

# Quartet-Based Phylogenetic Inference: Improvements and Limits

Vincent Ranwez and Olivier Gascuel

Département Informatique Fondamentale et Applications, Montpellier, France

We analyze the performance of quartet methods in phylogenetic reconstruction. These methods first compute four-taxon trees (4-trees) and then use a combinatorial algorithm to infer a phylogeny that respects the inferred 4-trees as much as possible. Quartet puzzling (QP) is one of the few methods able to take weighting of the 4-trees, which is inferred by maximum likelihood, into account. QP seems to be widely used. We present weight optimization (WO), a new algorithm which is also based on weighted 4-trees. WO is faster and offers better theoretical guarantees than QP. Moreover, computer simulations indicate that the topological accuracy of WO is less dependent on the shape of the correct tree. However, although the performance of WO is better overall than that of QP, it is still less efficient than traditional phylogenetic reconstruction approaches based on pairwise evolutionary distances or maximum likelihood. This is likely related to long-branch attraction, a phenomenon to which quartet methods are very sensitive, and to inappropriate use of the initial results (weights) obtained by maximum likelihood for every quartet.

## Introduction

The maximum-likelihood method (Felsenstein 1981) is widely used to infer molecular phylogenies. It has sound statistical foundations and performs well in computer simulations. Unfortunately, the computing time quickly becomes unacceptable as the number of taxa ( $n$ ) increases. When  $n$  is greater than about a dozen taxa, it is much faster to use maximum-likelihood to analyze each subset of four taxa than to use it to directly infer the evolutionary history of all these taxa. By combining four-taxon phylogenies (4-trees) obtained by maximum likelihood, quartet methods try to tap the strength of maximum likelihood within reasonable computing time.

The simplest quartet approach involves keeping for each group of four taxa (quartet) the most likely 4-tree and then seeking the complete phylogeny which contains as many of these 4-trees as possible. Methods based on this principle (Bandelt and Dress 1986; Jiang, Kearney, and Li 1998; Berry and Gascuel 2000) are interesting due to their good theoretical properties. They do not, however, account for the fact that some 4-trees are more reliable than others, and they implicitly give the same importance to all inferred 4-trees.

Several different approaches have been described for incorporating reliability into quartet-based phylogenetic analysis. In Erdős et al. (1997), 4-trees found to be unreliable are rejected and not used in phylogeny reconstruction. In Berry et al. (1999), two different 4-trees on the same quartet are used when a single 4-tree is unreliable. Finally, methods like quartet puzzling (QP) (Strimmer, Goldman, and von Haeseler 1997) and that of Willson (1999a) preserve the three possible 4-trees for each quartet and associate with them a weight proportional to the confidence they have in them. By using maximum likelihood to balance 4-trees, these algorithms

seem to follow a reasonable approach for inferring a tree having a high likelihood on  $n$  taxa.

However, inference of phylogenies containing only four taxa is sometimes contested, which questions the very principle of quartet methods. The sampling of taxa preceding phylogenetic reconstruction strongly influences the inferred phylogeny. If one seeks to establish the phylogenetic relationship between four groups of taxa by using a single representative for each of these groups, the result generally depends on which representatives are selected. This problem was studied by Philippe and Douzery (1994), who, based on 4-trees inferred by a parsimony method, concluded: "Reconstructing history with only four taxa is rather a game of chance." Adachi and Hasegawa (1999) extended this result to 4-trees inferred by maximum likelihood and concluded: "As Philippe and Douzery showed, it is now clear that an argument based on a quartet analysis of a single gene is very dangerous." However, it seems possible to overcome this difficulty, since in published simulations (Strimmer and von Haeseler 1996; Strimmer, Goldman, and von Haeseler 1997) the performance of QP is close to or even better than that of maximum likelihood. This gives rise to the following questions: Is it possible to improve QP? If so, how do these new quartet methods perform compared with other classical phylogenetic reconstruction methods?

In this article, we present weight optimization (WO), a new algorithm which also uses weighted 4-trees inferred by maximum likelihood. WO searches for the tree on  $n$  taxa such that the sum of the weights of the 4-trees induced by this tree is maximal. Finding the optimal tree in this sense is computationally difficult (Steel 1992), like most optimization tasks in phylogenetic inference (Swofford et al. 1996), and thus WO is based on heuristic approach which only guarantees finding a near-optimal tree. However, we shall see that its performance in optimizing this criterion is good enough and that better optimization does not improve the topological accuracy. We start by describing QP, and then we describe and demonstrate the properties of WO. Using computer simulations, we then compare the topological accuracies of QP and WO with those of other common phylogenetic reconstruction methods. WO significantly

Key words: phylogenetic reconstruction, quartet methods, tree consensus, maximum likelihood, parsimony, distance methods, computer simulations.

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.* 18(6):1103–1116, 2001

© 2001 by the Society for Molecular Biology and Evolution. ISSN: 0737-4038

improves QP, but, as discussed, the performance of these quartet methods is rather disappointing. Finally, we describe factors which seem to limit the topological accuracy of quartet methods. The implementation of WO, which uses tools of general interest in phylogenetic reconstruction, is described in the appendix.

## Methods

First, we define the terms employed in this article and the corresponding notation. Then, we describe QP and WO and study their properties.

### Definitions and Notation

A tree is a set of nodes connected by branches (or edges) so that there is a single path connecting two nodes. The degree of a node is the number of branches attached to it. The nodes with degree 1 (the leaves) are labeled by taxa; other nodes are called internal nodes. If all internal nodes have degree 3, then we say that the tree is fully resolved; if some have a higher degree, we say that the tree is partially resolved.

Removing a branch in a phylogeny separates taxa into two disjoint subsets such that their union equals the complete set of taxa. The branch is said to define the bipartition (or split) formed from the two subsets. When a bipartition separates two taxa from all the others, the pair of taxa is called an external pair. Let  $xy|zt$  denote the 4-tree that separates taxa  $x$  and  $y$  from taxa  $z$  and  $t$ . A quartet is a set of four taxa; for each quartet  $\{x, y, z, t\}$ , there are three possible 4-trees:  $xy|zt$ ,  $xz|yt$ , and  $xt|yz$ . When an  $n$ -tree  $T$  contains a bipartition which separates taxa  $x$  and  $y$  from taxa  $z$  and  $t$ ,  $T$  is said to induce the 4-tree  $xy|zt$ , and the 4-tree  $xy|zt$  is said to be consistent with  $T$ .

Let the pair  $(q, w)$  denote the weighted 4-tree ( $w4$ -tree) which associates a weight  $w$  with the 4-tree  $q$ . Let  $Q$  denote the set of weighted 4-trees used as the starting point for QP or WO.  $Q$  contains three  $w4$ -trees for each quartet.  $Q_{\max}$  is the set induced by  $Q$  which, for each quartet, contains only the 4-tree having the maximum weight (among three). In the unweighted case, for each quartet, one 4-tree has weight 1 and the two others have weight 0. The set of 4-trees induced by a tree  $T$  is denoted  $Q_T$ .

After assigning weights to the 4-trees, quartet methods search for the tree  $T$  which fits them best. A natural measurement of this fit is the sum of the weights of 4-trees induced by  $T$ . Thus, we search for the tree  $T$  which maximizes the  $W$  criterion, defined as

$$W(T) = \sum_{\substack{q \in Q_T \text{ and} \\ (q, w) \in Q}} w. \quad (1)$$

However, finding  $T$  defined as such is NP-hard (Steel 1992) and can require an exponentially long computing time. Therefore, we have to use heuristic algorithms (such as QP or WO), to find a near-optimal tree within a reasonable amount of computing time.

### Quartet Puzzling

To infer the phylogeny of  $n$  sequences, QP proceeds in three stages: (1) it uses the maximum-likelihood principle to weight all possible 4-trees; (2) based on these weights, it constructs a large number of  $n$ -trees; and (3) it computes the consensus tree of these  $n$ -trees.

Given a quartet, the likelihoods  $l_1$ ,  $l_2$ , and  $l_3$  of the three associated 4-trees are used to estimate their respective probabilities  $p_1$ ,  $p_2$ , and  $p_3$  of being the correct 4-tree. These probabilities are evaluated using Bayes theorem as (Strimmer, Goldman, and von Haeseler 1997)

$$p_i = \frac{l_i}{l_1 + l_2 + l_3}. \quad (2)$$

Strimmer, Goldman, and von Haeseler (1997) propose three different ways to use these probabilities to weight 4-trees. In the continuous case, the probabilities are directly used as weights, i.e.,  $w_i = p_i$ . In the binary (unweighted) case, the weight of the 4-tree with highest probability is set at 1, and the weights of the two others are set at 0. In the discrete case, the three  $w_i$ 's are discrete, least-squares approximations of the  $p_i$ 's. Assuming that  $p_1 \geq p_2 \geq p_3$ , the weights ( $w_1, w_2, w_3$ ) are approximated by  $(1, 0, 0)$ ,  $(1/2, 1/2, 0)$ , or  $(1/3, 1/3, 1/3)$ . For example, if  $p_1 = 0.5$ ,  $p_2 = 0.45$ , and  $p_3 = 0.05$ , then  $w_1 = 1/2$ ,  $w_2 = 1/2$ , and  $w_3 = 0$ , indicating that it is not clear which of the first two 4-trees is the correct one but that the third one is certainly incorrect.

