A Simulation Comparison of Phylogeny Algorithms under Equal and Unequal Evolutionary Rates

Mary K. Kuhner and Joseph Felsenstein
Department of Genetics, University of Washington

Using simulated data, we compared five methods of phylogenetic tree estimation: parsimony, compatibility, maximum likelihood, Fitch-Margoliash, and neighbor joining. For each combination of substitution rates and sequence length, 100 data sets were generated for each of 50 trees, for a total of 5,000 replications per condition. Accuracy was measured by two measures of the distance between the true tree and the estimate of the tree, one measure sensitive to accuracy of branch lengths and the other not. The distance-matrix methods (Fitch-Margoliash and neighbor joining) performed best when they were constrained from estimating negative branch lengths; all comparisons with other methods used this constraint. Parsimony and compatibility had similar results, with compatibility generally inferior; Fitch-Margoliash and neighbor joining had similar results, with neighbor joining generally slightly inferior. Maximum likelihood was the most successful method overall, although for short sequences Fitch-Margoliash and neighbor joining were sometimes better. Bias of the estimates was inferred by measuring whether the independent estimates of a tree for different data sets were closer to the true tree than to each other. Parsimony and compatibility had particular difficulty with inaccuracy and bias when substitution rates varied among different branches. When rates of evolution varied among different sites, all methods showed signs of inaccuracy and bias.

Introduction

A number of algorithms for estimating phylogeny from DNA sequence data are in use, and it is not always clear what the strengths and weaknesses of each method are, or which should be preferred in a given situation. A number of computer simulation studies of these methods have been done (for reviews, see Felsenstein 1988; Nei 1991). The majority of these studies have involved relatively small numbers of replications—the computational load of larger-scale tests has been simply too great. Furthermore, the biologically important case of unequal substitution rates at different sites in the molecule has seldom been explored.

Advances in both computer and algorithm speed have allowed us to simulate and analyze several thousand data sets, providing a thorough look at the performance of the various methods. In this study, we compare the performance of five major phylogeny algorithms—parsimony, compatibility, maximum likelihood, Fitch-Margoliash, and neighbor joining—on simulated DNA data, including cases of unequal rates of evolution either on different branches of the tree or at different sites in the sequence. The model trees used are randomly constructed, presenting a variety of different topologies comparable to those seen in real data. A method of tree comparison is used that allows the algorithms to be scored by how accurately they recover the true topology and branch length. We also examine a measure of bias: is the cloud of estimated trees produced by generating many data sets from the same true tree centered on the true tree, or on some other tree? We were able to analyze 5,000 estimations for each combination of rate and sequence length, ~10 times as many as in most other studies, providing a solid database to support conclusions about the relative effectiveness of these methods.

Key words: Parsimony, compatibility, maximum likelihood, Fitch-Margoliash, neighbor-joining, computer simulation, phylogeny.

Address for correspondence and reprints: Mary K. Kuhner, Department of Genetics SK-50, University of Washington, Seattle, Washington 98195.

Mol. Biol. Evol. 11(3):459–468. 1994.

0737-4038/94/1103-0013$02.00

Material and Methods

All programs used in this study were taken from PHYLIP version 3.4 (DNAPARS for parsimony, DNACOMP for compatibility, NEIGHBOR for the neighbor-joining method, and FITCH for the Fitch-Margoliash method), except for the maximum-likelihood program, for which we used a preliminary C language version of fastDNAml provided us by G. Olsen (this was several times faster than PHYLIP's DNAML program).

Generation of Phylogenetic Trees

Phylogenies of 10 taxa were randomly generated using a branching process. A single lineage was imagined

to just have split in two. If there is a constant probability dt that a lineage will split during the next short time interval of length dt, then, when there are k lineages, the time until the next one splits will be drawn from an exponential distribution with expectation $1/k$. The trees were simulated by drawing a time from this distribution (the initial value being $k = 2$) and lengthening each of the k branches by this amount and then choosing one of the k at random to be the one that splits. After the split there are now $k + 1$ branches, so k is increased by 1 and the process is repeated. The process stops when the split that would create the 11th branch is about to occur (when the time for it has already elapsed). This process produces an assortment of trees, some of which will be much more difficult for any method to correctly reconstruct than others. We feel that this represents the range of possible data more accurately than focusing attention on a specific model tree.

For each tree created, multiple independent data sets were generated by simulating evolutionary change along the tree. The simulation program used the Kimura (1980) two-parameter model of sequence evolution, with a transition/transversion ratio of 2.0, to randomly evolve DNA sequences according to an input phylogeny. Each site evolved independently, starting from a random nucleotide sampled equiprobably from A, G, C, and T and simulating change according to the Markov chain specified by the Kimura two-parameter model, with the time for the change given by the length of that branch in the tree. Changes in different branches were independent, starting from the nucleotide that was achieved in the common ancestor of the branches. Trees for which some sites evolve faster than others were simulated by multiplying the branch lengths by a constant before simulating the "fast" sites.

