DIMACS Technical Report 97-72
October 1997

A few logs suffice to build (almost) all trees (II)

by

Péter L. Erdős
Mathematical Institute of the Hungarian Academy of Sciences
Budapest P.O.Box 127, Hungary-1364
E-mail: elp@math-inst.hu

Michael A. Steel
Biomathematics Research Centre
University of Canterbury
Christchurch, New Zealand
E-mail: m.steel@math.canterbury.ac.nz

László A. Székely
Department of Mathematics
University of South Carolina
Columbia, South Carolina
E-mail: laszlo@math.sc.edu

Tandy J. Warnow
Department of Computer and Information Science
University of Pennsylvania, Philadelphia PA
E-mail: tandy@central.cis.upenn.edu

DIMACS is a partnership of Rutgers University, Princeton University, AT&T Labs-Research, Bell Labs and Bellcore.

DIMACS is an NSF Science and Technology Center, funded under contract STC–91–19999; and also receives support from the New Jersey Commission on Science and Technology.

ABSTRACT

Inferring evolutionary trees is an interesting and important problem in biology that is very difficult from a computational point of view as most associated optimization problems are NP-hard. Although it is known that many methods are provably statistically consistent (i.e. the probability of recovering the correct tree converges on 1 as the sequence length increases), the actual rate of convergence for different methods has not been well understood. In a recent paper we introduced a new method for reconstructing evolutionary trees called the Dyadic Closure Method (DCM), and we showed that DCM has a very fast convergence rate. DCM runs in $O(n^5 \log n)$ time, where n is the number of sequences, so although it is polynomial it has computational requirements that are potentially too large to be of use in practice. In this paper we present another tree reconstruction method, the Witness-Antiwitness Method, or WAM. WAM is significantly faster than DCM, especially on random trees, and converges at the same rate as DCM. We also compare WAM to other methods used to reconstruct trees, including Neighbor Joining (possibly the most popular method among molecular biologists), and new methods introduced in the computer science literature.

1 Introduction

Rooted leaf-labelled trees are a convenient way to represent historical relationships between extant objects, particularly in evolutionary biology (where such trees are called "phylogenies"). Molecular techniques have recently provided large amounts of sequence data that are being used to reconstruct such trees. These methods exploit the variation in the sequences due to random mutations that have occurred at the sites, and statistically-based approaches typically assume that sites mutate independently and identically down the tree. For such models, it is possible with high probability to recover the underlying *unrooted* tree from adequately long sequences generated by the tree. How long the sequences have to be to guarantee high probability of recovering the tree depends on the reconstruction method, the details of the model, and the number n of species. Determining bounds on that length and its growth with n has become more pressing since biologists have begun to reconstruct trees on increasingly larger numbers of species (often up to several hundred) from such sequences.

In a previous paper [19], we addressed this question for trees under the Neyman 2-state model of site evolution, and obtained the following results:

1. We established a lower bound of $\log n$ on the sequence length that every method, randomized or deterministic, requires in order to reconstruct any given n-leaf tree in any 2-state model of sequence evolution,

2. We showed that the Maximum Compatibility method of phylogenetic tree construction requires sequences of length *at least* $n \log n$ to obtain the tree with high probability, and

3. We presented a new polynomial time method (the *Dyadic Closure Method* (DCM)) for reconstructing trees in the Neyman 2-state model, and showed that polylogarithmic length sequences suffice for accurate tree reconstruction with probability near one on almost all trees, and polynomial length sequence length always suffices for any tree under reasonable assumptions on mutation probabilities.

Thus, the Dyadic Closure Method [19] has a very fast convergence rate, which on almost all trees is within a polynomial of our established lower bound of $\log n$ for any method. However, although DCM uses only polynomial time, it has large computational requirements (it has $O(n^2k + n^5 \log n)$ running time, and uses $O(n^4)$ space), which may make it infeasible for reconstructing large trees.

In this paper, we present the *Witness-Antiwitness Method* (WAM), a new and faster quartet-based method for tree reconstruction, which has the same asymptotic convergence rate as the Dyadic Closure Method. This method works significantly faster then DCM, with a worst-case bound $O(n^2k+n^4 \log n \log k)$, but actually runs provably faster under reasonable restrictions on the model (see Theorem 12 for details). The *provable* bounds on the running time of WAM depend heavily on the depth of the model tree. We introduced the "depth"

in [19] and showed that $depth(T)$ is bounded from above by $\log n$ for all binary trees T, and that random trees have depths bounded by $O(\log \log n)$.

In addition to presenting the new method, we present a framework for a comparative analysis of different distance based methods, and use this framework to compare the convergence rate of several promising distance-based tree reconstruction methods: *Neighbor Joining* [38], the Agarwala *et al.* [1] approximation algorithm and its variant [20], and the Naive Quartet Method, which we describe in this paper. We show that all these distance-based methods may require sequences to grow exponentially in the *diameter* of the tree, and analyze the diameter of random trees under two distributions. We show that the diameter of random trees is $\Omega(\sqrt{n})$ under the uniform distribution, and $\Omega(\log n)$ under the Yule-Harding distribution. Consequently, these other distance-based methods *may* require, on any tree, longer sequences than required by either *DCM* or *WAM* (and much longer on almost all trees), in order to be *guaranteed (by this analysis)* to yield an accurate estimation of T. Finally we generalize our methods and results to more general Markov models, improving results of Ambainis *et al.* in [4].

The structure of the paper is as follows. In Section 2 we provide definitions and discuss tree reconstruction methods in general. In Section 3, we describe the analytical framework for comparing the sequence lengths needed by different methods for exact accuracy in tree reconstruction, and we use this framework to provide an initial comparison between various distance-based methods. In Section 4, we describe the Witness-Antiwitness Tree Construction algorithm (WATC), and in Section 5, we describe the Witness-Antiwitness Method (WAM) in full. In Section 6, we analyze the performance of WAM for reconstructing trees under the Neyman model of site evolution, and compare its performance to other promising distance based methods. We extend the analysis of WAM to reconstructing trees under the general r-state Markov model in Section 7. In Section 8 we discuss some important issues relevant for users of our algorithms on real data, and discuss the significance of DCM and WAM for multiple sequence alignment. We close in Section 9 with some discussion of future research and the limitations of existing methods for inferring trees from real biological data.

2 Definitions

Notation: $\mathbb{P}[A]$ denotes the probability of event A; $\mathbb{E}[X]$ denotes the expectation of random variable X. We denote the natural logarithm by log. The set $[n]$ denotes $\{1, 2, \ldots, n\}$ and for any set S, $\binom{S}{k}$ denotes the collection of subsets of S of size k. $\mathbb{R}$ denotes the real numbers.

Definitions: (I) Trees. We will represent a phylogenetic tree T by a *semi-labelled* tree whose *leaves* (vertices of degree one) are labelled by extant species, numbered by $1, 2, \ldots, n$, and whose remaining internal vertices (representing ancestral species) are unlabelled. We will adopt the biological convention that phylogenetic trees are *binary*, so that all internal nodes have degree three, and we will also assume that T is *unrooted* (this is due to scientific and technical reasons which indicate that the location of the root can be either difficult

or impossible). We let $B(n)$ denote the set of all $(2n-5)!! = (2n-5)(2n-7)\cdots 3 \cdot 1$ semi-labelled binary trees on the leaf set $[n]$.

The path between vertices u and v in the tree is called the uv path, and is denoted $P(u,v)$. The *topological distance* $L(u,v)$ between vertices u and v in a tree T is the number of edges in $P(u,v)$. A *cherry* in a binary tree is a pair of leaves i,j with $L(i,j)=2$. The edge set of the tree is denoted by $E(T)$. Any edge adjacent to a leaf is called a *leaf edge*, any other edge is called an *internal edge*. For a phylogenetic tree T and $S \subseteq [n]$, there is a unique minimal subtree of T, containing all elements of S. We call this tree the *subtree* of T induced by S, and denote it by $T_{|S}$. We obtain the *contracted subtree induced by* S, denoted by $T^*_{|S}$, if we substitute edges for all maximal paths of $T_{|S}$ in which every internal vertex has degree two. We denote by $ij|kl$ the tree on four leaves i,j,k,l in which the pair i,j is separated from the pair k,l by an internal edge. When the contracted subtree of T induced by leaves i,j,k,l is the tree $ij|kl$, we call $ij|kl$ a *valid quartet split* of T on the quartet of leaves $\{i,j,k,l\}$. Since all trees are assumed to be binary, all contracted subtrees (including, in particular, the quartet subtrees) are also binary. Consequently, the set $Q(T)$ of valid quartet splits for a binary tree T has cardinality $\binom{n}{4}$.

(II) Sites. Let us be given a set C of character states (such as $C=\{A,C,G,T\}$ for DNA sequences; $C=\{\text{the 20 amino acids}\}$ for protein sequences; $C=\{R,Y\}$ or $\{0,1\}$ for purine-pyrimidine sequences). A *sequence of length* k is an ordered k-tuple from C—that is, an element of C^k. A collection of n such sequences—one for each species labelled from $[n]$—is called a *collection of aligned sequences.*

Aligned sequences have a convenient alternative description as follows. Place the aligned sequences as rows of an $n \times k$ matrix, and call *site* i the i^{th} column of this matrix. A *pattern* is one of the $|C|^n$ possible columns.

(III) Site substitution models. Many models have been proposed to describe the evolution of sites as a stochastic process. Such models depend on the underlying phylogenetic tree T and some randomness. Most models assume that the sites are independently and identically distributed (*i.i.d.*). In addition the models on which we test our algorithm also assume the Markov property that the random assignment of a character state to a vertex v is expressed from the random character state of its immediate ancestor, and a random substitution on the connecting edge. 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 [42]. We now briefly discuss this general Markov model. 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 modelled 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 oriented out from the root has an associated $r \times r$ stochastic transition matrix $M(e)$. The random character state at the root "evolves" down the tree—thereby assigning characters randomly to the vertices, from the root down to the leaves. For each edge $e=(u,v)$, with u between v and the root, $(M(e))_{\alpha\beta}$ is the probability that v has character state β given that u has character state α.

(IV) The Neyman model. The simplest stochastic model is a symmetric model for binary characters due to Neyman [36], and was also developed independently by Cavender [12] and Farris [23]. Let $\{0,1\}$ denote the two states. The root is a fixed leaf, the distribution π at the root is uniform. For each edge e of T we have an associated *mutation probability*, which lies strictly between 0 and 0.5. Let $p : E(T) \rightarrow (0, 0.5)$ denote the associated map. We have an instance of the general Markov model with $M(e)_{01} = M(e)_{10} = p(e)$. We will call this the *Neyman 2-state model*, but note that it has also been called the Cavender-Farris model.

The Neyman 2-state model is hereditary on subsets of the leaves—that is, if we select a subset S of $[n]$, and form the subtree $T_{|S}$, then eliminate vertices of degree two, 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. Furthermore, the mutation probabilities 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.

Lemma 1 *The probability p that the endpoints of a path P of topological length k are in different states is related to the mutation probabilities $p_1, p_2, ..., p_k$ of edges of P as follows:*

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

Lemma 1 is folklore and is easy to prove by induction.

(V) Distances. Any symmetric matrix, which is zero-diagonal and positive off-diagonal, will be called a *distance matrix*[1]. An $n \times n$ distance matrix D_{ij} is called *additive*, if there exists an n-leaf tree (not necessarily binary) with positive edge lengths on the internal edges and non-negative edge lengths on the leaf edges, so that D_{ij} equals the sum of edge lengths in the tree along the $P(i, j)$ path connecting i and j. In [10], Buneman showed that the following Four-Point Condition characterizes additive matrices (see also [40] and [57]):

Theorem 1 (Four Point Condition)
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 tree with positive lengths on internal edges and non-negative lengths on leaf edges representing the additive distance matrix is unique among the trees without vertices of degree two.

Given a pair of parameters (T, p) for the Neyman 2-state model, and sequences of length k generated by the model, let $H(i, j)$ denote the *Hamming distance* of sequences i and j and

[1]These "distances" may not satisfy the triangle inequality; this is therefore an abuse of the terminology, but is motivated by the observation that the distance corrections used in phylogenetics (and described below) do not always satisfy the triangle inequality. Since it is nevertheless the practice in systematics to refer to these quantities as "distances", we will do so here as well.

$h^{ij} = H(i,j)/k$ denote the *dissimilarity score* of sequences i and j. The *empirical corrected distance* between i and j is denoted by

$$d_{ij} = -\frac{1}{2}\log(1-2h^{ij}). \tag{1}$$

The probability of a change in the state of any fixed character between the sequences i and j is denoted by $E^{ij} = \mathbb{E}(h^{ij})$, and we let

$$D_{ij} = -\frac{1}{2}\log(1-2E^{ij}), \tag{2}$$

denote the *corrected model distance* between i and j. We assign to any edge e a positive length

$$l(e) = -\frac{1}{2}\log(1-2p(e)). \tag{3}$$

By Lemma 1, D_{ij} is the sum of the lengths (see previous equation) along the path $P(i,j)$ between i and j, and hence D_{ij} is an additive distance matrix. Furthermore, d_{ij} converges in probability to D_{ij} as the sequence length tends to infinity. These mathematical facts also have significance for biology, since under certain continuous time Markov models [43], which may be used to justify our models, $l(e)$ and D_{ij} are the expected number of back-and-forth state changes along edges and paths, respectively. A similar phenomenon and hence a similar distance correction exists for the general stochastic model [42], and is discussed in detail in Section 7.

(VI) Tree reconstruction. A *phylogenetic tree reconstruction method* is a function Φ that associates either a tree or the statement *Fail* to every collection of aligned sequences, the latter indicating that the method is unable to make such a selection for the data given. Some methods are based upon sequences, while others are based upon distances.

According to the practice in systematic biology (see, for example, [28, 29, 47]), a method is considered to be *accurate* if it recovers the unrooted binary tree T, even if it does not provide any estimate of the mutation probabilities. A necessary condition for accuracy, under the models discussed above, is that two distinct trees, T, T', do not produce the same distribution of patterns no matter how the trees are rooted, and no matter what their underlying Markov parameters are. This "identifiability" condition is violated under an extension of the *i.i.d.* Markov model when there is an unknown distribution of rates across sites as described by Steel *et al.* [44]. However, it is shown in Steel [42] (see also Chang [13]) that the identifiability condition holds for the *i.i.d* model under the weak conditions that the components of π are not zero and the determinant $\det(M(e)) \neq 0, 1, -1$, and in fact we can recover the underlying tree from the expected frequencies of patterns on just *pairs* of species.

Theorem 1 and the discussion that follows it suggest that appropriate methods applied to corrected distances will recover the correct tree topology from sufficiently long sequences. Consequently, one approach (which is guaranteed to yield a *statistically consistent* estimate) to reconstructing trees from distances is to seek an additive distance matrix of minimum

distance (with respect to some metric on distance matrices) from the input distance matrix. Many metrics have been considered, but all resultant optimization problems have been shown or are assumed to be NP-hard (see [1, 15, 22] for results on such problems).

(VII) Specific tree construction algorithms. In this paper, we will be particularly interested in certain distance methods, the *Four Point Method* (FPM), the *Naive Method*, *Neighbor Joining*, and the *Agarwala et al.* algorithm. We now describe these methods.

> **Four-Point Method (FPM)** Given a 4×4 distance matrix d, return the split $ij|kl$ which satisfies $d_{ij} + d_{kl} < \min\{d_{ik} + d_{jl}, d_{il} + d_{jk}\}$. If there is no such split, return *Fail*.

FPM is a tree reconstruction method in a narrow sense since it applies to four leaves only.