The next stage uses these weights on 4-trees to construct a collection of  $n$ -trees. To construct a single  $n$ -tree, QP follows an addition scheme. It first randomly determines an order among the taxa. The taxa are then added one by one to a partially constructed tree until a complete  $n$ -tree containing all taxa is obtained. Suppose that the taxa are denoted as  $1, 2, \dots, n$ , and that the addition order is  $(1, 2, 3, \dots, n)$ . The 4-tree of maximum weight among the three which resolve  $\{1, 2, 3, 4\}$  is used as starting point. At step  $i$ , the current tree contains the taxa from  $S_i = \{1, 2, 3, \dots, i-1\}$ , and QP seeks to add the taxon  $i$ . To determine the branch on which taxon  $i$  is added, QP proceeds as follows. First, with each branch of the current partially constructed tree, it associates a penalty which is initialized to 0. Then, it considers all  $w4$ -trees of type  $(xi|yz, w)$  with  $x, y$ , and  $z$  in  $S_i$ , which are the  $w4$ -trees "relevant" to the addition of  $i$ . For each of these  $w4$ -trees, QP adds the penalty  $w$  to branches of the path connecting  $y$  to  $z$ . The taxon  $i$  is then added onto the least penalized branch.

Figure 1 shows how, starting from a set of  $w4$ -trees, QP determines the branch of  $12|34$  on which taxon 5 is added. For example, to take the  $w4$ -tree  $(51|23, 0.4)$  into account, QP adds a penalty of 0.4 on branches belonging to the path  $(2, 3)$ . Indeed, if taxon 5 is added on one of these branches, the 4-tree  $51|23$  is not induced by the inferred tree. Note, however, that adding taxon 5 on the branch connected to taxon 4 also leads to a tree construction inconsistent with  $51|23$  and that QP does not penalize this branch.

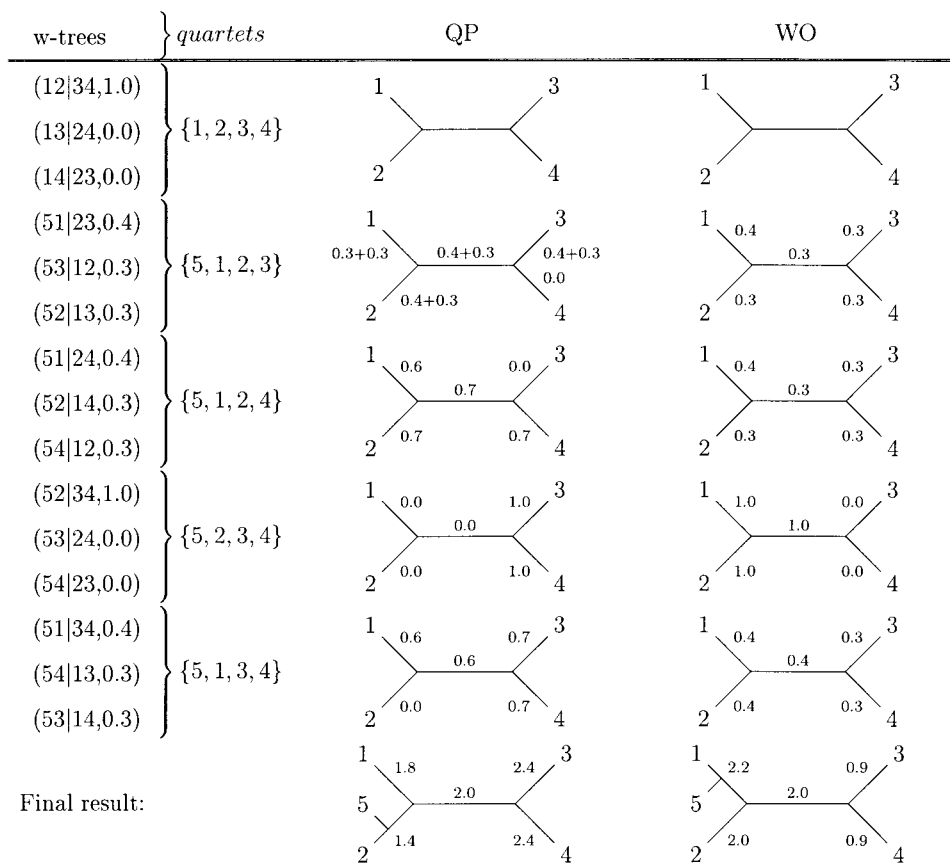


FIG. 1.—WO and QP (based on continuous weighting) reconstruction procedure on a simple example with five taxa. It is assumed that the initial 4-tree is 12|34 for both algorithms. The w4-trees relevant for the addition of taxon 5 are grouped according to the quartet they resolve. For each quartet, the w4-tree of maximum weight is the first one. The final result provides the score of each branch, which corresponds to the sum of the scores set for each of the 12 relevant w4-trees. This score indicates the branch on which taxon 5 is added. WO, unlike QP, reconstructs the correct tree.

This basic principle is altered according to the weighting scheme. In the binary case, only weights of 1 are reported in the tree, because weights of 0 do not have any influence. In the discrete case, QP randomly selects one of the likely 4-trees associated with a given quartet and then uses the binary procedure. For example, if weights are (1/2, 1/2, 0), then some *n*-trees will be reconstructed based on the assumption that the correct 4-tree is the first one, and some others will be reconstructed on the assumption that it is the second one.

The *n*-tree that is constructed varies according to the taxon addition order. In order to overcome this problem, QP randomly defines several addition orders, constructs the corresponding *n*-trees, and returns their consensus as final result. Strimmer and von Haeseler (1996) recommend that trees be constructed using as many different orders as possible and suggest that 1,000 is a reasonable value. Several consensus tree definitions exist. The current version of QP uses the “majority rule” consensus tree of Margush and McMorris (1981), whose bipartitions appear in more than half of the *n*-trees. Usually this consensus tree is partially resolved, and the simulation tests performed by Strimmer and von Haeseler (1996) and Strimmer, Goldman, and von Haeseler (1997) were done (personal communication) using the CONSENSE program from the PHYLIP package (Fel-

senstein 1989). This program usually provides a fully resolved tree with relatively high branch supports which systematically contains bipartitions of the majority-rule consensus tree.

### Weight Optimization

WO uses 4-tree weighting to dynamically define the taxon addition order. At each step, the selected taxon is added so as to generate the greatest increase of the *W* criterion (eq. 1). This does not guarantee that the optimal *n*-tree will be found, but this kind of “greedy” approach has proven effective in many optimization problems (Cormen, Leiserson, and Rivest 1992, pp. 329–356).

WO also uses continuous weighting based on likelihood, as introduced by (Strimmer, Goldman, and von Haeseler 1997) and described above. WO gives branches bonuses rather than penalizing them. This new formulation has the advantage of clarifying the link with the *W* criterion. When adding taxon *i*, WO chooses the branch giving the tree with the largest *W* score (eq. 1). To this end, WO considers every relevant w4-tree (*ix|yz*, *w*) and adds a bonus *w* to each branch of the current tree, such that the addition of *i* on this branch constructs a tree inducing *ix|yz*. Once all relevant w4-trees have been taken into account, the bonus of any

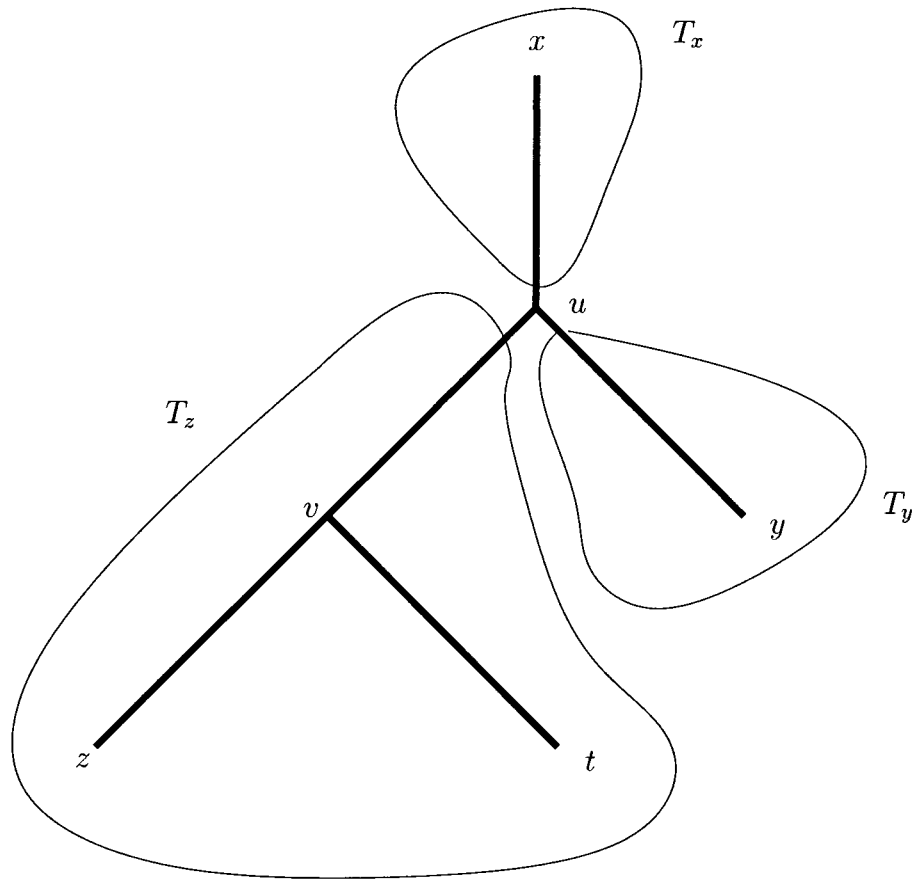


FIG. 2.—The median node  $u$  of  $x$ ,  $y$ , and  $z$  splits the tree into three subtrees,  $T_x$ ,  $T_y$ , and  $T_z$ .

branch  $b$  with the addition of taxon  $i$  is equal to the increase  $\delta W(b, i)$  of the  $W$  criterion induced by this addition. For the w4-tree  $(ix|yz, w)$ , branches receiving a bonus are determined as follows. There is a single internal node belonging to the three paths  $(x, y)$ ,  $(x, z)$ , and  $(y, z)$  called the median of  $x, y, z$  (Barthélemy and Guénoche 1990, pp. 57). This internal node is attached to three branches and thus defines three disjointed subtrees denoted as  $T_x, T_y$ , and  $T_z$ , such that  $x \in T_x, y \in T_y$ , and  $z \in T_z$  (fig. 2). If taxon  $i$  is added on a branch of  $T_x$ , then the path  $(i, x)$  has its branches within  $T_x$ , while the path  $(y, z)$  has all its branches within  $T_y \cup T_z$ . This ensures that paths  $(i, x)$  and  $(y, z)$  do not intersect and, consequently, that the constructed tree induces  $ix|yz$ . WO thus adds the bonus  $w$  onto branches of  $T_x$ , which is equivalent to adding a penalty  $w$  to the branches of  $T_y \cup T_z$ . However, this is very different from giving a penalty only to the branches of the path  $(y, z)$  as is the case with QP.

