Improvement of Distance-Based Phylogenetic Methods by a Local Maximul Likelihood Approach Using Triplets

Vincent Ranwez, Olivier Gascuel

Improvement of Distance-Based Phylogenetic Methods by a Local Maximum Likelihood Approach Using Triplets

Vincent Ranwez and Olivier Gascuel

Département Informatique Fondamentale et Applications, LIRMM, Montpellier Cedex 5, France

We introduce a new approach to estimate the evolutionary distance between two sequences. This approach uses a tree with three leaves: two of them correspond to the studied sequences, whereas the third is chosen to handle long-distance estimation. The branch lengths of this tree are obtained by likelihood maximization and are then used to deduce the desired distance. This approach, called TripleML, improves the precision of evolutionary distance estimates, and thus the topological accuracy of distance-based methods. TripleML can be used with neighbor-joining–like (NJ-like) methods not only to compute the initial distance matrix but also to estimate new distances encountered during the agglomeration process. Computer simulations indicate that using TripleML significantly improves the topological accuracy of NJ, BioNJ, and Weighbor, while conserving a reasonable computation time. With randomly generated 24-taxon trees and realistic parameter values, combining NJ with TripleML reduces the number of wrongly inferred branches by about 11% (against 2.6% and 5.5% for BioNJ and Weighbor, respectively). Moreover, this combination requires only about 1.5 min to infer a phylogeny of 96 sequences composed of 1,200 nucleotides, as compared with 6.5 h for FastDNAml on the same machine (PC 466 MHz).

Introduction

Using a sequence evolution model enables evaluation of the likelihood that a given phylogeny will yield the observed sequences. When a large set of sequences is studied, the likelihood of every possible phylogeny cannot be estimated within a reasonable time. This problem is generally handled with a heuristic approach, so that only a subset of promising phylogenies is studied. As long as only a few sequences are considered, maximum likelihood methods infer reliable phylogenies within a reasonable time (e.g., Kuhner and Felsenstein 1994). But despite improvements to the original maximum likelihood algorithm (Felsenstein 1981) to speed it up, especially that of Olsen et al. (1994), maximum likelihood methods remain so slow that they are only suitable for dealing with small data sets.

But estimation of the distances between all pairs of sequences can be done very fast on the basis of a maximum likelihood approach. Moreover, efficient algorithms are available for inferring a phylogeny that fits this matrix of pairwise distances, with the neighbor-joining (NJ) algorithm (Saitou and Nei 1987) being the most popular. This algorithm, according to the agglomerative process introduced by Sattah and Tversky (1977), selects a pair of taxa to be merged at each step. The two selected taxa are then replaced by a single new taxon, and the distance matrix is reduced by replacing the distances relative to the two merged nodes by those relative to the new node. NJ has low computational time complexity, so it can cope with very large data sets, and computer simulations (Saitou and Nei 1987; Nei 1991; Kuhner and Felsenstein 1994) have demonstrated its topological accuracy.

Key words: phylogenetic reconstruction, evolutionary distance, maximum likelihood, triplet method.

Address for correspondence and reprints: Olivier Gascuel, Département Informatique Fondamentale et Applications, LIRMM, 161 rue Ada, 34392 Montpellier Cedex 5, France. E-mail: gascuel@lirmm.fr.

Mol. Biol. Evol. 19(11):1952–1963. 2002

Variants of the original NJ algorithm, such as BioNJ (Gascuel 1997*a*) or Weighbor (Bruno, Socci, and Halpern 2000), have been proposed for increasing its topological accuracy. Other distance approaches have also been explored, e.g., least-squares tree fitting as implemented in the FITCH program (Felsenstein 1993). Yet, the topological accuracy of NJ and other distance-based approaches is lower than that of maximum likelihood methods (Kuhner and Felsenstein 1994; Swofford et al. 1996, p. 446).

Clearly, new phylogenetic approaches have to be found, so that evolutionary trees—more reliable than those of NJ and related methods—may be inferred even for large data sets. A possible solution is to use a distance method to restrict the number of trees studied by a maximum likelihood approach, as in the NJML method proposed by Ota and Li (2000). Another possible approach, explored by Strimmer and Von Haeseler (1996) and others, is to combine small four-taxon trees, obtained by maximum likelihood, to infer a larger tree that will hopefully have a high likelihood. But although it is likely that these quartet methods will be improved, their current performances are disappointing (Ranwez and Gascuel 2001).

Another direction is explored in this article. Indeed, irrespective of the distance-based methods used, the quality of the pairwise distances is essential. As we shall see, these distances can be better estimated by a local maximum likelihood approach based on triplets of taxa. We start by describing NJ and its variants. We then explain how to improve distance estimation used by these methods. Finally, we study, by computer simulations, the contribution of this new approach and conclude by analyzing some possible improvements.

The TripleML Approach

First, we recall how the likelihood of a given phylogeny is computed, following the approach introduced by Felsenstein (1981) and described at length in Swofford et al. (1996, pp. 430–442). Then, we detail NJ and

describe BioNJ (Gascuel 1997*a*) and Weighbor (Bruno, Socci, and Halpern 2000), which are two variants of NJ. Lastly, we present our distance estimation method and specify how it can be combined with any NJ-like algorithm.

Likelihood Computation

The starting point for using the maximum likelihood approach is to define a model of sequence evolution. The likelihood of a phylogeny T then depends on the phylogeny itself (including its branch lengths) and on the other model parameters. We place our study in the framework of the general time reversible model (Lanave et al. 1984), which generalizes most commonly used models such as Kimura's two parameter model (Kimura 1981) or F84 (Felsenstein 1993). Yang, Goldman, and Friday (1994) have shown that the parameters of such reversible models can reasonably be estimated before the phylogenetic reconstruction. We thus assume, as usual, that the model parameters have been estimated previously so that the likelihood only depends on the topology and the branch lengths of the phylogeny. Moreover, for these models, the likelihood of a known phylogeny T can be recursively computed regardless of the ancestral sequence position, and each site can be treated independently (Swofford et al. 1996, pp. 440–441). The probability of observing the n sequences of length l associated with leaves of T is the product of the probability of observing each of the l sites. Denoting S_a as the (unknown) ancestral sequence, the probability of site s regarding nucleotide b is obtained by multiplying the probability π_b that b was the ancestral nucleotide with the probability $L(S_a^s = b; T)$ that b was transformed into the n nucleotides observed at sites s of the leaf sequences of the tree T. Thus, the likelihood of a tree T, with known topology and branch lengths, is obtained using the following formula

$$L(T) = \prod_{s=1}^{l} \sum_{b \in \{A,C,G,T\}} \pi_b L(S_a^s = b; T). \qquad (1)$$

The likelihood term $L(S_a^s = b; T)$ is recursively computed. Let us assume that the tree T is composed of the two subtrees T_i and T_j, which are associated with ancestral sequences denoted as S_i and S_j, respectively. Then, the likelihood of T is computed using the likelihood of T_i and T_j and using the probabilities $P_{bc}(\delta)$ that for an evolutionary distance δ, a nucleotide b becomes c. Using these notations, we have

$$L(S_a^s = b; T) = \prod_{x \in \{i,j\}} \sum_{c \in \{A,C,G,T\}} P_{bc}(\delta_{ax}) L(S_x^s = c; T_x), \qquad (2)$$

where δ_{ax} denotes the evolutionary distance (branch length) between S_a and S_x, with $x = i$ or $x = j$. Likewise, the likelihood of T_i is computed using the likelihood of its subtrees. The recursive process continues until the subtree is reduced to a single leaf. This leaf—denoted as T_f—is associated with a single contemporary taxon of known sequence (denoted as S_f) that completely defines the likelihood of T_f

$$L(S_f^s = b; T_f) = \begin{cases} 1 & \text{when } S_f^s = b \\ 0 & \text{otherwise.} \end{cases} \qquad (3)$$

For a single site, each recursive step is done in constant time, and the number of recursive steps is proportional to the number (n) of taxa. Because this has to be done for each of the l sites, the time complexity of the likelihood computation is $O(nl)$. In the above description, we assumed that branch lengths were known, but they are generally unknown and must be adjusted so as to maximize the likelihood. This optimization is typically done by making a number of passes over the tree, adjusting branch lengths one at a time, and the passes are continued until the process converges (Olsen et al. 1994).

