Syst. Biol. 56(6):988–1010, 2007
Copyright © Society of Systematic Biologists
ISSN: 1063-5157 print / 1076-836X online
DOI: 10.1080/10635150701779808

Increasing the Efficiency of Searches for the Maximum Likelihood Tree in a Phylogenetic Analysis of up to 150 Nucleotide Sequences

DAVID A. MORRISON

Department of Parasitology (SWEPAR), National Veterinary Institute and Swedish University of Agricultural Sciences, 751 89 Uppsala, Sweden;
E-mail: David.Morrison@bvf.slu.se

Abstract.— Even when the maximum likelihood (ML) tree is a better estimate of the true phylogenetic tree than those produced by other methods, the result of a poor ML search may be no better than that of a more thorough search under some faster criterion. The ability to find the globally optimal ML tree is therefore important. Here, I compare a range of heuristic search strategies (and their associated computer programs) in terms of their success at locating the ML tree for 20 empirical data sets with 14 to 158 sequences and 411 to 120,762 aligned nucleotides. Three distinct topics are discussed: the success of the search strategies in relation to certain features of the data, the generation of starting trees for the search, and the exploration of multiple islands of trees. As a starting tree, there was little difference among the neighbor-joining tree based on absolute differences (including the BioNJ tree), the stepwise-addition parsimony tree (with or without nearest-neighbor-interchange (NNI) branch swapping), and the stepwise-addition ML tree. The latter produced the best ML score on average but was orders of magnitude slower than the alternatives. The BioNJ tree was second best on average. As search strategies, star decomposition and quartet puzzling were the slowest and produced the worst ML scores. The DPRml, IQPNNI, MultiPhyl, PhyML, PhyNav, and TreeFinder programs with default options produced qualitatively similar results, each locating a single tree that tended to be in an NNI suboptimum (rather than the global optimum) when the data set had low phylogenetic information. For such data sets, there were multiple tree islands with very similar ML scores. The likelihood surface only became relatively simple for data sets that contained approximately 500 aligned nucleotides for 50 sequences and 3,000 nucleotides for 100 sequences. The RAxML and GARLI programs allowed multiple islands to be explored easily, but both programs also tended to find NNI suboptima. A newly developed version of the likelihood ratchet using PAUP* successfully found the peaks of multiple islands, but its speed needs to be improved. [Large data sets; maximum likelihood; phylogeny; search strategies; tree islands.]

Maximum likelihood (ML) is a general statistical criterion in widespread use, and therefore it is no surprise that it has also been used for inference of molecular phylogenies for the past quarter of a century (Felsenstein, 1981). This criterion evaluates a hypothesis about evolutionary history in terms of the probability that the proposed model and the hypothesized history would give rise to the observed data (Huelsenbeck and Crandall, 1997). The presumption is that a hypothesized history with a higher probability of reaching the observed state is to be preferred to any hypothesized history with a lower probability. The method involves a search for the tree with the highest likelihood and therefore the greatest probability.

There are a number of reported advantages and disadvantages to the use of this criterion in evolutionary biology. However, here I do not address the issue of whether maximum likelihood is an appropriate criterion for reconstructing a phylogeny (e.g., whether it is the best estimate of the true tree) but simply the issue of how best to find the ML tree, presuming that this is the desired outcome. In this sense, "best" is considered to involve the ability to find the maximum likelihood (ML) tree within an acceptable time frame, where "acceptable" can vary from case to case.

The main obstacle to finding the ML tree is the frequent lack of speed of the computations (Sanderson and Shaffer, 2002). The application of the ML principle in phylogenetics is unusual in that the objective (the tree topology) is not an explicit part of the likelihood function (i.e., there is a different likelihood equation for each tree topology; Yang, 2006; Xia, 2006), and so there is no direct mathematical calculation for finding the ML tree. The only guaranteed strategy is to evaluate the ML value for every possible tree topology. Mathematically, this search problem is NP-hard (Roch, 2006), and so there is no known time-efficient and simultaneously thorough search strategy for nontrivial data sets. Therefore, heuristic strategies are used in most of the ML phylogenetic analyses performed in systematic biology (Bryant et al., 2005; Sullivan, 2005; Morrison, 2006).

Alternative tree-evaluation criteria, such as maximum parsimony and minimum evolution, are faster to evaluate than ML and so they can search tree space more thoroughly in a given time. Although it is possible to argue that the ML tree is "better" (in a specifiable sense) than the maximum parsimony or minimum evolution trees, it is more difficult to make a case that the result of a poor ML search is better than that of a more thorough maximum parsimony or minimum evolution search (Xia, 2006). If nothing else, the true evolutionary tree is likely to be within the confidence set of the ML tree (at least in principle, if the model used is adequate). The ability of a search strategy to find the globally optimal ML tree is therefore important.

Here, I undertake a comparison of a range of strategies designed to find the ML tree, and their associated computer programs, in terms of their success at locating the ML tree for realistic analyses of nucleotide sequences. It is important to note that I do not evaluate the computer programs per se but try to elucidate search strategies that seem to be most appropriate given a wide range of real data and heuristic algorithms—my focus is on the interaction with the data rather than on the programs themselves. Three distinct topics are discussed: the success of the search strategies, the generation of starting trees, and the exploration of multiple islands of trees.

Background

There have been several previous evaluations of strategies for finding the ML tree of nontrivial data sets (e.g., Takahashi and Nei, 2000; Guindon and Gascuel, 2003; Williams and Moret, 2003; Hordijk and Gascuel, 2005; Stamatakis, 2005, 2006; Stamatakis et al., 2005). However, these studies have concentrated on evaluation of a few computer programs in terms of finding a single "good" tree quickly. Here, I focus on finding strategies for locating the ML tree itself, no matter how long it takes; and I try to deal explicitly with the issue of multiple "islands" of near-optimal solutions. I use real data sets rather than simulated data, because only these data have the challenges that need to be met by realistic phylogenetic analyses (cf. Stamatakis, 2005, 2006; Stamatakis et al., 2005), and I have explicitly included multigene data sets, which offer a genuine challenge for ML analyses.

It is important to explore the characteristics of different tree-search strategies in relation to realistic data analyses, which involve multiple genes and islands as well as different numbers of sequences, because the successful design and use of heuristic strategies will depend on the particular characteristics of the solution space (Charleston, 1995; Davis et al., 2005), about which we currently know very little for maximum likelihood. No phylogenetic analysis can be considered to be complete after only one tree search, because we cannot then know how the results of that analysis relate to the set of possible alternative results, many of which may be better or at least only slightly worse.

In this context, I do not claim that I have necessarily found the ML tree for any of my data sets. For each data set, the evaluation of the search strategies is based solely on the best tree found to date. However, I have performed many analyses on each data set, using a variety of strategies, with the discovery of several islands of near-optimal trees, and so it seems probable that the best tree found so far is indeed the ML tree (cf. Davis et al., 2005). However, even if it is not, it still provides a suitable benchmark for comparison.

I focus on data sets with 50 to 150 sequences, which might currently be considered to be a medium-sized analysis. Data sets with <50 sequences can usually be analyzed in acceptable times even with inefficient search strategies (i.e., most strategies will converge on the same solution). The crux of the computational problem for 50 to 150 sequences is that the landscape of possible tree topologies can be very difficult to evaluate, so that users conventionally think that the tree searches will take a very long time (e.g., Mecham et al., 2006) and yet these are quite typical sizes for systematic studies (e.g., studies of genera/families). It is likely that data sets of this size will require qualitatively different approaches compared to smaller data sets, as the accuracy of heuristic searches decreases with increasing numbers of sequences (among other things). Analyzing data sets with >150 sequences certainly requires specialist approaches (Goloboff, 2002; Stamatakis, 2006). Alternatively, these larger data sets might be analyzable by divide-and-conquer strategies that break the data matrix into chunks with <150 sequences for individual analysis (e.g., Du et al., 2005).

I have used only programs that carry out reasonably comprehensive ML searches (i.e., I did not consider those that are mainly designed to evaluate user-supplied trees). I have also concentrated upon programs that run on a single processor, as this is still the main resource available to many (if not most) phylogeneticists. There are, however, currently multiprocessor implementations of several of the phylogeny programs that I have considered here.

There are a number of other computer programs that could have been used, but they did not provide at least the GTR+G model (general time-reversible model plus gamma-distributed rate variation across sites; see Morrison, 2006). The ML method of inference is available for both nucleic acid and amino acid data, as well as for several other molecular and nonmolecular data types. However, here I deal solely with nucleotide data.

THEORETICAL FRAMEWORK

There has been no recent review in the biological literature of methods for accelerating likelihood-based tree searches. So, I will start by summarizing those points that are directly relevant to this study, as well as providing a theoretical framework for evaluating the strategies.

The main idea behind phylogeny inference with maximum likelihood is to get estimates of (i) the parameters of the evolutionary (e.g., substitution) model; (ii) the branch lengths; and (iii) the tree topology (Bryant et al., 2005; Sullivan, 2005). All three of these estimations are time consuming and need to be reduced if an ML search method is to finish in a reasonable time. However, (i) and (ii) are the main bottlenecks in an ML analysis, because these optimizations need to be performed for each topology identified in (iii). There are a number of general ways to reduce the computation time of phylogenetic analyses (Sanderson and Shaffer, 2002), but here I focus specifically on ways to hasten the ML calculations themselves (see also Bader et al., 2006).

Several ways to speed up calculations of the substitution-model parameters (i.e., (i)) have been proposed: (a) prespecify the values based on prior calculations and leave them fixed during the entire tree search—e.g., use of a program such as ModelTest (Posada and Crandall, 1998), DT-ModSel (Minin et al., 2003), or Tree-Puzzle (Schmidt et al., 2002) to help derive reasonable estimates based on a quick tree or quartets; (b) alternate between tree searching and model estimation—e.g., reestimating the parameter values only after a specified number of topologies have been evaluated (Guindon and Gascuel, 2003); and (c) simplify the calculations—e.g., parameter approximations (Gu and Zhang, 1997).

Use of option (a) effectively reduces all calculations to a time requirement related to that of the JC (Jukes-Cantor) substitution model, which is the simplest nucleotide model. This is probably the most common form of analysis in phylogenetics (Sullivan et al., 2005), and it plays an important part in my use of several of the programs examined here, notably PAUP*. However, the alternative strategies are used by some of the other programs; this is explicitly noted at the time.

Speeding up branch-length estimation (i.e., (ii)) has received considerable attention (Yang, 2000), as there is no way to avoid performing the calculations for each topology assessed. Speed-ups include (a) performing an initial rough calculation, and only proceeding with the full optimization if the resulting score is close to the current known optimum—e.g., the ApproxLim parameter in the PAUP* computer program (Rogers and Swofford, 1998); (b) using calculation procedures that might be more efficient—e.g., Newton-Raphson optimization (Olsen et al., 1994) or Brent's rule (Guindon and Gascuel, 2003); (c) only updating a subset of the calculations for each new topology—e.g., conditional likelihoods (Guindon and Gascuel, 2003), subtree equality vectors (Stamatakis et al., 2002), column sorting (Kosakovsky Pond and Muse, 2004); (d) optimizing the branch lengths individually rather than simultaneously (Yang, 2000); and (e) optimizing the calculations for each substitution model (Bader et al., 2006).

Not all of the procedures that have been tried for (b) have turned out to be useful (e.g., Brent's rule), whereas methods (c) and (d) are commonly used. Option (a) is of particular relevance to this study, as use of the ApproxLim option in PAUP* seems to be an underappreciated aspect of that program (e.g., Artiss et al., 2001; Sallum et al., 2002). Nevertheless, it can have a dramatic effect on the efficiency of tree searches. Adjustment of the ApproxLim parameter must be done carefully, as described at http://hem.fyristorg.com/acacia/MLpaup.htm.

The only known exact method for ML topology search (i.e., (iii)) is to evaluate all possible trees. Branch-and-bound methods reduce the search space by eliminating some topologies that are guaranteed not to be optimal; but there is no simple way to calculate a good bound for ML (Hendy and Holland, 2003). For heuristic methods, the number of trees to be assessed can be reduced by considering only tree sets that, although not guaranteed to include the optimal tree, are very likely to do so. Strategies for this include (a) searching only in the neighborhood of some potentially optimal tree(s)—e.g., the most popular strategy is to use a stepwise-addition starting tree followed by branch swapping; (b) subdividing the taxa into groups, building the tree for each group, then joining the groups—e.g., quartet methods (Strimmer and von Haeseler, 1996; Vinh and von Haeseler, 2004), subtree methods (Vinh et al., 2005), disk-covering (Du et al., 2005); (c) using hybrid methods that combine likelihood evaluation with fast tree-building methods (Ota and Li, 2001; Ranwez and Gascuel, 2002; Jönsson and Söderberg, 2003; Hobolth and Yoshida, 2005); (d) generating a set of plausible candidate trees using some quick(er) method(s) and evaluating only those trees—e.g., the set of maximum parsimony trees (Ren et al., 2005), generalized neighbor-joining trees (Pearson et al., 1999), or Markov chain Monte Carlo trees (Suzuki et al., 2004); and (e) constraining the data by defining potentially monophyletic groups—e.g., constraining even one pair of taxa reduces the number of trees by $2 \times n^{-5}$ (e.g., 10,395 trees reduces to 945 trees for $n = 8$ taxa with 1 constrained pair).

Methods (a) and (b) are used by some of the programs examined here, as explicitly noted in Materials and Methods. I have paid particular attention to method (a) in terms of how best to generate the starting tree. Method (c) has not been evaluated here. Methods (d) and (e) require manual intervention by the user, and so I have not considered these in my experiments. Method (e), however, may actually be the most effective strategy from a biological perspective.

Landscape topography is usually used as the visual analogy to describe the solution space of phylogenetic trees. For ML, the topography forms what is called a likelihood surface, with parts of the solution space with near-optimal values being "hills," as illustrated in Figure 1.

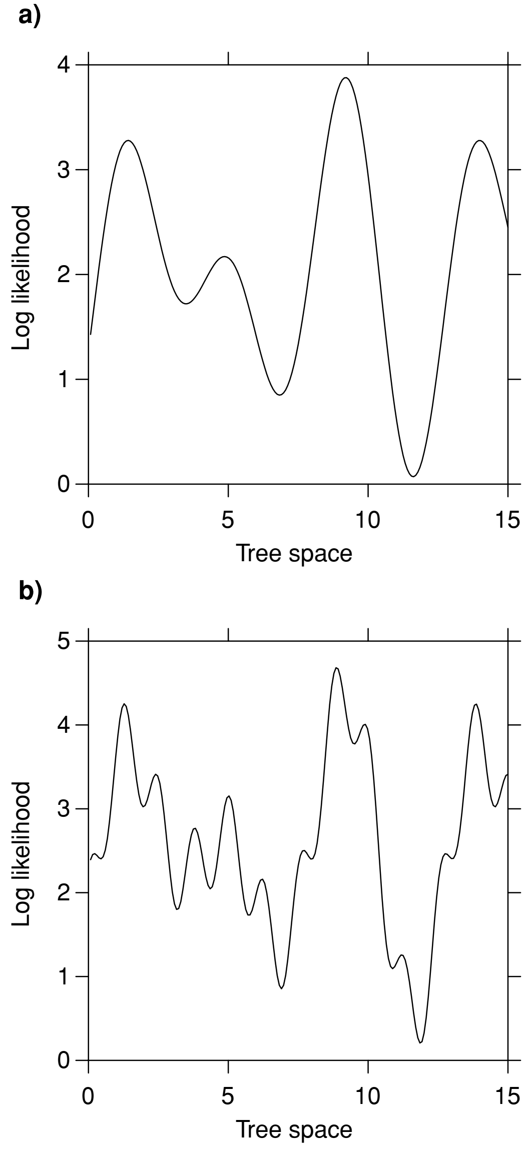

FIGURE 1. Diagrammatic illustration of maximum likelihood tree space showing (a) a relatively simple tree space and (b) a more complex tree space. The horizontal axis depicts a hypothetical arrangement of the trees (e.g., based on topological similarity), whereas the vertical axis depicts the negative log-likelihood score of each tree (using an arbitrary scale). The solid line represents the likelihood surface, which depicts the notion of hills or islands of trees.

