Algorithms for Phylogenetic Tree Reconstruction

L. A. Salter
Department of Mathematics and Statistics
University of New Mexico
Albuquerque, NM 87131, USA

Abstract *Scientists often wish to use the information contained in the DNA sequences of a collection of organisms, or taxa, to infer the evolutionary relationships among those taxa. These evolutionary relationships are generally represented by a labeled binary tree, called a phylogenetic tree. The phylogeny reconstruction problem is computationally difficult because the number of possible solutions increases rapidly with the number of taxa to be included in the tree. For example, for 30 taxa (a moderate number of taxa given the wide availability of DNA sequence data today!), there are* 8.69×10^{36} *possible unrooted bifurcating trees to be considered. Thus, scientists are faced with the task of choosing from among a large number of trees that tree(s) which gives the "best" representation of the evolutionary relationships among the taxa based on the data. Often, optimality criteria are applied to allow trees to be compared with one another. In this case, the phylogenetic tree reconstruction problem becomes that of finding the particular tree(s) that optimize the criteria. Exhaustive search of the space of phylogenetic trees is generally not possible for more than 11 taxa, and so algorithms for efficiently searching the space of trees must be developed. Branch-and-bound methods can reasonably be applied for up to about 20 taxa, but no computationally efficient exact algorithms have been developed for problems larger than this. Therefore, scientists generally rely on heuristic algorithms, such as stepwise-addition and star decomposition methods. However, such algorithms generally involve a prohibitive amount of computation time for large problems and often find trees that are only locally optimal. Recently, stochastic search algorithms, such as simulated annealing, genetic algorithms, and Markov chain Monte Carlo (MCMC) methods, have been developed and successfully applied to phylogenetic problems. We discuss and compare some of the methods available for phylogenetic tree reconstruction under an optimality criteria, and suggest some directions for future research.*

Keywords: phylogenetic trees, algorithms, optimality criteria

1 Introduction

We consider the problem of estimating evolutionary relationships among a collection of n organisms, or *taxa*. Such relationships are often represented by a labeled binary graph, called a *phylogenetic tree*, in which the edges of the graph represent ancestry-descent relationships and the nodes represent hypothetical ancestral organisms. Construction of phylogenetic trees for a group of taxa requires data for each of the taxa that contain information about their evolutionary history. Historically, morphological data were predominantly used for inferring phylogenies; however, the abundance of DNA sequence data currently available for a variety of organisms has led to phylogenetic inference based on these data. Proper estimation of phylogenetic trees is important, as such trees have a variety of uses. For example, they are used in classification and taxonomy, in the study of the relationships between parasite and host, in the molecular epidemiological study of viruses (e.g., HIV and HPV)

[16, 18], and in forensic medicine [8]. For these reasons, much attention has been devoted to the study of algorithms used to construct phylogenetic trees.

Given data for the n taxa under consideration, our goal is to find the particular phylogenetic tree(s) which best represent the evolutionary history of the taxa. One method for doing this is to specify an optimality criteria to select among the set of possible phylogenetic trees [1]. Many such optimality criteria have been specified, the two most popular being the maximum parsimony (MP) criteria and the maximum likelihood (ML) criteria. The MP criteria specifies that the tree requiring the fewest number of evolutionary changes to explain the data should be preferred (this is simply the Steiner tree problem [5]). The ML criteria selects the phylogenetic tree that has the highest probability of having given rise to the observed data under a specific stochastic model of evolution. If the ML criteria is used, then the lengths of the edges of the tree, which represent evolutionary time, must be estimated for each tree considered. In both cases, there may be more than one phylogeny with the optimal value of the criteria. The Steiner tree problem is known to be NP-hard [5]; while this has not been proven for the ML criteria, the problem behaves as if it is NP-hard as well.

Estimation of phylogenetic trees based on an optimality criteria is a difficult problem because the space of possible trees increases exponentially with the number of taxa. In general, the number of phylogenetic trees for a data set of n taxa is given by

$$\prod_{i=1}^{n-1}(2i-3). \tag{1}$$

For the ML criteria, we have the additional problem of inferring optimal edge lengths for the trees (there are $2n-3$ edges to consider for a tree with n taxa). Although the value of either criteria mentioned above can generally be computed rapidly for a given tree, the computation is not negligible. This, coupled with the enormous size of the search space for even moderately large problems, makes estimation of phylogenetic trees using an optimality criteria an extremely complex problem.

