# MAXIMAL ACCURATE FORESTS FROM DISTANCE MATRICES

C. DASKALAKIS, C. HILL, A. JAFFE,
R. MIHAESCU, E. MOSSEL, S. RAO

Abstract. We present a fast converging method for distance-based phylogenetic inference, which is novel in two respects. First, it is the only method (to our knowledge) to guarantee accuracy when knowledge about the model tree, i.e bounds on the edge lengths, is *not* assumed. Second, our algorithm guarantees that, with high probability, no false assertions are made. The algorithm produces a maximal edge-disjoint subforest of the model tree, with running time $O(n^4)$ in the worst case. Empirical testing has been promising, comparing favorable to Neighbor Joining and Maximum Parsimony, with the advantage of making no false assertions about the topology of the model tree; guarantees against false positives can be controlled as a parameter by the user.

## 1. Introduction

The shortcomings of "naive" distance methods in phylogenetic reconstruction, such as Neighbor Joining (NJ) [12], are well-known, and reconstructing trees from small subtrees is evidently both desirable and increasingly popular. All quartet-based methods are examples of this paradigm. However, this divide-and-conquer approach presents at least two serious difficulties: (1) identifying those subsets of taxa on which a tree topology can be accurately inferred; and (2) retaining accuracy when some subtree topologies cannot be correctly determined. In particular, quartet methods, such as the Dyadic Closure Method of [4] and the series of Disk-Covering Methods (DCM) [8, 13] are confined to considering only quartets of small diameter, so-called short quartets, in the hope that these provide enough information for a complete reconstruction. These methods, moreover, are compelled to reconstruct the entire tree; consequently, errors are incurred when attempting to combine subtrees when the given distance matrix simply does not justify the attempt.

The first DCM method, DCM1, is a good illustration of these difficulties. That method iterates over thresholds $\hat{D}(i,j)$ where $\hat{D}$ is the given distance matrix–estimated from sequences, for example. At threshold $w$, a graph $G_w$ is constructed, where the vertices of $G_w$ are the taxa, with an edge between $i, j$ whenever $\hat{D}(i,j) \leq w$. Trees are built on maximal cliques of a triangulation $G_w^*$ of $G_w$ using a base method such as NJ and merged according to a perfect elimination order of $G_w^*$. In some cases, there may be no accuracy guarantees for the trees built on maximal cliques of $G_w^*$, and the merging procedure–using strict consensus merger–is provable only when $\hat{D}$ is nearly additive (so that $G_w$ itself is chordal).

Much recent work in the study of distance-based methods has focused on the notion of *fast convergence.* Indeed, the work of [4, 5] can be considered a breakthrough in this vein; there, the authors delineate an algorithm which accurately infers almost all trees on $n$ leaves when provided sequences of length $O(poly(log(n))$, and all trees with $O(poly(n))$ length sequences. By way of comparison, the venerable NJ requires exponentially long sequences. A notable drawback of the Dyadic Closure method of [4], however, is the dearth of useful performance guarantees when sequence lengths are small. In this paper, we will present an algorithm which achieves fast convergence, to the same extent and with similar time complexity as in [5], and further, is guaranteed to return accurate subtrees even when sequences are too short to infer the whole tree correctly.

To this end, we adapt the work of [9], a method which reconstructs a collection of edge-disjoint subtrees of the model tree from which only a constant fraction of edges is omitted, when given $O(\log n)$ characters. We have improved on the framework of [9], for we do away with the need for parameters $f$ and $g$, the lower and upper bounds on the lengths of edges of the model tree. Specifically, we prove a *local quartet reliability criterion*, which is blissfully ignorant of $f$ and $g$. This permits our algorithm to produce an accurate subforest which is as large as possible from the data provided–it builds everything that can be built. Subsequently, such a forest can be used to boost other reconstruction methods by, for example, inferring sequences at ancestral nodes.

In the following subsection, we will present a number of definitions towards formulating the optimization problem for which our algorithm is a solution, namely, the Maximal Subforest (MS) problem. In Section 2 we delineate the subtree reconstruction and forest construction algorithms and analyze their performance. This section also constitutes a significant simplification of the arguments in [9], and the efficiency of our methods is such that we have been able to implement them. Experimental results are examined in Section 4. In Section 3, we prove that our method reconstructs almost all $n$-leaf trees accurately given sequences of length $O(poly(log(n)))$; our method achieves this guarantee with marked improvements in efficiency.

