Some combinatorial optimization problems in phylogenetics

Tandy Warnow
Department of Computer and Information Science
University of Pennsylvania

November 15, 1999

Abstract

The estimation of evolutionary history is a major intellectual project in biology, and yet is one of the hardest problems for a variety of mathematical, statistical, computational, and scientific reasons. In this paper we survey the issues and some of the new developments in the area, and outline the major research problems which still remain.

1 Introduction

Inferring phylogenetic (i.e. evolutionary) trees is a fundamental problem in biology with important applications in biomedical sciences, such as drug design, and protein structure and function prediction. Optimization problems related to evolutionary tree reconstruction are usually NP-hard or conjectured to be NP-hard, so that in practice heuristics are used to reconstruct trees from data sets of more than about 20 taxa (see [34] for an introduction to NP-hardness, and its consequences for algorithm design). Thus, for example, despite years of analysis, we still do not know whether we have found the "most parsimonious tree" for the *African Eve* data [104, 103]. Indeed, it seems that the problem of solving these optimization problems on large divergent data sets may elude our grasp. However, in the last five to ten years there has been increased interest in the optimization problems associated with phylogenetic reconstruction, and there have been new algorithms for evolutionary tree reconstruction developed as a result. Some of these may be useful for solving these hard phylogenetic analysis problems.

The scientific objective of the physical science of phylogenetic studies is not to solve a given optimization problem, but rather to recover the order of speciation or gene duplication events represented by the topology of the true evolutionary tree. (Locating the root of the evolutionary tree is a scientifically difficult task, so that a method is considered to have been successful if it recovers the topology of the *unrooted* tree.) This means that good or poor performance with respect to optimization problems is only important to the degree that it guarantees good or poor performance with respect to topology estimation. Consequently, experimental and analytical studies within systematic biology study the accuracy of the topology estimation of different methods, rather than the accuracy of methods with respect to associated optimization problems.

In this paper we outline some of the new results in the area, and describe the particular issues that confront algorithms designers when working in phylogenetic reconstruction.

1.1 Trees, data, and methods

An evolutionary tree (also called a *phylogenetic tree*) models the evolution of a set of taxa (species, biomolecular sequences, languages, etc) from a common origin. Thus, an evolutionary tree is rooted at the most recent common ancestor of the taxa, and the internal nodes of the tree are each labelled by a hypothesized or known ancestor. The common practice today is to use biomolecular sequences as representatives of the species set, so that the leaves of the tree are labelled by biomolecular (DNA, RNA, or amino acid) sequences. Morphological features are also used to assist in the reconstruction of phylogenetic trees. Both morphological features and aligned biomolecular sequences define *qualitative characters*, which means that they induce a partition of the species set into distinct *character states*. Thus, for example, the morphological feature *vertebrate-invertebrate* defines a binary (two-state) character. When using biomolecular sequences, each *site* (i.e. position) within the multiple alignment defines a character, so that the sequences having the same nucleotide (or amino-acid) at that site exhibit the same state of that character.

Thus, any given set of species set S can be represented by the values each species in S attains for each of the characters in a set C of characters; hence, we can represent the input to a phylogenetic reconstruction problem by a $|S| \times |C|$ matrix such that the ij^{th} entry is the state of the i^{th} species for the j^{th} character. A *phylogenetic tree* T is a tree whose leaves are labelled by the species in the set, and are numbered by $1, 2, \ldots, n$. The objective then of a phylogenetic reconstruction algorithm is to find a tree which best fits the data.

Our discussion of tree reconstruction is not concerned with the location of the root of the tree, because the location of the root is difficult to achieve with any degree of accuracy. However, rooted trees are reconstructed by systematic biologists, and the technique generally employed is to use an *outgroup*, which is a taxon which is clearly less related to the rest of the group than any two members in the group are to each other. Once the best unrooted tree containing the outgroup is constructed, the unrooted tree can be "rooted" on the edge separating the outgroup from the rest of the taxa. The problem with using "outgroups" is that what appears to be an outgroup may not in fact be an outgroup (if all "outgroup?" decisions were easy, then trees would be easy to construct using methods in [58]), and that if the taxon is definitely an outgroup, it may be difficult to locate the edge to which it should attach. In either case, obtaining accurate rooted versions of trees just increases the probability of error, since it adds another aspect of the tree which can be incorrectly analyzed.

Consequently, our objective in tree reconstruction is to obtain an accurate recovery of the topology of the unrooted tree, and this is in general accomplished through the use of related optimization criteria. Some criteria are based upon the sequence data, while others are based upon distances computed between sequences in the data, but unfortunately almost all of the resultant optimization problems have been shown to be NP-hard (and some even NP-hard to solve approximately!) [23, 65, 93]. Of the various sequence-based criteria used to evaluate trees, *parsimony, compatibility* and *maximum likelihood* are the most popular. Parsimony and compatibility are both NP-hard problems, so that "solutions" are generally obtained using heuristics (mostly hill-climbing), although exact algorithms based upon branch-and-bound approaches are used for small enough data sets (generally speaking of up to about 17 taxa). Solutions for the "maximum likelihood" tree are obtained through similar hill-climbing searches, but evaluating each fixed leaf-labelled topology is more computationally expensive (not even known to be solvable in polynomial time, for example). Consequently, maximum likelihood has not been used as frequently as parsimony for tree reconstruction.

A special case of the tree reconstruction problem exists called the "perfect phylogeny" problem, which has a surprisingly nice graph-theoretic equivalent. Although the perfect phylogeny problem is also NP-hard, every fixed parameter version of the problem can be solved in polynomial time. We present these algorithms in Section 3.

Distance-based approaches are also popular, and have a solid statistical foundation. While optimization-based approaches are desirable, almost all optimization problems relevant to distance-based reconstruction are again NP-hard (see [93, 65, 23]). For this reason, heuristic approaches, typically based upon "agglomerative clustering" are used to reconstruct trees. Some of these heuristic methods (notably neighbor-joining) are very popular, and have been shown experimentally to have good performance on small data sets [95]. In recent years, there have been advances in obtaining approximation algorithms with guaranteed performance for distance-based optimization problems. These advances and their performance with respect to topology estimation are discussed in Section 4.

1.2 Models of evolution and model-based inference

Many models have been proposed to describe the evolution of biomolecular sequences. Such models depend on the underlying phylogenetic tree and some randomness. Many models assume that the sites are independently and identically distributed (i.i.d.). In the most general stochastic model that we study the sequence sites evolve i.i.d. according to the general Markov model from the root [50]. Since the i.i.d. condition is assumed, it is enough to consider the evolution of a single site in the sequences. Substitutions (point mutations) at a site are generally modeled by a probability distribution π on a set of $r > 1$ character states at the root ρ of the tree (an arbitrary vertex or a subdividing point on an edge), and each edge e has an associated $r \times r$ doubly stochastic transition matrix $M(e)$.

The **Cavender-Felsenstein** model [18, 19, 20] (also called the *Cavender-Farris* model, and henceforth referred to as the "CF model") is the simplest possible Markov model of evolution. Let $\{0, 1\}$ denote the two states. The root is a fixed leaf and the distribution π at the root is uniform. For each edge e of a tree T we have an associated *mutation probability* that lies strictly between 0 and 0.5. Let $p : E(T) \to (0, 0.5)$ denote the associated map. Each site evolves down the tree identically and independently according to a Markov process, so that $p(e)$ denotes the probability that the character state in site i changes at the endpoints of the edge e.

Thus, the CF model is an instance of the general Markov model with

$$M(e) = \begin{pmatrix} 1 - p_e & p_e \\ p_e & 1 - p_e \end{pmatrix}.$$

We now describe a nice formula which is useful for understanding the performance of methods with respect to topology estimation. Given a leaf-labelled tree T and a subset S of the leaves, we denote the subtree of T induced by this set S by $T_{|S}$. If we suppress all nodes of degree two in $T_{|S}$ (by contracting edges incident with such nodes) we obtain the tree we denote by $T^*_{|S}$. Given $T^*_{|S}$, we can define mutation probabilities on the edges of $T^*_{|S}$ so that the probability distribution on the patterns on S is the same as the marginal of the distribution on patterns provided by the original tree T. The mutation probability that we assign to an edge of $T^*_{|S}$ is just the probability p that the endpoints of the associated path in the original tree T are in different states, and p is nicely related to the mutation probabilities $p_1, p_2, ..., p_k$ of edges of the path of the original tree:

$$p = \frac{1}{2}\left(1 - \prod_{i=1}^{k}(1 - 2p_i)\right) . \qquad (1)$$

Formula (1) is well-known and easy to prove by induction.

1.3 Encodings of Trees

Although trees can be represented in many ways, there is a particular representation which is used for various purposes (evolutionary tree reconstruction methods, consensus problems, and performance evaluation) called the *character encoding* which we now describe.

Given a tree T which is leaf-labelled by $1, 2, \ldots, n$, each edge e in the tree defines a bipartition π_e of the leaves of the tree in the natural way, where i and j are in the part of the bipartition if i and j are in the same component of $T - e$. It is not hard to see that the set $C(T) = \{\pi_e : e \in E(T)\}$ defines the tree T and that in fact T can be reconstructed from $C(T)$ in polynomial time. $C(T)$ is called the *character encoding of* T, or the *set of splits of* T.

1.4 Consistency, Power, and Robustness

The objective in phylogenetic reconstruction is to obtain an accurate estimate of the topology of the evolutionary tree which generated the observed sequences. We measure the performance of a phylogeny estimation method in terms of the *bipartitions* in the tree reconstructed by the method as compared to the bipartitions present in the true model tree; in other words, we compare $C(T)$ to $C(T^*)$, where T is the model tree and T^* is the reconstruction obtained by the method.

When $C(T) = C(T^*)$ then the method obtains exactly the correct topology, while if $C(T) \neq C(T^*)$, then the method has erred in some way. There are two types of errors that a method can make. Bipartitions in $C(T^*) - C(T)$ are inferred by the method but are not present in the model tree. These are the *false positives*, and are called *Type 1* errors. Bipartitions in $C(T) - C(T^*)$ are bipartitions that the method estimates as missing from the model tree but in fact are not missing; these are *false negatives*, and are also referred to as *Type 2 errors.*

Typically evolutionary trees are assumed to be *binary*, since speciation events are almost always bifurcations, and hence produce binary trees. In such cases, a method obtains an entirely accurate reconstruction of the topology if it does not fail to reconstruct any of the edges in the model tree; i.e. if it has no Type 2 errors. By contrast, a method can have no Type 1 errors and yet be completely uninformative (i.e. it may return a star-topology).

For this reason, it is reasonable to study the Type 2 errors that a method makes as a measure of its inaccuracy. Alternatively, we may consider the sum of the two errors (Type 1 + Type 2) as the measure of inaccuracy. The second way of measuring the error rate is the most typical approach used in phylogenetic studies.

Given a measure of inaccuracy (either Type 2 errors, or the sum of Type 1 and Type 2), we may study the conditions under which a method will be entirely accurate, and we may also study the rate at which the inaccuracy goes to 0. These issues are discussed in the phylogenetic literature under several names, of which the three most popular criteria are: **consistency, convergence rate,** and **robustness**. We will describe the usage of these terms as they are used in the systematic biology community, although to some extent each such usage is an abuse of the terminology.

A method is said to be a **consistent** method for inferring trees *under a given model of evolution* if for all trees in that model and for all bounds B on the error, the probability that the method has more than B errors on a random set of sequences of length k goes to 0 as k goes to infinity. Equivalently, the probability that the method makes any errors at all goes to 0, so that the method *converges* to the correct topology with probability 1. We note that this definition says nothing about the *rate* at which the method converges, so that a method may be consistent but it may require sequence lengths that exceed the size of genomes to give an accurate estimate of the topology. (Some consistent methods may even give very poor estimates of the topology on very long sequences!) Therefore, the study of the **convergence rate** (the rate at which the number of errors approaches 0, as a function of the sequence length) is also significant. **Robustness** is concerned with the issue of whether a method can recover the topology of the model tree when the assumptions of the model are violated.

It is very important to note that all these concepts (consistency, convergence rate, and robustness) are defined with respect to a particular model. Thus, some methods will be consistent estimators for one model but not for another, and the key question is whether the conditions under which a method is guaranteed to be consistent are biologically realistic or not. Unfortunately, for the most part we have only negative results with respect to robustness (see [97, 60] for conditions under which methods can be robust to relaxing the *iid* assumption).

2 Character Based Reconstruction

2.1 Parsimony

Parsimony is one of the most popular methods for phylogenetic tree inference, and yet it is a method whose applicability to phylogeny reconstruction is seriously and sometimes violently disputed in the systematic biology literature. In order to define the parsimony method, we begin with the following definitions.

Definition 1 *The* **Hamming distance** *between two sequences x and y of the same length is $|\{i : x_i \neq y_i\}|$ and is denoted $H(x, y)$. The* parsimony length *of a tree in which each node v is labelled by a sequence s^v of length k over Σ is the sum of the Hamming distances of sequences labelling endpoints of edges in the tree, i.e. $\sum_{(a,b)\in E} H(a, b)$. Given a set S of sequences, a* **most parsimonious tree** *for S is a tree leaf-labelled by S and assigned labels for the internal nodes, of minimum parsimony length. Thus, the* **parsimony criterion** *is to find a tree of minimum length.*

The motivation for the parsimony criterion is the observation that if evolution is assumed to operate only through point mutations (for example, substitutions of one nucleotide for another) then the parsimony length of a tree is the minimum possible number of evolutionary events needed to obtain the set of sequences observed at the leaves through point mutations.

