6

Phylogenetic Inference: Distance Methods

In **distance methods** or **distance matrix methods,** evolutionary distances are computed for all pairs of taxa, and a phylogenetic tree is constructed by considering the relationships among these distance values. There are many different methods of constructing trees from distance data. Here we discuss only the methods that have proved to be useful for actual data analysis.

6.1. UPGMA

The simplest method in this category is the **unweighted pair-group method using arithmetic averages** (**UPGMA**). This method is often attributed to Sokal and Michener (1958), but the method used by these authors is quite different from the currently used version. Its clear-cut algorithm appears in Sneath and Sokal's (1973) book. A tree constructed by this method is sometimes called a **phenogram,** because it was originally used to represent the extent of phenotypic similarity for a group of species in numerical taxonomy. However, it can be used for constructing molecular phylogenies when the rate of gene substitution is more or less constant. Particularly when gene frequency data are used for phylogenetic reconstruction, this model produces reasonably good trees compared with other distance methods (Nei et al. 1983; Takezaki and Nei 1996). In this case, a distance measure that has a smaller coefficient of variation seems to give better trees than other distance measures even if it is not strictly proportional to the number of gene substitutions (Takezaki and Nei 1996). UPGMA is intended to reconstruct a species tree, although topological errors often occur when the rate of gene substitution is not constant or when the number of genes or nucleotides used is small.

Algorithm

In UPGMA, a certain measure of evolutionary distance is computed for all pairs of taxa or sequences, and the distance values are presented in the following matrix form.

Taxon	1	2	3	4
2	d_{12}			
3	d_{13}	d_{23}		
4	d_{14}	d_{24}	d_{34}	
5	d_{15}	d_{25}	d_{35}	d_{45}

Here, d_{ij} stands for the distance between the i-th and j-th taxa. Clustering of taxa starts with a pair of two taxa with the smallest distance. Suppose that d_{12} is smallest among all distance values in the above matrix. Taxa 1 and 2 are then clustered with a branch point located at distance $b = d_{12}/2$. Here, we have assumed that the lengths of the branches leading from this branch point to taxa 1 and 2 are the same (see Example 6.1). Taxa 1 and 2 are then combined into a single composite taxon or cluster [u = (1-2)], and the distance between this u and another taxon $k (k \neq 1, 2)$ is computed by $d_{uk} = (d_{1k} + d_{2k})/2$. Therefore, we have the following new matrix.

Taxon	u = (1-2)	3	4
3	d_{u3}		
4	d_{u4}	d_{34}	
5	d_{u5}	d_{35}	d_{45}

Now suppose that distance d_{u3} is smallest. Then, taxa u and 3 are combined into a new composite taxon or cluster [v = (1-2-3)] with a branch point of $b = d_{u3}/2 = (d_{13} + d_{23})/(2 \times 2)$. The distance between the newly created cluster v and each of the remaining taxa (k's) is now computed by $d_{kv} = (d_{k1} + d_{k2} + d_{k3})/3$. We then have

Taxon	v = (1-2-3)	4
4	d_{v4}	
5	d_{v5}	d_{45}

Let us assume that d_{v4} is smallest in the above distance matrix. We then combine v = (1-2-3) and 4 with a branch point of $b = d_{v4}/2 = (d_{14} + d_{24} + d_{34})/(3 \times 2)$. It is obvious that the last taxon to join the tree is 5, and the branch point is given by $b = (d_{15} + d_{25} + d_{35} + d_{45})/(4 \times 2)$.

It is of course possible that in the second matrix the smallest distance is d_{45} (or any other one) instead of d_{u3}. In this case, taxa 4 and 5 are joined with the branch point of $b = d_{45}/2$, and a new composite taxon, v = (4-5), will be created. The distances between v and other taxa (3 and u are given by $d_{3v} = (d_{34} + d_{35})/2$ and $d_{uv} = (d_{14} + d_{15} + d_{24} + d_{25})/4$. Now suppose that d_{uv} is smallest. Then, taxa u and v are clustered, and taxon 3 will be the last to join the cluster. Of course, if d_{3v} is smallest, taxa 3 and v cluster first.

As is obvious from the above example, the distance between two clusters (A and B) is given by the following formula.

$$d_{AB} = \sum_{ij} d_{ij}/(rs) \qquad (6.1)$$

where r and s are the numbers of taxa in clusters A and B, respectively, and d_{ij} is the distance between taxon i in cluster A and taxon j in cluster B. The branch point between the two clusters is given by $d_{AB}/2$. For the purpose of computer programming, however, the above equation is not convenient, and other faster algorithms are used to compute d_{AB} (e.g., Swofford et al. 1996).

Statistical Tests of UPGMA Trees

Rooted and Unrooted UPGMA Trees

A tree obtained by UPGMA is usually presented as a rooted tree, because it is easy to infer the root of the tree under the assumption of a constant rate of evolution. However, UPGMA is a method of inferring both the topology and branch lengths similar to other methods, and we do not have to give the root to a UPGMA tree. In other methods of phylogenetic inference, an uprooted tree is usually constructed, because it is difficult to determine the root when the evolutionary rate varies from branch to branch. We can use the same approach and construct an unrooted UPGMA tree, disregarding the root usually given to a UPGMA tree. When we compare an UPGMA tree with trees constructed by other methods, we should use this unrooted UPGMA tree, because rooting can introduce an additional source of errors in tree building. Unrooted trees are also useful for testing the reliability of the tree obtained by using the bootstrap or other method, as will be discussed below.

Reliability of UPGMA Trees

Since a phylogenetic tree is usually constructed from a limited amount of data, it is important to examine the reliability of the tree obtained. As will be discussed in detail in chapter 9, there are two major methods of testing the reliability of the topology of a tree obtained by distance methods. In the case of UPGMA trees, we can use Nei et al.'s (1985) **interior branch test** or Felsenstein's (1985) **bootstrap test.** Both tests examine the reliability of each interior branch of a tree. If every interior branch length is proved to be positive, the tree is regarded as reliable from the statistical point of view. However, Nei et al.'s test becomes complicated when the number of taxa examined is large. A simpler way of testing the positiveness of an interior branch is to use the bootstrap test considering unrooted UPGMA trees (see chapter 9 for details). In this test, it is customary to compute a quantity equivalent to the probability of confidence (1 − Type I error) rather than the significance level. This value is called the **bootstrap confidence value** (P_B) or **bootstrap value.** If this value is higher than 95% (or 99% depending on the confidence level one wishes to have), the interior branch is considered to be statistically significant (Felsenstein 1985; Efron et al. 1996). In this book, we will use this bootstrap technique extensively.

It is known that when closely related DNA (or protein) sequences are used for constructing UPGMA trees, two or more trees (**tie trees**) may be

produced from the same distance (Kim et al. 1993; Backeljau et al. 1996; Takezaki 1998). These tie trees occur because two or more distance values in a distance matrix occasionally become identical. It is possible to enumerate all these tie trees (Rohlf 1993), but this enumeration is not very meaningful, since these tie trees are primarily caused by sampling errors of distance estimates and they are close to one another. A better way of treating this problem is to construct a **bootstrap consensus tree** (see section 9.3). This consensus tree also has a bootstrap value for each interior branch, and it can be treated in the same way as the above UPGMA tree with bootstrap values. Therefore, we will know the reliability of each branching pattern of the UPGMA tree. When only one UPGMA tree exists for a given data set, the bootstrap consensus tree is usually identical with the original UPGMA tree (Takezaki 1998).

Example 6.1. UPGMA Tree of Hominoid Species

Figure 6.1 shows the nucleotide sequences of a segment (896 nucleotides) of mitochondrial DNA (mtDNA) from humans, chimpanzees, gorillas, orangutans, and gibbons (Brown et al. 1982). In this data set, transitional nucleotide differences are considerably greater than transversional differences. The average transition/transversion ratio (R) obtained by Equation (3.18) is about 6.2. We therefore estimated the number of nucleotide substitutions (d) per site using Equation (3.12) (Kimura distance). The results obtained are presented in Table 6.1. (One site containing an alignment gap was removed.) It is seen that the value between humans and chimpanzees ($\hat{d} = 0.095$) is smallest, so that humans and chimpanzees are the first to be clustered with a branch point at $b_{HC} = 0.095/2 = 0.048$ (Figure 6.2A). Humans and chimpanzees are now combined into a single taxon, (HC). The $\hat{d}$ values between this taxon and gorillas, orangutans, and gibbons become $(0.113 + 0.118)/2 = 0.115$, $(0.183 + 0.201)/2 = 0.192$, and $(0.212 + 0.225)/2 = 0.218$, respectively. The other distance values remain unchanged. The smallest d value in the new d matrix is that (0.115) between (HC) and gorillas. Thus, gorillas join (HC) with a branch point at $b_{G(HC)} = 0.115/2 = 0.058$. If this type of computation is repeated, we finally obtain the phylogenetic tree given in Figure 6.2A.

The tree given in this figure is an unrooted tree, and there are two interior branches. The bootstrap values for the interior branches are written in boldface. In the present case, the bootstrap consensus UPGMA tree is virtually identical with the tree in Figure 6.2A. The branch separating the group of humans, chimpanzees, and gorillas from the two other species shows a bootstrap value of 100%, whereas the branch separating humans and chimpanzees from gorillas has a value of 90%. Therefore, this data set establishes the cluster of humans, chimpanzees, and gorillas but is not sufficient to resolve the branching pattern among these three species at the 95% confidence level. In this data set, application of the statistical test of rate constancy given in chapter 10 does not reject the hypothesis of a molecular clock. Therefore, the use of UPGMA for inferring species trees is justified.