1.1. **Definitions and notation.** Let $T$ be an edge-weighted, unrooted binary tree. (In the sequel, all trees are assumed to be unrooted.) Then, we define $\mathcal{L}(T)$ to be the set of leaves of $T$. For any subset $X$ of $\mathcal{L}(T)$, $T|X$ denotes the restriction of $T$ to $X$. We assume that $T$ is leaf-labelled by a set of taxa, $S$, of size $n$ and that $S$ is equipped with a distance matrix $\hat{D}$. For each taxon $v \in S$, let $L(v)$ denote a subset of $S$ such that if $\hat{D}(v,y) < \hat{D}(v,x)$ and $x \in L(v)$, then $y \in L(v)$. For $x, y \in S$, let $P(x,y)$ denote the set of edges of the path from $x$ to $y$ in $T$. We say that $L(u)$ and $L(v)$ are *edge-sharing* if there exist $x, y \in L(u)$ and $x', y' \in L(v)$ such that $P(x, y) \cap P(x', y')$ is nonempty; otherwise, $L(u)$ and $L(v)$ are *edge-disjoint*. For $U \subseteq S$, $\mathcal{E}(U)$ is the graph with vertex set $\{L(x)|x \in U\}$ and edges determined by the edge-sharing relation. Naturally, $\mathcal{E}(U)$ is called an edge-sharing graph on $U$. For convenience, we will freely identify a node $L(x)$ of $\mathcal{E}(S)$ with $x$ itself. Let $N(v)$ denote the set of neighbors of $v$ in $\mathcal{E}(S)$. Then, we define $SL(v) = L(v) \cup \bigcup_{u \in N(v)} L(u)$.

We will make use of the *strict consensus merger* [3] method for constructing supertrees. The strict consensus merger of two unrooted leaf-labelled trees is defined as follows. Let $t$ and $t'$ be trees. Let $L = \mathcal{L}(t) \cap \mathcal{L}(t')$and let $z = t|L$ and $z' = t'|L$; let $Z$ be the maximally resolved tree that is a contraction of both $z$ and $z'$. We call $Z$ the *backbone* of $t$ and $t'$. Finally, reattach the remaining pieces of $t$ and $t'$ to $Z$ appropriately. Note that the strict consensus merger of a pair of trees need not be unique. In particular, this method may attach two pieces of $t$ and $t'$ to the same edge of $Z$, returning a vertex of degree greater than three.

Generally, each taxon $s \in S$ is identified with a sequence over some alphabet $\Sigma$–for example, $\Sigma = \{A, C, G, T\}$. $S$ is equipped with a distance matrix $\hat{D}$, which is, by definition, symmetric, zero along the diagonal, and positive off the diagonal. The following several definitions and Theorem 1 motivate the algorithms of this paper.

**Definition 1.** *Let $T$ be an edge-weighted binary tree, leaf-labelled by $S$, and let $D$ be the associated additive matrix. Suppose $0 < \epsilon < M$. We say that $\hat{D} : S \times S \to \mathbb{R}^+$ is a* local $(\epsilon, M)$ distortion for $S' \subseteq S$ *if*

(1) *$\hat{D}$ is a distance matrix.*
(2) *$\hat{D}(x, y) = \infty$ implies $D(x, y) > M$, for all $x, y \in S'$*
(3) *$\hat{D}(x, y) < M$ implies $|\hat{D}(x, y) - D(x, y)| < \epsilon$, for all $x, y \in S'$*

**Definition 2.** *Let $T$ be an edge-weighted trivalent tree, leaf-labelled by $S$, and let $D$ be the associated additive matrix. Suppose $S = C_1 \sqcup ... \sqcup C_\alpha$ such that $T|C_i$ and $T|C_j$ are edge-disjoint for each $1 \leq i < j \leq \alpha$. For each $i \leq \alpha$, let $0 < \epsilon_i < M_i$ be given. Suppose $\hat{D} : S \times S \to \mathbb{R}^+$. We say that $\mathcal{C} = \{(C_i, \epsilon_i, M_i) : 0 \leq i \leq \alpha\}$ is a* local distortion decomposition of $\hat{D}$ *if $\hat{D}$ is a local $(\epsilon_i, M_i)$ distortion for $C_i$, for each $i = 1, ..., \alpha$.*

*Furthermore, let $f_i$ be the weight of the smallest edge in $T|C_i$, and let $\epsilon_i < \frac{f_i}{2}$; and let $r_i \leq \frac{M_i - 7\epsilon_i}{6}$, and assume $M_i > 7\epsilon_i$. For each $v \in C_i$, let $L(v)$ be the ball of radius $r_i$ about $v$. If $\mathcal{E}(C_i)$ are the connected components of $\mathcal{E}(S)$, then we say that $\mathcal{C}$ is* constructive.

The component reconstruction procedure presented below justifies the use of the word “constructive”; in the case described, we can accurately reconstruct $T|C_i$ in polynomial time.

