Reconstructing approximate phylogenetic trees from quartet samples

Sagi Snir [*] Raphael Yuster [†]

Abstract

The reconstruction of evolutionary trees (also known as *phylogenies*) is central to many problems in Biology. Accurate phylogenetic reconstruction methods are currently limited to a maximum of few dozens of species. Therefore, in order to construct a tree over larger sets of species, a method capable of inferring accurately trees over small, overlapping sets, and subsequently merging these sets into a tree over the complete set, is required.

A quartet tree is the smallest informative piece of information and quartet based methods are based on combining quartet trees into a big tree. However, even this case is NP-hard, and even when the set of quartet trees is compatible (agree on a certain tree). The general problem of approximating quartets, or *maximum quartet consistency* (MQC), even for compatible inputs, is open for nearly twenty years. Despite its importance, the only rigorous results for approximating quartets are the naive 1/3 approximation that applies to the general case and a PTAS when the input is the complete set of all $\binom{n}{4}$ possible quartets.

Even when it is possible to determine the correct quartet induced by every four taxa, the time needed to generate the complete set of all quartets may be impractical. A faster approach is to sample at random just $m \ll \binom{n}{4}$ quartets, and provide this sample as an input.

In this work we present the first approximation algorithm whose guaranteed approximation is strictly better than 1/3 when the input is any random sample of m compatible quartets. The approximation ratio we obtain is 0.425 for general m, and 0.468 when $m = \tilde{\omega}(n^2)$. An important ingredient in our algorithm involves solving a weighted Max-Cut in a certain graph induced by the set of input quartets. We also show an extension of the PTAS algorithm to handle dense, rather than complete, inputs.

1 Introduction

The study of evolution and the construction of phylogenetic (evolutionary) trees (or phylogenies) are classical subjects in Biology. DNA sequences from a variety of organisms are rapidly accumulating, providing large amounts of data to various phylogenetic reconstruction methods. The goal behind the "tree of life" project is to accurately construct the tree representing the evolutionary history of over several millions of taxa (species). This task cannot be achieved by the traditional accurate reconstruction methods. Therefore, the need to design methods capable of amalgamating small, accurately inferred trees into a large tree emerges.

Perhaps the simplest version of this task is *quartet based reconstruction*, in which all input trees are quartet trees (or simply *quartets*) - trees over four taxa. The study of quartets is of prime importance as quartets are the smallest informational unit and hence, quartets play a major role in other reconstruction methods [2, 3, 4, 5, 8, 11].

A set of quartets can be *consistent* (or *compatible*) in which case all quartets agree on (or *satisfied* by) some complete tree, or *inconsistent* if there is no such tree. Nevertheless, despite its simplicity, even when all quartets are consistent with some tree, finding such a tree is NP-hard [15]. The general problem of approximating quartets, or *maximum quartet consistency* (MQC), even for consistent inputs, is open for nearly twenty years, and the best approximation ratio is 1/3, obtained naively by a random tree. Exact polynomial algorithms exist for special cases only (see e.g. [4]). The only rigorous approximation result is a polynomial time approximation scheme (PTAS) by Jiang *et al.* [9] for the case when all $\binom{n}{4}$ quartets exist in the input.

Generating a large set of correct quartets, based on biological data, may be time consuming. Therefore, preparing a large input (moreover a complete input of $\binom{n}{4}$ quartets) may be too costly and impractical even for relatively small datasets. A faster approach is to sample a relatively small number of $m \ll \binom{n}{4}$ 4-taxa sets, and generate the input corresponding to the m quartets they define, and try to solve MQC on this input.

In this paper, we devise a new approximation algorithm for the MQC problem. Given a set of m quartets sampled uniformly from the set of all $\binom{n}{4}$ quartets, our algorithm achieves an approximation ratio of 0.425. To the best of our knowledge, this is the first algorithm achieving a better ratio than the naive 1/3. When $m \geq Cn^2 \log n$, a modified version of our algorithm achieves an approximation ratio of 0.468.

An important ingredient in our algorithm involves solving a weighted MaxCut in a certain graph induced by the set of quartets, a technique proved to be practically efficient as a heuristic for the same task [13]. We build a weighted graph based on the set of input quartets. We rely on combinatorial properties of trees that enable us to rigorously prove a lower bound for

[*] Department of Evolutionary Biology, University of Haifa, Haifa 31905, Israel. E–mail: ssagi@research.haifa.ac.il

[†] Department of Mathematics, University of Haifa, Haifa 31905, Israel. E–mail: raphy@math.haifa.ac.il

the maximum cut in our graph. Next we use the MaxCut approximation algorithm of Goemans and Willamson [7] to compute an approximate maximum cut in the graph. However, since our graph contains negative weights, special care must be taken in the establishment of the lower bound. We then translate our solution to a solution of MQC, by amalgamating the cut solution with an approximate solution from each of the parts separated by the cut.

When the input is dense enough so that $m = \Omega(n^2 \log n)$, we can further exploit the structure of the tree and also have some control over the negative weights in our graph. This yields a better approximation in this case.

Finally, we show how to generalize the aforementioned PTAS of Jiang *et al.* [9] to the case where $m = \Theta(n^4)$ (and is not necessarily a complete input). This generalization works for arbitrary input.