Figure 1 shows how, starting from a set of w4-trees, WO determines the branch of  $12|34$  on which taxon 5 must be added. To take the w4-tree  $(51|23, 0.4)$  into account, WO puts a single 0.4 bonus on the branch connected to taxon 1. Indeed, this is the only branch for which the addition of taxon 5 generates a tree inducing  $51|23$ . Note that by adding no bonus to the branch connected to taxon 4, WO, contrary to QP, penalizes this branch.

WO randomly selects three taxa (denoted as 1, 2, and 3) to initialize the tree  $T$  and then iteratively adds the remaining taxa. Unlike with QP, the taxon addition order is not random. Instead, at each step, we add the taxon providing the “safest” addition. The safety  $s(i)$  of the addition of  $i$  is defined as follows. Let  $M$  denote the branch with the highest bonus and  $m$  the branch giving the second highest bonus. Let  $\delta W(M, i)$  and  $\delta W(m, i)$  denote the increases in  $W(T)$  resulting from adding  $i$  onto branch  $M$  or  $m$ . We have

$$s(i) = \frac{\delta W(M, i) - \delta W(m, i)}{\delta W(M, i) + \delta W(m, i)}. \quad (3)$$

The greater the difference between  $\delta W(M, i)$  and  $\delta W(m, i)$ , the more contrasted the certainties relative to  $M$  and  $m$ , and the safer the addition of  $i$  onto  $M$ . This definition of safety differs from that of Willson (1999a), which uses  $s(i) = \delta W(M, i) - \delta W(m, i)$ . Unpublished tests show that, in the present case, it is preferable to normalize this value between 0 and 1, as in the above definition.

In order to completely define the addition order, we could alternatively select the 4-tree of maximum weight as the starting point. This requires examining all 4-trees, which is time- and memory-space-consuming, and simulations indicate that this does not improve the topological accuracy of WO. The simpler solution presented

here ensures that the starting 4-tree is a good one, since it is the best among the  $3 \times (n - 3)$  4-trees that resolve a quartet  $\{1, 2, 3, x\}$ , with  $x$  differing from 1, 2, and 3.

### Differences Between WO and QP

Let  $T$  denote the correct tree. If the weighting of the 4-trees is accurate enough to obtain  $Q_{\max} = Q_T$ , i.e., if for all quartets the correct 4-tree has a weight superior to that of the two others, then WO reconstructs the correct tree  $T$  with certainty. This property also holds for QP in the binary case (Strimmer and von Haeseler 1996), but it no longer holds when a discrete or continuous weighting is used. Moreover, WO constructs a single tree, whereas our simulations confirm that QP needs to construct a large number of trees to achieve satisfactory performance. As we shall explain, this not only makes WO faster than QP, but also saves memory space and allows us to deal with larger data sets.

The proof that WO infers the correct tree  $T$  once  $Q_{\max} = Q_T$  is achieved by proving that at each step of WO, the current tree is a subtree tree of  $T$ . This property is true for the 3-tree used as a starting point. Assuming that this property is true at the current step, we simply have to prove that for every taxon, adding it on the branch with maximum bonus for this taxon infers a subtree of  $T$ , which guarantees that the property is still true at the next step. Since the current tree is a subtree of  $T$ , it possesses a branch such that adding  $i$  onto it leads to a subtree of  $T$ . The bonus of this branch is the sum of the weights of the newly induced 4-trees, which are all correct. The bonus of any other branch is the sum of the weights of the newly induced 4-trees, but at least one of them is not correctly resolved. These two sums concern weights which are bijectively associated, depending on the relevant quartet to which they correspond. Based on our hypothesis that  $Q_{\max} = Q_T$ , this implies that the highest sum indicates the correct branch to add  $i$ , and thus recursively proves that at each WO step the current tree is a subtree tree of  $T$ .

Figure 1 illustrates a case in which QP (based on continuous weighting) fails to infer the correct phylogeny, although  $Q_{\max} = Q_T$ . In this example, QP and WO are used to infer the phylogeny of taxa 1, 2, 3, 4, and 5. Moreover, we assume that 12|34 is used as the starting point by both algorithms. The way QP and WO then use the 4-tree weighting to add taxon 5 is detailed in figure 1. After having taken all relevant w4-trees into account, WO adds taxon 5 on the branch connected to taxon 1 and thus infers the unique tree  $T$  such that  $Q_{\max} = Q_T$ . Moreover, as shown above, WO reconstructs this tree  $T$  irrespective of the three taxa used to initialize the tree. On the other hand, QP adds taxon 5 on the branch connected to taxon 2 and does not infer this tree. In this example, the discrete approximation of a 4-tree weight different from 0 and 1 is always 1/3. Thus, for each quartet, the discrete-weight version of QP has one chance in three to select the correct 4-tree, and, depending on these random selections, it may or may not reconstruct the correct phylogeny. Finally, the only consistent version of QP is that using binary weighting.

The time complexity of a phylogenetic reconstruction algorithm expresses the computing time it requires, depending on the number of treated taxa (and possibly other parameters). QP and WO both have  $O(n^4)$  complexity. In other words, their computing time increases proportionally to the fourth power of the number of taxa. In fact, QP as described by Strimmer and von Haeseler (1996) has  $O(n^5)$  complexity. However, both QP and WO can be implemented in  $O(n^4)$  by gathering the w4-trees which modify the bonuses (or penalties) of the same branches and by reusing calculations already carried out during previous steps. Although they have the same theoretical complexity, WO is faster than QP because it constructs a single tree, whereas QP constructs numerous trees (1,000 per default). This difference increases when the number of taxa grows, since the authors of QP say that “generally, the more taxa are involved the more runs of the puzzling step are advised.” However, WO remains slower than any method having  $O(n^3)$  complexity, such as distance algorithms like neighbor joining (NJ) (Saitou and Nei 1987) or BIONJ (Gascuel 1997). The  $O(n^4)$  implementation of WO is detailed in the appendix.

Space complexity expresses the memory space required according to the number of treated taxa. In the algorithm shown in figure 3, the size of the input data is  $O(n^4)$ , and the memory space required is thus  $O(n^4)$ , as for most quartet algorithms. However, it is possible to modify quartet methods in order to input DNA sequences and to compute the weight of a 4-tree each time it is needed. This decreases the memory space required but increases the computation time as soon as weights are needed more than once. QP needs to consult each weight at least 1,000 times (one per reconstructed tree). For such methods, recomputing the weights instead of storing them is not realistic. However, this approach is better adapted in our case, since in the algorithm shown in figure 3, the weight of a 4-tree is consulted only once. Denoting  $l$  as the length of DNA sequences, the memory space required then becomes  $O(nl + n^2)$ . This relatively low space complexity allows us to deal with larger data sets than quartet methods that need to store  $O(n^4)$  weights.

### Simulation Results

We start by describing how we generated our test sets and which criteria were used to compare the performances of the various phylogenetic reconstruction programs. We then compare the results obtained by the different methods.

#### Protocol

Our experimental tests followed a protocol used within a similar framework by Kumar (1996), and later by Gascuel (1997). Six model trees were considered, each consisting of 12 taxa (fig. 4). The first three (AA, BB, AB) satisfied the molecular-clock hypothesis, while the other three (CC, DD, CD) presented varying substitution rates among lineages. Each interior branch was one unit long ( $a$  for constant- and  $b$  for variable-rate

```

input : A set  $Q$  of w4-trees corresponding to the  $n$  taxa
output: A phylogeny of this  $n$  taxa

randomly select three taxa 1, 2 and 3;
 $last \leftarrow 3$ ;

initialize  $T$  with the tree (1, 2, 3);
 $R \leftarrow \{4, 5, \dots, n\}$ ;

while  $R \neq \emptyset$  do
  foreach taxon  $i$  of  $R$  do
    reinitialize branch weights to 0;
    foreach internal node of  $T$  splitting  $T$  in  $T_{last} \ni last, T_x, T_y$  do
      (1) foreach  $(x, y) \in T_x \times T_y$  do
        add the weight of  $(ix|y_{last})$  to  $S_{i,T_x}$ ;
        add the weight of  $(iy|x_{last})$  to  $S_{i,T_y}$ ;
        add the weight of  $(i_{last}|xy)$  to  $S_{i,T_{last}}$ ;
      (2) foreach subtree  $T_z \in \{T_x, T_y, T_{last}\}$  do
        add  $S_{i,T_z}$  to the bonus of each branch of  $T_z$ ;
    memorize the best branch for taxon  $i$  and its safety;
   $last \leftarrow$  the taxon providing the safest addition;
  add  $last$  on its best edge and remove it from  $R$ ;
return( $T$ );

```

FIG. 3.—The weight optimization (WO) algorithm.

trees; the lengths of external branches are given in multiples of  $a$  or  $b$ ). For each of these model trees, we studied four evolutionary conditions:

- a low evolutionary rate, for which the maximum pairwise divergence (MD) was about 0.1 substitutions per site ( $a = 0.00625$  and  $b = 0.005$ );
- 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  $b = 0.05$ );
- a very fast evolutionary rate, MD  $\approx 2.0$  ( $a = 0.125$  and  $b = 0.1$ ).