**Theorem 1** ([9]). *Let $T$ be an edge-weighted trivalent tree, leaf-labelled by $S$, and let $D$ be the associated additive matrix. Suppose $\hat{D}$ is an $(\epsilon, M)$*

*distortion for $S$ with $\epsilon < f/2$ and $M > 7\epsilon$, where $f$ is the weight of the smallest edge in $T$. Let $g$ be the weight of the largest edge in $T$. Let $\mathcal{E}(S)$ be the edge-sharing graph of $r$-balls around leaves where $r = \frac{M-7\epsilon}{6}$, and let $C_1, ..., C_\alpha$ be the components of $\mathcal{E}(S)$. Then $\mathcal{C} = \{(C_i, \epsilon, M)\}$ is a constructive local distortion decomposition, and $\alpha \leq 1 + \frac{60}{\sqrt{2}} 2^{-(M-\epsilon)/2g} \cdot n$. Moreover, the corresponding forest can be constructed in polynomial time.*

In principle, a binary search on $r$ might be expected to find the decomposition of Theorem 1. Observe, however that Theorem 1 takes the length of the shortest edge $f$ as a global criterion for accurate reconstruction of subtrees. But if edge-disjointness can be maintained, the length $f_i$ of the shortest edge in $T|C_i$ has no bearing on the reconstruction of $T|C_j$, when $i \neq j$. One should prefer to consider ball radii as large as possible, thereby increasing the sizes of the components of $\mathcal{E}(S)$, without incurring false resolutions. Thus, our work can be considered a solution of the following optimization problem:

**Definition 3** (Maximal Subforest Problem)**.** *Given a distance matrix $\hat{D}$ generated on a binary tree $T$, find a constructive local distortion decomposition of $\hat{D}$ such that the number of edge-dispoint components $\alpha$ is minimized.*

## 2. Our Algorithm

We start off by giving a high level picture of the algorithm, with the details of the various pieces to be described in later sections. Intuitively, in order to maximize the radii $r_i$ of Definition 2, when minimal edge weights are unknown, it is reasonable to grow radii incrementally. Thus, we sort the set of pairs $(x, y)$, $x, y \in S$, under $\hat{D}$. We would like to continue throwing in pairs $(x, y)$ just as long as we are certain of the accuracy of every $T|SL(v)$. Accuracy will be guaranteed by virtue of Algorithm 2 for quartet reliability.

### 2.1. **A Local Quartet Reliability Criterion.**

We describe a test which, given sequences at 4 leaves, returns the correct quartet split with high probability or fails if the sequences at the leaves are too noisy. For succinctness of description, we will present the test in the context of the Cavender-Farris-Neyman 2-state model, but as will become clear, it can be easily generalized to the general Markov model by virtue of the analysis in section 7 of [5].

We begin with a high level description of the CFN model and introduce some notation. Suppose $T$ is a rooted tree and $p : E(T) \to (0, 1/2)$ is a function associating to each edge a transition probability. Under the CFN model, a character is chosen at the root of the tree uniformly at random from $\Sigma = \{-1, 1\}$, and this value is propagated towards the leaves, mutating along each edge with probability $p(e)$. An equivalent description of the corresponding Markov model is the following: along every edge of the tree with probability $\theta(e) = 1 - 2p(e)$, the child copies its value from the father, and with probability $1 - \theta(e)$, it randomizes uniformly in $\{-1, 1\}$. It follows

**Algorithm 1** (Forest Reconstruction Algorithm)

```
Sort the set of pairs $E$ of vertices in ascending order under $\hat{D}$.
Let **Forest** be the set of subtrees of $T$; initially each subtree consists of a single leaf.
while $E \neq \emptyset$ do
  $(x, y) := pop(E)$
  $L(x) := L(x) \cup \{y\}$ and $L(y) := L(y) \cup \{x\}$
  Compute $\mathcal{E}(S)$ and $SL(\cdot)$ trees (Algorithm 3)
  if Algorithm 3 failed, i.e. a quartet induced by the new edge $(x, y)$ is wrong then
    $E := E \setminus \{(x, y)\}$; undo $L(\cdot)$ augmentations;
  else
    Construct $T|C$, $x, y \in C$, storing in **Forest** (Algorithm 4)
  end if
end while
```

easily from the above definitions that the probability $p(u, v)$ that the endpoints $u, v$ of a path $P(u, v)$ of topological length $k$ are in different states is related to the mutation probabilities $p_{e_1}, p_{e_2}, \dots, p_{e_k}$ of the edges of $P(u, v)$ by the formula $p(u, v) = \frac{1}{2}(1 - \theta(u, v))$ where

$$\theta(u, v) = \prod_{i=1}^{k} \theta(e_i)$$

This formula justifies the definition of $d(u, v) = -\frac{1}{2} \log \theta(u, v)$ as a path metric on the tree.

Now, given $k$ samples of the process at the leaves of the tree, $\{\sigma^k_{\mathcal{L}(T)}\}^k_{t=1}$, we can empirically estimate $\theta(u, v)$ for all $u, v \in \mathcal{L}(T)$, using the following empirical measure:

$$c(u, v) = \frac{1}{k} \sum_{t=1}^{k} \sigma^t_u \sigma^t_v$$

The local test for finding quartet splits reliably is described briefly in Algorithm 2 and its correctness is proved in Theorem 2.

**Theorem 2.** *If Algorithm 2 outputs a quartet split, then this split is correct with probability at least* $1 - \delta_1$.