The **Naive Method** uses the Four-Point Method to infer a split for every quartet i, j, k, l. Thus, if the the matrix is additive, the Four Point Method can be used to detect the valid quartet split on every quartet of vertices, and then standard algorithms [6, 14] can be used to reconstruct the tree from the set of splits. Note that the Naive Method is guaranteed to be accurate when the input distance matrix is *additive*, but it will also be accurate even from non-additive distance matrices under conditions which we will describe later (see Section 3). Most quartet-based methods (see, for example, [7, 45, 46]) begin in the same way, constructing a split for every quartet, and then accommodate possible inconsistencies using some technique specific to the method; the Naive Method, by contrast, only returns a tree if all inferred splits are consistent with that tree. The obvious optimization problem (find a maximum number of quartets which are simultaneously realizable) is of unknown computational complexity.

The **Agarwala et al.** algorithm [1] is a 3-approximation algorithm for the nearest tree with respect to the L_∞-metric, where $L_\infty(A, B) = \max_{ij} |A_{ij} - B_{ij}|$. Given input d, the result of applying the Agarwala *et al.* algorithm to d is an additive distance matrix D such that $L_\infty(d, D) \leq 3L_\infty(d, D^{opt})$, where D^{opt} is an optimal solution.

The use of the Agarwala *et al.* algorithm for inferring trees has been studied in two papers (see [21] for a study of its use for inferring trees under the Neyman model, and [4] for a study of its use for inferring trees under the general Markov model). However, both [21] and [4] consider the performance of the Agarwala *et al.* algorithm with respect to the *variational distance* metric. Optimizing with respect to this metric is related to - but distinct from - estimating the tree T, since it is concerned as well with the mutational parameters p.

The **Neighbor Joining** method [38] is a method for reconstructing trees from distance matrices, which is based upon agglomerative clustering. It is possibly the most popular method among molecular biologists for reconstructing trees, and does surprisingly well in some experimental studies; see, for example, [31, 32].

All these methods are known to be statistically consistent for inferring trees both under the Neyman 2-state model and under the general r-state Markov model of site evolution.

3 A framework for the comparison of distance-based methods

Although it is understood that all reasonable distance-based methods will converge on the true tree given sequences of adequate length, understanding the rate of convergence (as a function of sequence length) to the true topology is more complicated. However, it is possible sometimes to compare different distance-based methods, without reference to the underlying model. The purpose of this section is to provide a framework for an explicit comparison among different distance-based methods. We will use this technique to compare the 3-approximation algorithm of Agarwala *et al.* to the *Naive Method.* Our analysis of these two algorithms shows that on any distance matrix for which the first algorithm is guaranteed to reconstruct the true tree, so is the Naive Method. Since our new method, WAM, outperforms the Naive Method on every input, this analysis also establishes a comparison between the Agarwala *et al.* algorithm and WAM.

By the Four Point Condition (Theorem 1) every additive distance matrix corresponds to a unique tree without vertices of degree 2, and with positive internal edge lengths and non-negative leaf edge lengths.

Definition 1: Let $\mathcal{M}$ denote the class of additive distance matrices that correspond to binary trees. The *δ-neighborhood* around the distance matrix d, denoted $N(d, \delta)$, is the set of all distance matrices d' such that $L_\infty(d, d') < \delta$. A distance-based method Φ is said to be *combinatorially consistent* if $\Phi(D) = D$ for all $D \in \mathcal{M}$. A combinatorially consistent method Φ is *continuous at* $D \in \mathcal{M}$ if for every $\epsilon > 0$ there exists a $\delta > 0$ such that if $d \in N(D, \delta)$ then $\Phi(d) \in N(D, \epsilon)$.

Let T be the binary model tree with additive distance matrix D. Let d be the observed distance matrix. Furthermore let $\Delta = L_\infty(d, D)$. Finally, let x be the minimum edge-weight among internal edges in T under the correspondence with D. For every distance-based reconstruction method Φ, we seek a constant $c(\Phi)$ such that

$$c(\Phi) = \inf \{c : \Delta < cx \Longrightarrow \Phi(d) \text{ yields } T\}.$$

Lemma 2 (i) *Two additive distance matrices D and D' define the same topology if and only if for all quartets the relative orders of the pairwise sums of distances for that quartet are identical in the two matrices.*
(ii) *For every distance-based method Φ continuous at $D \in \mathcal{M}$, there is a $\delta > 0$ such that Φ is guaranteed to reconstruct the topology of T when applied to any $d \in N(D, \delta)$.*
(iii) *However, for every edge-weighted binary tree T with minimum internal edge weight x, and any $\vartheta > 0$, there is a different binary tree T' such that $L_\infty(D, D') = x/2 + \vartheta$, where D' is the additive distance matrix for T'.*
(iv) *Given any $n \times n$ distance matrix d, four indices i, j, k, l in $[n]$, let p_{ijkl} denote the difference between the maximum and the median of the three pairwise sums, $d_{ij} + d_{kl}, d_{ik} + d_{jl}, d_{il} + d_{jk}$. Let P be the maximum of the p_{ijkl} over all quartets i, j, k, l. Then there is no additive distance matrix D such that $L_\infty(d, D) < P/4$.*

Proof. Claim (i) is a direct consequence of the Four Point Condition (Theorem 1).
To prove (ii), put $\epsilon = x/2$, and set δ for ϵ in the definition of continuity at D, and assume $\Phi(d) = D'$ and T' is the tree corresponding to D'. We will show using Part (i) that T' has the same topology as T. Suppose that in T the quartet $\{i, j, k.l\}$ induces the topology $ij|kl$. Then $D_{ij} + D_{kl} + 2t = D_{ik} + D_{jl} = D_{il} + D_{jk}$, where t is the path length of the middle path segment. Applying $L_\infty(D, D') < \epsilon$ twice, we obtain $D'_{ik} + D'_{jl} > D_{ik} + D_{jl} - 2\epsilon = D_{ij} + D_{kl} + 2t - 2\epsilon > (D'_{ij} + D'_{kl} - 2\epsilon) + 2t - 2\epsilon \geq D'_{ij} + D'_{kl}$ whenever $t \geq 2\epsilon = x$. Hence, the relative order of distance pair sums is the same.
To prove (iii), for a given T, contract an internal edge e having minimum edge weight x, obtaining a non-binary tree T'. T' has exactly one vertex adjacent to four edges. Add $x/4$ to the weight of each of the four edges. Insert a new edge of weight ϑ to resolve the vertex of degree four, so that we obtain a binary tree T'', different from T. Let D be the additive distance matrix for T and let D'' be the additive distance matrix for T''. It is easy to see that then $L_\infty(D, D') = x/2 + \vartheta$.
For the proof of (iv), let D be an additive distance matrix with $L_\infty(d, D) = \epsilon < t/4$. For all quartets i, j, k, l, the median and the maximum of the three pairwise sums induced by i, j, k, l are identical in D. Now consider the quartet i, j, k, l for which $p_{ijkl} = t$. The maximum and the median of the three pairwise sums in d differ by p_{ijkl}. In order for the maximum and median of the three pairwise sums to be equal in D, at least one pairwise distance must change by at least $p_{ijkl}/4$. However $\epsilon < p_{ijkl}/4$, contradicting the assumption. □

Theorem 2 *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 Naive 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.*

Proof. To prove (i), assume that D^* is an additive distance matrix with $L_\infty(d, D^*) \leq \delta$, and let T^* denote the tree topology corresponding to D^*. According to Lemma 2 Part (i), D^* and D define the same tree iff the relative order of pairwise sums of distances agree for all quartets in the two matrices. Assume without loss of generality that $2P + D_{ij} + D_{kl} = D_{ik} + D_{jl}$ and $D^*_{ij} + D^*_{kl} = D^*_{ik} + D^*_{jl} + 2\epsilon$; these formulae hypothesize that the relative order of pairwise sums of distances over the quartet $\{i, j, k, l\}$ switches between D and D^*. Now $\epsilon > 0$ and $P \geq x$, since P is an internal path length in T. By the triangle inequality we have

$$L_\infty(D, D^*) \leq 2\delta. \tag{4}$$

We have

$$2P + 2\epsilon = D_{ik} + D_{jl} - D_{ij} - D_{kl} + D^*_{ij} + D^*_{kl} - D^*_{ik} - D^*_{jl} \tag{5}$$

and hence by the triangle inequality

$$2x < 2P + 2\epsilon \leq 8\delta \tag{6}$$

as required.

To prove (ii a), let D^* denote the output of the 3-approximation algorithm and T^* denote the corresponding tree. Following similar arguments, $L_\infty(d, D^*) \leq 3\delta$, so that corresponding to Formula (4) we have $L_\infty(D, D^*) \leq 4\delta$, and corresponding to Formula (6) we have $2x < 16\delta$. To prove (ii b), we now give an example where the 3-approximation algorithm can fail in which $L_\infty(D, d) = x/6$. Let d be distance matrix defined by $d_{uv} = d_{wx} = 7/3, d_{uw} = d_{vx} = 3$ and $d_{ux} = d_{vw} = 10/3$. By item (iv) of Lemma 2, it follows that there is no additive distance matrix D with $L_\infty(d, D) < 1/6$. Now let D be the additive distance matrix induced by the binary tree T on leaves u, v, w, x with topology $uv|wx$ and with edge length as follows: the central edge in T has weight 1 and all other edges have weight 13/12. Then, $L_\infty(D, d) = 1/6$ so that D is a closest additive distance matrix to d. Furthermore, $L_\infty(d, D) = x/6$, since $x = 1$ is the lowest edge weight in T. However there is another additive distance matrix induced by a different tree which lies within 3 times this minimal distance. Namely, let D'' be the additive distance matrix induced by the binary tree with topology $uw|vx$ with interior edge weighted 1/3 and other edges weighted 5/4. Then, $L_\infty(D'', d) = 1/2 = 3L_\infty(D, d) = 3\min_D\{L_\infty(D, d)\}$, as claimed. It is easy to see that this example can be embedded in any size distance matrix so that for all n such examples exist. For (ii c), suppose d is a distance matrix, D is its closest additive distance matrix, and x is the smallest weight of any edge in D. Then contract the edge e of weight x in T, the edge-weighted realization of D, and add $x/4$ to every edge originally incident to e. Let D' be the distance matrix of the new edge-weighted tree, T'. It follows that $L_\infty(D, D') = x/2$ and so that $L_\infty(d, D') \leq L_\infty(d, D) + L_\infty(D, D')$. If $L_\infty(d, D) = x/4$, then $L_\infty(d, D') \leq 3x/4$, by the triangle inequality. Hence the 3-approximation algorithm could return the topology of T or of T', and since they are different there is a possibility of making the wrong choice.

To prove (iii), arguments similar to the ones above obtain

$$2P + 2\epsilon = D_{ik} + D_{jl} - D_{ij} - D_{kl} + d_{ij} + d_{kl} - d_{ik} - d_{jl}$$

and $2x < 2P + 2\epsilon \leq 4\delta$. The required example is in Lemma 2 Part (iii). □

In other words, *given any matrix d of corrected distances, if an exact algorithm for the L_∞-nearest tree can be guaranteed - by this analysis - to correctly reconstruct the topology of the model tree, then so can the Naive Method.* This may suggest that there is an inherent limitation of the L_∞-nearest tree approach to reconstructing phylogenetic tree topologies. However, note that the analytical results are pessimistic; that is, they guarantee a high probability of an accurate performance once sequence lengths exceed some threshold, but do not guarantee a low probability of accurate performance for sequences below those lengths. Even so, these techniques are essentially the same ones that have been used in other studies to obtain analytical results regarding convergence to the true tree (see also [4, 21]).

4 The Witness-Antiwitness Tree Construction (WATC)

4.1 Introduction

In this section we describe the Witness-Antiwitness Tree Construction algorithm (WATC). This procedure, which is the heart of our Witness-Antiwitness Method (WAM), solves certain restricted instances of the NP-complete Quartet Consistency Problem [41], and solves them faster than the Dyadic Closure Tree Construction algorithm (DCTC) that we used as a procedure previously in our Dyadic Closure Method (DCM) [19]. We therefore achieve an improvement with respect to computational requirements over DCM, and pay for it by requiring somewhat longer sequences.

Let e be an edge in T. Deleting e but not its endpoints creates two rooted subtrees, T_1 and T_2; these are called *edi-subtrees*, where "edi" stands for "*e*dge-*d*eletion-*i*nduced". Each edi-subtree having at least two leaves can be seen as being composed of two smaller edi-subtrees. The algorithm we will describe, the Witness-Antiwitness Tree Construction algorithm, or WATC, constructs the tree "from the outside in", by inferring larger and larger edi-subtrees, until the entire tree is defined. Thus, the algorithm has to decide at each iteration at least one pair of edi-subtrees to "join" into a new edi-subtree. In the tree, such pairs can be recognized by the constraints (a) that they are disjoint, and (b) that their roots are at distance two from each other. These pairs of edi-subtrees are then said to be "siblings". The algorithm determines whether a pair of edi-subtrees are siblings by using the quartet splits. We will show that if the set Q satisfies certain conditions then WATC is guaranteed to reconstruct the tree T from Q.

The conditions that Q must satisfy in order for WATC to be guaranteed to reconstruct the tree T are slightly more restrictive than those we required in the DCTC method, but do not require significantly longer sequences. Sets Q which satisfy these conditions are said to be *T-forcing*. The first stage of WATC assumes that Q is *T-forcing*, and on that basis attempts to reconstruct the tree T. If during the course of the algorithm it can be determined that Q is *not* T-forcing, then the algorithm returns *Fail*. Otherwise, a tree T' is constructed. At this point, the second stage of WATC begins, in which we determine whether T is the unique tree that is consistent with Q. If Q fails this test, then the algorithm returns *Fail*, and otherwise it returns T.

Just as in the Dyadic Closure Method (DCM) we will need a search technique to find an appropriate set Q. Whereas binary search was a feasible technique for the DCM, it is no longer feasible in this case. Search techniques for an appropriate set Q are discussed in Section 5.

4.2 Definitions and preliminary material

Within each *edi*-subtree t, select that unique leaf which is the lowest valued leaf among those closest topologically to the root (recall that leaves are identified with positive integers). This is called the *representative* of t, and is denoted $rep(t)$. If the edi-subtree consists of a single leaf, then the representative leaf is identical with this single leaf, which also happens to be the root of the edi-subtree at the same time.

The *diameter* of the tree T, $diam(T)$, is the maximum topological distance in the tree between any pair of leaves. We define the depth of an edi-subtree t to be $L(root(t), rep(t))$, and denote this quantity by $depth(t)$. The *depth* of T is then $\max_t\{depth(t)\}$, as t ranges over all edi-subtrees yielded by internal edges of T. We say that a path P in the tree T is *short* if its topological length is at most $depth(T) + 1$, and say that a quartet i, j, k, l is a *short quartet* if it induces a subtree which contains a single edge connected to four disjoint short paths. The set of all short quartets of the tree T is denoted by $Q_{short}(T)$. We will denote the set of valid quartet splits for the short quartets by $Q^*_{short}(T)$.

For each of the $n-3$ internal edges of the n-leaf binary tree T we assign a *representative quartet* $\{i, j, k, l\}$ as follows. The deletion of the internal edge *and* its endpoints defines four rooted subtrees. Pick the representative from each of these subtrees to obtain i, j, k, l; by definition, the quartet i, j, k, l is a *short quartet* in the tree. We call the split of this quartet a *representative quartet split* of T, and we denote the set of representative quartet splits of T by R_T. Note that by definition

$$R_T \subseteq Q^*_{short}(T) \subseteq Q(T). \tag{7}$$

We will say that a set Q of quartet splits is *consistent* with a tree T if $Q \subseteq Q(T)$. We will say that Q is *consistent* if there exists a tree T with which Q is consistent, and otherwise Q is said to be *inconsistent*. In [19], we proved:

Theorem 3 *Let T be a binary tree on $[n]$. If R_T is consistent with a binary tree T' on $[n]$, then $T = T'$. Therefore, if $R_T \subseteq Q$, then either Q is inconsistent, or Q is consistent with T. Furthermore, Q cannot be consistent with two distinct trees if $R_T \subseteq Q$.*□

Let S be a set of n sequences generated under the Neyman model of evolution, and let d be the matrix of corrected empirical distances. Given any four sequences i, j, k, l from S, we define the *width* of the quartet on i, j, k, l to be $\max(d_{ij} + d_{ik}, d_{il} + d_{jk}, d_{jl} + d_{kl})$. For any $w \in R^+$, let Q_w denote the set of quartet splits of width at most w, inferred using the Four-Point Method.