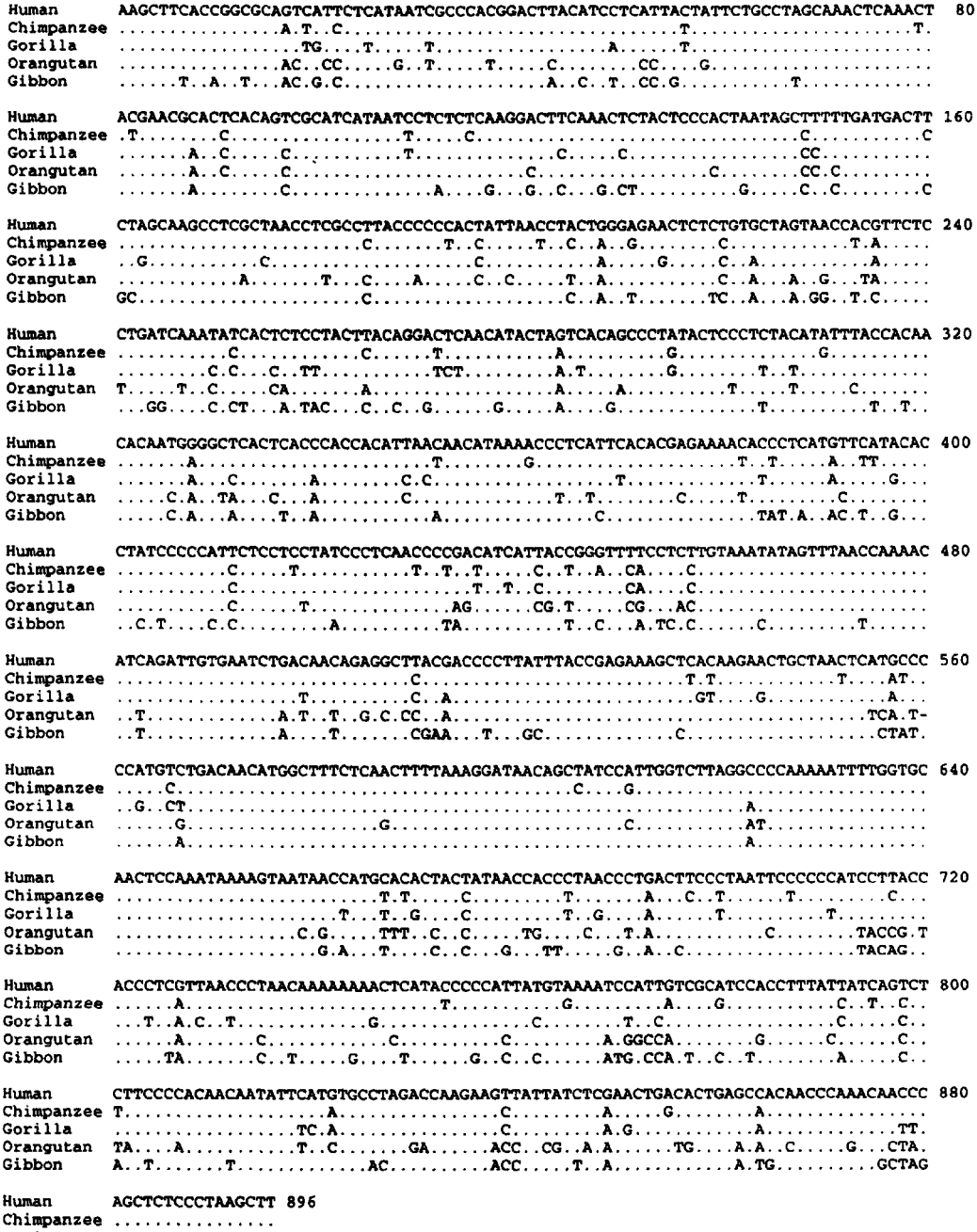

FIGURE 6.1. Sequences of an 896 bp fragment of primate mitochondrial DNAs. The orangutan sequence has one deletion at position 560. Data from the GenBank.

Table 6.1 Kimura distances for the data shown in Fig. 6.1.

	Human	Chimpanzee	Gorilla	Orangutan
Chimpanzee	0.095 ± 0.011			
Gorilla	0.113 ± 0.012	0.118 ± 0.013		
Orangutan	0.183 ± 0.016	0.201 ± 0.018	0.195 ± 0.017	
Gibbon	0.212 ± 0.018	0.225 ± 0.019	0.225 ± 0.019	0.222 ± 0.018

Note: One site containing an alignment gap was removed from the analysis.

6.2. Least Squares (LS) Methods

When the rate of nucleotide substitution varies from evolutionary lineage to lineage, UPGMA often gives an incorrect topology. In this case, we should use methods that allow different rates of nucleotide substitution for different branches. One group of such methods is least squares (LS) methods. There are several different LS methods, but the most commonly used ones are the ordinary LS and the weighted LS methods.

Topology Construction

In the **ordinary LS method** of phylogenetic inference (Cavalli-Sforza and Edwards 1967), we consider the following residual sum of squares

$$R_S = \sum_{i<j} (d_{ij} - e_{ij})^2 \qquad (6.2)$$

where d_{ij} and e_{ij} are the **observed** and **patristic distances** between taxa i and j, respectively. The patristic distance between taxa i and j is the sum of estimates of the lengths of all branches connecting the two taxa in a tree. For example, the patristic distance between humans and gorillas in

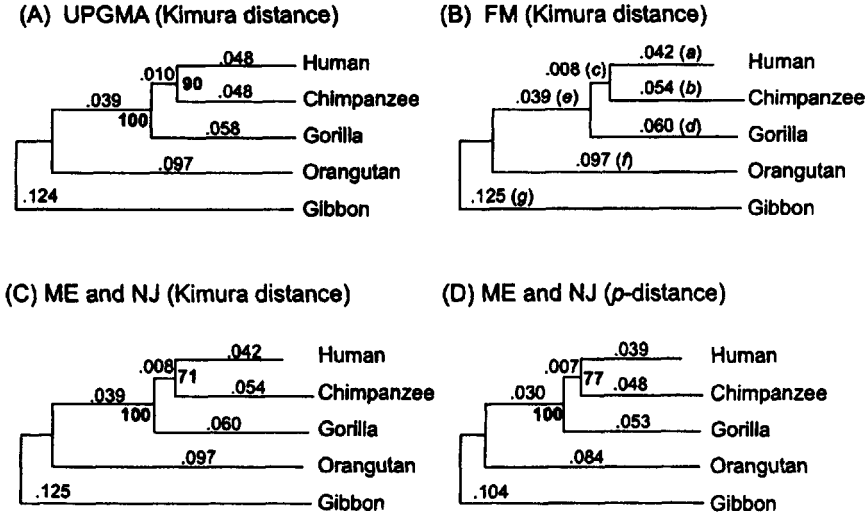

FIGURE 6.2. Evolutionary trees inferred by different distance methods. The UPGMA tree is unrooted. Bootstrap values are in boldface.

the tree of Figure 6.2B is $a + c + d = 0.110$. In the standard LS method, R_S is computed for all plausible topologies, and the topology with the smallest R_S value is chosen as the final tree.

Fitch and Margoliash (1967) used the following R_S value for choosing the final topology.

$$\sum_{i<j}[(d_{ij} - e_{ij})^2 / d_{ij}] \tag{6.3}$$

This procedure is called a **weighted LS method.** In practice, the R_S values defined in Equations (6.2) and (6.3) usually give the same topology or very similar topologies.

Theoretically, a better procedure would be to use the **generalized LS method** of computing R_S, in which both the variance and covariance of d_{ij}'s are taken into account (Cavalli-Sforza and Edwards 1967; Bulmer 1991). However, this method is very time consuming. Furthermore, when the d_{ij} values approach 0, the variance-covariance matrix becomes singular (Rzhetsky and Nei 1992b), and thus this method does not seem to give reliable phylogenetic trees.

Least-Squares Method with the Constraint of Nonnegative Branches

Using computer simulation, Saitou and Nei (1986) and Rzhetsky and Nei (1992a) studied the probability of obtaining the correct tree topology by the ordinary and weighted least-squares methods and showed that the probability is often lower than that of some other distance methods. Part of the reason seems to be that these methods often give negative estimates of branch lengths, which are theoretically unrealistic. Therefore, one way to improve the efficiency of this method would be to use the LS method with the constraint of nonnegative branches (Felsenstein 1995, 1997). Using computer simulation, Kuhner and Felsenstein (1994) indeed showed that this modified method increases the probability of obtaining the correct topology considerably. Estimation of branch lengths with the constraint of nonnegative values requires iterative computation of branch length estimates (Felsenstein 1995, 1997). It is also known that in the case of four taxa this method gives the same topology as that obtained by the neighbor joining method (see section 6.4) (Gascuel, 1994; M. Bulmer, personal communication, 1991).

Estimation of Branch Lengths

Fitch-Margoliash Method

To compute the residual sum of squares, R_S, we must first estimate the branch lengths and the e_{ij}'s for each topology. A simple way to estimate branch lengths is to use Fitch and Margoliash's method (1967). Although the estimates obtained by this method are not always the same as those obtained by the LS method, the differences are usually very small so that the Fitch-Margoliash method is still used. This method takes advantage

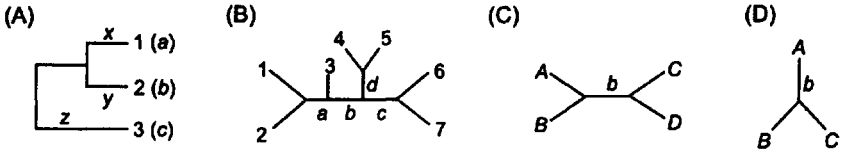

FIGURE 6.3. Estimation of branch lengths.

of the property that when there are only three taxa the estimates of branch lengths for all three taxa can be uniquely determined.

