Solving Large Scale Phylogenetic Problems using DCM2

Daniel H. Huson
Applied and Computational Mathematics
Princeton University
Princeton NJ USA
e-mail: huson@math.princeton.edu

Lisa Vawter
Bioinformatics
SmithKline Beecham
King of Prussia PA USA
e-mail: lisa_vawter@sbphrd.com

Tandy J. Warnow
Department of Computer Science
University of Arizona
Tucson AZ USA
e-mail: tandy@cs.arizona.edu

Abstract

In an earlier paper, we described a new method for phylogenetic tree reconstruction called the *Disk Covering Method*, or *DCM*. This is a general method which can be used with any existing phylogenetic method in order to improve its performance. We showed analytically and experimentally that when DCM is used in conjunction with polynomial time distance-based methods, it improves the accuracy of the trees reconstructed. In this paper, we discuss a variant on DCM, that we call *DCM2*. DCM2 is designed to be used with phylogenetic methods whose objective is the solution of NP-hard optimization problems. We show that DCM2 can be used to *accelerate* searches for Maximum Parsimony trees. We also motivate the need for solutions to NP-hard optimization problems by showing that on some very large and important datasets, the most popular (and presumably best performing) polynomial time distance methods have poor accuracy.

Introduction

The accurate recovery of the phylogenetic branching order from molecular sequence data is fundamental to many problems in biology. Multiple sequence alignment, gene function prediction, protein structure, and drug design all depend on phylogenetic inference. Although many methods exist for the inference of phylogenetic trees, biologists who specialize in systematics typically compute Maximum Parsimony (MP) or Maximum Likelihood (ML) trees because they are thought to be the best predictors of accurate branching order. Unfortunately, MP and ML optimization problems are NP-hard, and typical heuristics use hill-climbing techniques to search through an exponentially large space. When large numbers of taxa are involved, the computational cost of MP and ML methods is so great that it may take years of computation for a local minimum to be obtained on a single dataset (Chase *et al.* 1993; Rice, Donoghue, & Olmstead 1997). It is because of this computational cost that many biologists resort to distance-based calculations, such as Neighbor-Joining (NJ) (Saitou & Nei 1987), even though these may have poor accuracy when the diameter of the tree is large (Huson *et al.* 1998).

As DNA sequencing methods advance, large, divergent, biological datasets are becoming commonplace. For example, the February, 1999 issue of Molecular Biology and Evolution contained five distinct datasets of more than 50 taxa, and two others that had been pruned below that number for analysis because the large number of taxa made analysis difficult. Problems that require datasets that are both large and divergent include epidemiological investigations into HIV and dengue virus (Crandall *et al.* 1999; Holmes, Worobey, & Rambaut 1999), the relationships among the major life forms, e.g. (Baldauf & Palmer 1993; Embley & Hirt 1999; Rogers *et al.* 1999), and gene family phylogenies, e.g. (Modi & Yoshimura 1999; Gu & Nei 1999; Matsuika & Tsunewaki 1999).

Distant relationships are crucial to the understanding of very slowly-evolving traits. For example, one cannot hope to understand the transmission of HIV from non-human primates if one cannot place chimpanzee and human viruses on the same tree. In this case, the dataset contains distant sequences, because the molecular sequences evolve rapidly, and one must use a broad tree in order to map differences in the slowly-evolving trait of transmission onto that tree. One cannot understand the evolution of DNA and RNA-based life forms if one cannot place sequences from these viruses onto the same tree. One cannot understand the evolution of active sites in gene family members if one cannot place diverse members on a single tree. Clearly, algorithms for solving large, diverse phylogenetic trees will benefit the biological community.

The need for an accurate estimator of branch order has spurred our research into the questions: Are there fast (polynomial time) methods for phylogenetic reconstruction that are accurate with large numbers of taxa and across large evolutionary distances, or must we find good solutions to NP-hard optimization problems? If we must "solve" NP-hard optimization problems, can we discover techniques that will allow this to be done rapidly?

Our paper makes two contributions: First, we provide a comparative performance analysis of some of the best methods widely used among biologists: NJ, its relatives, BIONJ (BJ) (Gascuel 1997) and Weighbor (WJ)

(Bruno, Socci, & Halpern 1998), and heuristic MP, as implemented in the popular software package PAUP (Swofford 1996). We analyze datasets with large numbers of taxa using these methods, and show that the distance-based methods produce trees that are inferior to the trees produced by heuristic MP. Further, we show that the MP heuristic does not converge to a local optimum in a reasonable length of time on these datasets.

Our second contribution is a new technique for accelerating the solution of hard optimization problems on large phylogenetic datasets. This new technique is a variant of the Disk Covering Method (DCM), a divide-and-conquer method presented in (Huson, Nettles, & Warnow 1999). Unlike DCM, DCM2 is designed specifically to improve time performance of MP and ML methods. We present experimental evidence of this acceleration with MP on simulated datasets, and offer additional evidence that this method should work well with real biological data.

The paper is organized as follows: We describe Markov models of evolution and their utilization in our performance studies of phylogenetic reconstruction. We explain why divide-and-conquer is a natural approach to improving performance of MP. Finally, we describe our divide-and-conquer strategy, and present our experimental study of its performance on real and simulated data.

Phylogenetic Inference Under Markov Models of Evolution

The phylogenetic tree reconstruction problem can be formulated as a problem of statistical inference (or machine learning), in which the given sequence data are assumed to have been generated on a fixed but unknown binary tree under a Markov process. Thus, evolutionary events that happen below a node are assumed to be unaffected by those that happen outside of that subtree.

For example, the *Jukes-Cantor* model, a model of DNA sequence evolution, describes how 4-state sites (positions within the sequences) evolve identically and independently down a tree from the root. In the Jukes-Cantor model, the number of mutations on each edge of the tree is Poisson-distributed, and transitions between each pair of nucleotides are equally likely. While these assumptions are not entirely realistic, the standard technique for exploring performance of different phylogenetic methods within the biology community is based upon studying performance under simulations that use models similar to the Jukes-Cantor model.

Interesting theoretical results have been obtained for the Jukes-Cantor model, and for models in which the sites have different rates that are drawn from a known distribution. For example, it has been shown that many methods are guaranteed to be statistically consistent– they will obtain the true topology, given long enough sequences. Most distance-based methods (i.e. methods which estimate the number of mutations on each leaf-to-leaf path and use that estimated matrix to construct an edge-weighted tree) fall into this category. Neighbor-Joining (Saitou & Nei 1987) is an example of such a method. MP and ML are not distance-based methods, but instead use the input set of biomolecular sequences to infer the tree; the optimization criteria for MP and ML are different but related (see (Tuffley & Steel 1997)).

In this paper we describe a new technique for speeding up heuristics for NP-hard optimization problems in phylogenetics, and we shall explore its performance specifically with respect to the Maximum Parsimony problem. The Maximum Parsimony problem is the *Hamming Distance Steiner Tree Problem*, and is as follows. Let T be a tree in which every node is labelled by a sequence of length k over an alphabet $\mathcal{A}$. The *length* (also called the parsimony score) of the tree T is the sum of the Hamming distances of the edges of the tree, where the Hamming distance of an edge (v, w) is defined by $H(v, w) = |\{i : v_i \neq w_i\}|$ (i.e. the number of positions that are different). The Maximum Parsimony Problem is:

- **Input:** Set S of n sequences of length k over the alphabet $\mathcal{A}$.
- **Output:** Tree T with n leaves, each labelled by a distinct element in S, and additional sequences of the same length labelling the internal nodes of the tree, such that the length of T is minimized.