The objective of any tree-search strategy is to move through this tree space in such a manner that it will find the top of a hill. The problem for heuristic strategies is that a single analysis will lead to the top of a single hill, and if this is not the largest hill then the strategy can be considered to have failed (i.e., it has found a local optimum rather than the global optimum). This means that either multiple searches are required (e.g., Johnson et al., 2003; Vos, 2003), which is the conventional strategy, or there needs to be some mechanism for the search to move from one hill to another, using strategies such as landscape perturbation (Nixon, 1999; Quicke et al., 2001), simulated annealing (Salter and Pearl, 2001; Stamatakis, 2005), or genetic algorithms (Matsuda, 1996; Lewis, 1998; Lemmon and Milinkovitch, 2002).

If the hills are relatively smooth (Fig. 1a), then it is likely that most heuristic search strategies will succeed in finding a near-optimal tree (i.e., the top of one of the hills) in any single search, whereas a rougher topography (Fig. 1b) is likely to result in failure unless the strategy is designed to deal with that roughness (Charleston, 1995). Increasing numbers of taxa will increase the roughness (Kirkup and Kim, 2000), which makes the search difficult for conventional strategies that use simple hill-climbing techniques (e.g., the early maximum parsimony programs such as Hennig86, Nona, PAUP, Phylip; see Goloboff, 2002; Davis et al., 2005). This leads to the basic problem that needs to be dealt with when analyzing 50 to 150 taxa—not only is it difficult to find the biggest hill, it can be difficult to find the top of any hill.

If a particular "water level" is chosen in this analogy with landscape topography, then the hills are turned into islands (Maddison, 1991). That is, all of the trees with optimality scores greater than that of the chosen water level constitute a set of discrete islands of trees. It can be cogently argued that it is the set of trees on an island, rather than merely the optimal tree on that island, that is of real interest in a phylogenetic study. Furthermore, the set of islands, and their relationship to each other, is also of interest, rather than simply the island with the globally optimal tree. Thus, any program that produces a solution that is near to the optimum is not necessarily useful unless it also explores the tree space of multiple islands, because there may be a huge number of trees that are near optimal. In this sense, a single tree is an arbitrary choice from the solution set (an arbitrary tree from an arbitrary island). Note that it is not absolutely necessary in a phylogenetic analysis to find all of the nearly optimal trees, but it is necessary to find all of the nearly optimal islands and to sample some of the trees from each of them (Goloboff, 2002).

This idea of islands has only recently been discussed for likelihood analyses (e.g., Salter, 2001; Johnson et al., 2003; Vos, 2003), although it has a long history for parsimony analyses (Maddison, 1991; Page, 1993; Charleston, 1995). When there are multiple islands, it will be inadequate to compare tree-search strategies based solely on the reported likelihood score, as has been done in the past, because we also need to know whether the reported trees are on the same island (i.e., how different they are). This is particularly important if sequences from only one gene are being used, because (as reported here) there are likely to be many islands for such information-poor data sets. However, conflicting signals in multiple genes may also induce tree islands, so that single-island data sets may only exist when there is a large amount of congruent data.

Exploring tree islands in the context of a ML analysis is something about which we know very little. Even defining an island is more problematic than for a maximum parsimony analysis, where the island can reasonably be considered to comprise only equally optimal trees (Maddison, 1991). Under this definition, in an ML analysis there is likely to be only a single tree at the island peak (i.e., almost all trees are different if the log-likelihood is measured to several decimal places), which on its own tells us little about the relationships among trees and islands. There thus needs to be an explicit definition of the water level in an ML analysis, even if the choice is arbitrary (Salter, 2001), because all peaks will be connected if the water level is set low enough. In this sense, I have identified island peaks in my analyses and referred to these as "islands," although they will be different peaks on the same island for some water levels.

It is also possible that the choice of substitution model will affect the composition of an island, by changing the rank order of the tree likelihoods. The complexity of the likelihood surface can therefore be evaluated only under a specified substitution model; so, I have performed (almost) all of my analyses under a single model.

The potential size of an island is determined by how the trees are "connected" to each other during the search process. In terms of branch swapping (Page, 1993), the trees can be connected by nearest neighbor interchange (NNI), subtree pruning and regrafting (SPR), or tree bisection and reconnection (TBR), which form progressively larger islands of trees (see Allen and Steel, 2001, for the appropriate formulae for the neighborhoods defined by these procedures). It therefore seems that the most straightforward way to evaluate the success of a tree-search strategy is to examine the NNI, SPR, and TBR island around the tree offered as a solution. This is what I have done in my experiments, in order to locate the peak of the island closest to each output tree. However, it is important to note that if the starting tree is far from any island peak, then the order in which the trees are evaluated can determine which island is estimated to be the closest, because the tree evaluation is a greedy procedure.

Given that most data sets of the size considered here are likely to have multiple islands, or at least multiple local optima that could trap a tree-search strategy, it is important to have a framework within which to consider heuristic search techniques. The one that I have used is to recognize that there are three possible steps for locating the optimal topology: (1) get a starting tree (preferably a near-optimal one); (2) move rapidly to an island (preferably the optimal one); and (3) find the optimal tree within that island. Davis et al. (2005) refer to this as a two-stage search procedure, as they do not separately distinguish the first step (they consider only maximum-parsimony analyses, where stepwise addition is the usual procedure for step 1).

The significant conceptual point here is that the conventional hill-climbing search strategy (Bryant et al., 2005; Sullivan, 2005; Kosiol et al., 2006), as embodied in the early ML programs such as Phylip and its derivatives (e.g., fastDNAml and its subsequent derivatives) as well as PAUP*, does not distinguish steps 2 and 3, but instead carries out extensive branch swapping as it proceeds. In practice, this means that step 2 may not be rapid, because there is a lot of unproductive branch swapping, but step 3 will be thorough, because a local optimum will always be reached. More recent techniques usually aim to perform step 2 rapidly, by quickly excluding unlikely trees, but they then risk being less thorough at step 3, perhaps also using the smaller NNI definition of an island (rather than SPR or TBR). Furthermore, some tree-finding techniques, such as star decomposition (Saitou, 1988) and quartet-puzzling (Strimmer and von Haeseler, 1996), do not explicitly carry out steps 2 and 3 but consider alternative trees as they build the initial tree. My aim here is to investigate some of the currently available strategies in the context of this three-step conceptual framework.

Finally, it is important to distinguish strategies that operate deterministically from those that operate stochastically. The former reproduce the same moves through tree space each time they are run, while the latter do not necessarily do so. Deterministic strategies are usually applied only once for any particular data set, whereas stochastic strategies are usually applied multiple times, possibly producing a different result each time.

TABLE 1. Characteristics of the three main data sets analyzed. The parameter values of the nucleotide substitution model are maximum-likelihood estimates based on the optimal tree found. The best-fitting model was determined by the AIC criterion using the ModelTest program (v. 3.7; Posada and Crandall, 1998).

Feature	NAD4	Isospora	HIV
Sequence type	Mitochondrial protein	Nuclear rRNA	Viral genome
Number of sequences	94	93	94
Number of genes	1 partial	1 complete	9 complete + noncoding
Aligned sequence length	411	2076	10,922
Distinct data patterns (GTR model)	106	1096	7661
Average % identity	96.2	91.4	83.0
Average % gaps/missing	0.0	19.4	16.9
Average % GC	21.8	45.6	41.7
Best-fitting model	TrN+G+I	GTR+G+I	GTR+G+I
Best log-likelihood found	−2,084.781	−18,502.343	−235,805.975
Nucleotide frequency			
A	0.329	0.267	0.396
C	0.093	0.184	0.183
G	0.103	0.246	0.216
T	0.475	0.304	0.206
Relative substitution rate			
A↔C	0.664	0.779	1.501
A↔G	47.719	2.421	3.871
A↔T	0.913	1.414	0.817
C↔G	1.020	1.370	0.930
C↔T	26.285	3.771	5.202
G↔T	1.000	1.000	1.000
Gamma shape	0.956	0.355	0.772
Proportion invariable sites	0.635	0.242	0.224

MATERIALS AND METHODS

Data Sets

Three empirical data sets were used for the most detailed analyses. These data sets differed in a wide range of features (Table 1), although they all had approximately the same number (93 to 94) of sequences. No attempt was made to choose "representative" data sets, and it is impossible to assess how typical these data sets might be. However, one sample was at the population level, consisting of short sequences with high identity (referred to here as "NAD4"), one was an interspecies sample of a single gene (referred to as "Isospora"), and one consisted of whole viral genomes (labeled "HIV"). Based on the results, it is likely that the biggest difference between these data sets is the ratio of phylogenetic information to the number of sequences, increasing from NAD4 to Isospora to HIV.

The NAD4 data set was based on that of Troell et al. (2006), whereas the Isospora data set was based on that of Morrison et al. (2004). The HIV data set was derived from the 2001 HIV-1 Subtype Reference Sequences Alignment of the HIV Sequence Database (http://hiv-web.lanl.gov/content/hiv-db/mainpage.html). In the latter two cases, some adjustments were made to the original alignments, aligning obvious motifs across groups of sequences as well as highly similar sequence pairs. The alignments are available as supplementary material on the Systematic Biology Web page (http://www.systematicbiology.org/).

An additional 17 data sets were used for a subset of the analyses, in order to test the generality of some of the conclusions derived from the above three data sets. These additional data sets covered a wide range of sequence numbers and sequence characteristics, as well as taxonomic groups (Table 2). Some of these data sets were chosen because they have previously been used to investigate islands of ML trees, whereas others were chosen to broaden the representativeness of the collection of data sets. In almost all cases, adjustments were made to the original alignments, as above. Also, identical sequences were deleted, as were sequences with large amounts of missing data (e.g., >50%).

It is worth noting here that these data sets were chosen to cover a range of what is commonly referred to as "phylogenetic information," even though there is no current biological or mathematical definition of this concept. Informativeness in the context of a phylogeny can be expected to vary with respect to sequence length (or, more specifically, the number of site patterns) as well as to conflicting signals in the data. In the absence of conflict, information is likely to increase with sequence length, but to decrease with increasing amounts of conflict (e.g., among different gene trees), or to remain the same if no additional synapomorphies occur among the extra characters. Here, I have used sequence length as a

TABLE 2. Characteristics of the 17 additional data sets, along with their analysis results. The sets are listed in order of increasing log-likelihood score. The best-fitting model was determined by the AIC criterion using the ModelTest program (v. 3.7; Posada and Crandall, 1998). The number of suboptimal runs is the number of times the search did not reach the SPR peak (out of 10).

Feature	VOS	EF1A	GRIM	AGAP	FUNG	RANA
Source	Vos (2003)	Sota and Sasabe (2006)	Grimm et al. (2006)	Johnson et al. (2003)	Hambleton et al. (2003)	Hillis and Wilcox (2005)
Organisms	Molluscs	Insects	Plants	Insects	Fungi	Amphibians
Sequence type	Nuclear rRNA	Nuclear protein	Nuclear rRNA	Mitochondrial protein	Nuclear rRNA	Nuclear rRNA
Number of sequences	56	86	105	22	76	64
Number of genes	1 complete	1 partial	3 complete	2 partial	1 partial	1 complete + noncoding
Aligned sequence length	584	475	851	943	1768	1976
Distinct data patterns (GTR model)	234	76	500	550	778	890
Average % identity	97.8	97.1	94.2	76.6	92.9	88.8
Average % gaps/missing	21.7	1.1	14.2	0.6	5.7	2.0
Average % GC	57.9	42.0	59.4	26.7	47.8	40.3
Best-fitting model	K81uf+G	HKY+I+G	TIM+I+G	GTR+I+G	GTR+I+G	GTR+I+G
PAUP* ApproxLim setting (%)	5	5	4	4	3	1
Best log-likelihood found	−1835.579	−1844.664	−6026.033	−9855.973	−15,580.006	−21,812.654
No. SPR islands explored	17	14	3	2	2	2
No. Ratchet (Nixon) islands	10	7	1	2	2	2
No. GARLI islands	7	10	3	2	1	2
No. suboptimal GARLI runs	10	9	10	0	0	1
	LEWI	HPV	DNAM	LITT	RYDI	BASI
Source	Lewis and Lewis (2005)	Salter (2001)	Stewart et al. (2001)	Olson et al. (2003)	Rydin et al. (2002)	Matheny et al. (2007)
Organisms	Algae	Viruses	Microsporidia	Flatworms	Plants	Fungi
Sequence type	Nuclear rRNA	Virus protein	Nuclear rRNA	Nuclear rRNA	Nuclear + chloroplast	Nuclear protein
Number of sequences	150	38	150	158	111	106
Number of genes	1 complete	1 complete	1 complete	1 complete + 1 partial	4 partial	2 partial
Aligned sequence length	1758	1518	1269	2644	5911	3317
Distinct data patterns (GTR model)	982	1065	1130	1408	3486	2467
Average % identity	93.9	67.8	81.8	90.1	87.6	75.7
Average % gaps/missing	4.3	2.5	4.8	0.3	28.5	18.1
Average % GC	48.2	42.0	46.6	51.3	47.3	52.5
Best-fitting model	GTR+I+G	TVM+I+G	GTR+I+G	GTR+I+G	GTR+I+G	TVM+I+G
PAUP* ApproxLim setting (%)	0.79	2	1.3	0.70	0.49	1.77
Best log-likelihood explored	−27,334.710	−33,790.994	−39,561.179	−62,069.395	−77,162.156	−134,207.688
No. SPR islands found	17	2	18	6	2	14
No. Ratchet (Nixon) islands	9	2	8	5	2	6
No. GARLI islands	10	2	10	6	2	9
No. suboptimal GARLI runs	9	1	3	1	1	4
	MURP	QIU	MAMM	JANS	ROKA	
Source	Murphy et al. (2001a)	Qiu et al. (2005)	Murphy et al. (2001b)	Jansen et al. (2006)	Rokas and Carroll (2005)	
Organisms	Mammals	Plants	Mammals	Plants	Fungi	
Sequence type	Nuclear + mitochondrial	Nuclear + chloroplast	Nuclear + mitochondrial	Chloroplast	Nuclear	
Number of sequences	66	99	44	29	14	
Number of genes	14 partial + noncoding	7 partial	18 partial	61 complete	106 complete	
Aligned sequence length	10,694	13,531	17,028	45,573	120,762	
Distinct data patterns (GTR model)	6687	8174	10,727	14,051	53,279	
Average % identity	85.9	92.5	85.3	87.3	68.4	
Average % gaps/missing	15.0	16.5	11.2	6.0	1.9	
Average % GC	46.1	48.7	49.2	39.8	42.0	
Best-fitting model	GTR+I+G	GTR+I+G	GTR+I+G	GTR+I+G	GTR+I+G	
PAUP* ApproxLim setting (%)	1.21	0.68	1	2	2	
Best log-likelihood explored	−170,821.926	−173,216.181	−216,857.573	−339,203.481	−1,156,290.763	
No. SPR islands found	1	2	1	1	1	
No. Ratchet (Nixon) islands	1	2	1	1	1	
No. GARLI islands	1	2	1	1	1	
No. suboptimal GARLI runs	0	0	0	0	4	

heuristic for informativeness, which assumes that in the original data collection longer sequences were intended to provide more phylogenetic information.