Consider three taxa 1, 2, and 3, of which the evolutionary relationships are given by Figure 6.3A. The evolutionary distances between taxa 1 and 2, 1 and 3, and 2 and 3 are then given by

$$d_{12} = x + y \tag{6.4a}$$

$$d_{13} = x + z \tag{6.4b}$$

$$d_{23} = y + z \tag{6.4c}$$

where x, y, and z are the branch lengths for taxa 1, 2, and 3, respectively. Solving these simultaneous equations gives

$$x = (d_{12} + d_{13} - d_{23})/2 \tag{6.5a}$$

$$y = (d_{12} - d_{13} + d_{23})/2 \tag{6.5b}$$

$$z = (-d_{12} + d_{13} + d_{23})/2 \tag{6.5c}$$

These are LS estimates.

When there are four or more taxa, we first choose the two taxa with the smallest distance and denote them by A and B. All the remaining taxa are combined into a single composite taxon designated by C. The distance between taxa A and B is the same as the original distance (d_{12}), but the distance between taxa A and C is now represented by the simple average of the distances between A and all taxa in C. Similarly, the distance between B and C is the average of the distances between B and all taxa in C. For example, in the distance matrix in Table 6.1, humans and chimpanzees show the smallest distance. Therefore, we denote humans and chimpanzees by A and B, respectively, and the remaining species by C. From the distance estimates given in Table 6.1, we have $d_{AB} = 0.095$, $d_{AC} = (0.113 + 0.183 + 0.212)/3 = 0.169$, and $d_{BC} = (0.118 + 0.201 + 0.225)/3 = 0.181$. The values of x, y, and z therefore become 0.042, 0.054, and 0.124, respectively, from Equations (6.5). Here x and y represent the number of estimated nucleotide substitutions (a and b) for the human and chimpanzee lineages, respectively, and z is the distance between the composite taxon C and the branch point between humans and chimpanzees (Figure 6.2B).

We now combine taxa 1 and 2 and designate the composite taxon as (AB). We then recompute the distances between this composite taxon

(*AB*) and all other taxa and choose the two taxa that show the smallest value among all distances, including those that do not involve (*AB*). These two taxa are again designated by *A* and *B*, whereas *C* represents the composite taxon consisting of all the remaining taxa. The new *x*, *y*, and *z* values are computed by the same procedure. In the case of hominoid data, the distances between (*AB*) and the other taxa (gorillas, orangutans, and gibbons) have already been computed (0.115, 0.192, and 0.218, respectively) when we constructed the UPGMA tree, and the smallest distance in the new matrix is that between (*AB*) and gorillas. Therefore, (*AB*) and gorillas are designated as the new *A* and *B*, respectively, whereas *C* represents orangutans and gibbons. We now have $d_{AB} = 0.115$, $d_{AC} = (0.183 + 0.201 + 0.212 + 0.225)/4 = 0.205$, and $d_{BC} = (0.195 + 0.225)/2 = 0.210$. Therefore, we have $x = 0.055$, $y = 0.060$, and $z = 0.150$ from Equations (6.5). Branch lengths *c* and *d* of the tree in Figure 6.2B are then estimated by using the following relationships.

$$d_{AB} = (a + b)/2 + c + d$$

$$d_{AC} = (a + b)/2 + c + z$$

$$d_{BC} = d + z$$

We know that $(a + b)/2 = 0.048$ and $z = 0.150$. Therefore, we have $c = 0.008$ and $d = 0.060$ (Figure 6.2B). The above procedure is repeated until all branch lengths are estimated. In the case of hominoid data, the estimates of the three remaining branches (*e*, *f*, and *g*) are presented in Figure 6.2B.

We are now in a position to compute e_{ij}'s for all pairs of taxa and then the R_S values in Equations (6.2) and (6.3). The latter values become 0.000047 and 0.002264, respectively. To find the LS tree, however, we must consider all possible or all plausible trees. In practice, the number of topologies is usually very large so that only a small proportion of possible topologies is examined for computing the R_S values. In Fitch-Margoliash's (1967) method, the first topology is constructed by the algorithm described above. Once this topology is obtained, different topologies are examined by various branch-swapping algorithms. These algorithms will be explained in the next chapter, where the algorithms are important in relation to the construction of maximum parsimony trees.

Once the final tree topology is obtained by minimizing R_S, better estimates of branch lengths of the final tree may be obtained by the LS method, which will be described below. Mathematically, the LS estimates are more reliable than those obtained by the Fitch-Margoliash method, but in practice, the differences between them are usually very small when DNA or protein sequences are used.

Least Squares Methods

The standard method of estimating the branch lengths of a tree is to use the LS method. Rzhetsky and Nei (1992a, 1993) developed a fast algorithm for obtaining LS estimates of branch lengths for any given topol-

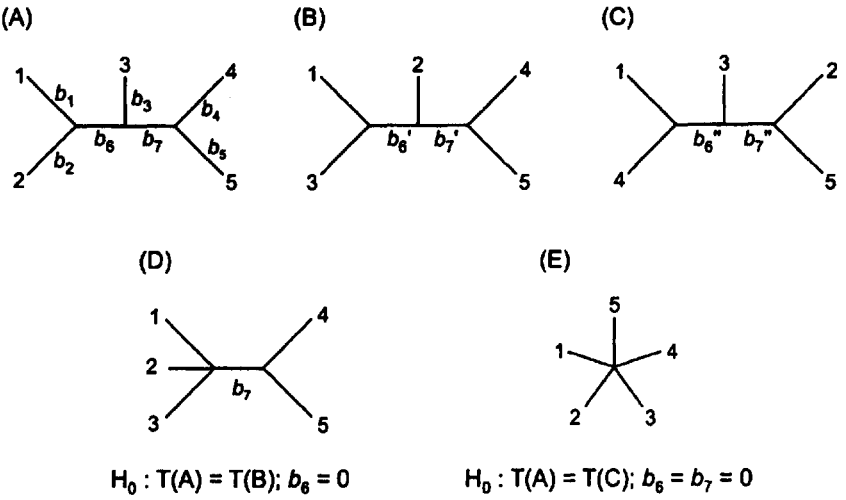

FIGURE 6.4. Three different topologies for five taxa and two "null trees" for testing topological differences.

ogy. Let us consider a hypothetical tree for five sequences given in Figure 6.4A and use the ordinary LS method to estimate the branch lengths denoted by $b_1, b_2, \ldots,$ and b_7. We represent an estimate of evolutionary distance between sequences i and j by d_{ij}. We can then write the d_{ij}'s as follows.

$$
\begin{aligned}
d_{12} &= b_1 + b_2 + \epsilon_{12} \\
d_{13} &= b_1 + b_3 + b_6 + \epsilon_{13} \\
d_{14} &= b_1 + b_4 + b_6 + b_7 + \epsilon_{14} \\
d_{15} &= b_1 + b_5 + b_6 + b_7 + \epsilon_{15} \\
d_{23} &= b_2 + b_3 + b_6 + \epsilon_{23} \\
d_{24} &= b_2 + b_4 + b_6 + b_7 + \epsilon_{24} \\
d_{25} &= b_2 + b_5 + b_6 + b_7 + \epsilon_{25} \\
d_{34} &= b_3 + b_4 + b_7 + \epsilon_{34} \\
d_{35} &= b_3 + b_5 + b_7 + \epsilon_{35} \\
d_{45} &= b_4 + b_5 + \epsilon_{45}
\end{aligned}
$$

where ϵ_{ij}'s are sampling errors. We assume that ϵ_{ij} is distributed with mean 0 and variance $V(d_{ij})$. If we use matrix algebra, the above set of equations may be written as

$$\boldsymbol{d} = \boldsymbol{A}\boldsymbol{b} + \boldsymbol{\epsilon} \qquad (6.6)$$

where $\boldsymbol{d}$, $\boldsymbol{b}$, and $\boldsymbol{\epsilon}$ are column vectors of d_{ij}'s, b_i's, and ϵ_{ij}'s, respectively; that is, $\boldsymbol{d}^t = (d_{12}, d_{13}, \ldots, d_{45})$, $\boldsymbol{b}^t = (b_1, b_2, \ldots, b_7)$, and $\boldsymbol{\epsilon}^t = (\epsilon_{12}, \epsilon_{13}, \ldots, \epsilon_{45})$. Here t indicates the transpose of a vector or a matarix. Note that vectors $\boldsymbol{d}$ and $\boldsymbol{\epsilon}$ have $r \equiv m(m-1)/2$ elements and $\boldsymbol{b}$ has $T \equiv 2m - 3$ elements, where m is the number of sequences. $\boldsymbol{A}$ is a matrix representing a topology, and in this case (topology [A] in Figure 6.4) it is given by

$$\mathbf{A} = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 \end{bmatrix} \tag{6.7}$$