For simplicity, we focus on evolutionary models that assume that every site evolves at the same rate. But the adaptation of our method for models that explicitly incorporate site-to-site variation is straightforward. Regarding a discrete rate distribution, the full likelihood of a given site is obtained by simply summing over rate categories the likelihoods of the site according to each rate, weighted by the probability that the site is drawn from each category (Yang 1993).

NJ and its Variants

In what follows, we use the simplified expression of NJ from Studier and Keppler (1988) rather than the original one. The equivalence of both expressions as well as their correctness are demonstrated by Gascuel (1994, 1997*b*).

At each step, NJ uses a distance matrix (δ_{ij}), where i and j are either taxa or clusters of taxa agglomerated during previous steps. On the basis of these distances, two taxa are selected to be merged; they lose their individual identities and are then referred to as a single cluster. Initially, each taxon constitutes its own cluster, and the dimension of the matrix, denoted as r, is thus equal to the number n of studied taxa. At each agglomeration, as two clusters are merged into one, r declines by 1 just like the number of clusters. Denoting Q_{12} as the criterion value for the agglomeration of the two clusters 1 and 2, the pair agglomerated is the one minimizing

$$Q_{12} = (r - 2)\delta_{12} - \Delta_1 - \Delta_2 \quad \text{where}$$

$$\Delta_x = \sum_{y=1}^{r} \delta_{xy}. \qquad (4)$$

Once the pair to be agglomerated is selected, NJ creates a new node i which represents the root of the new cluster. Then, NJ estimates the length of branches $(1, i)$ and $(2, i)$ using the formulae

$$\delta_{1i} = \frac{1}{2}\left(\delta_{12} + \frac{\Delta_1 - \Delta_2}{r - 2}\right) \quad \text{and}$$

$$\delta_{2i} = \frac{1}{2}\left(\delta_{12} + \frac{\Delta_2 - \Delta_1}{r - 2}\right). \qquad (5)$$

Finally, NJ reduces the distance matrix by replacing the

distances relative to taxa 1 and 2 by those between the new node i and any other node j using

$$\delta_{ij} = \frac{1}{2}(\delta_{1j} - \delta_{1i}) + \frac{1}{2}(\delta_{2j} - \delta_{2i}). \quad (6)$$

The process stops when $r = 2$, with the last branch length being equal to the last value in the distance matrix. NJ variants may use other estimates for δ_{1i}, δ_{2i}, and δ_{ij} and even a different criterion to select the pair that is agglomerated. Yet, all these methods share the same agglomerative scheme described above.

When taxa 1 and 2 are merged into a new taxon i, the new distances δ_{ij} can be estimated by any convex combination of $(\delta_{1j} - \delta_{1i})$ and $(\delta_{2j} - \delta_{2i})$. The NJ algorithm assumes that both estimates are equally important and gives both the same weight (1/2). BioNJ chooses the weights that provide the δ_{ij} estimate of minimal variance. In this way, better estimates are available for the following steps of the algorithm. BioNJ improves the topological accuracy of NJ, especially when the substitution rates are high and vary among lineages, while retaining its computation speed (Gascuel 1997*a*).

Weighbor uses a different criterion to select the pair it agglomerates. This criterion takes into account that the larger the evolutionary distance, the worst is its estimation. Whereas the NJ criterion is based on a minimum evolution approach, the Weighbor criterion embodies a likelihood function on the distances, which are modeled as Gaussian random variables. This distance model is also used to reduce the distance matrix. Weighbor is less sensitive to the long-branch attraction bias observed in NJ and BioNJ (Bruno, Socci, and Halpern 2000) but is significantly slower than both previous algorithms.

Overview of TripleML

For all these methods, the estimate of distance values is a key point. We introduce a new method that improves the precision of distance estimation and thus increases the topological accuracy of NJ and its variants. In these methods, there are two kinds of estimation, i.e., the prior estimation of distances between pairs of contemporary taxa and estimation done after each agglomeration step to evaluate distances between the new cluster and those already existing. As we shall see, both are improved by our approach.

The usual estimate of the distance δ_{ij} separating taxon i from taxon j is obtained by optimizing the likelihood of the ''tree'' containing these two taxa. This very simple tree is composed of one branch and two leaves and is called a 2-tree. The likelihood expression of this minimal phylogeny is much simpler than that of the general formula, and the likelihood optimization task is generally easy. For some models, (e.g., Kimura 1981), there is even an analytical solution to this optimization problem.

Instead of the usual approach of estimating the initial distance between taxa i and j on the basis of the corresponding 2-tree, we propose to estimate it using a 3-tree. Two leaves of this tree are the studied taxa, whereas the third, denoted as k, is chosen to handle long-distance difficulties. The branch lengths of this tree are determined so as to maximize its likelihood and are then used to obtain a more reliable estimation of δ_{ij}. This 3-tree is obtained by linking the three taxa i, j, and k to a common ancestral node a using tree branches with lengths δ_{ai}, δ_{aj}, and δ_{ak}, respectively. The distance δ_{ij} between taxa i and j is then estimated by $\delta_{ij} = \delta_{ai} + \delta_{aj}$. The quality of this δ_{ij} estimation depends on the third taxon, but likelihood optimization for all 3-trees containing i and j is too time consuming. Hence, a third taxon allowing a good estimation of δ_{ij} must be chosen before the 3-tree likelihood optimization. We thus rely on a two-stage approach. First, initial pairwise distances are estimated as usual. Second, these first estimates are used to select a third taxon for each pair (i, j) to improve the first rough estimation of δ_{ij}. The distance δ_{ij} is then estimated by optimizing the likelihood of the 3-tree containing these three taxa.

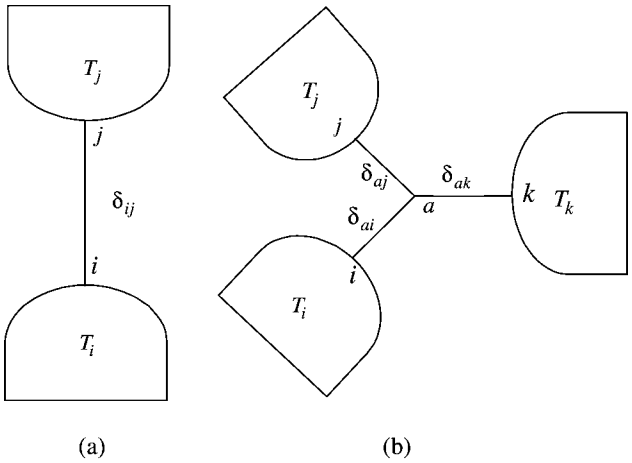

FIG. 1.—Estimation of the distance δ_{ij} separating T_i from T_j. (*a*) A first rough estimate of δ_{ij} is obtained using the $T_i \cup T_j$ phylogeny. (*b*) This first estimate is refined using a third subtree T_k and the phylogeny $T_i \cup T_j \cup T_k$; we then have $\delta_{ij} = \delta_{ai} + \delta_{aj}$.