For each tree  $T$  and evolutionary condition, we used SEQGEN (Rambaut and Grassly 1997) to generate 1,000 data files with sequences of length 300, and 1,000 data files with sequences of length 600. These sequences were obtained by simulating an evolving process along  $T$  according to the Kimura two-parameter model with a

transition/transversion rate of 2. We thus tested the different methods on 48,000 test sets corresponding to two sequence lengths, four evolution rates, and six model trees (the data files are available on our web page at <http://www.firmm.fr/~w3ifa/MAAS/>).

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). Generally, this evaluation is made either by counting how many times the tree  $\hat{T}$  proposed by the method has the same topology as the correct tree  $T$  or by considering a topological distance  $d(\hat{T}, T)$  between the inferred tree and the correct one. We used both criteria. The first penalizes methods which propose not-fully-resolved trees, since such trees always differ from the correct one. The second avoids this drawback but requires choosing a measurement of the difference between two tree topologies, which could favor one reconstruction method over another. For our tests, we used the bipartition distance introduced by

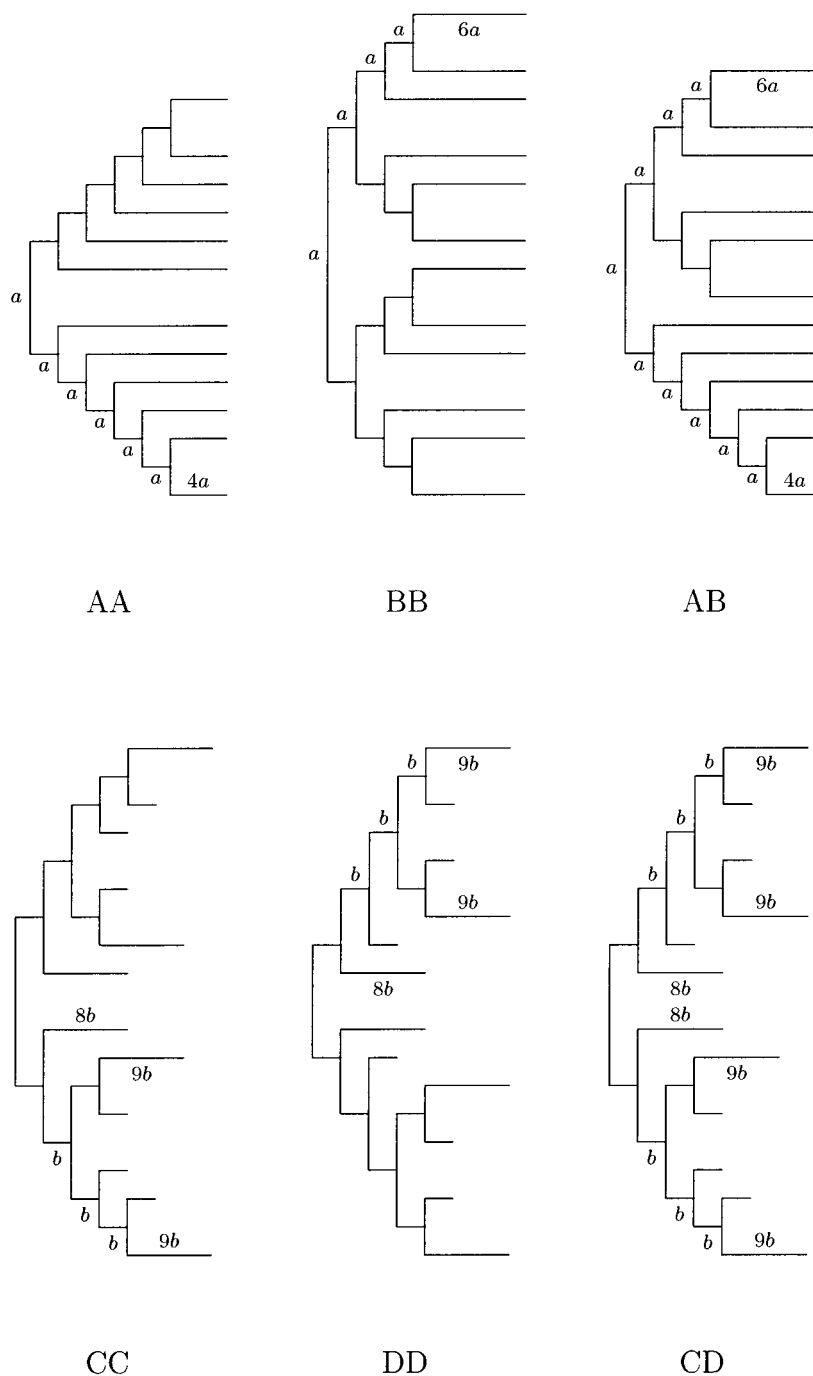


FIG. 4.—Model trees used for simulation. Each interior branch is one unit long ( $a$  for constant- and  $b$  for variable-rate trees), and the lengths of external branches are given in multiples of  $a$  or  $b$ . Low divergence refers to  $a = 0.00625$  substitutions per site and  $b = 0.005$ , which corresponds to a maximum pairwise divergence (MD) of about 0.1 substitutions per site. Medium divergence refers to  $a = 0.0185$  and  $b = 0.015$  ( $MD \approx 0.3$ ), high divergence refers to  $a = 0.0625$  and  $b = 0.05$  ( $MD \approx 1.0$ ), and very high divergence refers to  $a = 0.125$  and  $b = 0.1$  ( $MD \approx 2.0$ ).

Robinson and Foulds (1981), which is equal to the number of bipartitions present in one of the two trees and not the other. This is the topological distance used in most simulation studies.

Methods Tested

For our tests, we used the 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. The following programs were tested:

1. The most recent version (4.2) of quartet puzzling, denoted QP<sub>4.2</sub>. This version of QP is based on discrete weighting. By default, this program constructs

**Table 1**  
**Percentages of Correctly Inferred Trees (sequence length 300)**

TREE	MOLECULAR CLOCK				NO CLOCK				
	AA	BB	AB	Avg. <sup>a</sup>	CC	DD	CD	Avg. <sup>a</sup>	
MD ≈ 0.1 ...	QP <sub>4.2</sub>	1	3	2	2 (7)	3	3	4	3 (7)
	QP <sub>B</sub>	9	12	11	11 (5)	12	10	13	12 (5)
	QP <sub>C</sub>	4	22	10	12 (4)	12	11	12	11 (6)
	WO	10	9	12	10 (6)	12	12	15	13 (4)
	DNAPARS	19	16	15	17 (2)	13	15	17	15 (3)
	BIONJ	17	16	15	16 (3)	16	15	18	16 (2)
	FASTDNAML	23	20	21	21 (1)	16	18	18	17 (1)
	QP <sub>4.2</sub>	4	14	7	9 (7)	18	24	21	21 (7)
MD ≈ 0.3 ...	QP <sub>B</sub>	26	37	30	31 (3)	44	45	44	44 (5)
	QP <sub>C</sub>	12	48	24	28 (6)	36	39	38	38 (6)
	WO	30	29	32	30 (4)	46	47	48	47 (4)
	DNAPARS	28	36	26	30 (5)	46	53	51	50 (3)
	BIONJ	33	34	34	33 (2)	56	57	56	56 (2)
	FASTDNAML	58	53	54	55 (1)	70	68	69	69 (1)
	QP <sub>4.2</sub>	0	3	1	2 (7)	17	26	22	22 (6)
	QP <sub>B</sub>	13	23	19	18 (4)	46	50	48	48 (4)
MD ≈ 1.0 ...	QP <sub>C</sub>	4	29	13	15 (5)	33	41	37	37 (5)
	WO	23	16	20	20 (3)	53	56	56	55 (3)
	DNAPARS	6	8	6	7 (6)	7	9	10	9 (7)
	BIONJ	22	20	21	21 (2)	62	63	65	63 (2)
	FASTDNAML	44	36	37	39 (1)	82	83	81	82 (1)
	QP <sub>4.2</sub>	0	0	0	0 (7)	1	3	1	2 (6)
	QP <sub>B</sub>	1	4	2	2 (4)	13	17	12	14 (4)
	QP <sub>C</sub>	0	4	2	2 (3)	7	11	8	8 (5)
MD ≈ 2.0 ...	WO	2	1	2	2 (5)	19	25	19	21 (3)
	DNAPARS	0	0	0	0 (6)	0	0	0	0 (7)
	BIONJ	2	3	3	3 (2)	29	31	33	31 (2)
	FASTDNAML	8	4	5	6 (1)	59	62	59	60 (1)

NOTE.—MD = maximum pairwise divergence; QP<sub>4.2</sub> = the most recent version (4.2) of quartet puzzling; QP<sub>B</sub> = quartet puzzling based on binary weighting; QP<sub>C</sub> = quartet puzzling based on continuous weighting; WO = weight optimization.

<sup>a</sup> Average percentage of correctly inferred trees over the three model trees respecting (or not respecting) the molecular clock; the number in parentheses indicates the rank of the method with regard to this score.