4.3 The Dyadic Closure Method

The Dyadic Closure Method is based on the Dyadic Closure Tree Construction (DCTC) algorithm, which uses dyadic closure (see [19, 17]) to reconstruct a tree T consistent with

an input set Q of quartet splits. The *dyadic closure* of a set Q is denoted by $cl(Q)$, and consists of all splits that can be inferred by combining two splits at a time from Q, and from previously inferred quartet splits. In [19], we showed that the dyadic closure $cl(Q)$ could be computed in $O(n^5)$ time, and that if Q contained all the representative quartet splits of a tree, and contained only valid quartet splits, (i.e. if $R_T \subseteq Q \subseteq Q(T)$), then $cl(Q) = Q(T)$. Consequently, the DCTC algorithm reconstructs the tree T if $R_T \subseteq Q \subseteq Q(T)$. It is also easy to see that no set Q can simultaneously satisfy this condition for two distinct binary trees T, T', by Theorem 3, and furthermore, if Q satisfies this condition for T, it can be quickly verified that T is the unique solution to the reconstruction problem. Thus, when Q is such that for some binary tree T, $R_T \subseteq Q \subseteq Q(T)$, then the DCTC algorithm properly reconstructs T. The problem cases are when Q does not satisfy this condition for any T.

We handle the problem cases by specifying the output $DCTC(Q)$ to be as follows:

- *binary tree* T such that $cl(Q) = Q(T)$ (this type of output is guaranteed when $R_T \subseteq Q \subseteq Q(T)$)
- *Inconsistent* when $cl(Q)$ contains two contradictory splits for the same quartet, or
- *Insufficient* otherwise.

Note that this specification does not prohibit the algorithm from reconstructing a binary tree T, even if Q does not contain all of R_T. In such a case, the tree T will nevertheless satisfy $cl(Q) = Q(T)$; therefore, no other binary tree T' will satisfy $Q \subseteq Q(T'))$. Note that if $DCTC(Q) = Inconsistent$, then $Q \nsubseteq Q(T)$ for any binary tree T, so that if $Q \subseteq Q'$ then $DCTC(Q') = Inconsistent$ as well. On the other hand, if $DCTC(Q) = Insufficient$ and $Q' \subseteq Q$, then $DCTC(Q') = Insufficient$ also. Thus, if $DCTC(Q)$ is Inconsistent, then there is no tree T consistent with Q, but if $DCTC(Q)$ is Insufficient, then it is still possible that some tree exists consistent with Q, but the set Q is *insufficient* with respect to the requirements of the DCTC method.

Now consider what happens if we let Q be Q_w the set of quartet splits based upon quartets of width at most w. The output of the DCTC algorithm will indicate whether w is too big (i.e. when $DCTC(Q_w) = Inconsistent$), or too small (i.e. when $DCTC(Q_w) = Insufficient$). Consequently, DCTC can be used as part of a tree construction method, where splits of quartets (of some specified width w) are estimated using some specified method, and we search through the possible widths w using binary search.

In [19], we studied a specific variant of this approach, called the Dyadic Closure Method (DCM), in which quartet trees are estimated using the Four-Point Method (see Definition VII in Section 2). We analyzed the sequence length that suffices for accurate tree construction by DCM and showed that it grows very slowly; for almost all trees under two distributions on binary trees the sequence length that suffices for tree reconstruction under DCM is only polylogarithmic in n, once $0 < f \leq g < .5$ are fixed and $p(e) \in [f, g]$ is assumed. Thus, DCM has a very fast convergence rate. DCM uses $O(n^2 k + n^5 \log n)$ time and $O(n^4)$ space; therefore

it is a statistically consistent polynomial time method for inferring trees under the Neyman model of evolution. For practical purposes, however, the computational requirements of the DCM method are excessive for inferring large trees, where n can be on the order of hundreds.

4.4 Witnesses, antiwitnesses, and T-forcing sets

Recall that the Witness-Antiwitness Tree Construction Algorithm constructs T from the outside in, by determining in each iteration which pairs of edi-subtrees are siblings. This is accomplished by using the quartet splits to guide the inference of edi-subtrees. We now describe precisely how this is accomplished.

Definition 2: Recall that an *edi-subtree* is a subtree of T induced by the deletion of an edge in the tree. Two edi-subtrees are *siblings* if they are disjoint, the path between their roots contains exactly two edges, and there are at least two leaves not in either of these two edi-subtrees[2]. Let t_1 and t_2 be two vertex disjoint *edi*-subtrees. A *witness to the siblinghood of* t_1 *and* t_2 is a quartet split $uv|wx$ such that $u \in t_1$, $v \in t_2$, and $\{w, x\} \cap (t_1 \cup t_2) = \emptyset$. We call such quartets *witnesses.* An *anti-witness to the siblinghood of* t_1 *and* t_2 is a quartet split $pq|rs$, such that $p \in t_1$, $r \in t_2$, and $\{q, s\} \cap (t_1 \cup t_2) = \emptyset$. We will call these *anti-witnesses.*

Definition 3: Let T be a binary tree and Q a set of quartet splits defined on the leaves of T.

- Q has the **witness property for** T**:** Whenever t_1 and t_2 are sibling edi-subtrees of T and $T - t_1 - t_2$ has at least two leaves, then there is a quartet split of Q which is a witness to the siblinghood of t_1 and t_2.
- Q has the **antiwitness property for** T**:** Whenever there is a witness in Q to the siblinghood of two edi-subtrees t_1 and t_2 which are not siblings in T, then there is a quartet split in Q which is an antiwitness to the siblinghood of t_1 and t_2.

Theorem 4 *If* $R_T \subseteq Q$*, then* Q *has the witness property for* T*. Furthermore, if* $R_T \subseteq Q \subseteq Q(T)$*, and* t_1 *and* t_2 *are sibling edi-subtrees, then* Q *contains at least one witness, but no antiwitness, to the siblinghood of* t_1 *and* t_2*.* □

The proof is straightforward, and is omitted.

Suppose T is a fixed binary tree, and Q is a set of quartet splits defined on the leaves of T. The problem of reconstructing T from Q is in general NP-hard [41], but in [19] we showed that if $R_T \subseteq Q \subseteq Q(T)$ we can reconstruct T in $O(n^5)$ time, and validate that T is the unique tree consistent with Q. Now we define a stronger property for Q which, when it

[2]The last condition - that there are at least two leaves not in either of the two edi-subtrees - is nonstandard, but is assumed because it simplifies our discussion.

holds, will allow us to reconstruct T from Q (and validate that T is the unique tree consistent with Q) in $O(n^2 + |Q| \log |Q|)$ time. Thus, this is a significantly faster algorithm than the DCTC algorithm that we presented in [19].

Definition 4: T-forcing sets of quartet splits: A set Q of quartet splits is said to be *T-forcing* if there exists a binary tree T such that

1. $R_T \subseteq Q \subseteq Q(T)$, and
2. Q has the antiwitness property for T.

Two points should be made about this definition. Since $R_T \subseteq Q$, Q has the *witness* property for T, and it is impossible for Q to be both T-forcing and T'-forcing for distinct T and T', since by Theorem 3, R_T is consistent with a unique tree. Finally, note that the first condition $R_T \subseteq Q \subseteq Q(T)$ was the requirement we made for the Dyadic Closure Tree Construction (DCTC) algorithm in [19], and so T-forcing sets of quartet splits have to satisfy the assumptions of the DCTC algorithm, plus one additional assumption: having the *antiwitness* property.

4.5 WATC

The algorithm we will now describe operates by constructing the tree from the outside in, via a sequence of iterations. Each iteration involves determining a new set of edi-subtrees, where each edi-subtree is either an edi-subtree in the previous iteration or is the result of making two edi-subtrees from the previous iteration siblings. Thus, each iteration involves determining which pairs of edi-subtrees from the previous iteration are siblings, and hence should be joined into one edi-subtree in this iteration.

We make the determination of siblinghood of edi-subtrees by applying the witness and antiwitness properties, but we note that only certain splits are considered to be relevant to this determination. In other words, we will require that any split used either as a witness or an anti-witness have leaves in four distinct edi-subtrees that exist at the time of the determination of siblinghood for this particular pair. Such splits are considered to be *active*, and other splits are considered to be *inactive*. All splits begin as active, but become inactive during the course of the algorithm (and once inactive, they remain inactive). We will use the terms "active witness" and "active antiwitness" to refer to active splits which are used as witnesses and antiwitesses. We will infer that two edi-subtrees are siblings if and only if there is an active witness to their siblinghood and no active anti-witness. (Note that this inference will be accurate if Q has the witness and antiwitness properties, but otherwise the algorithm may make a false inference, or fail to make any inference.)

We represent our determination of siblinghood as a graph on the edi-subtrees we have currently found. Thus, suppose at the beginning of the current iteration there are p edi-subtrees, $t_1, t_2, \ldots, t_p$. The graph for this iteration has p nodes, one for each edi-subtree, and

we put an edge between every pair of edi-subtrees which have at least one witness and no anti-witness in the set of quartet topologies. The algorithm proceeds by then merging pairs of sibling edi-subtrees (recognized by edges in the graph) into a single (new) edi-subtree. The next iteration of the algorithm then requires that the graph is reconstructed, since witnesses and antiwitnesses must consist of four leaves, each drawn from distinct edi-subtrees (these are the *active* witnesses and antiwitnesses – thus, quartet splits begin as active, but can become inactive as edi-subtrees are merged).

The last iteration of the algorithm occurs when the number of edi-subtrees left is four, *or* there are no pairs of edi-subtrees which satisfy the conditions for siblinghood. If no pair of edi-subtrees satisfy the criteria for being siblings, then the algorithm returns *Fail*. On the other hand, if there are exactly four edi-subtrees, and if there are two disjoint pairs of sibling edi-subtrees, then we return the tree formed by merging each of the two pairs of sibling edi-subtrees into a single edi-subtree, and then joining the roots of these two (new) edi-subtrees by an edge.

If a tree T' is reconstructed by the algorithm, we will not return T' until we verify that

$$R_{T'} \subseteq Q \subseteq Q(T').$$

If the tree T' passes this test, then we return T', and in all other cases we return *Fail*.

We summarize this discussion in the following:

The WATC algorithm

Stage I:

♢ Start with every leaf of T defining an *edi*-subtree.

♢ While there are at least four *edi*-subtrees do:

- Form the graph G on vertex set given by the edi-subtrees, and with edge set defined by siblinghood; i.e., $(x, y) \in E(G)$ if and only if there is at least one witness and no antiwitness to the siblinghood of edi-subtrees x and y. All witnesses and antiwitnesses must be splits on four leaves in which each leaf lies in a distinct edi-subtree; these are the *active* witnesses and antiwitnesses.
  - **Case: there are exactly four edi-subtrees:**
    Let the four subtrees be x, y, z, w. If the edge set of the graph G is $\{(x, y), (z, w)\}$, then construct the tree T formed by making the edi-subtrees x and y siblings, the edi-subtrees z and w siblings, and adding an edge between the roots of the two new edi-subtrees; else, return *Fail*.
  - **Case: there are more than four edi-subtrees:**
    If the graph has at least one edge, then select one, say (x, y), and make the roots

of the *edi*-subtrees x and y children of a common root r, and replace the pair x and y by one *edi*-subtree. If no component edge exists, then Return *Fail*.

Stage II:

$\diamondsuit$ Verify that T satisfies the constraints $R_T \subseteq Q \subseteq Q(T)$. If so, return T, and else return *Fail*.

The runtime of this algorithm depends upon how the two *edi*-subtrees are found that can be siblings.

4.6 Implementation of WATC

We describe here a fast implementation of the WATC algorithm.

We begin by constructing a multigraph on n nodes, bijectively labelled by the species. Edges in this multigraph will be colored either green or red, with one green edge between i and j for each witness to the siblinghood of i and j, and one red edge between i and j for each antiwitness. Thus, each quartet split $ij|kl$ defines six edges in the multi-graph, with two green edges ((ij) and (kl)) and four red edges ($(ik), (il), (jk), (jl)$). Each green edge is annotated with the quartet that defined it and the topology on that quartet, so that the other edges associated to that quartet can be identified. Constructing this multi-graph takes $O(|Q|)$ time. Note that *edi*-subtrees x and y are determined to be siblings if there exists a green edge (x, y) but no red edge (x, y).

We will maintain several data structures:

- $Red(i, j)$, the number of red edges between nodes i and j, so that accesses, increments, and decrements to $Red(i, j)$ take $O(1)$ time,
- $Green(i, j)$, the set of green edges between nodes i and j, maintained in such a way that we can enumerate the elements in $|Green(i, j)|$ time, and so that we can union two such sets in $O(1)$ time,
- T_i, the i^{th} edi-subtree (i.e. the edi-subtree corresponding to node i), maintained as a directed graph with edges directed away from the root,
- $Tree$, an array such that $Tree[i] = j$ indicates that leaf i is in tree T_j. This is initialized by $Tree[i] = i$ for all i, and
- *Candidates*, the set of pairs of edi-subtrees which have at least one green edge and no red edges between them (and hence are candidates for siblinghood). We maintain this set using doubly-linked lists, and we also have pointers *into* the list from other datastructures ($Green(i, j)$) so that we can access, add, and delete elements from the set in $O(1)$ time.

Finding a sibling pair A pair of edi-subtrees are inferred to be siblings if and only if they have at least one green edges and no red edges between them. We maintain a list of possible sibling pairs of edi-subtrees in the set *Candidates*, and the members of *Candidates* are pairs of the form i, j where both i and j are edi-subtrees. (Testing whether i is a current edi-subtree is easy; just check that $Tree[i] = i$.) We take an element (i, j) from the set *Candidates* and verify that the pair is valid. This requires verifying that both i and j are current names for *edi*-subtrees, which can be accomplished by checking that $Tree[i] = i$ and $Tree[j] = j$. If (i, j) fails this test, we delete (i, j) from the set of *Candidates*, and examine instead a different pair. However, if (i, j) passes this test, we then verify that the pair i, j have at least one green edge and no red edges between them. For technical reasons (which we describe below), it is possible that $Green(i, j)$ will contain a *ghost* green edge. We now define what ghost green edges are, and how we can recognize them in $O(1)$ time.

Definition 5: A *ghost* green edge is a green edge (a, b) which was defined by a quartet split $ab|cd$, but which was not deleted after the edi-subtrees containing c and d were merged into a single edi-subtree.

Detecting whether a green edge is a ghost is done as follows. Recall that every green edge (a, b) is annotated with the quartet (a, b, c, d) that gave rise to it. Therefore, given a green edge (a, b), we look up the edi-subtrees for the members of the *other* green edge (c, d) (using the $Tree$ array), and see if c and d still belong to distinct edi-subtrees. If $Tree[c] = Tree[d]$ then (a, b) is a ghost green edge (since c and d were already placed in the same edi-subtree) and otherwise it is a true green edge.

Every ghost we find in $Green(i, j)$ we simply delete, and if $Green(i, j)$ contains only ghost edges, we remove (i, j) from the set *Candidates* (the edi-subtrees i and j are not actually siblings). If we find any non-ghost green edge in $Green(i, j)$, then (i, j) are inferred to be sibling edi-subtrees, and we enter the next phase.