Numerous algorithms for estimating phylogenetic trees under optimality criteria have been proposed. Exact solutions can be obtained when 20 or fewer taxa are considered, using either exhaustive enumeration or a branch-and-bound algorithm. Exact methods will be discussed in Section 2. For larger data sets, heuristic searches must be employed. The heuristic most commonly used in practice, which will be discussed in Section 3, is an uphill search based on rearranging branches in the tree, until no possible rearrangements of a given tree result in an improvement of the optimality criteria. Another frequently used heuristic is the method of Star Decomposition, which will be discussed in Section 4. Section 5 describes heuristics based on divide-and-conquer techniques that have recently been developed for the phylogenetic tree problem. Methods which incorporate probabilistic searches have been applied to the phylogenetic tree problem and will be described in Section 6. We conclude with some discussion of the future of algorithm development for the phylogenetic tree problem.

2 Exact Methods

2.1 Exhaustive Search

When the number of taxa is not too large (generally less than 11 or so), it is possible to find the phylogenetic tree(s) with the optimal value of the criteria by enumerating all of the possible trees and evaluating the criteria for each of these trees. A simple method (Figure 1) can be used to enumerate the trees. An initial labeled tree is constructed using the first three taxa under consideration. All possible four

[1] The other commonly used procedure in phylogenetics is to compute a matrix of pairwise distances between the taxa, and use such distance information in constructing the tree. This is often done using a clustering technique or by seeking a tree with minimal length, as specified by the distances between taxa.

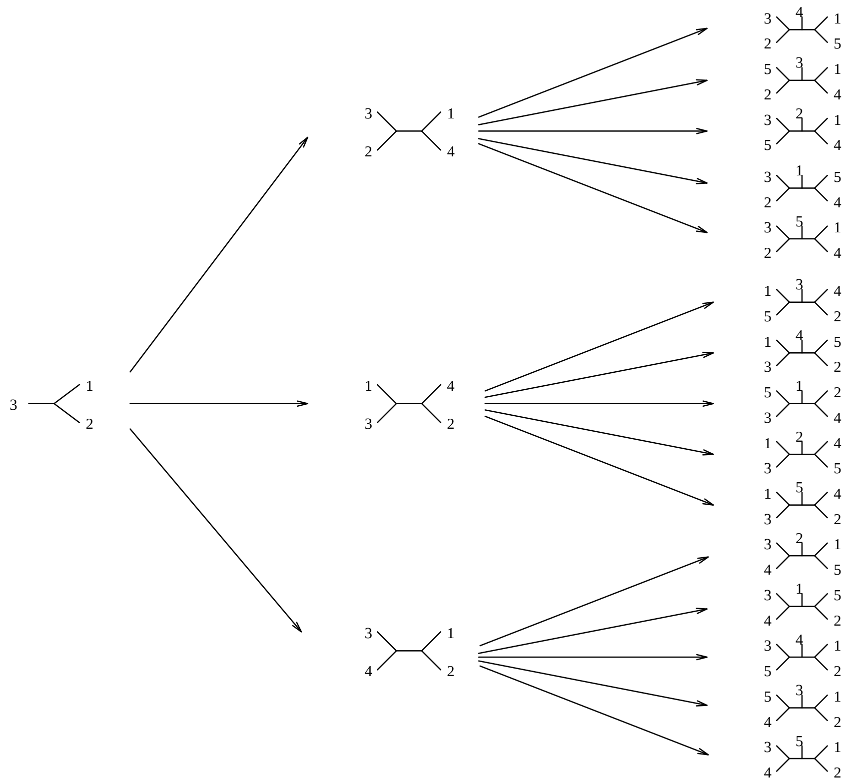

Figure 1: Enumeration of all possible trees for 5 taxa. An initial tree is formed from the first three taxa (first column). The fourth taxa is added to each possible branch of the initial tree to obtain the three trees in the middle column. The fifth taxa is then added to each possible branch in each of the four taxon trees to obtain the 15 trees in the third column. These 15 trees are the complete set of possible branching patterns for 5 taxa. Continuing in this manner would provide an enumeration of all possible trees for any number of taxa.

taxon trees are then constructed by adding the fourth taxa at all possible locations in the initial tree. Taxa are added in this manner until all taxa have been added, which results in the enumeration of all of the possible trees for n taxa. The optimality criteria is then evaluated for each of the trees, and the estimate of the phylogeny is the tree(s) that have the optimal value of the criteria.