Given an arbitrary set of sequences, the parsimony problem is to find a tree of minimum parsimony cost (i.e. a "most parsimonious tree"). Unfortunately, this is an NP-hard problem, even when the sequences are binary (i.e. the alphabet size is two) [75, 33, 81]. One approach that has been taken to obtaining good (hopefully optimal) trees with respect to parsimony is to examine as many different leaf-labelled topologies as possible, evaluating each one in turn for its best possible labelling of the internal nodes, and selecting the best of all the considered trees. Given a fixed leaf-labelled tree, computing the parsimony length can be achieved in polynomial time [108, 107], and this approach is the basis of the heuristics used in practice. In practice, parsimony can be solved exactly for up to about 17 sequences.

Approximating Parsimony Although parsimony is NP-hard, it can be 2-approximated in polynomial time in a very simple way. This is a well-known result in the community.

Given the set S of species defined as vectors in Σ^k, define the weighted complete graph $G(S)$ whose node set is bijectively labelled by the species in S, and where $w(i,j)$ is the Hamming distance between the i^{th} and j^{th} sequences.

Theorem 1 *Let T be a minimum spanning tree on $G(S)$. Then the parsimony length of T is at most twice that of the most parsimonious tree.*

Proof: Consider a most parsimonious tree, T^*, and consider the result of doubling the tree T^* to create an edge-weighted graph G in which every edge in T^* appears twice. This is an Eulerian graph since every node has even degree, and consequently G has an Eulerian tour, γ. Create from γ a smaller tour, γ', which contains only the nodes of $G(S)$ ordered in the way in which they appear in γ. Let the weight of the tour γ be denoted by $w(\gamma)$, and define it to be the sum of the weights of the edges in γ. Similarly define $w(\gamma')$. It is easy to see the following:

$$w(\gamma) = w(G) = 2w(T^*)$$

since G is Eulerian, and that

$$w(\gamma') \leq w(\gamma)$$

since Hamming distances satisfy the triangle inequality. Now note that if we delete any single edge from γ' we create a path P such that

$$w(P) < w(\gamma').$$

Hence the most parsimonious tree whose topology is a path on the input sequences is guaranteed to have a parsimony length which is at most twice that of the most parsimonious tree. Now consider T, a minimum spanning tree for the graph $G(S)$. Since P is also a spanning tree, it follows that

$$w(T) \leq w(P) < w(\gamma') \leq w(\gamma) = 2 * w(T^*).$$

∎

The main contribution of this result is an upper bound on the parsimony length of the most parsimonious tree for a given input. Using this method to try to infer something about the structure of the most parsimonious tree seems doubtful, since the proof itself shows that a ratio of 2 is achievable on a path, and the order of the nodes in the path is probably not particularly useful for inferring the topology of the most parsimonious tree.

2.2 Compatibility

For some types of data, such as morphological features, there are very specific constraints that are implied by the data that are not adequately described by either parsimony or maximum likelihood. For example, the qualitative character *vertebrate-invertebrate* imposes a very precise constraint on the topology of the evolutionary tree, and that is that there should be an edge in the tree whose removal separates the vertebrates from the invertebrates. We generalize this property as follows:

Definition 2 *A character c defined on S is* **compatible** *(also called* **convex***) on an evolutionary tree T for S if it is possible to label all the nodes of T so that for every state i of c, the nodes labelled i in T define a subtree of T (i.e. are* connected*).*

There is a natural optimization problem associated to this property, which is the *maximum compatibility* problem, defined as follows.

Definition 3 *Given a set S of species defined by a set C of characters, the* **maximum compatibility** *problem seeks a tree T on which a maximum number of characters in C are compatible.*

In the context we address, the species are typically defined by sequences of the same length, so that the character set C is simply the set of *sites* in the sequences. Unfortunately, the maximum compatibility problem is, in fact, equivalent to Maximum Independent Set problem, even for binary characters. Hence it is NP-hard even to approximate. See [22, 106] for these and related results.

2.3 Maximum Likelihood estimation

Given a model of evolution (for example, the Cavender-Felsenstein (CF) model [18, 19, 20], the general Markov model [50], or some other such model), the problem of finding the model tree which is most likely to have generated the observed sequences is the *maximum likelihood estimation* problem. Finding the maximum likelihood tree is very computationally expensive, however, even for the simplest models of evolution. For example, in the CF model of evolution, every edge is associated with a mutation probability. Given a leaf-labelled tree, maximizing the likelihood score of that tree requires finding the mutation probabilities on the edges that maximize the probability of observing the sequences at the leaves. The leaf-labelled tree for which the optimal assignment of mutation probabilities results in a maximum probability of observing the sequences at the leaves is the maximum likelihood tree. In practice, computing the optimal mutation probabilities per edge in a fixed leaf-labelled tree is much more computationally expensive than computing the optimal assignment of sequences to the internal nodes (as is required for computing the parsimony score of a fixed tree), so that finding the maximum likelihood tree is generally speaking more difficult than finding the most parsimonious tree.

3 Perfect Phylogeny

Related to both the maximum compatibility and maximum parsimony problems is the *perfect phylogeny* problem, as follows:

Definition 4 *A tree on which all the characters are compatible is called a* **perfect phylogeny**. *Given a set S of species defined by a set C of characters, the* **perfect phylogeny problem** *(also called the* character compatibility problem*) asks if there is a tree T on which all the characters in C are compatible.*

Note that the convexity property is independent of rooting for the tree, and that there is no information about distances in the tree, whether measured in time or in evolutionary change.

3.1 History of the perfect phylogeny problem

A phylogeny which has no back mutations and exhibits no parallel evolution has the property that every character is compatible; i.e. it is a *perfect phylogeny.* It must be evident that such phylogenies are unlikely when analyzing biomolecular sequences, especially when the sequences are DNA or RNA sequences. However, when working with morphological features, such as *presence or absence of a backbone*, the problem makes much more sense.

This objective was championed by LeQuesne in a series of papers [84, 85, 86, 87] and later given its firm mathematical foundation in the literature by a series of papers (see for example [26, 27, 56, 24]). The computational complexity of the problem remained open for many years (though it was assumed to be NP-hard, since it is closely related to other NP-hard problems, maximum parsimony and maximum compatibility). This was finally proved independently in 1991 by Steel [91] and Bodlaender, Fellows and Warnow [71].

Until 1990, the only progress on algorithms for the character compatibility problem was to show that binary character compatibility [88] and compatibility of two characters at a time [77, 79] could be solved in polynomial time. An important theoretical breakthrough came in 1974, when Buneman proved a beautiful result [73] showing that the problem reduced in polynomial time to the graph-theoretic *Triangulating Colored Graphs Problem* (TCG).

3.2 The triangulating colored graphs problem

We begin by defining some basic graph-theoretic terminology.

Definition 5 *A* **graph** $G = (V, E)$ *consists of a vertex set V and an edge set E, where each edge is an unordered pair of distinct vertices. A map $f : V \to Z$ is said to be a* **vertex-coloring** *and if the vertex-coloring satisfies $f(v) \neq f(w)$ for all vertices v, w such that $(v, w) \in E$, then the vertex-coloring is said to be* proper. *A* **path** *in the graph is a sequence of vertices $v_1, v_2, \ldots, v_r$ where $(v_i, v_{i+1}) \in E$ for $i = 1, 2, \ldots, r-1$, and a* cycle *is a path with $v_1 = v_r$ (i.e. it begins and ends in the same place). A graph is said to be* **acyclic** *if it has no cycles. A* **chord** *in a cycle is an edge which joins vertices which are not consecutive in the ordering given by the cycle. A graph G is said to be* **chordal** *or* **triangulated** *if every cycle of size at least four contains a chord. Such graphs are also called* **rigid circuit graphs**.

The **Triangulating Colored Graphs Problem** is as follows

Triangulating Colored Graphs Problem:

Input: A graph $G = (V, E)$, and a vertex-coloring $c : V \to Z$.

Question: Does there exist a graph $G' = (V, E')$ with $E \subset E'$ such that

- G' is chordal, and
- G' is properly colored by c?

In other words, is it possible to add edges between vertices in G, never adding an edge between vertices which have the same color, so that the resultant graph is triangulated? If this is possible, the resultant graph is called a c-**triangulation** of G, and G is said to be c-**triangulated**.

We now describe the relationship between the perfect phylogeny problem and the triangulating colored graphs problem.

Suppose S is a set of species defined by the set C of characters, and suppose that a perfect phylogeny T exists for $I = (S, C)$. We define a graph G_T based upon the perfect phylogeny G_T as follows. The nodes of G_T are the states of the characters in C, so that α_i is a node if $\alpha \in C$ and i is a state of α. The edges of G_T are those (α_i, β_j) for which $\exists v \in V(T)$ such that $\alpha(v) = i$ and $\beta(v) = j$. Thus, we can color the vertices of G_T with $|C|$ colors, by assigning color α to the nodes labelled α_i. Clearly, G_T contains no edge of the form (α_i, α_j), and this is a proper coloring. Furthermore, it can be shown that G_T is triangulated, since otherwise T would contain a cycle. Thus, the perfect phylogeny T defines a triangulated colored graph G_T.

We now describe the **partition intersection graph** derived from the input I (this graph was originally defined and studied by McMorris and Meacham in [89]). The vertices of G_I are also the character states of the different characters, and also given the same coloring. The edges of G_I are defined as follows: $(\alpha_i, \beta_j) \in E(G_I)$ if and only if $\exists s \in S$

such that $\alpha(s) = i$ and $\beta(s) = j$. It is clear that G_I is properly colored, and that if there is a perfect phylogeny T for the input I, then G_T is a supergraph of G_I on the same node set. In other words, *G_T is a c-triangulation of G_I*.

We summarize this discussion as follows.

Lemma 1 *Let $I = (S, C)$ be an input to the perfect phylogeny problem, and assume that a perfect phylogeny T exists for I. Then the partition intersection graph G_I has a c-triangulation.*

The converse of this is also true, in other words, if G_I has a c-triangulation then $I = (S, C)$ has a perfect phylogeny. The proof of the converse is more complicated, and is based upon the characterization of chordal graphs given by Buneman in 1974 [73], as follows:

Theorem 2 *(Buneman 1974) A graph $G = (V(G), E(G))$ is chordal if and only if there exists a tree $T = (V(T), E(T))$, a collection of subtrees $\mathcal{T}$ of T, and functions $f : V(G) \to \mathcal{T}$ and $g : V(T) \to \{maximal\ cliques\ of\ G\}$ such that*

- *$(a, b) \in E(G)$ if and only if $f(a) \cap f(b) \neq \emptyset$, and*
- *g is a bijection, and $a \in g(v)$ if and only if $v \in f(a)$.*

For example, if $G = K_n$, then G corresponds under this theorem to a tree with a single node, since G contains a single maximal clique. (Note that by "maximal clique" we mean a clique that is not properly contained in any other clique, but this clique need not be of maximum cardinality!)

The following corollary follows from this characterization.

Theorem 3 *An instance I to the character compatibility problem has a perfect phylogeny if and only if G_I can be c-triangulated.*

3.3 Fixed parameter Perfect Phylogeny

There are three natural parameters to the character compatibility problem: **n**, the number of species (or sequences); **k**, the number of characters (or length of the sequences); and **r**, the maximum number of states per character. Because perfect phylogeny is in general NP-complete, the best we can hope for is to find polynomial time algorithms for the various fixed parameter versions of the problem. Fortunately, such algorithms have recently been obtained, and in the next few sections we describe these results.

Algorithms when the number k of characters is fixed The algorithms which handle the case where the number of characters is fixed directly use Buneman's theorem. These algorithms first translate the input I (given as species defined by characters) into the partition intersection graph G_I, and then use graph-theory and Buneman's theorem to determine whether I admits a perfect phylogeny.

Theorem 4 *Let I be an input to the perfect phylogeny problem based upon two characters, and let G_I be the partition intersection graph. Then I has a perfect phylogeny if and only if G_I is acyclic. Hence we can determine whether a perfect phylogeny exists for two characters in $O(|S|)$ time.*

Proof: Computing the partition intersection graph G_I takes $O(|S|)$ time, since each sequence in S defines an edge in G_I. Suppose G_I can be c-triangulated, and let G' be a c-triangulation of G_I. It is easy to prove by induction that a two-colored graph is triangulated if and only if it is acyclic, and hence G' must be acyclic (since G_I is two-colored, and hence G' is also two-colored). Hence, G_I must also be acyclic. Determining acyclicity of G_I takes $O(|V(G_I)|)$ time. Now, if $|V(G_I)| \leq |E(G_I)|$ then G_I is cyclic, and hence we can assume that $|V(G_I)| > |E(G_I)|$. Note then that since G_I is the partition intersection graph for two characters, $|E(G_I)| = |S|$ since each edge in G_I corresponds to a unique element in S. Since computing the graph G_I takes $O(|S|)$ time, and determining if G_I is acyclic takes $O(|S|)$ time, the algorithm takes $O(|S|)$ time. ∎

In the biological literature, the first algorithms presented to determine compatibility of two characters were given in 1975 [77, 79], and proved correct two years later in [78]. By contrast, we have presented an extremely simple algorithm with a very short proof, by applying Buneman's theorem!

