BIOINFORMATICS

Vol. 17 Suppl. 1 2001
Pages S190–S198

Designing fast converging phylogenetic methods

Luay Nakhleh[1], Usman Roshan[1], Katherine St. John[2], Jerry Sun[1] and Tandy Warnow[1]

[1]Department of Computer Sciences, University of Texas at Austin, Austin, Texas, 78712, U.S., and [2]Department of Mathematics and Computer Science, Lehman College, CUNY, Bronx, New York, 10468, U.S.

Received on February 6, 2001; revised and accepted on April 2, 2001

ABSTRACT

Absolute fast converging phylogenetic reconstruction methods are provably guaranteed to recover the true tree with high probability from sequences that grow only polynomially in the number of leaves, once the edge lengths are bounded arbitrarily from above and below. Only a few methods have been determined to be absolute fast converging; these have all been developed in just the last few years, and most are polynomial time. In this paper, we compare pre-existing fast converging methods as well as some new polynomial time methods that we have developed. Our study, based upon simulating evolution under a wide range of model conditions, establishes that our new methods outperform both neighbor joining and the previous fast converging methods, returning very accurate large trees, when these other methods do poorly.

Contact: usman@cs.utexas.edu

INTRODUCTION

Performance studies of phylogenetic methods focus upon how accurately methods can reconstruct the unrooted underlying leaf-labeled tree (called the “topology”) under various model conditions. Recent research (Erdos et al., 1997, 1999; Huson et al., 1999a; Warnow et al., 2001) has developed a new class of phylogenetic methods, called *fast converging methods*, which provably recover the true tree topology with high probability given only polynomial length sequences. Earlier experimental studies have shown that some of these methods can recover significantly more accurate trees than standard methods, such as neighbor joining (NJ) (Sautou and Nei, 1987)– perhaps the most popular polynomial time method in phylogeny reconstruction. Since some of these fast converging methods are also polynomial time, they potentially provide a powerful alternative to NJ.

Our first simulation study confirms the observations that current fast converging methods (Csűrös, 2001; Huson et al., 1999a) can outperform NJ; however, our study also suggests they outperform NJ only for very large and evolutionarily divergent datasets. Under other conditions, the fast converging methods are much less accurate than NJ. With this in mind, we designed additional methods, some of which are provably fast converging. The best of these new methods are significantly more accurate than the previous fast converging methods; not only do they perform as well as NJ in our experiments, but they outperform NJ on smaller and less evolutionarily divergent datasets. Many of our new methods are polynomial time, and while slower than the NJ method, they still complete within a few minutes, even for datasets with hundreds of taxa.

The rest of the paper is organized as follows. First, we provide a review of the terminology that is used in the paper and a discussion of the theoretical results about fast-convergence. Second, we outline the experimental methodology. Third, we present our initial simulation study comparing two fast converging methods to NJ. Fourth, we discuss the development and performance analysis of our new methods. Finally, we address the performance of the methods on large trees and conclude with a discussion of the consequences of this study and suggestions for future research. In particular, we discuss how some of these methods can be used to provide excellent approximations to the maximum likelihood (or maximum parsimony) problems.

TERMINOLOGY & REVIEW

Models

The two models we use for the simulation study are the Jukes-Cantor model (JC) and Kimura 2-Parameter (K2P) model with a gamma distribution (K2P+Gamma). The JC and the K2P model (without the gamma distribution) are special cases of the General Markov (GM) model (Steel, 1994b).

The *Jukes-Cantor* (JC) model (Jukes and Cantor, 1969) is the simplest Markov model of biomolecular sequence evolution. In that model, a DNA sequence (a string over $\{A, C, T, G\}$) at the root evolves down a rooted binary tree T. The assumptions of the model are: (1) the sites (i.e., the positions within the sequences) evolve independently and

© Oxford University Press 2001

identically, (2) if a site changes state it changes with equal probability to each of the remaining states, and (3) the number of changes of each site on an edge e is a Poisson random variable with expectation $\lambda(e)$ (this is also called the "length" of the edge e). A JC tree is completely defined by the pair $(T, \{\lambda(e)\})$.

The *Kimura 2-Parameter* (K2P) model (Kimura, 1980) is a generalization of the JC model. As with JC, each site evolves down the tree under the Markov assumption, but there are two different types of nucleotide substitutions: transitions (mutations that change A to G or vice-versa, and C to T or vice-versa) and transversions (all other mutations). The probability of a given nucleotide substitution depends on the edge and upon the type of substitution. A K2P tree is defined by the triplet $(T, \{\lambda(e)\}, ts/tv)$, where ts/tv is the transition/transversion ratio; in our experiments, we fix this ratio to 2 (a standard setting).

These models describe how a single site (i.e. a position within the sequence at the root) evolves down the tree, and it is assumed that the sites evolve identically and independently. However, we can also assume that the sites have different rates of evolution, and that these rates are drawn from a known distribution. One popular assumption is that the rates are drawn from a gamma distribution with shape parameter α. We use $\alpha = 1$ for our experiments under K2P+Gamma. With these assumptions, we can specify a K2P+Gamma tree just by the pair $(T, \{\lambda(e)\})$.

Measures of accuracy

There are many ways of measuring error between trees, but when the trees are all constrained to be binary, the Robinson-Foulds (RF) measure (Robinson and Foulds, 1981) is the preferred technique. Each edge in a tree induces a bipartition on the set of leaves of the tree. The RF error is the proportion of bipartitions that are unique to each tree (i.e., the RF score is the normalized symmetric difference of the trees). When this value is 0, the topology of the trees are identical.