Distance Matrices

Matrices of corrected evolutionary distances were generated by PHYLIP program DNADIST, using the same Kimura two-parameter model and transition/transversion ratio that were used to generate the data. This program estimates the distance by maximum-likelihood estimation under that model.

Phylogeny Algorithms

Five methods were tested: parsimony, compatibility, maximum likelihood, Fitch-Margoliash (Fitch and Margoliash 1967), and neighbor joining (Saitou and Nei 1987). The programs for the first four methods listed used a heuristic search method, which attempts to find the best tree by a specific criterion but is not guaranteed to do so (programs that perform branch-and-bound searches and guarantee finding the optimal tree are very slow). The neighbor-joining method always uses a stepwise construction approach rather than a search for the optimal tree.

The algorithms that search for optimal trees can contain an optional final step, global rearrangement, which involves removing each branch in turn and trying all possible positions for it; this improves performance but approximately triples run time. It was not used for the two slowest programs, those for maximum likelihood and the Fitch-Margoliash method, as preliminary simulations suggested that it would not substantially improve their performance (data not shown).

The parsimony method uses the Wagner parsimony criterion (Eck and Dayhoff 1966; Kluge and Farris 1969; Fitch 1971). The related compatibility method (Le-Quesne 1969) tries to find the tree compatible with the largest number of sites, where compatibility is defined as not requiring any nucleotide to arise twice on the tree. The PHYLIP implementations of these programs do not estimate branch lengths.

The maximum-likelihood method is an extension of the method described earlier (Felsenstein 1981*a*) to use base frequencies calculated from the input data and uses a model of base substitution that allows not only unequal base frequencies but inequality of transition and transversion rates. The model is that invented by one of us (J.F.) and first described by Kishino and Hasegawa (1989). We supplied it with the correct transition/transversion ratio, 2.0.

The remaining two methods estimate trees on the basis of distance matrices. One uses the least-squares criterion of Fitch and Margoliash (1967), which searches for the tree minimizing the sum of squared differences between the actual distances and those on the tree. The power parameter, which controls the relative weighting of long versus short distances, was set to 2.0, Fitch and Margoliash's original value, because preliminary simulations showed this to be most accurate (data not shown). The method of assigning branch lengths for a given tree topology was not that used in Fitch and Margoliash's original program but used alternating least squares to find branch lengths that solve the normal equations for the least-squares estimate. The method of searching among tree topologies was, as we have indicated, similar to that used in the parsimony, compatibility, and maximum-likelihood programs.

The neighbor-joining method (Saitou and Nei 1987, simplified as in Studier and Keppler 1988) is a distance-matrix method that sequentially modifies an initial star phylogeny in order to minimize the total branch length. This approximates the method of minimum evolution (Rzhetsky and Nei 1992).

Both of the distance-matrix methods sometimes estimate negative branch lengths. Preliminary simula-

tions suggested that they would perform better if negative branches were disallowed. In the case of the Fitch-Margoliash method, disallowing negative lengths affects the search among topologies, and so it can change both topology and branch length accuracy; for the neighbor-joining method, only branch length is affected.

To disallow negative branches in the Fitch-Margoliash method, we constrained the negative branch to length 0 and then did a constrained least-squares solution for the two adjacent branches. Since our algorithm uses an iterative approach to solve for the least-squares branch lengths, such a change at one branch will affect the lengths of other branches throughout the tree.

Neighbor joining is a sequential construction algorithm and does not allow for this type of global adjustment to remove negative branches. Instead, when a negative branch occurred, we immediately set its length to 0, adjusting its sibling branch accordingly.

We considered including the method of minimum evolution (Rzhetsky and Nei 1992) in this study, but we were unable to find a suitable correction for negative-length branches. Without correction, the method's performance was poor in preliminary simulations (data not shown), but this is not a fair comparison.

Comparison of Trees

The success of the algorithms in recovering the correct tree was evaluated by two methods, both of which compute distances between trees. The first is the *dT* score of Robinson and Foulds (1981), which measures the number of internal branches that exist in one tree but not in the other; for trees of 10 taxa it varies between 0 (identical topologies) and 14. (This measure is described in more detail below.)

Trees with 0-length branches were not treated as multifurcating. Rather, we accepted whatever resolution of the multifurcation the estimating program produced.

For the three methods (maximum likelihood, Fitch-Margoliash, and neighbor joining) that estimate branch lengths, we have developed a distance between unrooted trees that is sensitive to the correctness of branch lengths. This distance measure, which we call the "branch score" (*Bs*), is the sum of squares of the differences between each branch's length in the true and deduced trees. Branches that appeared in one tree but not in the other were scored as if compared to a branch of length 0. This means that, for a sufficiently short branch, an algorithm that does not find the branch will score better than an algorithm that assigns it a great length. Like *dT*, the branch score is 0 for identical trees and increases as the match worsens. However, it depends on the absolute size of the trees being compared, and so branch scores cannot be directly compared among trees with different substitution rates. The branch score is closely related to *dT*, as *dT* is the branch score for trees in which all branches have length 1. Note that branch-score measures differences in topology as well as branch length, though it can be 0 for a comparison between two nonidentical topologies if all the discordant branches are of length 0.