Graph-theoretic algorithms for three characters exploiting properties of triangulated graphs were obtained by Kannan and Warnow [83], and then later by Bodlaender and Kloks [72] and Idury and Schaeffer [82]. Each of these algorithms uses linear time, but differ in their space requirements (Kannan and Warnow's algorithm requires quadratic space, and the others have linear space implementations).

There have also been polynomial time algorithms for the case in which the number of characters is bounded. The first such algorithm is by McMorris, Warnow, and Wimer [90]. To obtain their algorithm, McMorris *et al.* first demonstrate a relationship between k-colored graphs which can be c-triangulated and graphs which have treewidth bounded by $k-1$. They then show that determining whether a given k-colored graph can be c-triangulated can be accomplished by modifying the *partial k-tree* recognition algorithm of Arnborg, Corneil, and Proskurowksi in [7]. The McMorris, Warnow, and Wimer algorithm runs in $O(nk^2 + (rk)^{k+1})$ time. Another algorithm for this case is by Agarwala and Fernandez-Baca [3]. This $O(r^{k+1}nk)$ algorithm is combinatorial, and implies an $O((2e/k)^k e^2 k)$ algorithm for triangulating a k-colored graph having e edges.

Algorithms when the number r of states is fixed This fixed parameter version of the perfect phylogeny problem is the most relevant to practice because biomolecular data (and sometimes other biological data as well) typically have a small number of states. For example, in a column of a multiple alignment of DNA or RNA sequences which does not contain any gaps, there are only four states since there are only four nucleotides. Thus, algorithms which are fast when the number of states is bounded are potentially of use.

The first case of this type that was handled was for binary characters. Polynomial time algorithms for binary character compatibility were found by several authors, and there are several linear time (i.e. $O(nk)$) algorithms; see [36, 5] as examples.

Binary character perfect phylogeny is easy for a particular reason that is specific to them. Let T and T' be trees on n leaves that are leaf-labelled by the same underlying set S. Then we can say that T **refines** T' if T' can be obtained from T by **contracting** some edges in T. This definition allows us to define a partial order on the space of trees leaf-labelled by a fixed set S by saying $T \leq T'$ if T' refines T. Given this definition, it is possible to ask whether for each set S of species defined by character set C there is at most one *minimal* perfect phylogeny for S, C. This is in general not true. However, for binary characters, it is true, as the following lemma asserts.

Lemma 2 *If a perfect phylogeny exists for a set of binary characters, then the minimum such perfect phylogeny (under the refinement order) is unique.*

This makes it possible, for example, to infer the minimal perfect phylogeny for binary characters through sequential addition of leaves. The perfect phylogeny problem is more complicated for r-state characters, when $r \geq 3$, since in those conditions it no longer holds that minimal perfect phylogenies are unique.

Polynomial time algorithms for determining compatibility of three-state characters were found in 1990 by Dress and Steel [76] and Kannan and Warnow [64]. These algorithms are combinatorial rather than graph-theoretic, and have complexity $O(nk^2)$ and $O(n^2k)$ respectively, where n is the number of species and k the number of characters. Kannan and Warnow extended the techniques of their algorithm for three-state characters, and derived an $O(n^2k)$ algorithm for four-state character compatibility [64]. Four-state character compatibility applies to inferring perfect phylogenies from DNA or RNA sequences which do not have gaps, since in these cases the alphabet size is four.

In [4], Agarwala & Fernandez-Baca obtained the an algorithm which has running time $O(2^{3r}(nk^3 + k^4))$, so that the perfect phylogeny problem can be solved in polynomial time when the number of states per character is bounded. This result was later improved by Kannan and Warnow, who obtained an $O(2^{2r}nk^2)$ algorithm for the same case [67].

3.4 Application to Historical Linguistics

Although the concept of a perfect phylogeny was first proposed in the biological context, its applicability has been primarily limited to correctly selected morphological characters. With the increased reliance upon biomolecular data, reconstruction of perfect phylogenies is less common today than in the past.

However, a recent collaboration between Donald Ringe, a historical linguist, and Tandy Warnow, suggested that perfect phylogenies might make sense in the context of *natural language* evolution. Here, the characters are typically *meanings*, and the states for a given meaning are the different *cognate classes*. Their analysis of the Indo-European family of languages [109] was based initially upon the perfect phylogeny algorithm in

[67], but their dataset was pruned substantially because of the presence of *polymorphism*. The most frequent example of polymorphism in this context is two or more words for the same meaning, such as *big* and *large*. Ringe, Warnow, and Taylor (another historical linguist at the University of Pennsylvania) studied the conditions under which such polymorphism arises in linguistics, and modelled polymorphism as the confluence of two or more monomorphic characters. They designed an approach to handling polymorphic data based upon *inverting* the confluence process, and hence formulated the problem of determining whether each polymorphic character could be *separated* into the right number (which is determined through the linguistic evidence) of monomorphic characters, so that each monomorphic character was compatible with a single evolutionary tree (which would thus be a perfect phylogeny for the new set of characters). This problem was then shown to be equivalent to the following graph-theoretic question:

l-Triangulating colored graphs: Given a graph G with vertex coloring c, does there exist a supergraph G' of G such that G' is triangulated and the maximum monochromatic clique size in G' is l?

This is an NP-hard problem for which even the fixed-parameter versions are NP-hard. However, Warnow and her colleagues obtained algorithms which ran in polynomial time if both k and l are bounded, and used these in conjunction with the algorithm of [67] to to obtain a second (and probably more accurate) analysis of the Indo-European family. These results are given in [69]. Additional information about the methodology is described in [68, 70].

4 Distance-based methods

In this section we discuss some of the most promising distance-based methods that are used in systematic biology.

4.1 Basic concepts

Given a leaf-labelled tree T with positive edge weights, we can define the *path distance* between leaves i and j to be the sum of the weights of the edges in the path between i and j.

Definition 6 *A distance matrix D is* **additive** *if there exists a tree with positive edge weights such that $D_{ij} = \sum_{e \in P_{ij}} w(e)$, where P_{ij} denotes the path between leaves i and j in the tree, and $w(e)$ is the weight of edge e.*

The following theorem was proved in [55]:

Theorem 5 *Given an additive $n \times n$ positive distance matrix d, there is a unique positive edge-weighted tree in which n nodes in the tree are labelled $s_1, s_2, \ldots, s_n$, so that the path distance between s_i and s_j is equal to d_{ij}. Furthermore, the unique tree consistent with d is reconstructible in $O(n^2)$ time.*

Definition 7 *We will call any symmetric matrix which is zero-diagonal and positive off-diagonal a* distance matrix*. A* **distance method** *d maps $n \times n$ distance matrices to $n \times n$ additive distance matrices.*

4.2 Methods and Problems

There are many distance-based methods and optimization problems related to distance-based reconstruction. Because almost all optimization problems are NP-hard ([93, 23, 65]), almost all methods are based upon simple heuristics. Probably the most popular type of heuristics used for distance-based tree reconstruction are agglomerative clustering techniques (i.e. these methods determine siblinghood of pairs of taxa in some manner, and recurse on smaller sets of leaves). Surprisingly, these simple methods are *consistent* (i.e. accurate if given long enough sequences). In the following sections, we provide some techniques for establishing consistency, and a framework within which the convergence rate of different distance methods can be studied.

We now discuss two natural requirements we may wish to make of a distance method, which we will later prove will ensure that the method is a consistent estimator for inferring binary model trees. These two properties are *combinatorial consistency* and *continuity*. We will say that a method M is **combinatorially consistent** if $M(D) = D$ whenever D is additive. This property is true of just about all distance methods that are in use today, except those that seek to reconstruct ultrametric trees (i.e. rooted trees in which the distance from the root to any leaf is the same). This property is also automatically true of any method which solves or approximates (with a performance guarantee) an optimization problem of the form "*given distance matrix d, find a nearest additive matrix D*", where by "nearest" we permit any metric between distance matrices to be used. Furthermore, if we define a metric on distance matrices, we may naturally define continuity with respect to that metric. For example, the L_∞ metric is defined by $L_\infty(d, d') = max_{ij}|d_{ij} - d'_{ij}|$. A distance method M is then **continuous** at d (with respect to the L_∞ metric) if for all $\epsilon > 0$ there is a $\delta > 0$ such that $L_\infty(d, d') < \delta$ implies that $L_\infty(M(d), M(d')) < \epsilon$.

Definition 8 *We will say that a distance method method is* **reasonable** *if it is both combinatorially consistent and continuous at additive distance matrices corresponding to positively weighted binary trees.*

Almost all methods used to reconstruct trees from distances are reasonable. The importance of being "reasonable" will be shown in Section 4.5, in which we will prove that any method which is reasonable" is guaranteed to be *consistent* for estimating binary trees.

Many of the methods used in practice are based upon *agglomerative clustering*. Agglomerative clustering is a basic technique which constructs a tree by successively deciding which pair of leaves should be siblings, thus reducing the size of the input in each step. The particular technique by which the siblinghood decision is made, and the way in which the distance matrix is then modified, distinguishes the different clustering methods. Some of the most popular methods used in practice, such as the $O(n^2)$ *Neighbor*

Joining method (popularized by Saitou and Nei in [110]) and the $O(n^4)$ *Fitch-Margoliash* method [32] are based upon this technique. All of these methods, except those that reconstruct ultrametric trees, are "reasonable" and hence provably consistent estimators for binary trees.

Ultrametric trees are rooted and edge-weighted so that the distance from the root to every leaf is the same. Consequently, given an additive but not ultrametric distance matrix D, methods which reconstruct ultrametric trees will modify D, sometimes even changing its topology!

The reconstruction of ultrametric trees used to be popular among biologists when the "molecular-clock" hypothesis was accepted. This hypothesis asserts that mutations occur in a more-or-less clocklike fashion, so that differences between sequences should be proportional to the evolutionary time between the two sequences (i.e. to the time back to their most recent common ancestor). This is also expressed by saying that DNA sequences evolve at a constant rate across different lineages. The molecular-clock hypothesis has however been discredited, and there is mounting evidence that different lineages can evolve at unboundedly different rates, and that even mitochondrial DNA does not evolve at anything close to a constant rate. (See [105] for the original disproof of the molecular clock hypothesis, and [111, 112, 115, 114, 116] for other such results.)

Many new distance-based methods have been introduced, such as *BIONJ* [80], *Quartet Puzzling* [117], *the Short Quartet Method* [118], and *Agarwala's 3-approximation* [93] and its variant, the *Double-Pivot* [119]. However, these methods are not yet in use by the systematic biology community, and there has not yet been enough experimental performance analysis of these methods for their advantages and disadvantages on realistic data sets to be understood.

4.3 Statistical basis of distance-based methods

The idea behind distance-based methods is to compute distances between sequences so that these pairwise distances reflect the actual number of point mutations that occurred on the path between the leaves representing the two sequences. If this can be done so that the actual computed distances exactly equal the number of changes on the paths, then these distances are *additive* and hence can be used to reconstruct the evolutionary tree. Furthermore, as we have shown, reconstructing the underlying tree from additive matrices is easy to do in polynomial time.

There are two main hurdles in this basic approach. The first is that the distances must be computed appropriately, so that these distances will be additive. Hamming distances clearly fail this test because of the "multiple-hits" phenomenon where two sequences are identical on a particular site, but that site has changed state on the path between the two sequences. It turns out that computing additive distances from finite length sequences is not really possible to do, but it is nevertheless possible to define distances so that as the sequence length gets longer, the computed distances more closely approximate additive distances.

Corrected distance transformations were invented for this purpose. A corrected distance transformation simultaneously:

- represents the model tree (i.e. leaf-labelled tree with information about the evolutionary process governing each edge) as an edge-weighted tree, and
- defines distances between sequences generated on the tree, so that the following holds: *as the sequence length increases, the matrix of observed distances converges to the additive distance defined by the edge weighted tree.*

Using such corrected distance transformations then ensures that a distance method can be consistent. Corrected distance transformations exist for the CF model and for the general Markov model. We now describe the corrected distance transformation for the CF model, and why it makes distance methods consistent.

Given a CF tree T and sequences of length k generated at the leaves of T, let $H(i,j)$ denote the *Hamming distance* of sequences i and j and $h^{ij} = H(i,j)/k$ denote the *dissimilarity score* of sequences i and j. The *corrected distance* between i and j is denoted by $d_{ij} = -\frac{1}{2}\log(1-2h^{ij})$ and the model probability of change of character state between the sequences i and j is denoted by E^{ij} (i.e. E^{ij} denotes the expected value of h^{ij}). We let $D_{ij} = -\frac{1}{2}\log(1-2E^{ij})$ denote the *theoretical distance* between i and j, computed with E^{ij} instead of h^{ij}. If we assign to any edge e a positive weight $w(e) = -\frac{1}{2}\log(1-2p_e)$, then it follows from Equation (1) above that D_{ij} is exactly the sum of the weights along $P(i,j)$.

What we have shown is that a combinatorially consistent method applied to distances computed for *infinite length sequences* (and corrected appropriately using these corrected distance transformations) will with probability 1 reconstruct the correct topology. However, we never have infinite length sequences, so that we need to discuss whether the method attains the correct topology on some finite length sequences. For this to be true, we will need the continuity property. However, we will only be able to finish the proof after we establish conditions under which two additive matrices can be guaranteed to define the same topology. The next few sections will develop these results.

