A Linear Time Approximation Scheme for Maximum Quartet Consistency on Sparse Sampled Inputs

Sagi Snir * Raphael Yuster †

Abstract

Phylogenetic tree reconstruction is a fundamental biological problem. Quartet amalgamation - combining a set of trees over four taxa into a tree over the full set - stands at the heart of many phylogenetic reconstruction methods. This task has attracted many theoretical as well as practical works. However, even reconstruction from a *consistent* set of quartet trees, i.e. all quartets agree with some tree, is NP-hard, and the best approximation ratio known is 1/3. For a dense input of $\Theta(n^4)$ quartets that are not necessarily consistent, the problem has a polynomial time approximation scheme.

When the number of taxa grows, considering such dense inputs is impractical and some sampling approach is imperative. It is known that given a randomly sampled consistent set of quartets from an unknown phylogeny, one can find, in polynomial time and with high probability, a tree satisfying a 0.425 fraction of them, an improvement over the 1/3 ratio.

In this paper we further show that given a randomly sampled consistent set of quartets from an unknown phylogeny, where the size of the sample is at least $\Theta(n^2 \log n)$, there is a randomized approximation scheme that runs in linear time in the number of quartets. The previously known polynomial approximation scheme for that problem required a very dense sample of size $\Theta(n^4)$. We note that samples of size $\Theta(n^2 \log n)$ are sparse in the full quartet set. The result is obtained by a combinatorial technique that may be of independent interest.

keywords: phylogenetic reconstruction, quartet amalgamation, approximation scheme.

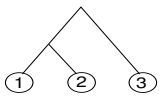
1 Introduction

The study of evolution and the construction of phylogenetic (evolutionary) trees are classical subjects in biology. Existing accurate phylogenetic techniques are capable of coping with a relatively small amount of data. DNA sequences from a variety of organisms are rapidly accumulating, challenging current approaches of phylogenetics. The *supertree* approach works by constructing small trees over overlapping sets of taxa, and subsequently, amalgamating these trees into a big tree over the full set.

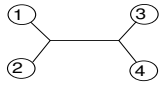
We distinguish between *rooted* and *unrooted* phylogenetic trees. In the rooted setting a rooted triplet tree (Figure 1:a) is the basic unit of information. We denote a triplet over the taxa a, b, c

*Department of Evolutionary Biology, University of Haifa, Haifa 31905, Israel.
E–mail: ssagi@math.haifa.ac.il
†Department of Mathematics, University of Haifa, Haifa 31905, Israel.
E–mail: raphy@math.haifa.ac.il

by $ab|c$ meaning that, in the underlying tree, the lowest common ancestor of a and b ($lca(a,b)$) is a descendant of $lca(a,c) = lca(b,c)$. Given a set of rooted triplets, there exists a polynomial time algorithm that constructs a tree consistent with the given set, or reports that no such tree exists [1, 8]. In the unrooted setting, the notion of lca is meaningless and therefore the basic unit of information is a quartet tree (Figure 1:b) - $ab|cd$ - meaning that there is a path in the underlying tree separating a and b from c and d. Quartet-based reconstruction methods have been extensively studied. For example, [15] describe the most used algorithm for inference from quartets, and another important heuristic is the quartet cleaning technique for correcting quartet errors [3, 4, 6, 9] The decision problem of whether there exists a tree satisfying all the quartets in an arbitrary given set is NP-complete [14]. This raises the problem of finding a tree maximizing the number of consistent quartets - *maximum quartet consistency* (MQC) [14].



(a)



(b)

Figure 1: In a *rooted* setting, a rooted triplet is the smallest informative tree versus an unrooted quartet in the *unrooted* setting. (a) - a rooted triplet tree 12|3. (b) - an unrooted quartet tree 12|34.

The MQC problem is central in many phylogenetic problems that reduce to solving MQC at some stage (see [11], Chapter 6, for an introduction). The complexity of approximating MQC is a longstanding open problem. At present, the best known polynomial time approximation algorithm has an approximation ratio of 1/3 (see Section 1.1). There are also a few results that assume some constraint either on the correctness or the density of the input. Most notably, Jiang et al. [9] designed a polynomial time approximation scheme (PTAS) for MQC when the input consists of all $\binom{n}{4}$ possible quartets. This was later generalized by the authors in [13] for inputs of size $\Theta(n^4)$. We mention here that the dual problem of Min Quartet Inconsistency (MQI), where one wants to find a phylogeny minimizing the number of input quartets that are inconsistent with such phylogeny has also been extensively studied [5, 18, 19].

The requirement that the input consists of $\Theta(n^4)$ quartets as in [9, 13] becomes prohibitive when the number of taxa grows even to moderate sizes. A faster approach is to sample a relatively small number of $m \ll \binom{n}{4}$ four-taxa sets, providing as input the corresponding m quartets they define, and try to solve MQC on this input. This version of the problem is *sampled-MQC*.

In a recent paper [13], the authors devised a new polynomial time approximation algorithm for sampled-MQC. Given a set of quartets sampled uniformly from the set of all $\binom{n}{4}$ quartets of an unknown phylogeny, the algorithm achieves an approximation ratio of roughly 0.425. Observe that since it is assumed that the quartets are sampled from some unknown phylogeny, the input

contains no errors. More generally, if the input contains at most an η fraction of errors, then the algorithm achieves an approximation ratio greater than $0.425 - 0.26\eta$. The result is obtained by constructing a *weighted quartet graph* and approximating a maximum weight cut in that graph.

The main result of this paper is that sampled-MQC from an unknown phylogeny admits a linear time randomized approximation scheme for sparse inputs. We prove that already for $m = \Theta(n^2 \log n)$, sampled-MQC from an unknown phylogeny admits an EPRAS (efficient polynomial time randomized approximation scheme [17]) that runs in $O(m)$ time. In other words, we compute, in $O(m)$ time, an n-taxa phylogenetic tree that satisfies, with high probability, at least $(1-\epsilon)m$ of the input quartets. This is an improvement over the input density of the other PTAS algorithms of [9, 13], but at the cost of assuming a uniform error-free (consistent) sampled input. It also improves significantly the previous 0.425 approximation, but at the cost of $\Omega(n^2 \log n)$ quartets. As we discuss in the final section, our algorithm also allows that a small fraction (the fraction depending on ϵ) of the supplied sampled quartets are erroneous.

1.1 Preliminaries

An (unrooted) phylogenetic tree is a tree whose internal vertices each have degree 3, and whose leaves are labeled by some taxa set (representing existing species). Throughout this paper all trees are phylogenetic trees, unless stated otherwise. For a tree T, we denote by $\mathcal{L}(T)$ the taxa set corresponding to the leaves of T.

Let T be a tree and $A \subseteq \mathcal{L}(T)$ a subset of the leaves of T. We denote by T_A, the subtree of T induced by A. Namely, T_A is the tree obtained from T by removing all leaves in $\mathcal{L}(T) \setminus A$ and paths leading exclusively to them, and subsequently internal vertices with degree two are contracted.