Thus, MP is an optimization problem whose objective is to find the tree with minimum length (i.e. parsimony score). Most methods for "solving" Maximum Parsimony operate by doing a hill-climbing search through the space of possible leaf-labelled trees, and computing the parsimony score of each considered tree. Computing the parsimony score of a given tree is polynomial, but the size of the search space is exponential in the number of leaves, and hence these methods are computationally very expensive. Maximum Likelihood, however, does not have a polynomial time point estimation procedure; thus, computing the Maximum Likelihood score of a given tree is itself computationally expensive. For this reason, although ML is statistically consistent and has many of the nice properties that MP has, it has not been the method of choice of biologists on even moderate-sized datasets.

Because MP is NP-hard and not always statistically consistent (Felsenstein 1978), one might ask *why use MP?* Biologists generally prefer MP because, with typical sequence lengths, it is more accurate than distance methods, and because MP makes specific biological predictions associated with the sequences at internal nodes. Distance-based methods make no such predictions.

Our own studies suggest that the the accuracy of both heuristic MP (as implemented in PAUP) and NJ may be comparable on many trees, as long as the number of leaves and tree diameter are small. However, on trees with high leaf number and large diameter, MP outperforms NJ, sometimes quite dramatically (Rice & Warnow 1997; Huson *et al.* 1998). The poor accuracy of

NJ at realistically short sequence lengths is in keeping with the mathematical theory about the convergence rate of NJ, which predicts that NJ will construct the true tree with high probability if the sequence length is exponential in the maximum evolutionary distance in the tree (Erdős *et al.* 1999). (The evolutionary distance between two leaves is the expected number of mutations of a random site on the path in the tree between the two leaves; this can be unboundedly large.)

Thus, distance methods seem to require longer sequences than biologists usually have (or could even get, even if all genomes were sequenced!), in order to get comparable accuracy to Maximum Parsimony. Hence, the convergence rate to the true tree (the rate at which the topological error decreases to 0) is a more important practical issue than is the degree of accuracy of an algorithm given infinite sequence length. Additionally, some biologists argue that the conditions under which MP is not statistically consistent are biologically unrealistic, and are thus not pertinent (e.g. (Farris 1983; 1986)). Finally, under a more general (and more biologically realistic) model of evolution, it has been recently shown that MP and ML are identical, in the sense that on every dataset, the ordering of tree topologies with respect to ML and MP are identical (Tuffley & Steel 1997). Thus, if the objective is the topology of the evolutionary tree and nots its associated mutation parameters, then solving MP is *equivalent* to solving ML under a biologically realistic model.

Performance of Polynomial Time Methods on Real Data

In this section, we illustrate the performance of three of the most promising polynomial time distance methods, NJ, BJ, and WJ, on three large and biologically important datasets that are considered by biologists to be difficult because of their large leaf number and large evolutionary distances between the leaves. These datasets are: (1) Greenplant221, (2) Olfactory252, and (3) rbcL436.

***rbc*L436 and greenplant221 datasets:** Because green plants are one of the dominant life forms in our ecology–they provide us with foods and medicines, and even oxygen–they are prominent subjects of study for biologists. *rbc*L (ribulose 1,5-biphosphate carboxylase large subunit) is a chloroplast-encoded gene involved in carbon fixation and photorespiration. Chase et al. (Chase *et al.* 1993) published one of the most ambitious phylogenetic analyses to date of 476 distinct *rbc*L sequences in an attempt to infer seed plant phylogeny. This work represents one of the largest collaborative efforts in the field of systematic biology, and has proved controversial (Rice, Donoghue, & Olmstead 1997), not only because of the visibility and importance of this particular phylogenetic problem, but because there is no accepted "good" method in the biological systematics community for phylogenetic analysis of such a large dataset. We have selected a subset of 436 sequences from this large dataset to form our rbcL436 dataset.

Largely as a result of the controversy following the Chase et al. publication, the plant systematics community organized the Green Plant Phylogeny Research Coordination Group (GPPRCG), funded by the US Department of Agriculture, to tackle the organization of data collection and analysis for this important phylogenetic problem. The GPPRCG realized that this issue of phylogenetic analysis of large datasets was crucial to them, and thus proposed a benchmark dataset at their joint meeting with the 1998 Princeton/DIMACS Large Scale Phylogeny Symposium.

The GPPRCG benchmark dataset consists of 18s ribosomal DNA (rDNA) sequence from 232 carefully-chosen exemplar plant taxa (Soltis *et al.* 1997). 18s rDNA is a slowly-evolving nuclear gene that is widely used in phylogenetic analysis of distantly-related taxa. Challenges issued by the GPPRCG for analysis of this dataset include rapidly finding shortest trees and exploring the effects of analyzing and recombining subsets of the data. We selected 221 taxa from this dataset of 232 to form our greenplant221 dataset.

Olfactory252: Olfactory (smell, taste and sperm cell surface) receptor genes are the most numerous subfamily of G protein-coupled receptors (GPCRs), the largest eukaryotic gene family (e.g. (Skoufos *et al.* 1999)). Because GPCRs are the basis of much animal cell-to-cell communication as well as sensing of the environment, understanding of their evolutionary history contributes to the understanding of animal physiology and to our own evolution as multicellular organisms. We have chosen a set of 252 olfactory receptors that (presumably) have small, fat-soluble molecules as ligands (Freitag *et al.* 1998) for our olfactory252 dataset.

Performance on These Datasets

All flavors of NJ analysis (NJ, BJ, WJ) performed badly on all datasets relative to MP trees and relative to biological expectations. The most egregious example of this was that none of the polynomial time methods succeeded in resolving eudicots from monocots in either the *rbc*L436 or the greenplant221 dataset. Eudicots, "advanced" plants with specific seed, flower, pollen, vasculature, root and other characteristics, are easily resolved by MP analyses (Chase *et al.* 1993; Rice, Donoghue, & Olmstead 1997). The various flavors of NJ also failed to resolve Rosidae (rose family), Poaceae (grass family) and Fabaceae (bean or legume family) placing each of the genera within these families distant from each other on the tree. Many other differences between the polynomial time analyses and MP analysis and botanical "knowledge" exist; these are simply the most egregious.

The three polynomial time methods we studied performed similarly poorly on the olfactory252 dataset. Genes within the olfactory receptor family evolve by

duplication on the chromosome, followed by divergence (Sullivan, Ressler, & Buck 1994). Thus genes that are near to each other on a chromosome are likely to be close relatives. MP analysis of the olfactory receptor confirms, in large part, the duplication and divergence scenario (5 of 6 groups of neighboring olfactory receptors group together on the MP tree). With the polynomial time methods, these 6 groups of neighboring receptors are split into 13-15 groups on the tree.

Why Divide-and-Conquer?

In previous sections, we provided empirical evidence that polynomial time distance-based methods are insufficiently accurate on some important large biological datasets, and we cited analytical evidence that the reason for this may be the maximum evolutionary distance within these datasets. Thus, we argue that distance-based methods may have problems recovering accurate estimates of evolutionary trees when the datasets contain high evolutionary distances. Methods such as MP and ML are not expected to have the same problem with large evolutionary distances (see, in particular, (Rice & Warnow 1997), an experimental study which demonstrated a sublinear increase in sequence length requirement as a function of the diameter of the tree), but large numbers of taxa cause these sorts of analyses to take so much time that they are unwieldy.