After each agglomeration, new δ_{ij} distances are estimated, and the distance matrix is reduced. NJ, BioNJ, and Weighbor estimate these new distances using formula (6) or analogous formulae based on distance averaging. Our method estimates them through the same maximum likelihood approach as that used for estimating the initial matrix. An agglomeration induces a new subtree representing a cluster of taxa, and the aim is to estimate the distances between this new subtree and the existing ones. The distance between a subtree T_i, with i as root, and a subtree T_j, with j as root, may be estimated by considering the phylogeny $T_i \cup T_j$ (fig. 1*a*) obtained from T_i and T_j by adding the branch (i, j). The branch lengths of $T_i \cup T_j$ may be adjusted to maximize its likelihood. This provides not only the desired δ_{ij} estimate but also new estimates of T_i and T_j branch lengths. To keep low computation times, we decided not to question previous estimates of T_i and T_j branch lengths and to locally optimize $T_i \cup T_j$ likelihood only with regard to δ_{ij}.

A better δ_{ij} estimate can then be obtained by taking into account a third subtree T_k, with the root denoted as k. In this case, we consider the phylogeny $T_i \cup T_j \cup T_k$ obtained from T_i, T_j, and T_k by linking the three nodes i, j, and k with a new node a using three branches with lengths δ_{ai}, δ_{aj}, and δ_{ak}, respectively (fig. 1*b*).

Thus, the distances δ_{ij} separating a newly agglomerated tree T_i from the other subtrees T_j are obtained in two stages. First, these distances are estimated through a local likelihood optimization of $T_i \cup T_j$. On the basis of these first estimates, a third subtree is then selected for each pair (T_i, T_j) to improve the first rough estimation of δ_{ij}. The δ_{ij} distance is finally estimated using local likelihood optimization on $T_i \cup T_j \cup T_k$.

We next describe the procedure we use to estimate initial pairwise distances and to reduce the distance matrix. We then explain how these distance estimations can be combined with NJ-like algorithms without increasing their computational time complexity.

Initial Pairwise Distance Estimation

Let S_i be the sequence of taxon i and S_j that of taxon j. Assuming that S_i is the ancestral sequence, the likelihood of the 2-tree T containing i and j is defined as follows from equations (1), (2), and (3):

$$L(T) = \prod_{s=1}^{l} \pi_{S_i^s} P_{S_i^s S_j^s}(\delta_{ij}). \tag{7}$$

The maximization of this likelihood provides the first rough δ_{ij} estimates. This optimization requires numerical techniques unless a direct analytical solution is available. These first estimates are then used to select a third taxon k with sequence S_k for each pair (i, j). Denoting T as the tree that contains these three taxa and assuming that S_a is the sequence of the ancestral node a, the likelihood of T is

$$L(T) = \prod_{s=1}^{l} \sum_{b \in \{A,C,G,T\}} \left[\pi_b \prod_{x \in \{i,j,k\}} P_{bS_x^s}(\delta_{ax}) \right]. \tag{8}$$

The branch lengths of T are adjusted so as to maximize this likelihood, and the distance between S_i and S_j is then re-estimated by $\delta_{ij} = \delta_{ai} + \delta_{aj}$. The optimization procedure and the criterion to select k are further detailed.

Using Maximum Likelihood to Reduce the Distance Matrix

After each agglomeration, a similar approach is used to estimate the δ_{ij} distances separating the newly agglomerated subtree T_i from other subtrees T_j (this latter may be reduced to a single taxon). To obtain the first estimates of these δ_{ij} distances, we locally optimize the likelihood of the tree $T = T_i \cup T_j$ (fig. 1*a*). This likelihood, assuming that S_i is the ancestral sequence, is defined as follows from equations (1) and (2):

$$L(T) = \prod_{s=1}^{l} \sum_{b \in \{A,C,G,T\}} \left[\pi_b L(S_i^s = b; T_i) \times \sum_{c \in \{A,C,G,T\}} P_{bc}(\delta_{ij}) L(S_j^s = c; T_j) \right]. \tag{9}$$

On the basis of these first estimates, we select a third subtree T_k for each pair (T_i, T_j). δ_{ij} is then re-estimated through a local likelihood optimization of the tree $T = T_i \cup T_j \cup T_k$ having

$$L(T) = \prod_{s=1}^{l} \sum_{b \in \{A,C,G,T\}} \left[\pi_b \times \prod_{x \in \{i,j,k\}} \sum_{c \in \{A,C,G,T\}} P_{bc}(\delta_{ax}) L(S_x^s = c; T_x) \right]. \tag{10}$$

All the $L(S_x^s = c; T_x)$ values must be known for computing this likelihood. We call this set of values the likelihood vector of T_x, and we denote it as $LV(T_x)$. For each of the l sites, $LV(T_x)$ contains four values, one for each possible nucleotide. So the likelihood vector of a subtree is made of $4l$ values. After each agglomeration, the likelihood vector of the new subtree T_i is computed and stored for further use. Subtrees are initially made of a single taxon; thus, their likelihood vectors are completely defined by their sequences (eq. 3). When two subtrees are merged into a new subtree T_i, the values of $LV(T_i)$ are computed from equation (2) using $LV(T_1)$, $LV(T_2)$, and the lengths δ_{1i} and δ_{2i} obtained from equation (5). After this agglomeration, $LV(T_1)$ and $LV(T_2)$ are useless. The number of likelihood vectors is initially equal to the number n of studied taxa, and this number then decreases after each agglomeration. The memory space required to store these values is thus in the same range $[O(nl)]$ as the memory required to store nucleotide sequences of the n taxa under study.

Selection of the Third Taxon (or Subtree)

In our approach, the estimation of the distance separating two contemporary taxa is a particular case of the estimation of the distance separating two subtrees T_i and T_j having, respectively, i and j as roots (fig. 1). Using a third subtree T_k brings more information and improves the estimate of the δ_{ij} distance. This phenomenon was already pointed out by Swofford et al. (1996, p. 499). Indeed, to obtain a more reliable phylogeny on a set of taxa, they advise that a tree be computed for a larger set (interspersed among those of interest) and that the tree be then pruned. Moreover, they specify that to be most effective, the additional taxa should be chosen so as to divide long branches reasonably evenly. On the basis of their scheme, we thus seek a subtree T_k such that the branch (i, j) is cut near its middle, which is measured—using the first estimates—by $(\delta_{ik} - \delta_{jk})^2$. When dividing the branch (i, j), we also create a new branch (a, k). Cutting a long branch by creating another long branch

would be of little gain. We thus want T_k to be close to T_i and T_j, which is measured by $\delta_{ik}\delta_{jk}$. Both measurements are of the same order because they are both distance products. To estimate the δ_{ij} distance, we therefore use the tree T_k that minimizes $(\delta_{ik} - \delta_{jk})^2 + \delta_{ik}\delta_{jk}$. Clearly, this criterion is minimal when $\delta_{ai} = \delta_{aj}$ and $\delta_{ak} = 0$. When *i*, *j*, and *k* are taxa, this criterion is thus minimal in the ideal case where *i* and *j* have an equidistant ancestor, and the sequence of this ancestor is known.

Note that in this approach, the third subtree is selected on the basis of the current distance matrix. Therefore, if the process were repeated after the re-estimation of distances using triplets, the choice of the third subtree could change. In practice, however, this rarely happens (only 4% of the cases in our simulations). Moreover, when it does, the criterion value for the formerly selected subtree is very close to that of the newly selected one, so it is not worthwhile to re-estimate the distance. Note also that other criteria can be used to select the third subtree. We have tested many, but none of them improves the performance obtained with the simple one provided above.