2 Quartets MaxCut

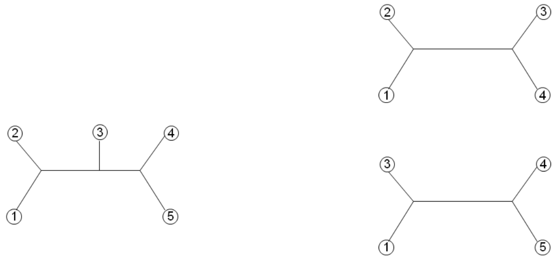
In this section we describe the central tool used in our approximation algorithms *Quartets MaxCut* (QMC). Before we present the algorithm, we provide some basic definitions, we formally describe the graph induced by a set of quartets, and provide some properties of this graph.

2.1 Preliminaries A *cut* $C = (S, \bar{S})$ in an edge-weighted graph G is a partition over the vertex set $V = V(G)$ into non-empty parts S and $\bar{S} = V - S$. An edge *belongs to the cut* if its endpoints are in distinct parts. We denote the set of edges of the cut by E_C. The *weight* $w(C)$ of a cut C is the sum of weights of the edges in E_C. A *maximum cut* of G is a cut of G with maximum weight. The task of finding a maximum cut is the *MaxCut* problem.

Recall that a full binary rooted tree is a tree whose internal vertices have two children each. A full binary undirected (also called unrooted) tree is a tree whose internal vertices have three neighbors each (a trivalent tree). Throughout this paper, all trees are assumed to be full unrooted binary trees, with leaves labeled bijectively by a taxa set $\mathcal{X}$. Such trees are called *phylogenetic trees*. For a tree $T = (V, E)$, we denote by $\mathcal{L}(T)$ the set of leaves of T.

The removal of an edge e in a tree splits the tree into two subtrees and therefore induces a split among the leaves of the tree. We identify an edge e by the split $(U, \mathcal{X} \setminus U)$ it generates and denote it by e_U. 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 where all leaves in $\mathcal{X} \setminus A$ and paths leading exclusively to them are removed, and subsequently internal vertices with degree two are contracted.

For two trees T and T', we say that T *satisfies* T', and 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 $\bigcup_i \mathcal{L}(T_i)$ that satisfies every tree $T_i \in \mathcal{T}$ (see Figure 1). Otherwise, $\mathcal{T}$ is *inconsistent*. When $\mathcal{T}$ is inconsistent, it is desirable to find a tree T^* over $\bigcup_i \mathcal{L}(T_i)$ that maximizes some objective function. T^* is denoted a *supertree* and the problem of finding T^* is the *supertree problem.*



(a) (b)

Figure 1: (a) A phylogenetic tree over five leaves. (b) Two trees over four leaves induced by the tree on the left.

A *quartet* tree (or just a quartet for short), is an undirected tree over four leaves $\{a, b, c, d\}$. We write a quartet over $\{a, b, c, d\}$ as $((a, b), (c, d))$ if there exists an edge e_U such that $a, b \in U$ and $c, d \notin U$. A 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 such that $|\mathcal{Q}_s(T)|$ is maximized. The problem is denoted as *maximum quartet consistency* (MQC). We note that MQC is NP-hard even when $\mathcal{Q}$ is consistent [15].

Let T be any tree with n leaves. Consider a random bijection π between a taxa set $\mathcal{X}$ of size n and the leaves of T. The corresponding labeled tree is denoted by T^π. As each of the $n!$ possible bijections is equally likely, we notice that a quartet $((a, b), (c, d))$ with labels from $\mathcal{X}$ is satisfied by T^π with probability $1/3$. We therefore have, by linearity of expectation:

Lemma 2.1. *Let $\mathcal{Q}$ be an arbitrary set of quartets over a taxa set $\mathcal{X}$ of size n, and let T^π be a random bijection between the leaves of a tree T and $\mathcal{X}$. Then the expected number of elements in $\mathcal{Q}$ satisfied by T is $|\mathcal{Q}|/3$.*

2.2 The Quartets MaxCut approach The quartets MaxCut (QMC) is a divide and conquer algorithm that operates on the taxa set $\mathcal{X}$ by first partitioning the set into a partition $\mathcal{P}$ of two or more parts. Then it operates on the sub problems induced by each part, and merges the sub-solutions obtained into a complete solution. In our cases the partition $\mathcal{P}$ is of two or three parts.

Let $\mathcal{Q}$ be a set of quartets with $|\mathcal{Q}| = m$. A quartet $q = ((a,b),(c,d)) \in \mathcal{Q}$ is said to be *unaffected* by a partition $\mathcal{P}$, if all $\{a,b,c,d\}$ are in one part of $\mathcal{P}$. Otherwise, it is *affected* by $\mathcal{P}$. For an affected $q = ((a,b),(c,d))$ we say that q is *satisfied* by $\mathcal{P}$ if some part contains precisely a and b, or some part contains precisely c and d. It is *violated* by $\mathcal{P}$ if at least one of the pairs a,c or a,d or b,c or b,d are in the same part, and the other two are not in that part. Otherwise, we have that one part contains only one of $\{a,b,c,d\}$ and some other part contains the other three. In this case we say that q is *deferred*. At every step of the algorithm, some quartets are satisfied, some are violated, and some continue to the next steps (i.e. either deferred or unaffected). A plausible strategy is to maximize the ratio between satisfied and violated quartets at every step. We will show that this strategy has *guaranteed* performance.