4.4 Buneman's Four-Point Condition

The following theorem of Buneman [17], called the *Four Point Condition*, provides a characterization of additive distance matrices which is of interest in its own right, and has several consequences for algorithm design.

Theorem 6 (Four Point Condition [17])
A matrix D is additive if and only if for all i,j,k,l (not necessarily distinct), the maximum of $D_{ij}+D_{kl}, D_{ik}+D_{jl}, D_{il}+D_{jk}$ is not unique. The edge weighted tree (with positive weights on internal edges and non-negative weights on leaf edges) representing the additive distance matrix is unique among the trees without vertices of degree two.

Proof: We only prove one direction, because it is easy and also illuminative. Suppose that D is additive so that there is a tree T with positive edge weights on the internal edges and non-negative edge weights on the edges incident with leaves, so that D_{ij} equals the path distance in T between i and j. Now consider a quartet i, j, k, l. These four nodes induce a subtree of T which is either a star or a resolved binary tree. It is easy to see that the four nodes induce a star if and only if the three pairwise sums are identical. In the case where the four nodes induce a binary tree in which i and j are separated from k and l by a path of positive weight, the smallest of the three pairwise sums will be $D_{ij} + D_{kl}$, while the other two pairwise sums will be identical. ▮

The sketch of the proof we have just described actually indicates that $D_{ij} + D_{kl} < D_{ik} + D_{jl} = D_{il} + D_{jk}$ if and only if the topology of the subtree of T induced by i, j, k, l is $ij|kl$ (i.e. there is an edge in T separating i, j from k, l). Consequently, if the distance matrix is additive, then the topology of the model tree can be obtained by simply inferring the topology of every quartet. It is then straightforward to construct the tree topology, since siblinghood of leaves (and subsequently of subtrees) can be easily inferred, and once the tree topology is reconstructed, the edge-weights realizing the distance method can also be obtained by solving linear equations. This is just one of many polynomial time methods for reconstructing the unique positively edge-weighted tree realizing the distance matrix (though this particular method uses much more time than the other methods!).

However, distances calculated and appropriately corrected from finite length sequences generated on a model tree are not actually additive, even though these distances do (with probability 1) converge to the additive distance defining the model tree. Consequently, the real issue is whether we can infer the model tree from distances that are close to but not identical to the additive matrix defining the model tree.

4.5 Topology Invariant Neighborhoods and Consistency

Since distance methods must be applied to nonadditive distance matrices, it is relevant to consider whether a method can return the topology of the model tree even when the distances are not additive. In order to answer this question, we consider the question of when two different additive matrices define the same topology. All of the results in this section are from [118].

Theorem 7 *Two additive distance matrices D and D' define the same topology if and only if for every quartet i, j, k, l, $D_{ij} + D_{kl}$ is the minimum of the three pairwise sums if and only if $D'_{ij} + D'_{kl}$ is the minimum of the corresponding three pairwise sums.*

Proof: First, note that the topology of a tree is defined by the topology the tree induces on every quartet of leaves in the tree. Given this observation, we note that Buneman's four-point condition shows that the topology of any quartet i, j, k, l can be inferred by examining the three pairwise sums, $D_{ij} + D_{kl}, D_{ik} + D_{jl}, D_{il} + D_{jk}$. The minimum of these three pairwise sums is $D_{ij} + D_{kl}$ if and only if the topology on i, j, k, l in the tree is $ij|kl$. Therefore, two additive distance matrices define the same topology if and only if they impose the same ordering on such pairwise sums. ▮

A surprising consequence of this theorem is that there is a positive neighborhood around each additive distance matrix defining a *binary* tree (i.e. all nodes of degree 3), on which *all* additive distance matrices define the same topology.

Theorem 8 *Let D be an additive distance matrix defining an edge-weighted binary tree T, and let x be the weight of the smallest edge in T. Let D' be another additive distance matrix defining a (not necessarily binary) edge-weighted tree T'. If $L_\infty(D, D') = max_{ij}|D_{ij} - D'_{ij}| < x/2$ then the topologies of T and T' are identical.*

Proof: It suffices to note that if $D_{ij} + D_{kl}$ is the minimum of the three pairwise sums, then it is less than the other two sums by $2x$. If $L_\infty(D, D') < x/2$, then $D'_{ij} + D'_{kl}$ is also less than both of $D'_{ik} + D'_{jl}$ and $D'_{il} + D'_{jk}$. Consequently, D and D' define the same topology (although with different edge weights). ▮

An immediate consequence of this theorem is the following:

Theorem 9 *All combinatorially consistent distance methods which are continuous at additive distance matrices defining binary trees are consistent methods for inferring topologies of binary model trees T.*

Proof: The proof is straightforward. Suppose that T is a binary model tree, x is its smallest edge weight, and M is a method which is both combinatorially consistent and continuous at additive matrices for binary trees (i.e. "reasonable"). Since M is continuous, then there is some δ such that if d satisfies $L_\infty(d, D) \le \delta$, then $L_\infty(M(D), M(d)) < x/2$. Since M maps distance matrices to additive distances, $M(d)$ is guaranteed to have the same topology as $M(D)$. Since M is combinatorially consistent, $M(D) = D$. Consequently $M(d)$ is an additive matrix which defines the topology of the model tree T. ▮

It is worth noting that all of the standard distance-based methods (with the exception of those that explicitly seek to reconstruct ultrametric trees) are continuous at binary trees, and combinatorially consistent, and hence are consistent methods for inferring binary evolutionary trees, but little is understood about the convergence rate of these methods.

However, the proof of this theorem establishes a mechanism by which the convergence rate of different methods can be compared: For each method M and for each binary model tree T and the additive matrix D defined by T, there is some positive δ such that given a distance matrix $d_0 \in N(D, \delta) = \{d : L_\infty(d, D) < \delta\}$, $M(d_0)$ is guaranteed to be an additive distance matrix having the same topology as T. It then follows that the larger the maximum δ for which this is true, the easier it is for a method to be guaranteed to obtain an accurate topology. We will return to this study in Section 4.10.

4.6 Ultrametric Tree Reconstruction

If distances computed on the basis of differences between biomolecular sequences are proportional to time since the sequences split off from a common ancestor (the "molecular clock" hypothesis), then a special kind of edge-weighted tree, called an "ultrametric tree", is an appropriate model of evolution.

Definition 9 **Ultrametric** *trees are rooted edge-weighted trees in which the distances from the root to any two leaves are the same.*

However, as we have discussed earlier, the molecular clock hypothesis is now generally discredited, and the reconstruction of ultrametric trees is no longer generally considered relevant to biomolecular evolutionary studies.

However, there are two reasons to discuss ultrametric tree reconstruction: first, there are some very nice algorithms which have been developed for obtaining optimal solutions to problems related to ultrametric tree reconstruction, and second, these algorithms have been shown to be useful in *approximating* the nearest fitting additive tree. We describe the algorithms for ultrametric tree reconstruction in this section, and show in the next section how they can be used to approximate additive trees.

We have already noted that optimization problems in distance-based reconstruction are typically NP-hard. When the desired tree is constrained to be ultrametric, it is however possible that the problem's complexity could become tractable, but in almost all cases, optimization problems for reconstructing ultrametric trees are still NP-hard [65, 93]. One notable exception is the problem of finding the nearest ultrametric distance matrix (i.e. distance matrix fitting an ultrametric tree) to a given distance matrix, with respect to the L_∞-criterion.

The first result of this type is due to Gower and Ross [35], who proved the following:

Theorem 10 *Given distance matrix d, there is a unique ultrametric distance matrix D satisfying*

1. *D is* dominated by *d (i.e. $D[ij] \leq d[ij]$ for all i, j), and*
2. *D dominates all other ultrametric distance matrices which are dominated by d (i.e. if D' is ultrametric and $D'[ij] \leq d[ij]$ for all i, j, then $D'[ij] \leq D[ij]$).*

Their proof was constructive: given d:

- Step 1: weight the complete graph on $1, 2, \ldots, n$ by the matrix d; i.e. $w(i, j) = d_{ij}$.
- Step 2: construct a minimum spanning tree T on K_n
- Step 3: define the ultrametric matrix U by letting $U[ij]$ be the maximum weight on the edge in T between i and j.

This algorithm produces an ultrametric tree which is called the *subdominant* ultrametric of the matrix d.

Surprisingly, the same tree can be constructed in a greedy *agglomerative* fashion, through the "Single-linkage algorithm," which takes as input a distance matrix d and computes a rooted tree whose edges can be weighted appropriately to obtain the subdominant ultrametric of d.

Single-linkage algorithm:

1. Begin with all taxa (leaves) in their own classes, and set the distance between two classes x, y to be $d(x, y)$.
2. While there is more than one class, DO:
   - Choose the classes C_1 and C_2 minimizing the quantity $d(C_1, C_2)$.
   - Join the subtrees for C_1 and C_2 into one rooted subtree, by making their roots children of the same root.
   - Create the class $C = C_1 \cup C_2$, and define $d(C, C') = min(d(C_1, C'), d(C_2, C'))$ for all classes $C' \neq C_1, C_2$.
   - If $C = S$ then return tree, else delete C_1 and C_2.

This algorithm [49] can be easily modified to assign weights to the edges of the tree so as to define an ultrametric tree, and it is easy to see (see [38] for the first such statement of this observation) that the topology this method reconstructs is the same as the topology obtained by Gower and Ross's algorithm for the subdominant ultrametric.

These algorithms were rediscovered in a later work by Farach, Kannan, and Warnow [65] in the context of a more general problem:

Matrix sandwich problem: *Given two distance matrices M_l and M_h which represent lower and upper bounds respectively, determine if there is an ultrametric matrix U satisfying $M_l[ij] \leq U[ij] \leq M_h[ij]$.*

This is a general class of problems since the matrix sought can be less constrained (i.e. we may seek an additive tree only, or an ultrametric satisfying additional constraints). The following then is obvious:

Theorem 11 *Let M_l and M_h be two $n \times n$ distance matrices. The following are equivalent:*

1. *There is an ultrametric matrix U such that $U \in [M_l, M_h]$, and*
2. *It is possible to weight the edges of the tree obtained by the Gower-Ross algorithm applied to distance matrix M_h so as to obtain an ultrametric matrix $U' \in [M_l, M_h]$.*

This theorem implies that a simple algorithm can determine whether there is an ultrametric matrix in an arbitrary sandwich, and the algorithm uses only polynomial time. This formulation has some nice properties, since in general it means that many different optimization problems related to obtaining nearest ultrametric trees can be solved in polynomial time, for various definitions of "nearest."

4.7 Approximation algorithms for nearest trees

Various approximation algorithms for obtaining "nearest" additive trees to a given distance matrix have been developed, all of which are based upon a fundamental observation relating ultrametric trees and additive trees. To explain this relationship we first define what a *centroid metric* is.

Definition 10 *A* **centroid metric** *is an additive metric which can be realized by edge-weighting a star topology (i.e. a tree with exactly one non-leaf node).*

The critical observation relating centroid metrics, additive metrics, and ultrametrics, was first observed by Farris, and communicated to Carroll who published it in 1976 in [120]:

Theorem 12 *Let D be an additive matrix, and let X be a centroid matrix. Then $D+X$ is an ultrametric matrix.*

This suggests a general strategy for reconstructing nearby additive metrics to given distance metrics, which we now describe.

The basic idea: If d is a matrix, D^{opt} is the nearest additive metric to d (under some optimization criterion). Let X be a centroid metric. Then $U^* = D^{opt} + X$ is an ultrametric matrix, and since D^{opt} is near to d, it may be that U^* is close to $d + X$. If we could get from $d + X$ to U^* we could then easily obtain D^{opt} from U^* by subtracting X. Hence, the problem in some sense "reduces" to finding a nearest ultrametric to $d + X$, for some suitably selected centroid metric X.

To summarize, the observation that an additive metric decomposes into the sum of a centroid (which can be arbitrarily selected) and an ultrametric suggests a *general algorithmic strategy*:

General algorithmic strategy for obtaining nearby additive metrics:

1. Given distance matrix d, compute a centroid metric X, and compute distance metric $d' = d + X$.

2. Use some method to find an ultrametric U which is close to d'.

3. Compute the additive metric $D = U - X$, and reconstruct the edge-weighted tree T realizing D. Return T.

This basic approach was possibly first discovered by Blanken *et al.* in 1982 [14], also used by Brossier in 1984 [15], and again used by Agarwala *et al.* in 1996 [93]. The major contribution of the Agarwala *et al.* paper was the observation that if the basic method were implemented by using the Single-Linkage algorithm and the topology obtained were correctly weighted, then the resultant additive matrix would be *guaranteed* to be no more than three times as far from the input matrix than the nearest additive matrix, *with respect to the L_∞ metric.* In other words, Agarwala and her colleagues showed that the basic approach could be implemented to produce a 3-approximation for the nearest tree, with respect to the L_∞-metric.

4.8 Reconstruction based upon combining subtrees

One general technique that can be used to reconstruct an evolutionary tree is to reconstruct all subtrees of a given size, and then combine these subtrees into one tree.