Statistical performance issues

We say that a phylogeny reconstruction method M is *statistically consistent* under a model of evolution if, for every model tree $(T, \{\lambda(e)\})$ and every $\varepsilon > 0$, there is a sequence length k such that M recovers the true tree with probability at least $1 - \varepsilon$, when the method is given sequences of length at least k generated on the tree T. Real data are of limited length. Therefore, the length k of the sequences affects the performance of the method M significantly. The *convergence rate* of a method M is the rate at which it converges to 100% accuracy as a function of the sequence length.

Absolute fast convergence

The largest and smallest edge-lengths clearly affect the sequence length needed by any method. So, we will examine the convergence rate issue by fixing arbitrarily the largest and smallest "edge-lengths" (see above). Once these bounds are fixed, we can consider the sequence length a method needs in order to recover the tree topology exactly with high probability. This sequence length "requirement" clearly grows with the number of leaves in the tree. Intuitively, we will say that a method is "fast-converging" if the sequence length that suffices in order to obtain the true tree with high probability is bounded from above by a polynomial in n. We now define this concept formally.

Since we examine several different models of evolution (e.g. JC and K2P+Gamma), we will let $\mathcal{M}$ denote the assumed model of evolution. We parameterize this model as follows:

DEFINITION 1. *Let* $f, g \geq 0$. *Define* $\mathcal{M}_{f,g} = \{(T, \{\lambda(e)\}) : \forall e \in E(T),\ f \leq \lambda(e) \leq g\}$.

We now define absolute fast convergence:

DEFINITION 2. *A phylogenetic reconstruction method* Φ *is* (absolute) fast-converging (afc) *for the model* $\mathcal{M}$ *if, for all positive* f, g, ε, *there is a polynomial* p *such that, for all* $(T, \{\lambda(e)\}) \in \mathcal{M}_{f,g}$, *on a set* S *of* n *sequences of length at least* $p(n)$ *generated on* T, *we have* $Pr[\Phi(S) = T] > 1 - \varepsilon$.

Previous fast converging methods

Several afc methods have been developed in the last years (see (Erdos et al., 1997, 1999; Huson et al., 1999a; Csűrös, 2001; Warnow et al., 2001)). Of these, HGT-FP (Csűrös, 2001) and DCM*-NJ (Warnow et al., 2001) are the most promising. We now briefly describe these methods.

HGT+FP: The Harmonic Greedy Triplets + Four Point Condition is a polynomial method which builds a tree by sequential insertion of taxa, using a quartet-based metric.

DCM*-NJ: DCM*-NJ is one of the "Disk-Covering Methods" (see also (Huson et al., 1999a,b; Warnow et al., 2001)). DCM*-NJ is not polynomial time, since it involves solving an NP-hard problem, although polynomial time versions of DCM*-NJ in which the NP-hard optimization problem is approximated by a greedy heuristic perform well in practice (Huson et al., 1999a). We build upon the design strategy of DCM*-NJ in order to develop our new afc methods (described below). Hence we will also describe briefly the two-phase structure of DCM*-NJ.

The input to DCM*-NJ is an $n \times n$ matrix d_{ij} of distances between each pair of sequences in the input.

Phase 1: For each $q \in \{d_{ij}\}$, compute a binary tree T_q. Let $\mathcal{T} = \{T_q : q \in \{d_{ij}\}\}$. (In order to ensure that each

T_q is binary, we use heuristics for refining incompletely resolved trees. We note that the specific heuristic we used in Huson et al. (1999b) differs from the one we use in the new afc methods in this paper.)

Phase 2: Select the best tree from $\mathcal{T}$.

The method used in the second phase of DCM*-NJ is the Short Quartet Support (SQS) method, which we now define. Let T be a tree on a set of taxa S, and let $Q(T)$ denote the set of trees induced by T on each set of four leaves; hence a quartet tree $t \in Q(T)$ if and only if the subtree of T induced by the taxa of t equals t.

DEFINITION 3. *Let d be a distance matrix on a set S of taxa. For a given quartet q on taxa from S, define $diam_d(q) = max\{d_{i,j} \mid \{i, j\} \subset q\}$. In other words, $diam_d(q)$ is the maximum distance between the taxa of q. For Q, a fixed set of quartets, we can define the set $Q_w = \{q \in Q : diam_d(q) = w\}$.*

DEFINITION 4. *Let T be a fixed tree leaf-labeled by a set S of taxa, Q a fixed set of quartets on S, and d the distance matrix on S. The* short quartet support *of T with respect to Q, denoted $sqs(T, Q)$, is $max\{w : \forall j \leq w,\ Q_j \subseteq Q(T)\}$.*

We now present a high-level version of SQS:

```
PROCEDURE SQS($\mathcal{T}$, $S$)
  • For each set of four taxa from $S$, compute the NJ quartet $q$; let $\mathcal{Q}$ be the set of all such quartets.
  • Return $T_i \in \mathcal{T}$ such that $sqs(T_i, \mathcal{Q})$ is maximum; if more than one such tree exists, return the one with the smallest index $i$.
```

Note that the short quartet support of a tree, as defined, is a fairly crude estimate of the quality of the tree; surprisingly, it is sufficient to ensure that DCM*-NJ is absolute fast converging. In fact, if we had picked any tree with maximum support, the result would have been a provably absolute fast converging method.

EXPERIMENTAL DESIGN

Simulation study

Simulation studies are the standard technique used in phylogenetic performance studies (see, for example, (Huelsenbeck and Hillis, 1993; Huelsenbeck, 1995; Kuhner and Felsenstein, 1994)). In a simulation study, a DNA sequence at the root of a model tree (i.e. tree topology with branch lengths) is evolved down the tree under some assumed stochastic model of evolution, such as the K2P or JC models. This process generates a set of sequences at the leaves of the tree. The sequences are then given to the phylogenetic reconstruction methods, with each method producing a tree for the set of sequences. These reconstructed trees are then compared against the model tree for topological accuracy. The process is repeated many times in order to obtain a statistically significant test of the performance of the methods under these conditions.