An element of this matrix is 1 when there is a corresponding branch and 0 otherwise (see the equations for d_{ij}'s). The LS estimate of $\boldsymbol{b}$ is then given by

$$\hat{\boldsymbol{b}} = (\boldsymbol{A}^t\boldsymbol{A})^{-1}\boldsymbol{A}^t\boldsymbol{d} = \boldsymbol{L}\boldsymbol{d} \tag{6.8}$$

where $\boldsymbol{L} = (\boldsymbol{A}^t\boldsymbol{A})^{-1}\boldsymbol{A}^t$. Obviously, an estimate of the length of the i-th branch is

$$\hat{b}_i = \boldsymbol{L}_i\boldsymbol{d} \tag{6.9}$$

where L_i is the i-th row of the matrix $\boldsymbol{L}$ (Rzhetsky and Nei 1992a). If we use this formula for the case of topology A in Figure 6.4, we obtain

$$\begin{aligned}
\hat{b}_1 &= \frac{1}{2}d_{12} + \frac{1}{6}(d_{13} - d_{23} + d_{14} - d_{24} + d_{15} - d_{25}) \\
\hat{b}_2 &= \frac{1}{2}d_{12} - \frac{1}{6}(d_{13} - d_{23} + d_{14} - d_{24} + d_{15} - d_{25}) \\
\hat{b}_3 &= \frac{1}{4}(d_{13} + d_{23} + d_{34} + d_{35}) - \frac{1}{8}(d_{14} + d_{24} + d_{15} + d_{25}) \\
\hat{b}_4 &= \frac{1}{2}d_{45} + \frac{1}{6}(d_{14} - d_{15} + d_{24} - d_{25} + d_{34} - d_{35}) \\
\hat{b}_5 &= \frac{1}{2}d_{45} - \frac{1}{6}(d_{14} - d_{15} + d_{24} - d_{25} + d_{34} - d_{35}) \\
\hat{b}_6 &= -\frac{1}{2}d_{12} + \frac{1}{4}(d_{13} + d_{23} - d_{34} - d_{35}) + \frac{1}{8}(d_{14} + d_{24} + d_{15} + d_{25}) \\
\hat{b}_7 &= \frac{1}{4}(d_{34} + d_{35} - d_{13} - d_{23}) + \frac{1}{8}(d_{14} + d_{24} + d_{15} + d_{25}) - \frac{1}{2}d_{45}
\end{aligned} \tag{6.10}$$

Similar expressions can be obtained for any other topology such as topology B or C in Figure 6.4 or for any number of sequences (m).

In practice, however, estimation of branch lengths by Equation (6.10) is not always easy, because a large amount of computational time is required when the number of sequences is large. Rzhetsky and Nei (1993) solved this problem by developing a simple method of estimating branch lengths without using matrix algebra. Consider tree B in Figure 6.3 as an example. If we choose one particular interior branch of this tree, this tree can be drawn in the form of tree C, where A, B, C, and D each represent a cluster of sequences. For example, for the interior branch b of tree B in Figure 6.3, A, B, C, and D represent clusters (3), (1, 2), (4, 5), and (6, 7) respectively. In this case, the branch length b in tree C can be estimated by the following equation

$$\hat{b} = \frac{1}{2}\{\gamma[d_{AC} / (m_A m_C) + d_{BD} / (m_B m_D)] + (1 - \gamma)[d_{BC} / (m_B m_C) + d_{AD} / (m_A m_D)] - d_{AB} / (m_A m_B) - d_{CD} / (m_C m_D)\} \quad (6.11)$$

where

$$\gamma = (m_B m_C + m_A m_D) / [(m_A + m_B)(m_C + m_D)]$$

Here, m_A, m_B, m_C, and m_D are the numbers of sequences in clusters A, B, C, and D, respectively, and d_{AC} is the sum of pairwise distances between cluster A (sequence 3) and cluster C (sequences 4 and 5). The distances d_{BD}, d_{BC}, d_{AD}, d_{AB}, *and* d_{CD} are defined in a similar fashion. By contrast, the LS estimate of the length (b) of an exterior branch of tree D in Figure 6.3 is given by

$$\hat{b} = [d_{AB} / m_B + d_{AC} / m_C - d_{BC} / (m_A m_B)]/2 \quad (6.12)$$

where d_{AB} is the sum of all pairwise distances between sequence A (representing one exterior branch) and all sequences belonging to cluster B, d_{AC} is the sum of distances between A and all sequences belonging to cluster C, d_{BC} is the sum of all pairwise distances between sequences in clusters B and C, and m_B and m_C are the numbers of sequences in clusters B and C, respectively.

The above equations simplify the computation of branch length estimates considerably. For example, $\hat{b}_1$ in Equation (6.10) can be obtained by using Equation (6.12). In this case, the tree is given by Figure 6.4A, and the sequences in clusters A, B, and C are 1, 2, and (3, 4, 5), respectively. Therefore, $d_{AB} = d_{12}$, $d_{AC} = d_{13} + d_{14} + d_{15}$, $d_{BC} = d_{23} + d_{24} + d_{25}$, $m_B = 1$, and $m_C = 3$, and we have $\hat{b}_1 = [d_{12} + (d_{13} + d_{14} + d_{15})/3 - (d_{23} + d_{24} + d_{25})/3]/2$, which is identical to $\hat{b}_1$ in Equation (6.10). Similarly, all the other branch length estimates can be obtained by either Equation (6.11) or (6.12). Once $\hat{b}_i$'s are obtained, e_{ij}'s in Equations (6.2) and (6.3) can easily be obtained by summing the $\hat{b}_i$'s for all the branches that connect sequences i and j, and therefore R_S can be computed.

Bryant (1997), Gascuel (1997b), and Bryant and Waddell (1998) recently developed a fast algorithm for computing $\hat{b}$, using Equations (6.11) and (6.12). The readers who are interested in this algorithm should refer to the original papers. This algorithm is used in PAUP* and MEGA2.

Figure 6.2B shows a tree obtained by the Fitch-Margoliash method with LS estimates of branch lengths. The distance used is the Kimura distance. This tree now has a shorter branch for humans than that of the UPGMA tree and a longer branch for chimpanzees, but the other branches are nearly the same as those of the UPGMA tree. In the present case, Fitch and Margoliash's original algorithm gives essentially the same results.

Theoretical Basis

The LS method is a well-established statistical method of parameter estimation. When the variables are normally distributed, it is as efficient as the maximum likelihood method. In the present case, if the number of nucleotides or amino acids examined is sufficiently large, b_i is expected to follow the normal distribution (Rzhetsky and Nei 1993). Therefore, the LS method is expected to give good estimates of branch lengths (b_i's).

However, our primary interest is to determine the topology of the tree, and if this topology is incorrect, branch length estimates do not have much biological meaning. The mathematical formulation presented in this section is not intended to estimate a topology, because there is no parameter specifying topology in the formula for R_S. What is then the theoretical basis of the LS method for "estimating" the correct topology? At this moment, we do not have a good answer to this question. We can simply argue that a topology of which the estimated branch lengths are closest to the observed ones should be a good topology. Indeed, if unbiased estimates of evolutionary distances are used and the number of nucleotides or amino acids used (n) becomes infinitely large, the R_S value will be 0 only for the correct topology. Therefore, if we regard a tree-building method as a statistic as Felsenstein (1978) did, the LS method is a consistent estimator of the true topology. Computer simulations (e.g., Sourdis and Krimbas 1987; Kuhner and Felsenstein 1994) have shown that the LS method with the constraint of nonnegative branches gives reasonably good results for topology construction when the number of nucleotides used is large.

6.3. Minimum Evolution (ME) Method

Principle

In this method, the sum (S) of all branch length estimates, i.e.,

$$S = \sum_{i}^{T} \hat{b}_i \qquad (6.13)$$

is computed for all or all plausible topologies, and the topology that has the smallest S value is chosen as the best tree. Here $\hat{b}_i$ denotes an esti-

mate of the length of the i-th branch, and T is the total number of branches, that is, $2m - 3$. For example, in the case of tree A of Figure 6.4, S is given by $\hat{b}_1 + \hat{b}_2 + \cdots + \hat{b}_7$, where $\hat{b}_i$ indicates an estimate of b_i. The idea of a minimum evolution method was first put forward by Edwards and Cavalli-Sforza (1963) without giving any justification or algorithm. Later, Kidd and Sgaramella-Zonta (1971) suggested that the total branch lengths [$L(S)$] be computed by summing the absolute values ($|\hat{b}_i|$) of all branch length estimates without any theoretical justification. (In the case of allele frequency data with which Kidd and Sgaramella-Zonta were concerned, LS estimates of b_i's often become negative.) Unfortunately, $L(\hat{S})$ does not have a nice statistical property that permits the fast computation of S values, and the statistical tests as developed by Rzhetsky and Nei (1992a, 1993) are not applicable to $L(S)$. Note also that in the presence of statistical errors estimates of short branch lengths may become negative by chance even for the correct topology (Sitnikova et al. 1995).

The theoretical foundation of the ME method is Rzhetsky and Nei's (1993) mathematical proof that when unbiased estimates of evolutionary distances are used, the expected value of S becomes smallest for the true topology irrespective of the number of sequences (m). This is a good statistical property, but a topology with the smallest S is not necessarily an "unbiased estimator" of the true topology (chapter 9).

Like the LS method, the ME method is supposed to examine all possible topologies and find one that has the smallest S value. For this purpose, one may use the algorithms presented in chapter 7. However, this is very time consuming, and for this reason Rzhetsky and Nei (1992a, 1993) suggested that the neighbor joining (NJ) tree (see section 6.4) be first constructed and then a set of topologies close to the NJ tree be examined to find a tree with a smaller S value (temporary ME tree). A new set of topologies close to this temporary ME tree (excluding previously examined topologies) are now examined to find a tree with an even smaller S value. This process will be continued until no tree with a smaller S is found, and the tree with the smallest S is regarded as the ME tree. The theoretical basis of this strategy is that the ME tree is generally identical or close to the NJ tree when m is relatively small (Saitou and Imanishi 1989; Rzhetsky and Nei 1992a), and thus the NJ tree can be used as a starting tree when m is large.

