Fast and Reliable Reconstruction of Phylogenetic Trees
with Very Short Edges
Extended Abstract

Ilan Gronau*   Shlomo Moran†   Sagi Snir‡

Abstract

Phylogenetic reconstruction is the problem of reconstructing an evolutionary tree from sequences corresponding to leaves of that tree. A central goal in phylogenetic reconstruction is to be able to reconstruct the tree as accurately as possible from as short as possible input sequences. The sequence length required for correct topological reconstruction depends on certain properties of the tree, such as its depth and minimal edge-weight. Fast converging reconstruction algorithms are considered state-of the-art in this sense, as they require asymptotically minimal sequence length in order to guarantee (with high probability) correct topological reconstruction of the entire tree. However, when the original phylogenetic tree contains very short edges, this minimal sequence-length is still too long for practical purposes. Short edges are not only very hard to reconstruct; their presence may also prevent the correct reconstruction of long edges.

In this paper we present a fast converging reconstruction algorithm which returns a partially resolved topology containing *all* edges of the original tree whose weight exceeds some (non-trivial) lower bound, which is determined by the input sequence length, as well as some properties of the tree, such as its depth. It does not depend, however, on the minimal edge-weight. This lower bound provides a partial reconstruction guarantee which is strictly stronger than the guarantees given by other fast converging algorithms. Our algorithm also has optimal complexity (linear space and quadratic-time) which, together with its partial reconstruction guarantee, makes it appealing for practical use.

1 Introduction.

Phylogenetic reconstruction methods attempt to find the evolutionary history of a given set of extant species (taxa). This history is usually described by an edge-weighted tree whose internal vertices represent past speciation events (extinct species) and whose leaves correspond to the given set of taxa. The amount of evolutionary change between two subsequent speciation events is indicated by the (positive) weight of the edge connecting them. The main objective of phylogenetic reconstruction is to reconstruct the *topology* of the tree given some data extracted from the taxa. The topology is essentially an unweighted version of the tree. Most reconstruction methods assume that the topology of the evolutionary tree is *fully resolved*, meaning that the tree is binary (i.e. each internal vertex is of degree 3). An algorithm is said to correctly reconstruct an edge of the tree if the topology it outputs contains an edge inducing the same split on the set of taxa.

A major challenge in phylogenetic reconstruction is the task of reconstructing the topology of the original tree as accurately as possible, given access to as little data as possible. This data is often given in the form of sequences of uniform length k (e.g. DNA sequences), each associated with a different taxon. In this case we would like to use sequences which are as short as possible. It was already shown in [12] that assuming arbitrary constant lower and upper bounds on edge-weights, all trees may be reconstructed correctly with high probability from sequences of polynomial length in the number of taxa (n). Methods which achieve this are often referred to as *fast converging* reconstruction methods. In [18] it was shown that this is optimal in the sense that it is impossible to reconstruct the entire tree with high probability from sequences shorter than some polynomial of n. In [18, 11] it is shown that by assuming an additional (specific) upper bound on edge weights, sequences of logarithmic length suffice.

Much attention has been focused on fast converging reconstruction methods in recent years [12, 13, 15, 6, 7, 16, 18, 19, 10]. These algorithms use distance-based methods which rely only on small entries of the input matrix. There are two types of fast converging algorithms: *absolute* fast converging algorithms do not require any prior knowledge about the original tree (see algorithms in [12, 13, 7, 10], and DCM-Buneman in [15]); *relative* fast converging algorithms require some prior knowledge on the original tree, such as its depth or a lower bound on edge-weights (see algorithms in [8, 6, 16, 19], and DCM-NJ from [15]). A technique for converting a relative fast converging algorithm to an absolute fast converging algorithm is presented in [17]. This technique is very general, but it increases the time complexity of the algorithm by $\Omega(n^4)$. Therefore, efficient absolute fast converging algorithms remain of

*Technion - Israel Institute of Technology, Haifa, 32000Israel
†Technion - Israel Institute of Technology, Haifa, 32000Israel
‡Netanya Academic College, Netanya, Israel

much interest.

The major drawback of almost all fast converging algorithms (relative and absolute) is that they provide a meaningful reconstruction guarantee only when the input sequences are long enough. The required sequence length depends on two parameters of the original tree: its minimal edge-weight and some notion of tree-depth. Consequently, when the tree contains some very short edges, we may be unable to obtain sequences of the required length, in which case these algorithms guarantee nothing about the correctness of their output. Recent works [19, 10] suggest algorithms which instead of returning a fully resolved topology, return a forest of edge-disjoint trees which is consistent (with high probability) with the original tree. The extent of topological resolution these algorithms provide depends on the sequence length as well as the minimal edge-weight of the original tree, but is independent on the depth of the tree.

An inherent limitation of these algorithms is that they must return a forest of *fully resolved* trees. The following example demonstrates how this may prevent them from reconstructing long edges in the presence of some short edges. Consider a full binary tree over $4n$ leaves obtained by taking n 4-leaf trees with an arbitrarily short internal edge, and attaching them (in the middle of the internal edge) to n leaves of a full binary tree whose edges have some (reasonable) fixed weights. Now assume that the input sequences are too short to ensure reliable resolution of the quartets corresponding to the 4-leaf trees. Then, any method which insists on returning a forest of fully resolved trees, is prevented from resolving *any* of the internal edges of the tree.

In this paper we cope with the above problem by devising an algorithm which is allowed to contract short edges, but it guarantees correct reconstruction (with high probability) of all "sufficiently long" edges. In particular, we guarantee two important things: (1) the returned topology contains no false splits (i.e. zero false-positive rate), and (2) the unreconstructed splits correspond only to "short" edges of the original tree. Our algorithm maintains a zero false-positive rate by contracting any low-confidence edge it encounters and introducing high degree vertices as a result. This approach is similar in a sense to the one taken by Buneman's classic reconstruction method [3, 1], but unlike that method, our method is fast converging (in fact it is *absolute* fast converging) and guarantees reconstruction of much shorter edges. In addition, our algorithm is faster than algorithms which construct Buneman trees (or refined Buneman trees).

The algorithm presented in this paper follows the incremental approach which was introduced in [2], and

applied to fast converging algorithms in [8, 7, 16]. In this approach, the topology is constructed by attaching the taxa one by one. In each iteration of such an algorithm, it attempts to find the correct place in the current topology to insert the new taxon. This is typically done by querying the vertices of the current topology as to the direction (relative to that specific vertex) in which the new taxon should be attached. When a vertex in the current topology corresponds to a vertex of the original tree, such directional queries can be simply answered by querying quartet splits (see [8, 7]). However, the current topology our incremental algorithm holds may contain vertices which correspond to non-trivial (contracted) subtrees of the original tree. This significantly complicates the task of reliably answering a directional query. We present an efficient method which does this without violating the optimal time complexity ($O(n^2)$) of the algorithm. Our *directional oracle* is partial in the sense that it may fail to return a direction when it is not confident enough in the answer. In such a case, the construction of false edges is avoided by contracting some of the previously constructed edges.