- 1,000 trees (as described above) and returns their majority-rule consensus tree.
2. Variants of quartet puzzling based on binary (QP<sub>B</sub>), discrete (QP<sub>D</sub>) and continuous (QP<sub>C</sub>) weighting, and on the consensus tree computed by the CONSENSE program, available in the PHYLIP package. Using CONSENSE (instead of the majority-rule consensus) led in our experiments to fully resolved trees and thus to inference of the correct tree more often. This was the consensus tree used in the tests of Strimmer and von Haeseler (1996) and Strimmer, Goldman, and von Haeseler (1997) (personal communication). We also tested variants of QP based on the reconstruction of a unique tree and on binary (QP<sub>B</sub><sup>1</sup>) and continuous weighting (QP<sub>C</sub><sup>1</sup>). These variants used the same maximum-likelihood computations as QP<sub>4.2</sub>.
3. WO based on the same maximum-likelihood computations as QP variants.
4. DNAPARS, the parsimony program (version 3.5c) of the PHYLIP package.
5. BIONJ, a distance method designed by Gascuel (1997). BIONJ improves the NJ algorithm of Saitou and Nei (1987) by using a statistical model of evolutionary distances obtained from sequences. The distances used as input were computed with DNADIST from the PHYLIP package, assuming the Kimura two-parameter model, as for other methods.

6. FASTDNAML, the maximum-likelihood program (version 1.2) introduced by Olsen et al. (1994). FASTDNAML comes from DNAML (Felsenstein 1981) and generally infers the same tree, but much faster.

#### Comparisons of the Different Methods

The results obtained by FASTDNAML, BIONJ, DNAPARS, WO, and the main variants of QP are detailed in tables 1–4. The results obtained are summarized in tables 5 and 6 by averaging over the two sequence lengths and the four evolutionary conditions. Tables 1, 2, and 5 provide the percentages of inferred trees that exactly corresponded to the correct tree. Tables 3, 4, and 6 indicate the average bipartition distance between the inferred tree and the correct one. The results obtained from these two types of comparisons were nearly equivalent, except for QP<sub>4.2</sub> which often inferred non-fully-resolved trees. We first analyze the performances of QP variants and compare them with that of WO. We then compare the performances of these quartet methods with those of other reconstruction methods.

Tables 5 and 6 indicate a significant decrease of the topological accuracy of QP when a single tree is reconstructed instead of 1,000 trees. Indeed, the performances of QP<sub>B</sub><sup>1</sup> and QP<sub>C</sub><sup>1</sup> are 12%–16% worse than those of QP<sub>B</sub> and QP<sub>C</sub>, respectively, in terms of percentage of cor-

**Table 2**  
**Percentages of Correctly Inferred Trees (sequence length 600)**

TREE		MOLECULAR CLOCK				NO CLOCK			
		AA	BB	AB	Avg. <sup>a</sup>	CC	DD	CD	Avg. <sup>a</sup>
MD ≈ 0.1 ...	QP <sub>4.2</sub>	17	29	21	22 (7)	29	28	27	28 (7)
	QP <sub>B</sub>	42	49	47	46 (5)	53	48	52	51 (4)
	QP <sub>C</sub>	36	59	45	47 (4)	50	45	49	48 (6)
	WO	45	44	46	45 (6)	52	49	52	51 (5)
	DNAPARS	57	53	52	54 (2)	54	57	56	55 (3)
	BIONJ	52	50	54	52 (3)	60	57	56	58 (2)
MD ≈ 0.3 ...	FASTDNAML	69	63	65	66 (1)	66	61	62	63 (1)
	QP <sub>4.2</sub>	48	61	54	54 (7)	74	79	78	77 (7)
	QP <sub>B</sub>	74	77	76	76 (3)	85	87	88	87 (4)
	QP <sub>C</sub>	66	84	74	75 (5)	82	87	87	85 (6)
	WO	77	74	76	76 (4)	88	89	89	89 (3)
	DNAPARS	65	77	69	70 (6)	83	86	88	86 (5)
MD ≈ 1.0 ...	BIONJ	76	77	81	78 (2)	90	92	92	91 (2)
	FASTDNAML	93	87	91	91 (1)	94	97	96	96 (1)
	QP <sub>4.2</sub>	21	43	26	30 (6)	76	84	79	80 (6)
	QP <sub>B</sub>	54	67	59	60 (4)	88	88	86	87 (4)
	QP <sub>C</sub>	32	74	51	52 (5)	83	89	85	86 (5)
	WO	63	62	64	63 (3)	91	92	90	91 (3)
MD ≈ 2.0 ...	DNAPARS	22	33	25	27 (7)	17	12	15	14 (7)
	BIONJ	62	62	65	63 (2)	92	92	93	93 (2)
	FASTDNAML	85	77	82	82 (1)	99	99	98	99 (1)
	QP <sub>4.2</sub>	0	1	0	1 (7)	24	36	30	30 (6)
	QP <sub>B</sub>	10	22	13	15 (4)	49	53	48	50 (4)
	QP <sub>C</sub>	2	25	9	12 (5)	37	48	43	43 (5)
MD ≈ 2.0 ...	WO	23	17	18	19 (2)	64	61	63	63 (3)
	DNAPARS	4	1	2	2 (6)	0	0	0	0 (7)
	BIONJ	22	18	17	19 (3)	76	71	73	73 (2)
	FASTDNAML	35	26	28	29 (1)	93	91	94	92 (1)

NOTE.—MD = maximum pairwise divergence; QP<sub>4.2</sub> = the most recent version (4.2) of quartet puzzling; QP<sub>B</sub> = quartet puzzling based on binary weighting; QP<sub>C</sub> = quartet puzzling based on continuous weighting; WO = weight optimization.

<sup>a</sup> Average percentage of correctly inferred trees over the three model trees respecting (or not respecting) the molecular clock; the number in parentheses indicates the rank of the method with regard to this score.

rectly inferred trees. This confirms the importance of reconstructing many trees for QP.

Our tests also confirm that using continuous weighting is slightly better than using discrete weighting (about 1% in terms of correctly reconstructed trees), but they question the idea that using discrete or continuous weighting (instead of binary weighting) improves the topological accuracy of QP. Indeed, tables 5 and 6 indicate that the average results obtained by QP<sub>B</sub> are better than those of QP<sub>D</sub> and QP<sub>C</sub> by about 2%–4% in terms of percentage of correctly inferred trees. This is likely explained by the fact that QP<sub>B</sub> is the only consistent variant of QP.

Tables 5 and 6 indicate that the topological accuracy of QP strongly depends on the model tree topology. The performances obtained by QP<sub>D</sub> and QP<sub>C</sub> are more than 20% worse for model tree AA than for tree BB in terms of percentage of correctly inferred trees, whereas other methods tend to have similar performances for both trees or better performances for tree BB than for tree AA. Moreover, the topological accuracy of QP<sub>C</sub> does not depend on the model tree. The topological bias of QP thus seems to be due to the use of the consensus, which tends to privilege trees having many external pairs (tree BB has four, whereas tree AA has only two). To test this assumption, we applied QP<sub>C</sub> 1,000 times on a test set made of 12 identical sequences and therefore containing no phylogenetic signal. The trees constructed

by QP<sub>C</sub> before the consensus step contained four external pairs on average, whereas those obtained after the consensus step always contained six (the largest possible number with 12 taxa). Moreover, on data files obtained from a model tree having six external pairs, QP<sub>C</sub> obtains an impressive performance, even better than that of FASTDNAML. This likely explains the previous excellent results obtained by QP<sub>D</sub> and QP<sub>C</sub> (Strimmer and von Haeseler 1996; Strimmer, Goldman, and von Haeseler 1997), because the model trees used to generate the sequences contained the greatest possible number of external pairs. QP<sub>B</sub> and QP<sub>4.2</sub> are also affected by this topological bias, but to a lesser extent (their performances are about 7%–8% worse for model tree AA than for tree BB). The difference with QP<sub>C</sub> and QP<sub>D</sub> is due to the fact that trees reconstructed by QP<sub>B</sub> are less diversified and that in the majority tree provided by QP<sub>4.2</sub>, only highly supported bipartitions are retained.

It appears from tables 3 and 4, that QP<sub>4.2</sub> is more efficient (in terms of bipartition distance) than QP<sub>C</sub> and QP<sub>B</sub> only when the evolution rate is so low (MD ≈ 0.1) and the sequences so short ( $l = 300$ ) that it is not uncommon that no mutation occurs along certain branches (Kumar 1996). In this case, QP<sub>4.2</sub> benefits from being the only method able (in our tests) to propose partially resolved trees, with about six bipartitions on average, whereas all other methods propose fully resolved trees having nine bipartitions. In all other conditions, QP<sub>B</sub> and

**Table 3**  
**Average Robinson and Foulds Distances Between the Reconstructed Tree and the Model (sequence length 300)**