Maximum Likelihood Optimization Process

Using TripleML requires optimizing numerous tree likelihoods at each stage. We use a simple optimization method that does not require any derivative computation, which makes its use easy with complicated sequence evolution models.

To pinpoint the value δ_{ij} (locally) maximizing the likelihood of $T_i \cup T_j$, we use the Brent optimization method of one parameter function, as described in Press et al. (1988, pp. 299–302). At each stage, this method defines three values *a*, *b*, and *M* such that $a < M < b$ and $f(a) < f(M)$ and $f(b) < f(M)$. The optimal value we are searching for is bracketed between *a* and *b*, and *M* is the point with the highest function value found so far. There is only one parabola joining these three points. The abscissa *s* of its pinnacle defines the new *M* value, and the interval (*a*, *b*) is refined to (*a*, *M*) or (*M*, *b*) depending on whether $s < M$ or $s > M$. At each iteration, the interval containing the sought maximum is reduced, and the optimization process stops when the desired precision is reached.

If T_i, T_j, and T_k branch lengths are fixed, the likelihood of $T_i \cup T_j \cup T_k$ only depends on δ_{ai}, δ_{aj}, and δ_{ak}. We then seek the values of these three lengths for which the likelihood of $T_i \cup T_j \cup T_k$ is (locally) maximal. Assuming that two of these three lengths are fixed turns the problem into a one-variable function optimization. So the third branch length can be determined using the Brent optimization as described above. This value is then supposed to be fixed, and the likelihood is optimized with regard to one of the two other branch lengths. The likelihood of the whole tree is thus optimized by making passes over the three branches and adjusting them one at a time. The passes are continued as long as they significantly increase the likelihood of $T_i \cup T_j \cup T_k$. We restrain the number of iterations involved for each branch optimization to two because, as pointed out by Olsen et al. (1994), this optimization effort can be invalidated by subsequent changes of other branch lengths.

Combining This Distance Estimation with the NJ Agglomeration Process

The previously described distance estimation can be used with any variant of NJ. This can be done by simply replacing the distance estimates of the method by the estimates we introduced. The only difference between NJ and BioNJ is the formula used to estimate the distances appearing after an agglomeration. Therefore, combining our distance estimation with NJ or with BioNJ leads to the same algorithm that we call NJ+TripleML. Our distance estimation can also be combined with Weighbor, and the resulting algorithm is denoted as Weighbor+TripleML. Because Weighbor is a much more complex algorithm than NJ, we only detail NJ+TripleML (fig. 2). The use of TripleML only modifies the initial pairwise distance computation (step 1) and the matrix reduction (step 3). For Weighbor+TripleML, we use the last version of Weighbor available on the Web (version 1.2) and replace its distance estimation by ours. This rather rough approach could likely be improved. For example, the variance and covariance computations could be adapted to our distance estimates.

Time Complexity Analysis

The time complexity of a phylogenetic reconstruction algorithm expresses the computing time it requires, depending on the number *n* and the length *l* of the treated sequences and possibly on some inner parameters of the method. The NJ algorithm consists of two successive steps. During the first step, the initial pairwise distances are estimated in $O(n^2l)$. Then, this matrix is used to infer in $O(n^3)$ a phylogeny through successive agglomerations, so the time complexity of NJ is $O(n^2l + n^3)$. This basically means that computing the distance matrix requires a time proportional to n^2l, whereas building the tree is proportional to n^3.

Using TripleML requires optimization of more likelihood functions but does not change the time complexity of the algorithm. The computation of every initial pairwise distance (step 1 in fig. 2) is done in two steps; the first is in $O(l)$ time complexity and the second in $O(n + l)$. The time required to initialize the distance matrix is thus $O(n^2l + n^3)$. As for NJ, the computing time required for each tree-building stage depends on the number *r* of remaining taxa. Selecting the best pair is done in $O(r^2)$, whereas estimating the new distances is done in $O(rl + r^2)$. The time required for one agglomeration (steps from 3.1 to 3.5) is thus $O(rl + r^2)$, and the time required by the entire tree building procedure (step 3) is then equal to $O(n^2l + n^3)$. So using TripleML raises the time complexity of the distance matrix initialization from $O(n^2l)$ to $O(n^2l + n^3)$ and that of the tree-building stage from $O(n^3)$ to $O(n^2l + n^3)$. But the complexity of the whole approach is globally unchanged and remains $O(n^2l + n^3)$.

Input : n DNA sequences and a model of sequence evolution
Output : A phylogeny of the n sequences
1. Estimate the initial pairwise distances (δ_{ij})
  1.1. For each pair of taxa i,j
    1.1.1. Estimate the distance δ_{ij} using a 2-tree (Equation 7)
  1.2. For each pair of taxa i,j
    1.2.1. Select the taxon k minimizing: $(\delta_{ik}-\delta_{jk})^2+\delta_{ik}\delta_{jk}$
    1.2.2. Re-estimate δ_{ij} using a 3-tree (Equation 8)
2. Initialize the parameter values
  2.1. Initialize the number r of remaining nodes with n
  2.2. Initialize the likelihood vectors of the n taxa
3. While r is greater than 2
  3.1. Select the pair 1,2 to agglomerate (Equation 4)
  3.2. Estimate the two branch lengths δ_{1i} and δ_{2i} (Equation 5)
  3.3. Create the new node i and compute $VL(T_i)$ (Equation 2)
  3.4. Remove 1 and 2; Add i; $r \leftarrow (r-1)$
  3.5. Estimate δ_{ij} for each remaining node j
    3.5.1. For each node j estimate δ_{ij} using Equation (9)
    3.5.2. For each node j
      i Select the node k minimizing: $(\delta_{ik}-\delta_{jk})^2+\delta_{ik}\delta_{jk}$
      ii Re-estimate δ_{ij} using $T_i \cup T_j \cup T_k$ (Equation 10)
4. Extract the last branch length from (δ_{ij})
5. Return the constructed tree.

FIG. 2.—The NJ+TripleML algorithm.

When an iterative process is used to optimize likelihood functions, the number of iterations also influences the computing time required by the method. In our tests, optimization of the single branch length of a 2-tree required about six parabola interpolations and that of a 3-tree required about four passes over the tree. This explains why, although NJ and NJ+TripleML have the same theoretical time complexity, NJ is much faster.

We speed up these optimization steps by compressing the data sets. Two sites with the same value for each studied taxon are said to have identical patterns, and the corresponding likelihood needs only to be computed once. So the first step of phylogenetic reconstruction programs generally consists of searching for sites identical over the whole set of studied taxa, and this compression step is generally done only once at the beginning of the program (Felsenstein 1993). We adapt this approach and search for identical patterns before each likelihood optimization. This significantly decreases the computing time of TripleML. Indeed, all our likelihood computations are done on trees containing only a subset of the studied taxa. When this subset is small—especially during the first steps—numerous sites are identical. For example, during the first step of initial pairwise distance estimation (using a 2-tree), there are at most 16 possible patterns and 64 patterns at most for the second step (using a 3-tree). On the data sets that we used to estimate the computing time of the various methods (table 3), using this improvement makes TripleML from five to eight times faster.

Simulation Results

We first describe the way we generated our data sets and specify the programs we tested. We then measure the influence of TripleML on distance estimation and compare the tested programs on the basis of their topological accuracy and computing time.