In the next section we provide the notations used in the paper. In Sections 3-6 we present our incremental algorithm in a 'top-down' fashion. In Section 3 we outline our algorithm and present the main concept of *partial directional oracle* which is a model-independent primitive for constructing phylogenetic trees in the presence of noisy information. In Section 4 we provide the main inductive argument for bounding (from above) the weight of edges our algorithm contracts. In Section 5 we present an implementation of a directional oracle which uses a reliable *quartet oracle*. In Section 6 we show how to implement our algorithm using noisy pairwise-distance estimates. In that section we provide the main theorem which describes the relation between the amount of noise in the input dissimilarities and the reconstruction guarantees of our algorithm. In Section 7 we analyze the time and space complexities of our algorithm, and in Section 8 we discuss the performance of our algorithm on a known model of sequence evolution. In that section we establish the relation between the length of input sequences and the weight of contracted edges. We conclude that section by generalizing the notion of fast convergence, and showing that our algorithm satisfies this stronger variant of fast convergence. Due to space limitations, some of the proofs are omitted from this extended abstract.

2   Definitions and Notations.

Trees: A tree T is an undirected connected and acyclic graph. $V(T)$ and $E(T)$ denote the sets of vertices and edges of T, respectively. $leaves(T)$ denotes the

leaf-set of T, and $internal(T) = V(T) \setminus leaves(T)$ denotes the set of its internal vertices. For a vertex $v \in V(T)$, the *neighborhood* of v, $N_T(v)$, is the set of vertices adjacent to v in T. The neighborhood of a subset $U \subseteq V$ is defined by $N_T(U) = \cup_{u \in U} N(u) \setminus U$. The degree of a vertex, $deg_T(v)$, is the size of its neighborhood in T. The *parent* of a leaf x in T, $parent_T(x)$, is the unique vertex in $N_T(x)$. T is said to be a *phylogenetic tree* over taxon-set S if $leaves(T) = S$, the degree of every internal vertex is at least 3, and every edge $e \in E(T)$ is associated with a strictly positive weight $w(e)$. The weight function w induces a metric $D_T = \{d_T(u,v)\}_{u,v \in V(T)}$ over $V(T)$, s.t. $d_T(u,v)$ is the total weight of $path_T(u,v)$ – the (unique) simple path connecting u and v in T. The diameter of T ($diam(T)$) is the maximum weight of a simple path in T.

A *subtree* of a tree T is a connected subgraph of T. The notion of distances is generalized for subtrees as follows: for two vertex disjoint subtrees t_1, t_2 of T, $d_T(t_1, t_2)$ denotes the total weight of $path_T(t_1, t_2)$, which is the (unique) shortest path in T connecting a vertex in t_1 and a vertex in t_2. Let t_1, t_2, t_3 be mutually disjoint subtrees of T. We say that t_2 *separates* t_1 from t_3 if $path_T(t_1, t_3)$ intersects t_2. If t_2 does not separate t_1 from t_3 we say that t_1 *and* t_3 *are on the same side of* t_2. In general, we use lower case t's to denote subtrees of a tree T, and whenever the tree T is clear from context, the subscript T may be removed from the corresponding notation.

Induced sub-phylogenies: Let T be a phylogenetic tree over a set of taxa S, and let $S' \subseteq S$. $T(S')$, the phylogenetic tree induced by T on S', is obtained by taking the minimal subtree of T which spans S', and removing all degree-2 vertices by iteratively replacing the two edges which touch such a vertex with a single edge. Note that every vertex in $V(T(S'))$ corresponds to a vertex of T and every edge in $E(T(S'))$ corresponds to a simple path in T. Edge-weights in $T(S')$ are determined according to the weight of corresponding paths in T. Therefore, $T(S')$ induces a sub-metric to the metric D_T induced by T, and D_T is used to describe distances induced by all sub phylogenies of T.

Internal Edge Contraction: The contraction of an edge $e = (u, u') \in E(T)$ replaces e with a single vertex v s.t. $N(v) = N(\{u, u'\})$. In such a case we say that edge e was contracted into vertex v. $\widehat{T}$ is said to be an *internal (edge) contraction* of T if it is obtained from T by a series of contractions of internal edges. Note that if $\widehat{T}$ is an internal contraction of T then $leaves(T) = leaves(\widehat{T})$. All edge contractions considered in this paper are internal. $\widehat{T}$ is said to be an ε-contraction of T if each (internal) edge in

T which is contracted in $\widehat{T}$ has weight at most ε. An internal contraction of T to $\widehat{T}$ induces a mapping between vertices of $internal(\widehat{T})$ and (vertex disjoint) subtrees of $internal(T)$: each vertex $\hat{v}$ of $internal(\widehat{T})$ either corresponds to a vertex of $internal(T)$, or to a subtree consisting of the internal edges contracted into $\hat{v}$. The subtree of T corresponding to a vertex $\hat{v}$ of $\widehat{T}$ is denoted by $t_{\hat{v}}$. Edge weights in a contraction $\widehat{T}$ of T are determined by the weight of the corresponding edges in T. This means that for neighboring vertices $\hat{u}, \hat{v}$ in $\widehat{T}$, we have $d_{\widehat{T}}(\hat{u}, \hat{v}) = d_T(t_{\hat{u}}, t_{\hat{v}})$, but otherwise, $d_T(t_{\hat{u}}, t_{\hat{v}})$ is generally larger than $d_{\widehat{T}}(\hat{u}, \hat{v})$.

We complete this section with two simple observations which are used later (often implicitly):

LEMMA 2.1. *Let t be a subtree of T s.t. $V(t) \subseteq internal(T)$. Then the tree $\widehat{T}$ obtained from T by replacing t with a single vertex $\hat{v}$ such that $N_{\widehat{T}}(\hat{v}) = N_T(V(t))$ is an edge contraction of T. If in addition the weights of all edges in t are at most ε, then $\widehat{T}$ is an ε-contraction of T.*

LEMMA 2.2. *[transitivity of ε-contraction] Let T_1, T_2, T_3 be three phylogenetic trees over S. If T_1 is an ε-contraction of T_2 and T_2 is an ε-contraction of T_3, then T_1 is an ε-contraction of T_3.*

3 A General Overview of The Algorithm.

Our incremental reconstruction algorithm works in iterations, where the objective in each iteration is to extend an edge contraction of $T(S')$ for some $S' \subseteq S$ to an edge-contraction of $T(S' \cup \{x\})$ for some $x \in S \setminus S'$.

Procedure *Incremental_Reconstruct(S)*:

- Select $x_0, x_1 \in S$,

- Initialize: $\widehat{T} \leftarrow (x_0, x_1)$; $S' \leftarrow \{x_0, x_1\}$.