For two trees T and T', we say that T' is *satisfied* by T, if $\mathcal{L}(T') \subseteq \mathcal{L}(T)$ and $T_{\mathcal{L}(T')} = T'$. Otherwise, T' is *violated* by T. For a set of trees $\mathcal{T} = \{T_1, \ldots, T_k\}$ with possibly overlapping leaves, we denote by $\mathcal{T}_s(T)$ the set of trees in $\mathcal{T}$ that are satisfied by T. We say that $\mathcal{T}$ is *consistent* if there exists a tree T^* over the set of leaves $\cup_i \mathcal{L}(T_i)$ that satisfies every tree $T_i \in \mathcal{T}$ (see Figure 2). Otherwise, $\mathcal{T}$ is *inconsistent*. When $\mathcal{T}$ is inconsistent, it is desirable to find a tree T^* over $\cup_i \mathcal{L}(T_i)$ that maximizes some objective function. T^* is called a *supertree* and the problem of finding T^* is the *supertree problem*.

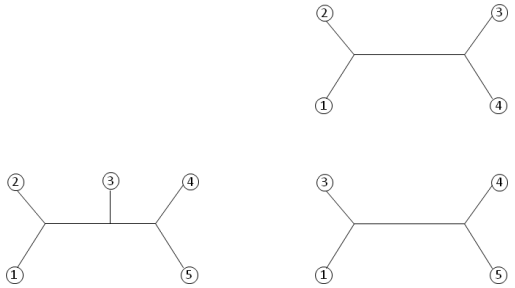


Figure 2: (a) A phylogenetic tree over five leaves. (b) Two quartets, 12|34 and 13|45 satisfied by the tree on the left.

A *quartet* tree (or simply *quartet*), is an undirected phylogenetic tree over four leaves $\{a, b, c, d\}$.

We write a quartet over $\{a, b, c, d\}$ as $ab|cd$ if the removal of the unique edge connecting the two internal vertices partitions the quartet into two components, one containing a, b and the other containing c, d, as in Figure 2. An important case of the supertree problem is when the set of input trees is a set of quartet trees $\mathcal{Q}$ and the task is to find a tree T that maximizes $|\mathcal{Q} \cap \mathcal{Q}(T)|$ where $\mathcal{Q}(T)$ is the set of all quartets induced by T (its full quartet set). The problem is denoted as *maximum quartet consistency* (MQC). We note that MQC is NP-hard already as a decision problem. Namely, it is NP-Complete to decide if a given set of quartets is consistent [14].

Notice that for every tree T with $|\mathcal{L}(T)| = n$, its full quartet set $\mathcal{Q}(T)$ consists of $\binom{n}{4}$ quartets, as each subset of four leaves defines a unique quartet satisfied by T. Consider the following trivial approximation algorithm for MQC. Take any tree T^* with n leaves, and randomly label them with the elements of $\mathcal{L}(T)$. As any four leaves a, b, c, d define one of three possible quartets (either $ab|cd$ or $ac|bd$ or $ad|bc$), only one of which is satisfied by T, we have that T^* satisfies an expected number of $1/3$ of the input quartet set. Surprisingly, no algorithm is known that improves upon the naive $1/3$ approximation, although the problem has been raised over two decades ago. In fact, even if we are guaranteed that the input $\mathcal{Q}$ satisfies $\mathcal{Q} \subset \mathcal{Q}(T)$ (namely, we are guaranteed that the optimal solution to MQC is $|\mathcal{Q}|$), no algorithm is known to achieve an outcome that is asymptotically better than $|\mathcal{Q}|/3$. As mentioned in the introduction, the MQC problem has a PTAS when $|\mathcal{Q}| = \Theta(n^4)$ [9, 13].

We now turn to sampled-MQC. As described in the introduction, we know the set of taxa $\mathcal{L}(T)$ of some unknown tree T, and given any four taxa we can (using biological information) infer the correct quartet. Clearly, if we have unlimited time and resources, we can generate all $\binom{n}{4}$ elements of $\mathcal{Q}(T)$ and solve the problem, as we have complete information, and the input is consistent. This, however, is unrealistic for very large n.

Motivated by this problem, sampled-MQC consists of an input $\mathcal{Q}$ of $m \ll \binom{n}{4}$ quartets sampled uniformly (say, with replacement), from $\mathcal{Q}(T)$. Recently, the authors [13] obtained an approximation algorithm for sampled-MQC that improves upon the naive $1/3$ approximation. They describe a randomized approximation algorithm that constructs a tree T^* satisfying an expected number of more than $0.425m$ elements of $\mathcal{Q}$, when the input is error-free. More generally, if the input contains at most an η fraction of errors, then the algorithm achieves an approximation ratio greater than $0.425 - 0.26\eta$.

The main result of this paper is a significant strengthening of the result of [13], for the case where the sample size is at least $\Theta(n^2 \log n)$. Notice that such a sample is very sparse in $\mathcal{Q}(T)$, as the size of the latter is $\binom{n}{4}$. We construct a linear time randomized approximation scheme for error-free sampled-MQC. The exact statement of our result follows.

Theorem 1.1. *Let $\epsilon > 0$ be fixed. Suppose that $\mathcal{Q}$ is a set of $m \geq \Theta(n^2 \log n)$ quartets sampled uniformly from the full quartet set $\mathcal{Q}(T)$ of some unknown tree T. Then there is an $O(m)$ time randomized algorithm that constructs a tree T^* that satisfies an expected number of $(1-\epsilon)m$ elements of $\mathcal{Q}$.*

It should be noted that the $O(m)$ term hides a large constant that depends on ϵ. The constant is of the form $(1/\epsilon)^{O(1/\epsilon)}$. As mentioned earlier, the proof of Theorem 1.1 can be made slightly more general. One does not need to assume that all m sampled quartets are error-free. As shown in the final section, our proof stays intact as long as the number of errors is a small fraction that depends quadratically on ϵ. The general (extremely high level) idea of the algorithm is to consider

relatively small *models* of phylogenetic trees. By scanning constantly many such models, we prove that one of them is guaranteed to be a model M of our unknown tree T. Given that model M, we prove how to expand it to a tree T^* which is also modeled by M. We prove that with small *constant* probability, T^* satisfies many quartets of $\mathcal{Q}$. By repeating this process a constant number of times, we eventually obtain, with high probability, a tree T^* which indeed satisfies many quartets of $\mathcal{Q}$.

The rest of this paper is organized as follows. In the next section we state and prove several notions and lemmas that are useful for the description of the algorithm. The algorithm and its proof of correctness are given in Section 3. The final section contains some concluding remarks.

2 Tree models

For the remainder of this paper, we assume that T is some (unknown) phylogenetic tree with n leaves. The leaves are labeled by a set $\mathcal{L}(T)$ of known labels, and $\mathcal{Q} \subset \mathcal{Q}(T)$ is a set of quartets obtained by sampling uniformly (with replacement) m elements of $\mathcal{Q}(T)$. The error parameter $\epsilon > 0$ is fixed, and we assume that $m \geq Cn^2 \log n$ where C is some constant depending only on ϵ, whose value is set in the proof as a result of optimization. Our goal is to construct a tree T^* that satisfies, with high probability, at least $(1-\epsilon)m$ quartets of $\mathcal{Q}$.

2.1 Tree models of constant size

Since all internal vertices of T have degree 3, it is a well-known fact that there is always an edge of T, whose removal partitions T into two components, each having at least $n/3$ of the leaves. Indeed, take an edge (x, y) whose removal partitions T to two components C_x and C_y, such that $||\mathcal{L}(C_x)| - |\mathcal{L}(C_y)||$ is minimized. If one of the components, say C_x, has more $2n/3$ leaves, then one of the edges incident with x in C_x, say an edge (x, z), has the property that its removal creates a component C_z with more than $n/3$ leaves, and hence choosing (x, z) instead of (x, y) causes a smaller imbalance, contradicting the choice of (x, y). So, we may now assume that $e = (x, y)$ has the stated property. It will be convenient to view T as a *rooted* tree. Subdivide e by introducing a new vertex r in its middle and make r the root. Hence, now T is a full binary tree, the children of r are x and y, and each of them is an ancestor of at least $n/3$ leaves (see Figure 3). Unless otherwise specified, we refer to this rooted version of T.