All ML evaluations were performed using the GTR+G+I nucleotide-substitution model (i.e., the general time-reversible model with among-site rate variation and a proportion of invariable sites; Morrison, 2006) even though this was not the optimal model for describing some of the smaller data sets (Tables 1 and 2). This strategy ensures experimental consistency, and makes the results directly comparable across the data sets. However, this implicitly assumes that all of the heuristics are affected in the same way by an inappropriate choice of model (or that the effects are randomized), in particular the possible overparameterizing of some of the data (i.e., trying to estimate too many parameter values from the available phylogenetic information). However, it has been shown for several published data sets that the tree produced by the best-fitting model (as determined by the AIC criterion) rarely differs from that produced using the GTR+G+I model (Morrison, unpublished data).

Computer Programs and Hardware

There are at least 20 computer programs currently available that are capable of assessing ML tree models for nucleotide data, but only 9 of these computer programs were used here. Not used were those programs that have not been designed for extensive tree searches, as they cannot be expected to locate the ML tree without considerable manual input from the user (i.e., by providing a set of trees to be evaluated). Also not assessed were those programs that do not implement at least the GTR+G nucleotide-substitution model in a straightforward manner (i.e., many of these programs do not extend beyond the HKY+I model). The programs used are described in the supplementary material on the Systematic Biology Web page (http://www.systematicbiology.org/). In almost all cases the programs were used with either their default option settings or those recommended by their authors (i.e., no attempt was made to optimize the programs for each data set).

Vos (2003) described a tree-search strategy based on reweighted tree perturbations that he called the "likelihood ratchet." However, instead of implementing a version of the "parsimony ratchet" of Nixon (1999), this strategy simply uses random character reweighting and the neighbor-joining procedure to generate a set of starting trees, which are then subjected to branch swapping until a specified time limit is reached. I explored a range of alternative combinations of (i) number of starting trees, (ii) branch-swapping strategy, and (iii) time limit. In most cases, the combinations were adjusted so that the total time to analyze each data set was approximately 24 hours. PAUP* was used for the analyses, and all of the command files were created using the PAUPRat (v. 1; Sikes and Lewis, 2001) computer program. In all cases, the parameters for the nucleotide-substitution model were fixed at the values determined for the optimal tree found using the successive-approximations strategy described in the supplementary material, and using the same ApproxLim percentages.

I also implemented a maximum likelihood version of the parsimony ratchet of Nixon (1999), in which random character reweighting is used to explore tree space by using the current estimate of the optimal tree as the starting tree for a round of branch swapping, followed by a round of branch swapping using the unweighted characters. I have called this the Ratchet (Nixon) strategy. Based on the results of preliminary analyses, searches were limited to 10 rounds of reweighting (with the successive-approximations tree being the initial tree), using only SPR branch swapping. Analyses were run with both 15% and 25% of the characters reweighted. PAUP* was used for the analyses, and all of the command files were created using the PAUPRat computer program. A fully commented template that can be used as the input file for the latter program is available as supplementary material on the Systematic Biology Web page. In all cases the parameters for the nucleotide-substitution model were fixed at the values determined for the optimal tree found using the successive-approximations strategy, and using the same ApproxLim percentages. Note that other programs could also be adapted to create the input file for the Ratchet (Nixon) strategy, such as PRAP (Müller, 2004) or perlRat (Bininda-Emonds, unpublished); and, indeed, PRAP now explicitly incorporates options for reproducing the ratchet analyses as described here (K. Müller, personal communication).

The analyses described here were performed on a range of different computer hardware, over a period of more than one year. The six computers used were (in order of speed) G4 iMac (1.0 GHz processor; 768 MB memory; OSX 10.3 operating system); G4 iMac (1.25 GHz; 256 MB; OSX 10.2; this computer could not run some of the data sets, due to memory limitations); G4 eMac (1.42 GHz; 512 MB; OSX 10.4); G5 iMac (1.8 GHz; 768 MB; OSX 10.3); Pentium4 PC (3.0 GHz; 1024 MB; Windows XP 5.1); Dual Xeon PC (2.66 GHz; 2048 MB; Red Hat Linux 9 i386; only one processor was used per analysis). These are typical of the range of computers that are likely to be available to a systematist.

Not only did these computers differ in speed, with the latter two completing analyses in less than one fifth of the time taken by the first computer, but they also differed slightly in the estimated likelihoods for the optimal trees. As discussed below, this is important in the case of PAUP*, which reports the likelihoods to five decimal places but for which the likelihoods usually differed in the third decimal place.

All analyses were performed serially, except for DPRml and MultiPhyl. That is, the parallel-processing versions available for some of the computer programs were not used. This is likely to be the most common way in which phylogenetic data are currently analyzed, although this is slowly changing. Furthermore, most of the computers were being used concurrently for other purposes (even if only occasionally), and so few direct comparisons of analysis speed (or memory usage) between programs or computers can be made. Any such comparisons discussed below refer to the 3.0-GHz Pentium4 PC.

Analyses

A three-step strategy was used for the evaluation of the three main data sets, involving a starting tree and a search for the optimal tree using each of the computer programs, followed by a thorough search of the nearby tree space to identify the optimal tree for the island found.

Most of the programs have a default strategy for obtaining a starting tree (step 1), which was used in all cases. The four commonly used strategies were a random tree, the neighbor-joining tree based on absolute differences (or the BioNJ tree), the stepwise-addition parsimony tree (sometimes followed by NNI branch swapping), and the stepwise-addition tree based on maximum likelihood. I therefore compared the ML score of these five trees directly using PAUP*, based on all 20 data sets. For the BioNJ tree I also compared 11 distance measures: P, JC, F81, TajNei, K2P, F84, HKY85, K3P, TamNei, GTR, and LogDet.

One strategy for exploring tree space in a heuristic manner is to use a range of starting trees, and so I evaluated various strategies for producing such a range. For the two stepwise-addition strategies, multiple trees can easily be generated by varying the input order of the sequences, which I did using PAUP*. For the neighbor-joining strategy it is possible to generate multiple trees using character weighting, as suggested by Vos (2003), which I did using the PAUPRat and PAUP* programs (as described above). It is also possible to use generalized neighbor joining, as described by Pearson et al. (1999), which retains multiple partial trees during the tree-building process, each of which leads to a different final tree. I used the default options of the GNJ (v. 1.0: Pearson et al., 1999) computer program except that I employed the neighbor-joining comparison criterion for evaluating the partial trees. Multiple trees can also be generated using a "random walk" of NNI interchanges from some central tree topology. For this strategy, I used the TreeFinder program (see the supplementary material) with the neighbor-joining tree as the center tree. For all of these strategies, I generated 200 trees, except for the random trees where I generated 1000 trees using PAUP*. The trees were compared using the Robinson-Foulds (symmetric difference) tree distance (Hillis et al., 2005).

The final trees from each computer program (step 2) were derived as described in the supplementary material. The ML value for each of these trees was then evaluated using PAUP* along with the fixed nucleotide-substitution model determined for the optimal tree found using the successive-approximations strategy (i.e., all tree topologies were compared using the same program and the same fixed substitution model). This was done in order to make the likelihood values directly comparable between analyses (Kosiol et al., 2006; Stamatakis, 2006), within the limitations of computer precision as described above. As a comparison, the trees for the NAD4 data set were also evaluated using the PhyML computer program, allowing the program to optimize the parameter values of the nucleotide-substitution model for each tree individually.

The exploration of the island around each final tree (step 3) was performed using PAUP*, separately employing NNI, SPR, and TBR branch swapping around that tree (i.e., three different definitions of an island for each tree), in order to locate the peak of the relevant island. In all cases the parameters for the nucleotide-substitution model were fixed at the values determined for the optimal tree found using the successive-approximations strategy, and using the same ApproxLim percentages. The diagrams of the tree islands were derived by using PAUP* to perform an SPR search that retained all trees with a likelihood exceeding a specified value (the chosen "water level"), starting from the optimal tree on each island and performing an SPR search around every tree retained. The trees found for each data set were then compared using the Robinson-Foulds (symmetric difference) tree distance and visualized using nonmetric multidimensional scaling (Hillis et al., 2005) via the Systat (v. 9.01; SPSS, 2000) computer program.

For the additional 17 data sets, testing the generality of some of the conclusions derived from the main three data sets, I used the GARLI computer program and the Ratchet (Nixon) strategy, both exactly as described above. These two search procedures proved to be among the most effective in the main analyses, either because of their proficiency at finding multiple islands of near-optimal trees or the speed of their analysis. For the Ratchet (Nixon), the percentage of reweighted characters was set to 25% for all data sets; and the ApproxLim percentages were as shown in Table 2. For the GARLI analyses, each tree found was subjected to a subsequent round of SPR branch swapping, in order to identify the peak of the associated SPR island.

Additional analyses were also performed on four of these extra data sets, based on the results from the initial analyses. The MURP data set was analysed using IQPNNI, MultiPhyl, PhyML, PhyNav, RaxML, and TreeFinder, exactly as described in the supplementary material (except that only the 10 hill-climbing runs were performed for RAxML). For the VOS, EF1a, and GRIM data sets, the GARLI and Ratchet (Nixon) analyses were repeated using the HKY+G nucleotide-substitution model.

Results and Discussion

Starting Trees

Step 1 of the maximum likelihood search strategy is to construct a starting tree, with preference typically given to one with a good ML score. Various trees have been proposed for this purpose, including a random tree, the neighbor-joining (NJ) tree based on absolute differences (including the BioNJ tree), the stepwise-addition parsimony tree (perhaps followed by NNI branch swapping), and the stepwise-addition ML tree. These five tree types are compared for all 20 data sets in Table 3.

Clearly, any of the standard methods produced a tree that was much closer to the optimal ML tree than was a random tree, but equally clearly these trees were still

TABLE 3. Log-likelihood score differences for various trees that could be used as starting trees for all 20 data sets. The scores are the difference between the log-likelihood of the starting tree and that of the best tree found for each data set (the "optimal" tree). The strategies as implemented in PAUP* are Random = random tree (average of 1000 trees), Stepwise-addition ML = default maximum-likelihood tree, Neighbor joining = default neighbor-joining tree, BioNJ = BioNJ tree based on absolute differences, Stepwise-addition parsimony = default parsimony tree, and Parsimony+NNI = default parsimony tree followed by nearest-neighbor branch swapping.

Tree-building strategy	NAD4	Isospora	HIV	VOS	EF1a
Random	1, 489.196	13, 630.833	85, 276.013	788.022	1, 439.948
Stepwise-addition ML	32.840	91.349	42.046	3.832	31.302
Neighbor joining	43.832	100.939	673.841	23.418	48.586
BioNJ	40.169	54.321	430.117	30.065	47.103
Stepwise-addition parsimony	50.864	89.088	214.642	2.154	61.494
Parsimony+NNI	—[a]	36.964	41.956	—[a]	—[a]
	GRIM	AGAP	FUNG	RANA	LEWI
Random	5, 175.521	740.978	8, 042.362	9, 623.949	15, 470.311
Stepwise-addition ML	3.960	12.327	17.744	28.187	226.882
Neighbor joining	37.643	23.034	90.270	63.632	241.937
BioNJ	41.536	28.348	87.732	65.862	256.003
Stepwise-addition parsimony	49.571	43.855	46.301	108.136	170.634
Parsimony+NNI	—[a]	43.855	—[a]	74.154	—[a]
	HPV	DNAM	LITT	RYDI	BASI
Random	6, 171.550	19, 821.418	29, 731.733	47, 921.197	11, 451.232
Stepwise-addition ML	8.775	609.800	116.898	117.306	316.900
Neighbor joining	12.626	767.518	161.155	951.437[b]	543.486
BioNJ	60.853	731.199	135.441	984.727[b]	532.150
Stepwise-addition parsimony	92.397	618.497	57.946	676.934	1, 002.151
Parsimony+NNI	89.138	—[a]	—[a]	—[a]	718.253
	MURP	QIU	MAMM	JANS	ROKA
Random	30, 959.949	46, 995.916	25, 734.244	80, 697.028	40, 843.174
Stepwise-addition ML	12.420	61.751	0.000	4.963	0.000
Neighbor joining	247.340	806.377	211.240	481.115	756.825
BioNJ	131.848	484.066	210.321	476.707	46.024
Stepwise-addition parsimony	177.445	99.714	335.619	103.364	1, 172.993
Parsimony+NNI	97.459	90.892	188.029	47.435	1, 172.993

[a] Large number of equal trees based on parsimony score.
[b] Several undefined distances, set to twice the maximum.

only a rough starting point. The stepwise-addition ML tree had the best ML score on average, although it was not always the best. The NJ and BioNJ trees were never the best tree, but the other three trees were for at least one data set. The NJ tree had the worst ML score on average, with the BioNJ tree faring much better. The stepwise-addition parsimony tree was very erratic, varying from the best score to the worst in different data sets. The parsimony+NNI strategy had a good score when it worked, but it failed for one third of the data sets, where it produced many trees with very different ML scores; in this case one must either evaluate all of the trees using ML and choose the best, or simply select an arbitrary tree. The NJ and BioNJ strategies can suffer from undefined distances, which PAUP* sets to twice the largest observed distance. On balance, the BioNJ tree was the most consistent performer after the stepwise-addition ML tree. This seems to justify its use as the starting tree in the IQPNNI, PhyML, and PhyNav programs, whereas RAxML uses the parsimony+NNI strategy.

For the three main data sets, none of these starting trees had particularly similar topologies to each other (Table 4), even when they had similar ML scores. Thus, the trees cannot be used as simple substitutes for each other. However, as a group they explored tree space quite well.

Under these circumstances, the ML score might not be the best criterion to use to judge the suitability of a starting tree. Instead, calculation speed might be a more appropriate criterion. In this regard, the stepwise-addition ML strategy was clearly the poorest one, as it took orders of magnitude more time than did any of the others (e.g., about 11 hours for the HIV data set on the fastest computer, compared to a few seconds for the other strategies).

I therefore explored the use of the BioNJ tree further by evaluating its performance with respect to 11 distance measures, for all 20 data sets (data not shown). The absolute distance (P) consistently produced trees with poor ML scores, whereas the 10 corrected distances fared much better (contrary to the findings of Tamura et al., 2004). The logdet distance (LogDet) produced the best ML score for 13 of the 20 data sets, but it also produced uniquely the worst score for 3 other data sets. The Tajima-Nei distance (TajNei), on the other hand, produced the best ML score for 7 of the 20 data sets and never ranked worse than 6th; it thus had the best average performance.

The importance of a starting tree is also related to the possible need to search multiple islands of near-optimal trees. One way of doing this is to use multiple starting trees that explore the appropriate parts of tree space. This issue is discussed in more detail in a later section.

TABLE 4. Robinson-Foulds (symmetric difference) distances among possible starting trees for the three main data sets. Single values are shown where there is only a single pair of trees, with ranges shown when there are multiple trees to compare. The "within strategy" distance is the distance among replicate trees.