- While $S' \neq S$ do

  1. Select a taxon $x \in S \setminus S'$ and set $S' \leftarrow S' \cup \{x\}$.

  2. *Attach_Taxon($\widehat{T}, x$)*

The crucial part of this process is pinpointing the *anchor* of x, which is the location of $parent(x)$ in the current topology, defined below:

DEFINITION 3.1. *Let $\widehat{T}$ be an edge contraction of $T(S')$, and let $x \in S \setminus S'$. The anchor of x in $\widehat{T}$, $anchor_{\widehat{T}}(x)$, is defined as follows: Let p_x be the parent of x in $T(S' \cup \{x\})$. If p_x is included in $t_{\hat{v}}$ for some $\hat{v} \in V(\widehat{T})$ (either by being a vertex in $t_{\hat{v}}$ or by lying on a path in T which corresponds to an edge of $t_{\hat{v}}$) then $anchor_{\widehat{T}}(x) = \hat{v}$. Otherwise, $anchor_{\widehat{T}}(x)$ is the unique edge $(\hat{u}, \hat{v}) \in E(\widehat{T})$ for which p_x is an internal vertex in $path_T(t_{\hat{u}}, t_{\hat{v}})$.*

The anchor of x in $\widehat{T}$ is found by querying vertices in $\widehat{T}$ for the location of x. These queries are posed to a *directional oracle* which receives the new taxon x and a vertex $\hat{v}$ of $\widehat{T}$ and is expected to output a neighbor $\hat{u}$ of $\hat{v}$, which indicates the direction of x with respect to $\hat{v}$.

DEFINITION 3.2. (PARTIAL DIRECTIONAL ORACLE) *Let T be a phylogenetic tree over S. A* partial directional oracle *for T is a function $PDO = PDO_T$ which receives queries of the form $(\widehat{T}, \hat{v}, x)$, where*

- $\widehat{T}$ *is an edge-contraction of $T(S')$, for some $S' \subset S$,*

- $\hat{v} \in V(\widehat{T})$,

- $x \in S \setminus S'$,

and outputs either a vertex $\hat{u} \in N_{\widehat{T}}(\hat{v})$ or 'null'. The output must satisfy the following requirements:

- *If $\hat{v} \in leaves(\widehat{T})$, then $PDO(\widehat{T}, \hat{v}, x) = parent_{\widehat{T}}(\hat{v})$.*

- *If $PDO(\widehat{T}, \hat{v}, x) = \hat{u}$, then $t_{\hat{u}}$ and x are on the same side of $t_{\hat{v}}$ in $T(S' \cup \{x\})$ (this property is also referred to as the* truthfulness *of the oracle). We sometimes abuse notation and simply say that $\hat{u}$ is on the same side of $\hat{v}$ as x.*

Our algorithm uses the partial directional oracle to compute the *insertion zone* of x in $\widehat{T}$ as follows:

DEFINITION 3.3. (INSERTION ZONE) *Given an edge-contraction $\widehat{T}$ of $T(S')$, some taxon $x \in S \setminus S'$, and a partial directional oracle PDO, the* insertion zone of x in $\widehat{T}$, *denoted by $\hat{t}_{inz}(\widehat{T}, x, PDO)$ (or simply $\hat{t}_{inz}$ when $\widehat{T}, x$ and PDO are clear from context), is the subtree of $\widehat{T}$ defined by the following procedure:*

Procedure Find_Insertion_Zone($\widehat{T}, x, PDO$):

1. $\hat{t}_{inz} \leftarrow \widehat{T}$

2. For every edge $(\hat{u}, \hat{v}) \in E(\hat{t}_{inz})$ s.t. $PDO(\widehat{T}, \hat{v}, x) = \hat{u}$ do: Delete from $\hat{t}_{inz}$ all the vertices which are separated from $\hat{u}$ by $\hat{v}$.

A simple inductive argument using the truthfulness of PDO implies that the above procedure returns a subtree of $\widehat{T}$ which includes $anchor_{\widehat{T}}(x)$.

OBSERVATION 3.1. $\hat{t}_{inz} = \hat{t}_{inz}(\widehat{T}, x, PDO)$ *is a subtree of $\widehat{T}$ which satisfies the following:*

1. $\hat{t}_{inz}$ *includes $anchor_{\widehat{T}}(x)$ (as an edge or a vertex).*

2. *For each leaf $\hat{v}$ of $\hat{t}_{inz}$, $PDO(\widehat{T}, \hat{v}, x) = parent_{\hat{t}_{inz}}(\hat{v})$.*

3. *For each internal vertex $\hat{v}$ of $\hat{t}_{inz}$ (if any), $PDO(\widehat{T}, \hat{v}, x) = $ 'null'.*

We conclude this general overview by describing how to attach x to $\widehat{T}$ given its insertion zone.

Procedure Attach_Taxon($\widehat{T}, x$):

1. $\hat{t}_{inz} \leftarrow \hat{t}_{inz}(\widehat{T}, x, PDO)$.

2. If $\hat{t}_{inz}$ is a single edge $(\hat{u}, \hat{v})$, then attach x to $\widehat{T}$ by introducing a new internal vertex $\hat{p}_x$ and replacing $(\hat{u}, \hat{v})$ with the three edges $(\hat{u}, \hat{p}_x), (\hat{v}, \hat{p}_x), (x, \hat{p}_x)$.

3. If $\hat{t}_{inz}$ has a single internal vertex $\hat{v}$ (i.e. $V(\hat{t}_{inz}) = N(\hat{v}) \cup \{\hat{v}\}$), then add to $\widehat{T}$ the edge $(\hat{v}, x)$.

4. Else (i.e. $\hat{t}_{inz}$ has at least one internal edge), contract all internal edges of $\hat{t}_{inz}$ into a new vertex $\hat{v}$ and add to $\widehat{T}$ the edge $(\hat{v}, x)$.

---

4 ε-Reliable Directional Oracles.

In this section we present the concept of ε-reliability, which is used for bounding the weight of edges contracted by our algorithm. Consider a single iteration of the algorithm, in which a taxon $x \in S \setminus S'$ is inserted to the current topology $\widehat{T}$ over S'. We consider an intermediate topology $\widehat{T}^{+x}$, which is the natural extension of $\widehat{T}$ to a contraction of $T(S' \cup \{x\})$.

DEFINITION 4.1. *Let $\widehat{T}$ be a contraction of $T(S')$ and let $x \in S \setminus S'$. $\widehat{T}^{+x}$ is the extension of $\widehat{T}$ to a contraction of $T(S' \cup \{x\})$ defined as follows:*

- *If the anchor of x in $\widehat{T}$ is a vertex $\hat{v} \in V(\widehat{T})$ then $\widehat{T}^{+x}$ is obtained by adding an edge $(\hat{v}, x)$ to $\widehat{T}$.*

- *Otherwise, the anchor of x in $\widehat{T}$ is an edge $(\hat{u}, \hat{v}) \in E(\widehat{T})$, and $\widehat{T}^{+x}$ is obtained by replacing the edge $(\hat{u}, \hat{v})$ with the three edges $(\hat{u}, \hat{p}_x), (\hat{p}_x, \hat{v}), (\hat{p}_x, x)$.*

A simple case analysis implies the following lemma:

LEMMA 4.1. *If $\widehat{T}$ is an ε-contraction of $T(S')$ then $\widehat{T}^{+x}$ is an ε-contraction of $T(S' \cup \{x\})$.*

Let $\widehat{T}_{post}$ be the topology resulting from the application of $Attach\_Taxon(\widehat{T}, x)$. We show that $\widehat{T}_{post}$ is an internal edge contraction of $\widehat{T}^{+x}$ by considering $\hat{t}_{inz}^{+x}$ – the minimal subtree of $\widehat{T}^{+x}$ which includes x and $\hat{t}_{inz}$.