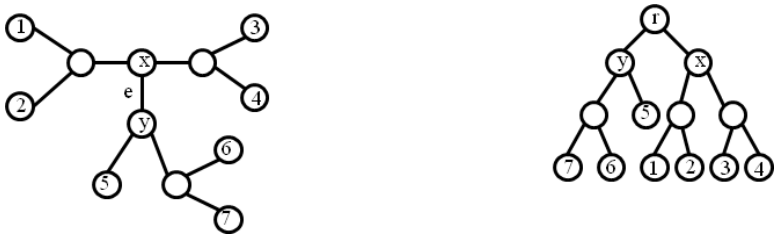


Figure 3: An unrooted phylogenetic tree on the left, and its rooted version on the right.

Let $\mathcal{I}(T)$ denote the set of internal vertices of T (including the root r), and notice that since $|\mathcal{L}(T)| = n$ and T is a full binary tree, then $|\mathcal{I}(T)| = n - 1$.

Recall that a *lowest common ancestor* of two vertices in a rooted tree is a vertex z that is a common ancestor of both x and y, and any other common ancestor of x and y is an ancestor of z. Denote by $lca(x, y)$ the lowest common ancestor of x and y. In particular, if x is an ancestor of y, then $lca(x, y) = x$. We say that a subset $K \subset \mathcal{I}(T)$ is *LCA-closed* if whenever $x, y \in K$ then also $lca(x, y) \in K$. We further demand that $r \in K$. Observe that every set of k vertices of a rooted tree can be made LCA-closed by adding to it at most k additional vertices.

There is a natural correspondence between an LCA-closed subset K, a topological minor of T it defines, and a partition it defines on $\mathcal{L}(T)$. We now state this correspondence formally.

Definition 1. *Let $K \subset \mathcal{I}(T)$ be LCA-closed. Let $M_T(K)$ be the rooted binary tree whose vertex set is K, and u is the parent of v in $M_T(K)$ if and only if u is the lowest among all ancestors of v in K. We call $M_T(K)$ a* tree model *of T.*

Notice that r is the root of $M_T(K)$, since it is the only vertex of K with no ancestor in K. Observe also that $M_T(K)$ is a contraction (in fact, a topological minor) of T.

For $v \in K$, let

$$A_v = \{x \in \mathcal{L}(T) \mid v \text{ is the lowest ancestor of } x \text{ in } K\} .$$

Let v_0 and v_1 be the two children of v in T (and notice that v_0 and v_1 are not necessarily in K and may or may not be in $\mathcal{L}(T)$). Then A_v is further divided into two parts, $A_{v,0}$ are those leaves that have v_0 as their ancestor while $A_{v,1}$ have v_1 as their ancestor.

Definition 2. *The set $\mathcal{P}_T(K) = \{A_{v,0} \mid v \in K\} \cup \{A_{v,1} \mid v \in K\}$ is the* leaf partition *of the model.*

Notice that $\mathcal{P}_T(K)$ is a partition of $\mathcal{L}(T)$ into $2|K|$ parts. It may be the case that some parts are empty; for example, if $v \in K$ and its child $v_0 \in K$ then $A_{v,0} = \emptyset$. See Figure 4 for an example of a model and its corresponding leaf partition.

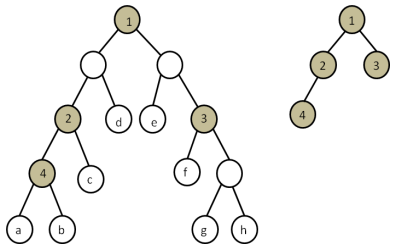


Figure 4: A tree with $\mathcal{L}(T) = \{a, b, c, d, e, f, g, h\}$ and its model formed by the LCA-closed set $\{1, 2, 3, 4\}$. The corresponding leaf partition is: $A_{1,0} = \{d\}$, $A_{1,1} = \{e\}$, $A_{2,0} = \emptyset$, $A_{2,1} = \{c\}$, $A_{3,0} = \{f\}$, $A_{3,1} = \{g, h\}$, $A_{4,0} = \{a\}$, $A_{4,1} = \{b\}$.

Definition 3. *A tree model $M_T(K)$ of a tree T is a* δ-model *if every element of $\mathcal{P}_T(K)$ has size at most δn.*

The next lemma proves that there are δ-models with $O(1/\delta)$ vertices.

Lemma 2.1. *There is a δ-model of T with at most $4/\delta$ vertices.*

Proof. We prove that there exists a (not necessarily LCA-closed) subset $K' \subset \mathcal{I}(T)$ with $r \in K'$, so that for each $v \in K'$, the set of leaves A_v has $\delta n/2 \leq |A_v| \leq \delta n$. (Notice that since $r \in K'$ then A_v is well-defined even if K' is not LCA-closed.) Since $\mathcal{P}' = \mathcal{P}_T(K')$ is a partition of $\mathcal{L}(T)$, this implies that $|K'| \leq n/(\delta n/2) = 2/\delta$. Now, since K' can be made into an LCA-closed set K by adding to K' at most $|K'|$ additional vertices, we have that $|K| \leq 4/\delta$. Since $\mathcal{P} = \mathcal{P}_T(K)$ is a refinement of $\mathcal{P}'$, then every element of $\mathcal{P}$ also has size at most δn, and the lemma follows.

We construct K' as follows. We initially set all the leaves in $\mathcal{L}(T)$ as *unmarked*. Next, we perform a postorder traversal of T. Whenever we reach a vertex $v \in \mathcal{I}(T)$, let U_v denote the set of yet unmarked leaves in the subtree rooted at v. If $|U_v| \geq \delta n/2$ then we add v to K', mark all elements of U_v, and notice that $A_v = U_v$. Observe that we must have $|U_v| \leq \delta n$. Otherwise, one of the two children of v, say w, would have had at least $\delta n/2$ unmarked leaves in U_w. But since w has already been traversed, we should have already added w to K' and marked all elements of U_w. □

2.2 Constant size nets for constant size models

Let $M_T(K)$ be a δ-model, and let $\mathcal{P}_T(K)$ be its leaf partition, consisting of subsets $A_{v,j}$ for $v \in K$ and $j = 0, 1$.

Definition 4. *The function* $f_T(K) : K \times \{0,1\} \to [0, \delta]$ *where* $f_T(K)(v, j) = |A_{v,j}|/n$ *is called the* size vector *of the model.*

We say that a function $f' : K \times \{0,1\} \to [0, \delta]$ is an α-*approximation* of $f_T(K)$ if $f'(v, j) \leq f_T(K)(v, j) \leq f'(v, j) + \alpha$ for all $v \in K$ and $j = 0, 1$.

Given $|K|$ and δ, a family of functions $\mathcal{F}$ is called a $(|K|, \delta, \alpha)$-*net* if for every possible function $f : K \times \{0,1\} \to [0, \delta]$, there exists an α-approximation of f in $\mathcal{F}$.