Processing a sibling pair Having found a pair i and j of edi-subtrees which are siblings, we need to update all the data-structures appropriately. We now describe how we do this.

First, we process every green edge e in $Green(i, j)$ by deleting the four red edges associated to e (this is accomplished by decrementing appropriate entries in the matrix Red). Note that we do not explicitly (or implicitly) delete the other green edge associated with edge e, and rather leave that green edge to be handled later; this is how *ghost* green edges arise.

After we finish processing every green edge, we merge the two edi-subtrees into one edi-subtree. We will use one index, say i, to indicate the number of the new *edi*-subtree created. We update T_i so that it has a new root, and the children of the new root are the roots of the previous edi-subtrees T_i and T_j, and we update the $Tree$ array so that all entries which previously held a j now hold i.

We also have to reset $Red(i, k)$ and $Green(i, k)$ for every other edi-subtree k, since the edi-

subtree labelled i has changed. We set $Red(i,k) = Red(i,k) + Red(j,k)$, and $Green(i,k) = Green(i,k) \cup Green(j,k)$ for all k. We then set $Red(j,k) = 0$ and $Green(j,k) = \emptyset$, if we wish (this is for safety, but is not really needed).

We also have to update the *Candidates* set. This involves deletions of some pairs, and insertions of others. The only pairs which need to be deleted are those i,k for which there is now a red edge between edi-subtrees i and k, but for which previously there was none. This can be observed during the course of updating the $Red(i,k)$ entries, since every pair (i,k) which should be deleted has $Red(i,k) = 0$ before the update, and $Red(i,k) > 0$ after the update. Pairs (i,k) which must be inserted in the *Candidates* set are those (i,k) which previously had $Green(i,k) = \emptyset$ and which now have $Green(i,k) \neq \emptyset$. Accessing, inserting, and deleting the elements of *Candidates* takes $O(1)$ time each, so this takes $O(1)$ additional time.

We now discuss the runtime analysis of the first stage of WATC:

Theorem 5 *The first stage of WATC uses $O(n^2 + |Q|)$ time.*

Proof. Creating the multi-graph clearly costs only $O(|Q|)$ time. Initializing all the datastructures takes $O(n^2)$ time. There are at most $O(|Q|)$ green edges in the multigraph we create, and each green edge is processed at most once, after which it is deleted. Processing a green edge costs $O(1)$ time, since $Tree$ can be accessed in $O(1)$ time. There are at most $n-1$ siblinghood detections, and updating the datastructures after detecting siblinghood only costs $O(n)$ time (beyond the cost of processing green edges). Implementing the datastructures $Green(i,j)$ and *Candidates* so that updates are efficient is easy through the use of pointers and records. Hence, the total cost of the first stage is $O(n^2 + |Q|)$. □

So suppose the result of the first phase constructs a tree T from the set Q of splits. The second stage of the WATC algorithm needs to verify that $R_T \subseteq Q \subseteq Q(T)$; we now describe how this is accomplished efficiently.

Given T, we can compute R_T in $O(n^2)$ time in a straightforward way: for each of the $O(n)$ edi-subtrees t, we compute the representative $rep(t)$ in $O(n)$ time. We then use the representatives to compute R_T, which has size $O(n)$, in $O(n)$ additional time. Verifying that $R_T \subseteq Q$ then takes at most $O(n \log n + |Q| \log |Q|)$ time. First we make sorted list of quartet splits by the lexicographic order of the 4 vertices involved. Sorting is in $O(|Q| \log |Q|)$ time. Then we use a binary search to determine membership, which costs $O(\log n)$ time for each element of R_T, since $|Q| = O(n^4)$. Verifying that $Q \subseteq Q(T)$ then can be done by verifying that $q \in Q(T)$ for each $q \in Q$. This is easily done in $O(1)$ time per q using $O(1)$ *lca* queries (to determine the valid split for each quartet which has a split in Q). Preprocessing T so that we can do *lca* queries in $O(1)$ time per query can be done in $O(n)$ time, using the algorithm of Harel and Tarjan [48]. Consequently, we have proven:

Theorem 6 *The second stage of WATC takes $O(n^2 + |Q| \log |Q|)$ time. Therefore, WATC takes $O(n^2 + |Q| \log |Q|)$ time.*

4.7 Proof of correctness of WATC

We begin by proving that the WATC algorithm correctly reconstructs the tree T provided that Q is T-forcing.

Theorem 7 *If Q is T-forcing, then $WATC(Q) = T$.*

Proof. If Q is T-forcing, then $R_T \subseteq Q$. Now if t and t' are sibling edi-subtrees, then let e be the edge in T whose deletion disconnects $t \cup t'$ from the rest of the tree T. Let q be the representative quartet split associated to e. This quartet split is a witness to the siblinghood of t and t', which will remain active throughout the iterations of the algorithm until the entire tree is constructed (otherwise there are only three edi-subtrees present at some point, and this is contradicted by the structure of the algorithm). Furthermore, since $Q \subseteq Q(T)$, there is no invalid quartet split, and consequently no antiwitness to the siblinghood of t and t'. Therefore, the algorithm will never fail to have opportunities to merge pairs of sibling edi-subtrees, and the only question is whether any pairs of edi-subtrees which are not actually siblings are incorrectly inferred to be siblings.

We prove that no incorrect decisions are made by the algorithm, by induction on the number of iterations. At the first iteration, every edi-subtree is a leaf, and these are correct. Now assume that so far the WATC algorithm applied to Q has constructed only correct edi-subtrees, and the next step merges two edi-subtrees, t_1 and t_2, into one, but that these are not actually siblings.

Since Q has the antiwitness property, there is an valid quartet split $ab|cd \in Q$ with $a \in t_1, c \in t_2$ and $\{b, d\} \cap (t_1 \cup t_2) = \emptyset$. We need only show that this antiwitness is still active at the time that we merged t_1 and t_2 into one edi-subtree.

Suppose that the split $ab|cd$ is not active at the time we merged t_1 and t_2. In this case, then the four leaves a, b, c, d are in fewer than four distinct edi-subtrees. The assumption $\{b, d\} \cap (t_1 \cup t_2) = \emptyset$ then implies that we have already created an edi-subtree t containing both b and d. This edi-subtree is true, since we have assumed all edi-subtrees constructed so far are accurate. Now, consider the edge e' whose deletion creates the subtree t. This edge cannot exist, if $ab|cd$ is a valid quartet split and neither b nor d are in $t_1 \cup t_2$. Consequently, the antiwitness $ab|cd$ is *still* active at the time we merged t_1 and t_2, contradicting that we made that merger, and hence all inferred edi-subtrees are correct. □

Theorem 8 *If the WATC algorithm returns a tree T given a set Q of quartet splits, then Q is consistent with T and with no other tree T'. If WATC does not return a tree T, then Q is not T-forcing.*

Proof. The proof is not difficult. If T is returned by WATC, then Q satisfies $R_T \subseteq Q \subseteq Q(T)$. Under this condition Q is consistent with T and with no other tree, by Theorem

3. Hence the first assertion holds. For the second assertion, if Q is T-forcing, then by the previous theorem WATC returns T after the first stage. The conditions for being T-forcing include that $R_T \subseteq Q \subseteq Q(T)$, so that the verification step is successful, and Q is returned. □

5 The Witness-Antiwitness Method (WAM)

In the previous section we described the WATC algorithm which reconstructs T given a T-forcing set of quartet splits, Q. In this section we describe a set of search strategies for finding such a set Q. These strategies vary in their number of queries on quartet split sets (ranging from $O(\log \log n)$ to $O(n^2)$), but also vary in the sequence length needed in order for the search strategy to be successful with high probability. All have the same asymptotic sequence length requirement as the Dyadic Closure Method ([19]), but differ in terms of the multiplicative constant.

Before we describe and analyze these search strategies, we begin with some results on the Four-Point Method, and on random trees.

5.1 Previous results

Lemma 3 [Azuma–Hoeffding inequality, see [3]]
Suppose $X = (X_1, X_2, \ldots, X_k)$ are independent random variables taking values in any set S, and $L : S^k \rightarrow \mathbb{R}$ is any function that satisfies the condition: $|L(\mathbf{u}) - L(\mathbf{v})| \leq t$ whenever $\mathbf{u}$ and $\mathbf{v}$ differ at just one coordinate. Then,

$$\begin{aligned} \mathbb{P}[L(\mathbf{X}) - \mathbb{E}[L(\mathbf{X})] \geq \lambda] &\leq \exp\left(-\frac{\lambda^2}{2t^2k}\right), \\ \mathbb{P}[L(\mathbf{X}) - \mathbb{E}[L(\mathbf{X}] \leq -\lambda] &\leq \exp\left(-\frac{\lambda^2}{2t^2k}\right). \qquad \square \end{aligned}$$

In [19], we proved:

Theorem 9 *Assume that z is a lower bound for the transition probability of any edge of a tree T in the Neyman 2-state model, $y \geq \max E^{ij}$ is an upper bound on the compound changing probability over all ij paths in a quartet q of T. The probability that FPM fails to return the correct quartet split on q is at most*

$$18 \exp\left(-(1 - \sqrt{1-2z})^2(1-2y)^2k/8)\right). \square \tag{8}$$

In [19] we also provided an upper bound on the growth of the depth of random trees under two distributions:

Theorem 10 (i) *For a random semilabelled binary tree T with n leaves under the uniform model, $depth(T) \leq (2 + o(1)) \log_2 \log_2 (2n)$ with probability $1 - o(1)$.*
(ii) *For a random semilabelled binary tree T with n leaves under the Yule-Harding distribution, after suppressing the root, $depth(T) = (1 + o(1)) \log_2 \log_2 n$ with probability $1 - o(1)$.* □

5.2 Search strategies

The first step is the computation of the distance matrices d and h, where h^{ij} is the dissimilarity score between i and j, and d_{ij} is based upon an appropriate distance correction. This is accomplished in $O(n^2 k)$ time. Let Q_w denote the set of splits inferred using the Four-Point Method on quartets whose width is at most w; recall that the width of a quartet i, j, k, l is the maximum of $d_{ij}, d_{ik}, d_{il}, d_{jk}, d_{jl}, d_{kl}$. The objective is to find a set Q_w such that Q_w is T-forcing.

Definition 6:

$$\mathcal{A} = \{w \in R^+ : R_T \subseteq Q_w\},$$

$$\mathcal{B} = \{w \in R^+ : Q_w \subseteq Q(T)\}.$$

We now state without proof the following observation which is straightforward.

Observation 1 *$\mathcal{A}$ is either $\emptyset$, or is (w_A, ∞) for some positive real number w_A. $\mathcal{B}$ is either $\emptyset$, or is $(0, w_B)$, for some positive real number w_B.* □

Sequential search for T-forcing Q_w A sequential search through the sets Q_w, testing each Q_w for being T-forcing by a simple application of WATC algorithm, is an obvious solution to the problem of finding a T-forcing set which will find a T-forcing set from shorter sequences than any other search strategy through the sets Q_w. However, in the worst case, it examines $O(n^2)$ sets Q_w, since w can be any of the values in $\{d_{ij} : 1 \leq i < j \leq n\}$, and hence it has high computational requirements.

Sparse-high search for a T-forcing Q_w We describe here a sparse search that examines at most $O(\log k)$ sets Q_w and hence has lower computational requirements, but may require longer sequences. Even so, we prove that the sequence length requirement has the same order of magnitude as the sequential search. This sparse search examines the high end of the values of w, and so we call it the *Sparse-high* search strategy.

Let $\tau < 1/4$ be given. We define Z_τ to be the set of quartets i, j, k, l such that $\max\{h^{ij}, h^{ik}, h^{il}, h^{jk}, h^{jl}, h^{kl}\} < 1/2 - 2\tau$. Note then that the set of splits (inferred using the Four Point Method) on quartets in Z_τ is $Q_{w(\tau)}$, where $w(\tau) = -\frac{1}{2}(\log(4\tau))$.

The sparse-high search examines $\tau = 1/8, 1/16, \ldots$, until it finds a τ such that $Z_\tau = Q_{w(\tau)}$ is T-forcing, or until $w(\tau)$ exceeds every d_{ij}.

We now define conditions under which each of these search strategies are guaranteed to find a T-forcing set Q_w. Recall the sets $\mathcal{A} = \{w : R_T \subseteq Q_w\}$, and $\mathcal{B} = \{w : Q_w \subseteq Q(T)\}$. We now define the following *assumptions:*

$$\mathcal{A} \cap \mathcal{B} \neq \emptyset, \tag{9}$$

$$\exists w^* \in \mathcal{A} \cap \mathcal{B}, \text{ s.t. } Q_{w^*} \text{ has the antiwitness property,} \tag{10}$$

$$\exists \tau^*, \text{ s.t. } \forall \tau \in [\tau^*/2, \tau^*], w(\tau) \in \mathcal{A} \cap \mathcal{B}, \text{ and } Q_{w(\tau)} \text{ has the antiwitness property.} \tag{11}$$

It is clear that if Assumptions (9) and (10) hold, then the sequential search strategy will be guaranteed to succeed in reconstructing the tree, and that the Sparse-high search strategy requires that Assumption (11) hold as well.

We now analyze the sequence length needed to get each of these assumptions to hold with constant probability.

6 How WAM performs under the Neyman 2-state model

In this section we analyse the performance of the Witness-Antiwitness Method (WAM), with respect to computational and sequence-length requirements. The analysis of the sequence length requirement follows a similar analysis for DCM in [19], but turns out to be more complicated, and results in constant times longer sequences. The analysis of the computational complexity of WAM is both in the worst case, and under the assumption that the tree topology is drawn from a random distribution. Finally, we compare the performance of WAM to other methods, with respect to both these issues.

6.1 Sequence length needed by WAM

Theorem 11 *Suppose k sites evolve under the Cavender-Farris model on a binary tree T, so that for all edges e, $p_e \in [f, g]$, where we allow $f = f(n)$ and $g = g(n)$ to be functions of n. We assume that* $\limsup_n g(n) < 1/2$. *Then both the sparse-high and sequential search based on the WATC algorithm returns the true tree T with probability* $1 - o(1)$, *if*

$$k > \frac{c \cdot \log n}{(1 - \sqrt{1 - 2f})^2 (1 - 2g)^{4depth(T)}} \tag{12}$$

where c is a fixed constant.

Proof. Note that the sparse-high search requires Assumptions (9), (10), and (11), while the sequential search only requires Assumptions (9) and (10). We will show that the given sequence length suffices for all three assumptions to hold with probability $1 - o(1)$.

We begin by showing that Assumption (9) holds, i.e. that $R_T \subseteq Q_w \subseteq Q(T)$ for some w.