LEMMA 4.2. *$\widehat{T}_{post}$ is obtained from $\widehat{T}^{+x}$ by contracting the internal edges (if any) of $\hat{t}_{inz}^{+x}$.*

We conclude this section by stating the conditions under which $\widehat{T}_{post}$ is an ε-contraction of $T(S' \cup \{x\})$.

DEFINITION 4.2. (ε-ENVIRONMENT) $\hat{t}_{env}(\widehat{T}, x, \varepsilon)$, the ε-environment of x in $\widehat{T}$, is the maximal subtree of $\widehat{T}^{+x}$ which includes x and whose internal edges have weight at most ε.

DEFINITION 4.3. (ε-RELIABILITY) A partial directional oracle PDO is said to be ε-reliable for $(\widehat{T}, x)$, if the insertion zone of x determined by PDO is included in the ε-environment of x, i.e.: $V(\hat{t}_{inz}) \subseteq V(\hat{t}_{env}(\widehat{T}, x, \varepsilon))$.

LEMMA 4.3. If the partial directional oracle PDO used in the calculation of $\hat{t}_{inz}$ is ε-reliable for $(\widehat{T}, x)$, then $\widehat{T}_{post}$ is an ε-contraction of $T(S' \cup \{x\})$.

Lemma 4.3 provides the main inductive argument used in the proof of our main result (Theorem 6.1). The validity of this argument requires that our partial directional oracle is ε-reliable in every iteration of the algorithm.

5   An Efficient Implementation of The Partial Directional Oracle Using Quartet Queries.

In this section we present our partial directional oracle and give explicit conditions on $\widehat{T}$ and x under which it is ε-reliable. Our directional oracle PDO uses queries on *quartet splits*. A taxon-quartet $q = \{a, b, c, d\} \subseteq S$ defines an induced sub-topology $T(q)$ which is either a star or a split $(x, y; z, w)$, where $\{x, y, z, w\} = q$, and $T(q)$ has a single internal edge separating $\{x, y\}$ from $\{z, w\}$. Reconstructing phylogenetic trees from quartet splits dates back as far as [3]. Under recently studied models of evolution, the reliability of quartet topologies inferred from estimated distances decreases dramatically when the diameter of the quartet becomes large [12, 19, 10]. This motivates the definition of an (r, ε)-*reliable quartet oracle*.

DEFINITION 5.1. (PARTIAL QUARTET ORACLE) Let T be a phylogenetic tree over S. A partial quartet oracle for T is a function which receives a quartet $q \subseteq S$ and returns either the (correct) split of q in T, or 'null'.
A quartet oracle is said to be (r, ε)-reliable if it returns the quartet split whenever given a quartet q, s.t. $diam(T(q)) \leq r$ and the single internal edge of $T(q)$ has weight greater than ε.

Recall that a partial directional oracle, when queried on a vertex $\hat{v}$ and taxon x, returns either 'null' or a neighbor $\hat{u}$ of $\hat{v}$ which is on the same side of $\hat{v}$ as x. The procedure described below seeks such a neighbor via a series of queries to a partial quartet oracle PQO. These quartet queries include the new taxon x and three other taxa s_1, s_2, s_3 of $\widehat{T}$ which represent three different

directions corresponding to three different neighbors of $\hat{v}$ in $\widehat{T}$, defined as follows:

DEFINITION 5.2. (DIRECTIONAL REPRESENTATIVES) Let $(\hat{u}, \hat{v})$ be an edge in $\widehat{T}$. A taxon s of $\widehat{T}$ is a valid directional representative of $(\hat{v} \rightarrow \hat{u})$ if $\hat{u} \in path_{\widehat{T}}(\hat{v}, s)$. The directional representative of $(\hat{v} \rightarrow \hat{u})$ is denoted by $s_{\hat{v}}(\hat{u})$.

The Partial Directional Oracle – $PDO(\widehat{T}, \hat{v}, x)$:

1. Initialize candidate set $C \leftarrow N_{\widehat{T}}(\hat{v})$.

2. If $C = \{\hat{u}\}$ ($\hat{v}$ is a taxon), return $\hat{u}$.

3. Otherwise ($|C| \geq 3$), proceed as follows:

4. **Triplets Tournament:**
   While $|C| > 1$ do:

   - If $|C| = 2$, then $C \leftarrow C \cup \{\hat{u}\}$, for some $\hat{u} \in N_{\widehat{T}}(\hat{v}) \setminus C$.

   - Select some triplet $\{\hat{u}_1, \hat{u}_2, \hat{u}_3\} \subseteq C$ and invoke $PQO(\{x, s_{\hat{v}}(\hat{u}_1), s_{\hat{v}}(\hat{u}_2), s_{\hat{v}}(\hat{u}_3)\})$.
     - If output is 'null', then remove $\hat{u}_1, \hat{u}_2, \hat{u}_3$ from C.
     - Otherwise, the output is $(x, s_{\hat{v}}(\hat{u}_i) ; s_{\hat{v}}(\hat{u}_j), s_{\hat{v}}(\hat{u}_k))$ (where $\{i, j, k\} = \{1, 2, 3\}$), then remove $\hat{u}_j, \hat{u}_k$ from C.

   If the tournament results in $C = \emptyset$, return 'null'.

5. **Validation:** $C = \{\hat{u}\}$ for some $\hat{u} \in N_{\widehat{T}}(\hat{v})$.

   (a) Select some vertex $\hat{u}_1 \in N_{\widehat{T}}(\hat{v}) \setminus \{\hat{u}\}$.

   (b) For every $\hat{u}_2 \in N_{\widehat{T}}(\hat{v}) \setminus \{\hat{u}, \hat{u}_1\}$, invoke $PQO(\{x, s_{\hat{v}}(\hat{u}), s_{\hat{v}}(\hat{u}_1), s_{\hat{v}}(\hat{u}_2)\})$.
      - If output is $(x, s_{\hat{v}}(\hat{u}) ; s_{\hat{v}}(\hat{u}_1), s_{\hat{v}}(\hat{u}_2))$, then continue.
      - Otherwise, stop and return 'null'.

   (c) Return $\hat{u}$ (if it survived all rounds).

Our partial directional oracle contains two main phases:

Triplets Tournament: In this phase, the set of all possible directions (represented by $N_{\widehat{T}}(\hat{v})$) is iteratively screened to end up with at most one candidate direction. In each iteration a quartet is queried and as a result at least two directions are eliminated from the set of candidates. If the tournament results in an empty candidate set, then the directional oracle returns 'null'. Otherwise, the tournament results in a single surviving candidate. The following validation phase is needed to guarantee that the surviving candidate (if any) indeed indicates the correct direction.