One way of choosing closely related topologies is to consider all topologies that are different from the temporary ME tree by topological distances $d_T = 2$ and 4. If this is repeated many times, avoiding all topologies previously examined, one can usually obtain the ME tree or a tree close to it. We call this procedure the **close neighbor interchange** (CNI) algorithm.

Computation of S *and* D

In a previous section, we have mentioned that the LS estimates of branch lengths are given by a function of distance estimates (d_{ij}'s), that is, $\hat{\boldsymbol{b}} = \boldsymbol{Ld}$. Therefore, S can be expressed as a linear function of d_{ij} or d_i, where d_{ij}'s are renumbered as d_i for $i = 1, 2, \ldots, m(m-1)/2$. That is,

$$S = \boldsymbol{yd} = \sum_{i=1}^{r} y_i d_i \tag{6.14}$$

where $r = m(m-1)/2$ (Rzhetsky and Nei 1992a). The coefficients y_i's are determined solely by the tree topology, and they can be computed if the topology matrix A in Equation (6.6) is defined. For example, in the case of tree A in Figure 6.4, S becomes

$$\begin{aligned} S_A = {} & d_{12}/2 + d_{13}/4 + d_{14}/8 + d_{15}/8 + d_{23}/4 \\ & + d_{24}/8 + d_{25}/8 + d_{34}/4 + d_{35}/4 + d_{45}/2 \end{aligned} \tag{6.15}$$

Similarly, for tree B in Figure 6.4 we have

$$\begin{aligned} S_B = {} & d_{12}/4 + d_{13}/2 + d_{14}/8 + d_{15}/8 + d_{23}/4 \\ & + d_{24}/4 + d_{25}/4 + d_{34}/8 + d_{35}/8 + d_{45}/2 \end{aligned} \tag{6.16}$$

However, we are primarily interested in the difference in S between two topologies. This difference (D) is given by

$$D = S_B - S_A = \sum_{i=1}^{r} (y_{Bi} - y_{Ai}) d_i \tag{6.17}$$

where y_{Ai} and y_{Bi} are the coefficients of the i-th distance in S for topologies A and B, respectively. Therefore, if y_{Ai}'s and y_{Bi}'s are computed for a pair of topologies, D can easily be obtained. For this purpose, it is not necessary to know individual S values. In the case of trees A and B in Figure 6.4, we know y_{Ai}'s and y_{Bi}'s, so that D is given by

$$D = -d_{12}/4 + d_{13}/4 + d_{24}/8 + d_{25}/8 - d_{34}/8 - d_{35}/8 \tag{6.18}$$

In practice, D may be subject to sampling error, and we are interested in testing the null hypothesis that the expected value [E(D)] of D is 0 for a given type of nucleotide or amino acid substitution. If D is significantly greater than 0, we may conclude that tree A is better than tree B. However, what is the biological meaning of this null hypothesis when two different topologies are compared? Actually, this hypothesis is equivalent to the null hypothesis that the lengths of the interior branches that produce different branching patterns (different partitions of sequences) for the two topologies are 0. Only in this case do trees A and B become identical. The tree corresponding to this null hypothesis is called the **null tree.** For example, in the comparison of trees A and B in Figure 6.4, the null tree is given by tree D, where $b = 0$. Therefore, the test of $E(D) = 0$ for the trees A and B is equivalent to testing the null hypothesis of $b_6 = 0$. Indeed, we can show that

$$D = S_B - S_A = b_6/8 - (2\epsilon_{12} - 2\epsilon_{13} - \epsilon_{24} - \epsilon_{25} + \epsilon_{34} + \epsilon_{35})/8 \tag{6.19}$$

Therefore, when $b_6 = 0$, the expectation of D is 0. In practice, we do not know which of the trees A and B is the correct one. So, D can be positive or negative. Similarly, $D = S_C - S_A$ can be written as

$$D = S_C - S_A = 3(b_6 + b_7)/4 - 3(\epsilon_{12} - \epsilon_{14} - \epsilon_{25} + \epsilon_{45})/8 \quad (6.20)$$

This indicates that we are testing the null hypothesis that both b_6 and b_7 in tree A are 0 and that the null tree for this null hypothesis is tree E in Figure 6.4. This principle applies to any pair of bifurcating trees, irrespective of the number of sequences.

To test the null hypothesis of $E(D) = 0$, we have to know the standard error of D. Rzhetsky and Nei (1992a, 1993) developed a simple algorithm to compute the standard error of D for several substitution models. As long as the number of sequences used is relatively small (say, $m \leq 50$), their method is easily applicable. However, if m is large, it requires a substantial amount of computer time. Another way of testing is to use a bootstrap method (Nei 1991). In this bootstrap method, S is computed for a given pair of topologies (i and j) for each set of resampled sequences (see chapter 9), and $D_{ij} = S_i - S_j$ is computed. If this is repeated many times, we can compute the standard error of D. Therefore, we can test the null hypothesis of $E(D) = 0$ by the Z test given in Equation (4.5). When there are several potentially correct trees, D_{ij} can be computed for all pairs of i and j using the same set of resampled sequences.

Example 6.2. ME Trees for Hominoid Species

In sections 6.1 and 6.2, we constructed the UPGMA and the Fitch-Margoliash trees for five hominoid species using Kimura distances. Let us now construct the ME tree using the same set of pairwise distances (Table 6.1). In the present case, there are only 15 possible topologies, so it is easy to identify the ME tree. The tree obtained is presented in Figure 6.2C. The topology and the branch lengths of the tree are virtually identical with those of the FM tree. We also constructed the ME trees using the p distance, Jukes-Cantor distance, and Kimura gamma distance with $a = 0.53$, but these trees had the same topology, and their branch lengths were similar to those obtained with the Kimura distance.

To see the differences in the S value and branch length estimates between different topologies, let us consider the three possible trees for humans (H), chimpanzees (C), orangutans (O), and gibbons (B) given in Figure 6.5. The S value and the branch length estimates given in these trees were obtained by using Jukes-Cantor distance. Topology I, in which humans and chimpanzees make a cluster, has the smallest S value, and the S values for the other topologies are significantly greater than that for topology I. The bootstrap value (100%) also supports this topology. Therefore, topology I is the most likely tree.

Theoretically, it can be shown that in the absence of sampling error the interior branch of the correct topology for four sequences are always nonnegative, whereas that of an incorrect topology is negative (Rzhetsky and Nei 1992a; Sitnikova et al. 1995). In the present case, the interior branch is positive in topology I but is negative in topologies II and III. These results also support topology I. Incidentally, Figure 6.5 includes the MP and ML trees, which will be discussed later. In these trees, all interior branches are nonnegative, so that the positiveness of interior branches cannot be used for distinguishing between the correct and incorrect

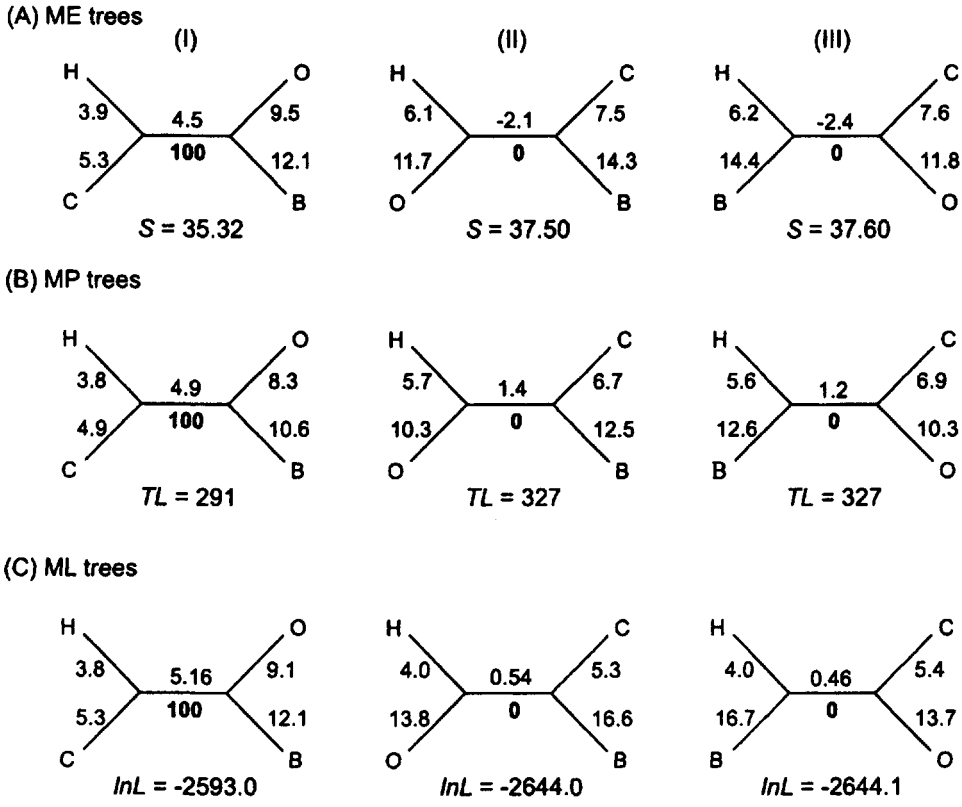

FIGURE 6.5. Estimates of branch lengths obtained by the ME, MP, and ML methods for a tree of humans (H), chimpanzees (C), orangutans (O), and gibbons (B). The Jukes-Cantor model was used in all calculations to make fair comparisons of ME and ML trees with MP trees. The bootstrap values for each case are shown below the interior branch. All branch lengths are in units of the number of substitutions per 100 sites.

topologies, although the interior branch of incorrect topologies tends to be smaller than that of the correct topology. However, the bootstrap test gives essentially the same conclusion as that for the ME tree.