We believe that a divide-and-conquer strategy may provides a means by which large datasets can be more accurately analyzed. Our reasons include statistical, computational and biological rationales for such an approach:

- Statistical reasons: Under *i.i.d.* Markov models of evolution, the sequence length that suffices to guarantee accuracy of standard polynomial time methods is *exponential* in the maximum evolutionary distance in the dataset (Erdős *et al.* 1999). Thus, if the divide-and-conquer strategy can produce subproblems each of which has a smaller maximum evolutionary distance in it, then the sequence length requirement for accuracy on the subproblems will be reduced. Experimental evidence suggests also that at every fixed sequence length, the accuracy of the distance methods decreases exponentially with increased divergence (Huson *et al.* 1998). Consequently, accuracy on subproblems will be higher than accuracy on the entire dataset, if the subproblems have lower divergence in them.
- Computational reasons: Large datasets are computationally challenging for methods which solve or attempt to solve NP-hard optimization problems. Data sets with even 50 leaf nodes can occupy a lab's single desktop computer for months, which is an unreasonable time claim on a machine which must perform other phylogenetic analyses as well as many other functions for the lab. Data sets of 100 or more taxa can take years (one dataset of 500 *rbc*L genes is still being analyzed, after several years of computation (Chase *et al.* 1993; Rice, Donoghue, & Olmstead 1997)). Smaller subproblems are generally analyzed more quickly, and heuristics performed on smaller subproblems are more accurate (since they can explore proportionally a greater amount of the tree space).
- Biological reasons:
  1. Because little is known about the effects of missing data on various methods of phylogenetic analysis, a biologist may be hesitant to include taxa for which not all sequence information is present. Additionally, some taxa may have naturally-occurring missing data resulting from insertions or deletions. Data set decomposition will allow direct comparison of sets of taxa for which comparable data are available. Data set decomposition increases the amount of sequence data available to each subproblem (so that the sequence length in the subproblem analyses is larger than that common to the full dataset), thus increasing accuracy in the estimation of the subproblems, as compared to the accuracy of the full problem.
  2. Many tantalizing biological problems, e.g. viral phylogenies, many gene family phylogenies, and the phylogeny of all living organisms, comprise organisms that are so distantly-related that a single multiple alignment of all sequences involved is difficult, if not impossible. Data set decomposition requires alignment only of sequences in the various subsets for phylogenetic analysis, as opposed to requiring global multiple alignment. This is not only computationally easier, but more likely to be accurate (even the approximation algorithms are computationally expensive, and most alignments are adjusted by eye). Thus, reduction to subproblems involving closely related taxa is likely to improve the accuracy of the underlying multiple alignment, and hence also of the phylogenetic tree reconstructed on the basis of the multiple alignment.
  3. Interesting problems in systematics are often involved large numbers of taxa that are distantly-related. The phylogenetic analysis necessary for study of cospeciation of hosts and parasites (Page *et al.* 1998); Caribbean bird biogeography (Burns 1997); HIV evolution (Paladin *et al.* 1998); human origins (Watson *et al.* 1997); the prediction of apolipoprotein E alleles that may be associated with Alzheimer's disease (Tempelton 1995) all are such problems!

Thus, a divide-and-conquer approach is a attractive solution because it attacks both the large evolutionary distance barrier to the use of distance-based methods and the large leaf number barrier to MP and ML methods. Divide-and-conquer approaches may also alleviate real-data issues, such as missing character data, heterogeneous data, and problems with multiple alignments.

In the following section, we will describe our divide-and-conquer strategy for reconstructing evolutionary trees. Our technique is closely based upon an earlier technique, called the *Disk-Covering Method*, or DCM. Therefore, we have decided to call our new method *DCM2*.

Description of DCM1 and DCM2

In (Huson, Nettles, & Warnow 1999), we described the first Disk-Covering Method, which we will call DCM1, and demonstrated both theoretically and experimentally (using simulations of sequence evolution) that the use of DCM1 could greatly improve the accuracy of distance based methods for tree reconstruction. The new technique we now propose, DCM2, has much the same structure, but differs in the specific instantiations of the technique.

General Structure of DCM

The input to both DCM1 and DCM2 is a set $S = \{s_1, \ldots, s_n\}$ of n aligned biomolecular sequences, and a matrix d containing an estimate of their interleaf distances. DCM1 and DCM2 operate in two phases. In the first phase, for some (or possibly all) of the $q \in \{d_{ij}\}$, a tree T_q is constructed. In the second phase, a "consensus tree" of all the T_q is obtained. The difference between DCM1 and DCM2 techniques lies primarily in how the tree T_q is constructed.

Phase I of DCM

For both DCM1 and DCM2, the construction of the tree T_q has three basic steps:

1. Decompose the dataset into smaller, overlapping subsets, so that within each subset there is less evolutionary distance than across the entire dataset,
2. Construct trees on the subsets, using the desired phylogenetic method, and
3. Merge the subtrees into a single tree, encompassing the entire dataset.

The Threshold Graph

Both DCM1 and DCM2 obtain the decomposition in the first step by computing a *threshold graph*, $G(d, q)$, defined as follows:

- The vertices of $G(d, q)$ are the taxa, $s_1, s_2, \ldots, s_n$.
- The edges of $G(d, q)$ are those pairs (s_i, s_j) such that $d_{i,j} \leq q$.

The graph $G(d, q)$ can be considered an edge-weighted graph, in which the weight of edge (i, j) is $d_{i,j}$. It is then minimally *triangulated*, which means that edges are added to the graph so that it no longer possesses any induced cycles of length four or greater (Buneman 1974; Golumbic 1980), but the weight of the largest edge added is minimized. Obtaining an optimal triangulation of a graph is in general NP-hard (Bodlaender, Fellows, & Warnow 1992), but graphs that arise as threshold graphs will be *triangulated or close to triangulated*, as we showed in (Huson, Nettles, & Warnow 1999). For graphs that are close to triangulated, optimal triangulations can be obtained very quickly using simple techniques. We have implemented a polynomial time greedy heuristic for triangulating the threshold graphs we obtain, that attempts to minimize the largest weight of any edge added; in our experiments, the heuristic has generally added very few edges.

This technique for dataset decomposition has two advantages, each of which contributes to the efficiency of DCM: Triangulated graphs are quite special, in that various NP-hard problems can be solved in polynomial time when restricted to triangulated graphs. Also, triangulated graphs have the property that the minimal vertex separators are *cliques*, and there are only a linear number of them, all of which can be found in polynomial time.

Difference Between DCM1 and DCM2

After the initial decomposition, DCM1 and DCM2 diverge. DCM1 computes the *maximal cliques* (that is, cliques which cannot be enlarged through the addition of vertices without losing the clique property); these maximal cliques define the subproblems for which DCM1 constructs trees. We refer to the technique used by DCM1 to obtain a decomposition as the *maxclique* decomposition technique.

DCM2, by contrast, does the following (see Figure 1): Let S be the input set of sequences, d the associated interspecies distance matrix, let $q \in \{d_{ij}\}$ be the selected *threshold*, and let G be the triangulation of the threshold graph $G(d, q)$.