Validation: Validation of the direction represented by the surviving neighbor $\hat{u}$ is done by another series of quartet queries which contain both x and $s_{\hat{v}}(\hat{u})$. If all quartet queries *positively* validate this direction (meaning that they put x and $s_{\hat{v}}(\hat{u})$ on the same side of the split), then $\hat{u}$ is returned. Otherwise, the directional oracle returns *'null'*.

LEMMA 5.1. *Assuming that PQO is a partial quartet oracle for T and all directional representatives are valid (Definition 5.2), then procedure PDO described above is a (truthful) partial directional oracle for T.*

Proof. Consider a valid input instance $(\widehat{T}, \hat{v}, x)$ for PDO. If $\hat{v}$ is a taxon then PDO returns the unique neighbor of $\hat{v}$ in $\widehat{T}$, as required. So assume $\hat{v}$ is an internal vertex of $\widehat{T}$. It is sufficient to show that for any vertex $\hat{u} \in N_{\widehat{T}}(\hat{v})$ which is *not* on the same side of $\hat{v}$ as x, there exists a vertex which fails $\hat{u}$ at step 5b of the validation phase (when chosen either as $\hat{u}_2$ or as $\hat{u}_1$). If there is a vertex $\hat{u}' \in N(\hat{v})$ which is on the same side of $\hat{v}$ as x, then $\hat{u}'$ fails the validation of $\hat{u}$. If there is no such vertex, then p_x, the parent of x in $T(leaves(\widehat{T}) \cup \{x\})$, is contained in $t_{\hat{v}}$. In this case, for every choice of $\hat{u}_1$ there is $\hat{u}_2 \notin \{\hat{u}, \hat{u}_1\}$ s.t. p_x lies on $path_T(t_{\hat{u}_1}, t_{\hat{u}_2})$. Hence x and $s_{\hat{v}}(\hat{u})$ are not on the same side of the split induced by T on $\{x, s_{\hat{v}}(\hat{u}), s_{\hat{v}}(\hat{u}_1), s_{\hat{v}}(\hat{u}_2)\}$, and the validation of $\hat{u}$ fails, as claimed. ∎

The following lemma states the cases in which PDO is *guaranteed* to return the correct direction.

LEMMA 5.2. *Given a contraction $\widehat{T}$ of $T(S')$, a vertex $\hat{v} \in internal(\widehat{T})$ and taxon $x \in S \setminus S'$, assume that the following holds:*

1. *The directional representatives $\{s_{\hat{v}}(\hat{u}') : \hat{u}' \in N(\hat{v})\}$ are valid (Definition 5.2).*

2. *$\hat{v}$ has a neighbor $\hat{u}$ in $\widehat{T}$ which is on the same side of $\hat{v}$ as x.*

3. *PQO is an (r, ε)-reliable quartet oracle for T, where r, ε satisfy the following:*

   (a) *$\varepsilon < \min\{d_T(t_{\hat{v}}, t_{\hat{u}}), d_T(t_{\hat{v}}, p_x)\}$, where p_x is the parent of x in $T(S' \cup \{x\})$.*

   (b) *$r \geq \max\{d_T(y, z)\}$, for all taxon-pairs $\{y, z\} \subset \{x\} \cup \{s_{\hat{v}}(\hat{u}') : \hat{u}' \in N(\hat{v})\}$.*

Then $PDO(\widehat{T}, \hat{v}, x) = \hat{u}$.

Proof. Consider all taxon-quartets queried by PDO of type $q = \{x, s, s_1, s_2\}$, where s is the directional representative of $(\hat{v} \rightarrow \hat{u})$ in $\widehat{T}$. By condition 3b we

have that $diam(T(q)) \leq r$. It is easy to see that the path in T corresponding to the internal edge of $T(q)$ intersects $t_{\hat{v}}$, and in addition it either contains p_x or intersects $t_{\hat{u}}$. Hence, by condition 3a above, the weight of this path is greater than ε. Therefore, by the (r, ε)-reliability of PQO we get that $PQO(q) = (x, s \; ; \; s_1, s_2)$, which implies that $\hat{u}$ survives all rounds of the triplets tournament and the validation phase. ∎

We conclude this section by determining the values of r, for which the (r, ε)-reliability of PQO implies the ε-reliability of PDO for given $(\widehat{T}, x)$. In order to establish the ε-reliability of PDO, we must show that the insertion zone of x in $\widehat{T}$ is contained in the ε-environment of x in $\widehat{T}$. By Observation 3.1(1,3), this can be proven by showing that $PDO(\widehat{T}, \hat{v}, x) \neq$ *'null'* for every leaf $\hat{v}$ of the ε-environment. The following is therefore directly implied by Lemma 5.2 and the above discussion:

COROLLARY 5.1. *Let $\widehat{T}$ be an internal contraction of $T(S')$ for some $S' \subset S$, and let x be a taxon in $S \setminus S'$. Assume that for every $\hat{v} \in leaves(\hat{t}_{env}(\widehat{T}, x, \varepsilon))$ we have:*

1. *$d_T(t_{\hat{v}}, x) \leq r_1$.*

2. *$diam(t_{\hat{v}}) \leq r_2$.*

3. *$\forall \hat{u} \in N_{\widehat{T}}(\hat{v}) : d(t_{\hat{v}}, s_{\hat{v}}(\hat{u})) \leq r_3$.*

If PQO is (r, ε)-reliable for $r \geq r_2 + r_3 + \max\{r_1, r_3\}$, then PDO is ε-reliable for $(\widehat{T}, x)$.

Discussion: The reconstruction guarantees of our algorithm (see Theorem 6.1) largely depend on an upper bound on r mentioned in Corollary 5.1 above. The value of r_3 is bounded using the *depth* of T (to be defined in the next section). The values of r_1 and r_2, on the other hand, are bounded using also the ε-*diameter* of T, which is the maximal weight of a path in T which consists only of edges whose weight is at most ε (see Definition 6.2). While it is reasonable to assume that, for small values of ε, the ε-diameter of T is small, in some extreme cases it can still be much larger than the depth. The question whether it is possible to bound r linearly in the depth alone (regardless the value of the ε-diameter) is hence probably of small practical interest, but it has some interesting theoretical implications (see concluding discussion in Section 8). In the full version of this paper we present a variant of our directional oracle which achieves this goal.

6 Applying the Algorithm on Noisy Tree Metrics.

In this section we specify conditions which guarantee that our partial directional oracle is ε-reliable in every

iteration of the incremental algorithm. As commonly done in phylogenetic reconstruction, we assume that the input to the algorithm is a *dissimilarity matrix* $\widehat{D}$ over S ($\widehat{D} = \{\hat{d}(i,j)\}_{i,j \in S}$), which is a noisy version of the metric D_T induced by the phylogenetic tree T. Under commonly studied models of evolution (as demonstrated in Section 8), this noise may be characterized by an increasing function of the pairwise distances. This is captured by the following definition:

DEFINITION 6.1. *Two dissimilarity matrices D_1, D_2 over S are said to be α-close, for some non-decreasing function $\alpha : \mathcal{R}^+ \to \mathcal{R}^+$, if for every taxon-pair $\{i,j\} \subseteq S$, we have*

$$|d_1(i,j) - d_2(i,j)| \leq \alpha(\min\{d_1(i,j), d_2(i,j)\}) .$$

For our analysis we assume that the input matrix $\widehat{D}$ and the tree-metric D_T are α-close for some efficiently computable non-decreasing function α. It is convenient to think of α as a sub-linear function (i.e. $\alpha(d) = o(d)$), though this requirement is not formally needed for most of our results and discussions. Our partial quartet oracle $FPM_{\widehat{D},\alpha}(q)$ is the following modified version of the well-known four-point method (FPM) [12].