Tree type	Stepwise-addition ML	Neighbor joining	Stepwise-addition parsimony	Within strategy
NAD4				
Stepwise-addition ML	—			
Neighbor joining	98	—		
Stepwise-addition parsimony	78	84	—	
Random[a]	144–148	178–182	152–156	174–182
ML (random order)[b]	45–74	76–101	54–81	21–79
Neighbor joining (weighted)[b]	52–77	55–93	50–71	5–69
Generalized neighbor joining[b]	80–116	54–120	76–110	2–138
Random walk[b]	98–118	22–40	84–110	22–78
Parsimony (random order)[b]	64–89	79–108	49–87	41–95
Isospora				
Stepwise-addition ML	—			
Neighbor joining	70	—		
Stepwise-addition parsimony	75	71	—	
Random[a]	168–172	176–180	175–179	172–180
ML (random order)[b]	32–73	47–83	54–86	17–88
Neighbor joining (weighted)[b]	47–68	33–53	52–72	4–51
Generalized neighbor joining[b]	58–98	18–100	65–93	2–100
Random walk[b]	74–100	24–40	75–103	24–80
Parsimony (random order)[b]	53–86	52–88	51–85	36–98
HIV				
Stepwise-addition ML	—			
Neighbor joining	56	—		
Stepwise-addition parsimony	32	54	—	
Random[a]	176–182	178–182	178–182	174–182
ML (random order)[b]	18–50	34–65	26–58	10–60
Neighbor joining (weighted)[b]	40–60	0–26	38–58	0–36
Generalized neighbor joining[b]	32–82	24–90	34–82	2–80
Random walk[b]	64–92	26–40	62–88	26–78
Parsimony (random order)[b]	18–54	42–72	20–62	16–66

[a] Based on 1000 trees.
[b] Based on 200 trees.

Finding the Optimal Tree

Step 2 of the search strategy is to move to the optimal island, whereas step 3 is to find the peak of that island. The results of separately performing these two steps are shown in Tables 5 (NAD4), 6 (Isospora), and 7 (HIV) for the three main data sets. These tables list the ML score for the single tree found by each of the deterministic strategies or the best tree found by each of the stochastic strategies (i.e., those that use multiple starting trees), along with the ML score for the optimal tree for the NNI, SPR, and TBR island closest to that tree.

The results for the NAD4 and Isospora data sets were qualitatively similar, whereas those for the HIV data set were quite different from these two data sets. So, I will first discuss the NAD4 and Isospora data sets together, and then highlight the differences found for the HIV data set.

For the NAD4 and Isospora data sets, the optimal tree (i.e., the best tree found, which is not guaranteed to be the optimal tree but is likely to be so) was found by only one method: the "successive approximations" strategy using PAUP*. Not too much should be made of the success of this particular strategy, because it was not successful for the HIV data set. However, it is worth noting that the iterative NJ+NNI+SPR search combination was at least as effective as the fixed SA+TBR combination (the default) and was a lot faster to compute.

The final tree (plus substitution model) was found for all three data sets after the first SPR iteration; that is, the second round of SPR swapping and the TBR swapping were redundant. Sullivan et al. (2005) found that more iterations were necessary before convergence to a single tree, but they started their iterations from a random starting tree. Second, although the changes in the substitution model were relatively small between iterations, they were not necessarily trivial, especially in the context of the number of nearly optimal trees found for all three data sets (see below). This is an important point, given that many phylogeneticists accept the fixed substitution-model parameter values derived from use of the ModelTest computer program (Posada and Crandall, 1998), which uses the neighbor-joining tree based on absolute differences. Even one round of NNI branch swapping would provide better estimates of the parameter values for the optimal tree, without necessarily greatly increasing the analysis time. Use of the BioNJ tree with the logDet (or Tajima-Nei) distance would also be beneficial.

For both the NAD4 and Isospora data sets, none of the other search strategies found the optimal tree. Indeed, almost all of the strategies found a unique tree. That is, none of the programs evaluated could find the best-known tree, unaided, using their default options, nor could they even find the same tree as each other. There are several possible explanations for this depressing result.

First, the calculation of the ML value of a tree often differs somewhat between the programs, so that it is theoretically possible that the optimal tree found by any one program may not be the optimal tree as assessed by another program. To examine this possibility, I evaluated each tree found for the NAD4 data set using both PhyML (allowing the substitution-model parameters to be optimized individually for each tree) and PAUP* (fixing the substitution-model parameters at the values for the best tree). In this comparison (Fig. 2), the rank-order of the trees does, indeed, differ between the two programs, although the same general ordering is observed. (Note that the scores are better for PhyML than for PAUP* because for PhyML the substitution model is optimized for each tree.) Nevertheless, the optimal tree according to both PhyML and PAUP* is not the tree found by PhyML using any of its three search strategies. Thus, it seems unlikely

TABLE 5. Maximum likelihood scores for the search analyses performed for the NAD4 data set. The strategies are as described in the text. NJ: neighbor-joining starting tree; SA: stepwise-addition ML starting tree; +NNI: search followed by nearest-neighbor interchange branch swapping; +SPR: search followed by subtree-pruning regrafting branch swapping; +TBR: search followed by tree-bisection reconnection branch swapping; best: best tree found from multiple starting trees. The optimal ML value is shown in boldface.

		Log-likelihood			
Program	Strategy	Tree found	+NNI	+SPR	+TBR
PAUP* 4.0b10	SA–NNI(10)^a	−2090.364	—	−**2084.781**	−**2084.781**
	SA–SPR(3) ^{a}	−2084.859	—	—	−2084.859
	SA–TBR	−2086.939	—	—	—
	NJ–Iterative	−**2084.781**	—	—	—
	Star decomposition	−2132.228	−2102.830	−2084.833	−2084.833
	Puzzling	−2188.069	−2125.096	−2087.271	−2087.271
Tree-Puzzle 5.2		−2240.428	−2118.336	−2084.859	−2084.859
IQPNNI 3.0		−2093.117	−2091.242	−**2084.781**	−**2084.781**
PhyNav 1.0		−2088.180	−2088.180	−2084.859	−2084.859
DPRml 1.0		−2097.834	−2094.177	−2084.859	−2084.859
MultiPhyl 1.0.6		−2111.740	−2102.417	−2084.833	−2084.833
PhyML 2.4.4	NNI	−2105.555	−2105.555	−2084.833	−2084.833
	NNI–SPR	−2093.847	−2089.958	−2084.859	−2084.859
	SPR	−2090.783	−2090.635	−2084.859	−2084.859
RAxML-VI 1.0	Hill climbing (best)	−2087.117	−2087.117	−2084.859	−2084.859
	Simulated annealing (best)	−2086.364	−2085.591	−2084.859	−2084.859
TreeFinder (May 2006)		−2102.114	−2097.820	−2086.939	−2086.939
GARLI 0.93	Random (best)	−2092.345	−2092.345	−2084.859	−2084.859
	Neighbor joining	−2087.130	−2086.996	−2084.859	−2084.859
Ratchet (Vos)	NNI(200)^a (best)	−2089.908	−2089.908	−**2084.781**	−**2084.781**
	SPR(200,5)^b (best)	−2095.121	−2094.808	−**2084.781**	−**2084.781**
	SPR(100,10)^b (best)	−2090.530	−2089.682	−**2084.781**	−**2084.781**
	TBR(100,13.5)^b (best)	−2092.254	−2091.729	−**2084.781**	−**2084.781**

[a] Number of random-addition sequences.
[b] Number of random-addition sequences, time limit in minutes for branch swapping each starting tree.

TABLE 6. Maximum likelihood scores for the search analyses performed for the Isospora data set. The strategies are as described in the text. Abbreviations are described in Table 5. The optimal ML value is shown in boldface.

		Log-likelihood			
Program	Strategy	Tree found	+NNI	+SPR	+TBR
PAUP* 4.0b10	SA–NNI(10)^a	−18,513.767	–-	−**18,502.343**	—
	SA–SPR(3)^a	−18,502.343	—	—	—
	SA–TBR	−18,503.237	—	—	—
	NJ–Iterative	−**18,502.343**	—	—	—
	Star decomposition	−18,863.270	−18,593.487	−**18,502.343**	—
	Puzzling	−18,674.783	−18,547.910	−**18,502.343**	—
Tree-Puzzle 5.2		−18,819.221	−18,620.729	−18,503.529	—
IQPNNI 3.0		−18,505.434	−18,505.434	−**18,502.343**	−**18,502.343**
PhyNav 1.0		−18,504.388	−18,504.388	−18,503.529	—
DPRml 1.0		−18,561.011	−18,538.438	−18,502.431	−18,502.431
MultiPhyl 1.0.6		−18,517.818	−18,512.837	−18,503.614	−18,503.614
PhyML 2.4.4	NNI	−18,507.281	−18,507.281	−18,503.529	—
	NNI–SPR	−18,507.515	−18,507.515	−18,503.529	—
	SPR	−18,523.807	−18,507.003	−**18,502.343**	—
RAxML-VI 1.0	Hill climbing (best)	−18,506.202	−18,503.898	−18,503.529	−18,503.529
	Simulated annealing (best)	−18,504.389	−18,503.535	−18,503.535	−18,503.535
TreeFinder (May 2006)		−18,507.390	−18,506.626	−**18,502.343**	—
GARLI 0.93	Random (best)	−18,502.347	−18,502.347	−18,502.347	—
	Random (2nd best)	−18,503.286	−18,503.286	−18,502.347	—
	NJ	−18,503.286	−18,503.286	−18,502.347	−18,502.347
Ratchet (Vos)	NNI(100,14)^b	−18,513.988	−18,508.323	−18,502.343	—
	SPR(100,14)^b	−18,523.723	−18,505.070	−18,503.529	—
	SPR(50,28)^b	−18,512.718	−18,503.737	−**18,502.343**	—
	SPR(25,57)^b	−18,505.281	−18,505.281	−18,503.529	—
	TBR (25,57)^b	−18,503.529	−18,503.529	−18,503.529	—

[a] Number of random-addition sequences.
[b] Number of random-addition sequences, time limit in minutes for branch swapping each starting tree.

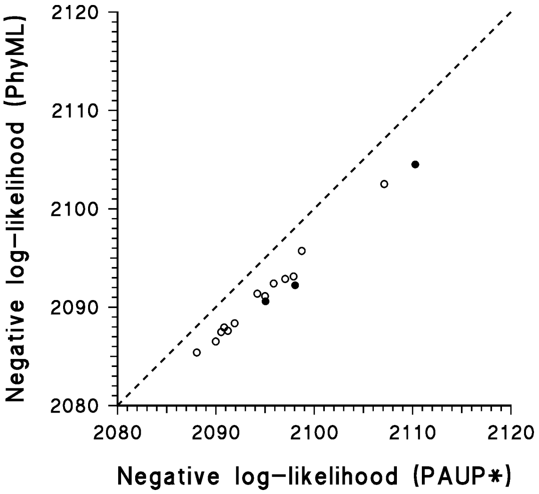

FIGURE 2. Comparison of the log-likelihood scores of the trees found by the various analyses of the NAD4 data set, as calculated by PhyML (allowing the substitution-model parameters to be optimized individually for each tree) and PAUP* (with the substitution-model parameters fixed at the values for the optimal tree). The dashed line represents identical scores. The filled symbols represent the trees from the three PhyML tree-searches. The tree from the Tree-Puzzle search cannot be shown at this scale.

that differences in likelihood calculations are generally responsible for the wide range of trees reported by the different programs.

Second, the topology of the tree space may be trapping the search strategies in a local (but not global) optimum. Alternatively, the tree-search strategy may not be finding any worthwhile ML optimum at all. Here, I will argue that there is strong evidence that these methods do find local optima, and that this also explains the difference in performance between the computer programs when comparing the HIV data set with the other two data sets. I evaluated these two possibilities by thoroughly exploring the space around each tree found by the computer programs, using NNI, SPR, and TBR branch swapping.

As shown in Tables 5 and 6, the trees found for the NAD4 and Isospora data sets were almost always at or close to the peak of a NNI island. That is, they are close to locally optimal based on the NNI definition of an island. However, few of the trees were close to the peak of their associated SPR island (except for those strategies that explicitly implement a SPR or TBR search). Indeed, there were relatively few SPR islands involved for either data set (Tables 5 and 6), implying that the search strategies were becoming trapped on different "hillocks" on the same few "hills." That is, the tree space for these two data sets looks more like that depicted in Figure 1b than in Figure 1a. Thus there are two issues involved: first, the search strategies could not locate the peak of the local SPR island; and second, some strategies were searching a suboptimal island. Only one of the search strategies happened to find the ultimate peak on the highest island.

TABLE 7. Maximum likelihood scores for the search analyses performed for the HIV data set. The strategies are as described in the text. Abbreviations are described in Table 5. The optimal ML value is shown in boldface.

		Log-likelihood			
TCHProgram	Strategy	Tree found	+NNI	+SPR	+TBR
PAUP* 4.0b10	SA–NNI(10)[a]	−235,806.123	—	−235,806.123	—
	SA–SPR(3)[a]	−**235,805.975**	—	—	−**235,805.975**
	SA–TBR	−235,808.839	—	—	—
	NJ–Iterative	−235,807.952	—	—	—
	Star decomposition	—[c]	—	—	—
	Puzzling	−238,164.092	−236,028.090	−235,808.839	−235,808.839
Tree-Puzzle 5.2		−239,809.839	−235,893.528	−**235,805.975**	—
IQPNNI 3.0		−235,811.440	−235,811.167	−235,808.839	—
PhyNav 1.0		−**235,805.975**	−**235,805.975**	−**235,805.975**	—
DPRml 1.0		−235,879.579	−235,807.445	−235,807.445	−235,807.445
MultiPhyl 1.0.6		−235,901.434	−235,873.005	−**235,805.975**	—
PhyML 2.4.4	NNI	−235,850.823	−235,847.978	−235,807.953	—
	NNI–SPR	−235,811.305	−235,810.556	−235,807.953	—
	SPR	−235,950.378	−235,813.461	−235,813.461	−235,813.461
RAxML-VI 1.0	Hill climbing (best)	−235,813.538	−235,806.994	−**235,805.975**	—
	Simulated annealing (best)	−235,821.100	−235,808.555	−235,806.123	−235,806.123
TreeFinder (May 2006)		−235,845.490	−235,845.490	−**235,805.975**	—
GARLI 0.93	Random (best)	−**235,805.975**	−**235,805.975**	−**235,805.975**	—
	(2nd & 3rd)	−235,806.123	−235,806.123	−235,806.123	—
	(4th & 5th)	−235,808.839	−235,808.839	−235,808.839	—
	NJ	−235,807.953	−235,807.953	−235,807.953	−235,807.953
Ratchet (Vos)	NNI(100,200)[b]	−235,880.082	−235,847.977	−235,807.953	—
	SPR(100,200)[b]	−235,910.309	−235,847.977	−235,807.953	—
	SPR(50,400)[b]	−235,857.830	−235,845.375	−235,807.953	—
	SPR(25,800)[b]	−235,807.937	−235,807.937	−**235,805.975**	—
	TBR(25,800)[b]	−235,810.556	−235,810.555	−235,807.953	—

[a] Number of random-addition sequences.
[b] Number of random-addition sequences, time limit in minutes for branch swapping each starting tree.
[c] Not attempted.

This conclusion implies that many of the computer programs are not really implementing step 3 of the general search strategy. That is, there seem to be two general search strategies currently available: some of the programs (e.g., PAUP*) perform a thorough search of part of tree space, which makes step 2 very slow but ensures that step 3 is implemented; and the other programs move rapidly to a local optimum on an island but do not finding the peak of that island. There thus seems to be a trade-off between steps 2 and 3, so that none of the programs implement both steps in an effective manner. Note that this conclusion does not address the issue of finding the optimal island, but only that of finding the peak of whatever island is being searched.

These issues become even more clear when comparing these two data sets with the HIV data set. In the latter case, several of the computer programs found the optimal tree unaided (Table 7), including: PAUP* with three random stepwise-addition sequences followed by SPR swapping; PhyNav with its single search; and GARLI with 10 random starting trees. Here, it seems that the tree space looks more like that depicted in Figure 1a than Figure 1b, so that finding the peak of the local SPR island is possible. However, it is also clear that the issue of finding the correct island still remains. For example, GARLI found three different islands with its five best trees (Table 7), finding the best island only once; and PAUP* found the best island only once among its three trees. That is, only PhyNav found the best island at one attempt.