*Proof.* By the Azuma-Hoeffding inequality, it is not hard to see that for all $i, j \in \{1, 2, 3, 4\}$,

$$\mathbb{P}\left[|\theta(i, j) - c(i, j)| \geq \alpha(k, \delta_1)\right] \leq 2 \cdot \exp\left\{-\frac{\alpha(k, \delta_1)^2 k}{2}\right\}$$

From the choice of $\alpha(k, \delta_1)$ it follows that, with probability at least $1 - \delta_1$, we have $|\theta(i, j) - c(i, j)| \leq \alpha(k, \delta_1)$ for all $i, j \in \{1, 2, 3, 4\}$ Without loss of generality, suppose that the correct quartet on the leaves $\{1, 2, 3, 4\}$ is $12|34$.

**Algorithm 2** (Quartet Reliability Criterion)

```
INPUT: $k$ samples of the CFN model on four leaves $\{1,2,3,4\}$ and a parameter $\delta_1 > 0$
OUTPUT: a quartet split of $\{1,2,3,4\}$ or "fail" if not enough data; if a quartet split is returned, it is correct with probability at least $1-\delta_1$

Take $\alpha(k,\delta_1) := \sqrt{\frac{2}{k}\ln\frac{12}{\delta_1}}$ and $\frac{1}{\epsilon} := min_{u,v\in\{1,2,3,4\}}\left\{\frac{c(u,v)}{\alpha(k,\delta_1)}\right\}$
if  $\epsilon \geq 1$ then
  return "fail" /* the estimation error is bigger than some estimation*/
end if
for $i,j \in \{1,2,3,4\}, i \neq j$ do
  for $k,l \in \{1,2,3,4\} - \{i,j\}, k \neq l$ do
    if $\sqrt{\frac{c(i,j)c(k,l)}{c(i,k)c(j,l)}} < 1 - \frac{2\epsilon}{1-\epsilon}$ then
      return $ij|kl$
    end if
  end for
end for
return "fail"
```

Suppose that the middle "edge" of the quartet split corresponds to a path $p$ in $T$ with $\theta(p) = \prod_{e\in p}\theta(e)$. Since the algorithm does not return "fail," it needs be $\epsilon < 1$, and we can show the following by easy calculations:

$$\sqrt{\frac{c(i,j)c(k,l)}{c(i,k)c(j,l)}}\begin{cases} < \theta(p)\cdot\left(1+\frac{2\epsilon}{1-\epsilon}\right), & \text{if } \{i,j\}=\{1,2\} \text{ and } \{k,l\}=\{3,4\} \\ > \frac{1}{\theta(p)}\cdot\left(1-\frac{2\epsilon}{1-\epsilon}\right), & \text{if } \{i,j\}=\{1,3\} \text{ and } \{k,l\}=\{2,4\} \end{cases}$$

It follows that if $\sqrt{\frac{c(i,j)c(k,l)}{c(i,k)c(j,l)}} < 1 - \frac{2\epsilon}{1-\epsilon}$ then the split $ij|kl$ is the correct quartet split. □

2.2. **Local Tree Reconstruction.** In this section we will prove that Algorithms 3 and 4 correctly reconstruct the subforest corresponding to a set of $L(\cdot)$'s as long as the sequence length permits correct estimation of the quartet splits. If this is not the case, the algorithms will fail without returning an incorrect tree. All the above claims are with high probability for $k > c(T,f,g)\log n$.

**Theorem 3.** *If algorithm 3 does not fail, then the tree output by algorithm 4 is correct with probability at least* $1 - n^4\delta_1$.

*Proof.* Suppose that algorithm 3 does not "fail". It follows that all quartets it considers pass the test of Algorithm 2. Now, since there are at most $\binom{n}{4}$ of them and each is estimated correctly with probability at least $1-\delta_1$, the probability that they are all estimated correctly is at least $1 - n^4\delta_1$. It only remains to argue that if all quartets are estimated correctly, the tree output by algorithm 4 is correct. Note that the $T|SL(v)$'s that are computed by algorithm 3 are correct so that the input to algorithm 4 is correct. So we have to show that 4 finds the supertree of these trees correctly. The proof of the later is given by lemmas 1, 2 and 3 which, also, provide a streamlined proof of the correctness of [9]. □

**Algorithm 3** (Construction of Edge Sharing Graph and $SL(\cdot)$ trees)

```
INPUT: $\{L(v)\}_{\forall v \in \mathcal{L}(T)}$
OUTPUT: Edge Sharing Graph and $T|SL(v)$'s or "fail"

$r_i := max_{u \in L(i)} \widehat{D}(u, i)$, $Q_i := \emptyset$ for all $i$
for $i = 1$ to $n$ do
  for every quartet $q$ that contains $i$ and has estimated width $\leq 6r_i$ do
    if $q$ does not pass test of algorithm 2 then
      return "fail";
    else
      store quartet in $Q_i$;
    end if
  end for
end for
for $i = 1$ to $n$ do
  Merge quartets in $Q_i$ into a tree using some base method;
end for
```