Data Sets

Our experimental tests followed a protocol used within a similar framework by Kumar (1996), Gascuel (1997*a*), and Ranwez and Gascuel (2001). Six 12-taxon model trees were considered (fig. 3). The first three (AA, BB, AB) satisfied the molecular-clock hypothesis, whereas the other three (CC, DD, CD) presented substitution rates that vary substantially among lineages. Each interior branch was one unit long (*a* for constant—and *b* for variable—rate trees; the lengths of external branches are given in multiples of *a* or *b*). For each of these model trees, we studied three evolutionary conditions:

- A low evolutionary rate, for which the maximum pairwise divergence (MD) was about 0.1 substitutions per site ($a = 0.00625$ and $b = 0.005$)

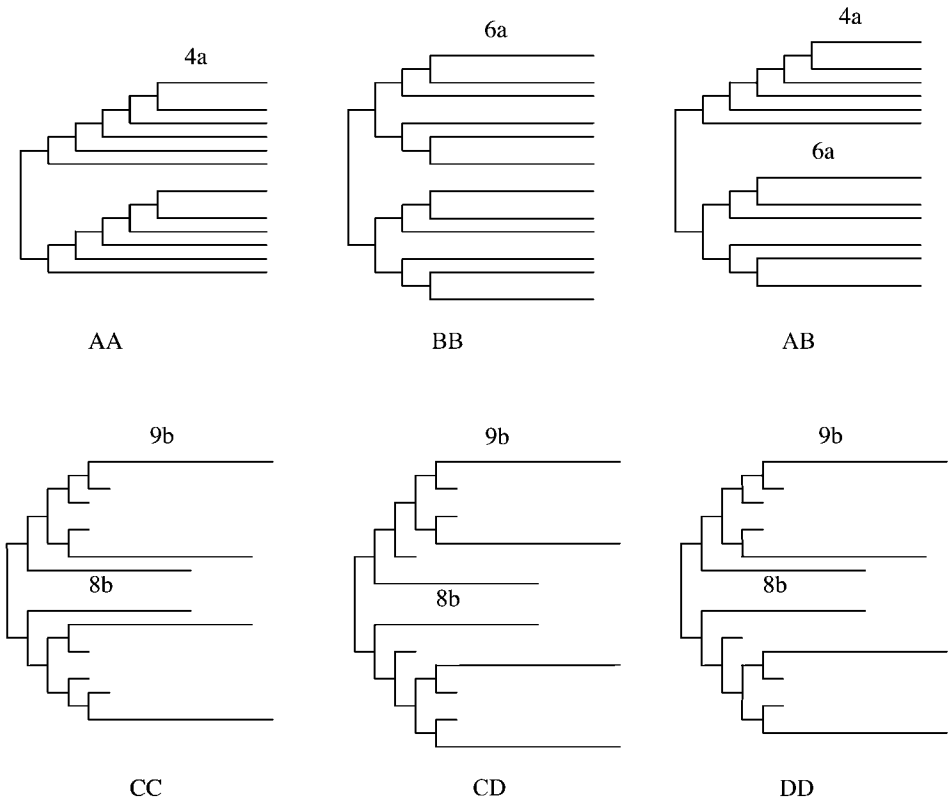

FIG. 3.—Model trees used for simulation. Each interior branch is one unit long (*a* for constant—and *b* for variable—rate trees), and the lengths of external branches are given in multiples of *a* or *b*. Low divergence refers to $a = 0.00625$ substitutions per site and $b = 0.005$, which corresponds to a maximum pairwise divergence (MD) of about 0.1 substitutions per site. Medium evolution divergence refers to $a = 0.0185$ and $b = 0.015$ (MD $\approx$ 0.3), and high divergence refers to $a = 0.0625$ and $b = 0.05$ (MD $\approx$ 1.0).

- A medium evolutionary rate, MD $\approx$ 0.3 per site ($a = 0.0185$ and $b = 0.015$)
- A fast evolutionary rate, MD $\approx$ 1.0 ($a = 0.0625$ and $b = 0.05$).

For each tree *T* and evolutionary condition, we generated 1,000 data files with sequences of length 300. We thus tested the different methods on 18,000 test sets corresponding to three evolution rates and six model trees.

The effectiveness of a phylogenetic reconstruction method depends on the model tree shape, the evolutionary rate, and whether the molecular-clock hypothesis stands. The tests described above allow comparison of the methods according to these different conditions, but they provide a broken up view. Moreover, the evolutionary conditions used for these trees are extreme, which highlights the contrast between the performances of the various methods but is not representative of the data set generally encountered by biologists. Therefore, we also used data sets with 24 sequences of length 600 based on 5,000 random trees. These complementary tests allowed comparison on trees whose internal branch lengths are not all equal and over a wide variety of tree shapes and evolutionary rates.

A true phylogeny, denoted as *T*, was first generated using the stochastic speciation process described by Kuhner and Felsenstein (1994), which corresponds to the usual Yule-Harding distribution on trees (Yule 1925; Harding 1971). The branch length expectation was set at 0.035 mutations per site. Using this generating process makes *T* ultrametric (or molecular-clock–like). This hypothesis does not hold in most biological data sets, so we created a deviation from the molecular clock. Every branch length of *T* was multiplied by $1.0 + \alpha X$, where *X* followed the standard exponential distribution [$P(X > \eta) = e^{-\eta}$], and α was a tuning factor to adjust the deviation from the molecular clock; α was set at 0.8. The average ratio between the mutation rate in the fastest-evolving lineage and the rate in the slowest-evolving lineage was then about 2.0. The smallest (among 5,000) value of this ratio was about 1.2 and the largest 5.0 (1.0 corresponds to the strict molecular clock), whereas the standard deviation was approximately 0.5. The maximum pairwise divergence (MD) ranged from 0.15 to 1.2, with an average of about 0.4.

The random trees were obtained using a software developed by Guindon and Gascuel (2002). We used SEQGEN_1.06 (Rambaut and Grassly 1997) to generate the sequences. For each tree *T*, these sequences were obtained by simulating an evolving process along *T* according to the Kimura two-parameter model with a transition/transversion ratio of 2. The data files are available on our Web page.

Programs Tested

For our tests, we used the latest program versions available on the Web. The different programs were run

with model options corresponding to the Kimura two-parameter model with a transition/transversion ratio of 2. We used default parameter values for other program options.

- We tested three classical distance methods: NJ (Saitou and Nei 1987), BioNJ (Gascuel 1997*a*), and Weighbor version 1.2 (Bruno, Socci, and Halpern 2000). The initial pairwise distance matrix used by these three methods was computed with DNAdist from the PHYLIP package, assuming the Kimura two-parameter model with a transition/transversion ratio of 2. We also provided the sequence lengths for Weighbor, which is the only one of these three methods to consider this information.
- We tested a simple variant of TripleML further denoted as 3Dist. This variant only uses our approach to estimate the initial pairwise distances, and the resulting initial distance matrix can feed into any distance method. In particular, these distances can be used with NJ, BioNJ, and Weighbor, and the resulting methods are denoted as NJ+3Dist, BioNJ+3Dist, and Weighbor+3Dist, respectively.
- We tested NJ+TripleML and Weighbor+TripleML (as explained previously, NJ+TripleML and BioNJ+TripleML are identical).
- We tested a maximum likelihood program: FastDNAml (Olsen et al. 1994). FastDNAml comes from DNAml (Felsenstein 1981) and generally infers the same tree but is much faster.

NJML (Ota and Li 2000) seemed like an interesting approach, and we would have liked to test it, but no version of it was available during this study (unpublished data).

Distance Estimation

Before comparing the topological accuracy of the various methods, we measure the influence of TripleML on distance estimation. The purpose of these tests is to evaluate the improvement resulting from both the initial triplet-based distance estimation and the local maximum likelihood approach that is used during the agglomerative process. To obtain such results, we compared the distances inferred by the distance methods with the distances induced by the true tree *T*, using the 24-taxon data sets.