A more precise definition of the branch score shows that its square root is a metric and thus should be called a distance. If we consider the set S of all species, each branch in the tree induces a partition of the elements of this set, dividing them into two sets—R_1 and R_2—according to whether they are connected to one end of the branch or the other. As the trees are unrooted, we do not distinguish between the partitions $\{R_1, R_2\}$ and $\{R_2, R_1\}$. Now consider the large set $(P_1, P_2, \ldots, P_N)$ of all possible partitions of S into two sets. For each tree, we can define an array B of nonnegative reals $(b_1, b_2, \ldots, b_N)$. The real number b_i is the branch length of the branch corresponding to partition P_i, unless that branch does not exist in the tree, in which case it is 0. Thus most elements of B will be 0: for 10 species there are 511 possible partitions, only 19 of which will correspond to branches in any one fully resolved tree.

For each tree, we can imagine calculating the corresponding array B. For two trees whose arrays are B and B', the branch score is simply the squared Euclidean distance between these arrays:

$$Bs(B, B') = \sum_{i=1}^{N} (b_i - b'_i)^2 . \qquad (1)$$

Robinson and Foulds's *dT* is simply the branch score where all of the non-0 values of b_i and b'_i are 1's, so that the squared difference $(b_i - b'_i)^2$ is 1, if the branch exists in one of the two trees and not in the other, and otherwise is 0.

Although we have not made use of them, extensions of the branch score are straightforward. To define a score for rooted trees, we need only use ordered rather than unordered partitions, so that in the partition $\{R_1, R_2\}$ it is R_2 that contains the root, distinguishing it from the partition $\{R_2, R_1\}$. If we wish to have a squared distance that is sensitive to relative branch lengths rather than to absolute branch lengths, we need only divide all the b_i by their sum. This ensures that two trees that are of the same topology and have proportional branch lengths will have a branch score of 0; however, this approach may run into trouble if some branches are allowed to have negative lengths.

Note that the branch score as we have defined and used it is sensitive both to difference in branch lengths and also to differences in tree topology, although, if a difference in topology occurs in a region that has very short branches, it may lead to a low branch score. In

simulations where the substitution rate was not equal at all sites, branch lengths of the "true" tree were computed by taking an average of the rates across all sites. When an algorithm produced multiple tied trees (possible only with parsimony and compatibility in this simulation), we took the average either of *dT*s or of the branch scores for all trees produced.

Bias

Several studies have addressed the consistency of phylogenetic algorithms: does the method converge to the true tree with an infinite amount of data? We chose instead to explore a related question, which is, perhaps, of more importance in actual studies with finite data sets: does the method have a systematic preference for something other than the true tree with a finite amount of data? Such a preference represents a bias in the method and can exist even when the method is consistent. We chose to measure bias by comparing the differences (measured by either *dT* or branch score) between reconstructions of the same tree (based on different data sets) with the differences between these reconstructions and the true tree. If the reconstructed trees form a cloud centered on the true tree, the differences between reconstructions should be greater, on average, than the differences between each reconstruction and the truth. If the cloud is centered on an incorrect tree, the reconstructions may be more similar to one another than to the truth.

To avoid computational burden and lack of independence, we sampled the difference between the first and second reconstructions, the third and fourth, and so on, producing a set of 50 differences for each series of 100 reconstructions. We compared the mean of this set of differences with the mean of the 100 differences between reconstructions and the true tree, and we scored the run as unbiased if the reconstructions were closer to the true tree than to one another. This is a conservative test for bias; if the cloud of trees is symmetrical around the true tree, then, since the individual trees would be independently placed in a Euclidean space, the squared distance between trees would be expected to be, on average, twice as great as the squared distance to the true tree. This means that there is some tendency for this measure of bias to conclude that bias is not present. A method that is consistent with infinite data may still be biased with finite data; conversely, a method that is inconsistent with infinite data might not have detectable bias with finite data if the difference between its preferred tree and the truth were small compared with the error in tree estimation based on a small data set.

Results

Throughout this study, "low" evolutionary rate corresponds to a rate of 0.01, and "high" corresponds to a rate of 0.1. Time is scaled in units such that the average time that elapses in a lineage until it branches is 1.0 mutation per site. Thus, a substitution rate of 0.01 per unit time is equivalent to having 0.01 times as great a probability that a single base changes as that the lineage splits. Note that, in a tree of 10 species, the expectation of the sum of all branch lengths is 1.93, when the branch lengths are given in terms of the probability of a lineage splitting. (This number is the sum of the expected times from each split to the next, which have expectations of $1/2$, $1/3$, . . . , $1/10$.) A substitution rate of 0.01 thus means that, in a tree of average total length, the average number of changes per site will be 0.0193. The number of changes reconstructed by a parsimony algorithm may be slightly less than that, owing to the minimization step of the algorithm.

Performance of the five algorithms was evaluated under a number of conditions: uniform clocklike evolution with a high or low rate; unequal rates on different branches (half the branches high, half low); and unequal rates at different sites (half the sites high, half low). Fifty random trees were generated for each condition; for each tree, 100 replicate data sets were generated for each of four length categories (100, 300, 1,000 and 3,000 bp), resulting in 5,000 phylogeny estimations for each combination of condition and length.