For k evolving sites (i.e. sequences of length k), and fixed $\tau > 0$, let us define the following two sets,

$$S_\tau = \{\{i,j\} : h^{ij} < 0.5 - \tau\},$$

and

$$Z_\tau = \{q \in \binom{[n]}{4} : \text{ for all } i,j \in q, \{i,j\} \in S_{2\tau}\},$$

and the following four events,

$$A = Q_{short}(T) \subseteq Z_\tau \tag{13}$$

$$B_q = \text{FPM correctly returns the split of the quartet } q \in \tbinom{[n]}{4} \tag{14}$$

$$B = \bigcap_{q \in Z_\tau} B_q \tag{15}$$

$$C = S_{2\tau} \text{ contains all } \{i,j\} \text{ with } E^{ij} < 0.5 - 3\tau \text{ and no } \{i,j\} \text{ with } E^{ij} \geq 0.5 - \tau \tag{16}$$

Note that B is the event that $Q_{w(\tau)} \subseteq Q(T)$, so that $A \cap B$ is the event that $Q^*_{short} \subseteq Q_{w(\tau)} \subseteq Q(T)$, or $w(\tau) \in \mathcal{A} \cap \mathcal{B}$. Thus, $\mathbb{P}[\mathcal{A} \cap \mathcal{B} \neq \emptyset] \geq \mathbb{P}[A \cap B]$. Define

$$\lambda = (1-2g)^{2depth(T)+3}. \tag{17}$$

We claim that:

$$\mathbb{P}[C] \geq 1 - (n^2 - n)e^{-\tau^2 k/2} \tag{18}$$

and

$$\mathbb{P}[A|C] = 1, \quad \text{if} \quad \tau \leq \lambda/6 \ . \tag{19}$$

To establish (18), first note that h^{ij} satisfies the hypothesis of the Azuma-Hoeffding Inequality (Lemma 3 with $X_l = 1$ if the l^{th} bits of the sequences of leaves i and j differ, and $X_l = 0$ otherwise, and $t = 1/k$). Suppose $E^{ij} \geq 0.5 - \tau$. Then,

$$\begin{aligned} & \mathbb{P}[\{i,j\} \in S_{2\tau}] = \mathbb{P}[h^{ij} < 0.5 - 2\tau] \\ \leq \ & \mathbb{P}\left[h^{ij} - E^{ij} \leq 0.5 - 2\tau - E^{ij}\right] \leq \mathbb{P}\left[h^{ij} - \mathbb{E}[h^{ij}] \leq -\tau\right] \leq e^{-\tau^2 k/2}. \end{aligned}$$

Since there are at most $\binom{n}{2}$ pairs $\{i,j\}$, the probability that at least one pair $\{i,j\}$ with $E^{ij} \geq 0.5 - \tau$ lies in $S_{2\tau}$ is at most $\binom{n}{2} e^{-\tau^2 k/2}$. By a similar argument, the probability that $S_{2\tau}$ fails to contain a pair $\{i,j\}$ with $E^{ij} < 0.5 - 3\tau$ is also at most $\binom{n}{2} e^{-\tau^2 k/2}$. These two bounds establish (18).

We now establish (19). For $q \in Q_{short}(T)$ and $i,j \in q$, if a path $e_1 e_2 ... e_t$ joins leaves i and j, then $t \leq 2depth(T)+3$ by the definition of $Q_{short}(T)$. Using these facts, Lemma 1, and the

bound $p_e \leq g$, we obtain $E^{ij} = 0.5\,[1-(1-2p_1)\cdots(1-2p_t)] \leq 0.5(1-\lambda)$. Consequently, $E^{ij} < 0.5 - 3\tau$ (by assumption that $\tau \leq \lambda/6$) and so $\{i,j\} \in S_{2\tau}$ once we condition on the occurence of event C. This holds for all $i,j \in q$, so by definition of Z_τ we have $q \in Z_\tau$. This establishes (19).

Define a set

$$X = \left\{ q \in \binom{[n]}{4} : \max\{E^{ij} : i,j \in q\} < 0.5 - \tau \right\}$$

(note that X is not a random variable, while Z_τ, S_τ are). Now, for $q \in X$, the induced subtree in T has mutation probability at least $f(n)$ on its central edge, and mutation probability of no more than $\max\{E^{ij} : i,j \in q\} < 0.5 - \tau$ on any pendant edge. Then, by Theorem 9 we have:

$$\mathbb{P}[B_q] \geq 1 - 18\exp\left(-(1-\sqrt{1-2f})^2\tau^2 k/8\right) \tag{20}$$

whenever $q \in X$. Also, the occurence of event C implies that

$$Z_\tau \subseteq X \tag{21}$$

since if $q \in Z_\tau$, and $i,j \in q$, then $i,j \in S_{2\tau}$, and then (by event C), $E^{ij} < 0.5 - \tau$, hence $q \in X$. Thus,

$$\mathbb{P}[B \cap C] = \mathbb{P}[(\bigcap_{q \in Z_\tau} B_q) \cap C] \geq \mathbb{P}[(\bigcap_{q \in X} B_q) \cap C]$$

where the second inequality follows from (21), as this shows that when C occurs, $\bigcap_{q\in Z_\tau} B_q \supseteq \bigcap_{q \in X} B_q$. Invoking the Bonferonni inequality, we deduce that

$$\mathbb{P}[B \cap C] \geq 1 - \sum_{q \in X} \mathbb{P}[\overline{B_q}] - \mathbb{P}[\overline{C}]. \tag{22}$$

Thus, from above,

$$\mathbb{P}[A \cap B] \geq \mathbb{P}[A \cap B \cap C] = P[B \cap C]$$

(since $\mathbb{P}[A|C] = 1$), and so, by (20) and (22),

$$\mathbb{P}[A \cap B] \geq 1 - 18\binom{n}{4}\exp\left(-(1-\sqrt{1-2f})^2\tau^2 k/8\right) - (n^2-n)e^{-\tau^2 k/2}.$$

Formula (12) follows by an easy calculation for $\tau = c \cdot \lambda$, for any $0 < c_1 \leq 1/6$.

We proceed to prove that Assumption (10) holds. Recall the definition of $Q_{w(\tau)} = \{FPM(q) : q \in Z_\tau\}$. Recalling Theorem 4, it is obvious that that event $A \cap B \cap D$ implies Assumptions (9, 10). We are going to show that $\mathbb{P}[A \cap B \cap D] = 1 - o(1)$ under the conditions of the theorem for a certain choice of τ, which is just slightly smaller than the τ that sufficed for the Assumption (9). Technically, we are going to show

$$\mathbb{P}[D|A \cap B \cap C] = 1. \tag{23}$$

proof of (23): $\overline{D} = \cup_{t_1,t_2} H_{t_1,t_2}$, where t_1, t_2 denote two disjoint edi-subtrees of T, and H_{t_1,t_2} denotes the event that there is a witness but no antiwitness exists for the siblinghood of t_1, t_2 in $Q_{w(\tau)}$. Therefore in order to prove (23), it suffices to prove:

$$\mathbb{P}[H_{t_1,t_2} | A \cap B \cap C] = 0. \tag{24}$$

Assume that there is a witness for the siblinghood of t_1, t_2 where t_1 and t_2 are *not* siblings. We will show that $Q_{w(\tau)}$ contains an antiwitness to the siblinghood of t_1 and t_2. Let the witness to the siblinghood of t_1 and t_2 be $ab|cd$, where $a \in t_1$, $b \in t_2$, and c, d not in $t_1 \cup t_2$. Let pq be an internal edge of the unique ab path in T containing the midpoint of the path $P(a,b)$ measured using the lengths defined by the corrected model distances D, and with p closer to a and q closer to b, i.e. the edge (p,q) maximizes the following quantity:

$$\min_{pq \ internal \ edge} \left(1 - 2E^{ap}, 1 - 2E^{qb}\right). \tag{25}$$

Let p' and q' be neighbors of p and q respectively that are not on the path between nodes a and b. Consider the *edi*-subtrees t_3 and t_4 rooted at p' and q' respectively, formed by deleting (p,p') and (q,q'), respectively. Set $u = rep(t_3)$, $v = rep(t_4)$.

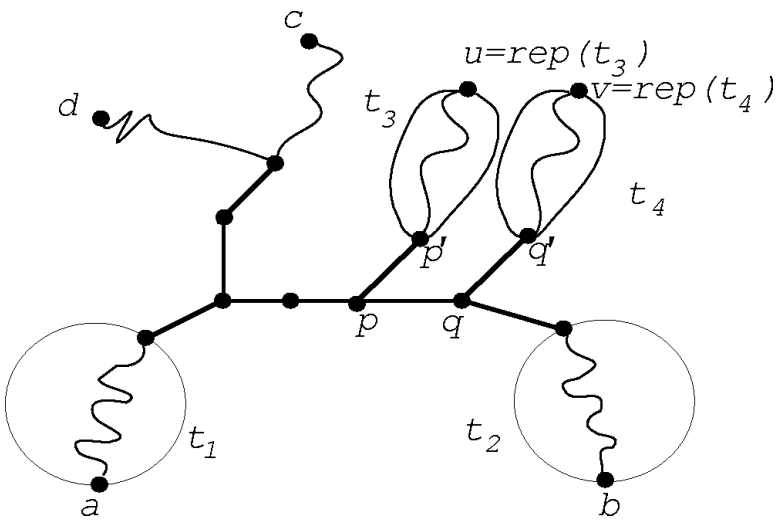

Fig. 1 : Finding an antiwitness.

We are going to show that

$$\{a, b, u, v\} \in Z_\tau, \tag{26}$$

and $au|bv \in Q_{w(\tau)}$. The proof of (26) is the only issue, since by (15) the split of $\{a,b,u,v\}$ is correctly reconstructed, and is $au|bv$ by construction. Clearly

$$\mathbb{P}\Big[H_{t_1,t_2} \Big| A \cap B \cap C\Big] \leq \mathbb{P}\Big[\{a,b,u,v\} \notin Z_\tau \Big| A \cap B \cap C\Big]. \tag{27}$$

The RHS of (27) can be further estimated by

$$\begin{aligned}\mathbb{P}\Big[h^{au} \geq 0.5 - 2\tau \Big| A \cap B \cap C\Big] \quad &+ \quad \mathbb{P}\Big[h^{av} \geq 0.5 - 2\tau \Big| A \cap B \cap C\Big] + \\ \mathbb{P}\Big[h^{bu} \geq 0.5 - 2\tau \Big| A \cap B \cap C\Big] \quad &+ \quad \mathbb{P}\Big[h^{bv} \geq 0.5 - 2\tau \Big| A \cap B \cap C\Big] + \\ &\mathbb{P}\Big[h^{uv} \geq 0.5 - 2\tau \Big| A \cap B \cap C\Big]. \end{aligned} \tag{28}$$

The fifth term $\mathbb{P}[h^{uv} \geq 0.5 - 2\tau | A \cap B \cap C] = 0$, since it is easy to find a short quartet which contains u, v; and therefore by (13), $h^{uv} < 0.5 - 2\tau$. Here is how to find a short quartet containing u and v. Let a' denote the neighbor of p on the ab path towards a, and let q denote the the neighbor of q on the ab path towards b. Consider the edi-subtree t_5 defined by pa', which contains the leaf a, and the edi-subtree t_6 defined by qb', which contains the leaf b. It is easy to check that $\{u, v, rep(t_5), rep(t_6)\}$ is a short quartet.

In order to finish the proof of (24), and hence the proof of (23), it suffices to show that the other four terms in (28) are zero as well. The third and fourth terms are symmetric to the first and second, and in fact the second has a worse bound than the first. Therefore it suffices to prove that

$$\mathbb{P}\Big[h^{av} \geq 0.5 - 2\tau \Big| A \cap B \cap C\Big] = 0. \tag{29}$$

We assume that $\{a, v\} \notin S_{2\tau}$, and show that consequently τ is large. Hence, for a properly small τ, Formula (29), and hence (23) holds. From $\{a, v\} \notin S_{2\tau}$, conditioning on C,

$$E^{av} > 0.5 - 3\tau, \tag{30}$$

and $\{a, b\} \in S_{2\tau}$, and hence, conditioning on C,

$$E^{ab} < 0.5 - \tau. \tag{31}$$

There is no difficulty to extend the definition of E^{ij} to cases when at least one of i, j is an internal vertex of the tree. Simple algebra yields from formula (30) and Lemma 1, that

$$6\tau \geq 1 - 2E^{av} = (1 - 2E^{pv})(1 - 2E^{pa}). \tag{32}$$

We have

$$1 - 2E^{pv} \geq (1 - 2g)^{depth(T)+2} = \sqrt{\lambda(1 - 2g)} \tag{33}$$

by the definition of λ (see Formula (17)) and the choice of v as representative. By formula (25), it is easy to see that

$$1 - 2E^{pa} \geq (1 - 2g)^2\sqrt{1 - 2E^{ab}} \tag{34}$$

Combining (31, 32, 33, and 34), we obtain $6\tau > \sqrt{\lambda(1 - 2g)}(1 - 2g)^2\sqrt{2\tau}$. This formula fails, if we select

$$\tau = c_2 \cdot (1 - 2g)^5\lambda \tag{35}$$

with a sufficiently small positive constant c_2.

Case 1. $p \notin t_1$ and $q \notin t_2$ (as in Fig. 1).
Then $au|bv \in Q_{w(\tau)}$ is an anti-witness, as desired.

When Case 1 does not hold, the only problem that can arise is if the valid split $au|bv$ does not satisfies the condition $\{u, v\} \cap (t_1 \cup t_2) = \emptyset$, and hence is not an antiwitness.

Case 2. $p \in t_1$ or $q \in t_2$.
Without loss of generality we may assume $p \in t_1$. Now we *redefine* the location of the edge pq on the ab path as follows. Let p denote the first vertex after $root(t_1)$ on the ab path and let q denote the second. Clearly $q \notin t_2$, since t_1 and t_2 are not siblings. We also redefine p', q', t_3, u, t_4, v according to the new p and q. Redefine a to be $rep(t_1)$ and call the old a as a^*. Now we are going to show (26) and that $au|bv \in Q_{w(\tau)}$ is the sought-for antiwitness (note a, u, v have been redefined, but b has not). Again, we have to see (27) and prove that (28) is termwise zero.

Now handle the case where $\{u, v\} \in S_{2\tau}$ exactly as in Case 1. Observe that E^{bu} and E^{bv} *decreased* during the redefinition, so a calculation like (29–35) still goes through. Observe that $L(a, u) \leq 2depth(T) + 2$, $L(a, v) \leq 2depth(T) + 3$, and hence $\{a, u\} \in S_{2\tau}$ and $\{a, v\} \in S_{2\tau}$, exactly as in the proof of (19). The only thing left to prove is $\{a, b\} \in S_{2\tau}$.

In order to prove $\mathbb{P}[h^{ab} \geq 0.5 - 2\tau | A \cap B \cap C] = 0$, since under the condition C, it suffices to prove $1 - 2E^{ab} > 6\tau$. However,

$$1 - 2E^{ab} = (1 - 2E^{a,root(t_1)})(1 - 2E^{root(t_1),b}) \geq (1 - 2g)^{depth(T)}(1 - 2g)^2\sqrt{1 - E^{a^*b}},$$

and we still have $\sqrt{1 - E^{a^*b}} > \sqrt{2\tau}$ according to (31). A calculation like the one resulting in (35) gives the result wanted. Now we are finished with the proof of (23).

Using these statements, $\mathbb{P}[A \cap B \cap D] \geq \mathbb{P}[A \cap B \cap D | A \cap B \cap C] \times \mathbb{P}[A \cap B \cap C] = \mathbb{P}[A \cap B \cap C] = \mathbb{P}[B \cap C]$, and we are back to the same estimates that proved Assumption (9), but we need a slightly smaller τ and consequently slightly larger k.

Note that the proof above applies to all $c_3 \in [c_2/2, c_2]$, if it applies to $c_3 = c_2$ and $c_3 = c_2/2$, so that Assumption (11) holds. □

Note that the proof also handled the problem that arises if some of the dissimilarity scores exceed 1/2, and so we cannot even compute corrected distances. The morale is that those pairs are not needed according to the proof. Therefore there is no need for additional conditioning for the shape of the observed data.

6.2 Runtime analyses of search strategies

Theorem 12 *(i) The running time of WAM based on sequential search is always* $O(n^2k + n^6 \log n)$.

(ii) The running time of WAM based on sparse-high search is always $O(n^2k + n^4 \log n \log k)$. *Assume now that our model tree is a random binary tree, under the uniform or Yule-Harding distribution, and all mutation probabilities are taken from an interval* $(p - \epsilon_n, p + \epsilon_n)$, *for a sufficiently small sequence* ϵ_n. *If* k *is as large as in (12), then with probability* $1 - o(1)$
(iii) The running time of WAM based on sequential search is $O(n^2k + n^3 polylogn)$.
(iv) The running time of WAM based on sparse-high search is $O(n^2k + n^2 polylogn)$.