We first considered the traditional estimation approach for the initial distances (2Dist), which is based on 2-tree likelihood optimization, and our triplet approach (3Dist). In this case, we had to compare the $n(n-1)/2$ nonzero distance estimates in the inferred matrices with the corresponding true pairwise distances.

Then, we measured the accuracy of the new distances inferred during the tree-building process by NJ, BioNJ, Weighbor, and NJ+TripleML. To avoid confounding this measurement with topological accuracy, we modified pair selection to enforce all these methods to reconstruct the correct tree *T*. This constraint makes NJ+TripleML and Weighbor+TripleML nearly identical, so the latter method was not tested. At the first step, all these methods infer $(n-2)$ new distances separating the pair root from the remaining taxa. At the second step, $(n-3)$ new distances are inferred, and the process stops at the last step where only one new distance is inferred. We then compared these $(n-2)+(n-3)+\ldots+1=(n-1)(n-2)/2$ new distances with the corresponding distances in the true tree.

In both cases, we used the ratio of unexplained variance to measure the fitness of the inferred distances. Let (δ_x) be the set of inferred distances and (d_x) the corresponding set of true distances. The ratio of unexplained variance is defined by:

$$\frac{\sum_x (\delta_x - d_x)^2}{\sum_x (d_x - \bar{d})^2} \qquad (13)$$

where $\bar{d}$ is the average of the d_x's. The closer (δ_x) is to (d_x), the smaller is the ratio. Results were averaged over the 5,000 24-taxon data sets.

Regarding initial pairwise distance estimation, the ratio of unexplained variance is about 7.93% for 2Dist and decreases to about 7.83% for 3Dist, whereas regarding the distances estimated during the agglomerative process, the ratio of unexplained variance is about 7.38% for NJ, 7.34% for BioNJ, and 7.33% for Weighbor, but only 6.97% for NJ+TripleML. These tests confirm that the use of a third taxon improves initial distance estimation. But the improvement is much more significant in the following steps, when TripleML reduces the distance matrix using a maximum likelihood approach, whereas NJ, BioNJ, and Weighbor only use distance averaging.

Topological Accuracy

The various reconstruction methods were judged on their ability to infer the correct topology (i.e., that of the tree used to generate the sequences). For tests based on the six model trees, this evaluation was simply done by counting how many times the tree $\hat{T}$ proposed by the method has the same topology as the correct tree *T*. For tests based on random trees, the exact topology is rarely found. This is because some branches are so short that no mutation occurs during simulation along these branches. The topology of $\hat{T}$ was then compared with that of the true tree *T* using a topological distance $d(\hat{T}, T)$ equivalent to that of Robinson and Foulds (1981). This distance is defined by the proportion of internal branches that are found in one tree and not in the other. It varies between 0.0 (both topologies are identical) and 1.0 (they do not share any internal branch). To compare the performance of a method with that of the NJ algorithm, we also measured the relative difference separating its performance from that of NJ. Denoting P_M as the performance of the method M, the relative difference between its topological accuracy and that of NJ corresponds to the ratio $(P_M - P_{NJ})/P_{NJ}$.

The results obtained by the different tested methods for the 5,000 random trees are detailed in table 1. As expected, BioNJ and Weighbor have a better topological accuracy than NJ does. Indeed, the relative difference

Table 1
Results with 5,000 Randomly Generated 24-Taxon Trees

	$d(T, \hat{T})$	Better than NJ	Worse than NJ	Equivalent to NJ
NJ	0.0829			
BioNJ	0.0807 (−2.6%)	12.04%	8.22%	79.74%
Weighbor	0.0784 (−5.5%)	22.10%	14.48%	63.42%
NJ + 3Dist	0.0808 (−2.5%)	13.38%	9.48%	77.14%
BioNJ + 3Dist	0.0787 (−5.1%)	18.80%	11.50%	69.70%
Weighbor + 3Dist	0.0773 (−6.8%)	24.44%	15.42%	60.14%
NJ + TripleML	0.0738 (−11.0%)	28.06%	13.46%	58.48%
Weighbor + TripleML	0.0732 (−11.8%)	31.06%	16.26%	52.68%
FastDNAml	0.0616 (−25.7%)	43.00%	14.12%	42.98%

NOTE.—For NJ, BioNJ and Weighbor, the initial pairwise distance matrix is computed as usual, while NJ + 3Dist, BioNJ + 3Dist, and Weighbor + 3Dist use initial pairwise distances obtained from triplets. NJ + TripleML and Weighbor + TripleML correspond to the combination of our distance estimation method with NJ and Weighbor, respectively. The distance $d(T, \hat{T})$ is defined as the ratio of internal branches wrongly inferred by the methods, and the relative difference between the performance of a method and that of NJ is indicated in parentheses. The last three columns indicate the percentage of data sets for which the distance between the tree inferred by a method is similar, greater, and equal, respectively, to the distance between the true tree and that inferred by NJ.

between the proportion of branches wrongly inferred by BioNJ and NJ is −2.6%, and this difference is −5.5% for Weighbor and NJ. As shown above, using a third taxon to estimate the initial pairwise distances improves their precision. For the three methods, computing the initial distances with 3Dist reduces the proportion of branches wrongly inferred, so that the topological accuracy of NJ+3Dist is equivalent to that of BioNJ, the accuracy of BioNJ+3Dist is close to that of Weighbor, and the relative increase between Weighbor+3Dist and NJ is about −6.8% (vs. −5.5% for Weighbor alone). As further detailed, these improvements are obtained with low additional computing time.

Using the full TripleML approach provides a much greater improvement. Indeed, the relative difference between the proportion of branches wrongly inferred by NJ+TripleML and NJ is −11%, and this difference is −11.8% for Weighbor+Triple and NJ (whereas it is −2.6% for BioNJ and −5.5% for Weighbor). These tests also confirm that the topological accuracy of FastDNAml is far better than that of NJ because the relative difference between FastDNAml and NJ is −25.7%. Therefore, the performance of TripleML combined with NJ (−11%) or with Weighbor (−11.8%) is midway between that of NJ alone and that of FastDNAml.

A question of interest is to know whether the topological accuracy of the tested methods is better than that of NJ for every data set or whether NJ is better on some. We answer this question by measuring for each method the percentage of data sets for which its topological accuracy is better, worse, and equal to that of NJ. These measures are provided in the last three columns of table 1. Although it is clear that all tested methods have a topological accuracy significantly better than that of NJ, these measures show that for numerous data sets, NJ reconstructs a better tree than other methods do and that for most data sets, NJ is as good as other methods. For example, NJ and BioNJ have the same topological accuracy for 80% of the data sets, and NJ is better than FastDNAml for 14% of the data sets. It is thus important to test the performance of the different methods for various tree shapes, and evolutionary conditions, to find the main factors that influence the topological accuracy of the different methods. The study on model trees demonstrates cases where there is no reason to use a method requiring much more computing time than NJ requires and cases where it is worth being patient.

Table 2 gives the percentage of phylogenies correctly reconstructed by the different tested methods, and the relative increase between their topological accuracy and that of NJ, for the six model trees. Results of other methods (parsimony, quartet puzzling, etc.) on the same data sets can be found in Ranwez and Gascuel (2001).

The relative difference between the performance of a method and that of NJ depends on the evolutionary rate. For any method, the higher the evolutionary rate, the greater the difference between its topological accuracy and that of NJ. Under the molecular-clock hypothesis, the difference between NJ and BioNJ or between NJ and Weighbor is only significant for high evolutionary rates (MD $\approx$ 1.0). Under these evolutionary conditions, the relative increase between these two variants and NJ reaches 13%, whereas for MD $\approx$ 0.1, the relative increase between NJ and BioNJ is very low, and Weighbor even has slightly worse results than NJ.