The Partial Quartet Oracle – $FPM_{\widehat{D},\alpha}(q)$:
Let $q = \{a,b,c,d\}$ and assume w.l.o.g. that
$\hat{d}(a,b) + \hat{d}(c,d) \leq \hat{d}(a,c) + \hat{d}(b,d) \leq \hat{d}(a,d) + \hat{d}(b,c)$.
– The quartet oracle returns $(a,b;c,d)$, if:

$$(6.1) \quad \alpha(diam_{\widehat{D}}(q)) < \frac{\hat{d}(a,c) + \hat{d}(b,d) - (\hat{d}(a,b) + \hat{d}(c,d))}{4}$$

where $diam_{\widehat{D}}(q) \triangleq \max\{\hat{d}(i,j) : i,j \in q\}$.

– Otherwise, it returns *'null'*.

LEMMA 6.1. *Assume that $\widehat{D}$ is α-close to D_T. Then for every $z \in \mathcal{R}^+$, $FPM_{\widehat{D},\alpha}$ is an (r_z, ε_z)-reliable partial quartet oracle for T, where $r_z = z - \alpha(z)$ and $\varepsilon_z = 4\alpha(z)$.*

The (r, ε)-reliability of $FPM_{\widehat{D},\alpha}$ is required for establishing the ε-reliability of PDO in Corollary 5.1. Lemma 6.1 (together with the monotonicity of α) implies that in order to establish (r, ε)-reliability for small values of ε, we will have to make sure that r is sufficiently small. This is done by establishing tight upper bounds on r_1, r_2, r_3 mentioned in Corollary 5.1. These bounds depend on the *depth* and ε-*diameter* of T (defined below), and are strongly influenced by the taxon-insertion order and by the way directional representatives are updated.

Order of Insertion: The algorithm starts with the two closest taxa x_0, x_1 (under $\widehat{D}$). In each consequent iteration it selects a taxon $x \in S \setminus S'$ closest to S' (where $\hat{d}(S', x) = \min_{y \in S}\{\hat{d}(x,y)\}$).

Updating Directional Representatives: The algorithm holds two representatives $s_{\hat{v}}(\hat{u}), s_{\hat{u}}(\hat{v})$ for each edge $(\hat{u}, \hat{v})$ in $\widehat{T}$. Consider the updates required after inserting taxon x into $\widehat{T}$. Let y denote the taxon closest to x in $\widehat{T}$ (under $\widehat{D}$), and let $\hat{p}_x$ denote the parent of x (after its insertion). The following updates take place for the new external edge $(x, \hat{p}_x)$: $s_{\hat{p}_x}(x) \leftarrow x$ and $s_x(\hat{p}_x) \leftarrow y$. If an edge $(\hat{u}, \hat{v})$ is split to $(\hat{u}, \hat{p}_x), (\hat{p}_x, \hat{v})$, the following updates take place: $s_{\hat{v}}(\hat{p}_x), s_{\hat{p}_x}(\hat{u}) \leftarrow s_{\hat{v}}(\hat{u})$, and $s_{\hat{u}}(\hat{p}_x), s_{\hat{p}_x}(\hat{v}) \leftarrow s_{\hat{u}}(\hat{v})$. Finally, if contractions (of the internal edges of $\hat{t}_{inz}$) take place, then edges touching the new vertex (resulting from contraction) inherit the directional representatives of the corresponding external edges of $\hat{t}_{inz}$.

DEFINITION 6.2. *For $\varepsilon \geq 0$, the ε-diameter of T (denoted by $diam(T, \varepsilon)$) is the maximum weight of a simple path in T consisting only of edges of weight at most ε.*

DEFINITION 6.3. *The* depth *of a tree T (denoted by $depth(T)$) is given by:*

$$\max_{v \in V(T), u \in N_T(v)} \left\{ \min \left\{ d_T(v,s) : \begin{array}{l} s \in leaves(T), \\ u \in path_T(v,s) \end{array} \right\} \right\}.$$

The bounds on r_1, r_3 rely on the following result:

LEMMA 6.2. *Assume that $\widehat{D}$ is a dissimilarity matrix over S which is α-close to the tree-metric D_T. Let $S' \subset S$ be a subset of the taxa, and let $x \in S \setminus S'$ be a taxon closest to S' under $\widehat{D}$. Further let $y \in S'$ be the taxon closest to x in S', and let p_x denote the parent of x in $T(S' \cup \{x\})$. Then,*

1. $d_T(x,y) \leq 2depth(T) + \rho$.

2. $d_T(x,p_x) \leq depth(T) + \rho$.

where $\rho = \alpha(2depth(T)) + \alpha(2depth(T) + \alpha(2depth(T)))$

Lemma 6.2 is proven by first showing that there exist $x' \in S \setminus S'$ and $y' \in S'$ s.t. $d_T(x', y') \leq 2depth(T)$, and then applying the α-closeness of $\widehat{D}$ and D_T. We are now ready to present the main result of this section:

THEOREM 6.1. *Consider a phylogenetic tree T over a taxon-set S. Let $\widehat{D}$ be a dissimilarity matrix which is α-close to the tree-induced metric D_T, for some non-decreasing function α, and let $\rho = \alpha(2depth(T)) + \alpha(2depth(T) + \alpha(2depth(T)))$. Assume that the following properties hold for some $\varepsilon, z \in \mathcal{R}^+$:*

1. The ε-diameter of T is at most H.

2. $\alpha(z) \leq \frac{\varepsilon}{4}$ and $z - \alpha(z) \geq 4depth(T) + 3\rho + 2H$.

Then, algorithm Incremental_Reconstruct (Section 3) which uses the quartet oracle $FPM_{\widehat{D},\alpha}$ returns a topology which is an ε-contraction of T.

Sketch of proof. By Lemma 6.1, the second condition of the theorem implies that $FPM_{\widehat{D},\alpha}$ is an (r, ε)-reliable quartet oracle for $r = 4depth(T) + 3\rho + 2H$. The proof is completed by proving (inductively) that the topology $\widehat{T}$ our algorithm holds throughout its execution satisfies the following properties:

1. $\widehat{T}$ is a ε-contraction of $T(S')$.

2. Every edge in $\widehat{T}$ has weight at most $depth(T) + \rho$.

3. For every directional representative $s_{\hat{v}}(\hat{u})$ in $\widehat{T}$, we have $d(t_{\hat{v}}, s_{\hat{v}}(\hat{u})) \leq 2depth(T) + \rho$.