(a) We compute a vertex separator X so that
   - X is a maximal clique, and
   - $G - X$ has components $A_1, A_2, \ldots, A_r$, so that $\max_i |X \cup A_i|$ is minimized.

(b) We construct trees t_i for each $A_i \cup X$.

(c) We *merge* the trees t_i, $i = 1, 2, \ldots, r$, into a single tree T_q on the entire dataset.

We will refer to this as the *dac* (divide and conquer) decomposition technique.

For triangulated graphs, finding all maximal cliques and finding the optimal vertex separator are polynomial time problems, and in the next section, we will describe the polynomial time method we use for merging subtrees into a single tree. Furthermore, we have found that our polynomial time heuristic for triangulating the threshold graphs suffices to obtain good accuracy. Consequently, as we have implemented them, both DCM1 and DCM2 are polynomial time meta-methods, but they need a specified phylogenetic method (which we call the "base method") in order to construct trees on subproblems. In this paper we will always use heuristic Maximum Parsimony as the base method.

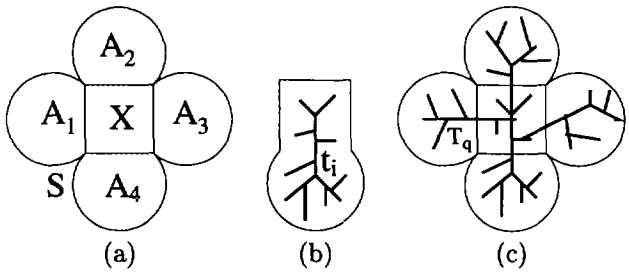

Figure 1: Here we schematically indicate the three steps that make up Phase I of DCM2. (a) First, a clique separator X for the taxon set S is computed (relative to the given threshold graph $G(d,q)$), producing subproblems $A_1 \cup X, A_2 \cup X, \ldots, A_r \cup X$. (b) Then, a tree t_i is computed for each subproblem $A_i \cup X$ using the specified base method. (c) Finally, the computed subtrees are merged together to obtain a single tree T_q for the whole dataset.

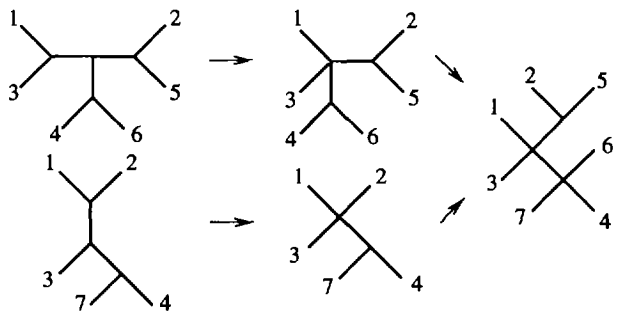

Figure 2: Merging two trees together, by first transforming them (through edge contractions) so that they induce the same subtrees on their shared leaves.

Merging Subproblems

The *Strict Consensus Subtree Merger (SCM)* method (described in detail below) takes two trees t and t' on possibly different leaf sets, identifies the set of leaves X that they share, and modifies t and t' through a minimal set of edge contractions, so that they induce the same subtree on X. Once they are modified in this way, the two trees can be merged together; see Figure 2. A collection of more than two trees is merged sequentially. For DCM2, although not for DCM1, the order of merging is irrelevant.

We call this the *Strict Consensus Subtree Merger* because the definition of the tree that will be induced on X after the merger is the "strict consensus" (Day 1995) of the two initially induced subtrees. This is defined to be the maximally resolved tree that is a common contraction of the two subtrees. We will call this subtree on X the *backbone*. Merging the two trees together is then achieved by *attaching* the pieces of each tree appropriately to the different edges of the backbone.

It is worth noting that the Strict Consensus Subtree Merger of two trees, while it always exists, may not be unique. In other words, it may be the case that some piece of each tree attaches onto the same edge of the backbone. We call this a *collision*. For example, in Fig. 2, the common intersection of the two leaf-sets is $X = \{1,2,3,4\}$, and the strict consensus of the two subtrees induced by X is the 4-star. This is the backbone, it has four edges, and there is a *collision* on the edge of the backbone incident to leaf 4, but no collision on any other edge. Collisions are problematic, as the Strict Consensus Subtree Merger will potentially introduce false edges or lose true edges when they occur. However, in (Huson, Nettles, & Warnow 1999), we showed that SCM is guaranteed to recover the true tree, when the input subtrees are correct and the triangulated graph is "big enough". In fact, the following can be proven using techniques similar to those in (Huson, Nettles, & Warnow 1999):

Theorem 1 *Let T be the true tree, and let $A_1, A_2, \ldots, A_p$ be the subproblems defined by either the maxclique- or dac-decompositions, for $q \geq$ d-width(T). If SCM is applied to trees $T_i = T|A_i$ for $i = 1,2,\ldots,p$, then T is reconstructed.*

Thus, if $q \geq$ *d-width*(T) (an amount to be the largest interleaf distance in a "short quartet", see (Huson, Nettles, & Warnow 1999) for details), then the true tree is reconstructed, given accurate subtrees. Exactly how to estimate q without knowing T is, however, an interesting and open question. In (Huson, Nettles, & Warnow 1999), we showed that when the sequences are long enough, it is not difficult to find the value $d-$*width*(T) for which every subtree is correct, and hence get the true tree on the whole dataset. For sequences that are too short, the problem of estimating q remains open.

Why dac-decompositions are preferable

There are two primary differences between a dac-decomposition and a maxclique-decomposition: dac-decompositions produce larger subproblems than maxclique-decompositions, but they also produce a much smaller number of subproblems (see Table 1). As a result, when the subtrees are merged, there is generally *less loss of resolution* in the DCM2 tree than in the DCM1 tree, as we will see in our experiments later in the paper. This is one reason that dac-decompositions are preferable in general. We are interested in getting a good estimate of the tree, but the DCM1 technique, although potentially faster than the DCM2 technique because the problems are smaller, produces trees that are too unresolved to be of interest.

Phase II of DCM

The general DCM technique allows for any collection of $q \in \{d_{ij}\}$ to be used for tree reconstruction, and then follows this with Phase II, in which a consensus of the different trees is made. For our purposes in DCM2, we will only select one q, and hence will not take a consensus of the resultant trees; consequently, Phase II

in DCM2 is different. The reason for our restriction to one q is purely computational: running Maximum Parsimony or Maximum Likelihood is computationally intensive, and the point of using DCM2 rather than standard heuristic MP or heuristic ML is to get a faster estimate of the true tree that is as good as that obtained through standard techniques.

For our experiments in this paper, we selected q to be the smallest d_{ij} for which the threshold graph is connected. This is not the optimal choice, if what we wish is maximum accuracy, as we will show later in our study, but we get reasonably good estimates of the tree this way, and much faster than by using more standard techniques.

Optimal Tree Refinement Heuristics

After selecting q and computing the subtrees t_i, we merge subtrees. Chances are great that this merger will contract edges, thus resulting in a tree T_q that is not binary, and may in fact have a significant loss of resolution. Consequently, we follow the construction of T_q by a phase in which we attempt to find the *optimal refinement* of the tree. Thus, we will attempt to solve the *Optimal Tree Refinement* (OTR) Problem with respect to a criterion, π, such as the Maximum Parsimony criterion, or the Maximum Likelihood criterion. We call this the $OTR - \pi$ problem.