An unrooted leaf-labelled tree can be defined by the topology it induces on the quartets of leaves in the tree. Thus, one approach to reconstructing evolutionary trees is to determine (using some technique) the topology on every quartet of leaves, and then combine these quartets if possible into one tree consistent with the entire set. It is easy to see that if all the quartets are consistent, it is easy to reconstruct the (unique) tree consistent with the constraints in polynomial time. However, quartet topologies are not always consistent, so that each quartet-based method must also specify a means for resolving inconsistencies or else permit the output of *inconsistent* to be returned on those inputs for which some topology is inaccurately reconstructed.

In essence, then, a quartet method takes as input a set Q of topologies on quartets, and determines a tree from this set. One problem with quartet-based approaches is that some quartets are simply harder to estimate than others (see [95] for a study of quartet estimation). Thus, one quartet-based approach is to take the quartets one has confidence in, and use those only to reconstruct the tree. Unfortunately, consistency of a set of quartets with a tree is in general NP-complete [91] (although the case where the set contains topologies for all quartets is solvable in polynomial time). Thus, quartet methods generally use all the possible quartets and specifically identify a heuristic step for handling inconsistencies, but have the flexibility to allow any method whatsoever for reconstructing trees on quartets.

These methods are historically popular though they have been replaced by the faster and possibly more powerful methods (such as neighbor-joining) introduced in recent years. Recently however there have been new quartet-based methods introduced which have very nice properties and interesting performance in experimental studies. Before we discuss the more sophisticated quartet based methods, we begin with the simplest of all possible methods, which we call the *Naive Method*.

4.8.1 The Naive Method

Consider the following quartet based method. For every set i, j, k, l, select the topology $ij|kl$ if and only if $D_{ij} + D_{kl} < min(D_{ik} + D_{jl}, D_{il} + D_{jk})$. If all the quartet topologies can be simultaneously realized in a single tree, then that tree can be reconstructed in polynomial time (simply determine siblinghood of pairs of leaves, and then of subtrees, and hence reconstruct the tree "from the outside-in").

The problem with this approach is that it may happen that one of the $\Omega(n^4)$ quartets may be incorrectly inferred, especially if the tree contains widely separated pairs of taxa, or very short edges. Nevertheless, this method is combinatorially consistent (i.e. it satisfies $M(D) = D$, when D is additive) and continuous at binary trees, and hence it is a consistent method for reconstructing binary evolutionary trees.

Most (but not all) quartet-based methods begin in essentially the same way as the Naive Method, in that they infer the topology of every quartet (using some method, typically a distance-based reconstruction, but sometimes other techniques are used) and then reconstruct the tree from the set of quartets. Since typically some quartets will be incorrectly estimated, most quartet-based methods must provide a mechanism for handling incompatible sets of quartets.

4.8.2 The Buneman Tree

One of the classical quartet-based approaches is the Buneman tree, suggested by Buneman in [17], as follows:

The topology of every quartet is inferred using the same basic approach as the Naive Method. This defines a set of quartet topologies, Q. If all the topologies in Q are simultaneously realizable with one tree, we return that tree. Otherwise, we seek a tree in which "every edge is supported by Q". We now define what this means.

Consider a bipartition of the leaves S into two sets A and B defined by an edge of a tree T. We will say that this bipartition is completely supported by a set Q of quartet topologies if for every $\{a, a'\} \subseteq A$ and $\{b, b'\} \subseteq B$, the topology on a, a', b, b' defined by Q separates a, a' from b, b'.

Buneman's approach [17] was to reconstruct the tree which contained *every* bipartition that was supported by Q. Although there are exponentially many possible bipartitions, the set of bipartitions that are completely supported by a set of topologies for all of the possible quartets is compatible, and hence defines a unique tree (see [121]). Furthermore, reconstructing the unique tree containing all the completely supported bipartitions can be accomplished in polynomial time; an $O(n^5)$ algorithm to reconstruct the **Buneman Tree** has been implemented in the SplitsTree phylogenetic software package (available at `ftp://ftp.uni-bielefeld.de/pub/math/splits`). A faster $O(n^4)$ algorithm has also been obtained by Berry and Gascuel [121].

4.9 Split Decomposition

An interesting quartet-based method was developed by Bandelt and Dress, which has a different objective than other methods. This method is the *split decomposition* method [8, 9] to construct networks of relationships. The basic idea in this method is the observation that observed sequence or distance data may not support a single tree, but rather may support a *set* of trees. Thus, for example, Buneman showed that additive distance matrices D are characterized as having the four-point condition: *for all* i, j, k, l, *the maximum and median of* $D_{ij} + D_{kl}, D_{ik} + D_{jl}, D_{il} + D_{jk}$ *are the same*; consequently the smallest of the three pairwise sums indicates the topology. But what happens if the distance matrix is not additive? Perhaps the three numbers are distinct, and the smallest is so close to the median that it is hard to determine with any confidence the actual topology of the quartet? In that case, the most accurate representation of the topology on the quartet i, j, k, l is that it is *either* of one of two trees, but not the third. Bandelt and Dress showed that it was possible to effectively represent such ambiguities using *"networks"* (which are really just graphs), and they developed software for representing these networks as planar graphs so that identifying sets of parallel lines would yield the different trees that are consistent with the data. This method and its variants has been implemented and constitutes the *SplitsTree* phylogenetic software package at `ftp://ftp.uni-bielefeld.de/pub/math/splits`.

4.9.1 The Short Quartet Method

Another quartet-based method that has been introduced is the *Short Quartet Method*, by Erdös *et al.* [118]. This method reconstructs trees based upon topologies of just a subset of the possible quartets containing the "short quartets", which are defined as follows.

Definition 11 *If e is an edge in a tree T, then deleting the edge e (but not the endpoints of e) creates two rooted subtrees, t_1 and t_2. Let d_i be the distance from the root of t_i to its nearest leaf. Then the* **depth** *of the edge e is defined to be $max(d_1, d_2)$. The depth of T, denoted* **depth(T)***, is the maximum depth of any edge in T. We say that i, j, k, l is a* **short quartet** *of a tree T if i, j, k, l are leaves in T and the maximum path length (counting only the number of edges) between any pair in i, j, k, l is at most $2depth(T) + 3$.*

Erdös *et al.* made the following critical observations, upon which their method rests:

1. The short quartets suffice to define the tree,
2. Given a set of quartets containing the short quartets, it is possible to determine the unique tree consistent with the implied topology constraints in polynomial time, and
3. Although the set of short quartets cannot be known in advance, they can be estimated in a greedy fashion, and this greedy method has good performance probabilistically.

It should be noted that the short quartet method either produces a tree consistent with all the quartet constraints, or it produces a star tree (i.e. the null tree). Thus, the short quartet method can fail entirely to obtain any information about the tree topology. However, the sequence length required for a complete recovery of the topology of the model tree was shown in Erdös *et al.* to grow only *polylogarithmically* for almost all model trees, once the range within which the mutation probabilities must fall is specified. By contrast, they showed that other methods (neighbor-joining, the naive method, the berry-gascuel method, the 3-approximation algorithms for the L_∞-nearest tree by Agarwala *et al.* [93], Cohen and Farach [119], and even a (hypothetical) exact algorithm for the L_∞-nearest tree *might require superpolynomial* length sequences for a completely accurate topology reconstruction, for almost all trees.

In the next section, we show how they established these analyses.

4.10 Convergence rates of different methods

We have determined that it is easy to prove a distance method is consistent for inferring binary evolutionary trees, and yet we have not established anything about the rate at which these methods converge to the correct topology. The purpose of this section is to provide some analytical foundations for such a study.

Using the framework of topology invariant neighborhoods, Erdös *et al.* [118] and Atteson [94] obtained results on the conditions under which different distance methods would be guaranteed to yield accurate topology reconstructions, which we summarize here:

Theorem 13 *(From [118] and [94]) Let D be an additive $n \times n$ distance matrix defining a binary tree T, d be a fixed distance matrix, and let $\delta = L_\infty(d, D)$. Assume that x is the minimum weight of internal edges of T in the edge weighting corresponding to D.*
(i) *A hypothetical exact algorithm for the L_∞-nearest tree is guaranteed to return the topology of T from d if $\delta < x/4$.*
(ii) (a) *The 3-approximation algorithm for the L_∞-nearest tree is guaranteed to return the topology of T from d if $\delta < x/8$.* (b) *For all n there exists at least one d with $\delta = x/6$ for which the method can err.* (c) *If $\delta \geq x/4$, the algorithm can err for every such d.*
(iii) *The Neighbor Joining Method is guaranteed to return the topology of T from d if $\delta < x/2$, and there exists a d for any $\delta = x/2$ for which the method can err.*
(iv) *The Naive Method is guaranteed to return topology of T from d if $\delta < x/2$, and there exists a d for any $\delta = x/2$ for which the method can err.*

The first implication of this is that for each of these methods, and for every ϵ, and for every model tree T, there is a sequence length k such that the probability that the method obtains a *completely* accurate topology on sequences of length k or greater is at least $1 - \epsilon$. Thus, the methods are all *consistent*, and furthermore, the sequence length needed for a *guarantee* of accuracy with high probability can also be calculated directly. For example, neighbor-joining can be guaranteed to be accurate if $\delta < x/2$, and so we can calculate the sequence length needed to get $\delta < x/2$ with probability $1 - \epsilon$, and similarly an exact algorithm for the L_∞-nearest tree is guaranteed to be accurate if $\delta < x/4$ so we can calculate the sequence length needed to get $\delta < x/4$ with probability $1 - \epsilon$. This argument, taken at face value, suggests that both neighbor-joining and the naive method may be accurate on shorter sequences than the 3-approximation algorithm of Agarwala *et al.* or of Cohen and Farach for the L_∞-nearest tree, since both neighbor-joining and the naive method are guaranteed to be accurate if $\delta < x/2$, while 3-approximation algorithms for the L_∞-nearest tree *can fail* if $\delta \geq x/4$. It also suggests that it is easier to obtain conditions for which neighbor-joining is guaranteed to be accurate than to obtain conditions for which the *exact* algorithm for the L_∞-nearest tree can be guaranteed to be accurate. However, all of these statements need to be studied more closely, and most likely experimentally, since these analytical results do not describe *all* the conditions under which a method is guaranteed to be successful, but only *some* of the conditions which guarantee success.

Returning to the question of *convergence rate*, we note:

Theorem 14 *(From [118]) Let T is a Cavender-Farris model tree with edge mutation probabilities in the range $[f, g]$. Then for every $\epsilon > 0$ there is a constant c such that if sequences of length k are generated on this tree, where*

$$k > \frac{c \cdot \log n}{f^2(1-2g)^{2diam(T)}}, \tag{2}$$

then with probability $1-\epsilon$ the result of applying the Agarwala et al. algorithm to corrected distances will be the true tree. The same formula (with the constant changed) exists for the Neighbor-Joining method, the Berry-Gascuel Method (which reconstructs the Buneman tree), and the Naive Method.

Erdös *et al.* also analyzed the convergence rate of the Short Quartet Method, and obtained an analysis which allows a direct comparison to the convergence rate of the other distance methods:

Theorem 15 *(From [118]) Let T be a Cavender-Farris model tree with edge mutation probabilities in the range $[f, g]$. Then for every $\epsilon > 0$ there is a constant c such that if sequences of length k are generated on this tree, where*

$$k > \frac{c \cdot \log n}{f^2(1-2g)^{4depth(T)}}, \tag{3}$$

then with probability $1-\epsilon$ the result of applying the Short Quartet Method to corrected distances will be the true tree.

The comparison between the sequence length requirement of the Short Quartet Method to that of the sequence length requirement of the other methods (as provided by this analysis) thus depends on a comparison between the depth and the diameter.

The diameter of T (which we have denoted by $diam(T)$) is the number of edges in the longest path in T. It is clear that for small trees, $diam(T)$ is small as well, but for large trees $diam(T)$ can be quite large. The diameter of random trees has been analyzed in [123] and [118], showing that $diam(T)$ grows on the order of $\sqrt{n}$ for random trees under the uniform distribution [123], and on the order of $\log n$ for random trees under the Yule-Harding distribution [118]. These findings indicate that the sequence length needed for *guaranteed* accuracy by either the Neighbor-Joining method or *even* by an exact algorithm for the L_∞-nearest tree can grow superpolynomially in the number of taxa in the dataset. On the other hand, the depth of a tree is never more than $\log n$, and in general can be shown to be bounded by $O(\log \log n)$ for random trees in either the uniform distribution or the Yule-Harding distributions [118]. Thus, the sequence length requirement for the Short Quartet Method generally grows *polylogarithmically*, and never more than polynomially, in the number of taxa in the dataset. However, these results must then be evaluated experimentally, since they imply on that good performance is guaranteed in certain conditions, but do not imply bad performance when those conditions do not hold.

5 How hard are these problems, really?

We note the following curious situation. Almost all optimization problems in the domain of phylogenetic tree reconstruction are NP-hard. Some (such as the L_∞-nearest tree [93] and the maximum compatibility problem [106]) have even been shown to be hard to approximate beyond certain constant approximation ratios (see [13] for more of the story on non-approximability results). Heuristics abound, and they are incredibly simple heuristics, mostly composed of hill-climbing strategies which begin with an initial tree and search the tree-space through branch-swapping or other such topological rearrangements of the tree. However, studies of these heuristics based upon sets of sequences of varying lengths randomly generated on model trees have shown that these heuristics obtain accurate reconstructions of the model tree as the sequence length increases, and even that they seem to solve their optimization problems once the sequences get long enough.

What is going on?