Proof. Computing the matrices h and d takes $O(n^2k)$ time. (All distance methods begin by computing these distance matrices, but this "overhead cost" is usually always mentioned in the running time analysis of a given method.) Let w_0 be defined to be the smallest $w \in h^{ij}$ such that Q_w is T-forcing. Let $i(w)$ be the *order* of w within the sorted h^{ij} values. Then, since each call of the WATC algorithm uses $O(n^2 + |Q| \log |Q|)$ time, the running time of the sequential search is $O(i(w_0)(n^2 + |Q_{w_0}| \log |Q_{w_0}|))$, after the preprocessing.

For (i), the sequential search application of the WATC algorithm is $O(n^6 \log n)$, since we need never do more than examine all sets Q_w, and the largest such set has cardinality $O(n^4)$.

For (ii), the sparse-high search calls the WATC algorithm at most $O(\log k)$ times, and each call costs at most $O(n^4 \log n)$ time.

The depth of a random tree (under either the uniform or Yule-Harding distributions) is with high probability $O(\log \log n)$ by Theorem 10, and so there are at most $O(polylogn)$ leaves which are no more than about $O(\log \log n)$ distance (measured topologically) from any fixed leaf. This is the only fact that we exploit from the assumption of randomness of the tree.

For two leaves i, j, recall that $L(i, j)$ denotes the topological distance between i and j. We are going to show that if τ is the value at which the search reconstructs the tree in the proof of Theorem 11, then with probability $1 - o(1)$ we have $L(i, j) = O(\log \log n)$, whenever $i, j \in q \in Q_\tau$. This yields $|Q_{w(\tau)}| = n \cdot polylog(n)$.

In the proof of Theorem 11, according to Formula (18), event C holds with probability $1 - o(1)$. In that proof $Q_{w(\tau)}$ is denoted by $Z_{\tau/4}$. Now

$$(1 - 2g)^{L(i,j)} = 1 - 2E^{ij} \geq \tau/2, \tag{36}$$

where the equality follows from Lemma 1, and the inequality follows from the conditioning on the event C. Plugging in (35) for τ immediately yields $L(i, j) = O(\log \log n)$.

To finish the proof of (iii), realize that the sequential search makes at most n^2 calls of the WATC algorithm.

To obtain (iv), observe that Formulae (35, 17), and $depth(T) = O(\log \log n)$ imply that the number of iterations in the sparse-high search is

$$-\log_2 \tau = O(-\log(1 - 2g) \cdot depth(T)) = O(\log \log n). \square$$

6.3 The performance of other distance methods under the Neyman 2-state model

In this section we describe the convergence rate for the WAM and DCM method, and compare it briefly to the rates for two other distance-based methods, the Agarwala *et al.* 3-approximation algorithm [1] for the L_∞-nearest tree, and Neighbor Joining [38]. We make the natural assumption that all methods use the same corrected empirical distances from Neyman 2-state model trees. The comparison we provide in this section will establish that our method requires *exponentially shorter* sequences in order to ensure accuracy of the topology estimation than the algorithm of Agarwala *et al.*, for almost all trees under uniform or Yule-Harding probability distributions. The trees for which the two methods need comparable sequence lengths are those in which the diameter and the depth are as close as possible—such as complete binary trees. Even in these cases, WAM and DCM will nevertheless need shorter sequences than Agarwala *et al.* to obtain the topology with high probability, as we showed it in Section 3. (Again, note that this analysis is inherently pessimistic, and it is possible that the methods may obtain accurate reconstructions from shorter sequences than suffice by this analysis.)

The Neighbor Joining method is perhaps the most popular distance-based method used in phylogenetic reconstruction, and in many simulation studies (see [31, 32, 39] for an entry into this literature) it seems to outperform other popular distance based methods. The Agarwala *et al.* algorithm [1] is a distance-based method which provides a 3-approximation to the L_∞ nearest tree problem, so that it is one of the few methods which provide a provable performance guarantee with respect to any relevant optimization criterion. Thus, these two methods are two of the most promising distance-based methods against which to compare our method. All these methods use polynomial time.

In [21], Farach and Kannan analyzed the performance of the Agarwala *et al.* algorithm with respect to tree reconstruction in the Neyman 2-state model, and proved that the Agarwala *et al.* algorithm converged quickly for the variational distance. Personal communication from S. Kannan gave a counterpart to (12): if T is a Neyman 2-state model tree with mutation rates in the range $[f, g]$, and if sequences of length k' are generated on this tree, where

$$k' > \frac{c' \cdot \log n}{f^2(1-2g)^{2diam(T)}} \tag{37}$$

for an appropriate constant c', and where $diam(T)$ denotes the "diameter" of T, then with probability $1 - o(1)$ the result of applying Agarwala *et al.* to corrected distances will return the topology of the model tree. In [5], Atteson proved the same result for Neighbor Joining though with a different constant. (The constant for Neighbor Joining is smaller than the constant for the Agarwala *et al.* algorithm, suggesting that Neighbor Joining can be guaranteed to be accurate from shorter sequences than Agarwala *et al.*, on any tree in the Neyman 2-state model. However, remember that this analysis is pessimistic, and it may be that correct reconstruction is possible from shorter sequences than this analysis suggests.)

Comparing this formula to (12), we note that the comparison of depth and diameter is the issue, since $(1-\sqrt{1-2f})^2 = \Theta(f^2)$ for small f. It is easy to see that $diam(T) \geq 2depth(T)$ for binary trees T, but the diameter of a tree can in fact be quite large (up to $n-1$), while the depth is never more than $\log n$. Thus, for every fixed range of mutation probabilities, the sequence length that suffices to guarantee accuracy for the Neighbor Joining or Agarwala *et al.* algorithms can be quite large (i.e. it can grow exponentially in the number of leaves), while the sequence length that suffices for the witness-antiwitness method will never grow more than polynomially.

In order to understand the bound on the sequence length needed by these methods, we now turn to an analysis of the diameter of random trees. The models for random trees are the *uniform* model, in which each tree has the same probability, and the *Yule-Harding* model, studied in [2, 8, 26]. This distribution is based upon a simple model of speciation, and results in "bushier" trees than the uniform model.

Theorem 13 (i) *For a random semilabelled binary tree T with n leaves under the uniform model, $diam(T) > \epsilon\sqrt{n}$ with probability $1 - O(\epsilon^2)$.*
(ii) *For a random semilabelled binary tree T with n leaves under the Yule-Harding distribution, after suppressing the root, $diam(T) = \Theta(\log n)$, with probability $1 - o(1)$.*

Proof. We begin by establishing (i). The result of Carter *et al.* [11] immediately implies that leaves a, b have distance $m+1$ with probability *exactly* $m!N(n-2,m)/(2n-5)!!$. For ϵ small, $m \leq \epsilon\sqrt{n}$, this probability is $\Theta(\frac{m}{n})$. Two leaves a, b have distance at most $\epsilon\sqrt{n}$ in the random semilabelled tree under the uniform model with probability $O(\epsilon^2)$ for $\epsilon \to 0$, thus establishing the diameter. Summing up the probabilities from $m = 1$ to $m = \epsilon\sqrt{n}$, we get the required $O(\epsilon^2)$.

We now consider (ii). First we describe rooted Yule-Harding trees. These trees are defined by the following constructive procedure. Make a random permutation $\pi_1, \pi_2, \ldots, \pi_n$ of the n leaves, and join π_1 and π_2 by edges to a root R of degree 2. Add each of the remaining leaves sequentially, by randomly (with the uniform probability) selecting an edge incident to a leaf in the tree already constructed, subdividing the edge, and make π_i adjacent to the newly introduced node. For a rooted Yule-Harding tree T^R, let $h(T^R)$ denote the maximum distance of any leaf from the root. Let T be the unrooted Yule-Harding tree obtained from T^R by suppressing the root, and identifying the two edges incident with the root. Let $diam(T)$ denote the diameter of T. Then, we always have:

$$h(T^R) \leq diam(T) \leq 2h(T^R) - 1.$$

Now Aldous [2] shows that $h(T^R)/\log n$ converges in distribution to a (nonzero) constant c. Then, with probability tending to 1, $diam(T)/\log n$ will lie between c and $2c$. □

In the following table, we summarize sequence length that *suffice* for accurate reconstruction with high probability of WAM and DCM, and compare these to the sequence lengths that suffice for the Agarwala *et al.* algorithm, according to the analyses that we

have given above (thus, our summary is based upon (12), (37), and theorems 10 and 13). Sequence lengths are given in terms of growth as a function of n, and assume that mutation probabilities on edges lie within the specified ranges.

		range of mutation probabilities on edges:	
		$[f, g]$ f, g are constants	$\left[\frac{1}{\log n}, \frac{\log \log n}{\log n}\right]$
binary trees	DCM/WAM	polynomial	polylog
worst-case	Agarwala *et al.*	superpolynomial	superpolynomial
random binary trees	DCM/WAM	polylog	polylog
(uniform model)	Agarwala *et al.*	superpolynomial	superpolynomial
random binary trees	DCM/WAM	polylog	polylog
(Yule-Harding)	Agarwala *et al.*	polynomial	polylog

7 Extension to general stochastic models

In this section we consider the generalization of the WAM and DCM for inferring trees in the general stochastic model. Just as in the case of the Neyman 2-state model, we find that WAM and DCM obtains accurate estimations of the tree from sequences whose length is never more than polynomial in the number of leaves (for every fixed range for the mutation probabilities), and in general only polylogarithmic in the number of leaves. This should be contrasted to the study of Ambainis *et al.* [4].

Suppose the sequence sites evolve *i.i.d.* according to the "general" Markov model - that is, there is some distribution of states π at the root of the tree, and each edge e has an associated stochastic transition matrix $M(e)$, and the (random) state at the root evolves down the tree under a natural Markov assumption, as in the general stochastic model of Definition (III).

Let $f_{ij}(\alpha, \beta)$ denote the probability that leaf i is in state α and leaf j is in state β. By indexing the states, $f_{ij}(\alpha, \beta)$ forms a square matrix, $F_{ij} = [f_{ij}(\alpha, \beta)]$. Then

$$\phi_{ij} = -\log \det(F_{ij}) \tag{38}$$

denotes the *corrected model distance* between i and j. (There will be a guarantee for $\det(F_{ij}) > 0$.)

The *corrected empirical distance* $\hat{\phi}_{ij}$ of two species is computed as in (38), but uses the matrix $\hat{F}_{ij}$ composed of the relative frequencies $\hat{f}_{ij}(\alpha, \beta)$ of i being in state α and j being in state β, instead of the probability $f_{ij}(\alpha, \beta)$:

$$\hat{\phi}_{ij} = -\log \det(\hat{F}_{ij}). \tag{39}$$

Then, ϕ_{ij} can be derived from a positive edge weighting of the model tree, provided that the identifiability condition described in Section 2 [Tree Reconstruction] holds. These mild conditions only require that $\det(M(e))$ not take on the values $0, 1, -1$, and that the components of π are nonzero (i.e. every state has a positive probability of occurrence at the root).

Note that $\det(M(e))$ takes the values 1 or -1 precisely if $M(e)$ is a permutation matrix. Also, for the Neyman 2-state model $\det(M(e)) = 1 - 2p(e)$, where $p(e)$ is the mutation probability on edge e; thus, $\det(M(e)) > 0$ and $\det(M(e))$ tend to 0 as p approaches 0.5, and tend to 1 as p approaches 0. In general, $(1/2)[1 - \det(M(e)]$ plays the role of $p(e)$ in the general model. Thus, a natural extension of our restriction $f \leq p(e) \leq g$ and from the Neyman 2-state model corresponds to

$$0 < 1 - 2x' \leq \det(M(e)) \leq 1 - 2x < 1, \tag{40}$$

for suitable x, x', and we will henceforth impose this restriction for all edges of the tree. For technical reasons, we also impose the mildly restrictive condition[3] that every vertex can be in each state μ with at least a certain fixed positive probability:

$$\pi(v)_\mu > \epsilon. \tag{41}$$

Now, let $\lambda(e)$ be the weight of edge e in the realization of ϕ on the (unrooted version) of the true underlying tree T.

Lemma 4 *Set* $\delta(x) = -0.5\log(1-2x)$. *Then*

$$\lambda(e) \geq -0.5\log(\det(M(e))) \geq \delta(x) \tag{42}$$

for every edge e *of* T.

Proof. The second inequality follows from the restriction we imposed above on $\det(M(e))$. The first inequality in (42) follows from similar arguments to those appearing in Steel [42]; for the sake of completeness we give a proof.

Let T be the unrooted version of T^ρ. Now the edges of T correspond bijectively to the edges of T^ρ, except perhaps for one *troublesome* edge of T which arises whenever the root of T^ρ has degree two - in that case, two edges e_1, e_2 of T^ρ adjacent to ρ are identified to form e. For convenience, we assume in this proof that ρ is not a leaf.

We now prove that $\lambda(e) \geq -0.5\log\det(M(e))$ for all (non-troublesome) edges e of T, and if T has a troublesome edge e corresponding to edges e_1 and e_2 in T^ρ, then $\lambda(e) \geq -0.5\log\left(\det(M(e_1))\det(M(e_2))\right)$.

[3]This condition certainly holds under the Neyman 2-state model, the Kimura 3-st model [35], and much more general models (providing each state has positive probability of occurring at the root). Indeed this last weaker condition might be enough, but it would seem to complicate the analysis quite a lot.

For any edge $e = (v, w)$ of T^ρ where w is a leaf, let

$$h(e) = -\log\det(M(e)) - 0.5\log\left[\prod_\mu \pi(v)_\mu\right]$$

while, for any edge $e = (v, w)$ of T^ρ for which neither of $v,\ w$ are leaves, let

$$h(e) = -\log\det(M(e)) - 0.5\log\left[\prod_\mu \pi(v)_\mu\right] + 0.5\log\left[\prod_\mu \pi(w)_\mu\right].$$

Thus, h describes a weighting of the edges of T^ρ and thereby a weighting h^* of the edges of T by setting h^* equal to h on the non-troublesome edges, and the convention that if T has a troublesome edge e arising from the identification of a pair $e_1,\ e_2$ of edges of T^ρ then $h^*(e) = h(e_1) + h(e_2)$. Now, h realizes the ϕ_{ij} values on T^ρ. Thus, h^* also realizes the ϕ_{ij} values, on T and since (as we show) the edge weighting is strictly positive, it follows, by classical results [10], that this is the unique such edge weighting of T. Thus $\lambda = h^*$.

Now for an edge $e = (v, w)$ of T^ρ where w is a leaf,

$$h(e) \geq -\log\det(M(e)) \geq -0.5\log\det(M(e))$$

as claimed. Alternatively, for an edge $e = (v, w)$ of T for which neither of $v,\ w$ are leaves, we have

$$h(e) = -\log\det(M(e)) - 0.5\log\left[\prod_\mu \pi(v)_\mu\right] + 0.5\log\left[\prod_\mu \pi(w)_\mu\right] .$$

In order to derive our desired inequality we establish a further result. Let us suppose $M = [M_{\mu\nu}]$ is any $r \times r$ matrix with non-negative entries and $\mathbf{x}$ is a row vector of length r with non-negative entries. We claim that

$$\prod_\mu (\mathbf{x}M)_\mu \geq |\det(M)| \prod_\mu x_\mu \ .$$

To obtain this, note that the left hand side is just:

$$\prod_\mu \left(\sum_\nu x_\nu M_{\nu\mu}\right) \geq \left(\sum_\sigma M_{\sigma(1)1} M_{\sigma(2)2} \ldots M_{\sigma(r)r}\right) \prod_\mu x_\mu,$$

where the second summation is over all permutations σ of $(1, 2, \ldots, r)$, and so this sum is at least $|\det(M)|$, since the permanent of a nonnegative matrix is never smaller than the absolute value of its determinant. Now, $[\pi(w)_1, \ldots, \pi(w)_r] = [\pi(v)_1, \ldots, \pi(v)_r]M(e)$, and so, applying the above inequality to the case $M = M(e)$ and $x = [\pi(v)_1, \ldots, \pi(v)_r]$, we obtain

$$\prod_\mu \pi(w)_\mu \geq \det(M(e)) \prod_\mu \pi(v)_\mu \ .$$

Thus,

$$0.5\log\det(M(e)) \leq 0.5\log\left[\prod_\nu \pi(w)_\nu\right] - 0.5\log\left[\prod_\mu \pi(v)_\mu\right]$$

and so,

$$\begin{aligned} h(e) &= -0.5 \log \det(M(e)) \\ &- \left(0.5 \log \det(M(e)) + \log \left[\prod_{\mu} \pi(v)_{\mu}\right] - \log \left[\prod_{\mu} \pi(w)_{\mu}\right]\right) \\ &\geq -0.5 \log \det(M(e)), \end{aligned}$$

as claimed.