Optimal Tree Refinement

- *Input:* A tree T leaf-labelled by sequences S and an optimization criterion, π.
- *Output:* A tree T' that is topologically a refinement of T (i.e. there will be a set $E_0 \subset E(T')$ of edges in T', such that the contraction of these edges in T' results in T), such that T' optimizes the criterion π.

For optimization criteria π that are NP-hard to solve, the $OTR - \pi$ problem is also NP-hard, but there is a potential that the problem may be solvable in polynomial time for bounded degree trees. We do not have any polynomial time algorithms for OTR-Parsimony, (however see (Bonet *et al.* 1998) for what we have established for this problem), so we designed heuristics to approximate OTR-Parsimony.

In this paper we will explore one such heuristic, which we call the IAS technique (for *Inferring Ancestral States*) (Vawter 1991; Rice, Donoghue, & Olmstead 1997). In this approach, we first infer ancestral sequences for all internal nodes of the tree so as to minimize the parsimony score of the sequences for the given tree topology. Then we attempt to resolve the tree around each incompletely resolved node v by applying heuristic MP to the set of sequences associated with the set of neighbors of v. We then replace the unresolved topology around v by the (generally) more refined topology obtained on this set.

Alternatively, instead of inferring sequences for internal nodes, another standard method would be to use nearest leaves in each subtree around an unresolved node to resolve the tree. Although not shown here, our preliminary experiments have indicated that this technique is generally inferior to the IAS technique with respect to topological accuracy and also with respect to parsimony scores.

Experimental Studies Involving DCM2

DCM2 is designed to get maximal performance for use with methods or heuristics for NP-hard optimization problems on real datasets. We therefore had several issues to address in order to get this optimal performance. We report on four basic experiments: *Experiment 1*, which addresses the selection of q, *Experiment 2*, which addresses the effects of the IAS heuristic for OTR on the accuracy of the reconstructed tree, *Experiment 3*, which compares DCM1 followed by Optimal Tree Refinement (DCM1+OTR) to DCM2+OTR, and *Experiment 4*, which compares the more successful of the two methods in *Experiment 3* to heuristic MP. In order to explore these questions, we need a measure of accuracy.

Measures of Accuracy

Let T be the true or model tree, and let T' be the inferred tree. Thus, T and T' are both leaf-labelled by a set S of taxa. Each edge e in T defines a bipartition π_e of S into two sets in a natural way, so that the set $C(T) = \{\pi_e : e \in E(T)\}$ uniquely defines the tree T. Similarly, $C(T')$ can be defined. Topological error is then quantified in various ways, by comparing these two sets. For example, *false negatives (FN)* are the edges of the true tree that are missing from the inferred tree; this is the set $C(T) - C(T')$. *False positives (FP)* are edges in the inferred tree that are not in the true tree; this is the set $C(T') - C(T)$.

Experimental Setup

Our experiments examined both real and simulated datasets. Because of the computational issues involved in solving MP, we have used a somewhat restricted version of heuristic MP, in which tree-bisection-reconnection branch swapping is used to explore tree space, but we limit the number of trees stored in memory to 100. We simulated sequence evolution using ecat (Rice 1997) and seqgen (Rambaut & Grassly 1997) under Jukes-Cantor evolution. We measure accuracy using both FP and FN rates, and we report parsimony scores on computed trees.

The model trees we use in this section are a 35 taxon subtree based upon the year-long parsimony analysis of the 476 taxon *rbc*L dataset given in (Rice, Donoghue, & Olmstead 1997) and a 135 taxon tree based on the "African Eve" dataset (Maddison, Ruvolo, & Swofford 1992). For our studies, we scaled up rates of evolution on these trees to provide trees of increasing diameter with large numbers of taxa to yield an interesting test case for MP heuristics.

Recall that we use the term DCM1 method to refer to the DCM method for reconstructing T_q using

the maxclique decomposition and the term DCM2 as the DCM method for reconstructing T_q using the dac-decomposition. For purposes of this paper, we will always assume that heuristic search Maximum Parsimony (HS) is used to solve the DCM subproblems and that we use the first q for which the threshold graph is connected. If we follow DCM1 or DCM2 with the IAS heuristic for OTR (optimal tree refinement), we will indicate this by referring to the methods as DCM1+OTR and DCM2+OTR.

Experiment 1

In this experiment we explored the effects of how we select q from the set of possible values.

In Figure 3, we show the result of a DCM1 analysis (no OTR heuristic was applied) of one dataset of 35 taxa, using heuristics for MP to reconstruct trees on each subproblem. For each threshold q large enough to create a connected threshold graph, we computed the tree T_q, compared its false positive and false negative rates, and the size of the largest subproblem analyzed using MP in constructing T_q. We see that as q increases, both error rates decline, but the maximum dataset size analyzed by MP increases. Thus, there is a tradeoff between accuracy and efficiency. For optimal accuracy, the best approach would be to take the largest threshold size that can be handled, while for optimal running time, the best approach would be to take the first threshold that makes a connected graph. Note also that for thresholds q above 1.3, i.e. for the last half of the range, the trees T_q have no false positives, and hence are *contractions* of the model tree. For such selections of threshold, an optimal refinement of T_q will recover the true tree for this dataset, or will produce a tree with an even lower parsimony scores.

This experimental observation is why we have elected to follow the reconstruction of T_q with a phase of refinement. Since the number of false negatives is small for most of the experimental range, the tree is almost fully resolved, and hence it will even be possible to calculate the optimal refinement exactly. However, for larger trees, we may find larger false negative rates for small q, and hence have a more computationally intensive OTR problem, if we wish to solve it exactly. Consequently, we use heuristics for the refinement phase on highly unresolved trees.

This 35-taxon experiment was performed on a biologically-motivated model tree. It suggested that on this tree, the use of DCM1 (maxclique-decomposition) would reduce the dataset size; note that for q=1.3, the subproblems are at most half the size of the full dataset. However, this may not be the case with real molecular sequence datasets. Will both DCM1 and DCM2 (dac-decomposition) reduce the dataset size with real biological data? If so, we could possibly expect to get reductions in running time; if not, then this approach is unlikely to work.

In particular, DCM has a limitation which could possibly affect its usefulness for analyzing real data: it will

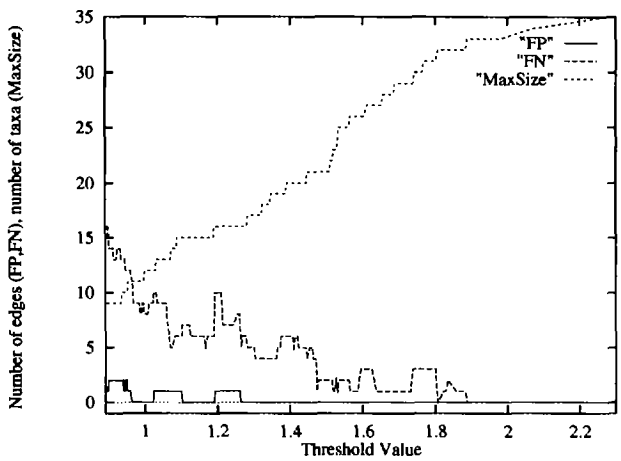

Figure 3: Here we depict the result of an experiment performed on 35 DNA sequences of length 2000 generated on a 35 taxon Jukes-Cantor model tree using a moderate rate of evolution. For all different threshold values that give rise to a connected threshold graph in the DCM algorithm, we computed the DCM1 tree (no OTR heuristic was applied). We plot the number of false positives (FP), false negatives (FN) and Maximum Problem size (MaxSize) produced by DCM1.

not improve the accuracy or running time of methods if the tree that generated the data is an "ultrametric". An ultrametric is a tree in which the distance from the root to each leaf is the same, and this occurs when evolutionary distances (measured by the number of mutations) are proportional to time. In other words, ultrametrics occur when the evolutionary process has the *strong molecular clock*. When the data are generated by a molecular clock, then the only threshold q that will permit a tree to be constructed, rather than a forest, is the maximum distance in the input matrix; in this case, there is no reduction to the dataset size at all.