6.4. Neighbor Joining (NJ) Method

Although the ME method has nice statistical properties, it requires a substantial amount of computer time when the number of taxa compared is large. Saitou and Nei (1987) developed an efficient tree-building method that is based on the minimum evolution principle. This method does not examine all possible topologies, but at each stage of taxon clustering a minimum evolution principle is used. This method is called the **neighbor joining (NJ) method** and is regarded as a simplified version of the ME method. When four or five taxa are used, the NJ and ME methods give identical results (Saitou and Nei 1987). There is some similarity between NJ and Sattath and Tversky's (1977) additive tree method (see also Fitch

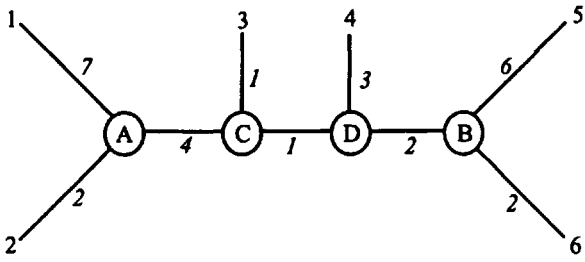

FIGURE 6.6. A phylogeny of six sequences with known branch lengths.

1981), but the former gives both the topology and branch lengths simultaneously.

One of the important concepts in the NJ method is **neighbors,** which are defined as two taxa that are connected by a single node in an unrooted tree. For example, taxa 1 and 2 in the tree of Figure 6.6 are neighbors, because they are connected by the only node A. Similarly, taxa 5 and 6 are neighbors, but all other pairs of taxa are not. However, if we combine taxa 1 and 2 and regard them as a single taxon, the combined taxon (1-2) and taxon 3 are now neighbors. It is possible to define the topology of a tree by successively joining neighbors and producing new pairs of neighbors. For example, the topology of the tree of Figure 6.6 can be described by the following pairs of neighbors: (1, 2), (5, 6), (1-2, 3), and (1-2-3, 4). Therefore, by finding these pairs of neighbors, one can obtain the tree topology.

Algorithm

Construction of a tree by the NJ method begins with a star tree, which is produced under the assumption that there is no clustering of taxa (Figure 6.7A). In practice, this assumption is generally incorrect. Therefore, if we estimate the branch lengths of the star tree and compute the sum of all branches (S_0), this sum should be greater than the sum (S_F) for the true or the final NJ tree. However, if we pick up neighbors 1 and 2 and consider the tree presented in Figure 6.7B, the sum (S_{12}) of all branch lengths should be smaller than S_0, although it may be greater than S_F. In practice, since we do not know which pair of taxa are true neighbors, we consider all pairs of taxa as a potential pair of neighbors and compute the sum of branch lengths (S_{ij}) for the i-th and j-th taxa using a topology similar to that given in Figure 6.7B. We then choose the taxa i and j that show the smallest S_{ij} value. Of course, actual distance values are subject to stochastic errors, so that the neighbors chosen in this way may not always be the true neighbors. Once a pair of neighbors are identified, they are combined into one composite taxon, and this procedure is repeated until the final tree is produced.

Mathematically, S_0 for the star tree is given by

$$S_0 = \sum_{i=1}^{m} L_{iX} = \frac{1}{m-1} \sum_{i<j}^{m} d_{ij}$$
$$= T/(m-1) \qquad (6.21)$$

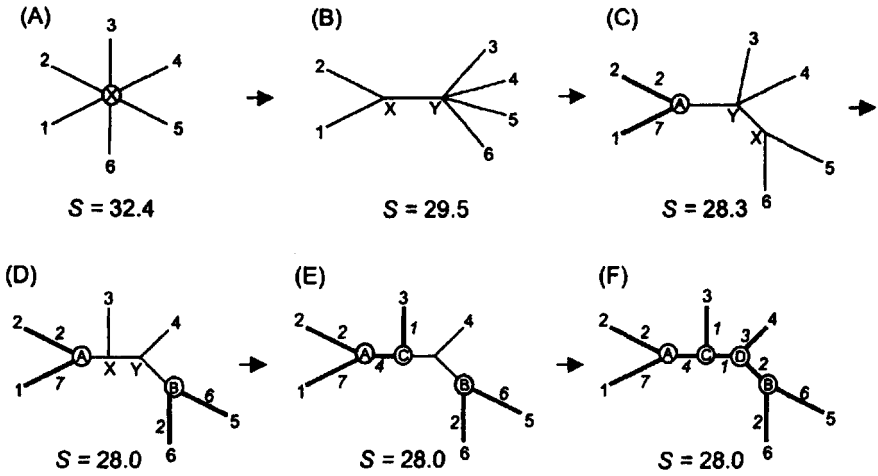

FIGURE 6.7. Illustration of the computational process in the neighbor-joining method.

where L_{iX} is the branch length estimate between nodes i and X, and $T = \Sigma_{i<j} d_{ij}$. In the present case, i stands for the i-th exterior node and X the interior node (Figure 6.7A). By contrast, Figure 6.7B indicates that S_{12} is given by the sum of $L_{1X} + L_{2X}$, L_{XY}, and $\Sigma_{i=3}^{m} L_{iY}$. Here, $L_{1X} + L_{2X} = d_{12}$, and

$$L_{XY} = \frac{1}{2(m-2)}\left[\sum_{i=3}^{m}(d_{1i} + d_{2i}) - (m-2)(L_{1X} + L_{2X}) - 2\sum_{i=3}^{m} d_{iY}\right]$$

$$\sum_{i=3}^{m} L_{iY} = \frac{1}{m-3}\sum_{3\le i<j} d_{ij}$$

Therefore, we have

$$\begin{aligned} S_{12} &= L_{1X} + L_{2X} + L_{XY} + \sum_{i=3}^{m} L_{iY} \\ &= \frac{1}{2(m-2)}\sum_{i=3}^{m}(d_{1i} + d_{2i}) + \frac{1}{2}d_{12} + \frac{1}{m-2}\sum_{3\le i<j} d_{ij} \end{aligned} \quad (6.22)$$

If we write $R_1 = \Sigma_{i=1}^{m} d_{1i}$ and $R_2 = \Sigma_{i=1}^{m} d_{2i}$, S_{12} can also be expressed as

$$S_{12} = \frac{2T - R_1 - R_2}{2(m-2)} + \frac{d_{12}}{2} \quad (6.23)$$

Obviously, S_{ij} can be computed in the same way if we replace 1 and 2 by i and j, respectively, in the above equations. Equation (6.23) requires less computational time than Equation (6.22). Furthermore, since T is the same for all pairs of i and j, S_{ij} can be replaced by

$$Q_{ij} = (m - 2)d_{ij} - R_i - R_j \tag{6.24}$$

for the purpose of computing the relative value of S_{ij} (Studier and Keppler 1988). Equation (6.24) is used for computer programming to facilitate the computation.

Once the smallest S_{ij} is determined, we can create a new node (A) that connects taxa i and j. The branch lengths (b_{Ai} and b_{Aj}) from this node to taxon i and taxon j are given by

$$b_{Ai} = \frac{1}{2(m-2)}\left[(m-2)d_{ij} + R_i - R_j\right] \tag{6.25a}$$

$$b_{Aj} = \frac{1}{2(m-2)}\left[(m-2)d_{ij} - R_i + R_j\right] \tag{6.25b}$$

(Saitou and Nei 1987; Studier and Keppler 1988). These values are known to be LS estimates for the topology under consideration (Saitou and Nei 1987). The next step is to compute the distance between the new node (A) and the remaining taxa (k; $3 \leq k \leq m$) (Figure 6.7C). This distance is given by

$$d_{Ak} = (d_{ik} + d_{jk} - d_{ij})/2 \tag{6.26}$$

If we compute all the distances using this equation, we have a new $(m - 1) \times (m - 1)$ matrix. From this matrix, we can compute a new S_{ij} matrix using Equation (6.23). However, we denote this new S_{ij} by S_{ij}', because this new S_{ij} does not include the lengths of exterior branches for the first pair of neighbors identified, and thus it is shorter than the real total sum (S_{ij}) of branch lengths at this stage of tree construction. To find the new pair of "neighbors," we choose a pair with the smallest S_{ij}' value. A new node B is then created for this pair of taxa, and a new $(m - 2) \times (m - 2)$ distance matrix is computed by using Equation (6.26). This procedure is repeated until all taxa are clustered in a single unrooted tree. The final tree obtained in this way is the NJ tree.

If one is interested in the reduction in S_{ij} in each cycle of neighbor joining, S_{ij} can be obtained by adding the lengths of all branches eliminated to S_{ij}'. This process of reduction in S_{ij} (represented by S) is shown in Figure 6.7, but in actual practice S_{ij} is rarely computed. In fact, most computer programs use Q_{ij} rather than S_{ij} or S_{ij}'.

To illustrate the computational procedure, let us consider the evolutionary distances given in cycle 1 of Table 6.2. These distances were obtained by adding the branch lengths for each pair of taxa of the tree in Figure 6.6. Therefore, all the distances satisfy the condition of additivity. The total sum of distances is $T = 162$. Therefore, S_0 for the star tree (Figure 6.7A) is 32.4 from Equation (6.21), since $m = 6$ in this case. We now compute S_{ij}'s for all pairs of i and j for topology B in Figure 6.7. For taxa 1 and 2, we have $d_{12} = 9$, $R_1 = 72$, and $R_2 = 52$, so S_{12} is 29.5 from Equation (6.23). Similarly, we compute S_{ij}'s for all other pairs of taxa, and they are shown in cycle 1 of Table 6.2. This table shows that the small-

Table 6.2 Distance and S_{ij}' matrices at sequential steps of the NJ algorithm.