The inequalities for h now extend to $h^* = \lambda$ for all (non-troublesome) edges of T. If T^{ρ} has a troublesome edge e then $\lambda(e) = h^*(e) = h(e_1) + h(e_2)$, and from the above we have $h(e_i) \geq -0.5 \log \det(M(e_i))$ for $i = 1, 2$. □

Theorem 14 *Let $x = x(n)$ and $x' = x'(n)$ be such that for all edges in the tree T, $0 < 1 - 2x' \leq det(M(e)) \leq 1 - 2x < 1$. Assume x' has an upper bound strictly less than $1/2$. Mutatis mutandis, algorithms FPM, DCM and WAM, Theorems 9, 11, and 12 generalize to the general stochastic model under (40, 41). WAM and DCM returns the binary model tree T with probability $1 - o(1)$ if*

$$k > \frac{c \cdot \log n}{x^2 (1 - 2x')^{4depth(T)}} \tag{43}$$

with a certain constant c.

Proof. Recall the definition of the corrected empirical distance, $\hat{\phi}_{ij}$. By analogy with the proof of Theorem 9 (also see Theorem 8 and Formula (17) from [19]),

$$\left|\phi_{ij} - \hat{\phi}_{ij}\right| < \delta(x)/2 \tag{44}$$

is the same, as x tends to zero, as requiring the counterparts of [19], Formulae (19) and (20) for a $\gamma > 0$:

$$\left|\det(F^{ij}) - \det(\hat{F}^{ij})\right| < x(1 - \gamma) \det(F^{ij})/2. \tag{45}$$

(We return to [19], Formula (20) later.) To apply Lemma 3, we need to know how $\det(\hat{F}^{ij})$ responds to the replacement at one site of a pattern by a different pattern. If $\hat{F}_1^{ij}$ is the resulting F-matrix for this perturbed data set, then

$$\hat{F}_1^{ij} = \hat{F}^{ij} + (1/k) D^{ij}$$

where D^{ij} has one entry of $+1$, one entry of -1, and all other entries 0. Consequently,

$$|\det(\hat{F}_1^{ij}) - \det(\hat{F}^{ij})| \leq c_1/k$$

for some constant c_1. Also, since $\det(\hat{F}^{ij})$ is a polynomial in the entries of F^{ij}, we have:

$$\left|\mathbb{E}[\det(\hat{F}^{ij})] - \det(F^{ij})\right| \leq c_2/k \tag{46}$$

for some constant c_2. Combining (46) with the triangle inequality gives

$$|\det(F^{ij}) - \det(\hat{F}^{ij})| \leq \left|\det(\hat{F}^{ij}) - \mathbb{E}[\det(\hat{F}^{ij})]\right| + c_2/k$$

and so,

$$\mathbb{P}\left[|\det(F^{ij}) - \det(\hat{F}^{ij})| > t\right] \leq \mathbb{P}\left[|\det(\hat{F}^{ij}) - \mathbb{E}[\det(\hat{F}^{ij})]| > (t - c_2/k)\right] \tag{47}$$

for any $t > 0$. Hence by Lemma 3, we have that, as x tends to zero,

$$\begin{aligned} & \mathbb{P}\left[|\det(F^{ij}) - \det(\hat{F}^{ij})| > x(1-\gamma)\det(F^{ij})/2\right] \\ \leq\ & 2\exp\left(-d(\frac{x(1-\gamma)det(F^{ij})}{2} - \frac{c_2}{k})^2 k\right) \end{aligned} \tag{48}$$

for a constant d. A counterpart of the troublesome [19], Formula (20) also can be established, with a slightly worse RHS than that of (48). Putting the pieces (44,45,46) together we see that:

$$\mathbb{P}\left[|\phi_{ij} - \hat{\phi}_{ij}| > \delta(x)/2\right] \leq 2\exp\left(-d(\frac{x(1-\gamma)[\det(F^{ij}) - \frac{c_3}{k}]}{2} - \frac{c_2}{k})^2 k\right). \tag{49}$$

Now, what can we say about $\det(F^{ij})$ in the exponent on the right-hand side of (49)? $\det(F^{ij})$ is just the product of $\det(M(e))$ over all edges on the path from i to j, times the product of $\pi(v_{ij})_\mu$ over all states μ, where $\pi(v)$ is the vector of probabilities of states at vertex v, and v_{ij} is the most recent common ancestor of i and j in the tree. Due to our hypotheses (41) and (42), we have

$$\mathbb{P}\left[|\phi_{ij} - \hat{\phi}_{ij}| > (1/2)\min_e\{\lambda(e)\}\right] \leq 2\exp\left(-D(\frac{x[(1-2x')^{d(i,j)} - \frac{c_3}{k}]}{2} - \frac{c_2}{k})^2 k\right)$$

where D is a constant (dependent on ϵ and γ) and $d(i,j)$ is the number of edges in T separating leaves i and j. Hence, for any fixed quartet q of diameter $diam(q)$,

$$\mathbb{P}[\text{FPM errs on } q] \leq \exp\left(-Dx^2(1-2x')^{2diam(q)}\right). \tag{50}$$

Thus we have an analogue of Theorem 9 (but also see Theorem 8 in [19]).

Now we show how to generalize the proof of Theorem 11. To avoid needless repetitions, we give details for the proof of Assumption (9) only, and leave the proofs of Assumptions (10) and (11) to the Reader. Note that the proof of correctness of DCM hinges exactly on Assumption (9). Having a distance function in the general model, the width and algorithmic operations based on width generalize in a straightforward way.

For k evolving sites (i.e. sequences of length k), and $\tau > 0$, let us define the following two sets, $S_\tau = \{\{i,j\} : \det(\hat{F}^{ij}) > 2\tau\}$ and $Z_\tau = \{q \in \binom{[n]}{4} :$ for all $i,j \in q, \{i,j\} \in S_{2\tau}\}$ (note the similarity between the definition for the set Z_τ, and that for the set Q_w of quartet splits of quartets of width at most w). We also define the following two events, $A = \{Q_{short}(T) \subseteq Z_\tau\}$ and $B =$ FPM correctly reconstructs the tree for all $q \in Z_\tau$. Thus, $\mathbb{P}[\mathcal{A} \cap \mathcal{B} \neq \emptyset] \geq \mathbb{P}[A \cap B]$.

Let C be the event: "$S_{2\tau}$ contains all pairs $\{i,j\}$ with $\det(F^{ij}) > 6\tau$, and no pair $\{i,j\}$ with $\det(F^{ij}) \leq 2\tau$". Define $\lambda = \epsilon^r(1-2x')^{2depth(T)+3}$. We claim that:

$$\mathbb{P}[C] \geq 1 - (n^2 - n)e^{-c\tau^2 k} \tag{51}$$

for a constant $c > 0$ and

$$\mathbb{P}[A|C] = 1, \text{ if } \tau \leq \lambda/6 \ . \tag{52}$$

Suppose $\det(F^{ij}) \leq 2\tau$. To establish (51), using arguments similar to those between (45) and (48) one easily sees that Lemma 3 applies and

$$\begin{aligned} \mathbb{P}\left[\{i,j\} \in S_{2\tau}\right] &= \mathbb{P}\left[\det(\hat{F}^{ij}) > 4\tau\right] \\ &\leq \mathbb{P}\left[\det(\hat{F}^{ij}) - \det(F^{ij}) \geq 2\tau\right] \leq e^{-c\tau^2 k} \end{aligned}$$

for a constant $c > 0$.

Since there are at most $\binom{n}{2}$ such pairs $\{i,j\}$ such that $\det(F^{ij}) \leq 2\tau$, the probability that at least one such pair lies in $S_{2\tau}$ is at most $\binom{n}{2} e^{-c\tau^2 k}$. By a similar argument, the probability that $S_{2\tau}$ fails to contain a pair $\{i,j\}$ with $\det(F^{ij}) > 6\tau$ is also at most $\binom{n}{2} e^{-c\tau^2 k}$. These two bounds establish (51).

We now establish (52). For $q \in Q_{short}(T)$ and $i,j \in q$, if a path $e_1 e_2 ... e_t$ joins leaves i and j, then $t \leq 2depth(T) + 3$ by the definition of $Q_{short}(T)$. Using these facts, and the bound $\det(M(e)) \geq 1 - 2x'$, we obtain $\det(F^{ij}) \geq \epsilon^r(1-2x')^t$. Consequently, $\det(F^{ij}) > 6\tau$ (by assumption that $\tau \leq \lambda/6$) and so $\{i,j\} \in S_{2\tau}$ once we condition on the occurrence of event C. This holds for all $i,j \in q$, so by definition of Z_τ we have $q \in Z_\tau$. This establishes (52).

Then for any quartet $q \in Q_{short}(T)$, if e is the central edge of the contracted subtree induced by q in T, then $\det(M(e)) \leq 1-2x$. Furthermore, conditional on C, for any pendant edge e, $\det(M(e)) > \min\{\det(F^{ij}) : i,j \in q\} > 2\tau$. Thus, by (50), the analogue of Theorem 9, and the Bonferroni inequality, following the corresponding proof from Theorem 11, we obtain

$$\mathbb{P}[B \cap C] \geq 1 - \binom{n}{4} \exp\left(-Dx^2(1-2x')^{2depth(T)+3}\right) \tag{53}$$

for a constant $D > 0$. Combining the above, and invoking (52), we have:

$$\mathbb{P}[A \cap B] \geq \mathbb{P}[B \cap C] \ .$$

From (51) and (53) we have:

$$\mathbb{P}[A \cap B] \geq 1 - \binom{n}{4} \exp\left(-Dx^2(1-2x')^{2depth(T)+3}\right) - (n^2 - n)e^{-d\lambda^2 k}.$$

Formula (43) follows by an easy calculation.

For biological reasons we can think that the number of states r is bounded by an absolute constant, so $r = O(1)$. Turning to the generalization of Theorem 12, observe that $\det(\hat{F}_{ij})$

is a rational number between 0 and 1 with denominator k^r. Since $\log(k^r) = O(\log k)$, the analysis of the performance of sparse-high search under the general Markov model is the same as under the Neyman-2 model. The case of sequential search is even simpler. □

Note that the proof also handled the problem that arises if some logarithms are to be taken of negative numbers and so we cannot even compute corrected distances. The morale is that those pairs are not needed according to the proof. Therefore there is no need for additional conditioning for the shape of the observed data.

8 Using WAM or DCM for inferring trees in practice

In an earlier paper we presented a similar tree construction method, the Dyadic Closure Method (DCM). The DCM method is related to WAM because both reconstruct trees by using splits on quartets of leaves which lie close together in the tree. In this section, we describe certain properties of these two methods (and other similar methods, which reconstruct trees based upon small and compact subtrees.

8.1 Peculiarities of biological data

Systematic biologists have identified desirable properties of tree reconstruction methods, which address issues that arise because biological data do not have the properties that are present in the assumptions of the mathematical models of sequence evolution. The following (not exhaustive) list presents some of the features that methods should have if they are to be truly useful in analyzing real data:

- *Flexibility in the type of input data that the method can use.* Most methods can accept only one type of input data at a time, and traditional distance methods in particular need to be able to infer a distance matrix from the entire data. This makes the use of a mixture of types of data (combining morphological characters with biomolecular characters, and using different types of biomolecular characters) problematic. On the other hand, although WAM is based upon distances, as we have described it, it is clear that the WATC algorithm can be applied to quartets inferred using other methods (including, for example, Maximum likelihood). In this case, the selection of which quartets to analyze can be based upon the strength of the likelihood score or some other objective criterior, rather than upon distances.

- *Ability to handle missing and inapplicable data.* It is typically problematic to include characters (genes, morphological features, etc.) which do not apply to all the taxa in the input. Distance methods, for example, cannot necessarily extrapolate from distances based upon subsets of the characters to the entire set of characters, unless it is reasonable to assume that all sites evolve at the same rate (real data may not satisfy

this constraint). However, DCM avoids the problem of missing or inapplicable data, since we never analyze more than four taxa at a time, and under those conditions we may use all characters that apply to all four taxa without worrying about whether these characters also apply to taxa outside that quartet. This may make significantly longer sequences available to the reconstruction effort of very large trees than can currently be accommodated using traditional methods. (See [25] for a discussion of this particular issue.)

- *Robustness to model violations.* The stochastic models of sequence evolution we have discussed make the unrealistic assumption that the sites evolve identically and independently. Surprisingly, WAM may be robust to violations of these assumptions. For example, the use of appropriately selected methods robust to such model violations on four-taxon trees will increase robustness to the i.i.d. assumption, and while this may entail an increase in computational expense (if we use a computationally expensive method, such as MLE), the increase will not be as great as if we had applied MLE to the entire data set.

- *Robustness to errors in the multiple sequence alignment on the input sequences.* The multiple sequence alignment problem (MSA) is one of the most computationally difficult problems involved in tree reconstruction. All optimization problems which have been proposed are NP-hard, and heuristics are used to infer alignments instead. Typically these heuristics are based upon a tree for the sequences, but since the tree reconstruction methods require (generally) an alignment to start with, there is a problem with circularity! Therefore, heuristics based upon the unaligned sequences are used to infer a tree, and then the tree is used to infer an alignment. Unfortunately, the reliability of these heuristics is not at all clear, and the problem is clearly compounded on large trees. Furthermore, it is easy to see that the reliability of the tree reconstruction methods may be seriously compromised if the multiple sequence alignment is incorrect, and that even small errors in the alignment may affect the reliability of the short edges in the tree. However, robustness to multiple sequence alignment errors may be reasonable to expect in WAM. Consider the following approach to using WAM in the presence of an unreliable multiple sequence alignment. We let the multiple sequence alignment define the pairwise distances (or else we use pairwise alignments to define distances). Then, given a set of four taxa, instead of using the distances to infer the topology on the tree for those four taxa, we determine a multiple sequence alignment on the sequences representing the taxa. In other words, we ignore the preliminary multiple sequence alignment, and start from scratch. Since these are closely related taxa, their alignment is easier to identify, and since there are only four taxa, we can solve various optimization problems exactly, rather than heuristically (this is an important point, since MSA is one of the most difficult problems from a computational point of view in computational molecular biology). Given this multiple sequence alignment on four taxa, we then infer the best topology, using the most statistically powerful method available - perhaps MLE, if we have the time. At this point, the point should be noted that WAM only requires that each four-taxon subtree (on a short quartet) be reconstructed correctly; consequently, we do not need to have a completely accu-

rate multiple sequence alignment on four taxa, but rather only a good enough one, so that it suffices to indicate the correct topology. We direct the interested reader to the following papers which discuss the multiple sequence alignment problem and present results about the computational complexity of the multiple sequence alignment problem [33, 52, 51, 53, 50, 27].