Earlier work showed convincingly that not all datasets had strong molecular clocks, and that in fact rates of evolution could vary quite dramatically (Vawter & Brown 1986). In fact, because the molecular clock varies more among more distantly-related taxa, datasets comprised of distantly-related sequences may be especially well-suited to DCM. Thus, a critical factor affecting the usability of DCM on the real datasets that we chose was whether the decomposition DCM obtained would produce significantly smaller subproblems.

We investigated this concern by examining several real datasets of significant interest to biologists. We should note that none of these datasets has been noted in the literature as a violator of molecular clocks, despite much effort toward solving their phylogenies. We will show here that despite their lack of obvious deviation from a molecular clock, these datasets can be decomposed under DCM into subproblems that are sig-

(1) Data set	(2) Taxa	(3) DCM1	(4)	(5) DCM2	(6)
		count	size	count	size
Greenplant	221	61	52%	3	74%
Olfactory	252	100	44%	9	68%
*rbc*L	436	296	17%	8	47%

Table 1: Problem size reduction obtained using DCM for the three data sets (1) described in the text. In column (2) we indicate the number of taxa in the original data set. In columns (3), (4) and (5), (6) we list the number of problems and the reduction obtained by each of the techniques, the latter scored as a percentage the largest subproblem size has of the original data set size.

nificantly smaller than the original problem, especially if the maxclique-decomposition (DCM1) is used, but even when we use the dac-decomposition (DCM2).

We examined problem size reduction in the *rbc*L436 dataset, the greenplant221 dataset and the olfactory252 dataset that we described earlier. Table 1 shows what we found.

As we see from Table1, DCM2 (dac-decomposition) produces larger subproblems than DCM1 (maxclique-decomposition); also, the subproblems contain more divergence, but there is generally a *much* smaller number of dac subproblems than maxclique subproblems. Nevertheless, both dac- and maxclique-decompositions do produce reductions, sometimes large ones, in the dataset size, and should therefore produce increases in the efficiency with which we can solve MP and other NP-hard optimization problems for these datasets.

Experiment 2

In this experiment we explored the improvement in accuracy, both with respect to parsimony score and with respect to topology estimation, when using the two tree reconstruction methods DCM1 (maxclique-decomposition) and DCM2 (dac-decomposition), applied to the smallest q which creates a connected threshold graph, with the IAS heuristic for optimal tree refinement.

We show the effect of the IAS heuristic upon the accuracy of DCM1 and DCM2 in Table 2 and Table 3, respectively for the 135-taxon "African Eve" dataset.

Note that any method for refining a tree will only decrease (or leave unchanged) the false negative rate, and increase (or leave unchanged) the false positive rate, because it only *adds* edges to a tree. We see that in many, thought not all, cases, IAS was optimal with respect to improving the topology estimation, and often reduced the parsimony score by a great amount (see, for example, the results for the higher mutation rate on this tree).

(1) Scale factor	(2) Seq. length	(3) DCM1	(4)	(5) DCM1+OTR	(6)
		Score	FP/FN	Score	FP/FN
0.1	250	10075	6/45	8763	15/15
0.1	500	19086	9/35	17601	14/14
0.1	1000	37312	2/23	34958	5/6
0.1	1500	53990	4/16	52366	4/4
0.1	2000	72705	2/18	69624	2/2
0.2	250	19009	4/98	13079	63/63
0.2	500	35842	12/95	26230	48/51
0.2	1000	67045	6/77	51996	37/37
0.2	1500	93090	11/70	78952	35/35
0.2	2000	131658	5/74	104410	36/36

Table 2: The effects of the IAS heuristic for optimal tree refinement (OTR) upon the tree computed using DCM1 (maxclique-decomposition). We generated sequences on a 135 taxon model tree under the Jukes-Cantor model of evolution for different values (1) of the maximal estimated number of substitutions per site on an edge and (2) sequence lengths. We report (3) the parsimony score obtained using DCM1, (4) the FP and FN rates obtained using DCM1, (5) the parsimony score obtained by refining the computed tree using the IAS heuristic, and (6) the FP and FN rates of that tree.

(1) Scale factor	(2) Seq. length	(3) DCM2	(4)	(5) DCM2+OTR	(6)
		Score	FP/FN	Score	FP/FN
0.1	250	10075	6/45	8763	15/15
0.1	500	17665	6/12	17412	6/6
0.1	1000	35213	2/8	34771	2/2
0.1	1500	52524	3/6	52158	3/3
0.1	2000	72705	2/18	69624	2/2
0.2	250	19009	4/98	13079	63/63
0.2	500	27322	17/39	25908	33/33
0.2	1000	53451	18/33	51669	29/29
0.2	1500	78452	25/28	78211	28/28
0.2	2000	131658	5/74	104410	36/36

Table 3: The effects of the IAS heuristic for optimal tree refinement (OTR) upon the tree computed using DCM2 (dac-decomposition). We generated sequences on a 135 taxon model tree under the Jukes-Cantor model of evolution for different values (1) of the maximal estimated number of substitutions per site on an edge and (2) sequence lengths. We report (3) the parsimony score obtained using DCM2, (4) the FP and FN rates obtained using DCM2, (5) the parsimony score obtained by refining the computed tree using the IAS heuristic, and (6) the FP and FN rates of that tree.

(1) Scale factor	(2) Seq. length	(3) DCM2+OTR Score	(4) DCM2+OTR FN/FP	(5) DCM1+OTR Score	(6) DCM1+OTR FN/FP
0.1	250	8763	15/15	8763	15/15
0.1	500	**17412**	**6/6**	17601	14/14
0.1	1000	**34771**	**2/2**	34958	5/6
0.1	1500	**52158**	**3/3**	52366	4/4
0.1	2000	69624	2/2	69624	2/2
0.2	250	13079	63/63	13079	63/63
0.2	500	**25908**	**33/33**	26230	48/51
0.2	1000	**51669**	**29/29**	51996	37/37
0.2	1500	**78211**	**28/28**	78952	35/35
0.2	2000	104410	36/36	104410	36/36

Table 4: We generated sequences on a 135 taxon model tree under the Jukes-Cantor model of evolution for different (1) values of the scale factor (see (Rambaut & Grassly 1997)) and 2) sequence lengths. We report (3) the parsimony score and (4) the number of false positives/negatives for the the tree computed by DCM2 (dac-decomposition) followed by the IAS optimal tree refinement technique, and (5) the parsimony score and (6) the number of false positives/negatives for the the tree computed by DCM1 (maxclique-decomposition) followed by IAS.