The issue of multiple islands of trees that have very similar ML scores is therefore clearly an important one when searching for the maximum-likelihood tree. However, it is important to note for the discussion here that for ML an island must be defined in each case with regard to a specific branch-swapping strategy and ML score (the "water level"; i.e., the base of the island). In particular, performing a search while retaining all trees above the water level expands the size of the island compared to a single search around the island peak.

The islands for the three data sets are shown in Figure 3, based on somewhat arbitrary definitions of the water level for each data set. Not all of the peaks found by the tree searches are on distinct islands based on these definitions, sometimes instead representing multiple peaks on a single island (notably for the Isospora data set). It is evident that any thorough analysis of these data sets must include all of these islands. Simply reporting a single optimal tree (i.e., the highest peak from one island) would be very misleading in terms of representing a ML estimate of the true phylogeny. The most extreme case was presented by the LITT data set, discussed below, where six SPR peaks were found that differed by ≤ 0.8143 log-likelihood units. Setting the water level at 1 log-likelihood unit below the ML tree, for this data set there were >1500 trees on a single island.

Finally, for almost all of the analyses the peak of the TBR island was identical with the peak of the SPR island. That is, for ML analysis of all three of the data sets, a TBR search was redundant, because its result merely

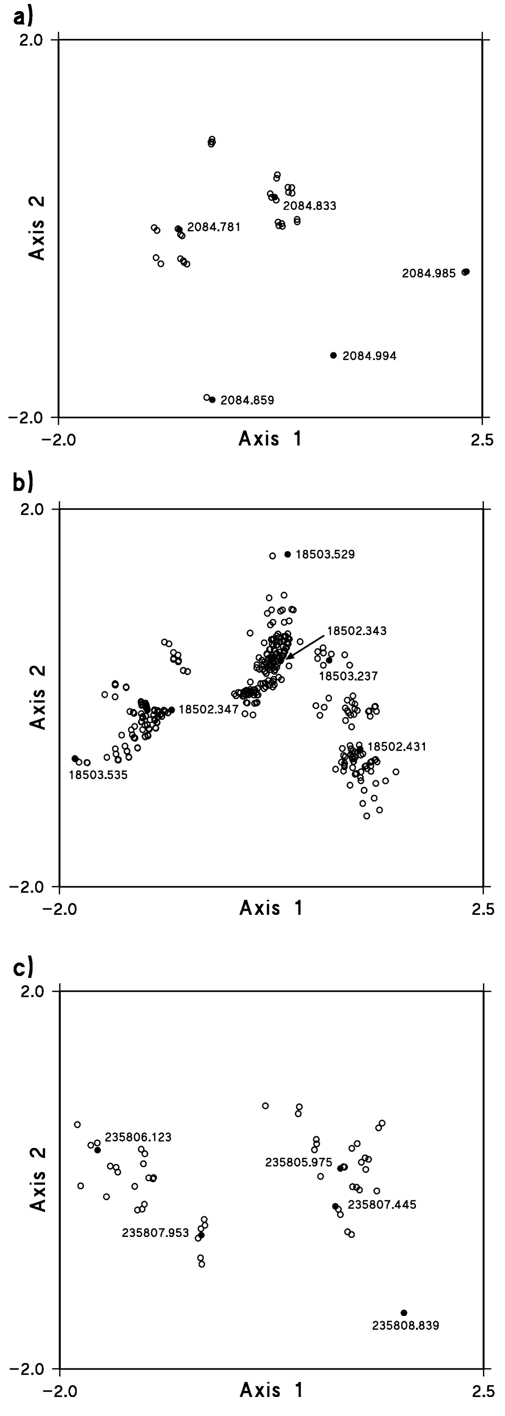

FIGURE 3. Nonmetric multidimensional scaling ordinations of the Robinson-Foulds distance between (a) 37 trees with –log-likelihood >2085.0 for the NAD4 data set, (b) 338 trees with −log-likelihood >18,503.6 for the Isospora data set, and (c) 54 trees with –log-likelihood >235,808.9 for the HIV data set. Each symbol represents a single tree, and the distance between symbols represents the Robinson-Foulds distance between those trees. The filled symbols represent the trees at the various SPR-island peaks, labeled with their –log-likelihood score. Note that not all of the solutions found by the phylogenetic analyses are shown here, due to the arbitrary cut-off value chosen for the "water level" defining the islands.

duplicated that of the SPR search (even though it examined a much larger set of trees). There were a couple of instances where a relatively poor tree was used as the starting tree for examining the island and the subsequent TBR search found a better peak than did the SPR search. There was also one instance where the SPR search found a better peak than did the TBR search. This counterintuitive result comes from the fact that the TBR search examines the trees in a different order to the SPR search, so that the two searches can potentially move to different islands as a result of the greedy nature of the search.

The important conclusion here is that a SPR search may be sufficient for ML analyses. For the data sets examined here, the TBR searches usually took four to five times as long as the SPR searches, making the SPR strategy considerably more efficient for arriving at the same result. Use of only SPR branch swapping thus seems to be a viable option for considerably speeding up ML analysis of nucleotide data. It could thus be the default option in programs such as PAUP*; and the absence of TBR branch swapping in programs such as Phylip (Felsenstein, 1989) may not necessarily be a handicap. If nothing else, several SPR searches (from different starting trees) will take less time than a single TBR search, and they are likely to be more effective at examining tree space.

Based on the abbreviated evaluations discussed below, I also analyzed the MURP data set using most of the different search strategies. I chose this data set because there appeared to be only one main island of trees (Table 2), and therefore this should be one of the easiest data sets for the programs to analyze.

All of these strategies found the same island, although they apparently disagreed as to which tree is the peak of that island (Table 8). There are three trees near this peak, being the three NNI rearrangements involving the position of the outgroup. GARLI, IQPNNI, MultiPhyl, and RAxML, along with the Ratchet (Nixon) strategy discussed below, chose one of these trees, PhyNav chose another, and PhyML and TreeFinder chose the third. PhyML and RAxML, which had multiple analyses, also chose other trees with one to four additional rearrangements among the terminal sequences; not all of these additional trees were necessarily on the same island. GARLI and the Ratchet (Nixon) strategy also used multiple analyses, but they consistently chose the same tree.

TABLE 8. Results of analyzing the MURP data set, showing either the single tree found or the range of trees found. The log-likelihood was determined by PAUP*, with the parameters of the nucleotide-substitution model fixed at the values determined for the optimal tree. The RF distance is the topological distance to the maximum-likelihood tree.

Method	Log-likelihood	RF distance
GARLI (10)[a]	−170,821.926	0
Ratchet (Nixon) (10)[a]	−170,821.926	0
MultiPhyl	−170,821.926	0
IQPNNI	−170,821.926	0
PhyNav	−170,822.978	2
TreeFinder	−170,823.352	2
PhyML (3)[a]	−170,823.352 to −170,833.513	2 to 6
RAxML (10)[a]	−170,821.926 to −170,866.165	0 to 10

[a] Number of analyses.

I evaluated the ML score for each of the three trees near the peak using IQPNNI, PAML (v. 3.15; Yang, 1997), PAUP*, PhyML, RaxML, and Tree-Puzzle. All of these programs agreed on the rank order shown in Table 8, except for PhyML, which ranked its chosen tree above the other two (it was ranked third by the other programs). Thus, at this level of resolution, we can expect differences among programs in likelihood evaluation to affect the choice of optimal tree—these different ML scores for the same tree may be equally valid theoretically. All of these programs evaluated the trees as differing by less than 1.5 log-likelihood units, and so distinguishing among them statistically is probably unreasonable. In this sense, all of the computer programs here (except PhyNav) performed much better than they did for the analyses of the three main data sets.

Individual Programs and Strategies

Obviously, not too many general conclusions can be made about the individual computer programs assessed in this study, based on only three or four data sets; nor was this the objective of the exercise. Nevertheless, some pertinent comments can be made about the programs, or more specifically the strategies they employ.

Star decomposition was the least successful of the strategies considered here, at least as implemented in PAUP*. The basic problem was not so much the likelihood score, although this was poor for the two analyses completed but simply the amount of time taken for processing. For the Isospora data set the analysis took 59 days of processing on the fastest computer, and so I made no attempt to analyze the larger HIV data set. This slowness seems to stem from two factors. First, for both data sets the number of trees evaluated was approximately the same as for TBR branch swapping (see Yang, 2006, for the formula for the number of star decomposition trees). Second, the ML score for every tree has to be evaluated fully rather than being approximated (i.e., ApproxLim does not apply).

Quartet puzzling was the next least successful of the strategies, both as implemented by PAUP* and by the Tree-Puzzle program. Indeed, the neighbor-joining tree with absolute differences and the stepwise-addition parsimony tree both had better ML scores for all three of the main data sets, in spite of the fact that they were produced in only a small fraction of the time. Moreover, quartet puzzling was a slower tree-search process than those implemented by most of the other programs. It is difficult, under these circumstances, to recommend this strategy for data sets of this size, in spite of the fact that researchers seem commonly to use it as a "quick" alternative to programs that employ extensive branch swapping. Interestingly, PAUP* produced a better puzzling tree for all three data sets than did Tree-Puzzle, although (based on the literature) Tree-Puzzle is far and away the most popular program used for quartet puzzling. One obvious difference between my use of the two programs was that Tree-Puzzle was used to optimize the parameter values of the GTR+G nucleotide-substitution model, whereas these were kept fixed for the PAUP* analysis using GTR+G+I.

The basic problem with the likelihood score for both the star decomposition and quartet puzzling strategies is that they do not explicitly implement the three-step search strategy discussed earlier. Both methods construct the final tree in a stepwise fashion that is effectively equivalent to step 1 of the three-step procedure. That is, there is no attempt to improve the tree that is initially built. Clearly, steps 2 and 3 are essential if we are to get close to the ML tree. All of the strategies discussed below employ some form of branch swapping in an attempt to improve upon their initial tree.

IQPNNI implements a branch-swapping development of quartet puzzling, and the results here indicate that it is a considerable improvement, operating more quickly and producing a tree with a better ML score. This conclusion concurs with the results of Vinh and von Haeseler (2004). Unfortunately, the results of the program were rarely competitive with those of the other programs evaluated, with the exception of the MURP data set.

PhyNav was in some ways the "surprise package" of the comparison. Its algorithm has not yet been published in the mainstream literature (Vinh et al., 2005), and the computer program does not appear to be in widespread use. Nevertheless, its results were always competitive with those of the other strategies assessed, and it was the only deterministic strategy to succeed with the HIV data set. That is, for this one data set it produced the "correct" answer with the least user effort and in the shortest time (i.e., the other successful strategies involved running several analyses only one of which succeeded). Not too much should be made of this latter result, because it may be fortuitous; and PhyNav was slightly out performed by some of the other programs on the MURP data set. Nevertheless, the consistency of the results across the four data sets is noteworthy. Further assessment of this package thus seems to be warranted, although its deterministic nature means that it cannot be used to explore multiple islands unless user-trees are used as the starting point. Both it and IQPNNI do, however, record the local optima encountered on the way to the final solution, which can be informative.

The PhyML program uses a deterministic strategy based on three possible branch-swapping regimes. Unfortunately, for all four data sets, the results were often relatively poor concerning the ML scores of the resulting trees. PhyML does not always rank trees in the same order as the other programs, and this may explain some of the behavior noted here. Moreover, none of the branch-swapping regimes seem to be superior to the others, as the SPR regime was most successful for the NAD4 data set, the NNI regime was most successful for the Isospora and MURP data sets, and the NNI+SPR regime was most successful for the HIV and MURP data sets. Indeed, there were large disparities between the results of the three strategies for two of the data sets. This diversity apparently contradicts the results of Hordijk and Gascuel (2005), who concluded that the SPR regime should generally be superior.

The search strategy in MultiPhyl is similar to that in PhyML, but I used SPR branch swapping rather than NNI. The results were very similar for the two programs, and consequently rather poor, as also found by Keane (2006). However, the way in which the likelihood scores are calculated is clearly different from PhyML, as MultiPhyl succeeded for the MURP data set.

The search strategy in DPRml is a straightforward development of that in the fastDNAml program, which in turn was derived from the Phylip package. The strategy is quite different to that used in PAUP*, as it uses NNI optimization at each step in the stepwise-addition sequence. DPRml actually performed better than MultiPhyl and PhyML for two of the three main data sets, which contradicts the results of Keane (2006). Furthermore, for two of these data sets (Isospora, HIV) DPRml explored a near-optimal island that none of the other search strategies discovered. It thus seems that this older search strategy can be quite effective under certain circumstances. Its main problem is that it is considerably slower than all of the recently developed strategies.

The RAxML program also produced somewhat disappointing results for the three main data sets considered here, although it did better for the NAD4 data set than did many of the other programs. The program's author originally recommended a somewhat complex stochastic strategy, with 10 iterations of a hill-climbing search followed by five iterations of a simulated-annealing search. This combined strategy took much longer to run than most of the other analyses, and the iterations tended to explore the same island. The simulated-annealing iterations took twice as long as the hill-climbing searches and produced very consistent results, although the output was superior for two of the three data sets. This combined strategy is not used in the latest version of the program (RAxML-HPC), thus simplifying matters and producing results faster.

The hill-climbing strategy produced somewhat variable results for the MURP data set, finding different trees within one NNI move on the same island, and once a different island, thus producing both the best and the worst trees found. This behavior by RAxML was also noted by Zwickl (2006); and it might result from the fact that the current versions do not explicitly use NNI rearrangements, thus opening up the possibility of missing the exact peak. The results were superior to those of PhyML for two of the three main data sets, as suggested by Stamatakis (2006) for empirical data.

TreeFinder uses an unpublished deterministic algorithm. Its success was "average" compared to the other programs, never being either the best or the worst for any of the four data sets. The analysis operates relatively quickly (e.g., equivalent to PhyML). It thus apparently produces results that are good approximations but are unlikely to be the true ML tree.

GARLI was consistently among the most effective programs, both in terms of exploring tree islands and in finding good trees on those islands. (Note that my use of different versions of the program seemed to have had little or no effect on the results.) The stochastic strategy requires multiple starting trees, with no obvious preference for any particular type of tree—the neighbor-joining tree was more successful for one of the main data sets but not for the other two, where random trees succeeded instead. The main limitation of the program seems to be a propensity for getting trapped in a local optimum when the data set has less phylogenetic information. That is, it finds NNI peaks rather than SPR peaks, and it only succeeds when these coincide. The speed of operation of the program seems to be related to the number of sequences rather than the number of characters (see below), so that the differences in size between the four data sets did not greatly affect the time taken by the analyses. This potentially makes the program very effective for medium-sized data sets.

The success of GARLI is not entirely surprising. The apparent success of genetic algorithms in phylogenetic analysis has encouraged more people to pursue this strategy than any other in relation to maximum-likelihood evaluation of sequences (Matsuda, 1995, 1996; Lewis, 1998; Katoh et al., 2001; Brauer et al., 2002; Lemmon and Milinkovitch, 2002; Jobb et al., 2004; Mak and Lam, 2004; Shen and Heckendorn, 2004; Oliveira, 2005; Poladian, 2005; Zwickl, 2006). Presumably, this active work will lead to further improvements.