For constants $|K|$ and δ, it is not difficult to construct a $(|K|, \delta, \delta^4)$-net of constant size, and in constant time. This is analogous to constructing the set of all vectors of length $2|K|$ whose coordinates are of the form $i\delta^4$ for $i = 0, \ldots, \lfloor \delta^{-3} \rfloor$. As there are at most $(1 + \delta^{-3})^{2|K|}$ such vectors, the claimed construction follows.

3 Proof of the main result

In our proof we will use

$$\delta = \frac{\epsilon}{5000} \ . \tag{1}$$

For the *proof* of our algorithm we need to fix and reference the following objects.

1. A rooting of T from some vertex r as described in Section 2.1. Recall that this makes T into a full binary tree, and each child of r is an ancestor of at least $n/3$ leaves.

2. A δ-model $M_T(K)$ of T with at most $4/\delta$ vertices, guaranteed to exist by Lemma 2.1. Label the vertices of $M_T(K)$ with $\{1, \ldots, |K|\}$.

3. The leaf partition $\mathcal{P}_T(K)$ of the model $M_T(K)$. Recall that $\mathcal{P}_T(K)$ is a partition of $\mathcal{L}(T)$ into $2|K|$ parts, denoted by $A_{v,j}$ for $v \in K$ and $j = 0, 1$.

4. The size vector $f_T(K)$ of the model. Recall that $f_T(K)(v, j) = |A_{v,j}|/n$.

Generally speaking, our algorithm will enumerate all possible δ-models, it will "guess" a close approximation of $f_T(K)$, and it will try to generate a partition of $\mathcal{L}(T)$ defined by a pair of a model and an approximation of $f_T(K)$. We will show that after guessing a constant number of times (a constant depending on δ and hence on ϵ) we are guaranteed that one of our trials forms a "correct" partition of $\mathcal{L}(T)$. We will then show that, under the assumption of a correct partition, the generated partition is *close* to the actual partition $\mathcal{P}_T(K)$. We will then show how to construct a phylogenetic tree T^* using the generated partition, which satisfies many quartets of $\mathcal{Q}$. We mention here an algorithm of Giotis and Guruswami [7] which gives a randomized PTAS for a different problem where the goal is also to compute some partition, and the main idea there is to guess a "sketch" of the solution and then to extend such sketch. The details of their algorithm, however, are completely different from ours.

We describe a general procedure PARTITION(M, f). The first parameter is a labeled binary tree M with k vertices, and the second parameter is a function $f : \{1, \ldots, k\} \times \{0, 1\} \to [0, \delta]$. We call PARTITION a constant number of times. For all integers k satisfying $1/\delta \leq k \leq 4/\delta$ we construct a (k, δ, δ^4)-net $\mathcal{F}_k$. This is done as explained in Section 2.2. As shown there, the number of elements in F_k is at most $(1/\delta)^{O(1/\delta)}$. We also construct the set $\mathcal{M}_k$ of all possible labeled binary trees with k vertices. The number of such trees is trivially less than $k^k = (1/\delta)^{O(1/\delta)}$ and there are several classical ways to generate them, perhaps the simplest is by using the classical bijection between labeled trees and Prüfer codes. For each k, for each $f \in \mathcal{F}_k$ and for each $M \in \mathcal{M}_k$ we call PARTITION(M, f). Hence, at some point we are *guaranteed* to call it with parameters (M_0, f_0) where M_0 is label-isomorphic to $M_T(K)$ and f_0 is a δ^4-approximation of $f_T(K)$.

What follows is a description of PARTITION *assuming* that the parameters are instantiated by (M_0, f_0). Its behavior in other calls is of no interest to us (it may return a partition that is not close to $\mathcal{P}_T(K)$).

3.1 PARTITION(M_0, f_0)

PARTITION tries, using f_0 and M_0, to construct a partition $\mathcal{P}^*$ of $\mathcal{L}(T)$ that is *close* (in a well defined sense) to $\mathcal{P}_T(K)$. We will show that with *constant positive probability*, it is guaranteed to succeed.

The main problem, of course, is that although we have M_0 and f_0 (and thus we assume from now on that we know $M_T(K)$ and have a good approximation of $f_T(K)$), we do *not know* the actual leaf partition $\mathcal{P}_T(K)$. However, we *do know* a close approximation of the *cardinality* of each element of $\mathcal{P}_T(K)$, since f_0 is close to $f_T(K)$.

We define:

1. $y_{v,j}$ to be the child of v in T that is the ancestor of all elements of $A_{v,j}$;
2. $S_{v,j} \subset \mathcal{L}(T)$ to be all the leaves that have $y_{v,j}$ as their ancestor. Notice that $A_{v,j} \subset S_{v,j}$.

For example, in Figure 4, we have that $y_{1,0}$ is the left child of the root 1, and $S_{1,0} = \{a, b, c, d\}$.

If v is an ancestor of u, we say that u is *below* v and use the notation $u < v$. The notation $u \leq v$ is used when we allow $u = v$.

Definition 5. *We say that a partition $\mathcal{P}^*$ of $\mathcal{L}(T)$ is* close *to $\mathcal{P}_T(K)$ if the following conditions hold.*

1. $\mathcal{P}^* = \{B_{v,j} \mid v \in K \ , \ j = 0,1\} \cup \{B^*\}$. *We call* B^* *the* exceptional part. *Hence,* $B^* = \mathcal{L}(T) \setminus \bigcup_{(v,j)\in K\times\{0,1\}} B_{v,j}$.

2. *For all* $v \in K$ *and* $j = 0,1$ *we have* $|A_{v,j} \setminus B_{v,j}| \leq 50\delta^2 n$.

3. *For all* $v \in K$ *and* $j = 0,1$ *we have* $B_{v,j} \subset S_{v,j}$.

4. *For all* $v \in K$ *and* $j = 0,1$ *we have* $|S_{v,j} \setminus \bigcup_{u\leq v}(B_{u,0} \cup B_{u,1})| \leq 50\delta^2 n$.

In fact, notice that the second requirement (which is the one we are after) is actually a consequence of the third and fourth requirements. We will show how to construct, with constant positive probability, a partition $\mathcal{P}^*$ that is close to $\mathcal{P}_T(K)$.

By performing a postorder traversal of M_0, we may assume that whenever we reach v, we have already defined sets $B_{u,i}$ for all pairs (u,i) such that $u < v$, and, furthermore, the sets already defined satisfy the desired properties. We will show how to define $B_{v,j}$ so that with *constant* probability, it also satisfies the properties above. Since the number of possible pairs (v,j) is only $2|K|$, this yields that, with constant positive probability, the constructed $\mathcal{P}^*$ is close to $\mathcal{P}_T(K)$.

3.1.1 Constructing $B_{v,j}$

Assume that we have reached vertex v in our postorder traversal and wish to construct $B_{v,j}$. Consider the set of leaves $X = S_{v,j} \setminus \bigcup_{u<v}(B_{u,0} \cup B_{u,1})$. Namely, X consists of the elements of $A_{v,j}$ together with all other elements of $S_{v,j}$ that have not been assigned to sets $B_{u,i}$. Although the algorithm does not *know* the set X (since it does not know $S_{v,j}$), it does know a good approximation for its cardinality. Since f_0 is a δ^4-approximation, we know each $|A_{u,i}|$ up to $\delta^4 n$. Namely, $f_0(u,i) \leq |A_{u,i}|/n \leq f_0(u,i) + \delta^4$. As there are at most $2|K| \leq 8/\delta$ possible pairs (u,i), the overall error in estimating $|S_{v,j}|$ is at most $8\delta^3 n$, since $S_{v,j} = \bigcup_{u\leq v}(A_{u,0} \cup A_{u,1})$. Hence, our estimate for X, which is just our estimate for $|S_{v,j}|$ minus the already computed size $|\bigcup_{u<v}(B_{u,0} \cup B_{u,1})|$ is also correct up to an error of $8\delta^3 n$.