(1) Scale factor	(2) Seq. length	(3) Score Model tree	(4) Score DCM2 +OTR	(5) Score HS
0.1	250	8685	**8763**	8911
0.1	500	17433	**17412**	17641
0.1	1000	34776	**34771**	35414
0.1	1500	52162	**52158**	53299
0.1	2000	69474	**69624**	71414
0.2	250	12974	13079	**13046**
0.2	500	25889	**25908**	25991
0.2	1000	51720	**51669**	51907
0.2	1500	78193	**78211**	78898
0.2	2000	103942	**104410**	104859

Table 5: Comparison of DCM2 (dac-decomposition) followed by OTR, to straight heuristic search Maximum Parsimony. We generated sequences on a 135 taxon model tree under the Jukes-Cantor model of evolution for different values (1) of the maximal estimated number of substitutions per site on an edge and (2) sequence lengths. we report (3) the parsimony score for the given model tree and data, (4) the parsimony score obtained using DCM2+OTR, and finally (5), the score obtained by applying HS directly to the dataset in the same amount of time.

Experiment 3

In this experiment, we made explicit comparisons between the two described versions of DCM:

- Maxclique-decomposition (DCM1) followed by the IAS heuristic for optimal tree refinement.
- Dac-decomposition (DCM2) followed by the IAS heuristic for optimal tree refinement

Our model tree was based on an MP reconstruction of the 135 taxon "African Eve" dataset. In Table 4, we report the comparison between DCM1 and DCM2, with respect to topological accuracy, as well as with respect to the obtained parsimony score. In both cases, we follow the construction of the tree with the IAS heuristic for optimal tree refinement (OTR). The parsimony score (Length) of each of the trees is given, and the False Negative (FN) and False Positive (FP) rates of each of the computed trees is also given.

In *every case* where the results differed, DCM2 followed by the IAS tree refinement technique is superior to DCM1 followed by IAS, with respect to optimizing either the topology estimation or the parsimony score (the better result of the two methods is put in **boldface**, to make the comparison easier to see).

Experiment 4

In the next experiments, we compared DCM2+OTR against the use of the standard PAUP MP heuristic to see if we obtained an advantage over standard techniques; see Table 5. For a given scale factor and sequence length, we generated a set of sequences at the leaves of the model tree under the *i.i.d.* Jukes-Cantor model of evolution using seqgen (Rambaut & Grassly 1997). On a multi-processor machine, we then ran both DCM-HS methods (either DCM1 or DCM2, using HS as base method) and straight Maximum Parsimony heuristic search (HS) in parallel. (Note that DCM makes calls to the same heuristic MP program that is used for the "straight MP" search; we also used the same parameter settings for the heuristic MP search, to ensure that the methods were fairly compared.) Once DCM using heuristic MP completed, we then applied the IAS heuristic for OTR (optimal tree refinement) to the output of the DCM program. If the resulting tree was still not fully resolved, we then ran the IAS heuristic for a second time. Five minutes after this completed, we terminated the straight heuristic MP search. We then compared the resultant trees on the basis of the parsimony score, and examined the trees obtained using one of the two DCM variants for their topological accuracy.

The aim of this experimental setup was to give DCM followed by OTR and HS approximately the same amount of CPU time. (It should be noted, however, that our current implementation of DCM and OTR makes a number of external calls to PAUP and thus is burdened with additional overhead.)

In nearly every case, DCM2+OTR outperformed standard use of the PAUP MP heuristic on these model trees (we put in **bold** the better of the two parsimony scores, to make the comparison easier). This result is a highly significant one to biologists, as it promises to provide a more rapid way to compute trees with low parsimony scores than heuristic MP as implemented in

PAUP, which is the gold standard of the systematics community. Because of the flexibility of DCM2 and of subsequent OTR, biologists can use system-specific knowledge to adapt DCM2 to particular phylogenetic problems. And because the metric used to judge trees, the parsimony score, is external to the method used, a biologist need not choose a specific flavor of DCM, but can experiment with conditions of analysis to reach lower parsimony scores.

Observations

Several observations can be made.

1. DCM2 (dac-decomposition) almost always outperformed DCM1 (maxclique-decomposition), with respect to the topological accuracy and the parsimony score, with both methods using heuristic search MP to solve their subproblems. However, both techniques produced incompletely resolved trees (with the loss of resolution greater for the tree reconstructed using maxclique).
2. Accuracy, with respect to both parsimony score and topology, was greatly improved by using the IAS heuristic for finding the best refinement of the tree with respect to Maximum Parsimony.
3. DCM2+OTR almost always outperformed straight Parsimony on our trees with respect to the parsimony score. DCM1+OTR frequently, but not always, outperformed straight Parsimony.

We conclude that for at least some large trees, DCM2+OTR obtains improved parsimony scores and better topological estimates of the true tree, and that these improvements can be great.

Conclusions

We have provided evidence, both statistical, experimental, and based upon real-data analyses, that even the best polynomial time distance methods are potentially misleading when used in phylogenetic analysis among distant taxa. Thus, for large scale tree reconstruction efforts, it is better to seek solutions to NP-hard optimization problems such as Maximum Likelihood or Maximum Parsimony than to use distance methods which may lead to an incorrect topology, albeit rapidly. We presented a new technique (a variant on the Disk-Covering Method, or DCM), and we showed experimentally that it is be useful for obtaining better solutions to Maximum Parsimony than can currently be obtained using standard heuristics for Maximum Parsimony.

The general DCM structure has been shown to be a flexible and potentially powerful tool, and while it remains to be seen whether the advantages we see on these model trees and the three real datasets tested will hold for real data in general, the flexibility of the DCM methods should allow us to engineer DCM so as to obtain substantially improved performance.

References

Baldauf, S., and Palmer, J. 1993. Animals and fungi are each other's closest relatives: congruent evidence from multiple proteins. *Proc. Natn. Acad. Sci. USA* 11558–11562.

Bodlaender, H.; Fellows, M.; and Warnow, T. 1992. Two strikes against perfect phylogeny. In *Lecture Notes in Computer Science, 623.* Springer-Verlag. 273–283. Proceedings, International Colloquium on Automata, Languages and Programming.

Bonet, M.; Steel, M.; Warnow, T.; and Yooseph, S. 1998. Better methods for solving parsimony and compatibility. Proceedings "RECOMB'98".

Bruno, W.; Socci, N.; and Halpern, A. 1998. Weighted Neighbor Joining: A fast approximation to maximum-likelihood phylogeny reconstruction. Submitted to Mol. Bio. Evol.

Buneman, P. 1974. A characterization of rigid circuit graphs. *Discrete Mathematics* 9:205–212.

Burns, K. 1997. Molecular systematics of tanagers (thraupinae): evolution and biogeography of a diverse radiation of neotropical birds. *Mol. Phylogenet. Evol.* 8(3):334–348.