	Distance Matrix					S_{ij} or S_{ij}' Matrix				
Cycle 1										
	1	2	3	4	5	1	2	3	4	5
1										
2	9					**29.5**				
3	12	7				32.5	32.5			
4	15	10	5			33.0	33.0	32.0		
5	20	15	10	11		33.5	33.5	32.5	32.0	
6	16	11	6	7	8	33.5	33.5	32.5	32.0	30.5
Selected Pair (1, 2) with branch lengths (7, 2); A = (1, 2)										
Cycle 2										
	A	3	4	5		A	3	4	5	
A										
3	5					19.7				
4	8	5				20.3	20.3			
5	13	10	11			21.0	21.0	20.7		
6	9	6	7	8		21.0	21.0	20.7	**19.3**	
Selected Pair (5, 6) with branch lengths (6, 2); B = (5, 6)										
Cycle 3										
	A	3	4			A	3	4		
A										
3	5					**11.0**				
4	8	5				11.5	11.5			
B	7	4	5			11.5	11.5	11.0		
Selected Pair (a, 3) with branch lengths (4, 1); C = (A, 3)										

est S_{ij} is $S_{12} = 29.5$. Thus, we infer that taxa 1 and 2 are neighbors. The fact that these two taxa are indeed a pair of neighbors is seen in Figure 6.6. The branch lengths of taxa 1 and 2 from the new node A in Figure 6.7 can be obtained by Equations (6.25) and becomes 7 and 2, respectively. These branch lengths are also identical with the true values of the tree in Figure 6.6. We now compute the distance between the new node A and taxon k using Equation (6.26). In the next step of neighbor joining (cycle 2 in Table 6.2), taxa 5 and 6 are found to be a pair of neighbors, because $S_{56}' = 19.3$ is the smallest S_{ij}' value. Therefore, we create a new node B and compute b_{5B} and b_{6B}. They are 6 and 2, respectively, which are again identical with those of the true tree (Figure 6.7). In cycle 3, taxa A and 3 show the smallest S_{ij}' value (= 11.0). Therefore, we now create a new node C. It is then obvious that node B and taxon 4 form a cluster. This creates another node D and completes the entire process of neighbor joining. We can now estimate branch lengths b_{4D}, b_{CD}, and b_{BD} in Figure 6.7F, and they become 3, 1, and 2, respectively. The final tree obtained is shown in Figure 6.7F. Both the topology and branch lengths of this tree are identical with those of the true tree in Figure 6.6.

However, this complete recovery of the original tree occurred because we used additive distances without any backward and parallel mutations. In real data, there are almost always backward and parallel mutations in some sequences, so it is not always easy to reconstruct the true tree. Therefore, it is important to conduct some statistical tests about the reliability of the tree obtained.

Using the sequence data in Figure 6.1, we produced two NJ trees for hominoid species using Kimura and p distances. Tree C in Figure 6.2 represents the NJ tree with Kimura distance, whereas tree D is the NJ tree with p distance. In these trees, the branch lengths were estimated by the ordinary LS method after the topology was determined. Note that the topology of those trees is identical with that of the ME tree obtained for the same data set.

Justification and Modifications

As mentioned above, the NJ method is based on the principle of minimum evolution but generates only one final topology with branch length estimates. Some authors criticized this method for this reason and suggested that the ME method rather than this method be used. Actually, it is possible to modify the NJ method to generate more topologies. Kumar (1996b) developed an algorithm in which not only the minimum S_{ij} but also several S_{ij}'s close to the minimum are considered as indicators of potential neighbors in each cycle of S_{ij} computation. This method generates as many topologies as desired and allows us to compare S values for different topologies. Therefore, one can choose the topology that shows the smallest S value. This is a hybrid method between the ME and NJ methods. A similar method has been proposed by Pearson et al. (1999).

As shown by Rzhetsky and Nei (1993), the ME method is expected to give the correct topology if the number of nucleotides examined (n) is sufficiently large and an unbiased estimate of nucleotide substitutions is used as a distance measure. When n is small and m is large, however, the S value (S_m) of the ME tree tends to be smaller than that (S_c) of the correct tree because of sampling errors. In fact, Nei et al. (1998) have shown that S_m is always equal to or smaller than S_c and that the probability of occurrence of $S_m < S_c$ is quite high when n is small (chapter 9). This indicates that it is not rewarding to spend excessive time to find the true ME tree when n is relatively small, because the true ME tree tends to be incorrect. Computer simulations by Saitou and Imanishi (1989), Rzhetsky and Nei (1992a), Gascuel (1997a, 1997b), and Nei et al. (1998) have also shown that the probability of obtaining the correct topology is nearly the same for both the ME and NJ methods. In other words, NJ is a fast method of constructing phylogenetic trees and is appropriate for analyzing a large data set. It is also capable of conducting bootstrap tests rapidly.

Backeljau et al. (1996) stated that NJ may produce two or more tie trees for the same data set. According to Takezaki (1998), NJ tie trees occur very rarely when the computation is done with high precision (much less than MP tie trees). Furthermore, even if multiple tie trees occur, they do not

pose any serious problem if a bootstrap consensus tree is produced. Therefore, we do not have to worry about them.

Gascuel (1997a) proposed the so-called **BIONJ method** to improve the efficiency of NJ in obtaining the correct topology. The computational algorithm is the same as that of NJ except that different weights are given to d_{ik}, d_{jk}, and d_{ij} in Equation (6.26) to minimize the variance of d_{Ak}. Using computer simulation, he showed that BIONJ is slightly better than NJ when sequence divergence is high. However, our limited experience with actual data analysis has shown that the two methods almost always give the same or very similar trees.

Example 6.3. NJ, ME, and BIONJ Trees for Simulated Sequence Data

To obtain some idea about the accuracy of NJ and BIONJ trees, let us consider the results of a small computer simulation presented in Figure 6.8. With real data, it is usually very difficult to know the true tree, so that it is virtually impossible to compare the reconstructed tree with the true tree. In a computer simulation, we can use a model tree and let a set of DNA sequences evolve following the model tree with a given pattern of nucleotide substitution. We can then reconstruct a tree using the DNA sequences generated and compare the reconstructed tree with the true tree.

The model and the realized trees for 24 DNA sequences used in the present simulation are shown in trees A and B of Figure 6.8, respectively. The number of substitutions in each branch of the realized tree were obtained by using pseudorandom numbers under the assumption that nucleotide substitution occurs following Kimura's model with a transition/transversion ratio (R) of 5 (see Saitou and Nei [1987] for the detail of the simulation). The number of nucleotides per sequence used in this simulation was 500. In both trees A and B, the branch lengths are measured in terms of the number of nucleotide substitutions per site. Note that the branch lengths of the realized tree are much more variable than those of the model tree because of stochastic errors. The topology of the realized tree is identical with that of tree A except for one trifurcating node that occurred because one interior branch corresponding to branch a of the model tree did not have any nucleotide substitution.

Figure 6.8C shows the NJ tree obtained by using the Jukes-Cantor distance for the 24 sequences that were generated by computer simulation. Comparison of this tree with the realized tree shows that tree C has one topological error. That is, the trifurcating node for sequences 8, 9, and 10 in the realized tree is decomposed into two consecutive bifurcating nodes in tree C, though the branch length between the two nodes is very close to 0. This occurred because NJ is designed to construct a bifurcating tree. Except for this minor difference, the topology of this tree is identical with that of the realized tree. The branch length estimates are also very close to those of tree B. Note that here we used the Jukes-Cantor distance instead of the Kimura distance, which is more appropriate in the present case. Yet, the reconstructed tree is very close to the true realized tree. When we used the Kimura distance, we obtained a tree that was very