Consider first the case where our estimate for $|X|$ is less than $49\delta^2 n$. In particular, we are guaranteed that $|X| \leq 49\delta^2 n + 8\delta^3 n \leq 50\delta^2 n$. In this case, we simply define $B_{v,j} = \emptyset$. Notice that since X contains $A_{v,j}$, this still satisfies the conditions required of $B_{v,j}$ in Definition 5.

So, we may now assume that $|X| \geq 49\delta^2 n$. Now, consider the tree T_X whose root is $y_{v,j}$ and whose leaf set is X. Again, the algorithm does not know T_X, but it can guess, with constant positive probability, some important information regarding its structure.

Each vertex t of T_X, when removed from T_X, partitions $T_X - t$, and hence also partitions X, into three parts (some of which may be empty). One part is the component containing the parent of t (if $t = y_{v,j}$ then this part is empty). The two other parts contain each a child of t (if t does not have two children then these parts could possibly be empty). So, denote the corresponding partition of X by $X_0(t), X_1(t), X_2(t)$ where $X_0(t)$ are the leaves of the part of $T_X - t$ that contain $y_{v,j}$ (see Figure 5). In particular $X_0(y_{v,j}) = \emptyset$ and if $t \in X$ (namely t a leaf of T_X) then $X_1(t) = X_2(t) = \emptyset$ while $X_0(t) = X - t$.

Lemma 3.1. *There exists* $t \in T_X$ *for which* $|X_0(t)| \leq 16\delta^2 n$ *but* $|X_0(t) \cup X_1(t)| > 16\delta^2 n$ *and* $|X_0(t) \cup X_2(t)| > 16\delta^2 n$.

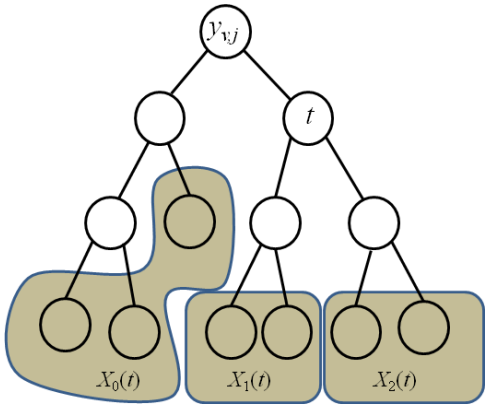


Figure 5: A vertex t of T_X and the corresponding $X_j(t)$ for $j = 0, 1, 2$.

Proof. Let t be a furthest vertex from $y_{v,j}$ in T_X for which $|X_0(t)| \leq 16\delta^2 n$. Clearly t is not a leaf of T_X since for leaves we have $|X_0(t)| = |X| - 1 \geq 49\delta^2 n - 1 > 48\delta^2 n$. Also, t must have two children in T_X since otherwise, if t_1 is its only child then $X_0(t_1) = X_0(t)$, and t_1 is further than t from $y_{v,j}$. So, let t_1 and t_2 be the two children of t, where t_j belongs to the subtree of $T_X - t$ whose leaves are $X_j(t)$ for $j = 1, 2$. If $|X_0(t) \cup X_1(t)| \leq 16\delta^2 n$ then observe that since $X_0(t_2) = X_0(t) \cup X_1(t)$, then t_2 would have been furthest. Therefore, $|X_0(t) \cup X_1(t)| > 16\delta^2 n$, and symmetrically, $|X_0(t) \cup X_2(t)| > 16\delta^2 n$. □

We call a vertex t satisfying Lemma 3.1 a *center* of T_X. So fix some center t, and consider the cardinalities, $|X_j(t)| = \alpha_j n$ for $j = 0, 1, 2$. Our algorithm guesses values α_j^* for $j = 0, 1, 2$ that approximate the α_j. We say that the guess is *valid* if $|\alpha_j^* - \alpha_j| \leq \delta^3$ for $j = 0, 1, 2$. As there are three values to guess in the range $[0, 1]$, the probability of a valid guess is $(\delta^3)^3 = \delta^9$, which is a constant positive probability depending on δ.

We may now assume that the α_j^* are a valid guess for $j = 0, 1, 2$. Now, if $\alpha_1^* \geq 16\delta^2$, we also guess a leaf $w_1 \in X_1(t)$. The probability that we have guessed correctly a leaf in $X_1(t)$ is therefore at least $16\delta^2 - \delta^3$, hence a constant positive probability. Similarly, if $\alpha_2^* \geq 16\delta^2$, we guess a leaf $w_2 \in X_2(t)$. So, assume that we have guessed correctly.

The construction of $B_{v,j}$ is done by considering three cases. The first case is when $\alpha_1^* < 16\delta^2$. The second case is when $\alpha_2^* < 16\delta^2$. The third case is when both are at least $16\delta^2$. Since the first and second case are symmetric, we consider, without loss of generality, only the first case and third case.

Before describing these cases, we fix some notation. Let $B = \bigcup_{u<v}(B_{u,0} \cup B_{u,1})$. Let $D = \mathcal{L}(T) - S_{v,j}$. Namely, D is the set of leaves that do not have $y_{v,j}$ as their ancestor. Since $y_{v,j}$ is not the root of T (it may be a child of the root in the case where $v = r$ is the root), then D contains all the leaves of some child of the root of T, so $|D| \geq n/3$. Notice that D, B, X are pairwise disjoint and

$$D \cup B \cup X = \mathcal{L}(T) \,.$$

Consider a sample of $Cn^2 \log n$ quartets where C is a sufficiently large constant. Eliminate from this sample all quartets that contain an element of B. As $|D| \geq n/3$, we still have a completely

random sample Q of $q \geq C' n_0^2 \log n_0$ quartets over $D \cup X$ where $n_0 = |D \cup X| > n/3$, and C' is a suitably large constant. Observe that, indeed, the probability that a sampled quartet has no taxa in B is at least $(1/3)^4$ so it suffices to pick $C \approx 81C'$. Notice that for two leaves $a, b \in D \cup X$, the probability that they appear in a specific element of Q is denoted by p and is *precisely*

$$p = \frac{12}{n_0(n_0-1)} .$$

For simplicity, denote $|D| = \eta n_0$, $|X_i(t)| = \alpha_i n = \beta_i n_0$. Observe that

$$\eta + \beta_0 + \beta_1 + \beta_2 = 1 . \tag{2}$$

Finally, let $\beta_i^* = \alpha_i^* n/n_0$, and observe that as $n/n_0 < 3$ we have $|\beta_i - \beta_i^*| \leq 3\delta^3$.

3.1.1.1 The case $\alpha_1^* < 16\delta^2$. Since $|X| \geq 49\delta^2 n$, we have that $\alpha_0 + \alpha_1 + \alpha_2 \geq 49\delta^2$. As t is a center we have $\alpha_0 \leq 16\delta^2$. Since $|\alpha_j^* - \alpha_j| \leq \delta^3$ we surely have $\alpha_2^* > 16\delta^2$ so we are in the case where we have guessed $w_2 \in X_2(t)$.