In our study, we have used model trees based upon biological datasets as well as randomly generated model trees. We have also explored performance under two different models: the JC model, and the K2P+Gamma model. Finally, unlike most previous studies ((Bininda-Edmonds et al., 2001; Csűrös, 2001; Hillis, 1996; Huson et al., 1999a) are some of the few exceptions), we have examined performance for a wide range of numbers of taxa, ranging from moderately large (50 taxon) trees to very large (1600 taxon) trees. Due to space constraints, we will only present a subset of our data, though we will discuss the variations we see in the results as well.

In order to obtain statistically robust results, we followed the advice of McGeoch (McGeoch, 1992) and Moret (Moret, 2001) and used a number of *runs*, each composed of a number of *trials* (a trial is a single comparison), computed the mean outcome for each run, and studied the mean and standard deviation over the runs of these events. This approach is preferable to using the same total number of samples in a single run, because each of the runs is an independent pseudorandom stream. With this method, one can obtain estimates of the mean that are closely clustered around the true value, even if the pseudorandom generator is not perfect.

The standard deviations of the mean outcomes in our studies is very low, less than 0.02. This is only two percent, since the possible values range from 0 to 1. We graph the average of the mean outcomes for the runs, but omit the standard deviation from the figures.

Model trees

We examined two types of model trees. The first type is random model trees, and the second type is biologically based model trees. Both are used in the phylogenetic performance literature.

Random Model Trees: For each number n of taxa, we randomly generated model tree topologies from the uniform distribution on binary n leaf trees (where the leaves are labeled by $1 \ldots n$). For each edge of each tree topology, we generated a random number (from the uniform distribution) between 1 and 100, and used that number as $\lambda(e)$, the expected number of changes on a random site. We then scaled each such "base" model tree by values between 0.01 and 0.0001. This process produces trees with average branch lengths of 0.5 and 0.005. Due to space constraints we will only show a subset of these experiments.

Biologically based Model Trees: A biologically based model tree is a rooted tree with branch lengths that are

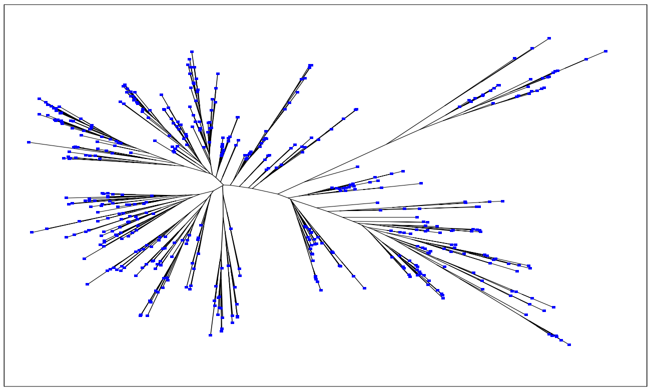

Fig. 1. The rbc*L* 500-taxon tree obtained by parsimony analysis by Rice *et al.* (Rice et al., 1997) of a collection of 500 rbc*L* gene (DNA) sequences.

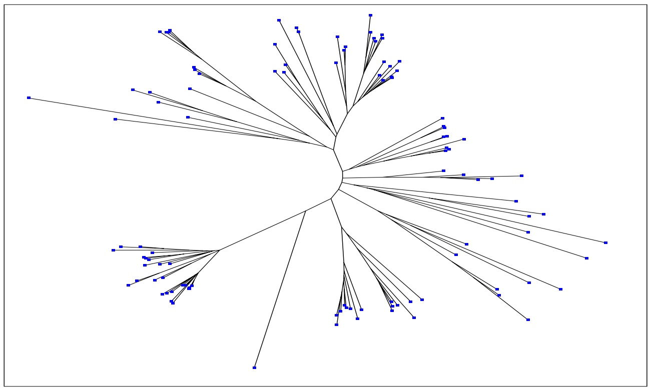

Fig. 2. The Archaea 107-taxon tree is from the Ribosomal Database Project (Maidak et al., 2000) and was constructed using Weighbor (Bruno et al., 2000).

inferred on the basis of a phylogenetic analysis of a real dataset. We have used several biologically based model trees in our studies. In each case, we used the model tree as a "base", and scaled the edge lengths of the tree up and down to produce a family of model trees, in order to test the performance of different methods under various conditions. Due to space limitations, we report on the performance for scalings selected so that NJ has only 20% error on sequences of length 1000. The trees we studied are:

500 rbcL tree: Our first biological model tree (see Figure 1) is based upon a parsimony analysis of a collection of 500 rbc*L* gene (DNA) sequences (the parsimony analysis was performed by Rice *et al.* (Rice et al., 1997)). This is the same model tree used by Csűrös in (Csűrös, 2001). In addition to the scaling factor described above, we also used the setting from (Csűrös, 2001).

107 Archaea tree: The second biological model tree is the Archaea 107 taxon tree (see Figure 2) obtained from the Ribosomal Database Project (Maidak et al., 2000). It was constructed using Weighbor (Bruno et al., 2000) from RNA sequences. This tree proved more challenging than the larger rbc*L* tree for many of the methods studied (see the New Methods section and the Very Large Datasets section).

85 Crenarch tree: We also studied performance on the Crenarch 85 taxon tree from (Maidak et al., 2000), constructed using Weighbor (Bruno et al., 2000) from RNA sequences. The performance on this tree echoed that on the Archaea tree, and we omit the details of those experiments (see the New Methods section for further discussion).

140 Eukaryote tree: This 140 taxon tree is a subtree of the Eukaryote 2055 taxon tree from (Maidak et al., 2000), constructed using maximum likelihood from RNA sequences. As with the 85 Crenarch tree, the performance on this tree echoed that on the Archaea tree, and we omit the details of those experiments (see the New Methods section for further discussion).