The answer seems to be surprisingly simple. Indeed, the optimization problems are NP-hard to solve exactly, but the data are *not* arbitrary. We have made a very strong assumption when we say that the data are generated by these model trees, and this makes a significant difference, as it turns out.

5.1 Simple heuristics work, under certain conditions!

Observation 1 *All optimization criteria which are* consistent *estimators of binary evolutionary trees, and which can be solved exactly on a fixed leaf-labelled topology, can be solved exactly in time which is polynomial in the number of leaves of the tree with arbitrarily high probability, provided that the sequences are generated under the assumed model of evolution, and the sequences are long enough.*

Proof: Fix an optimization problem Π, a model tree T on n leaves, and $\epsilon > 0$. Now let an ordering of the leaves in T be fixed, and for each i, let k_i be the sequence length needed to obtain an accurate topology estimation of the subtree of T induced by the first i sequences with probability at least $1-\epsilon$ to be obtained through an exact solution to the optimization problem. (Note that k_i is finite since the optimization problem is consistent.) Now let $k = max_i k_i$, and note that k is finite. Now let sequences of length k be generated randomly on T. Construct the topology of the best tree for the first four sequences, and then iteratively add sequences to the tree in the order given, each time finding the best of the possible extensions. Since evaluating each fixed tree topology takes only polynomial time, this is a polynomial time method. Since $k > k_i$, this approach makes a mistake on step i with probability bounded by $1-\epsilon$. Hence, with high probability this technique reconstructs the topology of the model tree. Then, solve the optimization problem exactly on the leaf-labelled topology. Since the optimization problem is consistent, the model tree and the optimal tree for the optimization problem are identical in topology, and since the optimization problem can be solved exactly on leaf-labelled topologies, this results in an exact solution to the optimization problem. ∎

A cautionary note: Other heuristics also have such guarantees, and we will discuss these in a minute; however, we caution the reader *not* to assume that this method would obtain an exact solution to the NP-hard optimization problem for all datasets generated on Cavender-Farris trees, nor even for all datasets with high probability! Rather, all this shows is that this *heuristic* will perform well in simulation studies in which the sequence length is increased arbitrarily. In other words, this heuristic will be *consistent* and hence for long enough sequences will actually reconstruct the optimal solution to its NP-hard objective criterion.

Quartet based methods also work An alternate proof would reconstruct all the topologies of all the quartets of leaves in T. Let n sequences evolve on this tree. Reconstruct the topology of every quartet of sequences using the optimization criterion, and then reconstruct the tree (if it exists) which is consistent with all the topologies on quartets. With probability $1-\epsilon$ all the quartets are correctly computed, and hence the entire tree topology can be reconstructed correctly. Since the optimization criterion is consistent, the model tree has the same topology as all trees which optimize the criterion, so that the result of reconstructing the tree is a leaf-labelled tree which can be extended (in a suitable way) to solve the optimization problem. Now solve the optimization problem on the leaf-labelled topology.

5.2 Solving the L_∞-nearest tree

We have shown two very simple heuristics which provide guaranteed good performance for solving NP-hard optimization problems when the sequences are long enough, and provided that the optimization problems are consistent estimators for model tree inference and can be solved exactly on fixed leaf-labelled topologies. Now we show a specific method which uses polynomial time and which "solves" (in the same sense) the NP-hard optimization problem, the L_∞-nearest tree.

Exact solution for the L_∞-nearest tree

- Given d, use the Agarwala *et al.* algorithm to obtain an additive distance matrix D satisfying $L_\infty(d, D) \leq 3 * L_\infty(d, D^{opt})$.
- Compute the topology of the tree T for the additive matrix D.
- Weight the edges of T optimally to minimize the distance to d with respect to the L_∞ metric.

Theorem 16 *Let T be a binary model tree, x the minimum weight of any edge in T, and D^* the additive matrix associated to T. Let n sequences evolve on T, and let d be the matrix of corrected distances computed on these sequences. If d satisfies $L_\infty(d, D^*) < x/8$, then the algorithm above solves the L_∞-nearest tree problem.*

Proof: The proof is quite simple. If D^{opt} is the nearest additive metric to d with respect to the L_∞-metric, then $x/8 > L_\infty(d, D^*) > L_\infty(d, D^{opt})$. Now suppose D is the additive matrix returned by the Agarwala *et al.* algorithm given distance d. Then $L_\infty(d, D) \leq 3 * L_\infty(d, D^{opt})$, so that $L_\infty(D, D^*) \leq L_\infty(D, d) + L_\infty(d, D^*) < 3x/8 + x/8 = x/2$. But then D and D^* both define the same topology by Theorem 8. By the same theorem, D^* and D^{opt} also have the same topology since $L_\infty(D^*, D^{opt}) < x/2$. Consequently, the Agarwala *et al.* algorithm returns a tree whose topology is identical to that of the nearest tree to d with respect to the L_∞ metric. Weighting that tree topology optimally takes polynomial time. ∎

5.3 Commentary

The Agarwala *et al.* algorithm in [93] is a variant on a classical approach to approximating the nearest trees, originally introduced by Carroll and Pruzansky in 1980 [124]. On any input the Agarwala *et al.* algorithm obtains a 3-approximate solution to the L_∞-nearest tree, and hence on *long enough* input, it will with high probability also provide a method to solve the L_∞-nearest tree. The same, however, can be said of the simple methods we described above (iterative addition of different taxa with respect to some ordering on the leaves, or quartet based methods). There are therefore several methods for reconstructing optimal L_∞-nearest trees from long enough sequences, and the major distinction between the different methods is the sequence length requirement of each method.

Our other observations are as follows:

1. *Not all polynomial time heuristics have guaranteed performance on long enough sequences.* Both quartet-based methods and iteratively adding nodes to the best tree found so far have this property. On the other hand, Mike Steel (personal communication) has pointed out that a branch-swapping approach may *not* be consistent if it begins with an arbitrarily selected tree, since there is no proof that such a strategy will terminate at a global optimum, nor that it will take only polynomial time. It is therefore interesting and worth noting that the practice in systematics is to begin the hill-climbing search for optimal trees with trees that are constructed according to this iterative procedure. This explains why experimental studies based upon simulating sequence evolution show that these heuristics do provide accurate solutions to NP-hard problems, once the sequences are long enough.

2. *The convergence rate can be affected by the choice of sequence ordering!* Note that the ordering of the sequences can be chosen well or badly. For example, if the topology induced by the first four taxa falls in the *Felsenstein Zone* [125] (a particular portion of the parameter space for four-taxon model trees in which parsimony is known to fail with probability 1 on unbounded length sequences), then techniques based upon reconstructing the tree which begin with those four sequences will in general require much longer sequences before complete accuracy is likely than other orderings of the leaves.

3. *When attempting to solve parsimony, or other "inconsistent" estimators, the particular choice of heuristic can lead to incorrect reconstructions.* The problem with parsimony is that it *can* be inconsistent on some model trees. This can cause significant problems, *even if* parsimony is consistent for the specific tree underlying a particular data set. As a specific example, parsimony may be a consistent estimator

for model trees which are caterpillars (i.e. trees consisting of one long path, with leaves hanging off the path), in which all edges have some specific small mutation probability (this is a conjecture, which seems to be true in simulation studies, but we do not know whether it is true). Suppose that the first four sequences that are selected consist of two leaves from the very middle, and the first and the last leaves. Such a set of four sequences forms a tree in the "Felsenstein Zone". If these are the first four sequences upon which the entire tree will be reconstructed, then at the very start of the process the tree being constructed will be incorrect. Hence, heuristics used to reconstruct the "most parsimonious" tree will *not* automatically result in consistent estimators, even if parsimony is consistent for the specific trees being reconstructed, and in such cases more sophisticated algorithms may be needed.

4. *These results do not fully apply to maximum likelihood estimation!* We do not know if the likelihood function can be maximized in polynomial time on a fixed leaf-labelled tree, and hence this result does not imply that maximum likelihood can be solved in polynomial time given long enough sequences. However, if we solve maximum likelihood exactly in each iteration (and do not just find "approximate" solutions), then these results do apply.

5. *These observations may not apply to real data.* These proofs of accuracy of polynomial time heuristics for NP-hard problems most definitely require that the optimization problems be consistent estimators for the trees that generated the data. The proofs of consistency for all reasonable distance-based methods most definitely rely upon the *iid* assumption, which real data are known to violate. Consequently, the performance guarantee we have shown for simple heuristics applied to simulated data may not hold when applied to real data! Again, in such cases, if the optimization problem is deemed appropriate for tree reconstruction purposes, we may need more sophisticated methods than simple "sequential insertion" techniques, even if followed by hill-climbing strategies (which, as Mike Steel pointed out, need not be consistent nor only require polynomial time).

6 Are all optimization problems equally important?

We have noted that many optimization problems seem to be solvable if the sequences are long enough and are generated by a Cavender-Farris tree. We have also noted that many optimization problems are consistent estimators, again under the assumption that the sequences are generated by Cavender-Farris trees. Does this mean that the tree reconstruction problem is essentially solved?

Not at all. First, as we have noted, real data are not generated on Cavender-Farris trees. This is a critical point which suggests that we still need to study the performance of different heuristics on more realistic simulation studies, and possibly we still need to develop better algorithms for finding the best trees, for various definitions of "best". However, the problem is not "solved" satisfactorily even for the case of data generated on Cavender-Farris trees, because in any event sequences are finite, and performance on finite length sequences is not easy to predict from performance in the limit. In addition, there is significant evidence that some optimization problems simply return much more accurate topology estimations from realistic length sequences than other optimization problems. In fact, a recent paper by Gascuel [80] introducing BIONJ, a statistically based variant on the neighbor-joining method, examined the relationship between topology estimation and optimizing a different numeric objective criterion, the *minimum evolution* (ME) score. Gascuel noted:

> BIONJ finds trees which are not shorter than NJ's, in the sense of the ME criterion. Specifically, it seems that BIONJ trees are just a little shorter with constant-rate trees than NJ's, but longer with varying-rate trees – where it outperforms NJ. Moreover, it appears that trees found by both NJ and BIONJ are more often too short (*i.e.*, shorter than the true tree) than too long. This explains why searching for trees shorter than NJ trees may not increase the topological accuracy.

Similarly, we note that although we can obtain near-optimal solutions for the L_∞-nearest tree by using the Agarwala *et al.* algorithm [93] (or its variant, the Double-Pivot [119]), both experimental [61] and analytical studies [127, 118] suggest that these methods are not likely to produce a correct estimates of the topology of the underlying evolutionary tree at realistic sequence lengths as, for example, neighbor-joining.

The critical distinction between phylogenetic reconstruction and numerical taxonomy is that the objectives of the two problems are not really the same. In phylogenetic reconstruction, the objective is the reconstruction of the topology of the evolutionary tree that gave rise to the observed sequences, while in numerical taxonomy the objective is to minimize some distance between distance matrices. Although we have analytical results that prove convincingly that exact (or approximate) solutions to numerical taxonomy optimization problems *will* be consistent, and hence will provide accurate reconstructions of the topology from long enough sequences, it is increasingly clear that the required sequence lengths for different optimization criteria can be quite different (see [118, 127]). Since sequence lengths are inherently bounded (and not by extremely large numberes, either), this has the consequence of shifting the attention from what happens in the limit to how the method behaves on finite data.

It is also important to note that performance issues concerned with topology (as is generally the objective in the performance literature in systematic biology) do not directly correlate to performance with respect to other metrics such as the "Minimum Evolution" criterion, as discussed above and in [80], nor the "variational distance", as is studied in [126, 6]). For this reason, studies such as [119, 126, 6] which do not look explicitly at the topology estimation question may not be predictive of performance with respect to topology. Indeed, experimental studies in [61, 80] have suggested that accuracy of topology is often not something easy to predict on the basis of analytical results related to distance-based optimization!

There is also a critical distinction between performance in the limit (i.e. from long enough sequences) and performance on finite data, and understanding the distinction may, in general, require experimental studies, since analytical studies are both hard to obtain and very difficult to interpret. Thus, theorems about the sequence lengths needed to obtain accurate reconstructions of the topology of the evolutionary tree given in [118, 126, 6] are given in terms of sequence lengths that *suffice* to guarantee good performance but indicate nothing about behavior on shorter sequences. Worse, these analytical results are given in terms of "big-oh" notation, and thus hide critically important constants. In effect, such analytical results *only* give information about performance in the limit, and are in any event pessimistic.

7 What about Real Data?

There is a danger in this analysis which must not be overlooked.

We have assumed that the data are generated by an appropriate Markov process and that the sites evolve identically and independently. However, biological data do not fit these models particularly well. For example, the assumption that the sites evolve identically and independently is known to be violated by biomolecular sequences; third positions within codons evolve much more rapidly than the other positions, since these are often silent mutations and do not affect the encoding into aminoacids. Unfortunately, if we allow the assumption that the sites evolve identically to be violated (but do not violate any other assumption), then it is no longer clear how well the different methods work. No method is consistent under such general circumstances, and we do not yet know anything about the degree to which accuracy drops for the various methods. Thus, the issue of *robustness* to model violations is pressing and critical to real applications, and unfortunately little understood.