2.2 Branch-and-Bound Methods

When the number of taxa exceeds 11 but is less than about 20, branch-and-bound methods may be used to find optimal phylogenetic trees under any criteria that is non-decreasing as additional taxa are added to the tree. Assuming that we want to minimize the value of the optimality criteria (which would be the case for the MP problem discussed in Section 1, for example), we first specify an upper bound, C, on the minimum value of the optimality criteria. We then consider the enumeration of the set of all trees for n taxa as discussed in the previous section (Figure 1). Specifically, we begin with the three taxon tree and then consider the first four taxon tree constructed from this initial tree. If the value of the optimality criteria exceeds the upper bound C that we have specified, then we need not consider any of the trees which are obtained by adding branches to this four taxon tree. If the tree does not exceed C, then we evaluate each of the five taxon trees obtained by adding a branch to this tree.

We proceed throughout the space of trees in this way. Whenever a tree exceeds C, we eliminate from consideration all trees obtained by adding branches to this tree, and then proceed backward to the largest k-taxon tree that did not exceed C. We then consider any $(k+1)$-taxon trees that have not been previously evaluated. If at any time in the search an n-taxon tree is encountered with a value C^* of the optimality criteria that is less than C, C is reset to C^*, and the search is continued.

The branch-and-bound method is useful in that it guarantees that a phylogenetic tree which optimizes the criteria is found. Clever programming, including an appropriate choice of the initial upper bound C, can greatly improve the running time of the algorithm [26]. A drawback of the method is that it does not necessarily give information about near-optimal trees - information that is often valuable in phylogenetic inference.

3 Branch-swapping Methods

For large data sets, the vast majority of phylogenetic tree estimates are obtained by using the uphill searching algorithms employed in popular phylogenetic analysis packages such as PAUP* [25], PHYLIP [7], and fastDNAML [17]. These packages all employ similar search strategies, which begin by building an initial tree through stepwise addition of taxa to the tree. The search begins with the unique three taxon tree formed from the first three taxa in the input data set. A four taxon tree is then constructed by considering all possible locations in the three taxon tree in which the fourth taxa could be inserted, and then selecting the four taxon tree which gives the best value of the optimality criteria. Taxa are added to the tree in this way until a tree containing all n of the taxa has been constructed. At each stage of addition of taxa to the tree, rearrangements of the current tree by some method of branch-swapping are attempted; if any single branch-swap results in a tree with a better value of the optimality criteria, that tree becomes the current tree to which the next taxa will be added.

After the initial n taxon tree has been constructed, thorough branch-swapping is performed to determine whether any nearby tree might have a better value of the optimality criteria. When a tree is found for which no improvement in the value of the optimality criteria can be made by any single branch-swapping move, that tree is returned as the estimate of the phylogenetic tree.

Although algorithms of this type are the most commonly used among biologists, their performance is not well understood. Empirical studies indicate that the time required by such algorithms is $O(n^3)$ [7, 23], and that the algorithms may often return estimates that are only locally optimal [17, 23]. While it is clear that the estimates obtained by such algorithms are dependent on the starting point (i.e., the order of the sequences in the input data set), little attention has been paid to the selection of good starting points. It is generally agreed that such algorithms must be run numerous times with random starting points to improve the likelihood that the optimal solution is found. However, for large problems (i.e., more than 40 sequences), the time required for this type of approach becomes prohibitive.