Experimental platform

Machines: The experiments were run over a period of approximately three months on approximately 280 different processors running the Debian Linux operating system. These included two clusters: the phylofarm cluster of 9 dual-processor machines, which are dedicated to the design and study of algorithms for phylogenetic reconstruction, and the SCOUT cluster: a cluster of 132 processors (16 4-way IBM Netfinity servers with 533-MHz Xeon processors and 1GB memory/box, 32 2-way IBM Netfinity servers with 733-MHz Pentium III processors and 512MB memory/box, 2 2-processor 733-MHz Netfinity boxes acting as file and checkpoint servers). The SCOUT cluster is funded by NSF EIA-9985991 and shared among five researchers. In addition, we also had nighttime use of approximately 150 Pentium III processors located in public undergraduate laboratories.

Software: We used the program `Seq-Gen` (Rambaut and Grassly, 1997) to randomly generate a DNA sequence for the root and evolve it through the tree under the JC model of evolution and the K2P + Gamma model. We calculate evolutionary distances appropriately for each model (see (Li, 1997)).

The software for DCM-NJ was written by Daniel Huson. To calculate the maximum likelihood scores of the trees we used PAUP* 4.0 (Swofford, 1996). To visualize the trees, we used the `splitstree` package (Huson, 1998). For job management across the cluster and public laboratory machines, we used the Condor software package (Condor, 2001). We generated the rest of this software (a combination of C++ programs and Perl scripts) explicitly for these experiments. The software for HGT+FP was provided by Csűrös.

COMPARING AFC METHODS TO NJ

Our first study focuses on the two most promising absolute fast converging methods under these two models of evolution. JC was chosen since the original studies (Csűrös, 2001) showing the HGT+FP method outperformed NJ on large trees with high evolution were done under this model of evolution. K2P+Gamma was chosen due to its popularity in many recent phylogenetic studies.

Comparison to NJ: We compared DCM*-NJ and HGT+FP to the popular neighbor joining (NJ) method of Saitou and Nei (Sautou and Nei, 1987). NJ is statistically consistent under the General Markov model of evolution. We do not know if NJ is absolute fast converging under these models (the only proven upper bound for its convergence rate is exponential (Atteson, 1999)).

Experimental Procedure: We compared these methods on a large number of model trees, both biological and random. We generated 50 sets of sequences of length 8000 under JC and then ran experiments on the first 200, 600, 1000, 2000, 4000 and 8000 sites of the same set of sequences.

Results & Discussion: The relative performance between the three methods is quite clear. We show only our results for the rbc*L* 500 tree experiments, due to space limitations.

On the rbc*L* 500 tree under JC (see Figure 3), our results confirm Csűrös' results (Csűrös, 2001) on the same tree, and show that HGT+FP can outperform NJ on this tree given long enough sequences, but is worse than NJ on shorter sequences. DCM*-NJ and HGT+FP both outperform NJ at sequence lengths above 4000, but NJ is better than DCM*-NJ and HGT+FP for sequence lengths below 4000. A comparison between DCM*-NJ and HGT+FP shows that DCM*-NJ has better performance than HGT+FP at all sequence lengths.

On the other model trees, the comparison was similar: HGT+FP was less accurate than DCM*-NJ, and the relative performance between NJ and these methods depended upon the number of taxa and the rate of evolution: as these parameters increased, NJ's performance decreased until the other methods were better than it (see the Very Large Datasets section for more details).

Summary: We conclude that these afc methods, HGT+FP and DCM*-NJ, can outperform NJ, but not consistently; they are often worse than NJ. In general it seems that they obtain improved performance only under restrictive conditions. In particular, even for those trees (mostly large and evolutionarily divergent trees) for which they do offer an advantage, the advantage seems to be limited to very long sequences.

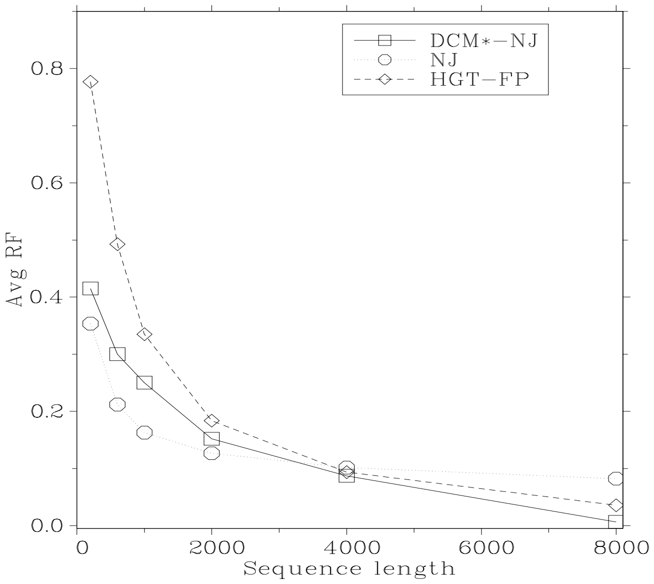

Fig. 3. DCM*-NJ vs. NJ vs. HGT+FP on the rbc*L* 500-taxon tree, under the JC model. Average branch length is 0.264.

NEW METHODS

In this section we describe our new phylogeny reconstruction methods. Our objective here is three-fold: first, the methods should be polynomial time and preferably as fast as NJ. In all cases, the methods must be fast enough that speed is not a consideration. Second, the methods should outperform both NJ and the previous fast-converging methods (with respect to topological accuracy) in an interesting portion of the parameter space. (For our concerns, we would like the new methods to outperform NJ and the previous fast converging methods on trees with just a few hundred taxa.) And lastly, the methods should not be worse than NJ or the previous fast-converging methods (with respect to topological accuracy) except in uninteresting portions of the parameter space (where NJ itself gets very poor reconstructions, such as missing 50% of the edges). Our earlier studies, including the ones we presented above, show that all the earlier afc methods (e.g. DCM*-NJ and HGT+FP) fail the last criterion.