The original idea for the Ratchet (Vos) strategy seems to have been to explore tree space using multiple starting trees (derived using character weighting and the neighbor-joining tree) along with limiting the search time used for branch swapping on each starting tree (on the assumption that the search will find the locally optimal tree relatively quickly and then spend a lot of time on a fruitless search around that tree). I found this approach to be unsuccessful for all three of the main data sets. The basic problem was the limit placed on the search time. In order to finish the analysis in a reasonable time, there is a trade-off between the number of starting trees and the length of swapping time on each of the trees. I found that fewer starting trees and longer swapping time was the better combination (contrary to the conclusion of Vos, 2003), but that very long swapping times were needed, with no way of knowing beforehand how long those times should be. Therefore, the strategy (as implemented by me) never found even locally optimal trees, let alone the globally optimal tree. The Ratchet (Nixon) strategy proved to be a more practical approach to the problem (see below). However, it is worth noting that adjusting the ApproxLim value, as discussed in more detail below, would definitely improve the practicability of the Ratchet (Vos) strategy.

Exploring Multiple Islands of Trees

One of the most important conclusions from the analyses described above is that for all three of the main data sets there were several distinct SPR islands of trees that had very similar ML scores (Figure 3). Therefore, even if a particular search strategy succeeds in finding the peak of a particular island, this single result is likely to be an inadequate representation of the data.

This idea has been rarely discussed for likelihood analyses, although it has received attention for parsimony analyses. Perhaps this dichotomy has resulted from the fact that the integer maximum-parsimony score often leads to many equally optimal trees whereas the real ML score leads to a single tree provided that enough decimal places are used. If this is so, then the distinction is illusory. First, likelihood calculations are not accurate to an infinite number of decimal places; for example, PAUP* produces different values on different computers to no better than three decimal places. Second, different computer programs evaluate the likelihood using different algorithms, and therefore can (indeed, will) produce different likelihood scores for the same tree. Third, the differences between likelihood scores can be trivial in a practical sense, so that there is no worthwhile mathematical or biological distinction between the trees.

Therefore, all likelihood analyses should explicitly consider the possibility that there are multiple islands each with multiple trees, and users should perform a search strategy that allows them to detect the peaks of those islands that do exist. Those islands can then be explored more thoroughly; for example, by using NNI or SPR branch swapping. There are two strategies that are known to be effective for exploring multiple islands of trees: perform a series of searches from different starting trees or use the ratchet method to move directly from island to island.

Of the programs investigated here, only GARLI, PAUP*, RaxML, and TreeFinder explicitly provide a method for producing multiple starting trees and can thus be used easily to explore multiple islands. However, it would be possible to provide multiple starting trees to programs such as IQPNNI, PhyML, and PhyNav by using the Vos (2003) character reweighting strategy to produce multiple BioNJ trees. I have not explored this latter option here.

The ratchet strategy is clearly the most efficient strategy for maximum parsimony analyses (Davis et al., 2005), but this does not mean that it is necessarily so for ML analyses. Indeed, use of this strategy for maximum likelihood seems never to have been reported in the literature. To this end, I devised a ratchet strategy based on the results of the above analyses. This strategy differs from the usual parsimony version (Nixon, 1999) by (i) using successive approximations to get the initial tree (neighbor-joining starting tree based on logdet distance, followed by NNI and then SPR branch swapping); (ii) performing only SPR branch swapping for all subsequent iterations (parsimony analyses use TBR branch swapping); and (iii) performing only one replicate of 10 iterations of character reweighting (parsimony analyses often use 10 replicates of 200 iterations). I varied the percentage of characters that were reweighted by comparing 15% and 25% (which covers the range recommended for parsimony analyses).

I then applied this strategy to the three main data sets, with the results shown in Table 9. As can be seen, in all

TABLE 9. Maximum likelihood scores for analyses of the three main data sets via the Ratchet (Nixon) strategy.

Percent of characters weighted	Island number	Log-likelihood of peak	Number of times found
NAD4			
15	1	−2084.781	8
	2	−2084.833	3
25	1	−2084.781	3
	2	−2084.833	3
	3	−2084.859	3
	4	−2084.994	1
	5	−2088.797	1
Isospora			
15	1	−18,502.343	3
	2	−18,502.431	4
	3	−18,503.237	2
	4	−18,503.529	1
	5	−18,504.297	1
25	1	−18,502.343	2
	2	−18,502.431	3
	3	−18,503.237	2
	4	−18,503.529	2
	5	−18,503.614	2
HIV			
15	1	−235,805.975	1
	2	−235,806.123	6
	3	−235,807.445	1
	4	−235,807.953	3
25	1	−235,805.975	4
	2	−235,806.123	1
	3	−235,807.953	6

cases the strategy did successfully move among islands of trees, finding up to five islands for the different data sets. It did, however, have a tendency to visit one island more often than the others, not necessarily the optimal island (but presumably the largest island). Moreover, the strategy preferentially found the better islands, although this is likely to result from using the best-known tree as the initial tree for two of the data sets. There seemed to be little to choose between the different reweighting percentages, although there is some hint that 25% performed better for the smallest data set (NAD4) and 15% for the largest (HIV).

Perhaps the major limitation of this strategy for large data sets was the time taken for the searches. For the HIV data set, each search took approximately 16 hours on the fastest computer, so that each iteration took 1.3 days. Thus, the entire analysis took 17 days, including the time to get the initial tree. This is not unreasonable, and it is certainly shorter than the "several months" that some users seem to believe ML analyses will take (e.g., Mecham et al., 2006). However, it does emphasize why the use of alternative and faster analyses is so prevalent in the literature.

I therefore did a series of trials with the 25% reweighted analysis of the HIV data, to see how much the searches could be accelerated by adjusting the ApproxLim value. A value of 0.90% allowed the analysis to finish in 9.5 days while still following the same search path among trees (i.e., it found the same islands in the same order). Values of 0.88% and 0.89% caused the searches to follow different paths, which did not always lead to the same final trees (i.e., missing some of the islands). A value of ≤0.87% caused the searches to "stick" on the original tree. Thus, with careful planning the analysis could be finished in <10 days. However, clearly the Ratchet (Nixon) strategy cannot easily be scaled to larger ML problems, the way it can be for parsimony analyses, because of the rapid speed decrease with increasing numbers of sequences. Unless the efficiency is greatly increased, it is not an option for hundreds, let alone thousands, of sequences.

To further explore the effectiveness of the Ratchet (Nixon) strategy, I compared its results to those from the GARLI program for the 17 additional data sets. I chose GARLI because it seemed to be one of the most effective of the computer programs previously evaluated, in terms of both processing time and ability to find multiple islands for this size of data set. I carefully adjusted the ApproxLim value for the Ratchet (Nixon) analyses to try to get each of the search times down below 10 days. It is important to note that the adjustment must take into account the reweighted searches as well as the unweighted searches—investigating just one type of search can lead to overreducing the value.

The data sets were deliberately chosen to represent a wide range of features for which phylogenetic analyses are known to vary, including the number of sequences and sequence length. In particular, three data sets seemed to have a relatively small amount of phylogenetic information (VOS, EF1a, GRIM) and can therefore be expected to be challenging to analyze, whereas five data sets had a large amount of information (MURP, MAMM, JANS, QIU, ROKA) and can be expected to be straightforward in terms of analysis (although not necessarily fast). The results are shown in Table 2.

The two strategies generally produced similar results in terms of which data sets they analyzed most effectively. Most of the apparently informative data sets (AGAP, FUNG, RANA, HPV, RYDI, MURP, QIU, MAMM, JANS) produced only a small number of islands, all of which were explored by both strategies. There was no consistency among the data sets as to whether the GARLI or the Ratchet (Nixon) strategy explored the most islands. However, several of the data sets that I originally assumed to be informative behaved differently to the others.

First, the ROKA data set had relatively few sequences but many nucleotide positions from many genes and was thus presumably the most informative data set analyzed. Nevertheless, GARLI succeeded in finding the SPR peak in only 6/10 iterations. This may be a result of the relatively small percentage identity among the sequences (68%) compared to the data sets where GARLI succeeded (85% to 93%), although the identity of the AGAP data set was also relatively small (77%). Alternatively, there may be conflicting signals among the genes that are affecting GARLI. This apparent failure thus needs further investigation.

Second, the LEWI and DNAM data sets produced a large number of trees, with very little overlap in the islands explored by each of the two search strategies.

For the LEWI data set, the Ratchet (Nixon) found three islands that were better than the best GARLI island, whereas for the DNAM set GARLI found both the best and the worst island but with Ratchet (Nixon) performed better on average. For both data sets, many of the GARLI iterations failed to find the peak of the SPR islands (Table 2), and for the DNAM data set PAUP* did not always agree with GARLI about the tip of the SPR islands, producing exactly the same ML score as for the GARLI tree but choosing a tree one to four RF steps away. The Ratchet (Nixon) thus performed better than did GARLI for these two data sets.

It seems likely that the complex nature of the tree space for these data sets was created by relative lack of phylogenetic information—1500 aligned nucleotide positions may simply be insufficient information to reliably reconstruct a tree for 150 sequences, even in the absence of conflicting signals. Increasing the number of sequences means both more internal branches and shorter branches to be estimated (for a constant tree diameter and depth), which thus combine to require more data. Indeed, simulations have shown that, even without conflict, sequences need to be several thousand nucleotides long in order to reliably construct phylogenies for hundreds of taxa (Hillis, 1996). In this regard, the performance of the BASI and LITT data sets was instructive, as both were multi-gene data sets with >100 sequences, but they seemed to have somewhat simpler tree space with fewer islands (Table 2). Indeed, after accounting for missing data and gaps, the effective sequence length for each of these data sets was about 1000 nucleotides longer than for the LEWI and DNAM sets. Moreover, the effective length of the RYDI data set (also >100 sequences) was another 1500 nucleotides longer, and this data set had only two islands, which were explored by both search strategies. Here, phylogenetic information does indeed seem to have been added with increased sequence length.

This simplistic conclusion about informativeness calls into question current attempts to use a single gene, such as small subunit (SSU) nuclear rDNA, to reconstruct the evolutionary history of large numbers of taxa (hundreds or thousands of sequences), no matter how sophisticated the tree-search strategy might be. The data sets used here indicate that about 500 aligned nucleotides are needed for a smooth likelihood landscape for 50 sequences and about 3000 nucleotides for 100 sequences. If the relationship between length and number of sequences is approximately linear (which is unlikely, even in the absence of conflict), then this suggests that, for example, SSU rDNA sequences (ca. 2000 nucleotides) will be useful for only about 80 sequences, whereas 5000 to 6000 nucleotides will be needed for 150 sequences. On the other hand, if it is a power relationship (which seems more likely), then the data are consistent with a tripling of the sequence length for each additional 50 sequences. This predicts that the SSU rDNA can handle 90 sequences but that 150 sequences will require 9000 aligned nucleotides. These alternative predictions are hard to test at the moment, because there are only a couple of multi-gene data sets with >150 sequences.

The apparently uninformative VOS, EF1a, and GRIM data sets were particularly difficult for both search strategies (as expected), with many islands being found (Table 2). Furthermore, the two search strategies differed considerably in terms of how their difficulties were manifested.

GARLI's main problem was the previously noted tendency to become trapped in local optima for less informative data sets. Indeed, for the VOS, EF1a, and GRIM data sets GARLI only once found the peak of an SPR island, whereas for the MURP, QIU, MAMM, and JANS data sets, it always found the peak. Without prior knowledge of how much phylogenetic information there is in any one data set (because it cannot be predicted directly from sequence length), it seems likely that it will be impossible to identify a priori whether GARLI will become trapped or not. In cases of doubt (e.g., 10 iterations produce 10 different trees), it will be necessary either to perform subsequent SPR branch swapping (as I did here), or to perform a very large number of iterations (to see how often the same set of trees is found from different starting trees).

For the Ratchet (Nixon) strategy, the biggest problem for the three difficult data sets was getting PAUP* to consistently identify a set of optimal trees. The source of the problem seemed to be inconsistency in producing a ML score for each tree. During the search for each iteration, PAUP* would often retain a relatively large number of trees as being equally optimal, but subsequent independent evaluation of those trees would produce a range of different ML scores (sometimes involving more than 1 log-likelihood unit). That is, there would be different numbers of trees found for the same island in different iterations (e.g., when an island contained three distinct trees PAUP* would find two, three, four, or five trees in different iterations). This problem was compounded by the different likelihood values produced by different computers (i.e., the same analysis on different computers would produce different results).

An apparently related problem occurred for the FUNG and RYDI data sets. For the FUNG data set, PAUP* consistently chose a tree with a polychotomy in spite of the fact that the likelihood score was reported to be identical to those for two of the three resolutions of that polychotomy. Moreover, GARLI chose one or the other of the two resolutions in different iterations, but never chose the polychotomous tree, because it currently deals only with binary trees. Similarly for the RYDI data set, for each island PAUP* found a single tree with two polychotomies (the same two on each island), whereas in nine cases GARLI found several possible resolutions of those polychotomies (in the 10th case it found a genuinely suboptimal tree on one of the islands). PAUP* estimated all of these trees found for each island to have the same likelihood to five decimal places but still reported only the one with the polychotomies. This emphasizes the fact that small differences between likelihood values cannot be used as evidence that the trees are not equally optimal, nor can production of a single tree be used as evidence that there are not other equal trees nearby. Maximum likelihood was not meant to be easy.

Another problem was encountered for the VOS data set, where for some iterations literally hundreds of trees were retained for the search based on reweighted characters. Because only one of these trees is used as the starting tree for the subsequent unweighted search, this is enormously wasteful of processing time. I eventually resorted to the solution of Vos (2003) and set a time limit on each of the weighted searches (24 hours). Note that my results for this data set are quite different from those of Vos (2003), who assumed that the likelihood landscape for these data was not complex, which may have been true before I realigned the data to remove the spuriously informative sites.

It is possible that many of these problems resulted from the substitution model (GTR+I+G) being overparameterized for these three data sets (see Table 2), so that PAUP* (in particular) was trying to estimate parameter values with insufficient data at hand. I assessed this possibility by repeating these three analyses with the simpler HKY+G nucleotide-substitution model. For two of the data sets, EF1a and GRIM, this seemed to improve the situation somewhat. GARLI reduced the number of suboptimal iterations (9 to 8 for EF1a, 10 to 6 for GRIM), and PAUP* reduced the number of suboptimal trees retained by each iteration (12 to 8 for GRIM). However, the same problems persisted, although reduced, and so the complexity of tree space was still a source of problems. Furthermore, for the VOS data set use of the simpler model actually resulted in the retention of >200 equally optimal trees for the unweighted searches, and so this analysis was abandoned. Phylogenetic informativeness seems to be the key to simplifying tree space and thus ensuring a successful maximum-likelihood analysis, rather than choice of substitution model.

In all cases except two (i.e., 18 out of 20 data sets), the GARLI program finished its processing faster than the Ratchet (Nixon) strategy. The two exceptions were the MURP data set, where both GARLI and the Ratchet (Nixon) took 6 hours per iteration on the fastest computer, and the ROKA data set, where GARLI took 9 hours per iteration but the Ratchet (Nixon) took 5.5 hours. The speed of the GARLI program thus seems to be affected more by the number of characters than by the number of taxa. That is, it responds to the number of unique data patterns, as would be expected given the nature of the genetic algorithm used. Conversely, the Ratchet (Nixon) strategy slows down rapidly as the number of taxa increases but is affected less by the number of characters. Therefore, it is worth briefly considering strategies for speeding-up these analyses.

First, the SPR search used by the Ratchet (Nixon) strategy is inefficient in that it frequently repeats exactly the same tree search for the unweighted characters. For example, if a search based on reweighted characters has not moved to a new island or returns to a previous island, then most of the subsequent unweighted search will simply be an unnecessary repeat of the previous search. This issue can be dealt with by intelligent bookkeeping, such as is already used in PAUP* for multiple stepwise-addition searches based on random input order, where a search iteration is terminated once it reaches an island that was encountered in a previous iteration. This could potentially save a large amount of time, particularly if there are few islands being visited, as each new search is then restricted to those parts of islands that have not been searched before.