4 Star Decomposition Methods

The star decomposition method of obtaining an estimate of the phylogeny can be applied for any criteria that can be evaluated for multifurcating trees. The method begins with a tree with a single internal node from which n branches lead to each of the n external nodes (i.e., a star tree). Next, the $\binom{n}{2}$ possible trees that can be formed by pairing two of the taxa into a new group are evaluated, and the tree with the best value of the optimality criteria is selected. The number of branches leading from the central node is reduced by one, and the process is repeated, until a bifurcating tree is obtained.

The most popular of the star decomposition methods in widespread use is the neighbor-joining method [21], which is based on the distances between taxa rather than on optimizing the MP or ML criteria. Star decomposition algorithms have also been implemented for the ML problem in the case of both DNA and amino acid sequence data [3, 20, 27]. An advantage of the method is that phylogenetic trees can be estimated very rapidly. However, the algorithm is likely to return estimates that are only locally optimal. One use of estimates obtained by star decomposition methods is as starting points for branch-swapping algorithms, rather than building initial trees by stepwise addition as described in Section 3.

5 Divide-and-Conquer Techniques

Algorithms based on divide-and-conquer techniques have also been proposed for the phylogenetic tree estimation problem. Notable among these are the Short Quartet Methods [6], Disk-Covering Methods [9], and Quartet Puzzling [24]. The general idea behind these methods is that the problem of inferring the optimal phylogenetic tree for the n taxa under consideration can be divided into subproblems, which consist of inferring the phylogenetic tree for subsets of the n taxa. Finding optimal trees for the subsets should simplify the problem, since locating optimal phylogenetic trees is generally easier for smaller problems. Once estimates of the phylogeny are obtained for each of the subsets, the resulting subtrees are combined to obtain a supertree containing all of the n original taxa. An advantage of divide-and-conquer strategies is that they obtain estimates of the tree very quickly and appear to be reasonably accurate [9]. Implementation of such strategies can be complex, however, since they involve consideration of how the n taxa should be divided into subproblems and how best to construct a supertree from the various subtrees. Forming an overall supertree from the subtrees can be especially troublesome in the case in which the subtrees are incongruent.

6 Stochastic Search Methods

6.1 Simulated Annealing Methods

The simulated annealing algorithm has been applied to the phylogenetic tree problem by several authors [4, 5, 12, 23]. Each of these algorithms work by randomly rearranging a current tree, using, for example, one of the branch-swapping methods employed by the algorithms discussed in Section 3, and then evaluating the optimality criteria for the rearranged tree. A tree which has improved the value of the optimality criteria always replaces the current tree; however, a tree for which the value of the optimality criteria is worse than the value for the current tree might still be chosen to become the current tree. This choice is made based on the difference in values of the optimality criteria, and on a sequence of parameters that is commonly called the *cooling schedule* in the simulated annealing literature. The sequence of parameters is selected in such a way that the probability that a tree with a worse value of the optimality criteria is accepted decreases as the algorithm proceeds. The idea behind the algorithm is that its ability to move to trees with poorer values of the optimality criteria should allow the algorithm to avoid becoming trapped in locally optimal trees and hence improve its ability to find the globally optimal tree. Numerous results concerning the convergence of such algorithms under strict cooling schedules have been obtained [1, 2, 13], but these generally require prohibitive amounts of time to obtain the optimal solution. In practice, a compromise must be reached that allows a reasonable running time for the algorithm and still gives good ability to locate the global optimum.

The simulated annealing algorithms developed for the phylogenetic tree problem have had varying degrees of success. Barker [4] reports success for his algorithm, called LVB, for the MP problem when very large problems ($>$ 100 taxa) are considered. Salter [22] reported that simulated annealing was successful for the MP problem for problems of varying sizes, including smaller problems, but that the time required to obtain optimal solutions greatly exceeded the time required to perform many heuristic uphill searches of the type mentioned in Section 3 from random starting points. For the ML problem, however, the situation appears to be very different. Salter and Pearl [23] report that their simulated annealing algorithm outperforms commonly used heuristics such as PAUP* and PHYLIP both in terms of time and in terms of its ability to locate the globally optimal solution. In empirical studies, their algorithm required only about $O(n^2)$ time.