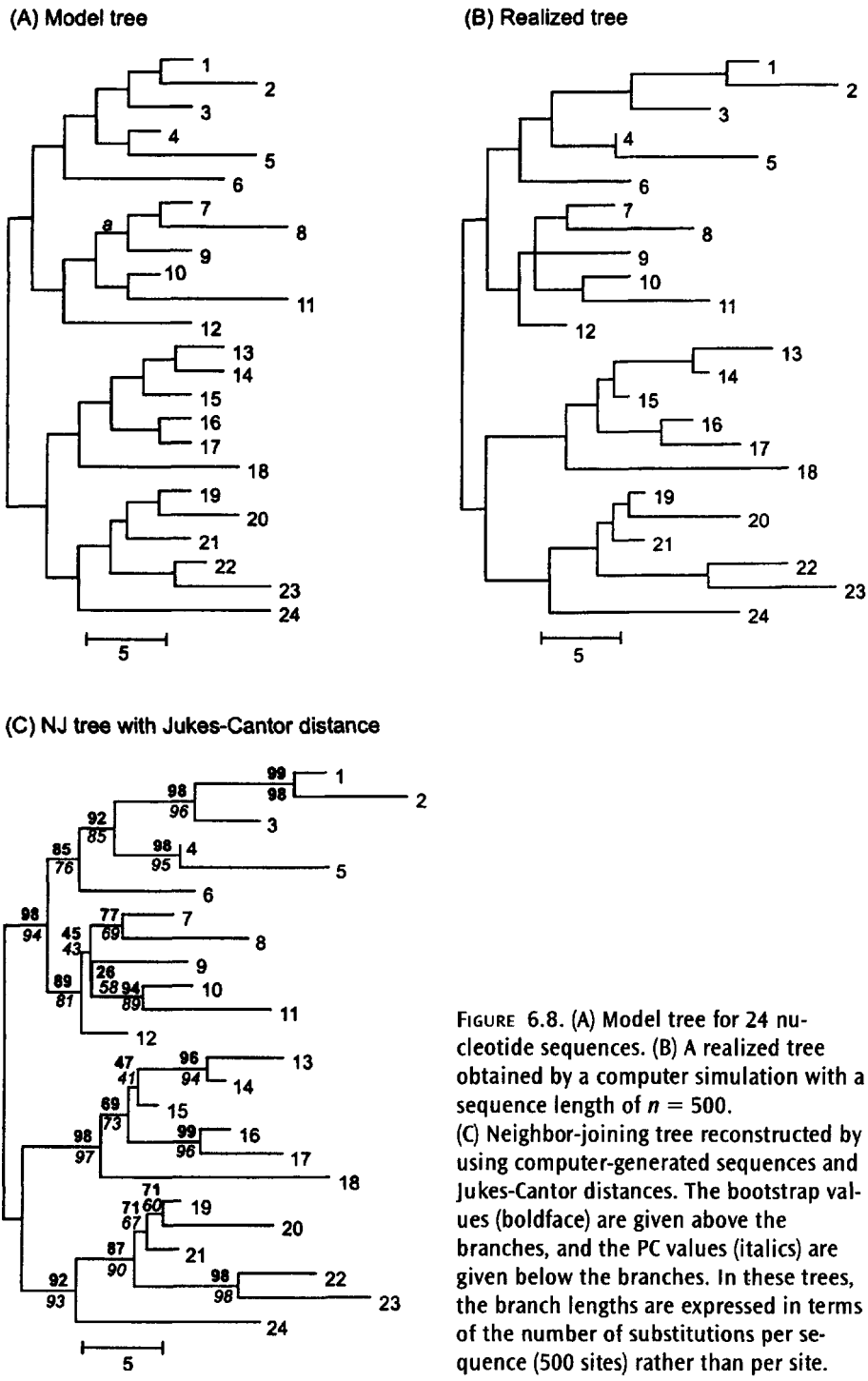

FIGURE 6.8. (A) Model tree for 24 nucleotide sequences. (B) A realized tree obtained by a computer simulation with a sequence length of $n = 500$. (C) Neighbor-joining tree reconstructed by using computer-generated sequences and Jukes-Cantor distances. The bootstrap values (boldface) are given above the branches, and the PC values (italics) are given below the branches. In these trees, the branch lengths are expressed in terms of the number of substitutions per sequence (500 sites) rather than per site.

similar to tree C, though the Kimura distance tree had slightly longer branch lengths near the root as expected. The similarity of the two reconstructed trees is of course due to the fact that the extent of sequence divergence is low in the present case. We also constructed the ME and BIONJ trees, but they had essentially the same topology and branch lengths as those of tree C.

In the above example, the topological error of a reconstructed tree occurred because one interior branch of the realized tree had no nucleotide substitution, as mentioned above. In fact, zero-length interior branches in realized trees are a source of topological errors in reconstructed trees, particularly when there are many such branches. Unfortunately, we usually do not know such interior branches in real data, and therefore it is difficult to evaluate the effect of this factor, though parsimony methods are capable of identifying such branches under certain conditions (chapter 7). However, such interior branches almost always give low bootstrap values, whether or not the branch pattern obtained is correct. In fact, the interior branch of the NJ trees associated with the zero-length branch in Figure 6.8 has a bootstrap value of 26%. Therefore, if we disregard low-bootstrap interior branches, we can conclude that the NJ, ME, and BIONJ methods reconstruct the true tree quite accurately.

However, this happened partly because the pattern of nucleotide substitution used was relatively simple. In most DNA sequences, the actual substitution pattern is much more complicated than the Kimura model, and this would introduce topological errors even when d's are only moderately large. For this reason, a number of statistical tests of the reliability of an inferred tree have been developed. This problem will be discussed in chapter 9.

6.5. Distance Measures to Be Used for Phylogenetic Reconstruction

In chapters 2–4, we discussed various distance measures for estimating the number of nucleotide or amino acid substitutions (d) considering different mathematical models. In general, a distance measure based on a complex mathematical model requires many parameters to be estimated, and this increases the variance of the estimate of d. Theoretically, it is possible to choose a mathematical model most appropriate for a given set of data using certain statistical criteria. Several such statistical methods are now available (Kishino and Hasegawa 1989; Bulmer 1991; Goldman 1993; Rzhetsky and Nei 1995; Yang 1995a), but in these methods the increment of variance by adding more parameters are not considered. Therefore, the best distance measure identified by these criteria is not necessarily most appropriate for reconstruction of phylogenetic trees, although they are usually useful for branch length estimation.

Generally speaking, the accuracy of an inferred tree depends on at least two factors: (1) the linear relationship of the distance used with the number of substitutions and (2) the standard error or the coefficient of variation of the estimate of the distance measure. For Kimura's (1980) model of nucleotide substitution, several authors have attempted to produce

better distance measures than the original estimator, taking into account these two factors (Schöniger and von Haeseler 1993; Goldstein and Pollock 1994; Tajima and Takezaki 1994), but the practical utility of these distance measures is still unclear.

At the present time, there is no general statistical method for choosing an appropriate distance measure (or mathematical model) for constructing tree topologies. However, computer simulations and empirical studies have led to the following guidelines for the purpose of topology construction (modified from Nei 1996).

1. When the Jukes-Cantor estimate of the number of nucleotide substitutions per site (d) is about 0.05 or less ($d \leq 0.05$), use the p or Jukes-Cantor distance whether there is a transition/transversion bias or not or whether the substitution rate (r) varies with nucleotide site or not. In this case, the Kimura distance and more complicated distance measures give essentially the same value as the p or Jukes-Cantor distance (Figure 3.1), but their variances are greater than those of the latter distances. The p distance tends to give good results, particularly when the number of nucleotides or amino acids used is small.

2. When $0.05 < d < 1.0$ and the number of nucleotides examined is large, use the Jukes-Cantor distance unless the transition/transversion ratio (R) is high, say, $R > 5$. When this ratio is high and the number of nucleotides examined (n) is very large, use the Kimura distance or the gamma distance. However, when the number of sequence is large and n is relatively small, the p distance often gives better results unless the evolutionary rate varies extensively with evolutionary lineage (Takahashi and Nei 2000). In recent years, a number of authors have used maximum likelihood estimates of the HKY gamma distance, apparently because in theory this distance takes care of the GC content and transition/transversion biases as well as the variation in substitution rate among different sites (e.g., Honda et al. 1999). However, computer simulations with 48 nucleotide sequences have shown that with most reasonable model trees this distance generally gives a poorer performance than the p or the Jukes-Cantor distance even if the HKY gamma model is used for generating sequence data (Takahashi and Nei 2000). This is because the HKY gamma distance has a large variance compared with the p or the Jukes-Cantor distance. When the number of nucleotides examined is very large (>10,000) and the rate of nucleotide substitution varies extensively with evolutionary lineage, a complicated distance measure (e.g., HKY gamma distance) may give better results (Takezaki and Gojobori 1999).

3. When $d > 1$ for many pairs of sequences, the phylogenetic tree constructed is generally unreliable for a number of reasons (e.g., large variances of $\hat{d}$'s and sequence alignment errors). We therefore suggest that these data sets should be avoided as much as possible. In this case, one may eliminate the portion(s) of the gene that evolves very fast and use only the remaining region(s) as is often done with immunoglobulin variable region genes (Ota and Nei 1994a; Rast et al. 1994). One may also use a different gene that evolves more slowly.

4. Many distance measures for estimating the number of nucleotide substitutions per site (d) often becomes inapplicable when the distance is very large and n is small. This happens because the mathematical for-

mulas for distance estimation usually involve logarithmic terms, and the arguments of the logarithms often become negative. Theoretically, this problem can be avoided by expanding the logarithmic terms into an infinite series, but the variance of the distance estimated in this way is quite large (Tajima 1993b; Rzhetsky and Nei 1994). Therefore, highly divergent sequences should not be used for topology construction. In this case the p distance is often more efficient for obtaining a reliable topology, because it is always applicable and has a smaller variance.

5. When a phylogenetic tree is constructed from the coding regions of a gene, the distinction between synonymous (d_S) and nonsynonymous (d_N) substitutions may be helpful, because the rate of synonymous substitutions is usually much higher than that of nonsynonymous substitution. When relatively closely related species are studied for a large number of codons and $d_S < 0.5$, one may use d_S for constructing a tree. This procedure is expected to reduce the effect of variation in substitution rate among different sites, because synonymous substitutions are subject to selection less often than nonsynonymous substitutions. However, when relatively distantly related species are studied, d_N or amino acid distances seem to be better. Note also that d_S or p_S sometimes reaches the saturation level rather quickly (chapter 5).

6. As a general rule, if two distance measures give similar distance values for a set of data, use the simpler one because it has a smaller variance. When the rate of nucleotide substitution is nearly the same for all evolutionary lineages and there is no strong transition/transversion bias, the p distance seems to give correct trees more often than other distances, even if sequence divergence is high (Schöniger and von Haeseler 1993; Tajima and Takezaki 1994; Takahashi and Nei 2000). When the substitution rate varies with evolutionary lineage, however, this may not be the case. It is important not to trust computer outputs of tree construction without scrutinizing the pattern of nucleotide or amino acid substitution, differences in nucleotide frequencies among the first, second, and third codon positions, temporal changes of nucleotide frequencies, and so forth. In real data analysis, there are so many unknown factors that the phylogenetic tree produced should be interpreted with caution and common sense.

Note that the above guidelines are for constructing phylogenetic trees. For estimating branch lengths or evolutionary times, unbiased estimators are generally better than biased estimators.

This page intentionally left blank