	TREE	MOLECULAR CLOCK				NO CLOCK			
		AA	BB	AB	Avg. <sup>a</sup>	CC	DD	CD	Avg. <sup>a</sup>
MD ≈ 0.1 ...	QP <sub>4,2</sub>	4.3	3.8	4.1	4.1 (4)	3.6	3.7	3.6	3.6 (1)
	QP <sub>B</sub>	4.3	4.2	4.2	4.3 (6)	4.3	4.4	4.2	4.3 (7)
	QP <sub>C</sub>	4.9	3.5	4.2	4.2 (5)	4.3	4.3	4.1	4.2 (6)
	WO	4.1	4.6	4.2	4.3 (7)	4.3	4.2	4.0	4.2 (5)
	DNAPARS	3.4	3.6	3.7	3.5 (2)	3.8	3.8	3.6	3.7 (3)
	BIONJ	3.7	4.0	3.8	3.8 (3)	3.9	3.8	3.7	3.8 (4)
	FASTDNAML	3.1	3.3	3.3	3.3 (1)	3.8	3.6	3.6	3.7 (2)
MD ≈ 3.0 ...	QP <sub>4,2</sub>	3.1	2.2	2.9	2.7 (7)	1.8	1.7	1.8	1.8 (7)
	QP <sub>B</sub>	2.4	2.0	2.3	2.2 (4)	1.6	1.6	1.6	1.6 (4)
	QP <sub>C</sub>	3.2	1.5	2.5	2.4 (6)	1.8	1.7	1.7	1.8 (6)
	WO	2.2	2.4	2.3	2.3 (5)	1.6	1.5	1.5	1.5 (3)
	DNAPARS	2.3	2.0	2.4	2.2 (3)	1.8	1.5	1.6	1.6 (5)
	BIONJ	2.1	2.2	2.2	2.1 (2)	1.3	1.2	1.3	1.2 (2)
	FASTDNAML	1.2	1.4	1.3	1.3 (1)	0.8	0.8	0.8	0.8 (1)
MD ≈ 1.0 ...	QP <sub>4,2</sub>	4.3	3.6	4.1	4.0 (6)	1.9	1.7	1.8	1.8 (6)
	QP <sub>B</sub>	3.3	3.1	3.2	3.2 (4)	1.5	1.5	1.4	1.5 (4)
	QP <sub>C</sub>	4.1	2.7	3.5	3.4 (5)	1.9	1.7	1.7	1.8 (5)
	WO	2.5	3.6	3.0	3.0 (3)	1.3	1.3	1.2	1.3 (3)
	DNAPARS	4.4	4.9	4.9	4.8 (7)	5.8	5.7	5.8	5.7 (7)
	BIONJ	2.7	3.2	3.0	2.9 (2)	1.1	1.1	1.1	1.1 (2)
	FASTDNAML	1.5	2.2	2.1	1.9 (1)	0.4	0.4	0.5	0.4 (1)
MD ≈ 2.0 ...	QP <sub>4,2</sub>	6.6	6.5	6.6	6.6 (5)	4.3	4.1	4.2	4.2 (5)
	QP <sub>B</sub>	6.2	6.8	6.7	6.6 (6)	4.3	4.0	4.3	4.2 (4)
	QP <sub>C</sub>	6.6	6.5	6.6	6.6 (4)	4.5	4.1	4.3	4.3 (6)
	WO	5.2	7.6	6.3	6.4 (3)	3.8	3.5	3.6	3.6 (3)
	DNAPARS	8.6	11.7	9.8	10.0 (7)	13.9	12.7	13.2	13.3 (7)
	BIONJ	5.6	6.9	6.3	6.3 (2)	3.2	3.1	3.1	3.1 (2)
	FASTDNAML	4.6	6.4	5.7	5.6 (1)	1.2	1.1	1.1	1.1 (1)

NOTE.—MD = maximum pairwise divergence; QP<sub>4,2</sub> = the most recent version (4.2) of quartet puzzling; QP<sub>B</sub> = quartet puzzling based on binary weighting; QP<sub>C</sub> = quartet puzzling based on continuous weighting; WO = weight optimization.

<sup>a</sup> Average percentage of correctly inferred trees over the three model trees respecting (or not respecting) the molecular clock; the number in parentheses indicates the rank of the method with regard to this score.

QP<sub>C</sub> tend to be better ranked than QP<sub>4,2</sub>, according to the bipartition distance as well as the percentage of correctly inferred trees.

However, WO is better ranked than QP<sub>B</sub> and QP<sub>C</sub> for all trees but the BB tree according to both topological criteria (tables 5 and 6). Its performance is better overall than that of QP and less dependent on the correct tree topology.

Nevertheless, the performance of WO is still worse than those of the other tested methods. WO is always less accurate than FASTDNAML and BIONJ, and the only cases in which WO is ranked better than DNAPARS correspond to high evolution rates (MD ≈ 1.0 and MD ≈ 2.0), for which the parsimony methods are well known to be poorly suited.

Tables 5 and 6 clearly demonstrate that FASTDNAML is the most accurate method. On average, it finds the correct tree 12%–13% more often than BIONJ, which is the next best method. The topological accuracy of DNAPARS depends on the evolution rate. For a low evolution rate, which corresponds to the assumptions made by parsimony, DNAPARS is a little more accurate than BIONJ; for a medium evolution rate, BIONJ has an advantage over DNAPARS, and this advantage becomes dramatic for high evolution rates.

Since there are about  $n^4$  quartets, the time complexity of QP and WO ( $O(n^4)$ ; see above) is thus minimal. This complexity makes it possible to deal with

problems involving about one or two hundred taxa, while BIONJ and DNAPARS, which have  $O(n^3)$  complexity, may deal with problems involving a few thousand taxa. However, note that BIONJ and DNAPARS are heuristic algorithms, like WO or QP, and do not systematically provide optimal trees according to the criterion they optimize (e.g., parsimony in the case of DNAPARS). We tested the different programs on a PC with a 466-MHz processor and 128 MB RAM. The average computing time required for one of our data sets was less than 0.1 s for both BIONJ and DNAPARS, about 0.6 s for WO, 2.64 s for QP, and 4.13 s for FASTDNAML. On a larger data set, containing 25 taxa and 1,896 sites, the average computing time required was about 1.6 s for BIONJ (including the computation of distances by DNADIST), 2.9 s for DNAPARS, 53.0 s for WO, 101.0 s for QP, and 318.0 s for FASTDNAML. In this case, bootstrapping the data is easy for BIONJ and DNAPARS, but becomes problematic for the other methods.

## Discussion

We first describe complementary tests revealing that the disappointing results of WO are not due to inefficient optimization of the criterion. Then, we discuss the difficulties encountered by quartet methods.

**Table 4**  
Average Robinson and Foulds Distances Between the Reconstructed Tree and the Model (sequence length 600)

TREE	MOLECULAR CLOCK				NO CLOCK					
	AA	BB	AB	Avg. <sup>a</sup>	CC	DD	CD	Avg. <sup>a</sup>		
MD ≈ 0.1 . . . .	QP <sub>4.2</sub>	1.9	1.5	1.7	1.7 (7)	1.4	1.5	1.5	1.5 (7)	
	QP <sub>B</sub>	1.6	1.4	1.5	1.5 (4)	1.3	1.4	1.3	1.3 (4)	
	QP <sub>C</sub>	1.9	1.1	1.6	1.5 (5)	1.3	1.5	1.4	1.4 (6)	
	WO	1.5	1.6	1.6	1.6 (6)	1.3	1.4	1.4	1.3 (5)	
	DNAPARS	1.2	1.2	1.3	1.2 (2)	1.1	1.2	1.2	1.2 (3)	
	BIONJ	1.3	1.3	1.3	1.3 (3)	1.0	1.2	1.2	1.1 (2)	
	FASTDNAML	0.7	0.9	0.9	0.8 (1)	0.9	1.0	1.0	0.9 (1)	
	MD ≈ 0.3 . . . .	QP <sub>4.2</sub>	0.8	0.6	0.7	0.7 (7)	0.4	0.3	0.3	0.3 (6)
		QP <sub>B</sub>	0.6	0.5	0.5	0.5 (3)	0.3	0.3	0.2	0.3 (4)
		QP <sub>C</sub>	0.8	0.4	0.6	0.6 (5)	0.4	0.3	0.3	0.3 (5)
WO		0.5	0.6	0.5	0.5 (4)	0.2	0.3	0.2	0.2 (3)	
DNAPARS		0.8	0.5	0.7	0.7 (6)	0.4	0.3	0.3	0.3 (7)	
BIONJ		0.5	0.5	0.4	0.5 (2)	0.2	0.2	0.2	0.2 (2)	
FASTDNAML		0.1	0.3	0.2	0.2 (1)	0.1	0.1	0.1	0.1 (1)	
MD ≈ 1.0 . . . .		QP <sub>4.2</sub>	1.8	1.0	1.5	1.4 (6)	0.4	0.3	0.3	0.3 (6)
		QP <sub>B</sub>	1.1	0.8	1.0	1.0 (4)	0.3	0.3	0.3	0.3 (4)
		QP <sub>C</sub>	1.9	0.6	1.2	1.2 (5)	0.4	0.3	0.3	0.3 (5)
	WO	0.8	0.9	0.9	0.9 (3)	0.2	0.2	0.2	0.2 (3)	
	DNAPARS	2.5	2.1	2.4	2.3 (7)	3.6	4.8	4.2	4.2 (7)	
	BIONJ	0.9	0.9	0.8	0.9 (2)	0.2	0.2	0.2	0.2 (2)	
	FASTDNAML	0.3	0.5	0.4	0.4 (1)	0.0	0.0	0.0	0.0 (1)	
	MD ≈ 2.0 . . . .	QP <sub>4.2</sub>	4.4	3.8	4.3	4.2 (6)	1.7	1.4	1.6	1.6 (5)
		QP <sub>B</sub>	3.5	3.2	3.4	3.4 (4)	1.5	1.4	1.5	1.4 (4)
		QP <sub>C</sub>	4.3	3.0	3.8	3.7 (5)	1.7	1.4	1.6	1.6 (6)
WO		2.5	3.7	3.1	3.1 (3)	1.0	1.2	1.0	1.1 (3)	
DNAPARS		5.5	7.8	6.7	6.7 (7)	13.5	11.3	12.4	12.4 (7)	
BIONJ		2.7	3.3	3.1	3.1 (2)	0.7	0.9	0.8	0.8 (2)	
FASTDNAML		1.9	2.7	2.4	2.3 (1)	0.2	0.2	0.1	0.2 (1)	

NOTE.—MD = maximum pairwise divergence; QP<sub>4.2</sub> = the most recent version (4.2) of quartet puzzling; QP<sub>B</sub> = quartet puzzling based on binary weighting; QP<sub>C</sub> = quartet puzzling based on continuous weighting; WO = weight optimization.  
<sup>a</sup> Average percentage of correctly inferred trees over the three model trees respecting (or not respecting) the molecular clock; the number in parentheses indicates the rank of the method with regard to this score.