We construct $B_{v,j}$ as follows: Given $z \in \mathcal{L}(T) \setminus B = X \cup D$, we count the number of quartets of Q in which w_2 and z are in opposite sides. If this number is at most $qp(2\beta_2^* - (\beta_2^*)^2 + 6\delta^3 + \delta^2/2)$ then we place z in $B_{v,j}$. Otherwise, we don't.

The next three lemmas prove that, with high probability, the constructed set $B_{v,j}$ satisfies $X_2(t) \subset B_{v,j} \subset X$. These lemmas show that for a $z \in D \cup X$, we can, with high probability, *differentiate* between the case $z \in D$ and the case $z \in X_2(t)$. For those $z \in X_0(t) \cup X_1(t)$ we will not be able to differentiate. Using this differentiation, we can find a subset $B_{v,j}$ so that $X_2(t) \subset B_{v,j} \subset X$, as required.

Lemma 3.2. *With probability at least $8/9$, for all $z \in D$, the number of elements of Q in which w_2 and z are on opposite sides is at least*

$$qp(2\beta_2 - 2\beta_2^2 + 2\eta(\beta_0 + \beta_1) - \delta^2/2) . \tag{3}$$

Proof. Consider some $z \in D$. Given that w_2 and z are in the same quartet of Q, what is the probability that they are on opposite sides? Let a, b denote the other two elements of the quartet. Clearly, if $a \in X_2(t)$ and $b \notin X_2(t)$ then the quartet must be $aw_2|zb$. Similarly, if $b \in X_2(t)$ and $a \notin X_2(t)$ then the quartet must be $bw_2|za$. Also, if $a \in D$ and $b \in X_0(t) \cup X_1(t)$ we must have $bw_2|za$. Similarly, if $b \in D$ and $a \in X_0(t) \cup X_1(t)$ we must have $aw_2|zb$. It follows that, given that w_2 and z are in the same quartet of Q, they are on opposite sides with probability *at least*

$$2\beta_2\eta + 2\beta_2(\beta_0 + \beta_1) + 2\eta(\beta_0 + \beta_1) - o_n(1) = 2\beta_2 - 2\beta_2^2 + 2\eta(\beta_0 + \beta_1) - o_n(1)$$

where the r.h.s. uses (2) (the term $o_n(1)$ denotes a quantity that goes to zero with n, and is due to the fact that a and b are sampled *without* replacement, as they must be distinct). Hence, the expected number of elements of Q in which w_2 and z are on opposite sides is greater than

$$qp(2\beta_2 - 2\beta_2^2 + 2\eta(\beta_0 + \beta_1) - \delta^2/4) .$$

But $q \geq C' n_0^2 \log n_0$ and hence $qp > C' \log n$. Since each element of Q is sampled uniformly and independently, we have, using a standard large deviation inequality (see [2], Theorem A.1.13), that the probability of being below the expectation by more than $qp\delta^2/4$ is smaller than

$$\exp(-(qp\delta^2/4)^2/(2qp(2\beta_2 - 2\beta_2^2 + 2\eta(\beta_0 + \beta_1) - \delta^2/4))) .$$

As this expression is trivially less than $\exp(-qp\delta^4/128)$ and since $qp > C' \log n$ we get that by choosing $C' = 256/\delta^4$, the probability of being below the expectation by more than $qp\delta^2/4$ is less than $1/(9n)$. Hence, by the union bound, with probability at least $8/9$, for all $z \in D$ we have that the number of elements of Q in which w_2 and z are on opposite sides is at least $qp(2\beta_2 - 2\beta_2^2 + 2\eta(\beta_0 + \beta_1) - \delta^2/2)$, as claimed. □

Lemma 3.3. *With probability at least $8/9$, for all $z \in X_2(t)$, the number of elements of Q in which w_2 and z are on opposite sides is at most*

$$qp(2\beta_2 - \beta_2^2 + \delta^2/2) \ . \tag{4}$$

Proof. Consider some $z \in X_2(t)$. Given that w_2 and z are in the same quartet of Q, what is the probability that they are on the same side? Let a, b denote the other two elements of the quartet. Clearly, if both a and b are not in $X_2(t)$ then the quartet must be $zw_2|ab$. Hence, given that w_2 and z are in the same quartet of Q, they are on opposite sides with probability *at most*

$$1 - (1 - \beta_2)^2 + o_n(1) = 2\beta_2 - \beta_2^2 + o_n(1) \ .$$

Hence, the expected number of elements of Q in which w_2 and z are on opposite sides is less than

$$qp(2\beta_2 - \beta_2^2 + \delta^2/4) \ .$$

Again, the probability of being above the expectation by more than $qp\delta^2/4$ is smaller than $1/(9n)$ using $C' = 256/\delta^4$ (this time we use the large deviation inequality from [2], Theorem A.1.11, as we need to bound the expectation from above). Hence, with probability at least $8/9$, for all $z \in X_2(t)$ we have that the number of elements of Q in which w_2 and z are on opposite sides is at most $qp(2\beta_2 - \beta_2^2 + \delta^2/2)$, as claimed. □

Lemma 3.4. *Let $B_{v,j}$ consist of all $z \in X \cup D$ for which the number of quartets containing w_2 and z in opposite sides is at most $qp(2\beta_2^* - (\beta_2^*)^2 + 6\delta^3 + \delta^2/2)$. Then, with probability at least $7/9$ we have $X_2(t) \subset B_{v,j} \subset X$.*

Proof. Recall that $|\beta_2^* - \beta_2| \leq 3\delta^3$. Hence,

$$qp(2\beta_2 - \beta_2^2 + \delta^2/2) \leq qp(2\beta_2^* - (\beta_2^*)^2 + 6\delta^3 + \delta^2/2) \ .$$

Thus, by (4), with probability at least $8/9$, we have that $X_2(t) \subset B_{v,j}$.

It remains to prove that with probability at least $8/9$ we have that $B_{v,j} \subset X$, or, equivalently, that $B_{v,j} \cap D = \emptyset$. By (3) it suffices to prove that

$$2\beta_2 - 2\beta_2^2 + 2\eta(\beta_0 + \beta_1) - \delta^2/2 > 2\beta_2^* - (\beta_2^*)^2 + 6\delta^3 + \delta^2/2 \ .$$

As $|\beta_2^* - \beta_2| \leq 3\delta^3$ it suffices to prove that $2\beta_2 - 2\beta_2^2 + 2\eta(\beta_0 + \beta_1) - \delta^2/2 > 2\beta_2 - \beta_2^2 + 12\delta^3 + \delta^2/2$ which is equivalent to showing that

$$2\eta(\beta_0 + \beta_1) - \beta_2^2 > \delta^2 + 12\delta^3 \ . \tag{5}$$

Recall that $\eta = |D|/n_0$ and $|D| \geq n/3$ so $\eta \geq 1/3$. As t is a center we have, by Lemma 3.1, that $\alpha_0 + \alpha_1 \geq 16\delta^2$ so $\beta_0 + \beta_1 > 16\delta^2$. Since $|A_{v,j}| \leq \delta n$ and since X consists of $A_{v,j}$ and at most $50\delta^2 n$ additional vertices from sets corresponding to pairs (u, j) where u is below v we have, in particular, that $\alpha_2 \leq \delta + 50\delta^2$. Thus, $\beta_2 \leq 3\delta + 150\delta^2 < 3.1\delta$. Hence the left hand side of (5) is at least $\frac{2}{3} \cdot 16\delta^2 - 9.61\delta^2 > 1.056\delta^2$, proving (5). □

Notice that by Lemma 3.4, with high probability (at least 7/9), the constructed $B_{v,j}$ misses at most $|X_1(t) \cup X_0(t)| < 49\delta^2 n$ vertices of X, and in particular satisfies the requirements in the definition of a close partition.