For the case of unequal rates on different branches, the simulation program that assigned rates to branches did so with a 50% probability of a high rate (10 times as high as the low rate). Thus the number of branches that had high and low rates would vary from tree to tree.

In the tables, the mean *dT* or branch scores for the condition/length combination are presented, followed by the difference of each method's score from the mean. (Note that a lower score indicates better performance.) Asymptotically, with very large amounts of data, one can show that the branch score will decline inversely with the number of sites. This behavior is very nearly realized for many of the tables discussed below.

Bias scores are presented separately; each score is based on 50 independent data trees, for each of which the 100 data sets generated were scored as showing or not showing bias. The number shown is the number of trees that appeared biased, so that a method that was always found to be biased would have a score of 50.

It should be noted that all of the methods except neighbor joining use a heuristic search that is not guaranteed to find the maximal tree, and details of the way this search is done could influence the programs' success. It is possible that different implementations of the algorithms would give different results, although we suspect that the results presented here are not strongly dependent on the success of the heuristic search. A

Table 1
Comparison of Methods under Low (0.01) Substitution Rate

A. Accuracy of Topologies

Sites	Mean dT	Algorithm				
		Pars	Comp	ML	F-M	N-J
100	7.945	0.03	0.06	−0.66	0.22	0.35
300	4.463	0.12	0.17	−0.41	0.02	0.10
1,000	1.837	0.11	0.14	−0.20	−0.04	−0.02
3,000	0.736	0.02	0.04	−0.07	0.01	0.00

B. Trees (of 50) Showing Bias in Topology

Sites	Algorithm				
	Pars	Comp	ML	F-M	N-J
100	0	0	0	0	0
300	0	0	0	0	0
1,000	1	0	0	1	0
3,000	2	1	0	0	0

C. Accuracy of Branch Lengths

Sites	Mean Bs	Algorithm		
		ML	F-M	N-J
100	95.972	1.12	−0.10	−1.02
300	32.965	0.10	−0.04	−0.06
1,000	9.807	−0.05	0.02	0.03
3,000	3.279	−0.02	0.01	0.01

D. Trees (of 50) Showing Bias in Branch Lengths

Sites	Algorithm		
	ML	F-M	N-J
100	0	0	0
300	0	0	0
1,000	0	0	0
3,000	0	0	0

NOTE.—Part A gives both mean dT for the five algorithms and the difference between each algorithm's dT score and the mean. Part B gives the number of trees (of 50) in which the mean dT distances between estimated and true trees were smaller than those among estimated trees, a measure of whether the estimated trees are centered around the true tree. Parts C and D give similar information—using branch score × 10^5, rather than dT—for the three algorithms that calculated branch lengths. Note that both dT and branch score increase with increasing inaccuracy; the method with the most negative difference from the mean was the most accurate. Pars = parsimony; Comp = compatibility; ML = maximum likelihood; F-M = Fitch-Margoliash; N-J = neighbor joining; and Bs = branch score.

preliminary test in which an exact branch-and-bound search was used with the parsimony criterion showed no differences, in behavior, from the heuristic search tested here (data not shown).

Uniform Rates

Tables 1 and 2 show the relative performance of the algorithms under a clocklike model of evolution. On the whole, the algorithms were quite successful at estimating the correct tree; under the lower rate, 1,000 bp were required to reduce the average errors per tree to <1 (dT<2.0), while under the higher rate only 300 bp were needed. Accuracy of the branch length estimates increased smoothly with increasing sequence length.

Under both uniform rates parsimony was somewhat more successful than compatibility, while the dis-

Table 2
Comparison of Methods under High (0.1) Substitution Rate

A. Accuracy of Topologies

Sites	Mean dT	Algorithm				
		Pars	Comp	ML	F-M	N-J
100	2.767	−0.01	0.29	−0.18	−0.06	−0.05
300	1.333	0.02	0.16	−0.19	0.00	0.01
1,000	0.659	0.02	0.09	−0.12	0.01	0.01
3,000	0.383	−0.01	0.02	−0.07	0.03	0.03

B. Trees (of 50) Showing Bias in Topology

Sites	Algorithm				
	Pars	Comp	ML	F-M	N-J
100	0	0	0	0	0
300	0	0	0	0	0
1,000	2	1	1	0	0
3,000	1	3	0	0	1

C. Accuracy of Branch Lengths

Sites	Mean Bs	Algorithm		
		ML	F-M	N-J
100	1,708.245	−25.41	−4.03	29.44
300	553.576	−23.98	7.93	16.06
1,000	162.536	−8.99	3.30	5.69
3,000	54.298	−3.06	1.25	1.82

D. Trees (of 50) Showing Bias in Branch Lengths

Sites	Algorithm		
	ML	F-M	N-J
100	0	0	0
300	0	0	0
1,000	0	0	0
3,000	0	0	0

NOTE.—See Note to table 1.

tance methods (Fitch-Margoliash and neighbor joining) had nearly identical results. The distance methods were slightly inferior to parsimony both with short sequences with low rates and with long sequences with high rates and were slightly superior in the other cases. Both were somewhat less accurate than maximum likelihood. The parsimony and compatibility methods showed signs of bias with longer sequences, perhaps owing to occasional generation of trees with very long branches, as such branches are known to be erroneously grouped together by parsimony (Felsenstein 1978). For low evolutionary rates all methods were equally successful at estimating branch lengths, while for higher rates maximum likelihood was slightly more successful; all three methods were unbiased.