Our new methods differ from DCM*-NJ in two ways. First, we obtain binary trees in Phase 1 in the following way. Given an unresolved tree, we assign DNA sequences to internal nodes using the Fitch maximum parsimony algorithm. Then we apply NJ to the neighbors around each unresolved node in order to resolve the node. Secondly, and more importantly, we modify the second phase, in which we select the best tree from the set $\mathcal{T}$ of trees $\{T_q\}$ (one for each $q \in \{d_{ij}\}$.

The importance of using a good technique to select a tree from a set of trees has been observed by others as well: the original HGT method also was based upon a two-phase structure in which a collection of trees is constructed, and then a best tree selected from the set. In (Csűrös and Kao, 1999), they used the Minimum Evolution Criterion to select the best tree from the set, and observed that it produced significantly better trees than their earlier techniques. In this section, we define three additional selection techniques:

- **Threshold Support (TS):** We define the threshold support of a tree T, denoted by $ts(T)$, as follows:
$$ts(T) = |\{w \in \{d_{ij}\} : Q_w \subseteq Q(T)\}|.$$
- **Maximum Parsimony (MP)**. The maximum parsimony score of a tree is obtained easily in polynomial time, see (Fitch, 1971).
- **Maximum Likelihood (ML):**. The Maximum Likelihood score of a tree is not easily computed, although heuristics exist (Steel, 1994a). We use the reasonably efficient heuristics for ML in PAUP* (Swofford, 1996).

See (Hillis et al., 1996) for a discussion of both maximum parsimony and maximum likelihood as selection criteria. Each of these techniques thus produces a different two-phase phylogenetic method, which we call DCM-NJ+SQS (this is DCM*-NJ), DCM-NJ+TS, DCM-NJ+MP, and DCM-NJ+ML, with the obvious meaning. Of these four phylogenetic methods, only DCM-NJ+SQS and DCM-NJ+TS are provably afc. While DCM-NJ+ML is also statistically consistent, we do not have any bound on its convergence rate; DCM-NJ+MP is not even statistically consistent under the simplest models (Felsenstein, 1978).

Comparing DCM-NJ Variants

Model trees and parameters: We studied the methods under all the biological trees and several random trees of up to 200 taxa. Due to space limitations, we report only on the performance on two biologically based trees: the 107 Archaea tree, and the 500 rbc*L* tree. The performance on the other trees was similar. We scaled the edge lengths of each tree up to create challenging conditions, with the average branch length of the 500 rbc*L* tree set to 0.278, and the average branch length of the 107 Archaea tree set to 0.143.

Dataset generation: For each model tree and parameter setting, we generated 50 sets of sequences each of length 16000 under the K2P+Gamma model. We then ran the experiments on the first 200, 400, 600, 1000, 2000, 4000, 8000 and 16000 sites on the same set of sequences.

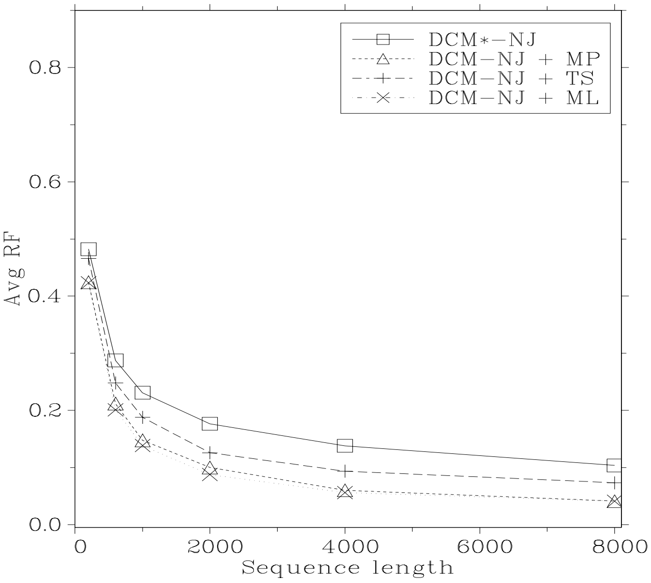

Fig. 4. Comparing variants of DCM-NJ on the Archaea 107-taxon tree under the K2P+Gamma model. Average branch length is 0.143.

Modification to DCM-NJ methods: In order to decrease running time, we modified the new methods to produce only a small subset of the possible trees, by restricting the set of $q \in d_{ij}$ to only 50 values, rather than the entire set of $\Theta(n^2)$ distances. Our brief experiments suggest that reducing the number of thresholds can reduce the topological accuracy, but generally not by much; furthermore, it greatly reduces the running time. Hence improved topological accuracy can be obtained by examining more, or all, of the different thresholds.

Discussion: A comparison between DCM-NJ+TS and DCM-NJ+SQS on the 107 taxon tree (see Figure 4) reveals that DCM-NJ+TS is an improvement over DCM-NJ+SQS. Other experiments (not shown) show DCM-NJ+TS consistently performs at least as well as DCM-NJ+SQS.

The distinction in performance between the four methods is noticeable on most of the trees (see Figure 4). Figure 5 shows good relative performance by all methods. In summary, it is clear that the optimal methods are DCM-NJ+MP and DCM-NJ+ML, followed by DCM-NJ+TS, and then by DCM-NJ+SQS. Furthermore, DCM-NJ+MP and DCM-NJ+ML are indistinguishable in most tests.