8.2 Confidence in the output tree

If we apply the Witness-Antiwitness Method (WAM) to a given set of sequences, we may succeed in recovering the tree correctly, or we may reconstruct a tree different from the true tree, or we may not return any tree at all (i.e. the algorithm may return *Fail*). In this section, we address the possibility of the two types of failure.

We first discuss the case where WAM returns the wrong tree. All phylogenetic methods may fail to reconstruct the correct tree, and the question is whether the errors made by a method can be detected and corrected (or at least minimized). The conditions necessary for WAM to reconstruct *any* tree are sufficiently restrictive that it seems reasonable to expect that when WAM succeeds in reconstructing a tree that the reconstructed tree should be identical (with some high probability) to the true tree; indeed, this has been experimentally verified (T. Warnow, personal communication). However, when WAM does reconstruct the wrong tree, the same conditions indicate the types of errors WAM is likely to make, and also indicate how to correct the errors that WAM may make.

In systematic biology studies, errors in topology estimation are quantified according to how the bipartitions implied by the reconstructed tree compare to the bipartitions in the model tree. Thus, every edge in a leaf-labelled tree defines a partition of the leaves into two sets, and so the tree topology can be exactly described by the set of bipartitions induced by the edges of the tree. Thus, we can say that an edge e in the model tree appears in the reconstructed tree if the bipartition induced by that edge in the model tree also exists for the reconstructed tree. Edges in the model tree which do not appear in the reconstructed tree are *false negatives*, while edges which appear in the reconstructed tree but not in the model tree are *false positives*. While an exact topology reconstruction is the objective, quantifying and understanding the types of mistakes (false positives vs. false negatives) is useful, and may lead to *corrections* to the initial estimate of the tree topology. For example, if the reconstructed tree has some possibly ill-supported edges, these edges can be contracted (thus replacing a degree three node by a degree four node), and this results in possibly reducing the false positive rate, at the expense of possibly increasing the false negative rate. However, systematic biologists prefer false negatives to false positives, since all evolutionary assertions made by a tree without false positives are accurate, even if they provide only an incomplete picture of the evolutionary process. The key to success in this is being able to identify which edges are not well supported; this is not always easy.

However, in trees reconstructed by WAM it is generally an easy task. By its construction, edges in a tree reconstructed by WAM are likely to be accurate (i.e. they also appear in

the model tree) if they are large enough; consequently, only the very short edges need to be reconsidered as possible false positives. There are several techniques which can be used to determine whether those edges are well-supported (likelihood ratio tests, for example), and all edges which are insufficiently supported can then be contracted. The outcome then will be a tree which is likely to be a *contraction* of the true tree, while retaining all long enough edges from the model tree. Because the threshold for "long enough" is actually very low, this is a very nice property of WAM.

The other type of "failure" is when no tree is constructed. We see this "failure" to return a tree as providing an advantage over methods (such as all popular methods) which will reconstruct trees from any input, even if their input is not generated by a tree at all! By contrast, the return of "failure" by WAM can provide a test of the fit between the data and the model, and hence make it possible to reject (and accept) trees with greater confidence. However, if WAM does not construct a tree, adjusting the method to permit a search for the maximum number of quartets which can be simultaneously satisfied is a promising approach, and heuristics may be used to approximate the optimal solution.

8.3 Experimental results

The analysis of our algorithm heavily depends on a quantity that we introduced, the *depth* of a tree, while the performance of other algorithms heavily depend on the *diameter* of the tree, so that these quantities occur as exponents with the same base. For an n-leaf binary tree the diameter may be n, while the depth is at most $\log n$; furthermore, random binary trees from two natural distributions (uniform and Yule-Harding) have depth $O(\log \log n)$. The $\log \log n$ function is so slow growing, that it can be considered as bounded for n's that may occur in practice. Trees with the smallest possible depth—depth one—are caterpillars. Therefore it is instructive to check the performance of our algorithm on caterpillars, the binary trees with depth one, since very large random trees, from the point of view of depth, are rather similar to caterpillars.

In a joint paper with K. Rice [16], we presented the results of a preliminary experimental performance analysis comparing a popular method Neighbor Joining [38] and a primitive version of WAM. This study was obtained by simulating sequence evolution under the Neyman 2-state model on a 50-taxon caterpillar in which we had uniform edge mutation probabilities on all edges except the two most extreme edges, which each had mutation probabilities 5 times as great as the remaining edges. We varied the mutation probabilities on the edges while maintaining the ratio between every pair of edges, and we generated sequences of varying lengths. The performance advantage enjoyed by WAM over Neighbor Joining is very dramatic on this experimental study.

9 Summary and future research

In this paper we presented a new method, WAM (a speed-up of a method, DCM, that we proposed before), for constructing phylogenetic trees based upon constructing trees on (short) quartets and constructing trees consistent with the inferred topological constraints. We analysed a variant of the method in which we compute the topologies of quartets using a simple method derived from Buneman's Four Point Condition. We analysed the convergence rate of our method and compared it to the analysis of the convergence rate of the Agarwala *et al.* algorithm provided by Sampath Kannan. This comparison showed that the short quartet based methods can reconstruct Cavender-Farris trees from much shorter sequences than the 3-approximation algorithm of Agarwala *et al.* for the L_∞-nearest tree problem. We also showed that our method will be accurate on any distance matrix for which even an *exact* algorithm for the (NP-hard) L_∞-nearest tree can be guaranteed, on the basis of the L_∞-metric alone, to be accurate.

Although our results seem surprisingly strong, there is an intuitive explanation for why we might expect to obtain such an improvement in the performance over even an exact algorithm for the L_∞-nearest tree. That is, the *slack* permitted by the L_∞ metric allows *all* distances to change within the required bound, and thus does not constrain the topology of the reconstructed tree as closely as other criteria might.

However, the real gain in our method is obtained by relying upon the depth of the tree rather than the diameter. Almost all (actually, all others, to our knowledge) distance-based methods are sensitive to the diameter, by contrast, and hence suffer when attempting to reconstruct large trees containing widely divergent sequences. Hence short quartet based methods seem utterly appropriate for problems like the Eve hypothesis, the evolution of the HIV virus, or the Tree of Life. The research project [56] used mitochondrial DNA, described as "fast ticking clock", to discover the phylogeny of modern humans, and concluded with the controversial "Eve hypothesis": an African female who lived about 200,000 years ago, was a common ancestor of all present humans. Their tree [56] on p. 69 has 182 leaves, and by a hand investigation seems to have diameter 19 and depth 4. Even so, our predicted improvement in performance needs to be verified on real data.

To apply short quartet based methods successfully to very large data sets may require further improvements on the method: (1) faster procedures for DCTC and/or WATC algorithm (2) use and analysis of sequence-based methods for obtaining quartet splits (3) proper handling of "slightly" inconsistent sets of quartet splits (4) extensions to allow non-binary trees.

10 Acknowledgements

Thanks are due to Scott Nettles for fruitful discussions of efficient implementations, and David Bryant and Éva Czabarka for proofreading the manuscript.

Péter L. Erdős was supported in part by the Hungarian National Science Fund contract T 016 358. László A. Székely was supported by the National Science Foundation grant DMS 9701211, the Hungarian National Science Fund contracts T 016 358 and T 019 367, and and European Communities (Cooperation in Science and Technology with Central and Eastern European Countries) contract ERBCIPACT 930 113. Michael A. Steel was supported by the New Zealand Marsden Fund and the New Zealand Ministry of Research, Science and Technology. Tandy J. Warnow was supported by an NSF Young Investigator Award CCR-9457800, a David and Lucille Packard Foundation fellowship, and generous research support from the Penn Research Foundation and Paul Angello.

This research started in 1995 when the authors enjoyed the hospitality of DIMACS during the Special Year for Mathematical Support to Molecular Biology, and was completed in 1997 while enjoying the hospitality of Andreas Dress, at Universität Bielefeld, in Germany.

References

[1] R. Agarwala, V. Bafna, M. Farach, B. Narayanan, M. Paterson, 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, 365–372.

[2] D. J. Aldous, 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, 1995, 1–18.

[3] N. Alon and J. H. Spencer, *The Probabilistic Method,* John Wiley and Sons, New York, 1992.

[4] A. Ambainis, R. Desper, M. Farach, and S. Kannan. Nearly tight bounds on the learnability of evolution. To appear, *Proc. of the 1998 Foundations of Computer Science.*

[5] K. Atteson. On the performance of neighbor-joining. To appear, *Proc. of COCOON 1997.*

[6] H.-J. Bandelt and A. Dress, Reconstructing the shape of a tree from observed dissimilarity data, *Adv. App. Math.*, **7** (1986), 309–343.

[7] V. Berry and O. Gascuel, Inferring evolutionary trees with strong combinatorial evidence, *Proc. COCOON 1997.*

[8] J. K. M. Brown, Probabilities of evolutionary trees, *Syst. Biol.* **43** (1994), 78–91.

[9] D. J. Bryant and M. A. Steel, Extension operations on sets of leaf-labelled trees, *Adv. App. Math.*, **16** (1995), 425–453.

[10] P. Buneman, The recovery of trees from measures of dissimilarity, in *Mathematics in the Archaeological and Historical Sciences*, F. R. Hodson, D. G. Kendall, P. Tautu, eds.; Edinburgh University Press, Edinburgh, 1971, 387–395.

[11] M. Carter, M. Hendy, D. Penny, L. A. Székely, and N. C. Wormald, On the distribution of lengths of evolutionary trees, *SIAM J. Disc. Math.* **3** (1990), 38–47.

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

[13] J. T. Chang and J. A. Hartigan, Reconstruction of evolutionary trees from pairwise distributions on current species, *Computing science and statistics: Proceedings of the 23rd symposium on the interface* (1991), 254–257.

[14] H. Colonius and H. H. Schultze, Tree structure for proximity data, *Brit. J. Math. Stat. Psychol.*, **34** (1981), 167–180.

[15] W.H.E. Day, Computational complexity of inferring phylogenies from dissimilarities matrices, *Information Processing Letters*, 30:215-220, 1989.

[16] P. L. Erdős, K. Rice, M. Steel, L. Székely, and T. Warnow, The Short Quartet Method. to appear, *Mathematical Modelling and Scientific Computing,* Principia Scientia.

[17] P. L. Erdős, M. A. Steel, L. A. Székely and T. Warnow, Local quartet splits of a binary tree infer all quartet splits via one dyadic inference rule. *Computers and Artificial Intelligence*, Vol. 16, No. 2, pp.217-227, 1997.

[18] P. L. Erdős, M. A. Steel, L. A. Székely and T. Warnow, Inferring big trees from short quartets. *Proceedings of ICALP 1997.*

[19] P. L. Erdős, M. A. Steel, L. A. Székely and T. Warnow, A few logs suffice to build (almost) all trees (I), Submitted, *Random Structures and Algorithms.*

[20] M. Farach, and J. Cohen, Numerical Taxonomy on Data: Experimental Results, *ACM-SIAM Symposium on Discrete Algorithms*, 1997.

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

[22] M. Farach, S. Kannan, and T. Warnow, A robust model for inferring optimal evolutionary trees, *Algorithmica* **13** (1995) 155–179.

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

[24] D. Feng and R. Doolittle, Progressive sequence alignment as a prerequisite to correct phylogenetic trees, *J. Mol. Evol.* **25**(1987), 351–360.

[25] Green Plant Phylogeny Research Coordination Group, Summary report of Workshop #1: Current Status of the Phylogeny of the Charophyte Green Algae and the Embryophytes. University and Jepson Herbaria, University of California, Berkeley, June 24-28, 1995. 7 January, 1996.

[26] E. F. Harding, The probabilities of rooted tree shapes generated by random bifurcation, *Adv. Appl. Probab.* **3** (1971), 44–77.

[27] J. Hein, A new method that simultaneously aligns and reconstructs ancestral sequences for any number of homologous sequences, when the phylogeny is given, *Mol. Biol. Evol.* **6**(1989), 649–668.

[28] D. Hillis, Approaches for assessing phylogenetic accuracy, *Syst. Biol.* **44** (1995), 3–16.

[29] D. Hillis, Inferring complex phylogenies, *Nature* Vol **383** 12 September, 1996, 130–131.

[30] P. Hogeweg and B. Hesper, The alignment of sets of sequences and the construction of phylogenetic trees, an integrated method, *J. Mol. Evol.* **20**(1984), 175–186.

[31] Huelsenbeck, J.1995. Performance of phylogenetic methods in simulation. *Syst. Biol.* 44:17-48.

[32] Huelsenbeck, J.P. and D. Hillis. 1993. Success of phylogenetic methods in the four-taxon case. *Syst. Biol.* 42:247-264.

[33] T. Jiang, E. Lawler, and L. Wang, *Aligning sequences via an evolutionary tree: complexity and approximation*, ACM STOC '94.

[34] S. Kannan, personal communication.

[35] M. Kimura, Estimation of evolutionary distances between homologous nucleotide sequences. *Proc. Nat. Acad. Sci. USA*, 78: 454-458, 1981.

[36] J. Neyman, Molecular studies of evolution: a source of novel statistical problems. in *Statistical Decision Theory and Related Topics*, Gupta, S. S. and J. Yackel (eds), New York, Academic Press, 1971, 1–27.

[37] K. Rice and T. Warnow, Parsimony is hard to beat!, Proceedings, COCOON 1997.

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

[39] 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.

[40] Y. A. Smolensky: A method for linear recording of graphs, *USSR Comput. Math. Phys., 2* (1969), 396–397.

[41] M. A. Steel, The complexity of reconstructing trees from qualitative characters and subtrees, *J. Classification*, **9** (1992), 91–116.

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

[43] M. Steel, M. D. Hendy, D. Penny, Invertible models of sequence evolution, submitted to *Mathematical Biosciences.*

[44] M. A. Steel, L. A. Székely, and M. D. Hendy, Reconstructing trees when sequence sites evolve at variable rates, *Journal of Comput. Biol.*, **1** (1994), 153–163.

[45] K. Strimmer and A. von Haeseler, Quartet Puzzling: a quartet Maximum Likelihood method for reconstructing tree topologies, *Mol. Biol. Evol.*, **13** (1996), 964–969.

[46] K. Strimmer, N. Goldman, and A. von Haeseler, Bayesian probabilities and Quartet Puzzling, *Mol. Biol. Evol.*, **14** (1997), 210–211.

[47] D. L Swofford, G. J. Olsen, P. J. Waddell, 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.

[48] D. Harel and R.E. Tarjan. Fast Algorithms for finding nearest common ancestors, *SIAM J. Computing*, **13** (1984), 338-355.

[49] N. Takezaki and M. Nei, Inconsistency of the maximum parsimony method when the rate of nucleotide substitution is constant, *J. Mol. Evol.*, **39** (1994), 210–218.

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

[51] L. Wang and T. Jiang, On the complexity of multiple sequence alignment, *J. Comput. Biol.* **1** (1994), 337–348.

[52] L. Wang, T. Jiang and E. Lawler, Approximation algorithms for tree alignment with a given phylogeny, *Algorithmica* **16** (1996), 302–315.

[53] L. Wang, T. Jiang and D. Gusfield, A more efficient approximation scheme for tree alignment in *Proc. First Annual International Conference on Computational Molecular Biology*, Jan. 1997, Santa Fe, NM.

[54] T. Warnow, *Combinatorial algorithms for constructing phylogenetic trees*, PhD thesis, University of California-Berkeley, 1991.

[55] M. S. Waterman, T. F. Smith, M. Singh, and W. A. Beyer, Additive evolutionary trees, *J. Theoret. Biol.*, **64** (1977), 199–213.

[56] Allan C. Wilson, Rebecca L. Cann, The recent African genesis of humans, *Scientific American* April 1992, 68–73.

[57] K. A. Zaretsky, Reconstruction of a tree from the distances between its pendant vertices, *Uspekhi Math. Nauk (Russian Mathematical Surveys), 20* (1965), 90–92 (in Russian).