**Better Optimizing the *W* Criterion**

It could be conjectured that the results obtained by WO are disappointing because it uses a naive approach that is inefficient in optimizing the *W* criterion. When the correct tree *T* is not inferred by WO, two cases arise. Either *T* is the single optimum according to *W*—and in this case the problem is due to WO incorrectly optimizing the *W* criterion—or *T* is nonoptimal (or not the single optimum)—and in this case the problem concerns the *W* criterion, which is not discriminating enough.

In order to know which of these two problems limits the performance of WO, we isolated the second kind of error. Instead of defining a dynamic taxon addition order, we used 1,000 random-addition orders and constructed the 1,000 corresponding trees by optimizing the *W* criterion at each step. Then, we selected the tree among these 1,000 trees which had the highest *W* criterion value. This algorithm (WO<sup>1,000</sup>) does not guarantee that the resulting tree is optimal. However, in our tests, it always inferred a tree with a criterion value at least as good as that of the correct tree *T*. Thus, the only errors made by WO<sup>1,000</sup> corresponded to cases in which the *W* criterion was not sufficiently discriminating. We observed that the performance of WO<sup>1,000</sup> was hardly any better than that of WO: the average improvement over all model trees and conditions was no more than 2%, according to the percentage of correctly inferred

trees. This clearly shows that only a tiny fraction of the errors of WO are due to insufficient optimization.

**Possible Limits of Quartet Methods**

The quartet methods are highly sensitive to 4-tree inference errors. For example, trees CC and CC' in figure 5 differ only with respect to the resolution of the nine specific quartets {1, 2, 3, *x*} with *x* ∈ {4, 5, 6, 7, 8, 9, 10, 11, 12}. For 12 taxa, there are 495 quartets. Therefore, it is sufficient to change the resolution of 2% (9/495) of the quartets of tree CC to obtain the set of 4-trees which exactly correspond to tree CC'. In this case, it is obvious that all quartet methods infer tree CC' instead of tree CC. In fact, in the unweighted case, a change in the resolution of five quartets is enough to infer the wrong tree, whereas in the weighted case this depends on the weights. A change in the weights of the w4-trees corresponding to these nine quartets such that *Q*<sub>max</sub> = *Q*<sub>CC'</sub>, is enough to make all reasonable methods infer the wrong tree. More generally, with *n* taxa, there are trees which differ only on *n* - 3 quartets, whereas the total number of quartets is about *n*<sup>4</sup>. Thus, the minimum rate of badly inferred 4-trees sufficient for misleading all quartet methods quickly approaches 0 as the number of taxa increases.

In our simulations, we observed cases in which most of the quartets were correctly weighted and the

**Table 5**  
**Percentages of Correctly Inferred Trees (averaged over all sequence lengths and evolutionary conditions)**

TREE	MOLECULAR CLOCK				No CLOCK			
	AA	BB	AB	Avg. <sup>a</sup>	CC	DD	CD	Avg. <sup>a</sup>
QP <sub>4.2</sub> .....	11	19	14	15 (9)	30	35	33	33 (10)
QP <sub>B</sub> .....	29	36	32	32 (4)	49	50	49	49 (4)
QP <sub>D</sub> .....	20	41	28	30 (7)	43	47	45	45 (6)
QP <sub>C</sub> .....	20	43	28	30 (5)	43	46	45	45 (5)
QP <sub>B</sub> <sup>I</sup> .....	22	18	19	20 (8)	37	35	36	36 (8)
QP <sub>C</sub> <sup>I</sup> .....	14	14	13	14 (10)	30	30	31	30 (9)
WO .....	34	32	34	33 (3)	53	54	54	54 (3)
DNAPARS .....	25	28	24	26 (6)	27	29	30	29 (7)
BIONJ .....	36	35	36	36 (2)	60	60	61	60 (2)
FASTDNAML .....	52	46	48	49 (1)	72	72	72	72 (1)

NOTE.—MD = maximum pairwise divergence; QP<sub>4.2</sub> = the most recent version (4.2) of quartet puzzling; QP<sub>D</sub> = quartet puzzling based on discrete weighting; QP<sub>C</sub> = quartet puzzling based on continuous weighting; QP<sub>B</sub><sup>I</sup> = variant on quartet puzzling based on the reconstruction of a unique tree and on binary weighting; QP<sub>C</sub><sup>I</sup> = variant on quartet puzzling based on the reconstruction of a unique tree and on continuous weighting; WO = weight optimization.

<sup>a</sup> Average percentage of correctly inferred trees over the three model trees respecting (or not respecting) the molecular clock; the number in parentheses indicates the rank of the method with regard to this score.

correct tree was not reconstructed. For example, considering a medium evolution rate, despite the fact that there were about 90% of the correct quartets in  $Q_{max}$ , we observed that the correct tree was not well inferred in comparison with other inference methods. This suggests that the inference errors of the 4-trees were not homogeneously distributed among the quartets and that the major part of the errors concerned specific quartets. This phenomenon is illustrated by the example in figure 5. In this example, the five quartets {1, 2, 3,  $x$ } with  $x \in \{5, 6, 7, 8, 12\}$  are subject to long-branch attraction, which tends to support the 4-trees of CC' rather than those of CC. Moreover, the four other quartets which differentiate CC from CC' are also (to a lesser extent) affected by this phenomenon. Therefore, it appears that long-branch attraction influences the resolution of specific quartets and could be one of the main difficulties quartet methods are faced with. In global character methods such as FASTDNAML or DNAPARS, these long branches are split, thanks to the presence of other taxa, thus making the problem easier to solve. The analysis

for distance methods is less clear. It is surprising that although based on pairwise evolutionary distances (i.e., two-by-two maximum-likelihood analysis), they are more accurate than current quartet methods, which are based on a more extensive four-by-four maximum-likelihood analysis.

We stress that poor weighting of a few specific quartets is enough to lead quartet methods to infer an incorrect bipartition. Moreover, once a wrong bipartition is inferred due to badly weighted 4-trees, this may influence the entire tree topology. For example, assume that the correct tree is CC (fig. 5) and that due to long-branch attraction some 4-trees are badly weighted and indicate taxa 1 and 5 as an external pair; then, the three bipartitions respectively gathering together taxa {1, 2}, {1, 2, 3}, and {4, 5} can no longer be inferred. In this case, even if all quartets {1, 2, 3,  $x$ } with  $x \in \{4, 5, 6, 7, 8, 9, 10, 11, 12\}$  are correctly weighted, the branch separating taxa 1 and 2 from the other is not inferred. This clearly indicates the importance and difficulty of correctly weighting 4-trees.

**Table 6**  
**Average Topological Distances Between the Inferred and Model Trees (averaged over all sequence lengths and evolutionary conditions)**

TREE	MOLECULAR CLOCK				No CLOCK			
	AA	BB	AB	Avg. <sup>a</sup>	CC	DD	CD	Avg. <sup>a</sup>
QP <sub>4.2</sub> .....	3.4	2.9	3.2	3.2 (7)	1.9	1.8	1.9	1.9 (4)
QP <sub>B</sub> .....	2.9	2.8	2.9	2.8 (4)	1.9	1.9	1.9	1.9 (5)
QP <sub>D</sub> .....	3.5	2.5	3.0	3.0 (6)	2.0	1.9	1.9	1.9 (7)
QP <sub>C</sub> .....	3.5	2.4	3.0	3.0 (5)	2.0	1.9	1.9	2.0 (6)
QP <sub>B</sub> <sup>I</sup> .....	3.8	4.6	4.2	4.2 (9)	2.9	3.0	2.9	2.9 (8)
QP <sub>C</sub> <sup>I</sup> .....	4.9	4.9	4.9	4.9 (10)	3.4	3.4	3.2	3.3 (9)
WO .....	2.4	3.1	2.7	2.8 (3)	1.7	1.7	1.6	1.7 (3)
DNAPARS .....	3.6	4.2	4.0	3.9 (8)	5.5	5.2	5.3	5.3 (10)
BIONJ .....	2.4	2.8	2.6	2.6 (2)	1.5	1.5	1.4	1.5 (2)
FASTDNAML .....	1.7	2.2	2.0	2.0 (1)	0.9	0.9	0.9	0.9 (1)

NOTE.—MD = maximum pairwise divergence; QP<sub>4.2</sub> = the most recent version (4.2) of quartet puzzling; QP<sub>D</sub> = quartet puzzling based on discrete weighting; QP<sub>C</sub> = quartet puzzling based on continuous weighting; QP<sub>B</sub><sup>I</sup> = variant on quartet puzzling based on the reconstruction of a unique tree and on binary weighting; QP<sub>C</sub><sup>I</sup> = variant on quartet puzzling based on the reconstruction of a unique tree and on continuous weighting; WO = weight optimization.

<sup>a</sup> Average percentage of correctly inferred trees over the three model trees respecting (or not respecting) the molecular clock; the number in parentheses indicates the rank of the method with regard to this score.