The induction is pretty straightforward. Lemma 6.2 is used to prove conditions 2 and 3. Condition 1 is proved with Lemma 4.3 and Corollary 5.1, using the induction hypothesis to obtain the following bounds on r_1, r_2, r_3:

$$r_1 = 2depth(T) + 2\rho + H \ / \ r_2 = H \ / \ r_3 = 2depth(T) + \rho.$$

The bounds on r_2, r_3 follow immediately from conditions 1 and 3. The bound on r_1 is obtained by observing the path in $T(S' \cup \{x\})$ connecting x and $t_{\hat{v}}$, where $\hat{v}$ is an arbitrary leaf of $\hat{t}_{env}$: condition 2 implies a bound of $depth(T) + \rho$ on the weight of the first and last edges, and the weight of the rest of the path is bounded by H since it consists of edges whose weight is at most ε. ∎

Note: Assuming that α is sub-linear, the value of z required for the reconstruction guarantee of Theorem 6.1 is linear in the depth and ε-diameter of T. Linearity in the depth is in a sense unavoidable, as demonstrated in [18]. Dependence on the ε-diameter is avoidable by modifying the directional oracle. As mentioned in the concluding discussion of Section 5, in the full paper we present an (efficient) variant of our directional oracle which enables a variant of Theorem 6.1 in which z is linearly bounded solely in the depth of T, and independent on its ε-diameter.

7 Complexity Analysis.

The space complexity of the algorithm (disregarding the space needed for storing the input dissimilarity matrix $\widehat{D}$) is obviously linear in n (the number of taxa), since the current topology $\widehat{T}$ and the directional representatives can easily be maintained in linear space

throughout the algorithm. The time complexity of the algorithm is quadratic, which is asymptotically optimal for algorithms reconstructing a phylogenetic tree with unbounded vertex-degrees, even from the exact tree-induced metric (see [9]). Each iteration involves selecting a taxon for insertion and applying *Attach_Taxon*. Note that computing the next taxon to be inserted (the one closest to S') can be done in linear time as done in Dijkstra-style algorithms [5]. The most time consuming task in *Attach_Taxon* is computing the insertion zone. This can be done by querying the directional oracle on all vertices of $\widehat{T}$, and then pruning $\widehat{T}$ in a DFS-traversal according to the queries' results. The DFS-traversal and pruning is clearly linear in n. All we have to show is that the total time complexity of all queries to the directional oracle PDO is linear in n. Consider a query corresponding to vertex $\hat{v}$ in $\widehat{T}$. After each iteration of the triplets tournament phase, at least two neighbors of $\hat{v}$ are eliminated from the candidate set, so this phase ends after at most $\frac{1}{2}deg_{\widehat{T}}(\hat{v})$ iterations. The validation phase simply scans the neighborhood of $\hat{v}$, and so it ends after at most $deg_{\widehat{T}}(\hat{v})$ iterations. Thus the total time complexity of directional queries is linear in the sum of vertex-degrees, which is linear in n.

8 Reliable Reconstruction from Biological Sequences.

In Section 6 (Theorem 6.1) we established the relation between the weight of contracted edges and the amount of noise separating the input dissimilarities from the tree-induced distances. The noise in the input was assumed to be bounded by a non-decreasing function α (see Definition 6.1). In this section we discuss the noise induced by a stochastic process of sequence evolution. Such a stochastic process induces a probability distribution over all possible inputs to the algorithm. In such a case, the noise function α is 'probabilistic' in the sense that it bounds the noise only *with sufficiently high probability*.

The results presented in this section assume the Cavender-Farris-Neyman [4, 14, 20] (CFN) model of *binary* sequence evolution, but they may be generalized to more complex models as well. This model assumes a rooted tree T, whose edges are associated with *changing probabilities* $\{p_e\}_{e \in E(T)}$. The process of evolution is modelled by uniformly randomizing a binary state (0 or 1) at the root and propagating mutations along the tree edges according to their changing probabilities. A *site* is defined by the n random states generated by the above process at the leaves of the tree. Note that under a given model-tree, the probability distribution of a specific site is well defined. Repeating this process k times, yields n binary sequences of length k, (corresponding

to k sites), which may serve as input to a *phylogenetic reconstruction method.*

The (additive) metric D_T induced by the model-tree T is defined by assigning a *weight* to each edge e in T: $w_e = -\frac{1}{2}\ln(1-2p_e)$. For $u, v \in V(T)$, denote by p_{uv} the *compound changing probability* between u and v, which is the probability of observing different states in u and v. It is well known (see e.g. [12]) that:

$$d_T(u,v) = \sum_{e \in path(u,v)} w_e = -\frac{1}{2}\ln(1-2p_{uv}).$$

Given a pair of sequences (of length k) corresponding to taxa i, j, the *observed* compound changing probability $\hat{p}_{ij}$ is estimated by the normalized hamming distance (i.e. the number of sites in the two sequences with different states divided by k). The observed pairwise dissimilarity is defined accordingly – $\hat{d}(i,j) = -\frac{1}{2}\ln(1-2\hat{p}_{ij})$. Note that $\hat{p}_{ij}$, and hence also $\hat{d}(i,j)$, are random variables defined by p_{ij} and k. The following claim provides the main result bounding the deviation of observed dissimilarities from the tree-induced distances:

CLAIM 8.1. *Let d be the tree-induced distance between two taxa, and let $\hat{d}$ be the observed dissimilarity between these two taxa. Then for any $\beta > 0$ we have:*

$$(8.2) \quad \Pr\left(|d - \hat{d}| > \beta\right) \le 2\exp\left(-\frac{k}{2}\frac{(1-e^{-2\beta})^2}{e^{4\min\{d,\hat{d}\}}}\right).$$

Sketch of proof. Let $p, \hat{p}$ be s.t. $d = \frac{1}{2}\ln(1-2p)$ and $\hat{d} = \frac{1}{2}\ln(1-2\hat{p})$. Note that $\hat{p}$ is the average of k i.i.d. *Bernoulli*(p) random variables. So Hoeffding's inequality [21] can be used to bound the deviation of $\hat{p}$ from its expectation p. The deviation of $\hat{d}$ from d is then obtained by expressing $\hat{d} - d$ as a function of $\hat{p} - p$.

LEMMA 8.1. *The tree-induced metric D_T and observed dissimilarity matrix $\widehat{D}$ are α_k-close with probability $1 - o(1)$ for α_k defined as follows:*

$$(8.3) \qquad \alpha_k(d) = -\frac{1}{2}\ln\left[1 - e^{2d}\sqrt{\frac{6\ln(n)}{k}}\right],$$

and when the expression inside the brackets in §8.3 is not positive, $\alpha_k(d) \overset{\triangle}{=} \infty$.