Unequal Rates per Branch

Table 3 shows results for unequal substitution rates on different branches. A randomly chosen 50% of branches evolved at the high rate, while the remainder evolved at the low rate. The ratio between high and low rates was 10-fold, which happens to be the same as the ratio between the rates used in tables 1 and 2. Thus, if the accuracy with which a branch length could be constructed depended only on the latter's own length, we would expect the means from table 3 to be midway between those from tables 1 and 2. They are in fact quite close to this, though generally a little higher (reflecting the difficulties caused by unusually long or short branches).

This case was expected a priori to be difficult for the parsimony method (Hendy and Penny 1989; Zharkikh and Li 1993)—and, presumably, also for the closely related compatibility method. Both of them performed relatively poorly, with parsimony slightly superior to compatibility. However, even with such grossly unequal substitution rates both methods produced less than one error per tree, on average, with ⩾1,000 bp. As expected, both methods had trouble with bias.

The Fitch-Margoliash method was somewhat more successful than neighbor joining, with this type of data, and maximum likelihood was more successful than either of these distance methods. These methods were generally unbiased. The same pattern held in the branch length estimates.

Unequal Rates per Site

Table 4 shows results for substitution rates varying by site. Half the sites evolved at the low rate, and the other half evolved at the high rate. This case allowed for fairly accurate estimation with short sequences—comparable to the results in table 3—but lengthening the sequences produced much less improvement in accuracy. All methods showed signs of bias.

Table 3
Comparison of Methods When Substitution Rate Varies by Branch

A. Accuracy of Topologies

		ALGORITHM				
SITES	MEAN *dT*	Pars	Comp	ML	F-M	N-J
100	4.980	0.15	0.37	−0.45	−0.09	0.02
300	3.079	0.15	0.33	−0.37	−0.13	0.03
1,000	1.660	0.14	0.21	−0.31	−0.07	0.03
3,000	0.889	0.18	0.21	−0.29	−0.09	−0.02

B. Trees (of 50) Showing Bias in Topology

	ALGORITHM				
SITES	Pars	Comp	ML	F-M	N-J
100	0	0	0	0	0
300	0	0	1	0	0
1,000	5	3	0	0	0
3,000	8	9	0	0	1

C. Accuracy of Branch Lengths

		ALGORITHM		
SITES	MEAN *Bs*	ML	F-M	N-J
100	669.892	−15.97	1.42	14.56
300	224.530	−7.18	1.46	5.72
1,000	67.032	−2.56	0.65	1.90
3,000	22.260	−0.93	0.27	0.66

D. Trees (of 50) Showing Bias in Branch Lengths

	ALGORITHM		
SITES	ML	F-M	N-J
100	0	0	0
300	0	0	0
1,000	0	0	0
3,000	0	0	0

NOTE.—See Note to table 1.

The compatibility method was designed to deal with cases in which some characters (in this case, sites) evolve so quickly that they are meaningless. With long sequences compatibility was as successful as the distance methods and was much better than parsimony; however, with shorter sequences it was inferior to parsimony, perhaps because it discarded too much of the limited data available.

Maximum likelihood was inferior to the distance methods, with very short sequences, but was considerably superior to any other method, with long sequences,

Table 4
Comparison of Methods When Substitution Rate Varies by Site

A. Accuracy of Topologies

		ALGORITHM				
SITES	MEAN *dT*	Pars	Comp	ML	F-M	N-J
100	4.492	0.05	0.47	−0.05	−0.28	−0.19
300	3.115	0.16	0.34	−0.35	−0.12	−0.03
1,000	2.367	0.26	0.21	−0.68	0.08	0.13
3,000	2.072	0.32	0.14	−0.81	0.13	0.22

B. Trees (of 50) Showing Bias in Topology

	ALGORITHM				
SITES	Pars	Comp	ML	F-M	N-J
100	9	4	0	7	6
300	21	10	5	21	17
1,000	26	26	11	26	27
3,000	29	29	15	27	29

C. Accuracy of Branch Lengths

		ALGORITHM		
SITES	MEAN *Bs*	ML	F-M	N-J
100	44,128.535	−2,148.75	1,207.21	941.54
300	39,979.094	−2,498.87	1,255.88	1,242.98
1,000	38,728.660	−2,454.44	1,169.76	1,284.68
3,000	38,362.602	−2,397.06	1,120.55	1,276.52

D. Trees (of 50) Showing Bias in Branch Lengths

	ALGORITHM		
SITES	ML	F-M	N-J
100	33	38	38
300	38	40	39
1,000	40	42	42
3,000	42	44	44

NOTE.—See Note to table 1.

and gave better branch length estimates. It was also less often biased than the other methods. Unequal rates per site violate a fundamental assumption of the likelihood algorithm used (Felsenstein 1981*a*), but it appears that the method is fairly robust to violation of this assumption. However, the likelihood results were still inferior to those of table 3, suggesting that unequal rates per site do interfere with tree estimation.