**Algorithm 4** (Component reconstruction)

```
INPUT: $SL(\cdot)$ trees of a connected component $C$ of $\mathcal{E}(S)$
OUTPUT: $T|C$

Let $v_1, ..., v_r$ be a perfect elimination order of the leaves of a component
$C$ of $\mathcal{E}(S)$ (by lemma 1 $C$ is triangulated).
for $1 \leq i \leq r$ do
  Let $X_i = SL(v_i) \cap \{v_{i+1}, ..., v_r\}$
  Get $t_i = T|X_i \cup \{v_i\}$ by restricting $T|SL(v_i)$
end for
Set $T_r = t_r$
for $i = r - 1$ to 1 do
  $T_i :=$ strict consensus merger of $t_i$ and $T_{i+1}$
end for
return $T_1$
```

**Lemma 1.** [7] *Let $G$ be a graph. Then the following are equivalent: (1) $G$ is a subtree intersection graph; (2) $G$ is chordal; (3) $G$ admits a perfect elimination ordering.*

**Lemma 2.** *Suppose $\mathcal{E}(S)$ is correct and $T|SL(v)$ is accurate for each $v \in C$. Then, for each $i \leq n$, $T_i = T|\{v_i, ..., v_r\}$. Moreover, $T_1 = T|C$.*

*Proof.* The argument is similar to that in [8]. We include it for the sake of completeness. We proceed by induction on $i$. The claim is obvious for $i = r$. Assume $T_{i+1} = T|\{v_{i+1}, ..., v_r\}$. Observe that $\mathcal{L}(t_i) \cap \mathcal{L}(T_{i+1}) = X_i$, so $X_i$ is the leaf set of the backbone $Z$ of the merger of $t_i$ and $T_{i+1}$. As $t_i$ and $T_{i+1}$ are both correct, we know that there is no edge contraction in the merger, so we need only show that there are no collisions.

The only possible collision is the following. Suppose $e$ is an edge of $Z$, and both $v_i$ and a subtree $\tau$ of $T_{i+1}$ are attached at $e$. Clearly, $\mathcal{L}(\tau) \subseteq \{v_{i+1}, ..., v_r\} - X_i$. We will derive a contradiction to this fact. In the true tree $T$, $e$ corresponds to a path $P$ with endpoints, say, $a$ and $b$. Let $T_0$ denote the subtree of $T$ consisting of the internal nodes and edges of $P$ along with the subtrees attached at those nodes. Now, observe that (1) $v_i \in \mathcal{L}(T_0)$ and $\mathcal{L}(\tau) \subset \mathcal{L}(T_0)$. Furthermore, (2) we know $\mathcal{L}(T_0) \cap X_i = \emptyset$, just because $Z$, $t_i$ and $T_{i+1}$ are correct. Finally, we will prove below that (3) $\mathcal{E}(\mathcal{L}(T_0))$ is path connected.

By (3), let $\pi$ be a simple path in $\mathcal{E}(\mathcal{L}(T_0))$ from $v_i$ to a node in $\mathcal{L}(\tau)$, and let $x$ be the first node of $\pi$ which lies in $\mathcal{L}(\tau)$; that is, we may assume that

$$\pi = (v_{j_1} = v_i, v_{j_2}, ..., v_{j_k} = x)$$

with $v_{j_l} \notin \mathcal{L}(\tau)$ whenever $l < k$. By (2), we know that each $v_{j_l}$ is in $\{v_1, ..., v_i\}$. We claim now that there must be an edge $(v_i, x)$ in $\mathcal{E}(C)$. For suppose that $j_1 > ... > j_p$ and $j_{p+1} > j_p$. Then there must be an edge $(v_{j_{p-1}}, v_{j_{p+1}})$ in $\mathcal{E}(C)$ because $v_1, ..., v_n$ is a perfect elimination ordering. Hence, $v_{j_p}$ can be removed from $\pi$ without breaking the path. By induction on $k$, then, there must be an edge $(v_i, x)$ in $G$ as claimed. It follows that $x \in X_i$, which is a contradiction. Thus, there are no collisions, and the claim is proven. □

**Lemma 3.** *$\mathcal{E}(\mathcal{L}(T_0))$ is path-connected.*

*Proof.* Let $T_a$ denote the subtree of $T$ rooted at $a$ containing no internal nodes of $P$. Define $T_b$ similarly. Let $v \in \mathcal{L}(T_0)$. Since $\mathcal{E}(C)$ is path connected, let $\pi$ be a simple path from $v$ to a leaf of $T_a$, and let $x$ be the last node of $\mathcal{E}(C)$ along this path, so that $L(x)$ and $L(z)$ are edge-sharing for some $z \notin \mathcal{L}(T_a)$. Thus, if we take $(a, c)$ to be a terminal edge of $P$, we can see that $L(x)$ must contain a node $x'$ which lies in $\mathcal{L}(T_a)$ and such that $P(x, x')$ contains $(a, c)$. Let $y, y'$ and $(b, d)$ be the corresponding construction for $T_b$.