DCM-NJ+ML/MP vs. NJ

We then compared our best methods, i.e., DCM-NJ+MP and DCM-NJ+ML, to neighbor joining (NJ) and to HGT+FP. In all our experiments DCM-NJ+MP and DCM-NJ+ML were more accurate than the other methods.

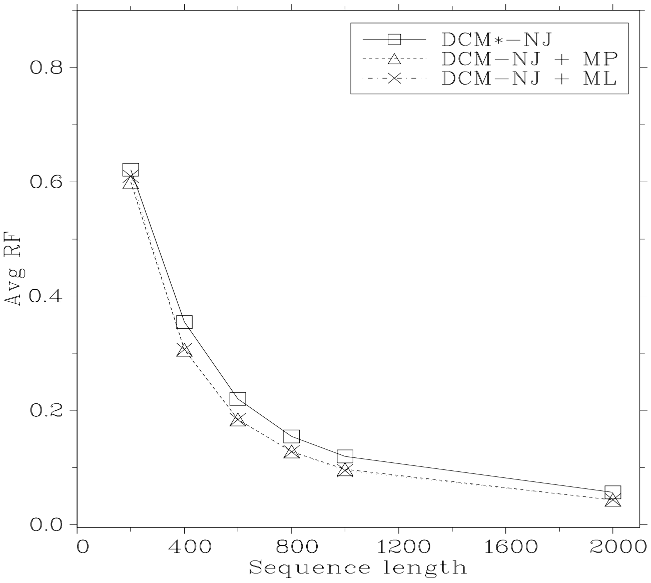

Fig. 5. Comparing variants of DCM-NJ on the rbc*L* 500-taxon tree under the K2P+Gamma model. Average branch length is 0.278.

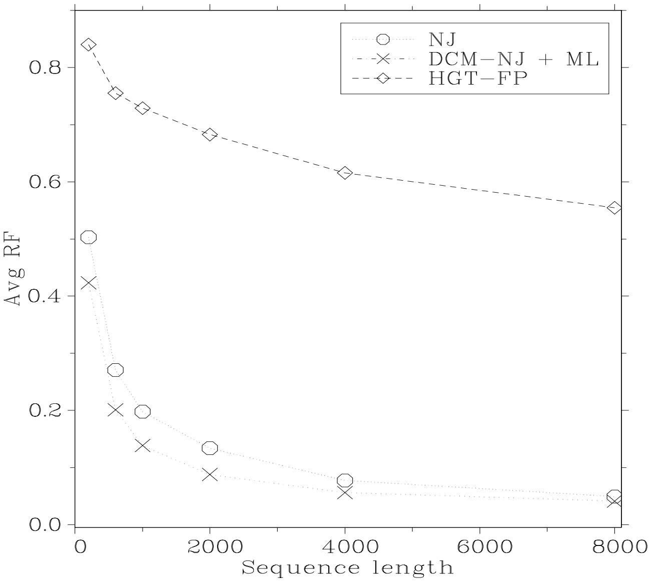

Fig. 6. DCM-NJ+ML vs. NJ vs. HGT+FP on the Archaea 107-taxon tree under the K2P+Gamma model. Average branch length is 0.143.

See, for example, Figure 6 and Figure 7. A comparison between DCM-NJ+MP and DCM-NJ+ML is interesting. In almost all our experiments they performed essentially the same (the small improvement obtained in Figure 4 is the greatest advantage we saw of ML over MP). This is interesting since DCM-NJ+ML is statistically consistent, and possible afc, while DCM-NJ+MP is neither.

VERY LARGE DATASETS

The earlier experiments show that DCM-NJ+ML (and DCM-NJ+MP) outperform both NJ and the earlier afc methods. However, we did not look at very large trees, that is, trees of more than 1000 taxa. In this section, we ask "How will topological errors grow with increasing numbers of taxa, if we fix the average branch length and the total sequence length available?" This question thus addresses the feasibility of inferring the tree-of-life, where the overall evolutionary distance and the number of taxa will both be large. We examine this by fixing the average branch lengths to two "nice" values.

Parameters: We generated 100 random tree topologies of 50, 100, 200, 400, 800 taxa and 10 topologies of 1600 taxa with random branch lengths selected so that the average branch lengths were either .05 or .005. For each tree topology we then generated sequences of length 1000 under K2P+Gamma model of evolution. Due to time constraints we could use only 10 runs for 1600 taxa.

Methods: We compared the error rates of DCM-NJ+MP, DCM-NJ+SQS, NJ and HGT+FP on each dataset.

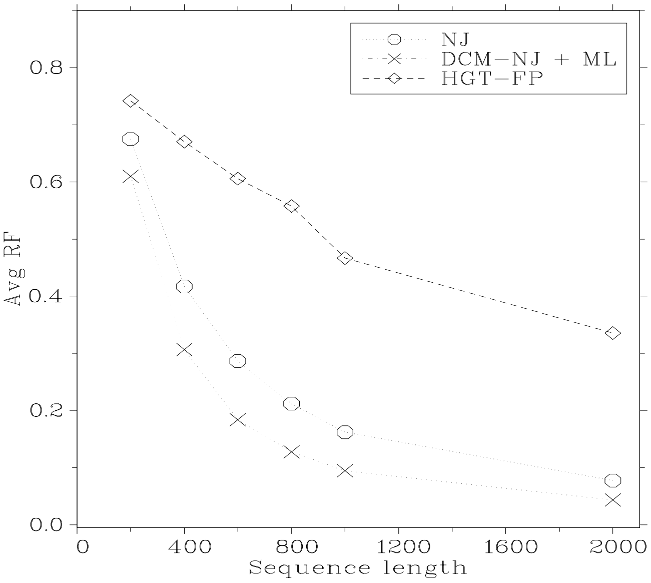

Fig. 7. DCM-NJ+ML vs. NJ vs. HGT+FP on the rbc*L* 500-taxon tree under the K2P+Gamma model. Average branch length is 0.278.