None of the three methods that estimated branch lengths gave reasonable results in this case, and all were severely biased. This is not surprising, as unequal rates per site drastically violate the assumptions both of maximum likelihood and of the distances that form the basis of the distance-matrix methods.

Correction of Negative Branch Lengths

Table 5 shows the results for one case, high constant mutation rate, comparing the performance of the distance methods with and without correction for negative branch lengths. The Fitch-Margoliash method was corrected by holding the negative branch or branches at 0 and making a constrained least-squares fit. Since this was done at each step in tree estimation, it could influence the final choice of topology, and table 5 shows that topology as well as branch length was estimated more accurately with this correction. The neighbor-joining method was corrected by setting the length of any branch estimated as negative to 0 and then reducing its sibling branch by the same amount. This does not change the topology, but it makes a small improvement in branch length estimation. Results with other mutation rates were comparable (data not shown).

Speed of Algorithms

An accurate algorithm may be effectively useless if it is too slow. For comparison purposes, we determined the run time of each algorithm on 10 data sets of 300 bp each, simulated under the low substitution rate, with

Table 5
Comparison of Distance Methods With and Without Correction of Negative Branch Lengths

A. Accuracy of Topologies

		ALGORITHM			
SITES	MEAN *dT*	F-M	F-M corr	N-J	N-J corr
100	2.444	0.61	−0.23	−0.19	−0.19
300	1.065	0.53	−0.18	−0.17	−0.17
1,000	0.48	0.20	−0.06	−0.07	−0.07
3,000	0.28	0.11	−0.03	−0.04	−0.04

B. Accuracy of Branch Lengths

		ALGORITHM			
SITES	MEAN *Bs*	F-M	F-M corr	N-J	N-J corr
100	1,795.483	292.69	−133.48	−72.60	−86.61
300	583.620	73.52	−28.11	−22.36	−23.05
1,000	169.589	6.20	−3.65	−1.27	−1.27
3,000	54.133	0.79	−0.59	0.10	0.10

NOTE.—For explanation of structure of table, see Note to table 1.
F-M = uncorrected Fitch-Margoliash method (allowing negative length branches); F-M corr = corrected Fitch-Margoliash method; N-J = uncorrected neighbor-joining method; and N-J corr = corrected neighbor-joining method.

10 and 15 taxa. We also timed the distance-matrix–generation program DNADIST, since construction of distance matrices is a necessary step in using the distance methods on sequence data. The results are presented in table 6. Even with the overhead needed to make distance matrices, NEIGHBOR is much faster than the other algorithms, while fastDNAml is extremely slow (the vast majority of this study's computing time was consumed by fastDNAml runs). fastDNAml has since been made much faster by algorithmic improvements (G. Olsen, personal communication), but for large data sets other methods may be more practical.

Table 6
Relative Speed of Algorithms

ALGORITHM[a]	TIME[b] (s) 10 Taxa	15 Taxa
NEIGHBOR	0.08	0.2
DNAPARS	3.5	14.9
DNACOMP	5.6	27.3
FITCH	6.6	39.7
DNADIST (C)	17.1	41.8
fastDNAml (C)	102.4	539.9

[a] All programs are Pascal versions from PHYLIP 3.4—except for DNADIST, for which the slightly faster C version from PHYLIP 3.5 was used, and fastDNAml, for which the C code was provided by Gary Olsen.

[b] Measured on a DECstation 5000/200 by using 10 data sets of 300 bp, except for the measurement for NEIGHBOR with 10 taxa, which was estimated using a run with 100 data sets to avoid rounding error. All other times are rounded to the nearest 0.1 s.

Discussion

Comparison with Other Studies

Previous studies have in general been smaller, with ≤1,000 replications per condition (Tateno 1985), as opposed to our 5,000; most have been closer to 100–500 replications per condition. Many studies examined methods other than the five analyzed here; we restrict attention to studies that include at least two of the same methods. The studies discussed examined trees of 4–32 taxa. With more taxa, all methods are expected to become less accurate; it is difficult to predict how the number of taxa will affect the relative accuracy of methods.

A number of studies (Li et al. 1987; Sourdis and Nei 1988; Jin and Nei 1990) have compared neighbor joining and parsimony. Although their results differ in detail, the general result was that neighbor joining is more accurate than parsimony, especially in the case of unequal rates per branch. This is in agreement with our results.

Hasegawa and Yano (1984) compared parsimony and maximum likelihood, and they found results similar to ours, with likelihood slightly better with equal rates per branch and substantially better with unequal rates per branch.