Proof. The noise function α_k is obtained by extracting the value of β for which the bound on the probability on the RHS of §8.2 equals $\frac{2}{n^3}$. Hence we get the following inequality for every taxon pair $i, j \in S$:

$$\Pr\left(|d(i,j) - \hat{d}(i,j)| > \alpha_k(\min\{d(i,j), \hat{d}(i,j)\})\right) \le \frac{2}{n^3}.$$

Now, by applying a simple union bound, we get that D_T and $\widehat{D}$ are α_k-close with probability at least $1 - \left[\binom{n}{2}\frac{2}{n^3}\right] = 1 - o(1)$. ∎

Note that, for given n and k, α_k (defined in §8.3) is indeed an increasing function of d. The noise, as described by α_k, depends also on the length of input sequences k: the longer the sequences, the smaller the noise. This give us (through Theorem 6.1) the relation between k and the weight of edges our algorithm contracts.

THEOREM 8.1. *Let $\widehat{D}$ be a dissimilarity matrix obtained from n binary taxon-sequences of length k which evolved according to the CFN model along a phylogenetic tree T. Let $\varepsilon > 0$ be s.t.*

$$(8.4) \quad k \ge 6\ln(n)\frac{\exp(7\varepsilon + 16depth(T) + 8diam(T,\varepsilon))}{\left(1 - e^{-\frac{\varepsilon}{2}}\right)^2}.$$

Then when executed on $\widehat{D}$, algorithm Incremental_Reconstruct (Section 3) returns an ε-contraction of T with probability $1 - o(1)$.

Sketch of proof. Assume D_T and $\widehat{D}$ are α_k-close, something which happens by Lemma 8.1 w.p. $1 - o(1)$. According to Theorem 6.1, all we need to show is that there exists some $z > 0$ which satisfies two conditions:

1. $\alpha_k(z) \le \frac{\varepsilon}{4}$.

2. $z - \alpha_k(z) \ge 4depth(T) + 3\rho + 2diam(T,\varepsilon)$.

Let $z = \alpha_k^{-1}\left(\frac{\varepsilon}{4}\right)$ (enforcing the first condition). By inverting α_k and assuming k which satisfies §8.4, we get that $z \ge 4depth(T) + 2diam(T,\varepsilon) + \frac{7}{4}\varepsilon$. The second condition is, therefore, proven by showing that $\rho \le \frac{\varepsilon}{2}$ (where ρ is as defined in Theorem 6.1). ∎

This result straightforwardly implies the fast convergence of our algorithm. In order to establish this, we need to prove that our algorithm reconstructs the correct topology of T with high probability from sequences of $poly(n)$ length, when edge-weights of T are assumed to be within some interval $[f, g]$ (where f, g are positive constants independent of n). Assuming this gives us that $depth(T) \le g\log_2(n)$, so by setting $\varepsilon = f - \gamma$ (for some arbitrarily small γ), Theorem 8.1 implies that our algorithm returns the correct topology with high probability, whenever $k \ge 6\ln(n) \cdot \phi(f) \cdot e^{16g\log_2(n)} = c_1\log(n)n^{c_2}$, where c_1, c_2 depend solely on f, g. The resulting function is clearly polynomially-bounded in n. Note that our algorithm is also *absolute* fast converging, since its input consists only of the dissimilarity matrix $\widehat{D}$, and the sequence-length k (from which the noise function α_k is calculated), and thus requires no knowledge of any of the parameters of the originating tree.

Fast convergence can be viewed as (asymptotically) obtaining the minimal sequence length required for

the correct reconstruction of a tree, as a function of its depth and minimal edge-weight. A natural generalization of this would be to obtain, for any 'user-defined' ε, the (asymptotically) minimal sequence length required for correct reconstruction of all edges of weight greater than ε, as a function of the tree depth. The current formulation of Theorem 8.1 does not realize this generalization since its result depends also on the ε-diameter of T. However, as mentioned earlier, in the full version of this paper we present a variant of our directional oracle which eliminates the dependence on the ε-diameter of T and thus provides a variant of Theorem 8.1 which achieves this generalization. This result enables us to bound the relation between the length of input sequences and the weight of contracted edges in terms of the tree-depth (and n). An interesting research direction is to improve this relation and provide tighter bounds for it.

References

[1] V. Berry and D. Bryant. Faster reliable phylogenetic analysis. In *RECOMB '99: Proceedings of the third annual international conference on Computational molecular biology*, pages 59–68, 1999.

[2] W. Beyer, M. Singh, T. Smith, and M. Waterman. Additive evolutionary trees. *J Theor Biol*, 64(2):199–213, January 1977.

[3] P. Buneman. The recovery of trees from measures of dissimilarity. *Mathematics in the Archeological and Historical Sciences*, pages 387–395, 1971.

[4] J. Cavender. Taxonomy with confidence. *Math Biosci*, 40:271–280, 1978.

[5] T. Cormen, C. Leiserson, R. Rivest, and C. Stein. *Introduction to Algorithms*. MITP, 2001. 2nd edition.

[6] M. Cryan, L. Goldberg, and P. Goldberg. Evolutionary trees can be learned in polynomial time in the two-state general markov model. *SIAM Journal on Computing*, 31(2):375–397, 2001.

[7] M. Csűrös. Fast recovery of evolutionary trees with thousands of nodes. *Journal of Computational Biology*, 9(2):277–297, 2002.

[8] M. Csűrös and M. Kao. Recovering evolutionary trees through harmonic greedy triplets. In *SODA: ACM-SIAM Symposium on Discrete Algorithms*, pages 261–270, 1999.

[9] J. Culberson and P. Rudnicki. A fast algorithm for constructing trees from distance matrices. *Information Processing Letters*, 30(4):215–220, February 1989.

[10] C. Daskalakis, C. Hill, A. Jaffe, R. Mihaescu, E. Mossel, and S. Rao. Maximal accurate forests from distance matrices. In *RECOMB '06: Proceedings of the tenth annual international conference on Computational molecular biology*, pages 281–295, 2006.

[11] C. Daskalakis, E. Mossel, and S. Roch. Optimal phylogenetic reconstruction. In *STOC '06: Proceedings of the thirty-eighth annual ACM symposium on Theory of computing*, pages 159–168, 2006.

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

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

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

[15] D. Huson, S. Nettles, and T. Warnow. Disk-Covering, a fast-converging method for phylogenetic tree reconstruction. *J Comp Biol*, 6:369–386, 1999.

[16] V. King, L. Zhang, and Y. Zhou. On the complexity of distance-based evolutionary tree reconstruction. In *SODA: ACM-SIAM Symposium on Discrete Algorithms*, pages 444–453, 2003.

[17] B. Moret, K. St. John, and T. Warnow. Absolute convergence: true trees from short sequences. In *SODA: ACM-SIAM Symposium on Discrete Algorithms*, pages 186–195, 2001.

[18] E. Mossel. Phase transitions in phylogeny. *Trans Amer Math Soc*, 356:2379–2404, 2004.

[19] E. Mossel. distorted metrics on trees and phylogenetic forests. *ACM Transactions on computational biology and bioinformatics*, 4:108–116, 2007.

[20] J. Neymann. Molecular studies of evolution: A source of novel statistical problems. In S. Gupta and Y. Jackel, editors, *Statistical Decision Theory and Related Topics*, pages 1–27. Academic Press, New York, 1971.

[21] L. Wasserman. *All of Statistics*. Springer, New York, 2004.