Discussion: In both experiments (the low branch length case, see Figure 8, and the moderate branch length case, see Figure 9), certain trends are clear. As the number of taxa increases, we see an increase in the error rate (the y-axis is the average RF error) for the NJ tree, but evidently no increase in error for HGT+FP nor for the

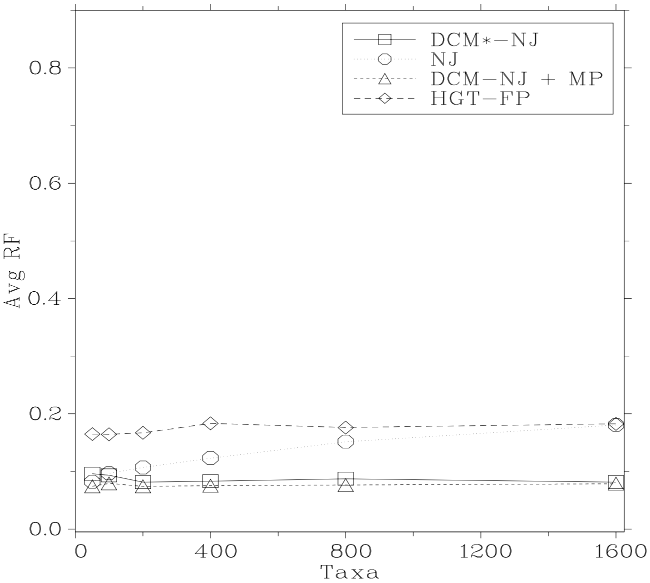

Fig. 8. DCM-NJ+MP vs. DCM*-NJ vs. NJ vs. HGT+FP on random trees under the K2P+Gamma model. Sequence length is 1000. Average branch length is 0.005.

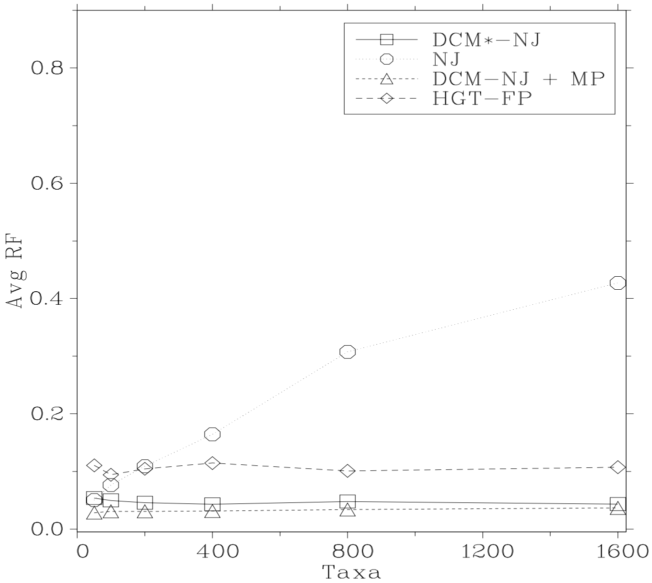

Fig. 9. DCM-NJ+MP vs. DCM*-NJ vs. NJ vs. HGT+FP on random trees under the K2P+Gamma model. Sequence length is 1000. Average branch length is 0.05.

two variants of DCM-NJ we study (i.e., DCM-NJ+MP and DCM*-NJ). The relative performance between HGT+FP and the DCM-NJ variants is clear: the best method is DCM-NJ+MP, followed by DCM*-NJ, and then followed by HGT+FP. The relative performance between NJ and the other methods depends upon the number of taxa and the rate of evolution. For the low branch-length trees, NJ outperforms HGT+FP until 1600 taxa, though the curve suggests that beyond this number NJ will be worse than HGT+FP. However, except for the 50 taxon case, NJ is worse than the DCM-NJ variants. For the moderate branch-length trees, NJ is much worse than the DCM-NJ variants throughout, and even worse than HGT+FP for the majority of the range.

The figures suggest that the relative advantage obtained by using DCM-NJ+MP *will increase* as the number of taxa increases. This means that truly large phylogenetic analyses which might not be feasible under NJ may be feasible using methods such as DCM-NJ+MP.

Finally, we wish to address the surprisingly flat curve for the error rates of HGT+FP, DCM*-NJ and DCM-NJ+MP. A flat error rate increase is impossible, as we know mathematically that all methods will have an increase in error as the number of taxa increases, due to the information content. We make, therefore, the following conjecture. Suppose that NJ's convergence rate is actually polynomial in n rather than exponential. (This would not contradict the theory in (Atteson, 1999), which is just an upper bound.) If this were so, then DCM*-NJ, DCM-NJ+TS, and perhaps even DCM-NJ+ML would have convergence rates that are bounded from above by a polynomial in $O(\log \log n)$ (see (Erdos et al., 1997, 1999; Huson et al., 1999a)) on random trees. The error curve of such a method might very well seem to be initially flat, as these do.

CONCLUSIONS

In all our experiments, DCM-NJ+MP and DCM-NJ+ML were at least as accurate as all the other methods we tested. This was true for all sequence lengths, all model trees, and all scalings. Furthermore, DCM-NJ+MP and DCM-NJ+ML were more accurate than the popular NJ method on a large portion of the parameter space. No earlier polynomial time method has been able to provide this kind of performance advantage, to our knowledge. Furthermore, these methods are polynomial time, and while slower than NJ, they are still fast enough to be acceptable. For example, DCM-NJ+MP completes its analysis on a 107 taxon tree in under three minutes.