Thus, what we really need are methods which provide reasonable estimates of the topology of the tree in the presence of model violations. However, these may be harder to obtain than the results we obtained (see Observation 1) about how well simple methods perform with respect to reconstructing model trees when the sequences evolve on a tree with the *iid* site evolution, and the sequences are long enough (the proof we gave will not in general apply for real data, since in such cases a method may be consistent for estimating a particular tree, and yet not be a consistent estimator for some subtrees of that tree!). Thus, most likely we will need methods which provably obtain accurate solutions to hard optimization problems (such as parsimony or the least squares metric), and which do not operate through sequential addition of taxa.

Finally, we note that we have not discussed the problem of obtaining an accurate alignment of the biomolecular sequences from which the tree will be reconstructed. This "multiple alignment" problem is an absolutely critical component of the tree reconstruction problem *in practice*, since (as we have shown) all tree reconstruction approaches assume an alignment before constructing the tree. While obtaining an accurate alignment in some cases is not problematic, in other cases it is potentially very difficult, and we do not yet know how robust different methods are to errors in the underlying sequence alignment.

Unfortunately, the multiple sequence alignment problem is one of the most difficult components of the procedure, and indeed may be computationally one of the most complex problems in computational biology. It is often assumed that the correct evolutionary tree for the sequences is given in advance, but as we have seen tree reconstruction often requires an alignment and hence this is a sometimes problematic assumption. Consequently, methods have been developed which reconstruct trees and a multiple sequence alignment simultaneously (see Hein's method [37]), but these methods are not known to optimize any objective criterion. If a tree is known for the sequences, then a "tree alignment" is the objective, where a tree alignment is a multiple alignment which is defined as the extension of the induced pairwise alignments indicated by the edges of the tree; hence every node of the tree is labelled by a sequence. This problem was introduced by Sankoff in [96]. Sankoff presented an exponential time algorithm for obtaining an optimal tree alignment on a fixed tree in [96], and the problem was later shown to be NP-hard [52]. Later it was shown to be approximable, and even to admit a PTAS if the degree of the tree is bounded [99, 100]. Unfortunately, the approximation algorithms are potentially not useful for tree reconstruction purposes (the 2-approximation is probably not good enough, and the PTAS is surely too slow). Thus, this specific problem remains a bottleneck for some tree reconstruction problems.

8 Further reading

We strongly recommend that interested readers see the very interesting surveys that have been written by Felsenstein [57, 30] and Swofford *et al.* [102]. Experimental studies are particularly illuminative and present evidence about performance that is not readily obtainable through analytical techniques. Of the few experimental studies of large data sets, some [101, 61] have been discouraging about the predicted performance of standard distance methods on large trees, although others [129] find that methods are capable of accurate reconstructions of large trees from sequences that are long but not astronomical. Many experimental papers have examined smaller trees; for example Huelsenbeck has written very interesting papers exploring the parameter space for four-taxon trees quite thoroughly [95], while others have examined larger trees [62, 63, 122].

The material on classical methods for distance-based reconstruction is obtained from [11], which is a rich source of theorems and citations into the theoretical literature in numerical taxonomy.

New and interesting algorithms for tree reconstruction are being developed by various theoretical computer scientists, especially in the areas of *consensus tree* reconstruction, which we have explicitly avoided discussing. It is a rich area in its own right. An entry into those interesting questions can be obtained in [1, 2, 12, 16, 21, 25, 28, 29, 66, 40, 43, 128, 44, 45, 46, 47, 48, 51, 53, 106, 10, 91, 54]. There are many new algorithms being developed by the theoretical computer science community which may not be as immediately useful, but which are very interesting in their own right. For example, "bounded imperfection parsimony" has been addressed by Fernandez-Baca and Lagergren [31], and problems for tree reconstruction based upon partial orders on pairwise distances has been considered

[39, 42, 41].

9 Acknowledgements

The author gratefully acknowledges the support of the National Science Foundation which supported this research through two grants, CCR-9457800 and SBR-9512092, Paul Angello, and the David and Lucille Packard Foundation. Much thanks to Laszlo Szekeley for editing the text, Ken Rice, Junhyong Kim, Mike Steel, and Mike Sanderson for interesting conversations about statistical and scientific issues related to tree reconstruction, and Menaka Sampath for proofreading.

10 Bibliography

References

[1] Adams, E. III, *Consensus techniques and the comparison of taxonomic trees*, Syst. Zool., 21: 390-397, 1972.

[2] Adams, E. III, *N-trees as Nestings : Complexity, Similarity, and Consensus*, Journal of Class., 3: 299-317, 1986.

[3] Agarwala, R. and D. Fernández-Baca. *Simple Algorithms for Perfect Phylogeny and Triangulating Colored Graphs.* In the special issue on Algorithmic Aspects of Computational Biology, International Journal of Foundations of Computer Science, Vol. 7, No. 1, pp. 11–21, 1996.

[4] Agarwala, R. and D. Fernández-Baca. *A Polynomial-Time Algorithm for the Perfect Phylogeny Problem when the Number of Character States is Fixed.* SIAM Journal on Computing, Vol. 23, No. 6, pp. 1216–1224, December 1994. Also available in Proceedings of the 34th Annual Symposium on Fundamentals of Computer Science, pp. 140–147, 1993.

[5] Agarwala, R., D. Fernández-Baca, and G. Slutzki. *Fast Algorithms for Inferring Evolutionary Trees.* Journal of Computational Biology, Vol. 2, No. 3, pp. 397–408, Fall 1995. An earlier version of this paper is available in Proceedings of the 30th Allerton Conference on Communication, Control, and Computing, pp. 594–603, 1992.

[6] Ambainis, A., Desper, R., Farach, M., and S. Kannan (1997), *Nearly tight bounds on the learnability of evolution,* to appear, FOCS 1997.

[7] Arnborg, A., Corneil, D., and A. Proskurowski. *Complexity of finding embeddings in a k-tree.* SIAM J. Algebraic Discrete Meth., 8 (1987), pp. 277-284.

[8] Bandelt, H-J., and A. Dress (1986), *Reconstructing the shape of a tree from observed dissimilarity data*, Advances in Applied Mathematics, 7 pp. 309-343.

[9] Bandelt, H-J., and A. Dress (1992), *A canonical decomposition theory for metrics on a finite set,* Advances in Mathematics, Vol. 92, No. 1, pp. 47-105.

[10] Barthélemy, J. and Janowitz, F. *A formal theory of consensus*, SIAM J. Disc. Math., 3:305-322 (1991).

[11] Barthélemy, J. and A. Guénoche, Trees and Proximity Representations. 1991, John Wiley and Sons LTd. West Sussex, England.

[12] Barthélemy, J. and McMorris, F. *The median procedure for n-Trees*, Journal of Classification, 3:329-334 (1986).

[13] Bellare, M. and M. Sudan, *Improved non-approximability results*, Proceedings of the Twenty-Sixth Annual ACM Symposium on Theory of Computing, (Montreal), ACM, pp. 184-193.

[14] Blanken, R., Klotz, L., and A. Hinnebush. (1982) *Computer comparisons of new and existing criteria for constructing evolutionary trees from sequence data,* J. Mol. Evol. 19, 9-19.

[15] Brossier, G. (1985) *Approximation des dimmilarités par des arbres additifs,* Math. Sci. Hum. 91, 5-21.

[16] Bryant, D. "Hunting for binary trees in binary character sets: efficient algorithms for extraction, enumeration, and optimization", Research Report #124, Department of Mathematics and Statistics, Canterbury University, Christchurch, New Zealand, April 1995.

[17] Buneman, P. Mathematics in the archaeological and historical sciences (F.R. Hobson, D.G. Kendal, and P. Tautu, eds.) University Press, Edinburgh, p. 387.

[18] Cavender, J.A. *Taxonomy with confidence,* Math. Biosci., **40** (1978), 271–280.

[19] Cavender, J.A. and J. Felsenstein, *Invariants of phylogenies: simple case with discrete states,* J. Classification, **4** (1987), 57–71.

[20] Farris, J.S. *A probability model for inferring evolutionary trees,* Syst. Zool., **22** (1973), 250–256.

[21] Day, W.H.E. (1985), *Optimal algorithms for comparing trees with labelled leaves*, Journal of Classification 2, pp. 7-28.

[22] Day, W.H.E., and Sankoff, D. (1986), *Computational complexity of inferring phylogenies by compatibility*, Systematic Zoology, 35, pp. 224-229.

[23] Day, W.H.E. *Computational complexity of inferring phylogenies from dissimilarity matrices*, Bull. of Math. Biol, Vol 49, No. 4, pp. 461-467, 1987.

[24] Estabrook, G.R., Johnson, C.S., Jr., and F.R. McMorris (1976), *A mathematical foundation for the analysis of cladistic character compatibility*, Math. Biosci. 29, pp. 181-187.

[25] Estabrook, G.F., and McMorris, F.R. (1980), *When is one estimate of evolutionary relationships a refinement of another?*, J. Math. Biosci. 10, pp. 327-373.

[26] Estabrook, G.F., Johnson, C.S. Jr. and F.R. McMorris, *An idealized concept of the true cladistic character*, Math. Biosci. 23, 1975, pp. 263-272.

[27] Estabrook, G.F., Johnson, C.S. Jr. and F.R. McMorris, *An algebraic analysis of cladistic characters*, Discrete Math., 16, 1976, pp. 141-147.

[28] Farach, M. Przytycka, T. and M. Thorup, *On the agreement of many trees*, Information Processing Letters, to appear.

[29] Farach, M. and Thorup, M. (1994), *Optimal evolutionary tree comparisons by sparse dynamic programming*, Proceedings of the 35th annual IEEE Foundations of Computer Science, 1994, pp. 770–779, to appear SIAM Journal on Computing.

[30] Felsenstein, J. (1988) *Phylogenies from molecular sequences: inferences and reliability,* Ann. Rev. Genetics, 22:521-565.

[31] Fernández-Baca, D., and J. Lagergren, *A polynomial time algorithm for near-perfect phylogeny*, Proceedings ICALP 1996, pp. 670-680.

[32] Fitch, W.M., and E. Margolaish (1967), *Construction of phylogenetic trees,* Science, 155, 279-284.

[33] Foulds, L.R., and R.L. Graham. 1982. *The Steiner Problem in Phylogeny is NP-Complete*, Adv. Appl. Math. 3, 43-49.

[34] Garey, M.R. and Johnson, D.S, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W.H. Freeman and Company, NY, 1979.

[35] Gower, J., and G. Ross (1969). *Minimum spanning trees and single linkage cluster analysis, Appl. Stat.* 18, 54-64.

[36] Gusfield, D. (1991), *Efficient algorithms for inferring evolutionary trees*, Networks 21, pp. 19-28.

[37] Hein, J. (1990) Unified approach to alignment and phylogenies, in *Methods in Enzymology*, vol 183.

[38] Jardine, C., Jardine, N., and R. Sibson. (1967) *The structure and construction of taxonomic hierarchies,* Math. Biosci. 1, 173-179.

[39] Kannan, S. and T. Warnow. *Tree reconstruction from partial orders.* SIAM J. Computing. Vol. 24, No. 3, pp. 511-519, 1995.

[40] Kao, M. *Tree contractions and evolutionary trees,* manuscript. (1995).

[41] Kearney, P. *A six-point condition for ordinal matrices,* to appear, Journal of Computational Biology, 1997.

[42] Kearney, P., Hayward, R.B., and H. Meijer, (1997) *Inferring evolutionary trees from ordinal data,* Proceedings of the 8th Annual ACM-SIAM Symposium on Discrete Algorithms, pp. 418-426.

[43] Keselman, D. and A. Amir, *Maximum agreement subtree in a set of evolutionary trees - metrics and efficient algorithms*, Proceedings of the 35th annual IEEE Foundations of Computer Science, 1994, pp. 758–769.

[44] Nelson, G. *Cladistic analysis and synthesis: Principles and definitions, with a historical note on Adanson's* Familles des Plantes *(1763-1764).* Syst. Zool., 28:1-21, 1979.

[45] Nelson, G. and N.I. Platnick, *Systematics and biogeography : Cladistics and vicariance*, Columbia Univ. Press, New York, 1981.

[46] Page, R.D.M. (1990), *Tracks and Trees in the Antipodes: a reply to Humphries and Seberg.* Syst. Zool. 39(3):288-299, 1990.

[47] Page, R. D. M. *Genes, Organisms, and Areas: The Problem of Multiple Lineages*, Systematic Biology, 42(1), 77-84, 1993.

[48] Page, R. D. M. (1993), *Reconciled Trees and Cladistic Analysis of Historical Associations Between Genes, Organisms, and Areas*, manuscript.

[49] Sokal, R., and P. Sneath, (1963), *Principles of Numerical Taxonomy*, Freeman, San Francisco.

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

[51] Steel, M.A., and Warnow, T.J. (1993), *Kaikoura tree theorems: Computing the maximum agreement subtree,* Information Processing Letters 48, pp. 72-82.

[52] Wang, L. and T. Jiang, (1994) *On the complexity of multiple sequence alignment,* J. Computational Biology, 1:337-348.

[53] Wareham, H. T. (1985) “An Efficient Algorithm for Computing Ml Consensus Trees”, Honors Dissertation, Department of Computer Science, Memorial University of Newfoundland, St. John's, Newfoundland.

[54] Warnow, T.J. (1994), *Tree compatibility and inferring evolutionary history*, Journal of Algorithms, 16, pp. 388-407.

[55] Waterman, M.S., Smith, T.F., Singh, M. and W.A. Beyer, *Additive evolutionary trees,* Journal Theoretical Biol. 64:199-213, 1977.