3.1.1.2 The case $\alpha_1^* \geq 16\delta^2$ and $\alpha_2^* \geq 16\delta^2$. In this case we have selected $w_1 \in X_1(t)$ and $w_2 \in X_2(t)$.

We construct $B_{v,j}$ as follows. As in Section 3.1.1.1 we use the same rule stated there to distinguish between $z \in D$ and $z \in X_2(t)$. Namely, given $z \in X \cup D$, we count the number of quartets of Q in which w_2 and z are in opposite sides. If this number is at most $qp(2\beta_2^* - (\beta_2^*)^2 + 6\delta^3 + \delta^2/2)$ then we place z in a set U_2. Symmetrically, we count the number of quartets of Q in which w_1 and z are in opposite sides. If this number is at most $qp(2\beta_1^* - (\beta_1^*)^2 + 6\delta^3 + \delta^2/2)$ then we place z in a set U_1. Finally, we define $B_{v,j} = U_1 \cup U_2$.

Lemma 3.5. *Let U_2 consist of all $z \in X \cup D$ for which the number of quartets containing w_2 and z in opposite sides is at most $qp(2\beta_2^* - (\beta_2^*)^2 + 6\delta^3 + \delta^2/2)$, let Let U_1 consist of all $z \in X \cup D$ for which the number of quartets containing w_1 and z in opposite sides is at most $qp(2\beta_1^* - (\beta_1^*)^2 + 6\delta^3 + \delta^2/2)$, and let $B_{v,j} = U_1 \cup U_2$. Then, with probability at least 5/9 we have that $X_1(t) \cup X_2(t) \subset B_{v,j} \subset X$.*

Proof. By the proof of Lemma 3.4 we have that with probability at least 7/9, $X_2(t) \subset U_2 \subset X$. Symmetrically, with probability at least 7/9, $X_1(t) \subset U_1 \subset X$. Hence, $B_{v,j} = U_1 \cup U_2$ satisfies, with probability at least 5/9, that $X_1(t) \cup X_2(t) \subset B_{v,j} \subset X$, as claimed. □

Notice that by Lemma 3.5, with high probability (at least 5/9), the constructed $B_{v,j}$ misses at most $|X_0(t)| \leq 16\delta^2 n$ vertices of X, and in particular satisfies the requirements in the definition of a close partition.

3.1.2 Analysis

We have proved that PARTITION, when called with the arguments (M_0, f_0), returns a partition $\mathcal{P}^*$ that is close to $\mathcal{P}_T(K)$ with constant positive probability. We have shown in Section 3.1.1 that, the probability of constructing a particular $B_{v,j}$ correctly (by correctly we mean that it satisfies the requirements of a close partition) is composed of the following ingredients. First we need to correctly choose a valid guesses α_j^* for $j = 0, 1, 2$ that approximate the α_j. We have shown that this happens with probability at least δ^9. We then need to guess a leaf $w_1 \in X_1(t)$ or a leaf $w_2 \in X_2(t)$ (or both). We have shown that the probability that this occurs is at least $16\delta^2 - \delta^3$ for each. Once this is done, Lemma 3.4 and Lemma 3.5 show that $B_{v,j}$ is constructed correctly with probability at least 5/9. Hence, the probability to construct $B_{v,j}$ correctly is at least $\delta^9 \cdot (16\delta^2 - \delta^3)^2 \cdot (5/9) > \delta^{13}$.

In order for PARTITION(M_0, F_0) to return a partition $\mathcal{P}^*$ that is close to $\mathcal{P}_T(K)$, a correct $B_{v,j}$ must be generated for all $v \in K$ and for all $j = 0, 1$. Thus, the generated P^* is close to $\mathcal{P}_T(K)$ with probability at least $(\delta^{13})^{2|K|}$. Since $|K| \leq 4/\delta$, this probability is at least $(\delta^{13})^{8/\delta}$.

The running time of a call to partition is $O(m)$ as it simply scans the input quartets one by one, and performs a decision in constant time per quartet. When constructing $B_{v,j}$, each scanned quartet is first checked to have all its four elements in $\mathcal{L}(T) \setminus B$. If this is not the case, the quartet is ignored. If all the elements are in $\mathcal{L}(T) \setminus B$, then in the case of Section 3.1.1.1, for example, if w_2 is an element of the quartet then we increase the count for the leaves on the opposite side of w_2. At the end of the scan we therefore know for each $z \in \mathcal{L}(T) \setminus B$, the number of quartets of Q

in which w_2 and z are in opposite sides. As shown in Section 3.1.1.1, we use this count to decide whether to place z in $B_{v,j}$ or not. A similar sequential scan in performed in the other cases.

3.2 Constructing a tree from a close partition

In Section 3.1 we have proved that PARTITION, when called with the parameters (M_0, f_0), constructs, with constant positive probability, a partition $\mathcal{P}^*$ of $\mathcal{L}(T)$ that is close to $\mathcal{P}_T(K)$. Hence, if we run PARTITION(M_0, f_0) a constant number of times, we are guaranteed that, with high probability, it will construct a $\mathcal{P}^*$ that is close to $\mathcal{P}_T(K)$. To complete the proof of Theorem 1.1, it suffices to show that with high probability, a $\mathcal{P}^*$ that is close to $\mathcal{P}_T(K)$ can be used to construct a tree T^* that satisfies a fraction of $(1-\epsilon)$ elements of a random sample of size at least $Cn^2 \log n$.

So, for the remainder of this section we assume that $\mathcal{P}^*$ is close to $\mathcal{P}_T(K)$. Recall that $\mathcal{P}^* = \{B_{v,j} \mid v \in K \ , \ j = 0, 1\} \cup \{B^*\}$. For each $B_{v,j}$ we construct a tree $T_{v,j}$ as follows. $T_{v,j}$ is an arbitrary rooted full binary tree, except for the root which has a unique child, and whose set of leaves is precisely $B_{v,j}$. In the event that $B_{v,j} = \emptyset$ then we also define $T_{v,j}$ to be an empty tree. Notice that $T_{v,j}$ has precisely $2|B_{v,j}|$ vertices.

We construct a tree T^* by attaching to M_0 the $2|K|$ trees $T_{v,j}$ at appropriate places as follows. There are three cases. If $B_{v,j} = \emptyset$ we do nothing with $T_{v,j}$ as it is an empty tree. So assume that $B_{v,j} \neq \emptyset$. If v is a leaf of M_0 then we attach $T_{v,j}$ to M_0 by identifying the root of $T_{v,j}$ with v. Notice that both trees $T_{v,0}$ and $T_{v,1}$ are attached at v so v has two children in T^*. If v is an internal vertex of M_0, then it has two emanating edges, leading towards its children. Denote these edges by e_0 and e_1. We subdivide the edge e_j for $j = 0, 1$, introducing a new vertex and identify this new vertex with the root of $T_{v,j}$.

Notice that, considered as an unrooted tree (we can simply put an edge connecting the two children of the root of T^*, which is also the root of M_0, and eliminate the root, thereby making T^* unrooted), T^* is a phylogenetic tree. Furthermore $\mathcal{L}(T^*) = \mathcal{L}(T) - B^*$. We now prove that, with high probability, T^* satisfies a large fraction of the input quartet set.