FUTURE RESEARCH

There are several future research directions that we plan to take. First, the new methods that incorporate biologically significant optimization methods, such as maximum likelihood (ML) and maximum parsimony (MP), as part of the selection phase can be used as very fast heuristics for obtaining good initial starting points for ML or MP searches. Our experiments (data not shown due to space limitations) shows that these methods return much better MP and ML trees than the NJ tree returns, and almost as quickly. These optimization problems are of major interest to systematists, and these methods (or similar methods) may be very helpful.

More generally, the methods we have developed are all specific examples of a general phylogenetic-method booster. In fact, this research is part of an ongoing project to explore the power of the DCM-style methods, which began with (Huson et al., 1999a).

ACKNOWLEDGMENTS

The authors wish to thank the computer support staff at the UT-Austin Computer Sciences Department for their excellent assistance throughout the project. Katherine St. John is supported in part by NSF grant 99-73874 and the Texas Institute for Computational and Applied Mathematics. Tandy Warnow is supported by a fellowship from the Lucile and David Packard Foundation.

REFERENCES

Atteson, K. (1999). The performance of the neighbor-joining methods of phylogenetic reconstruction. *Algorithmica 25*, 251–278.

Bininda-Edmonds, O., S. Brady, J. Kim, and M. Sanderson (2001). Scaling of accuracy in extremely large phylogenetic trees. In *Proceedings of the Pacific Symposium on Biocomputing (PSB01)*, pp. 547–557.

Bruno, W., N. Socci, and A. L. Halpern (2000). Weighted neighbor joining: A likelihood-based approach to distance-based phylogeny reconstruction. *Mol. Biol. Evol. 17(1)*, 189–197.

Condor (Copyright 1990-2001). Condor high throughput computing program. Developed at the Computer Sciences Department of the University of Wisconsin; http://www.cs.wisc.edu/condor/.

Csűrös, M. (2001). Fast recovery of evolutionary trees with thousands of nodes. To appear in RECOMB 2001.

Csűrös, M. and M. Y. Kao (1999). $O(n^2 log L)$ - time accurate recovery of evolutionary trees with more than one thousand leaves: an experimental combination of harmonic greedy triplets and the minimum evolution principle. Preprint, Yale U.

Erdos, P., M. Steel, L. Székély, and T. Warnow (1997). A few logs suffice to build almost all trees –I. *Random Structures and Algorithms 14*, 153–184.

Erdos, P., M. Steel, L. Székély, and T. Warnow (1999). A few logs suffice to build almost all trees –II. *Theor. Comp. Sci. 221*, 77–118.

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

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

Hillis, D. M. (1996). Inferring complex phylogenies. *Nature 383*, 130–131.

Hillis, D. M., C. Moritz, and B. Mable (1996). *Molecular Systematics*. Sinauer Pub., Boston.

Huelsenbeck, J. (1995). Performance of phylogenetic methods in simulation. *Syst. Biol. 44*, 17–48.

Huelsenbeck, J. and D. Hillis (1993). Success of phylogenetic methods in the four-taxon case. *Syst. Biol. 42*, 247–264.

Huson, D. (1998). SplitsTree: A program for analyzing and visualizing evolution data. *Bioinformatics 14(10)*, 68–73.

Huson, D., S. Nettles, and T. Warnow (1999a). Disk-covering, a fast-converging method for phylogenetic tree reconstruction. *Comput. Biol. 6*, 369–386.

Huson, D., L. Vawter, and T. Warnow (1999b). Solving large scale phylogenetic problems using DCM2. In *ISMB99*, pp. 118–129.

Jukes, T. H. and C. Cantor (1969). Mammalian protein metabolism. (21-132).

Kimura, M. (1980). A simple method for estimating evolutionary rates of base substitutions through comparative studies of nucleotide sequences. *J. Mol. Evol. 16*, 111–120.

Kuhner, K. and J. Felsenstein (1994). A simulation comparison of phylogeny algorithms under equal and unequal evolutionary rates. *Mol. Biol. Evol. 11*, 459–468.

Li, W. H. (1997). *Molecular Evolution*. Sinauer, Massachuesetts.

Maidak, B. et al (2000). The RDP (ribosomal database project) continues. *Nucleic Acids Res. 28*, 173–174.

McGeoch, C. C. (1992). Analyzing algorithms by simulation: variance reduction techniques and simulation speedups. *ACM Comp. Surveys 24*, 195–212.

Moret, B. M. E. (2001). Towards a discipline of experimental algorithmics. To appear in Monograph in Discrete Mathematics and Theoretical Computer Science; http://www.cs.unm.edu/ moret/dimacs.ps.

Rambaut, A. and N. C. Grassly (1997). Seq-gen: An application for the Monte Carlo simulation of dna sequence evolution along phylogenetic trees. *Comp. Appl. Biosci. 13*, 235–238.

Rice, K., M. Donoghue, and R. Olmstead (1997). Analyzing large datasets: *rbcL* 500 revisited. *Systematic Biology*.

Robinson, D. F. and L. R. Foulds (1981). Comparison of phylogenetic trees. *Mathematical Biosciences 53*, 131–147.

Sautou, N. and M. Nei (1987). The neighbor-joining method: A new method for reconstructing phylogenetic trees. *Mol. Biol. Evol. 4*, 406–425.

Steel, M. A. (1994a). The maximum likelihood point for a phylogenetic tree is not unique. *Systematic Biology 43(4)*, 560–564.

Steel, M. A. (1994b). Recovering a tree from the leaf colourations it generates under a Markov model. *Appl. Math. Lett. 7*, 19–24.

Swofford, D. L. (1996). PAUP*: Phylogenetic analysis using parsimony (and other methods). Sinauer Associates, Underland, Massachusetts, Version 4.0.

Warnow, T., B. M. Moret, and K. S. John (2001). Absolute convergence: true trees from short sequences. *Proceedings of ACM-SIAM Symposium on Discrete Algorithms (SODA 01)*, 186–195.