Given the set of quartets $\mathcal{Q}$ over a taxa set $\mathcal{X}$, we build the following weighted (multi) graph $G = G(\mathcal{Q}) = (V,E)$ with $V = \mathcal{X}$ and E as follows: For every $q = ((a,b),(c,d)) \in \mathcal{Q}$ we add the six edges $\{(a,c),(a,d),(b,c),(b,d),(a,b),(c,d)\}$ to E. We distinguish between the edges and denote the edges $\{(a,c),(a,d),(b,c),(b,d)\}$ as *good edges* and $\{(a,b),(c,d)\}$ as *bad edges*. Observe that between two vertices in $G(\mathcal{Q})$ there can be good and bad edges simultaneously, originating from different quartets (see Figure 2). We denote by E_g and E_b the set of good and bad edges respectively.

We note that a cut in G corresponds to a partition of the taxa set into two parts. Given a cut $C = (S,\bar{S})$ in the graph $G(\mathcal{Q})$ we notice:

Observation 2.1. *For an affected quartet $q \in \mathcal{Q}$*
1. q contributes 4 good edges to the cut if q is satisfied.
2. q contributes 2 good edges and 1 bad edge to the cut if q is deferred.
3. q contributes 2 good edges and 2 bad edges to the cut if q is violated.

Figure 2 shows graphically the effect of a cut in a graph on the two quartets generating that graph. The above observation links between the number of quartets satisfied/violated/deferred and the number of good/bad edges in the cut. Once we have defined $G(\mathcal{Q})$ we seek to find a cut C maximizing

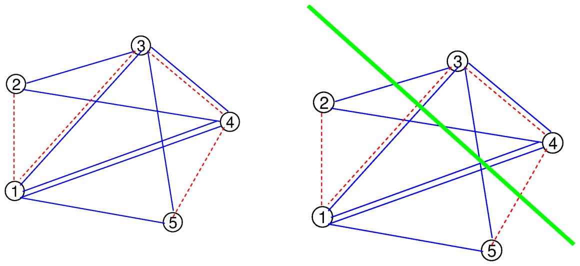


Figure 2: Left: The graph induced by the quartets $((1,2),(3,4))$ and $((1,3),(4,5))$ from Figure 1. Right: A cut separating $\{1,2,5\}$ from $\{3,4\}$, therefore satisfies quartet $((1,2),(3,4))$ but violates $((1,3),(4,5))$ and hence contains 6 good and 2 bad edges.

$$|E_g \cap E_C| - \alpha |E_b \cap E_C| \tag{2.1}$$

where $\alpha > 0$ is a *weight ratio parameter* between good and bad edges.

Solving (2.1) is equivalent to finding a maximum cut (solving MaxCut) in a graph in which good edges have unit weight and bad edges have weight $-\alpha$. However, since MaxCut is NP-hard [6], we use the seminal approximation algorithm of Goemans and Williamson [7] (denoted here by GW). The principle in GW is to embed the vertices on the unit n-dimensional sphere and effectively solve a relaxed semidefinite program.

3 A general approximation algorithm

We present a polynomial time approximation algorithm whose expected approximation ratio is 0.425 when the input is consistent and, more generally, an expected approximation ratio of $0.425 - 0.26\eta$ when a fraction $1-\eta$ of the input is non-conflicting. In both cases it is assumed that the m input quartets are generated uniformly at random from the set of all $\binom{n}{4}$ quartets.

3.1 A 0.425 approximating algorithm for consistent input We assume a set $\mathcal{Q}$ of m quartets over a taxa set $\mathcal{X}$ with $|\mathcal{X}| = n$ are generated as follows. Four taxa are chosen randomly and the correct quartet induced by them is taken. This process is repeated m times, independently.

Notice that since our quartets are consistent there exists *some* tree T which satisfies all the input quartets (the problem is, of course, that we don't know T, we just know it exists). A well known property [12] is that any binary tree, such as T, contains an edge s_0 that induces a $(\mathcal{X}_1, \mathcal{X} \setminus \mathcal{X}_1)$ partition over the leaves where

$|\mathcal{X}_1| = \beta$ and $\frac{1}{3} \leq \beta \leq \frac{1}{2}$. We denote it as the $(\frac{1}{3}, \frac{2}{3})$-property. We also set the weight ratio parameter from Equation (2.1) to $\alpha = 2$, as this choice maximizes our performance guarantee. Let $w(s_0)$ denote the weight of the cut in $G(\mathcal{Q})$ corresponding to the cut $(\mathcal{X}_1, \mathcal{X} \setminus \mathcal{X}_1)$ induced by s_0. The following lemma establishes a lower bound on the expectation $E[w(s_0)]$ of $w(s_0)$.