[56] Estabrook, G.R., Johnson, C.S., Jr., and F.R. McMorris (1976), *A mathematical foundation for the analysis of cladistic character compatibility*, Math. Biosci., 29, 1976, pp. 181-187.

[57] J. Felsenstein, *Numerical methods for inferring evolutionary trees*, The Quarterly Review of Biology, Vol. 57, No. 4, Dec. 1982.

[58] Kannan, S. Lawler, E. and T. Warnow, *Determining the evolutionary tree*, (1996), J. of Algorithms, 21, pp. 26-50.

[59] Kim, J. 1996. *General inconsistency conditions for maximum parsimony: effects of branch lengths and increasing numbers of taxa.* Syst. Biol. 45(3): 363-374.

[60] Schoniger, M. and A. von Haeseler Performance of maximum likelihood, neighbor joining, and maximum parsimony methods when sequence sites are not independent. *Syst. Biol.* (1995) 44:4 533-547.

[61] Rice, K.A. and T. Warnow 1997. Parsimony is hard to beat! *COCOON97 Conference Proceedings*, in press.

[62] Kuhner, M. and J. Felsenstein, 1994. *A simulation comparison of phylogeny algorithms under equal and unequal evolutionary rates.* Mol. Biol. Evol. 11:459-468.

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

[64] Kannan, S. and T. Warnow. 1994. *Inferring Evolutionary History from DNA Sequences,* SIAM J. on Computing, Vol. 23, No. 4, pp. 713-737. (A preliminary version of this paper appeared at FOCS 1990.)

[65] Farach, M., Kannan, S. and T. Warnow. 1996. *A Robust Model for Finding Optimal Evolutionary Trees,* Algorithmica, special issue on Computational Biology, Vol. 13, No. 1, pp. 155-179. (A preliminary version of this paper appeared at STOC 1993.)

[66] Kannan, S., Warnow, T. and S. Yooseph. 1995. *Computing the local consensus of trees,* The Association for Computing Machinery and the Society of Industrial Applied Mathematics, Proceedings, ACM/SIAM Symposium on Discrete Algorithms, 1995, pp. 68-77. To appear, SIAM J. Computing.

[67] Kannan, S. and T. Warnow. 1995. *A fast algorithm for the computation and enumeration of perfect phylogenies when the number of character states is fixed,* Proceedings, ACM/SIAM Symposium on Discrete Algorithms, 1995. To appear, SIAM J. Computing, 1997.

[68] Warnow, T. *Mathematical approaches to comparative linguistics.* Proceedings of the National Academy of Sciences, 1997.

[69] Bonet, M., Phillips, C.A., Warnow, T., and S. Yooseph. 1996. *Constructing evolutionary trees in the presence of polymorphic characters,* ACM Symposium on the Theory of Computing, 1996.

[70] Ringe, D., Warnow, T., Taylor, A., Michailov, A., and L. Levison. 1997. Computational cladistics and the position of Tocharian. In V. Mair (Ed.), *The Bronze Age and Early Iron Age Peoples of Eastern Central Asia*, a special volume of the Journal of Indoeuropean Studies.

[71] Bodlaender, H. Fellows, M. and Warnow, T. 1992: *Two strikes against perfect phylogeny,* Proceedings, International Congress on Automata and Language Processing.

[72] Bodlaender, H. and Kloks, T. 1993: *A simple linear time algorithm for triangulating three-colored graphs, Journal of algorithms,* 15, pp. 160-172.

[73] Buneman, P. 1974: *A characterization of rigid circuit graphs,* Discrete mathematics, 9:205-212.

[74] Camin, J. and Sokal, R. 1965: *A method for deducing branching sequences in phylogeny,* Evolution, 19: pp. 311-326.

[75] Day, W.H.E. 1983: *Computationally difficult parsimony problems in phylogenetic systematics,* Journal of Theoretical Biology, 103: 429-438.

[76] Dress, A. and Steel, M.A. 1992: *Convex Tree Realizations of Partitions,* Applied Math Letters, 5(3): 3-6.

[77] Estabrook, G.F. and Landrum, L. 1975: *A simple test for the possible simultaneous evolutionary divergence of two aminoacid positions,* Taxon. 24: 609-613.

[78] Estabrook, G.F. and McMorris, F.R. 1977: *When are two qualitative taxonomic characters compatible?,* J. Math. Biol., 4: 195-200.

[79] Fitch, W.M. 1975: *Toward finding the tree of maximum parsimony,* Proc. Eighth International Conference on Numerical Taxonomy, G.F. Estabrook, ed., pp. 189-230, W.H. Freeman, San Francisco.

[80] Gascuel, O., *BIONJ: an improved version of the NJ algorithm based on a simple model of sequence data,* Cahier du GERAD G-97-18, to appear in Molecular Biology and Evolution, July 1997.

[81] Gusfield, D. 1984: *The steiner tree problem in phylogeny,* Technical Report 332, Yale University, Department of Computer Science.

[82] Idury, R. and Schaffer, A. 1993: *Triangulating three-colored graphs in linear time and linear space,* to appear, SIAM J. Discrete Mathematics.

[83] Kannan, S. and Warnow, T. 1992: *Triangulating three-colored graphs,* SIAM J. Discrete Mathematics, May 1992, pp. 249-258.

[84] LeQuesne, W.J. 1969: *A method of selection of characters in numerical taxonomy,* Syst. Zool., 18: 201-205.

[85] LeQuesne, W.J. 1972: *Further studies on the uniquely derived character concept,* Syst. Zool., 21: 281-288.

[86] LeQuesne, W.J. 1974: *The uniquely evolved character concept and its cladistic application,* Syst. Zool., 23: 513-517.

[87] LeQuesne, W.J. 1977: *The uniquely evolved character concept,* Syst. Zool., 26, pp. 218-223.

[88] McMorris, F.R. 1977: *On the compatibility of binary qualitative taxonomic characters,* Bull. Math. Biol. 39: 133-138.

[89] McMorris, F.R. and C.A. Meacham, *Partition intersection graphs*, Ars Combinatorica, 16-B, pp. 135-138.

[90] McMorris, F.R., Warnow, T. and Wimer, T. 1993: *Triangulating vertex colored graphs, SIAM journal of discrete mathematics*, Vol. 7, No. 2, pp. 296-306.

[91] Steel, M.A. 1992: The complexity of reconstructing trees from qualitative characters and subtrees, *Journal of Classification*, Vol. 9:91-116. 1992.

[92] Swofford, D, 1993: PAUP (Phylogenetic Analysis Using Parsimony). It can be ordered from the Center for Biodiversity, Illinois Natural History Survey, 607 East Peabody Drive, Champaign, Illinois 61820, U.S.A.

[93] Agarwala, R., Bafna, V., Farach, M., Narayanan, B., Paterson, M., and M. Thorup. *On the approximability of numerical taxonomy: fitting distances by tree metrics.* Proceedings of the 7th Annual ACM-SIAM Symposium on Discrete Algorithms, 1996.

[94] Atteson, K. *On the performance of the neighbor-joining method*, to appear, Proceedings of COCOON 1997.

[95] Huelsenbeck, J. and D. Hillis, *Success of phylogenetic methods in the four-taxon case.* Syst. Bio. 42(3):247-264, 1993.

[96] Sankoff, D. (1975). *Minimum mutation trees of sequences,* SIAM J. on Applied Mathematics. 28(1):35-42.

[97] Steel, M.A., L.A. Székely, and M.D. Hendy, *Reconstructing trees when sequence sites evolve at variable rates,* Journal of Computational Biology, Volume 1, No. 2, 1994, pp. 153-163.

[98] Gusfield, D. (1991), *Efficient algorithms for inferring evolutionary trees*, Networks 21, pp. 19-28.

[99] Wang, L., Jiang, T., and E. Lawler, *Approximation algorithms for tree alignment with a given phylogeny,* Algorithmica 16, 1996, pp. 302-315.

[100] Wang, L. and D. Gusfield, *Improved approximation algorithms for tree alignment,* Journal of Algorithms, to appear.

[101] Strimmer, K. and A. von Haeseler. 1996. *Accuracy of neighbor joining for n-taxon trees,* Syst. Biol. 45 (4): 516-523.

[102] Swofford, D.L., Olsen, G.J., Waddell, P., and D. M. Hillis, Chapter 11: Phylogenetic inference, in: *Molecular Systematics*, D. M. Hillis, C. Moritz, B. K. Mable, eds., 2nd edition, Sinauer Associates, Inc., Sunderland, 1996, 407–514.

[103] Templeton, A. Human origins and analysis of mitochondrial DNA sequences. *Science* , Vol. 255, 737-739, 1992.

[104] Wilson, A. C. and R. L. Cann, The recent African genesis of humans, *Scientific American* April 1992, 68–73.

[105] Vawter, L, and Wm. Brown, Science 234 (4773): 194-196 (Oct 10 1986) Nuclear and mitochondrial DNA comparisons reveal extreme rate variation in the molecular clock.

[106] Phillips, C.A. and T. Warnow. 1996. The Asymmetric Median Tree: a new model for building consensus trees. Discrete Applied Mathematics, Special Issue on Computational Molecular Biology, 71, pp. 311-335.

[107] J. A. Hartigan, *Minimum mutation fits to a given tree,* Biometrics 29, 1973, pp. 53-65.

[108] Fitch, Wm. 1971: Toward defining the course of evolution: minimum change for a specified tree topology, *Syst. Zool.*, 20:406-416.

[109] Warnow, T., Ringe, D. and A. Taylor. 1996. Reconstructing the evolutionary history of natural languages, Association for Computing Machinery and the Society of Industrial and Applied Mathematics, Proceedings of ACM-SIAM Symposium on Discrete Algorithms (SODA), 1996, pp. 314-322.

[110] Saitou, N., and M. Nei. 1987. *The neighbor-joining method: A new method for reconstructing phylogenetic trees.* Mol. Biol. Evol. 4:406-425.

[111] Bruns, T.D. and T.M. Szaro (1992). *Rate and mode differences between nuclear and mitochondrial small-subunit rRNA genes in mushrooms.* Mol. Biol. Evol. 9 (5): 836-855.

[112] Sharp, P. and W.H. Li (1989). *On the rate of DNA sequence evolution in Drosophila*, J. Mol. Evol. 28 (5): 398-402.

[113] Sourdis, J. and M. Nei, *Relative efficiencies of the maximum parsimony and distance-matrix methods in obtaining the correct phylogenetic tree.* Mol. Biol. Evol. (1996) 5:3 293-311.

[114] Li, W.H. and M. Tanimura (1987), *The molecular clock runs more slowly in man than in apes and monkeys.* Nature 326 (6108): 93-96.

[115] Li, W.H., Tanimura M., and P.M. Sharp (1987), *An evaluation of the molecular clock hypothesis using mammalian DNA sequences.* J Mol Evol 25 (4): 330-342.

[116] Gillespie, J.H. (1984). *The molecular clock may be an episodic clock.* Proc. Natl. Acad. Sci. U S A 81 (24): 8009-8013.

[117] Strimmer, K. and A. von Haeseler, *Quartet Puzzling: a quartet maximum likelihood method for reconstructing tree topologies,* Mol. Biol. Evol., 1996, 964–969.

[118] Erdös, P., Steel, M., Szekeley, L. and T. Warnow. 1997. Inferring big trees from short sequences. Proceedings of International Congress on Automata, Languages, and Programming 1997.

[119] J. Cohen and M. Farach, Numerical Taxonomy on Data: Experimental Results. SODA '97 and RECOMB '97.

[120] Carroll, J.D. (1976). Spatial, non-spatial, and hybrid models for scaling, *Psychometrika*, 41 (4) 439-463.

[121] Berry, V. and O. Gascuel, *Inferring evolutionary trees with strong combinatorial evidence*. To appear, proceedings of COCOON 1997.

[122] Berry, V. and O. Gascuel (1996), *On the interpretation of bootstrap trees: appropriate threshold of clade selection and induced gain,* Mol. Biol. Evol. 13(7): 999-1011.

[123] Aldous, D. (1995). *Probability distributions on cladograms*, in: Discrete Random Structures, eds. D. J. Aldous and R. Permantle, Springer-Verlag, IMA Vol. in Mathematics and its Applications. Vol. 76, 1-18.

[124] Carroll, J.D., and S. Pruzansky. (1980) Discrete and hybrid scaling models. In *Similarity and Choice*, E.B. Lanterman and H. Freger, eds. Hans Huber, Berne.

[125] Felsenstein, J. *Cases in which parsimony or compatibility methods will be positively misleading*, Syst. Zoology, 27:401-410, 1978.

[126] Farach, M. and S. Kannan (1996). *Efficient algorithms for inverting evolution, Proceedings of the ACM Symposium on the Foundations of Computer Science*, 230–236.

[127] Atteson, K. *Results on Neighbor-Joining's Convergence Rate*, to appear, COCOON 1997.

[128] McMorris, F.R. and Steel, M. (1994), *The complexity of the median procedure for binary trees.* Proceedings of the 4th Conference of the International Federation of Classification Societies, Paris 1993, to be published in the series "Studies in Classification, Data Analysis, and Knowledge Organization" by Springer Verlag.

[129] Hillis, D. (1997). *Inferring complex phylogenies*, Nature, Vol. 383, pp. 130-131.