Second, it is not strictly necessary to find the optimal tree during the reweighted search, because all that is required is to move away from the current tree (Nixon, 1999; Goloboff, 2002). Therefore, this search could be shortened by using the currently available PAUP* commands to set a limit on the search time (TimeLimit); the number of branch swaps to occur (RearrLimit); the number of branches across which swaps can occur (ReconLimit); or the number of equally optimal trees swapped and saved (MulTrees). The latter command is closest in spirit to the original parsimony ratchet (Nixon, 1999). A useful additional command would be to provide a limit on the number of branch swaps allowed to occur after a better topology is found (i.e., not swapping to completion on the last tree). These limits may make the Ratchet (Nixon) strategy directly competitive with the GARLI program.

Third, it is worthwhile spending some time to pare down the ApproxLim percentage in PAUP* as much as possible, as I did for several of the data sets here—even 0.01% can make a big difference to the search speed (e.g., the BASI, LEWI, LITT, QIU, and RYDI data sets). All of the large data sets that I tested could be analyzed at ApproxLim = 2% without problems, and this could be used as a default value from which to assess the effect of further reductions. The biggest success was for the LITT data set (158 sequences), which took 25 days to be analyzed at ApproxLim = 1% but only 7 days at ApproxLim = 0.7% (while following exactly the same search path). The LEWI data set (150 sequences) took the longest to analyze, at 11.5 days (ApproxLim = 0.79%).

Fourth, any other computer program that implements character weighting (e.g., RAxML, TreeFinder) could be used for the searches, thus increasing the search speed. However, in this case we would need to address the problem of possibly suboptimal trees being found on each island.

Other than the number of unique data patterns, the only obvious source of variation in speed for the GARLI program is its use of random starting trees, which means that considerable time is spent moving to an island of near-optimal trees. Using better starting trees might thus be one viable option, although they would need to be on different islands if tree space is to be explored effectively. Perhaps using all three of the stepwise-addition, neighbor-joining, and parsimony trees as starting trees would explore potentially optimal islands just as effectively.

The GARLI algorithm is straightforward to implement as a coarse-grained multi-processor application (sometimes called “embarrassingly parallel”). Each of the search iterations can be run on a separate processor because the iterations are independent of each other (starting from different trees). However, this is not true of the Ratchet (Nixon), because each iteration starts from the tree found in the immediately preceding iteration (thus preserving hard-won information between iterations). Clement et al. (1999) try to bypass this limitation by starting each iteration from the best tree (or one of the best trees) found in previous iterations, thus making the iterations semi-independent. This makes a third definition of the "ratchet strategy," in addition to those of Nixon (1999) and Vos (2003). Fine-grained multiprocessing is likely to be similar for GARLI and the Ratchet (Nixon).

Multiple Starting Trees

As discussed above, the most commonly used strategy for exploring multiple islands of trees is to perform a series of searches from different starting trees. This raises the issue of how a set of starting trees should best be created. A thorough exploration of this issue is beyond my scope here, but some preliminary assessment seems to be worthwhile, in terms of the variability among the trees.

To this end, for each of the three principal data sets, I generated sets of starting trees using a range of available strategies: random trees; random input order for the stepwise-addition ML tree; random input order for the stepwise-addition parsimony tree; character weighting for the neighbor-joining tree; generalized neighbor joining; and a random walk with the neighbor-joining tree as the center tree. The variability of the trees produced is shown in Table 4, measured as the Robinson-Foulds distance between the trees.

As expected, the random trees are approximately equally distant from each other, implying that they are evenly spread through tree space, whereas each of the other strategies produced a more tightly clustered set of trees. There seems to be little to choose between most of the strategies, in the sense that each produced a set of trees clustered around the "default" tree for that strategy (e.g., the default parsimony or neighbor-joining tree), and each of the clusters had roughly the same variability. In this sense, each set of trees thoroughly explored one part of tree space but completely ignored all other parts, and it is obvious that the different sets were exploring somewhat different parts of tree space. The use of these tree sets will only be an effective strategy if the optimal islands are near each other (as they usually seem to be), but not otherwise.

It is interesting to note that the within-set variability of the trees was of the same order as the variability between the three "default" trees. In this sense, it is likely that combined use of the stepwise-addition ML, neighbor joining, and stepwise-addition parsimony trees would explore different islands just as effectively as use of a set of trees based on any one of these strategies.

The only obvious exception to these generalizations is that the strategies based on generalized neighbor joining and on the use of character weighting with neighbor joining both tended to produce some trees that were very similar to each other within the set. Use of these trees would presumably be wasteful of search time, and it might therefore be appropriate to apply a preliminary filter in order to remove such trees from the set to be used. This would particularly affect the Ratchet (Vos) strategy.

The other relevant observation to make concerns the relative time taken to produce the sets of trees. There was little to choose between the strategies in this regard except for the stepwise-addition ML trees, which took very much longer to produce than did the other tree types. For example, the set of 200 trees for the HIV data set took the equivalent of 3 months of time on the fastest computer. Given that there seems to be no great advantage to using this particular type of tree, at least as starting trees for these three data sets, it seems to be inappropriate that they should be used as the default trees in any computer program.

To explore the use of these different starting trees further, it would be necessary to perform a search from each of the trees in each set, in order to evaluate how effective the range of trees is at locating different islands (e.g., does the set of parsimony trees do better or worse at exploring islands than, say, the set of generalized neighbor-joining trees?). This would be a very ambitious undertaking, especially if done in conjunction with use of several of the programs as the search strategy.

Conclusions

The basic conclusion from the analyses performed here is that it is currently feasible to perform a thorough analysis of up to 150 multigene sequences, including investigation of multiple SPR (and TBR) islands of nearly optimal trees, in <2 weeks on a single personal computer. The exact time will depend on the complexity of the tree space, and whether GARLI or the Ratchet (Nixon) is the faster analysis strategy. GARLI is usually the faster strategy for informative data sets but the Ratchet (Nixon) is more successful at finding island peaks for less informative data sets. Thorough investigation of all of the trees on the islands, however, will take considerably longer, as it requires subsequent NNI or SPR branch swapping. These analyses would all be made much faster if deployed on a multiprocessor, parallel or distributed computing system, of course.

Some researchers might be disappointed with this conclusion, because it does not explicitly include concepts such as bootstrapping that are designed to provide some information about clade robustness. Traditionally, bootstrapping would require the entire tree-building analysis to be repeated many times, and this is not feasible if any one analysis requires a couple of weeks of computer time. In contrast, the use of decay values as a measure of support is facilitated by the methods discussed here, because in this case the ML value has to be found for both the constrained and unconstrained trees (Müller, 2005). Nevertheless, the approach presented here clearly relies on using the set of nearly optimal trees as the best means of assessing clade robustness. That is, any clade that does not appear in all (or most) of the nearly optimal trees is not a robust clade. This implies that the result of the tree-building analysis should be presented not as a consensus bootstrap tree but rather as a consensus "nearly optimal" tree, perhaps with separate consensuses for each island (cf. Maddison, 1991). Appropriate methods for doing this in a likelihood context are discussed by Jermiin et al. (1997), although I make no specific recommendation here about how to quantitatively define "nearly optimal."

The second conclusion is that there appears to be little to choose between different types of starting trees. The stepwise-addition ML tree takes many orders of magnitude longer to compute than the BioNJ tree based on logDet distances, and the small superiority of its ML score does not justify this time. In much less time, a successive-approximations strategy (BioNJ tree followed by NNI branch swapping) can be used for estimating both the nucleotide-substitution model and a very good starting tree. However, exploring the tree space of multiple islands using multiple starting trees is a more complex issue, and a ratchet strategy is likely to be superior for 50 to 150 sequences.

The third conclusion is that many users of PAUP* are using that program inefficiently when performing ML analyses. In particular, use of the stepwise-addition ML starting tree is unnecessarily slow; SPR branch swapping seems to be all that is generally necessary (not TBR); and adjustment of the ApproxLim parameter is almost mandatory for large data sets. A simple successive-approximations strategy can easily be implemented (starting tree followed by NNI and then SPR branch swapping) both to define the nucleotide-substitution model and to provide a very good estimate of the optimal tree. The problems identified by Davis et al. (2005) with the use of the default options in PAUP* for maximum parsimony analyses do not seem to affect ML analyses as much, as large numbers of equally optimal trees are retained only for data sets with low amounts of phylogenetic information.

The fourth conclusion is that most of the current heuristic algorithms that are advertised as providing faster analyses than PAUP* often do so at the expense of finding the optimal tree. These algorithms are prone to getting stuck in local optima, even when the topography of the tree space appears to be relatively smooth. Such methods cannot be relied upon, especially in situations where there are multiple islands of nearly optimal trees (i.e., uninformative data sets). It is possible that the programs would perform much better if their parameters were optimized for each data set, but it is not obvious how to develop a general protocol for doing this.

Some researchers might also be disappointed with this conclusion, especially those responsible for developing the computer programs. Indeed, the authors of most of the programs that I have used here do not actually intend their program to find the ML tree but merely to find a nearly optimal tree in a reasonable time. The ML tree is unlikely to be the true evolutionary tree, and thus there is no great incentive to spend too many days on a search for it. It is for this reason that I have focused on the need to find the set of nearly optimal trees. Nevertheless, someone clearly needs to investigate more thoroughly the effect of the various approximations that are being employed, notably on how the rank order of the trees compares to the rank order of trees calculated by a full likelihood approach. In the meantime, all "fast" programs should be used in conjunction with a more accurate program in order to guarantee that a good tree has been located.

Clearly, maximum likelihood analysis is a complex business, and although I have documented some general principles I cannot provide any detailed recommendations for analyzing phylogenetic data (i.e., a protocol for analysis). Such recommendations would presumably involve some explicit form of preliminary analysis that could be used to select the optimal search settings for performing the definitive phylogenetic analysis (cf. the work of Davis et al., 2005). All I can really do here is suggest that commitment to a thorough analysis using either GARLI or the Ratchet (Nixon) will probably pay dividends (each failed at least once when the other succeeded, so some data sets seem to be more amenable to analysis by one method rather than the other) and perhaps encourage people to develop faster and/or more accurate versions of these algorithms.

Acknowledgements

Thanks to Thomas Keane for help running the DPRml analyses. Thanks also to Stéphane Guindon, Gangolf Jobb, Thomas Keane, Kai Müller, Alexandros Stamatakis, Le Sy Vinh, Rutger Vos, Simon Whelan, and Derrick Zwickl for helpful discussion and/or comments on the manuscript.

References

Allen, B. L., and M. Steel. 2001. Subtree transfer operations and their induced metrics on evolutionary trees. Ann. Combin. 5:1–15.

Artiss, T., T. R. Schultz, D. A. Polhemus, and C. Simon. 2001. Molecular phylogenetic analysis of the dragonfly genera *Libellula*, *Ladona*, and *Plathemis* (Odonata: Libellulidae) based on mitochondrial cytochrome oxidase I and 16S rRNA sequence data. Mol. Phylogen. Evol. 18:348–361.

Bader, D. A., U. Roshan, and A. Stamatakis. 2006. Computational grand challenges in assembling the tree of life: Problems & solutions. Pages 128–170 *in* Advances in computers volume 68: Computational biology and bioinformatics (C.-W. Tseng, ed.). Academic Press, New York.

Brauer, M. J., M. T. Holder, L. A. Dries, D. J. Zwickl, P. O. Lewis, and D. M. Hillis. 2002. Genetic algorithms and parallel processing in maximum-likelihood phylogeny inference. Mol. Biol. Evol. 19:1717–1726.

Bryant, D., N. Gautier, and M.-A. Poursat. 2005. Likelihood calculations in molecular phylogenetics. Pages 33–62 *in* Mathematics of evolution and phylogeny (O. Gascuel, ed.). Oxford University Press, Oxford, UK.

Charleston, M. A. 1995. Toward a characterization of landscapes of combinatorial optimization problems, with special attention to the phylogeny problem. J. Comput. Biol. 2:439–450.

Clement, M., Q. Snell, G. Judd, and M. Whiting. 1999. High performance phylogenetic inference. Pages 335–336 *in* Proceedings of the 8th IEEE international symposium on high performance distributed computing. IEEE Computer Society, Washington, DC.

Davis, J. I., K. C. Nixon, and D. P. Little. 2005. The limits of conventional cladistic analysis. Pages 119–147 in Parsimony, phylogeny, and genomics (V. A. Albert, ed.). Oxford University Press, Oxford, UK.

Du, Z., A. Stamatakis, F. Lin, U. Roshan, and L. Nakhleh. 2005. Parallel divide-and-conquer phylogeny reconstruction by maximum likelihood. Lect. Notes Comp. Sci. 3726:776–785.

Felsenstein, J. 1981. Evolutionary trees from DNA sequences: A maximum likelihood approach. J. Mol. Evol. 17:368–376.

Felsenstein, J. 1989. PHYLIP: Phylogeny inference package (version 3.2). Cladistics 5:164–166.

Goloboff, P. A. 2002. Techniques for analyzing large data sets. Pages 70–79 *in* Techniques in molecular systematics and evolution (R. DeSalle, G. Giribet, and W. Wheeler, eds). Birkhäuser Verlag, Basel.

Grimm, G. W., S. S. Renner, A. Stamatakis, and V. Hemleben. 2006. A nuclear ribosomal DNA phylogeny of *Acer* inferred with maximum likelihood, splits graphs, and motif analyses of 606 sequences. Evol. Bioinform. Online 2:279–294.

Gu, X., and J. Zhang. 1997. A simple method for estimating the parameter of substitution rate variation among sites. Mol. Biol. Evol. 14:1106–1113.

Guindon, S., and O. Gascuel. 2003. A simple, fast, and accurate algorithm to estimate large phylogenies by maximum likelihood. Syst. Biol. 52:696–704.

Hambleton, S., A. Tsuneda, and R. S. Currah. 2003. Comparative morphology and phylogenetic placement of two microsclerotial black fungi from *Sphagnum*. Mycologia 95:969–975.

Hendy, M. D., and B. R. Holland. 2003. Upper bounds on maximum likelihood for phylogenetic trees. Bioinformatics 19:ii66–ii72.

Hillis, D. M. 1996. Inferring complex phylogenies. Nature 383:130–131.

Hillis, D. M., T. A. Heath, and K. St John. 2005. Analysis and visualization of tree space. Syst. Biol. 54:471–482.

Hillis, D. M., and T. P. Wilcox. 2005. Phylogeny of the New World true frogs (*Rana*). Mol. Phylogenet. Evol. 34:299–314.

Hobolth, A., and R. Yoshida. 2005. Maximum likelihood estimation of phylogenetic tree and substitution rates via generalized neighbor-joining and the EM algorithm. Pages 41–50 *in* Proceedings of the 1st international conference on algebraic biology (H. Anai and K. Horimoto, eds.). Universal Academy Press, Tokyo.

Hordijk, W., and O. Gascuel. 2005. Improving the efficiency of SPR moves in phylogenetic tree search methods based on maximum likelihood. Bioinformatics 21:4338–4347.

Huelsenbeck, J. P., and K. A. Crandall. 1997. Phylogeny estimation and hypothesis testing using maximum likelihood. Annu. Rev. Ecol. Syst. 28:437–466.