Lemma 3.1. *$E[w(s_0)] \geq \frac{32m}{27}$.*

Proof. Let us define three disjoint sets as follows:
$\mathcal{Q}_u(s_0)$ - the set of quartets unaffected by s_0. Denote $|\mathcal{Q}_u(s_0)|$ by $m_u(s_0)$,
$\mathcal{Q}_d(s_0)$ - the set of quartets deferred by s_0. Denote $|\mathcal{Q}_d(s_0)|$ by $m_d(s_0)$,
$\mathcal{Q}_s(s_0)$ - the set of quartets satisfied by s_0. Denote $|\mathcal{Q}_s(s_0)|$ by $m_s(s_0)$.
When it is clear from the context, we will remove the split indication (s_0). We also note that since $\mathcal{Q}$ is consistent, no quartets are violated by s_0 and we get: $\mathcal{Q} = \mathcal{Q}_u \cup \mathcal{Q}_d \cup \mathcal{Q}_s$.

Now, the weight of a cut is the sum of contributions of all quartets affected by it. In particular, at s_0 we get:

$$(3.2) \quad w(s_0) = 0m_u + (2-\alpha)m_d + 4m_s = 4m_s \ .$$

We claim that:
$E[m_u] = m(\beta^4 + (1-\beta)^4)$.
$E[m_d] = 4m(\beta^3(1-\beta) + \beta(1-\beta)^3)$.
$E[m_s] = 6m(\beta^2(1-\beta)^2)$.

Indeed, for a quartet to be unaffected by s_0 all its taxa must "fall" in the same partition. The probability of all taxa falling in the smaller side is β^4 and the probability of all taxa falling at the bigger side is $(1-\beta)^4$. For a quartet to be deferred by s_0 one taxa must fall at one part and all the other taxa at the other part. There are four possibilities to choose this single taxa, so the probability of a single taxa falling at the smaller part is $4\beta(1-\beta)^3$ and at the bigger part the probability is $4(1-\beta)\beta^3$. For a quartet to be satisfied by s_0 two of its taxa must fall at one part and the other must fall at the other part. There are six possibilities to choose a pair of taxa for the smaller part and hence the probability is $6\beta^2(1-\beta)^2$. The claim on expectations now directly follows by linearity of expectation as there are m quartets.

Together with (3.2) and with $\alpha = 2$ we have:

$$(3.3) \qquad E[w(s_0)] = 24m(\beta^2(1-\beta)^2) \ .$$

(3.3) has two extremal points at $\frac{1}{2}$ and 0, so in the interval $[\frac{1}{3}, \frac{1}{2}]$ the minimum is obtained on the boundary $\frac{1}{3}$ and the lemma follows. ■

We note that all claims on expectations in the last lemma and anywhere else in this paper can be also phrased in terms of high concentration, using a standard large deviation Chernoff estimate (see, e.g., [1], Appendix A). For example, one can easily obtain that for every $\epsilon > 0$, $\Pr[w(s_0) < (\frac{32}{27} - \epsilon)m] < e^{-\epsilon^2 m/32}$. However, for clarity of exposition, we state and prove all our results in terms of expectations.

The lower bound on the weight of s_0 (Lemma 3.1) serves as a lower bound for a maximum cut C^* in $G(\mathcal{Q})$. Let C_{GW} denote the cut returned by the GW algorithm.

Lemma 3.2. *$E[w(C_{GW})] \geq 0.552m$.*

Proof. GW achieves an approximation ratio of 0.878 if all weights are non-negative. However, GW has a special consideration for negative weights: In this case C_{GW} satisfies $w(C_{GW}) \geq 0.878w(C^*) - 0.122w^-$ where w^- is the absolute sum of the negative weights in the graph. In our case we have $|\mathcal{Q}| = m$ quartets, each one has two negative edges with weight -2. Therefore we obtain:

$$(3.4) \quad E[w(C_{GW})] \geq 0.878E[w(C^*)] - 0.122w^-$$
$$\geq 0.878\frac{32m}{27} - 0.122 \cdot 4m = 0.552m \ .$$

■

Recall that we are interested in satisfying the maximum number of quartets and not necessarily finding the optimal cut in $G(\mathcal{Q})$. Moreover, while s_0 corresponds to a real split in T and therefore no quartets are violated at s_0 this is *not* necessarily true for C_{GW}.

Definition 3.1. *$\mathcal{Q}_v(C_{GW})$ denotes the set of quartets violated by C_{GW} and $m_v(C_{GW}) = |\mathcal{Q}_v(C_{GW})|$.*

Observation 3.1.

$$(3.5) \quad E[m_v(C_{GW})] \leq 2E[m_s(C_{GW})] - 0.276m.$$

Proof. By Observation 2.1 we get that a quartet violated by C_{GW} contributes two bad and two good edges. Now since $\alpha = 2$ we get:

$$(3.6) \qquad w(C_{GW}) = 4m_s(C_{GW}) - 2m_v(C_{GW}).$$

By Lemma 3.2 the result follows. ■

So far we have dealt with quartets that are either satisfied or violated by the first cut. However, we are left with the sets of quartets deferred by C_{GW}, denoted by $\mathcal{Q}_d(C_{GW})$, and quartets unaffected by C_{GW}, denoted by $\mathcal{Q}_u(C_{GW})$. The cut C_{GW} partitions the taxa set $\mathcal{X}$ into $\{\mathcal{X}_1, \mathcal{X}_2\}$. We thus have that each quartet in $\mathcal{Q}_u(C_{GW})$ is either a quartet whose leaves are in $\mathcal{X}_1$ or else its leaves are in $\mathcal{X}_2$. Similarly, each quartet in $\mathcal{Q}_d(C_{GW})$ now corresponds to a rooted triplet in $\mathcal{X}_1$ (in the case where the quartet has a single taxa in $\mathcal{X}_2$) or to a rooted triplet in $\mathcal{X}_2$ (in the case where the quartet has a single taxa in $\mathcal{X}_1$). As in Lemma 2.1 we can construct labeled rooted trees T_1 and T_2 where the leaves of T_i are labeled by $\mathcal{X}_i$ and so that at least 1/3 of these triplets and quartets are satisfied. This yields the following approximation algorithm for quartet consistency - MaxCut Approx:

```
MCA($\mathcal{Q}$)
1. Construct $G(\mathcal{Q})$ with $\alpha = 2$.
2. Apply GW MaxCut on $G(\mathcal{Q})$ and obtain a
   partition $\{\mathcal{X}_1, \mathcal{X}_2\}$ over the taxa set $\mathcal{X}$.
3. Construct a random tree $T_1$ with root $\rho_1$
   over $\mathcal{X}_1$ and similarly $T_2$ over $\mathcal{X}_2$.
4. Let $T$ be the tree obtained by connecting
   $\rho_1$ and$\rho_2$.
5. Return $T$.
```

Theorem 3.1. *Let T_{MCA} be the tree returned by the MCA algorithm applied on a consistent set of randomly chosen quartets $\mathcal{Q}$ with $m = |\mathcal{Q}|$. Then we have $E[|\mathcal{Q}_s(T_{MCA})|] \geq 0.425m$.*

Proof. Recall that $\mathcal{Q} = \mathcal{Q}_u(C_{GW}) \cup \mathcal{Q}_d(C_{GW}) \cup \mathcal{Q}_s(C_{GW}) \cup \mathcal{Q}_v(C_{GW})$. Also note that the total set of satisfied quartets, $\mathcal{Q}_s(T)$ is the union of the three disjoint sets $\mathcal{Q}_s(C_{GW}) \cup \mathcal{Q}_s(T_1) \cup \mathcal{Q}_s(T_2)$ (we slightly abuse notation here since $\mathcal{Q}_s(T_i)$ actually consists of quartets and triplets, but recall that each satisfied triplet corresponds to a deferred quartet of C_{GW} that was eventually satisfied). Now, by Lemma 2.1,

$$\begin{aligned}
& |\mathcal{Q}_s(T_{MCA})| \\
= \;& |\mathcal{Q}_s(C_{GW}) \cup \mathcal{Q}_s(T_1) \cup \mathcal{Q}_s(T_2)| \\
= \;& m_s(C_{GW}) + \frac{1}{3}(m_u(C_{GW}) + m_d(C_{GW})) \\
= \;& m_s(C_{GW}) + \frac{1}{3}(m - (m_s(C_{GW}) + m_v(C_{GW}))) \\
= \;& \frac{1}{3}m + \frac{2}{3}m_s(C_{GW}) - \frac{1}{3}m_v(C_{GW}).
\end{aligned}$$

Hence, when we take expected values we get that $E[|\mathcal{Q}_s(T_{MCA})|] = \frac{m}{3} + \frac{2}{3}E[m_s(C_{GW})] - \frac{1}{3}E[m_v(C_{GW})]$. However, recall that by Observation 3.1 we know that $2E[m_s(C_{GW})] - E[m_v(C_{GW})] \geq 0.276m$. Therefore we obtain:

$$E[|\mathcal{Q}_s(T_{MCA})|] \geq \frac{m}{3} + \frac{0.276m}{3} \;. \tag{3.7}$$

■

3.2 Noisy input analysis

The analysis in the previous subsection assumes an error free data. However, realistic inputs are sometimes subject to noise and therefore we cannot always expect all quartets to be consistent. Our algorithm MCA can be adapted to achieve performance guarantee when a bounded fraction of the quartets are erroneous.

Theorem 3.2. *When a fraction $1 - \eta$ of the set of m quartets agree with T, the expected number of quartets satisfied by the tree returned by algorithm MCA is at least $0.425m - 0.26\eta m$.*

Proof. Notice that the case $\eta = 0$ coincides with the analysis in the noise-free case. In fact, the proof proceed along the lines of the analysis of the noise-free case, and generalizes it. Recall the split s_0 in T that partitions the taxa set into two parts where each part is of size at least $\frac{n}{3}$. In the current case however, in addition to the three disjoint sets $\mathcal{Q}_u(C_{s_0}), \mathcal{Q}_d(C_{s_0}), \mathcal{Q}_s(C_{s_0})$ we also have $\mathcal{Q}_v(C_{s_0})$ - the set of quartets violated at s_0 and equivalently we denote them as $m_v(s_0)$. As in the proof of Lemma 3.1 we have:

$$\begin{aligned}
E[m_u] &= m(\beta^4 + (1-\beta)^4), \\
E[m_d] &= 4m(\beta^3(1-\beta) + \beta(1-\beta)^3), \\
E[m_s] &= 6(1-\eta)m(\beta^2(1-\beta)^2), \\
E[m_v] &= 6\eta m(\beta^2(1-\beta)^2).
\end{aligned}$$