The performance of a method is also related to the shape of the true tree. Any method has a certain tendency to reconstruct chains or balanced trees, depending on whether its reconstruction process is based on iterative taxon insertion plus branch swapping (like FastDNAml) or on agglomeration (like the other tested methods). This phenomenon, studied by Gascuel (2000), explains why the performance of FastDNAml is better for tree AA (chain) than for BB (balanced), whereas the trend is reversed for the other methods.

Yet, the most significant differences between methods are related to molecular clock. BioNJ and Weighbor significantly improve the topological accuracy of NJ when the evolutionary rates vary among lineages but are just slightly better than NJ when the molecular-clock hypothesis stands. For example, with a medium evolu-

Table 2
Percentage of Correct Inference with Model Trees

		Molecular Clock				No Clock			
	Tree	AA	BB	AB	Avg	CC	DD	CD	Avg
MD ≈ 0.1...	NJ	16.4	14.5	14.0	14.97	14.6	13.8	16.4	14.93
	BioNJ	16.6	15.6	14.8	15.67 (+5%)	16.1	14.9	18.5	16.50 (+11%)
	NJ + TripleML	19.3	18.9	18.4	18.87 (+26%)	17.3	17.1	17.1	17.17 (+15%)
	Weighbor	15.5	14.1	13.8	14.47 (−3%)	16.0	15.9	17.9	16.60 (+11%)
	Weighbor + TripleML	17.7	19.0	17.9	18.20 (+22%)	17.6	16.2	18.2	17.33 (+16%)
	FastDNAml	22.6	19.8	21.3	21.23 (+42%)	15.5	17.8	17.8	17.03 (+14%)
MD ≈ 0.3...	NJ	32.2	33.9	31.3	32.47	46.8	50.4	47.5	48.23
	BioNJ	32.7	33.7	33.8	33.40 (+3%)	56.4	56.7	56.3	56.47 (+17%)
	NJ + TripleML	42.1	46.9	42.7	43.90 (+35%)	64.5	62.9	64.0	63.80 (+32%)
	Weighbor	32.8	31.7	33.2	32.57 (+0%)	57.8	60.2	59.2	59.07 (+23%)
	Weighbor + TripleML	40.7	46.0	42.5	43.07 (+33%)	65.0	65.8	68.2	66.33 (+38%)
	FastDNAml	57.6	52.9	54.4	54.97 (+69%)	70.0	67.7	69.4	69.03 (+43%)
MD ≈ 1.0...	NJ	20.1	17.3	18.9	18.77	47.7	48.8	49.4	48.63
	BioNJ	22.4	20.1	21.0	21.17 (+13%)	62.2	63.1	64.6	63.30 (+30%)
	NJ + TripleML	27.0	26.8	27.1	26.97 (+44%)	70.8	68.8	72.0	70.53 (+45%)
	Weighbor	21.4	21.2	20.9	21.17 (+13%)	68.4	71.0	72.2	70.53 (+45%)
	Weighbor + TripleML	24.7	26.4	25.3	25.47 (+36%)	76.1	80.1	77.8	78.00 (+60%)
	FastDNAml	44.0	35.7	37.2	38.97 (+108%)	81.6	83.2	81.2	82.00 (+69%)

Note.—MD, maximum pairwise divergence. NJ + TripleML and Weighbor + TripleML correspond to the combination of our distance estimation method with NJ and Weighbor, respectively. The topological measure is the percentage of correctly inferred trees. Avg is the average percentage of correctly inferred trees over the three model trees respecting (or not respecting) the molecular clock; the number within parentheses indicates the relative increase between the performance of the method considered and that of NJ.

tionary rate (MD ≈ 0.3), the relative increase between BioNJ and NJ is about 17% when rates vary and only about 3% for the molecular clock. Similarly, the relative increase between Weighbor and NJ is about 23% for varying rates and about 0% otherwise.

Using TripleML markedly improves the topological accuracy, and this improvement holds even when the molecular clock stands. For example, with a medium evolutionary rate (MD ≈ 0.3), the relative increase between NJ+TripleML and NJ is about 32% when evolutionary rates vary among lineages and about 35% under molecular clock (whereas under the same conditions, the relative increase for BioNJ drops from 17% to 3%). Similarly, the relative increase between Weighbor+TripleML and NJ is about 38% when evolutionary rates vary among lineages, and 33% under the molecular-clock hypothesis (whereas under the same conditions, the relative increase for Weighbor drops from 23% to 0%). The difference between NJ+TripleML and Weighbor+TripleML reflects the difference between NJ and Weighbor. Their performances are close under the molecular-clock hypothesis, and NJ+TripleML even slightly outperform Weighbor+TripleML for MD ≈ 0.1 and MD ≈ 1.0. For varying rates, Weighbor+TripleML is better adapted than NJ+TripleML. For example, with MD ≈ 0.3, the relative difference between NJ+TripleML and NJ is about 32%, whereas for Weighbor+TripleML, this difference is about 38%.

FastDNAml yields impressive results with the molecular clock. For example, with MD ≈ 0.3, the relative increase between its performance and that of NJ is about 69%. Yet, when rates vary among lineages, its performance is not much better than that of Weighbor+TripleML (e.g., 43% vs. 38% with MD ≈ 0.3).

Thus, irrespective of the tree shape and the evolutionary rate, using TripleML provides methods whose performances are almost midway between the performances of NJ and FastDNAml when the molecular-clock hypothesis stands and are quite close to that of FastDNAml when the evolutionary rates markedly vary among lineages.

Computing Time

To get an idea of the time required by the different methods, we tested them on data sets of various sizes. If we assume two trees AB (with $a = 0.0185$) that have a common ancestor and are linked by a branch of unitary length a, then the resulting tree has 24 taxa. We repeated this procedure twice to construct a 96-tree. For these 24-tree and 96-tree, we generated two data files with sequences of lengths 600 and 1,200, which were obtained using the same process as described above.

Table 3 provides the computing time required by each method depending on the number and length of sequences, using a PC with a 466-MHz processor and a 128-MB RAM. Note that these results are partly specific to our data sets and must therefore only be used to estimate the magnitude of the data that the methods can deal with.

NJ and BioNJ require the same computing time. They are the fastest methods, and most of their computing time is spent computing the initial distance matrix. Conversely, Weighbor performs complicated and extensive computations, and its computing time is significantly longer compared with those of NJ and BioNJ. On the largest data set made of 96 sequences of 1,200 nucleotides, Weighbor requires about 50 s, whereas NJ and BioNJ only require about 10 s.

Using 3Dist only slightly increases the computing time. For the largest data set, NJ or BioNJ+3Dist require about 16 s (instead of 10 s), and the increase is

Table 3.
Computing Times

		$n = 24$	$n = 96$
$l = 600$....	NJ/BioNJ	<1 s	5 s
	Weighbor	1 s	47 s
	NJ/BioNJ + 3Dist	1 s	12 s
	Weighbor + 3Dist	1 s	54 s
	NJ + TripleML	3 s	55 s
	Weighbor + TripleML	3 s	1 min 36 s
	FastDNAml	4 min 45 s	157 min
$l = 1{,}200$ ..	NJ/BioNJ	1 s	9 s
	Weighbor	1 s	52 s
	NJ/BioNJ + 3Dist	1 s	16 s
	Weighbor + 3Dist	2 s	58 s
	NJ + TripleML	5 s	1 min 37 s
	Weighbor + TripleML	5 s	2 min 20 s
	FastDNAml	7 min 40 s	385 min

NOTE.—*n*, number of sequences; *l*, length of sequences. For NJ, BioNJ and Weighbor, the initial pairwise distance matrix is computed as usual, whereas NJ + 3Dist, BioNJ + 3Dist and Weighbor + 3Dist use initial pairwise distances obtained from triplets. NJ + TripleML, and Weighbor + TripleML correspond to the combination of our distance estimation method with NJ and Weighbor, respectively.

not significant with Weighbor, which requires about 1 min with or without 3Dist.