6.2 Genetic Algorithms

Genetic algorithms (GAs) have also been developed for the phylogenetic tree problem, notably by Lewis [10] for the ML problem using DNA sequences and by Matsuda [14] for the ML problem using amino acid sequences. GAs are based on the analogy of the genetic transmission of desirable characteristics to successive generations to the search for solutions that have certain optimal characteristics. GAs work by first requiring that a solution to the function of interest be representable in the form of a vector (this vector is analogous to a chromosome in the biological analogy). Next, a certain set of operations on the vectors are defined, each of which has an analogy to the biological setting. For example, two vectors may experience crossing over, in which case analogous sections of the two vectors are exchanged. GAs start with an initial population of vectors, and then generate a new population of progeny of these vectors according to the set of possible operations on the vectors and some rules about the survival of vectors to the next generation. The value of the optimality criteria (the "fitness value" in the biological analogy) is used to determine which members of a population will produce progeny for the next generation and which will themselves survive to the next generation. Such algorithms have been found to work well in many problems, including various instances of the traveling salesman problem [19].

The GAs of Lewis [10] and Matsuda [14] have provided a viable alternative for estimation of phylogenetic trees under the ML criteria. For example, Lewis demonstrated that his algorithm, called GAML, outperforms the commonly used program PAUP* for ML tree estimation for a problem with 55 taxa. One difficulty with the use of GAs is that many choices must be made in their implementation, e.g., the size of the population at each generation must be determined, a "cross over" between two trees must be defined, a "mutation" of one tree into another must be defined, etc. The sensitivities of GAs to the various choices available in their implementations have not been thoroughly studied, at least in the case of phylogenetic tree estimation.

6.3 Markov Chain Monte Carlo Methods

Recently, Markov Chain Monte Carlo (MCMC) methods have been applied to the phylogenetic tree problem by several authors [11, 15, 28]. These methods are useful when an estimate of the posterior distribution of phylogenetic trees under specific prior assumptions is desired. Generally, the estimate of the optimal phylogenetic tree is that tree which attains the highest posterior probability. MCMC methods have the advantage that a distribution of trees is given, and thus "confidence sets" can be constructed. In addition, the distribution of trees provided can be used to study the variability in any aspect of the phylogeny that is of interest. However, when the number of taxa is large, MCMC methods give poor estimates of the posterior probability of any individual phylogenetic tree. Furthermore, they may be computationally intensive when large data sets are considered.

7 Conclusions

The problem of finding optimal phylogenetic trees is important, as scientists are increasingly relying on phylogenetic reconstructions to form and test evolutionary hypotheses. Though much study has been devoted to solving this problem, there is still room for improvement of the methods described here for phylogenetic estimation. For example, systematic study of the performance of the various algorithms to determine which might be most appropriate in certain situations has not been undertaken. Furthermore, many of the algorithms described here readily lend themselves to parallel implementations. Although some work in the parallel programming of algorithms for phylogenetic tree estimation has been done, much of this study has been limited to "embarrassingly parallel" applications. For these reasons, extensions of the algorithms described here and development of new, more efficient algorithms for phylogenetic tree estimation remains an exciting area of study.

References

[1] E. Aarts and J. Korst. *Simulated Annealing and Boltzman Machines.* Wiley and Sons, First edition, 1989.

[2] E. Aarts and P. J. VanLaarhoven. A new polynomial time cooling schedule. *Proc. IEEE Int. Conf. On Computer-Aided Design, Santa Clara*, x:206–208, 1989.

[3] J. Adachi and M. Hasegawa. *MOLPHY: Programs for molecular phylogenetics I - PROTML: Maximum likelihood inference of protein phylogeny.* Computer Science Monographs, No. 27. Institute of Statistical Mathematics, Tokyo, 1992.

[4] D. Barker. *LVB 1.0: Reconstructing Evolution with Parsimony and Simulated Annealing.* University of Edinburgh, 1997.

[5] A. Dress and M. Kruger. Parsimonious phylogenetic trees in metric spaces and simulated annealing. *Advances in Applied Mathematics*, 8:8–37, 1987.