Similarly we obtain:

$$E[w(s_0)] \geq 4\cdot 6(1-\eta)m\frac{4}{81} - 2\cdot 6\eta m\frac{4}{81} = m\frac{16}{27}(2-3\eta) \;.$$

Again we use this value as lower bound for the maximum cut in $G(\mathcal{Q})$. Recall that the cut returned by the GW, C_{GW} satisfies

$$\begin{aligned}
& E[w(C_{GW})] \\
\geq \;& 0.878\left(\frac{16m}{27}(2-3\eta)\right) - 0.122 * 4m \\
= \;& 4E[m_s(C_{GW})] - 2E[m_v(C_{GW})] \;.
\end{aligned} \tag{3.8}$$

Similarly to the noise-free case, the two random trees at both sides of C_{GW} satisfy 1/3 of the quartets either deferred or unaffected by the MaxCut step. Since the total number of satisfied quartets by the MCA algorithm is the sum of the quartets satisfied at the MaxCut stage and the random tree stage, we obtain:

$$\begin{aligned}
& |\mathcal{Q}_s(T_{MCA})| \\
= \;& m_s(C_{GW}) + \frac{1}{3}(m_u(C_{GW}) + m_d(C_{GW})) \\
= \;& m_s(C_{GW}) + \frac{1}{3}(m - m_s(C_{GW}) - m_v(C_{GW})) \\
= \;& \frac{m}{3} + \frac{1}{3}(2m_s(C_{GW}) - m_v(C_{GW})) \;.
\end{aligned}$$

By (3.8) we obtain that $E[|\mathcal{Q}_s(T_{MCA})|] \geq \frac{m}{3} + \frac{1}{3} \cdot \frac{1}{2}\left(0.878\left(\frac{16m}{27}(2-3\eta)\right) - 0.122 * 4m\right)$. ■

4 Improved approximations for mildly dense instances

As in the previous section, we assume a set $\mathcal{Q}$ of m quartets are generated randomly from a tree T over $\mathcal{X}$ leaves with $|\mathcal{X}| = n$. The previous section assumed nothing about m (the density of the input). In this section we prove that if the input is mildly dense, then a better approximation algorithm can be obtained. By *mildly dense* we require that each pair of taxa appear together in more than a constant number of quartets. For our purposes it would suffice that $m \geq Cn^2 \log n$ for some suitable absolute constant C. Notice that such instances may still be very far from dense instances as the latter have $m = \Theta(n^4)$.

Our improved algorithm is based upon two nontrivial ingredients that we add to the quartets Max-Cut algorithm, each contributing its own improvement to the approximation ratio. We call them the *exact*-3-*cut* method and the *negative edge cancellation* method. Already the exact 3-cut method yields the claimed improvement to 0.468 mentioned in the introduction. The negative edge cancellation method improves this slightly more to at least 0.472. Due to space limitations we will only describe in full the exact 3-cut method, while the negative edge cancellation method will be more briefly sketched.

4.1 Exact 3-cut We design an approximation algorithm that satisfies an expected number of $0.4689m$ quartets whenever $m > Cn^2 \log n$.

Consider an edge of T whose removal from T partitions the leaves to sets of size zn and $(1-z)n$ which are as balanced as possible (namely $z \leq 1/2 \leq 1-z$ and z is maximal). Recall also that by the $(\frac{1}{3}, \frac{2}{3})$-property, $z \geq 1/3$.

Observation 4.1. *When $z \geq 6/13 - \delta$, where δ is a sufficiently small absolute constant, the algorithm from Section 3 achieves approximation ratio of $0.4689m$.*

Proof: Using the same analysis an in the Section 3, with $\beta = 6/13 - \delta$ instead of $\beta = 1/3$ in (3.3), yields, in Lemma 3.1, that $E[w(s_0)] \geq 1.4823m$. Lemma 3.2 gives, in turn, that $E[w(C_{GW})] \geq 0.8134m$. This, in turn, gives in (3.7) that $E[|\mathcal{Q}_s(T_{MCA})|] \geq \frac{m}{3} + \frac{0.8134m/2}{3} \geq 0.4689m$. ■

We therefore proceed as follows. We run the MCA algorithm and see if the tree it constructs gives us at least $0.4689m$ satisfied quartets. If yes, then we are done. Otherwise, we *know* that any edge of T splits the leaves to two sets where the smaller set has at most $6n/13 - \delta n$ elements in it. We may now suppose that $1/3 \leq z < 6/13 - \delta$.