Suppose $u, v \in \mathcal{L}(T_0)$. Since $\mathcal{E}(C)$ is connected, there is a simple path $(u = w_1, w_2, ..., w_q = v)$ in $\mathcal{E}(C)$. Suppose $w_1, ..., w_j, w_{j+s+1} \in \mathcal{L}(T_0)$ and $w_{i+1}, ..., w_{i+s} \in \mathcal{L}(T_a)$. Then $L(w_j)$ and $L(w_{j+s+1})$ must be edge-sharing at $(a, c)$. We may, then, remove the excursion in $\mathcal{L}(T_a)$, obtaining the path $(w_1, ..., w_j, w_{j+s+1}, ..., w_q)$. Continuing in this manner, we remove from the path all excursions out of $\mathcal{L}(T_0)$. It follows that $\mathcal{E}(\mathcal{L}(T_0))$ is path connected. □

2.3. **Time complexity.** Suppose that $r$ is the largest radius of a leaf set $L(u)$ in a run of Algorithm 1, and let $f$ be the length of the shortest edge in the tree $T$. Then for every taxon $v$,

$$|SL(v)| \leq 2^{\frac{6r}{f}-1} = \kappa(n, f, k)$$

Thus, the base method for tree reconstruction is only deployed against $SL(v)$'s whose size is bounded by $\kappa(n, f, k)$. By the fast convergence analysis of our algorithm (section 3) it follows that for every tree our algorithm will reconstruct the whole topology for $r = O(\log n)$. On the other hand, for a typical tree (one drawn, for example, uniformly at random from the set of leaf-labelled trees) the algorithm will get the correct tree for $r = O(\log \log n)$, so the base method will be applied on trees of size $O(\log n)$.

Computing $\mathcal{E}(S)$ requires no more than $O(n^2\kappa^4)$ time, using, for example, a quartet based method. Moreover, at each iteration, at most $2\kappa$ many $SL(v)$'s are modified. A perfect elimination order of a chordal graph on $n$ vertices can be computed in $O(n^2)$ time, and computing the strict consensus merger of two trees takes $O(n)$ time. So every call of Algorithms 3 and 4 takes at most

$$n^2 \cdot \kappa^4 + 2\kappa \cdot \kappa^4 + O(n^2) + n \cdot O(n) = O(n^2\kappa^5)$$

Now, since there are at most $n^2$ iterations in Algorithm 1, the total running time is $O(n^4\kappa^5)$, in the typical case is $\tilde{O}(n^4)$.

Finally, we note that, for simplicity of presentation, the described algorithms are not optimized. Using hash tables to store the results of Algorithm 2 and the partial $T|SL(v)$ trees, each quartet is evaluated once along the coarse of the algorithm, and $T|SL(v)$ trees are built at each step on top of partially reconstructed topologies; it is not hard to show that the running time is $O(n^4)$ in the worst case.

## 3. Log-length sequences

In this section, we will prove that our method reconstructs almost all $n$-leaf trees provided that the sequence length $k$ is $O(poly(log(n)))$ under the Cavender-Farris-Neyman 2-state model of evolution [2, 6, 11]. More specifically, we argue that our method achieves the same performance guarantees as does the Dyadic Closure Method of [4]. A key notion in the analysis is the *depth* of a tree $T$, defined as follows: for an edge $e$ of $T$, let $T_1$ and $T_2$ be the rooted subtrees obtained by deleting $e$, and let $d_i(e)$ denote the topological distance from the root of $T_i$ to its nearest leaf in $T_i$; subsequently, we define

$$depth(T) = \max_e\{\max(d_1(e), d_2(e))\}$$

letting $e$ range over the set of internal edges of $T$. A quartet $\{i, j, k, l\}$ is called *short* if $T|\{i, j, k, l\}$ consists of a single edge connected to four disjoint paths of topological length no more than $depth(T) + 1$. Let $Q_{short}$ denote the set of short quartets of $T$. Given a set of quartets $Q$, we let $Q^*$ denote the set of the quartet topologies induced by $T$.

Given sequences $x, y$ of length $k$, let $h_{xy} = H(x, y)/k$ where $H(x, y)$ is the Hamming distance of the sequences. Let $E_{xy} = \mathbb{E}[h_{xy}]$

Let $Q_w$ denote the set of quartet topologies $q$ such that $h_{ij} \leq w$ for all $i, j \in q$. In [4], it is proved that if $Q^*_{short} \subseteq Q_w$ and $Q_w$ is consistent, then $cl(Q_w) = Q(T)$ where $cl(Q)$ is the dyadic closure of a set of quartet topologies. But observe that if $Q^*_{short} \subseteq Q_w \subseteq Q_{6w} \subseteq Q(T)$ for some $w$, then Algorithm 1 correctly reconstructs $T$. Let $E$ denote this event, and further, define the following events: $A$ for $Q^*_{short} \subseteq Q_w$; $B$ for $Q_{6w} \subseteq Q(T)$; and $C$ for "$Q_w$ contains all quartets containing pairs $i, j$ such that $E_{ij} < b$, and $Q_{6w}$ does not contain any pairs $i, j$ such that $E_{ij} > 13b$." If $i, j$ lie in a short quartet, then $E_{ij} \leq \frac{1-e^{-2g(2depth(T)+3)}}{2} = b$. We take $w = 2b$.