Chase, M. W.; Soltis, D. E.; Olmstead, R. G.; Morgan, D.; Les, D. H.; Mishler, B. D.; Duvall, M. R.; Price, R. A.; Hills, H. G.; Qiu, Y.-L.; Kron, K. A.; Rettig, J. H.; Conti, E.; Palmer, J. D.; Manhart, J. R.; Sytsma, K. J.; Michaels, H. J.; Kress, W. J.; Karol, K. G.; Clark, W. D.; Hedrn, M.; Gaut, B. S.; Jansen, R. K.; Kim, K.-J.; Wimpee, C. F.; Smith, J. F.; Furnier, G. R.; Strauss, S. H.; Xiang, Q.-Y.; Plunkett, G. M.; Soltis, P. M.; Swensen, S. M.; Williams, S. E.; Gadek, P. A.; Quinn, C. J.; Eguiarte, L. E.; Golenberg, E.; Jr, G. H. L.; Graham, S. W.; Barrett, S. C. H.; Dayanandan, S.; and Albert, V. A. 1993. Phylogenetics of seed plants: An analysis of nucleotide sequences from the plastid gene rbcl. *Annals of the Missouri Botanical Garden* 80:528–580.

Crandall, K.; Kelsey, C.; Imamichi, H.; Lame, H.; and Salzman, N. 1999. Parallel evolution of drug resistance in HIV: failure of nonsynonymous/ synonymous substitution rate ratio to detect selection. *Mol. Biol. Evol.* 16:372–382.

Day, W. 1995. Optimal algorithms for comparing trees with labelled leaves. *Journal of Classification* 2:7–28.

Embley, T., and Hirt, R. 1999. Early branching eukaryotes? *Current Opinion Genet. Devel.* 8:624–629.

Erdős, P. L.; Steel, M. A.; Székely, L. A.; and Warnow, T. 1999. A few logs suffice to build (almost) all trees I. *Random Structures and Algorithms* 14(2):153–184.

Farris, J. 1983. The logical basis of phylogenetic systematics. In Platnick, N., and Fun, V. A., eds., *Advances in Cladistics.* Columbia Univ. Press. 7–36.

Farris, J. 1986. On the boundaries of phylogenetic systematics. *Cladistics* 2:14–27.

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

Freitag, J.; Ludwig, G.; Andreini, I.; Rossler, P.; and Breer, H. 1998. Olfactory receptors in aquatic and terrestrial vertebrates. *J. Comp. Physiol.* 183(5):635–650.

Gascuel, O. 1997. BIONJ: An improved version of the NJ algorithm based on a simple model of sequence data. *Mol. Biol. Evol.* 14:685–695.

Golumbic, M. 1980. *Algorithmic Graph Theory and Perfect Graphs.* Academic Press Inc.

Gu, X., and Nei, M. 1999. Locus specificity of polymorphic alleles and evolution by a birth-and-death process in mammalian MHC genes. *Mol. Biol. Evol.* 147–156.

Holmes, E.; Worobey, M.; and Rambaut, A. 1999. Phylogenetic evidence for recombination in dengue virus. *Mol. Biol. Evol* 16:405–409.

Huson, D.; Nettles, S.; Rice, K.; and Warnow, T. 1998. Hybrid tree reconstruction methods. In *Proceedings of the "Workshop on Algorithm Engineering", Saarbrücken*, 172–192.

Huson, D.; Nettles, S.; and Warnow, T. 1999. Obtaining highly accurate topology estimates of evolutionary trees from very short sequences. To appear in: "RECOMB'99".

Maddison, D. R.; Ruvolo, M.; and Swofford, D. L. 1992. Geographic origins of human mitochondrial DNA: phylogenetic evidence from control region sequences. *Systematic Zoology* 41:111–124.

Matsuika, Y., and Tsunewaki, K. 1999. Evolutionary dynamics of Ty1-copia group retrotransposims in grass shown by reverse transcriptase domain analysis. *Mol. Biol. Evol.* 16:208–217.

Modi, W., and Yoshimura, T. 1999. Isolation of novel GRO genes and a phylogenetic analysis of the CXC chemokine subfamily in mammals. *Mol. Biol. Evol.* 16:180–193.

Page, R.; Lee, P.; Becher, S.; Griffiths, R.; and Clayton, D. 1998. A different tempo of mitochondrial DNA evolution in birds and their parasitic lice. *Mol. Phylogenet. Evol.* 9(2):276–293.

Paladin, F.; Monzon, O.; Tsuchie, H.; Aplasca, M.; Learn, G.; and Kurimura, T. 1998. Genetic subtypes of HIV-1 in the Philippines. *AIDS* 12(3):291–300.

Rambaut, A., and Grassly, N. 1997. An application for the monte carlo simulation of DNA sequence evolution along phylogenetic trees. *Comput. Appl. Biosci.* 13(3):235–238.

Rice, K., and Warnow, T. 1997. Parsimony is hard to beat! In Jiang, T., and Lee, D., eds., *Lecture Notes in Computer Science, 1276.* Springer-Verlag. 124–133. COCOON '97.

Rice, K.; Donoghue, M.; and Olmstead, R. 1997. Analyzing large data sets: *rbc*L 500 revisited. *Syst. Biol.* 46(3):554–563.

Rice, K. 1997. ECAT, an evolution simulator. http://www.cis.upenn.edu/~krice.

Rogers, A.; Sandblom, O.; Doolittle, W.; and Philippe, H. 1999. An evaluation of elongation factor 1-alpha as a phylogenetic marker for eukaryotes. *Mol. Biol. Evol* 16:218–233.

Saitou, N., and Nei, M. 1987. The Neighbor-Joining method: a new method, for reconstructing phylogenetic trees. *Mol. Biol. Evol.* 4:406–425.

Skoufos, E.; Healy, M.; Singer, M.; Nadkarni, P.; Miller, P.; and Shepherd, G. 1999. Olfactory receptor database: a database of the largest eukaryotic gene family. *Nucl. Acids Res.* 27(1):343–345.

Soltis, D.; Soltis, P. S.; Nickrent, D. L.; Johnson, L. A.; Hahn, W. J.; Hoot, S. B.; Sweere, J. A.; Kuzoff, R. K.; Kron, K. A.; Chase, M. W.; Swensen, S. M.; Zimmer, E. A.; Chaw, S.-M.; Gillespie, L. J.; Kress, W. J.; and Sytsma, K. J. 1997. Angiosperm phylogeny inferred from 18s ribosomal DNA sequences. *Annals of the Missouri Botanical Garden* 84:1–49.

Sullivan, S.; Ressler, K.; and Buck, L. 1994. Odorant receptor diversity and patterned gene expression in the mammalian olfactory epithelium. *Prog. Clin. Biol. Res.* 390:75–84.

Swofford, D. L. 1996. PAUP*: Phylogenetic analysis using parsimony (and other methods), version 4.2.

Tempelton, A. 1995. A cladistic analysis of phenotypic associations with haplotypes inferred from restriction endonuclease mapping or DNA sequencing. V. Analysis of case/control sampling designs: Alzheimer's disease and the apoprotein E locus. *Genetics* 140:403–409.

Tuffley, C., and Steel, M. A. 1997. Links between maximum likelihood and maximum parsimony under a simple model of site substitution. *Bulletin of Mathematical Biology* 59(3):581–607.

Vawter, L., and Brown, W. 1986. Nuclear and mitochondrial DNA comparisons reveal extreme rate variation in the molecular clock. *Science* 234:194–196.

Vawter, L. 1991. *Evolution of the Blattodea and of the small subunit ribosomal RNA gene.* Ph.D. Dissertation, Univ. of Michigan, Ann Arbor MI USA.

Watson, E.; Forster, P.; Richards, M.; and Bandelt, H.-J. 1997. Mitochondrial footprints of human expansions in africa. *Am. J. Hum. Genet.* 61(3):691–704.