Lemma 4.1. *There is an internal vertex x of T whose removal from T creates three sub-trees that partition the set of leaves into three parts of sizes $\alpha n, \beta n, \gamma n$ where $\alpha \leq \beta \leq \gamma$, $\alpha + \beta + \gamma = 1$, and $\alpha \geq 1/13 + 2\delta$ and $\gamma \leq 6/13 - \delta$.*

Proof: Consider an edge $e = (x, y)$ that partitions the leaves optimally as above into sets of size zn and $(1-z)n$. We know that $z \leq 6/13 - \delta$ and $1 - z \geq 7/13 + \delta$. Suppose that x belongs to the part with at least $(1-z)n \geq (7/13+\delta)n$ leaves. If we remove x from the tree we get one part with at most $(6/13-\delta)n$ leaves and two other parts having sum at least $(7/13 + \delta)n$. None of these two other parts can have size smaller than $(1/13 + 2\delta)n$ as otherwise this would have contradicted the optimality of the partitioning edge e. ■

Let, therefore (A, B, C) be some partition of the leaves obtained by removing an internal vertex of T, so that $|A| = \alpha n$, $|B| = \beta n$ and $|C| = \gamma n$ and α, β, γ satisfy the conditions of Lemma 4.1. We know that such a partition exists, so let's fix one. We will describe a polynomial time algorithm that, in each step, generates a partition of the leaves, and that with high probability, one of the partitions it generates is *precisely* (A, B, C).

Before describing the details of the algorithm, we need a definition and a lemma. Let $q = 12m/n^2$ and notice that q is the expected number of quartets containing a given pair of taxa. For a vertex $x \in A$ we say that x is *A-interior*, if for all $y \in B \cup C$, there are at least $(2/3)(\alpha^2 - \delta)q$ quartets in which (x, y) is a good edge.

Lemma 4.2. *With very high probability there exists an A-interior vertex.*

Proof: Fix $y \in B \cup C$. Since in each quartet that contains y and three vertices of A, two of these three vertices have a good edge with y, it follows that there exists $x \in A$ that in at least $2/3$ of the quartets that contain x, y and two additional vertices of A, (x, y) is a good edge. But for such an x and y, the expected number of such quartets is $\alpha^2 q$. Since $q > C \log n$, a standard Chernoff large deviation inequality shows that with very high probability, there are at least $(\alpha^2 - \delta)q$ such quartets. Thus, we have proved that for this specific y, there exists with very high probability an $x \in A$ so that there are at least $(2/3)(\alpha^2 - \delta)q$ quartets in which (x, y) is a good edge. But notice that if in any quartet (x, u, v, y) with $u, v \in A$ we have that (x, y) is a good edge, then in any quartet (x, u, v, y') with $y' \in B \cup C$ we also have that (x, y') is a good edge. Hence, this property of x is guaranteed to hold with high probability for *all* $y \in B \cup C$. Hence, such an x is A-interior. ■

Similarly, we define a vertex $x \in B$ as B-interior, if for all $y \in A \cup C$, there are at least $(2/3)(\beta^2 - \delta)q$ quartets in which (x, y) is a good edge. Similarly, with very high probability there exists a B-interior vertex.

Our algorithm proceeds as follows. Consider all possible quintuples (x, y, a, b, c) where x and y are distinct leaves, and a, b, c are positive integers satisfying $a + b + c = n$, $(1/13 + 2\delta)n \le a \le b \le c \le (6/13 - \delta)n$. Notice that there are $O(n^4)$ possible quintuples. Our goal is to let x play the role of an A-interior vertex, let y play the role of a B-interior vertex, and let a, b, c play the roles of the cardinalities of A, B, C respectively. Since we are scanning all possible quintuples, we know from Lemma 4.1 and Lemma 4.2 that with very high probability, at least one quintuple matches these roles.

So fix a quintuple (x, y, a, b, c) for which x is A-interior, y is B-interior, and a, b, c are the correct cardinalities. We now show that when our algorithm examines *this* quintuple, it will, with very high probability, precisely construct A, B, C.

What can we say about the edges (x, u) whenever $u \in B \cup C$? The expected number of quartets in which such an edge is good is, by Lemma 4.2 *at least* $q(2\alpha\beta + 2\alpha\gamma + 2\beta\gamma + (2/3)(\alpha^2 - \delta))$. On the other hand, if $u \in A$ then the expected number of quartets in which such an edge is good is *at most* $q(1 - (1 - \alpha)^2)$. Now, there is a clear separation between these two values. Indeed,

$$[2\alpha\beta + 2\alpha\gamma + 2\beta\gamma + 2/3(\alpha^2 - \delta)] - [1 - (1 - \alpha)^2] = \\ 2\beta\gamma - (1/3)\alpha^2 - (2/3)\delta^2 > 1/9 .$$

This implies that A can be constructed precisely; we will place in it precisely those u for which the number of quartets in which (x, u) is good is at most $q(1 - (1 - a/n)^2) + \delta q$. We write a/n instead of α to stress the point that the algorithm does not know the value of α, but when scanning the "correct" quintuple it will be the case that $\alpha n = a$. Notice that we also added a "confidence" of δq since $q(1 - (1 - \alpha)^2)$ is only an expectation, and hence with very high probability we do not deviate from it by more than δq.

