nk}) = -2a$ irrespective of k, and $S_{1N} - S_{12}$ should be negative. This is contradictory to our assumption that S_{12} is smallest. Therefore, OTUs 1 and N are not neighbors. Similarly, it can be shown that OTUs 2 and N are not neighbors—and thus that OTUs 1 and 2 should be the neighbors. Since we know that our assertion is true for $N = 4$, it is true for any N (≥ 4).

LITERATURE CITED

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

CASE, S. M. 1978. Biochemical systematics of members of the genus *Rana* native to western North America. Syst. Zool. **27**:299–311.

FAITH, D. P. 1985. Distance methods and the approximation of most-parsimonious trees. Syst. Zool. **34**:312–325.

FARRIS, J. S. 1972. Estimating phylogenetic trees from distance matrices. Am. Nat. **106**:645–668.

———. 1977. On the phenetic approach to vertebrate classification. Pp. 823–850 *in* M. K. HECHT, P. C. GOODY, and B. H. HECHT, eds. Major patterns in vertebrate evolution. Plenum, New York.

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

FITCH, W. M., and E. MARGOLIASH. 1967. Construction of phylogenetic trees. Science **155**: 279–284.

HARTIGAN, J. A. 1973. Minimum mutation fits to a given tree. Biometrics **29**:53–65.

JUKES, T. H., and C. R. CANTOR. 1969. Evolution of protein molecules. Pp. 21–132 *in* H. N. MUNRO, ed. Mammalian protein metabolism. Vol **3**. Academic Press, New York.

KLOTZ, L. C., and R. L. BLANKEN. 1981. A practical method for calculating evolutionary trees from sequence data. J. Theor. Biol. **91**:261–272.

LI, W.-H. 1981. Simple method for constructing phylogenetic trees from distance matrices. Proc. Natl. Acad. Sci. USA **78**:1085–1089.

NEI, M. 1987. Molecular evolutionary genetics. Columbia University Press, New York.

ROBINSON, D. F., and L. R. FOULDS. 1981. Comparison of phylogenetic trees. Math. Biosci. **53**:131–147.

SAITOU, N. 1986. Theoretical studies on the methods of reconstructing phylogenetic trees from DNA sequence data. Ph.D. diss. The University of Texas Health Science Center, Houston.

SAITOU, N., and M. NEI. 1986. The number of nucleotides required to determine the branching

order of three species with special reference to the human-chimpanzee-gorilla divergence. J. Mol. Evol. **24**:189–204.

SATTATH, S., and A. TVERSKY. 1977. Additive similarity trees. Psychometrika **42**:319–345.

SOKAL, R. R., and P. H. A. SNEATH. 1963. Principles of numerical taxonomy. W. H. Freeman, San Francisco.

SOURDIS, J., and C. KRIMBAS. 1987. Accuracy of phylogenetic trees estimated from DNA sequence data. Mol. Biol. Evol. **4**:159–168.

TATENO, Y., M. NEI, and F. TAJIMA. 1982. Accuracy of estimated phylogenetic trees from molecular data. I. Distantly related species. J. Mol. Evol. **18**:387–404.

WALTER M. FITCH, reviewing editor

Received August 5, 1986; revision received February 18, 1987.