Saitou and Imanishi (1989) compared Fitch-Margoliash, neighbor joining, parsimony, and maximum likelihood. Their results were generally similar to ours, with two exceptions. In Saitou and Imanishi's study, maximum likelihood performed slightly worse than neighbor joining when rates were equal and performed slightly better when rates varied by branch. We found maximum likelihood to be more accurate than neighbor joining, even with equal rates.

Saitou and Imanishi (1989) also found the Fitch-Margoliash method to be less accurate than neighbor joining, whereas we found it to be slightly superior in most cases. This is apparently due to a difference between their and our implementations of the Fitch-Margoliash method. Saitou and Imanishi used a program that made one pass through the tree when estimating branch lengths (N. Saitou, personal communication), whereas FITCH makes multiple passes. This difference in implementation of the Fitch-Margoliash algorithm may account for its weaker performance in their simulations.

Branch Length Estimation

Even in cases where there was substantial difference in their ability to estimate topology, all three of the methods that estimated branch length did almost equally well at branch length estimation. This is probably because the branch score is mainly dominated by the accuracy of the long branches. All three methods were approximately equally good at reconstructing long branches; the difference visible in their dT scores involves ability to reconstruct short branches. None showed signs of bias, except in the case of unequal rates per site.

General Success of Algorithms

In the most favorable case studied (high uniform substitution rate with sequences of length 3,000), the most successful algorithm produced an error ~1 tree in 6. Clearly the general structure of the tree is quite accurately recovered with such data.

We tested three departures from this most favorable case. Lowering the substitution rate, so that there was very little variability in the input data, tended to downplay differences among methods—all methods were approximately equally good at recovering the true tree with relatively invariable data. Varying the rate across branches caused substantial problems for the parsimony-based methods while leaving likelihood and distance methods fairly accurate. It is noteworthy that the distance methods did not show signs of bias (by our conservative test), with short sequences, despite the result of Zharkikh and Li (1993) showing that neighbor joining (and, pre-

sumably, other distance algorithms) may be biased with short sequences, owing to errors in distance-matrix estimation.

Varying the rate among sites, on the other hand, caused all methods to become biased—apparently preferring another tree or trees to the true tree. This is caused by failure to correct for the additional multiple hits and convergence events that occur at quickly evolving sites. Since functional DNA is known to show rate variation among sites, this is an important consideration.

Several approaches could be taken to deal with unequal rates at different sites. In the parsimony-based methods, sites could be weighted to emphasize those with lower rates. Compatibility itself is an example of such a method, in which 0 weight is given to any site that requires more than the minimum number of changes. In this study, compatibility was fairly successful with unequal rates. Successive character weighting (Farris 1969) or threshold methods (Felsenstein 1981*b*), giving lessened but not 0 weight to sites that are not fully compatible with the tree, might be more successful, since they would not discard the potential information in the rapidly evolving sites. Preliminary testing of Farris's approach has been reported by Fitch and Ye (1992), but little is yet known about the usefulness of threshold or iterative-weighting parsimony methods.

The difficulties encountered by the distance-matrix methods, with unequal rates per site, are probably due to nonadditivity of the estimated distances. It is straightforward to show, for the simple case of two species, that, if data are generated using a mixture of two Kimura two-parameter models with different mutation rates, the estimated distance between the two species will be biased downward. With large numbers of sites the estimated distance will converge with certainty to this incorrect value. Thus, branch length estimation is biased in the two-species case, and there is no reason to believe that adding additional species will improve matters. Since branch length estimation is intimately involved with topology estimation for both of the distance-matrix methods considered here, branch length bias leads inevitably to topology inaccuracy.

One cure would be to have a distance that allowed for a mixture of rates of substitution at different sites. Jin and Nei (1990) describe a method for constructing distance matrices that assumes a gamma distribution of rates, and they show that this corrected method gives better results than the uncorrected method when sites are evolving unequally, even when the underlying distribution is not close to a gamma distribution. Olsen (1987) proposes a similar method, using a log-normal distribution. With properly corrected distance matrices, the distance methods should regain their consistency on this type of data.

One can also correct for heterogeneity of rate of substitution at different sites, in maximum-likelihood methods. J. Felsenstein and G. Churchill (unpublished data) have developed a maximum-likelihood method in which several classes of sites with evolutionary rates in a fixed ratio are assumed, and the likelihood is then calculated over all possible classes for each site. This could improve the performance of the likelihood method (at the cost of approximately multiplying its run time by the number of classes assumed). It requires an arbitrary decision about the number and relationship of rate classes, but the equal rate assumption of the current methods is equally arbitrary. An alternative method (Yang 1993) assumes that rates come from a gamma distribution. This approach is quite slow, but it may have room for algorithmic improvement.