It's easy to see that

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

$$= \mathbb{P}[C] \cdot \mathbb{P}[A|C] \cdot \mathbb{P}[B|A, C] = \mathbb{P}[C] \cdot \mathbb{P}[B|C]$$

We will bound probability $\mathbb{P}[\overline{B}|C]$ first. Suppose $q = \{u, v, w, z\} \in \binom{n}{4}$ s.t. $\forall i, j \in q : E_{ij} \leq 13b$. Then, the quartet split of $q$ is found with probability at least $1 - \delta_1$ if:

(I) $(1 - 26b)\left(1 + \frac{2\epsilon}{1-\epsilon}\right) < \left(1 - \frac{2\epsilon}{1-\epsilon}\right) \Leftrightarrow \epsilon < \frac{13b}{2-13b}$

(II) $\frac{1}{\epsilon} = min_{i,j \in \{u,v,w,z\}}\left\{\frac{c(i,j)}{\alpha(k,\delta_1)}\right\} > 1$

If $k > \frac{8 \ln \frac{12}{\delta_1}(2-13b)^2}{(1-26b)^2(13b)^2}$, by the Azuma-Hoeffding inequality it follows that the probability that event $I \cap II$ does not hold is at most $6 \exp\{-\frac{(1-26b)^2 k}{8}\}$ so $\mathbb{P}[I \cap II] \geq 1 - \exp\{-\frac{(1-26b)^2 k}{8}\}$. Now, we can lower bound the probability of estimating quartet $q$ correctly as follows:

$$\mathbb{P}[\text{q is estimated correctly}] \geq 1 - \delta_1 - \mathbb{P}[\overline{II \cap I}] \geq 1 - \delta_1 - \exp\left\{-\frac{(1-26b)^2 k}{8}\right\}$$

Since the quartets are at most $\binom{n}{4}$ we can bound the probability of $\mathbb{P}[B|C]$ roughly as follows:

$$\mathbb{P}[B|C] \geq 1 - \binom{n}{4}\delta_1 - \binom{n}{4} \exp\left\{-\frac{(1-26b)^2 k}{8}\right\}$$

It remains to bound $\mathbb{P}[C]$. Define $S_r = \{\{i, j\} \mid h_{ij} < \frac{1}{2} - r\}$. Then, if $i, j$ are such that $E_{ij} \geq \frac{1}{2} - 13b$, then

$$\mathbb{P}[\{i, j\} \in S_{12b}] = \mathbb{P}[h_{ij} < \frac{1}{2} - 12b] \leq$$

$$\leq \mathbb{P}[h_{ij} - E_{ij} < \frac{1}{2} - 12b - E_{ij}] \leq \mathbb{P}[h_{ij} - E_{ij} \leq -b] \leq e^{-b^2k/2}$$

by the Azuma-Hoeffding inequality. A similar analysis shows that if $E_{ij} < \frac{1}{2} - 3b$, then $\mathbb{P}[\{i,j\} \notin S_{2b}] \leq e^{-b^2k/2}$. Thus, $\mathbb{P}[C] \geq 1 - \binom{n}{2}e^{-b^2k/2}$, and $\mathbb{P}[E]$ is not less than

$$1 - \binom{n}{4}\delta_1 - \binom{n}{4}\exp\left(-\frac{(1-26b)^2}{8}k\right) - \binom{n}{2}e^{-b^2k/2}$$

We have, therefore, proved

**Lemma 4.** *Suppose $k$ sites evolve on binary tree $T$ according to the Cavender-Farris-Neyman model, such that $f \leq D(e) \leq g$ for each edge $e$ of $T$. Then Algorithm 1 reconstructs $T$ with probability $1 - o(1)$ whenever*

$$k > \frac{c \cdot \ln \delta_1}{(1-26b)^2b^2} = \frac{c' \cdot \log n}{(1-26b)^2b^2}$$

$$\textit{and } \delta_1 \textit{ is chosen } \delta_1 < n^{-5}$$

*where $b = \frac{1-e^{-2g(2depth(T)+3)}}{2}$.*

In [4], it is also proven that a random $n$-leaf binary tree $T$ has

$$depth(T) \leq (2 + o(1)) \log\log 2n$$

with probability $1 - o(1)$. Thus,

**Theorem 4.** *Under the Cavender-Farris-Neyman model, Algorithm 2 correctly reconstructs almost all trees on $n$ leaves with sequences of length $k = O(poly(\log n))$.*

## 4. Experiments

Our methodology was as follows. We used the r8s package to produce edge-weighted, rooted trees, scaling the mutation probabilities recovered from the edge weights by the formula $d(e) = -\frac{1}{2}\log(1 - 2p(e))$. For each run of a given tree topology, a root sequence was selected uniformly at random. Sequences were evolved at the leaves according to edge mutation probabilities. We examined topologies with $n$ leaves, for $n = 10, 20, 30, 50$ and 100, and for each $n$, sequence lengths $k = 10, 40, 160$ and 2560.