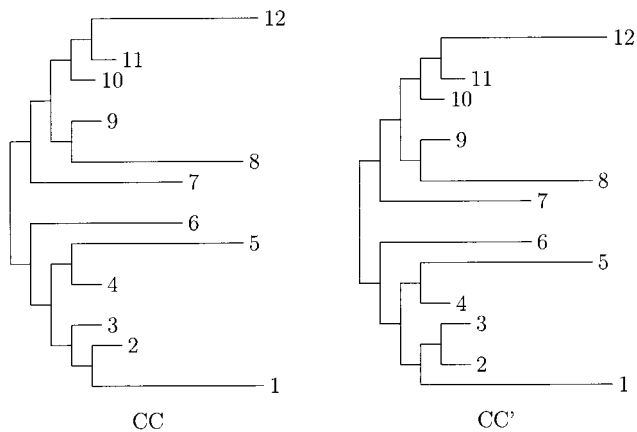


FIG. 5.— Only a few errors in quartet inference is enough to mislead quartet methods. The nine quartets  $\{1, 2, 3, x\}$  with  $x \in \{4, 5, 6, 7, 8, 9\}$  are the only ones with different topologies in CC and CC'. Moreover, they are all (to some extent) subject to long-branch attraction.

Although the likelihood approach used by QP (Strimmer, Goldman, and von Haeseler 1997) and then by WO seems well founded, this weighting system seems ill-adapted to the quartet approach. As an illustration, it should be noted that the likelihood of each 4-tree is well defined but does not allow us to predict the likelihood of the whole tree. Moreover, the continuous version of WO (presented above) finds the correct tree on average only about 3% more often than the binary variant (results not shown). The inadequacy of the weighting system currently used by QP and WO could thus explain their disappointing results.

## Conclusions

We have presented a new phylogenetic reconstruction algorithm based on weighted 4-trees. This algorithm is generally more efficient than QP, and its topological accuracy depends less on the correct tree topology. Moreover, it is faster and offers better theoretical guarantees. Unfortunately, our simulations seem to indicate that this method is less efficient overall than distance and maximum-likelihood methods. We have been working on quartet methods for a long time (Berry and Gascuel 1997, 2000) and were the first to be disappointed by the current results. Moreover, unpublished tests (V. Berry, personal communication) on quartet cleaning methods (Berry et al. 1999) seem to confirm our observations.

A great deal of work was carried out on ways of combining 4-trees to obtain a complete  $n$ -tree. The weaknesses of the various quartet methods tested in this paper are very likely due to the method of weighting the 4-trees, rather than the method of combining them. Indeed, as explained above, considering only four taxa requires correct management of long-branch attraction. We hope that this will become possible with the new approaches of Lyons-Weiler and Takahashi (1999), who suggest ways in which calculation of likelihood could handle this problem, or by Willson (1999b), who pro-

poses a version of parsimony that is more resistant to long-branch attraction.

## Acknowledgment

We thank David Bryant for his helpful comments and advice.

## LITERATURE CITED

- ADACHI, J., and M. HASEGAWA. 1999. Instability of quartet analyses of molecular sequence data by the maximum likelihood method: the cetacea/artiodactyla relationships. *Cladistics* **5**:164–166.
- BANDELT, H.-J., and A. DRESS. 1986. Reconstructing the shape of a tree from observed dissimilarity data. *Adv. Appl. Math.* **7**:309–343.
- BARTHÉLEMY, J. P., and A. GUÉNOCHE. 1990. *Trees and proximity representation*. John Wiley and Sons, Chichester, England.
- BERRY, V., and O. GASCUEL. 1997. Inferring evolutionary trees with strong combinatorial evidence. *Lect. Notes Comput. Sci.* **1276**:111–123.
- . 2000. Inferring evolutionary trees with strong combinatorial evidence. *Theor. Comput. Sci.* **240**:271–298.
- BERRY, V., T. JIANG, P. KEARNEY, and M. LI. 1999. Quartet cleaning: improved algorithms and simulations. *Lect. Notes Comput. Sci.* **1643**:313–324.
- CORMEN, T. H., C. E. LEISERSON, and R. L. RIVEST. 1992. *Introduction to algorithms*. MIT Press, Cambridge, Mass.
- ERDÖS, P., M. STEEL, L. A. SZÉKELY, and T. WARNOW. 1997. Constructing big trees from short sequences. *Lect. Notes Comput. Sci.* **1256**:827–837.
- FELSENSTEIN, J. 1981. Evolutionary trees from DNA sequences: a maximum likelihood approach. *J. Mol. Evol.* **17**:368–376.
- . 1989. *Phylogeny inference package* (version 3.2). *Cladistics* **5**:164–166.
- GASCUEL, O. 1997. BIONJ: an improved version of the NJ algorithm based on a simple model of sequence data. *Mol. Biol. Evol.* **14**:685–695.
- JIANG, T., P. KEARNEY, and M. LI. 1998. Orchestrating quartets: approximation and data correction. Pp. 416–425 in R. MOTWANI, ed. *Proceedings of the 39th IEEE Annual Symposium on Foundations of Computer Science*, Los Alamitos, Calif.
- KUMAR, S. 1996. A stepwise algorithm for finding minimum evolution trees. *Mol. Biol. Evol.* **13**:584–593.
- LYONS-WEILER, J., and K. TAKAHASHI. 1999. Branch length heterogeneity leads to nonindependent branch length estimates and can decrease the efficiency of methods of phylogenetic inference. *J. Mol. Evol.* **49**:392–405.
- MARGUSH, T., and F. MCMORRIS. 1981. Consensus  $n$ -trees. *Bull. Math. Biol.* **43**:239–244.
- OLSEN, G., H. MATSUDA, R. HAGSTROM, and R. OVERBEEK. 1994. A tool for construction of phylogenetic trees of DNA sequences using maximum likelihood. *Comput. Appl. Biosci.* **10**:41–48.
- PHILIPPE, H., and E. DOUZERY. 1994. The pitfalls of molecular phylogeny based on four species, as illustrated by the cetacea/artiodactyla relationship. *J. Mamm. Evol.* **2**:133–152.
- RAMBAUT, A., and N. GRASSLY. 1997. Seq-gen: an application for the Monte Carlo simulation of DNA sequence evolution along phylogenetic trees. *Comput. Appl. Biosci.* **13**:235–238.
- 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.
- STEEL, M. 1992. The complexity of reconstructing trees from qualitative characters and subtrees. *J. Classif.* **9**:91–116.
- STRIMMER, K., N. GOLDMAN, and A. VON HAESELER. 1997. Bayesian probabilities and quartet puzzling. *Mol. Biol. Evol.* **14**:210–211.
- STRIMMER, K., and A. VON HAESELER. 1996. Quartet puzzling: a quartet maximum-likelihood method for reconstructing tree topologies. *Mol. Biol. Evol.* **13**:964–969.
- SWOFFORD, D. L., G. J. OLSEN, P. WADDEL, and D. HILLIS. 1996. Phylogenetic inference. Pp. 407–514 in C. M. DAVID, M. HILLIS, and B. MABLE, eds. *Molecular systematics*. 2nd edition. Sinauer, Sunderland, Mass.
- WILLSON, S. J. 1999a. Building phylogenetic trees from quartets by using local inconsistency measure. *Mol. Biol. Evol.* **16**:685–693.
- . 1999b. A higher parsimony method to reduce long branch attraction. *Mol. Biol. Evol.* **16**:694–705.

## APPENDIX

 **$O(n^4)$  Implementation of WO**

Let  $T$  denote the current partially constructed tree. An internal node splits  $T$  into three subtrees, denoted as  $T_x$ ,  $T_y$ , and  $T_z$  (fig. 2). Gathering the relevant quartets  $\{i, x, y, z\}$  such that  $x \in T_x$ ,  $y \in T_y$ ,  $z \in T_z$ , and  $i$  is a taxon not yet added allows efficient computation of branch bonuses relative to the addition of  $i$ . Indeed, instead of searching  $T_x$  for each w4-tree ( $ix|yz, w$ ) in order to add a bonus  $w$  on all branches of  $T_x$ , it is more efficient to compute the sum  $S$  of these bonuses and to add it onto branches of  $T_x$  by a single tree search. When a new taxon  $z_{\text{new}}$  is added onto one subtree, say,  $T_z$ ,  $S$  is updated just by adding to it weights of the new w4-trees ( $ix|yz_{\text{new}}, w$ ). The WO implementation (fig. 3) exploits these observations. It associates to each subtree  $T_x$ , and for each not-yet-added taxon  $i$ , a value denoted as

$S_{i, T_x}$ . These  $O(n^2)$  values are updated after each new addition.

The algorithm shown in figure 3 initializes result tree  $T$  with three randomly selected taxa. Then, it initializes the set  $R$  of taxa not yet added and stores the last added taxon. Afterward, while  $R$  is not empty, each remaining taxon is considered and the associated bonuses are computed; the taxon with the highest safety is added to  $T$  on the branch having the highest bonus for this taxon. This taxon is removed from  $R$  and becomes the new last added taxon. For a particular taxon  $i$ , the branch bonuses are computed as follows. First, the branch bonuses are set at 0; then (loop 1), each internal node is considered, and the values to propagate for taxon  $i$  in the three corresponding subtrees are updated to take into account the last added taxon; finally (loop 2), these three values are added to bonuses of branches belonging to the corresponding subtrees. Note that the addition of a new taxon generates three new subtrees whose initial  $S$  values are set at 0 and updated during the next passage through loop (1).

Most of the computational time is due to loop (1) and loop (2). Since they are executed sequentially, the complexity of the whole algorithm is the same as the complexity of the algorithm without loop (1) added to the complexity of the algorithm without loop (2). By removing loop (1), we clearly obtain an  $O(n^4)$  algorithm (four nested loops of  $n$  steps). By removing loop (2), we obtain an algorithm which consults each w4-tree only once, which ensures that it is also an  $O(n^4)$  algorithm. Thus, the whole complexity of the WO algorithm is  $O(n^4)$ . In our experiments, we used a refined implementation (still in  $O(n^4)$ ) based on the tree search introduced by Berry and Gascuel (2000) to propagate bonuses along subtrees.

MANOLO GOUY, reviewing editor

Accepted February 22, 2001