We hope in future studies to be able to evaluate the success of these approaches at correctly inferring the phylogeny when rates vary across sites. In the meantime, what can be said about the use of phylogeny algorithms on real data sets? In cases where the rates per site and per branch are expected to be reasonably equal, all of the methods analyzed in this study perform quite well and can be expected to recover a correct or nearly correct tree from an adequately large data set. The suspicion of unequal rates among taxa is a strong reason not to use the parsimony or compatibility methods, and trees produced by parsimony or compatibility should be taken somewhat skeptically if unequal rates among taxa are likely to exist (though it is worth noting that the "fast" rate in our simulations was 10 times higher than the slow rate, a fairly extreme difference). When rates are unequal among sites, as in protein-coding sequences, all methods encounter difficulties (with maximum likelihood being the least affected). One possible test for such difficulties would be to construct trees independently for silent and nonsilent codon positions. If these trees do not agree with the tree from the entire data set, the latter should be regarded with skepticism. Our results suggest that adding additional sites is not very helpful in improving the estimates when per-site rates are unequal; either restricting attention to only one class of sites (silent or nonsilent) or using one of the correction methods described above may be more useful.

Acknowledgments

Jon Yamato made a large contribution to programing and running these simulations, for which a mere acknowledgment cannot be adequate. We thank Gary Olsen for providing an early version of fastDNAml, and we thank Naruya Saitou for readily explaining details of his implementations of phylogeny methods. We also thank Walter M. Fitch and an anonymous reviewer for helpful comments on the manuscript. This work was

supported by National Science Foundation grants BSR 89-18333 and DEB-9207558 to J.F.

LITERATURE CITED

ECK, R. V., and M. O. DAYHOFF. 1966. Atlas of protein sequence and structure. National Biomedical Research Foundation, Silver Spring, Md.

FARRIS, J. S. 1969. A successive approximation approach to character weighting. Syst. Zool. **18**:374–385.

FELSENSTEIN, J. 1978. Cases in which parsimony and compatibility methods will be positively misleading. Syst. Zool. **27**:401–410.

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

———. 1981*b*. A likelihood approach to character weighting and what it tells us about parsimony and compatibility. Biol. J. Linnean Soc. **16**:183–196.

———. 1988. Phylogenies from molecular sequences: inference and reliability. Annu. Rev. Genet. **22**:521–566.

FITCH, W. M. 1971. Toward defining the course of evolution: minimum change for a specified tree topology. Syst. Zool. **20**:406–416.

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

FITCH, W. M., and J. YE. 1992. Weighted parsimony: does it work? Pp. 147–154 *in* M. M. MIYAMOTO and J. CRACRAFT, eds. Phylogenetic analysis of DNA sequences. Oxford University Press, New York.

HASEGAWA, M., and T. YANO. 1984. Maximum likelihood method of phylogenetic inference from DNA sequence data. Bull. Biometr. Soc. Jpn. **5**:1–7.

HENDY, M. D., and D. PENNY. 1989. A framework for the quantitative study of evolutionary trees. Syst. Zool. **38**:297–309.

JIN, L., and M. NEI. 1990. Limitations of the evolutionary parsimony method of phylogenetic analysis. Mol. Biol. Evol. **7**:82–102.

KIMURA, M. 1980. A simple model for estimating evolutionary rates of base substitutions through comparative studies of nucleotide sequences. J. Mol. Evol. **16**:111–120.

KISHINO, H., and M. HASEGAWA. 1989. Evaluation of the maximum likelihood estimate of the evolutionary tree topologies from DNA sequence data, and the branching order in Hominoidea. J. Mol. Evol. **29**:170–179.

KLUGE, A. G., and J. S. FARRIS. 1969. Quantitative phyletics and the evolution of anurans. Syst. Zool. **18**:1–32.

LEQUESNE, W. J. 1969. A method of selection of characters in numerical taxonomy. Syst. Zool. **18**:201–205.

LI, W.-H., K. H. WOLFE, J. SOURDIS, and P. M. SHARP. 1987. Reconstruction of phylogenetic trees and estimation of divergence times under non-constant rates of evolution. Cold Spring Harb. Symp. Quant. Biol. **52**:847–856.

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, New York.

OLSEN, G. J. 1987. Earliest phylogenetic branchings: comparing rRNA-based evolutionary trees inferred with various techniques. Cold Spring Harb. Symp. Quant. Biol. **52**:825–837.

RZHETSKY, A., and M. NEI. 1992. A simple method for estimating and testing minimum-evolution trees. Mol. Biol. Evol. **9**:945–967.

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

SAITOU, N., and T. IMANISHI. 1989. Relative efficiencies of the Fitch-Margoliash, maximum-parsimony, maximum-likelihood, minimum-evolution, and neighbor-joining methods of phylogenetic tree construction in obtaining the correct tree. Mol. Biol. Evol. **6**:514–525.

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

SOURDIS, J., and M. NEI. 1988. Relative efficiencies of the maximum parsimony and distance-matrix methods in obtaining the correct phylogenetic tree. Mol. Biol. Evol. **5**:298–311.

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.

TATENO, Y. 1985. Theoretical aspects of molecular tree estimation. Pp. 293–312 *in* T. OHTA and K. AOKI, eds. Population genetics and molecular evolution. Japan Scientific Society, Tokyo; and Springer, Berlin.

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

ZHARKIKH, A., and W.-H. LI. 1993. Inconsistency of the maximum-parsimony method: the case of five taxa with a molecular clock. Syst. Biol. **42**:113–125.

STANLEY SAWYER, reviewing editor

Received June 8, 1993

Accepted December 28, 1993