Using TripleML significantly increases the computing time. On the largest data set, NJ+TripleML requires about 1.5 min and Weighbor+TripleML about 2.5 min. But for 96 sequences of 600 nucleotides, NJ+TripleML is close to Weighbor alone. Therefore, using TripleML significantly increases the topological accuracy of distance-based methods while retaining a computing time similar to that of Weighbor.

Despite the difference between their computing times, it thus appears that all methods discussed previously are (relatively) fast and much more suited to very large data sets than is FastDNAml, which already requires more than 6 h for our largest 96-taxon data set.

Conclusions

We have presented a new method for estimating evolutionary distances that we called TripleML. This approach uses the same process for initial pairwise distance estimation and for distance matrix reduction during tree building. All distances are estimated by local maximum likelihood using a third taxon (or cluster) to improve long-distance estimation. Combining TripleML with Weighbor or NJ provides methods whose topological accuracy is much better than that of traditional distance-based methods and is often close to that of the full maximum likelihood approach (as implemented in FastDNAml), while retaining low computing time. We also describe a variant of TripleML, called 3Dist, that only uses our approach to estimate the initial distance matrix. In our tests, combining 3Dist with any distance method significantly increases its topological accuracy, with almost no additional computing time. Moreover, 3Dist does not require any change in the method itself. Using the full TripleML approach provides greater performance improvements, but 3Dist is better adapted for very large data sets containing several hundreds (or thousands) of sequences. NJ+TripleML and 3Dist will be available in the near future at our Web page (http://www.lirmm.fr/w3ifa/MAAS/).

Our results demonstrate that using a third (carefully selected) taxon improves the estimation of large distances. Yet, with very large distances, the use of a third taxon may not be sufficient to obtain branch lengths short enough to be precisely estimated. So a possible development of TripleML could be the use of several well-chosen intermediary taxa to correctly estimate very long distances.

Our results, as those of NJML (Ota and Li 2000), demonstrate the effectiveness of combining distance-based and maximum likelihood approaches. Building and testing other combinations is an important direction for further research.

LITERATURE CITED

BRUNO, W. J., N. D. SOCCI, and A. L. HALPERN. 2000. Weighted neighbor joining: a likelihood-based approach to distance-based phylogeny reconstruction. Mol. Biol. Evol. **17**: 189–197.

FELSENSTEIN, J. 1981. Evolutionary trees from DNA sequences: a maximum likelihood approach. J. Mol. Evol. **17**:368–376.

———. 1993. PHYLIP: phylogeny inference package. Version 3.5c.

GASCUEL, O. 1994. A note on Sattah and Tversky's, Saitou and Nei's, and Studier and Keppler's algorithms for inferring phylogenies from evolutionary distances. Mol. Biol. Evol. **11**:961–963.

———. 1997*a*. BIONJ: an improved version of the NJ algorithm based on a simple model of sequence data. Mol. Biol. Evol. **14**:685–695.

———. 1997*b*. Concerning the NJ algorithm and its unweighted version, UNJ. Pp. 149–170 *in* B. MIRKIN, F. R. MCMORRIS, F. S. ROBERTS, and A. RZHETSKY, eds. Mathematical hierarchies and biology. American Mathematical Society, Providence, R.I.

———. 2000. Evidence for a relationship between algorithmic scheme and shape of inferred trees. Pp. 157–168 *in* W. GAUL, O. OPITZ, and M. SCHADER, eds. Data analysis, scientific modeling and practical applications. Springer, Berlin.

GUINDON, S., and O. GASCUEL. 2002. Efficient biased estimation of evolutionary distances when substitution rates vary across sites. Mol. Biol. Evol. **19**:534–543.

HARDING, E. F. 1971. The probabilities of rooted-tree shapes generated by random bifurcation. Adv. Appl. Probab. **3**:44–77.

KIMURA, M. 1981. Estimation of evolutionary distances between homologous nucleotide sequences. Proc. Natl. Acad. Sci. USA **78**:454–458.

KUHNER, M. K., and J. FELSENSTEIN. 1994. A simulation comparison of phylogeny algorithms under equal and unequal evolutionary rates. Mol. Biol. Evol. **11**:459–468.

KUMAR, S. 1996. A stepwise algorithm for finding minimum evolution trees. Mol. Biol. Evol. **13**:584–593.

LANAVE, C., G. PREPARATA, C. SACCONE, and G. SERIO. 1984. A new method for calculating evolutionary substitution rates. J. Mol. Appl. Genet. **20**:86–93.

NEI, M. 1991. Relative efficiencies of different tree-making methods for molecular data. Pp. 90–128 *in* M. M. MIYAMOTO and J. CRACRAFT, eds. Phylogenetic analysis of DNA sequences. Oxford University Press, Oxford.

OLSEN, G. J., H. MATSUDA, R. HAGSTROM, and R. OVERBEEK. 1994. FastDNAml: a tool for construction of phylogenetic trees of DNA sequences using maximum likelihood. Comput. Appl. Biosci. **10**:41–48.

OTA, S., and W. H. LI. 2000. NJML: a hybrid algorithm for the neighbor-joining and maximum-likelihood methods. Mol. Biol. Evol. **17**:1401–1409.

PRESS, W. H., B. P. FLANNERY, S. A. TEUKOLSKY, and W. T. VETTERLING. 1988. Numerical recipes in C: the art of scientific computing. Cambridge University Press, Cambridge, U.K.

RAMBAUT, A., and N. C. GRASSLY. 1997. Seq-Gen: an application for the Monte Carlo simulation of DNA sequence evolution along phylogenetic trees. Comput. Appl. Biosci. **13**:235–238.

RANWEZ, V., and O. GASCUEL. 2001. Quartet-based phylogenetic inference: improvements and limits. Mol. Biol. Evol. **18**:1103–1116.

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

SAITOU, N., and M. NEI. 1987. The neighbor-joining method: a new method for reconstructing phylogenetic trees. Mol. Biol. Evol. **4**:406–425.

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

STRIMMER, K., and A. VON HAESELER. 1996. Quartet puzzling: a quartet maximum-likelihood method for reconstructing tree topologies. Mol. Biol. Evol. **13**:964–969.

STUDIER, J. A., and K. J. KEPPLER. 1988. A note on the neighbor-joining algorithm of Saitou and Nei. Mol. Biol. Evol. **5**:729–731.

SWOFFORD, D. L., G. J. OLSEN, P. J. WADDELL, and D. M. HILLIS. 1996. Phylogenetic inference. Pp. 407–509 *in* M. D. HILLIS, C. MORITZ, and B. K. MABLE, eds. Molecular systematics. Sinauer Associates, Sunderland, Mass.

YANG, Z. 1993. Maximum-likelihood estimation of phylogeny from DNA sequences when substitution rates differ over sites. Mol. Biol. Evol. **10**:1396–1401.

YANG, Z., N. GOLDMAN, and A. FRIDAY. 1994. Comparison of models for nucleotide substitution used in maximum-likelihood phylogenetic estimation. Mol. Biol. Evol. **11**:316–324.

YULE, G. 1925. A mathematical theory of evolution, based on the conclusions of Dr. J. C. Willis. Philos. Trans. R. Soc. Lond. B Biol. Sci. **213**:21–87.

MANOLO GOUY, reviewing editor

Accepted July 16, 2002