Jansen, R. K., C. Kaittanis, C. Saski, S.-B. Lee, J. Tomkins, A. J. Alverson, and H. Daniell. 2006. Phylogenetic analysis of *Vitis* (Vitaceae) based on complete chloroplast genome sequences: Effects of taxon sampling and phylogenetic methods on resolving relationships among rosids. BMC Evol. Biol. 6:32.

Jermiin, L. S., G. J. Olsen, K. L. Mengersen, and S. Easteal. 1997. Majority-rule consensus of phylogenetic trees obtained by maximum-likelihood analysis. Mol. Biol. Evol. 14:1296–1302.

Jobb, G., A. von Haeseler, and K. Strimmer. 2004. Treefinder: A powerful graphical analysis environment for molecular phylogenetics. BMC Evol. Biol. 4:18.

Johnson, R. N., P.-M. Agapow, and R. H. Crozier. 2003. A tree island approach to inferring phylogeny in the ant subfamily Formicinae, with especial reference to the evolution of weaving. Mol. Phylogen. Evol. 29:317–330.

Jönsson, H., and B. Söderberg. 2003. An approximate maximum likelihood approach, applied to phylogenetic trees. J. Comput. Biol. 10:737–749.

Katoh, K., K. Kuma, and T. Miyata. 2001. Genetic algorithm-based maximum-likelihood analysis for molecular phylogeny. J. Mol. Evol. 53:477–484.

Keane, T. M. 2006. Computational methods for statistical phylogenetic inference. PhD thesis, The National University of Ireland Maynooth, Ireland.

Kirkup, B., and J. Kim. 2000. From rolling hills to jagged mountains: scaling of heuristic searches for phylogenetic estimation. Unpublished manuscript. Department of Ecology and Evolutionary Biology, Yale University, USA. 26 August 2000. Available at http://kim.bio.upenn.edu/web/papers/kirkup_kim.pdf.

Kosakovsky Pond, S. L., and S. V. Muse. 2004. Column sorting: rapid calculation of the phylogenetic likelihood function. Syst. Biol. 53:685–692.

Kosiol, C., L. Bofkin, and S. Whelan. 2006. Phylogenetics by likelihood: evolutionary modeling as a tool for understanding the genome. J. Biomed. Informatics 39:51–61.

Lemmon, A. R., and M. C. Milinkovitch. 2002. The metapopulation genetic algorithm: an efficient solution for the problem of large phylogeny estimation. Proc. Nat. Acad. Sci. USA 99:10516–10521.

Lewis, L. A., and P. O. Lewis. 2005. Unearthing the molecular phylodiversity of desert soil green algae (Chlorophyta). Syst. Biol. 54:936–947.

Lewis, P. O. 1998. A genetic algorithm for maximum-likelihood phylogeny inference using nucleotide sequence data. Mol. Biol. Evol. 15:277–283.

Maddison, D. R. 1991. The discovery and importance of multiple islands of most-parsimonious trees. Syst. Zool. 40:315–328.

Mak, T. S. T., and K. P. Lam. 2004. On computing maximum likelihood phylogeny using FPGA. Lect. Notes Comp. Sci. 3203:1188.

Matheny, P. B., Z. Wang, M. Binder, J. M. Curtis, Y. W. Lim, R. H. Nilsson, K. W. Hughes, V. Hofstetter, J. F. Ammirati, C. Schoch, E. Langer, G. Langer, D. J. McLaughlin, A. W. Wilson, T. Frøslev, Z.-W. Ge, R. W. Kerrigan, J. C. Slot, Z.-L. Yang, T. J. Baroni, M. Fischer, K. Hosaka, K. Matsuura, M. T. Seidl, J. Vauras, and D. S. Hibbett. 2007. Contributions of *rpb2* and *tef1* to the phylogeny of mushrooms and allies (Basidiomycota, Fungi). Mol. Phylogen. Evol. 43:430–451.

Matsuda, H. 1995. Construction of phylogenetic trees from amino acid sequences using a genetic algorithm. Genome Inform. 6:19–28.

Matsuda, H. 1996. Protein phylogenetic inference using maximum likelihood with a genetic algorithm. Pacific Symp. Biocomp. 1:312–323.

Mecham, J., M. Clement, Q. Snell, T. Freestone, K. Seppi, and K. Crandall. 2006. Jumpstarting phylogenetic analysis. Int. J. Bioinform. Res. Appl. 2:19–35.

Minin, V., Z. Abdo, P. Joyce, and J. Sullivan. 2003. Performance-based selection of likelihood models for phylogeny estimation. Syst. Biol. 52:674–683.

Morrison, D. A. 2006. Phylogenetic analyses of parasites in the new millennium. Adv. Parasitol. 63:1–124.

Morrison, D. A., S. Bornstein, P. Thebo, U. Wernery, J. Kinne, and J. G. Mattsson. 2004. The current status of the small subunit rRNA phylogeny of the coccidia (Sporozoa). Int. J. Parasitol. 34:501–514.

Müller, K. 2004. PRAP—Computation of Bremer support for large data sets. Mol. Phylogen. Evol. 31:780–782.

Müller, K. 2005. The efficiency of different search strategies in estimating parsimony jackknife, bootstrap, and Bremer support. BMC Evol. Biol. 5:58.

Murphy, W. J., E. Eizirik, W. E. Johnson, Y. P. Zhang, O. A. Ryder, and S. J. O'Brien. 2001a. Molecular phylogenetics and the origins of placental mammals. Nature 409:614–618.

Murphy, W. J., E. Eizirik, S. J. O'Brien, O. Madsen, M. Scally, C. J. Douady, E. Teeling, O. A. Ryder, M. J. Stanhope, W. W. de Jong and M. S. Springer. 2001b. Resolution of the early placental mammal radiation using Bayesian phylogenetics. Science 294:2348–2351.

Nixon, K. C. 1999. The parsimony ratchet, a new method for rapid parsimony analysis. Cladistics 15:407–414.

Oliveira, C. A. S. 2005. An algorithm for the maximum likelihood problem on evolutionary trees. J. Combin. Optimiz. 10:61–75.

Olsen, G. J., H. Matsuda, R. Hagstrom, and R. Overbeek. 1994. FastDNAml: A tool for construction of phylogenetic trees of DNA sequences using maximum likelihood. Comp. Appl. Biosci. 10:41–48.

Olson, P. D., T. H. Cribb, V. V. Tkach, R. A. Bray, and D. T. J. Littlewood. 2003. Phylogeny and classification of the Digenea (Platyhelminthes: Trematoda). Int. J. Parasitol. 33:733–755.

Page, R. D. M. 1993. On islands of trees and the efficacy of different methods of branch swapping in finding most-parsimonious trees. Syst. Biol. 42:200–210.

Pearson, W. R., G. Robins, and T. Zhang. 1999. Generalized neighbor-joining: More reliable phylogenetic tree reconstruction. Mol. Biol. Evol. 16:806–816.

Poladian, L. 2005. A GA for maximum likelihood phylogenetic inference using neighbour-joining as a genotype to phenotype mapping. Pages 415–422 in Proceedings of the 2005 conference on genetic and evolutionary computation (GECCO'05). ACM Press, New York.

Posada, D., and K. A. Crandall. 1998. ModelTest: Testing the model of DNA substitution. Bioinformatics 14:817–818.

Qiu, Y.-L., O. Dombrovska, J. Lee, L. Li, B. A. Whitlock, F. Bernasconi-Quadroni, J. S. Rest, C. C. Davis, T. Borsch, K. W. Hilu, S. S. Renner, D. E. Soltis, P. S. Soltis, M. J. Zanis, J. J. Cannone, R. R. Gutell, M. Powell, V. Savolainen, L. W. Chatrou, and M. W. Chase. 2005. Phylogenetic analyses of basal angiosperms based on nine plastid, mitochondrial, and nuclear genes. Int. J. Plant Sci. 166:815–842.

Quicke, D. L. J., J. Taylor, and A. Purvis. 2001. Changing the landscape: a new strategy for estimating large phylogenies. Syst. Biol. 50:60–66.

Ranwez, V., and O. Gascuel. 2002. Improvement of distance-based phylogenetic methods by a local maximum likelihood approach using triplets. Mol. Biol. Evol. 19:1952–1963.

Ren, F., H. Tanaka, and Z. Yang. 2005. An empirical examination of the utility of codon-substitution models in phylogeny reconstruction. Syst. Biol. 54:808–818.

Roch, S. 2006. A short proof that phylogenetic tree reconstruction by maximum likelihood is hard. IEEE/ACM Trans. Comput. Biol. Bioinform. 3:92–94.

Rogers, J. S., and D. L. Swofford. 1998. A fast method for approximating maximum likelihoods of phylogenetic trees from nucleotide sequences. Syst. Biol. 47:77–89.

Rokas, A., and S. B. Carroll. 2005. More genes or more taxa? The relative contribution of gene number and taxon number to phylogenetic accuracy. Mol. Biol. Evol. 22:1337–1344.

Rydin, C., M. Källersjö, and E. M. Friis. 2002. Seed plant relationships and the systematic position of Gnetales based on nuclear and chloroplast DNA: Conflicting data, rooting problems, and the monophyly of conifers. Int. J. Plant Sci. 163:197–214.

Saitou, N. 1988. Property and efficiency of the maximum likelihood method for molecular phylogeny. J. Mol. Evol. 27:261–273.

Sallum, M. A. M., T. R. Schultz, P. G. Foster, K. Aronstein, R. A. Wirtz, and R. C. Wilkerson. 2002. Phylogeny of Anophelinae (Diptera: Culicidae) based on nuclear ribosomal and mitochondrial DNA sequences. Syst. Entomol. 27:361–382.

Salter, L. A. 2001. Complexity of the likelihood surface for a large DNA dataset. Syst. Biol. 50:970–978.

Salter, L. A., and D. K. Pearl. 2001. Stochastic search strategy for estimation of maximum likelihood phylogenetic trees. Syst. Biol. 50:7–17.

Sanderson, M. J., and H. B. Shaffer. 2002. Troubleshooting molecular phylogenetic analyses. Annu. Rev. Ecol. Syst. 33:49–72.

Schmidt, H. A., K. Strimmer, M. Vingron, and A. von Haeseler. 2002. Tree-Puzzle: Maximum likelihood phylogenetic analysis using quartets and parallel computing. Bioinformatics 18:502–504.

Shen, J., and R. B. Heckendorn. 2004. Discrete branch length representations for genetic algorithms in phylogenetic search. Lect. Notes Comp. Sci. 3005:94–103.

Sikes, D. S., and P. O. Lewis. 2001. PAUPRat: PAUP* implementation of the parsimony ratchet. Beta software, version 1. Distributed by the authors. Department of Ecology and Evolutionary Biology, University of Connecticut, Storrs, USA, June 2001.

Sota, T., and M. Sasabe. 2006. Utility of nuclear allele networks for analysis of closely related species in the genus *Carabus*, subgenus *Ohomopterus*. Syst. Biol. 55:329–344.

SPSS Inc. 2000. SYSTAT 9 for Windows. SPSS, Chicago, Illinois.

Stamatakis, A. 2005. An efficient program for phylogenetic inference using simulated annealing. Page 198b *in* Proceedings of the 19th international parallel and distributed processing symposium (IPDPS'05), and the 4th international workshop on high performance computational biology (HiComB'05). IEEE Press, Piscataway, New Jersey.

Stamatakis, A. 2006. RAxML-VI-HPC: Maximum likelihood-based phylogenetic analyses with thousands of taxa and mixed models. Bioinformatics 22:2688–2690.

Stamatakis, A., T. Ludwig, and H. Meier. 2005. RAxML-III: A fast program for maximum likelihood-based inference of large phylogenetic trees. Bioinformatics 21:456–463.

Stamatakis, A., T. Ludwig, H. Meier, and M. J. Wolf. 2002. Accelerating parallel maximum likelihood-based phylogenetic tree calculations using subtree equality vectors. Pages 1–16 *in* Proceedings of the 15th IEEE/ACM conference on supercomputing. ACM Press, New York.

Stewart, C. A., D. Hart, D. K. Berry, G. J. Olsen, E. A. Wernert, and W. Fischer. 2001. Parallel implementation and performance of fastDNAml—a program for maximum likelihood phylogenetic inference. Pages 20–20 *in* Proceedings of the 14th IEEE/ACM conference on supercomputing. ACM Press, New York.

Strimmer, K., and A. von Haeseler. 1996. Quartet puzzling: A quartet maximum-likelihood method for reconstructing tree topologies. Mol. Biol. Evol. 13:964–969.

Sullivan, J. 2005. Maximum likelihood methods for phylogeny estimation. Methods Enzymol. 395:757–779.

Sullivan, J., Z. Abdo, P. Joyce, and D. L. Swofford. 2005. Evaluating the performance of a successive-approximations approach to parameter optimization in maximum-likelihood phylogeny estimation. Mol. Biol. Evol. 22:1386–1392.

Suzuki, R., T. Taniguchi, and H. Shimodaira. 2004. An approximate maximum likelihood method for phylogenetic tree analysis based on low-temperature Markov chain Monte Carlo. Genome Inform. 15:p081.

Takahashi, K., and M. Nei. 2000. Efficiencies of fast algorithms of phylogenetic inference under the criteria of maximum parsimony, minimum evolution, and maximum likelihood when a large number of sequences are used. Mol. Biol. Evol. 17:1251–1258.

Tamura, K., M. Nei, and S. Kumar. 2004. Prospects for inferring very large phylogenies by using the neighbor-joining method. Proc. Nat. Acad. Sci. USA 101:11030–11035.

Troell, K., A. Engström, D. A. Morrison, J. G. Mattsson, and J. Höglund. 2006. Global patterns reveal strong population structure in *Haemonchus contortus*, a nematode parasite of domesticated ruminants. Int. J. Parasitol. 36:1305–1316.

Vinh, L. S., H. A. Schmidt, and A. von Haeseler. 2005. PhyNav: A novel approach to reconstruct large phylogenies. Pages 386–393 *in* Classification, the ubiquitous challenge: Proceedings of the 28th annual conference of the Gesellschaft für Klassifikation e.V. (C. Weihs and W. Gaul, eds.). Springer-Verlag, Heidelberg.

Vinh, L. S., and A. von Haeseler. 2004. IQPNNI: Moving fast through tree space and stopping in time. Mol. Biol. Evol. 21:1565–1571.

Vos, R. A. 2003. Accelerated likelihood surface exploration: The likelihood ratchet. Syst. Biol. 52:368–373.

Williams, T. L., and B. M. E. Moret. 2003. An investigation of phylogenetic likelihood methods. Pages 79–86 in Proceedings of the 3rd IEEE symposium on bioinformatics and bioengineering (BIBE'03). IEEE Press, Piscataway, New Jersey.

Xia, X. 2006. Molecular phylogenetics: mathematical framework and unsolved problems. Pages 171–191 *in* Structural approaches to sequence evolution (U. Bastolla, M. Porto, H. E. Roman, and M. Vendruscolo, eds.). Springer-Verlag, New York.

Yang, Z. 1997. PAML: A program package for phylogenetic analysis by maximum likelihood. Comp. Appl. Biosci. 13:555–556.

Yang, Z. 2000. Maximum likelihood estimation on large phylogenies and analysis of adaptive evolution in human influenza virus A. J. Mol. Evol. 51:423–432.

Yang, Z. 2006. Computational molecular evolution. Oxford University Press, Oxford, UK.

Zwickl, D. J. 2006. Genetic algorithm approaches for the phylogenetic analysis of large biological sequence datasets under the maximum likelihood criterion. PhD thesis, The University of Texas at Austin.

First submitted 10 January 2007; reviews returned 23 April 2007; final acceptance 9 October 2007
Associate Editor: Laura Kubatko
Editor-in-Chief: Jack Sullivan