For each topology $T$ and sequence length $k$, we evaluated our method against NJ and maximum parsimony (MP) routines (both drawn from the PAUP package).To evaluate the performance of NJ and MP, we simply computed the Robinson-Foulds (RF) distance between the model tree and that produced by the program. RF distance simply counts the number of both mottied and erroneous bipartitions. Notice, however, that RF distance is not immediately relevant to our method's product; therefore, we compute, for each subtree $T'$, the RF distance between $T'$ and $T|\mathcal{L}(T')$.

On the whole, the performance of our method seems to be similar to that of NJ and MP, in the simple terms of RF distance. (The reader is referred to the preliminary table below.) One must note, however, that for NJ and MP, RF distance indicates a number of erroneous edges–that is, false assertions. By contrast, for our method, RF distance is, essentially, a count of omitted edges (although a small number of errors remained in some simulations); our method makes few or no false claims. This is a significant advantage immediately, for the product inference can be trusted as a claim about the ancestry of the given taxa. Furthermore, and perhaps most compellingly, the product subtrees can be handed up to other methods for further analysis. For example, the methods of [10] can be used to infer sequences at ancestral nodes, permitting deeper reconstructions with assurance of quality.

## References

[1] Buneman, P. 1971 . The recovery of trees from measures of dissimilarity, 387395 . In *Mathematics in the Archaeological and Historical Sciences*, Edinburgh University Press, Edinburgh .

[2] Cavender, J. 1978. Taxonomy with confidence. *Mathematical Biosciences*, 40:271-280.

[3] Day, W. 1995. Optimal algorithms for comparing trees with labelled leaves. *J. Class.* 2, 7-28.

[4] Erdos, P., Steel, M., Szekely, L., Warnow, T. 1999. A few logs suffice to build (almost) all trees (part 1). *Random Structures and Algorithms*, 14(2):153-184.

[5] Erdos, P., Steel, M., Szekely, L., Warnow, T. 1999. A few logs suffice to build (almost) all trees (part 2). *Theoretical Computer Science*, 221:77-118.

[6] Farris, J. 1973. A probability model for inferring evolutionary trees. *Systematic Zoology*, 22:250-256.

[7] Golumbic , M. 1980. *Algorithmic Graph Theory and Perfect Graphs.* Academic Press, New York.

[8] Huson, D., Nettles, S., Warnow, T. 1999. Disk-Covering, A fast converging method for phylogenetic tree reconstruction. *Journal of Computational Biology*, 6:369-386.

[9] Mossel, E. Distorted metrics on trees and phylogenetic forests. 2005. (preprint).

[10] Mossel, E. Phase Transitions in Phylogeny. 2004. *Trans. Amer. Math. Soc.* 356 no.6 2379–2404. (electronic)

[11] Neyman, J. 1971. Molecular studies of evolution: a source of novel statistical problems. In *Statistical Decision Theory and Related Topics*, S. Gupta and J Yackel (ed.) Academic Press, New York.

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

[13] Usman, R., Moret, B., Warnow, T., Williams, T. 2004. Rec-I-DCM3: A fast algorithmic technique for reconstructing large phylogenetic trees. Proc. IEEE Computer Society Bioinformatics Conference CSB 2004, Stanford Univ.

### Results of our method applied to $n$ taxa with sequences of length $k$.

| | $k$ | | | | |
|---|---|---|---|---|---|
| $n$ | 10 | 40 | 160 | 640 | 2560 |
| 10 | 5(0) | 1(2) | 1(2) | 1(2) | 1(2) |
| 20 | 11(0) | 3(2) | 3(2) | 3(2) | 3(2) |
| 30 | 16(1,2) | $6(1^2, 2^2)$ | 4(1,2) | 4(1,2) | 4(1,2) |
| 50 | $29(2.5^2)$ | $10(1^2, 3^2)$ | $7(1^5)$ | $2(1^2)$ | 7(0) |
| MP | | | 6.5 | | |
| NJ | | | 7.5 | | |
| 100 | $81(1^3)$ | $13(2^3, 3^3, 4^2)$ | $7(1^5)$ | $2(1^2)$ | 7(0) |
| MP | 50 | | | | |
| NJ | 70 | 21.5 | | | |

Figure 1: $i(j^k)$ indicates that that our method produced $i$ trees, $k$ of which had RF distance $j$. Some results of MP and NJ are given on the same model tree are given.