[6] P. Erdos, M. Steel, L. Szekely, and T. Warnow. Local quartet splits of a binary tree infer all quartet splits via one dyadic inference rule. *Computers and Artificial Intelligence*, 16(2):217–227, 1997.

[7] J. Felsenstein. *Phylogenetic Inference Package (PHYLIP), Version 3.5.* University of Washington, Seattle, 1993.

[8] D. M. Hillis and J. P. Hulsenbeck. Support for dental HIV transmission. *Nature*, 369:24–25, 1994.

[9] D. Huson, S. Nettles, and T. Warnow. Disk-covering, a fast converging method for phylognetic tree reconstruction. *Journal of Computational Biology*, 6(3):369–386, 1999.

[10] P. Lewis. A genetic algorithm for maximum-likelihood phylogeny inference using nucleotide sequence data. *Molecular Biology Evolution*, 15(3):277–283, 1998.

[11] S. Li, D. Pearl, and H. Doss. Phylogenetic tree construction using Markov chain Monte Carlo. *Journal of the American Statistical Association, to appear*, 2000.

[12] M. Lundy. Applications of the annealing algorithm to combinatorial problems in statistics. *Biometrika*, 72(1):191–198, 1985.

[13] M. Lundy and A. Mees. Convergence of an annealing algorithm. *Mathematical Programming*, 34:111–124, 1986.

[14] H. Matsuda. Protein phylogenetic inference using maximum likelihood with a genetic algorithm. In *Pacific Symposium on Biocomputing*, pages 512–523, 1996.

[15] B. Mau, M. Newton, and B. Larget. Bayesian phylogenetic inference via Markov chain Monte Carlo methods. *Biometrics*, 55:1–12, 1999.

[16] F. E. McCutchan, M. O. Salminen, J. K. Carr, and D. S. Burke. HIV-1 genetic diversity. *AIDS*, 10 (suppl 3):S13–S20, 1996.

[17] G. Olsen, H. Matsuda, R. Hagstrom, and R. Overbeek. FastDNAml: A tool for construction of phylogenetic trees of DNA sequences using maximum likelihood. *Computations in Applied Biosciences*, 10(1):41–48, 1994.

[18] C.-K. Ong, S. Nee, A. Rambaut, H.-U. Bernard, and P. H. Harvey. Elucidating the population histories and transmission dynamics of papillomaviruses using phylogenetic trees. *Journal of Molecular Evolution*, 44:199–206, 1997.

[19] C. R. Reeves. *Modern Heuristic Techniques for Combinatorial Problems.* John Wiley and Sons, Inc., 1993.

[20] N. Saitou. Maximum likelihood methods. *Meth. Enzymol.*, 183:584–598, 1990.

[21] N. Saitou and M. Nei. The neighbor-joining method: a new method for reconstructing phylogenetic trees. *Molecular Biology Evolution*, 4:406–425, 1987.

[22] L. Salter. *Simulation-based estimation of phylogenetic trees.* PhD thesis, The Ohio State University, Columbus, OH 43210, 1999.

[23] L. Salter and D. Pearl. A stochastic search strategy for estimation of maximum likelihood phylogenetic trees. *in revision, Systematic Biology*, 2000.

[24] K. Strimmer and A. von Haeseler. Quartet puzzling: A quartet maximum-likelihood method for reconstructing tree topologies. *Molecular Biology and Evolution*, 13(7):964–969, 1996.

[25] D. Swofford. *PAUP*. Phylogenetic analysis using parsimony (* and other methods). Version 4.* Sinauer Associates, 1998.

[26] D. Swofford, G. Olsen, P. Waddell, and D. Hillis. *Phylogenetic Inference, in Molecular Systematics (edited by D. Hillis, C. Moritz, and B. Mable)*, pages 407–514. Sinauer Associates, Inc., Second edition, 1996.

[27] Z. Yang. *Phylogenetic Analysis by Maximum Likelihood (PAML), Version 1.3.* Department of Integrative Biology, University of California at Berkeley, 1997.

[28] Z. Yang and B. Rannala. Bayesian phylogenetic inference using DNA sequences: a Markov chain Monte Carlo method. *Molecular Biology Evolution*, 14(7):717–724, 1997.