Lemma 3.6. *For a random sample of m quartets, the expected number of quartets satisfied by T^* is at least $m(1-\epsilon/3)$. Hence, by Markov's Inequality, with probability at least $2/3$ we have that T^* satisfies at least $(1-\epsilon)m$ quartets.*

Proof. By linearity of expectation, it suffices to prove that a single randomly sampled quartet is satisfied by T^* with probability at least $1 - \epsilon/3$.

Let $E_{v,j} = A_{v,j} \cap B_{v,j}$. Call the sets $E_{v,j}$ the *essential sets*. The construction of T^* guarantees that if a, b, c, d are in pairwise *distinct* essential sets then the quartet they induce in T is identical to the quartet they induce in T^*. On the other hand, if one of a, b, c, d is not in an essential set, or if two of them are in the same essential set, this need not be the case.

Now, observe first that since $\mathcal{P}^*$ is close to $\mathcal{P}_T(K)$ then we have $|E_{v,j}| \geq |A_{v,j}| - 50\delta^2 n$. As there are $2|K| \leq 8/\delta$ possible sets $A_{v,j}$ we have that

$$\left| \bigcup_{(v,j) \in K \times \{0,1\}} E_{v,j} \right| \geq \left| \bigcup_{(v,j) \in K \times \{0,1\}} A_{v,j} \right| - 400\delta n = n(1 - 400\delta).$$

Thus, a randomly chosen leaf is not in an essential set with probability at most 400δ.

What is the probability that two leaves of a randomly sampled quartet $ab|cd$ are in the same part of $\mathcal{P}_T(K)$? As each part contains at most a δ fraction of the leaves, this probability is at most

δ. As there are 6 pairs in a, b, c, d, with probability at least $1 - 6\delta$ each leaf of $ab|cd$ is in a distinct part. Overall, using (1), we have that with probability at least

$$1 - 6\delta - 4 \cdot (400\delta) = 1 - 1606\delta > 1 - \epsilon/3$$

each element of a randomly sampled quartet is in a distinct essential set. Hence, a randomly sampled quartet is also a quartet of T^* with probability at least $1 - \epsilon/3$. □

4 Discussion

The problem of establishing an approximation value for maximum quartet consistency problem (MQC) that is better than the trivial 1/3 approximation is open for nearly two decades. MQC is a central problem in phylogenetics and has become even more so with the exponential growth of molecular data and the emergence of the supertree approach for large scale phylogenetic reconstruction [10]. This work extends the current knowledge in the two fronts: the type of approximation - from a constant factor to any ε, and the density of the input - from $\Theta(n^4)$ to $\Omega(n^2 \log n)$. On the other hand, it relies on the assumption that the input is a random sample of consistent quartets. As sampled input appears to be inevitable [12, 16], we believe this contribution is important. Furthermore, our algorithm can be easily generalized to allow for a small fraction of errors in the sampled input. To see this, observe that by increasing the value of the constant C' in Lemma 3.2 and Lemma 3.3 from $256/\delta^4$ to, say, $1000/\delta^4$ we can bound the deviation from the expectations stated in these lemmas to a value less than $qp\delta^2/8$ instead of the value $qp\delta^2/4$ used in these lemmas. Hence, even if we allow for a fraction of $\delta^2/8 = \Theta(\epsilon^2)$ errors, the statements of these lemmas stay intact.

The issue of whether MQC should serve as a reconstruction quality measurement was discussed in the past and naturally arises here. Indeed in simulation studies it is possible to compare the resulting tree to the model tree. The recent result of [16] shows that, in the supertree realm, this is not always optimal. Moreover, in real life situations, this tree is not known, or possibly does not exist (in case of conflicting input subtrees). Finally, we note that with the sparsity of the inputs handled in this paper, it is likely that an optimal tree is not unique.

Acknowledgments

We thank the referees for carefully reading the manuscript and for invaluable suggestions.

References

[1] A.V. Aho, Y. Sagiv, T.G. Szymanski, and J.D. Ullman. Inferring a tree from lowest common ancestors with an application to the optimization of relational expressions. *SIAM Journal on Computing*, 10(3):405–421, 1981.

[2] N. Alon and J. Spencer. *The Probabilistic Method, 2nd edition.* John Wiley, New York, 2000.

[3] V. Berry and O. Gascuel. Inferring evolutionary trees with strong combinatorial evidence. *Theoretical Computer Science*, 240(2):271–298, 2001.

[4] V. Berry, T. Jiang, P. Kearney, M. Li, and T. Wareham. Quartet cleaning: improved algorithms and simulations. In *European Symposium on Algorithms*, pages 313–324, 1999.

[5] G. Della Vedova, T. Jiang, J. Li, and J. Wen. Approximating minimum quartet inconsistency. In *Proceedings of the Thirteenth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 894–895, San Francisco, 2002. ACM/SIAM.

[6] G. Della Vedova and T. Wareham. Optimal algorithms for local vertex quartet cleaning. *Bioinformatics*, 18(10):1297–1304, 2002.

[7] I. Giotis and V. Guruswami. Correlation clustering with a fixed number of clusters. *Theory of Computing*, 2(1):249–266, 2006.

[8] M. Henzinger, V. King, and T. Warnow. Constructing a tree from homeomorphic subtrees, with applications to computational evolutionary biology. In *Proceedings of the seventh ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 333–340, Atlanta, 1996. ACM/SIAM.

[9] T. Jiang, P.E. Kearney, and M. Li. A polynomial time approximation scheme for inferring evolutionary trees from quartet topologies and its application. *SIAM Journal on Computing*, 30(6):1942–1961, 2000.

[10] U. Roshan, B.M.E. Moret, T.L. Williams, and T. Warnow. Rec-i-dcm3: A fast algorithmic technique for reconstructing large phylogenetic tree. In *Proceedings of the IEEE Computational Systems Bioinformatics conference (CSB)*, 2004.

[11] C. Semple and M.A. Steel. *Phylogenetics*. Oxford University Press, 2003.

[12] S. Snir and S. Rao. Quartets maxcut: A divide and conquer quartets algorithm. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 7(4):704–718, 2010.

[13] S. Snir and R. Yuster. Reconstructing approximate phylogenetic trees from quartet samples. In *Proceedings of the 21st ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1035–1044, Austin, 2010. ACM/SIAM.

[14] M. Steel. The complexity of reconstructing trees from qualitative characters and subtress. *Journal of Classification*, 9(1):91–116, 1992.

[15] K. Strimmer and A. von Haeseler. Quartet puzzling: A quartet maximum-likelihood method for reconstructing tree topologies. *Molecular Biology and Evolution*, 13(7):964–969, 1996.

[16] M. Swenson, Rahul Suri, C. Linder, and Tandy Warnow. An experimental study of quartets maxcut and other supertree methods. In Vincent Moulton and Mona Singh, editors, *Workshop on Algorithms in Bioinformatics*, volume 6293 of *Lecture Notes in Computer Science*, pages 288–299, 2010.

[17] V. Vazirani. *Approximation Algorithms*. Springer, 2004.

[18] G. Wu, M. Kao, G. Lin, and J. You. Reconstructing phylogenies from noisy quartets in polynomial time with a high success probability. *Algorithms for Molecular Biology*, 3, 2008.

[19] G. Wu, J. You, and G. Lin. A polynomial time algorithm for the minimum quartet inconsistency problem with $o(n)$ quartet errors. *Information Processing Letters*, 100(4):167–171, 2006.