Similarly, when y is B-interior, what can we say about the edges (y, u) whenever $u \in A \cup C$? The expected number of times such an edge is good is *at least* $q(2\alpha\beta + 2\alpha\gamma + 2\beta\gamma + (2/3)(\beta^2 - \delta))$. On the other hand, if $u \in B$ then the expected number of times such an edge is good is *at most* $q(1 - (1 - \beta)^2)$. Again, we can show that there is a clear separation between these two values. Indeed,

$$[2\alpha\beta + 2\alpha\gamma + 2\beta\gamma + 2/3(\beta^2 - \delta)] - [1 - (1 - \beta)^2] = \\ 2\alpha\gamma - (1/3)\beta^2 - (2/3)\delta^2 > 1.5\delta .$$

(Here we use the fact that $\alpha \ge 1/13 + 2\delta$ and $\gamma \le 6/13 - \delta$). This implies that B can be constructed precisely. But this also means that with very high probability, when scanning the "correct" (x, y, a, b, c), then also the correct (A, B, C) is constructed. Notice that non-correct quintuples may either fail to construct a 3-cut (since a vertex u may "want" to be both in A and in B) or may construct a 3-cut for which some quartets are violated. But the point is that we are guaranteed that some quintuple (namely, the correct one) will generate a cut (A, B, C) for which no quartet is violated. Moreover, for this cut, any quartet that does not have three or more leaves in the same part is satisfied by the partition. This means that we satisfy all but a fraction of

$$\alpha^4 + \beta^4 + \gamma^4 + 4(\alpha^3\beta + \beta^3\alpha) + 4(\alpha^3\gamma + \gamma^3\alpha) + 4(\gamma^3\beta + \beta^3\gamma)$$

quartets. If we now construct random trees on each of the three parts we satisfy an expected amount of 1/3 of these deferred and unaffected quartets. Overall, the expected number of satisfied quartets is at least a fraction of

$$1 - (2/3)(\alpha^4 + \beta^4 + \gamma^4 + 4(\alpha^3\beta + \beta^3\alpha) + \\ 4(\alpha^3\gamma + \gamma^3\alpha) + 4(\gamma^3\beta + \beta^3\gamma))$$

of the quartets, and this value is much larger than 0.4689, as required.

4.2 Negative edge cancellation When we applied Lemma 3.2, we have assumed the worst case scenario that the total sum of the negative weights is $4m$. This, of course, is always true if we think of $G(\mathcal{Q})$ as a multigraph; it has precisely $2m$ bad edges with weight -2 each. However, the MaxCut algorithm treats $G(\mathcal{Q})$ as a weighted graph. That is, if u, v have x good edges between them and y bad edges between them then the weight of that edge is $x - 2y$. In other words, the x good edges cancel some (or maybe all) of the negative weights incurred by the pair u, v. Now, if the number of input quartets is small (say $m = o(n^2)$) then most pairs u, v have only a single edge between them, and indeed the total negative weight is $4m(1 - o(1))$. This, however, ceases to be the case when $m = \omega(n^2)$. In this case, the expected multiplicity of the edge (u, v) in the multigraph $G(\mathcal{Q})$ is larger than a constant. Moreover, if u, v are a pair of vertices corresponding to pair of taxa that are sufficiently far apart in the tree T, then, in fact, we can quantify the amount of negative cancellation that we obtain in the pair u, v.

Let us be slightly more concrete. Let e_A be an edge separating the tree into two subtrees T_1 and T_2 where $\mathcal{L}(T_1)$ has size αn and $\mathcal{L}(T_2)$ has size $(1 - \alpha)n$, where $1/3 \leq \alpha \leq 1/2$. Furthermore, let e_B be an edge separating T_2 into two subtrees T_3 and T_4 where the respective sizes of $\mathcal{L}(T_3)$ and $\mathcal{L}(T_4)$ are $(1-\alpha)\beta n$ and $(1-\alpha)(1-\beta)n$, and $1/3 \leq \beta \leq 2/3$. Assume w.l.o.g. that e_A is incident with a vertex in T_3. Let $u \in \mathcal{L}(T_1)$ and $v \in \mathcal{L}(T_3)$. Now consider two types of quartets that contain u and v. The first one (we call q_1-type), is of the form $((u,v),(i,j))$ where $i,j \in \mathcal{L}(T_4)$; these quartets contribute a bad edge between u and v. The expected number of such quartets is

$$(4.9) \qquad \frac{12m}{n^2}((1-\alpha)(1-\beta))^2 + o(m/n^2) \ .$$

The second one (q_2-type) is of the form $((i,u),(v,j))$ where $i \in L(T_1)$ and $j \in \mathcal{L}(T_2)$; these quartets contribute a good edge between u and v. The expected number of such quartets is

$$(4.10) \qquad \frac{24m}{n^2}\alpha(1-\alpha) + o(m/n^2) \ .$$

Since bad edges have negative weight (-2), while good edges have positive weight ($+1$), and since taking (4.9) twice is still much smaller than taking (4.10) once, we have that *all* negative weight between u, v is canceled. As there are $\alpha(1-\alpha)\beta n^2 - o(n^2)$ possible pairs u, v with $u \in \mathcal{L}(T_1)$ and $v \in \mathcal{L}(T_3)$ we have that the expected negative weight cancellation is

$$(4.11)\, \alpha(1-\alpha)\beta n^2 \frac{24m}{n^2}(1-\alpha)^2(1-\beta)^2 - o(m) \ .$$

Similarly and symmetrically, we could have done the same thing by splitting T_1 into two subtrees T_5 and T_6 via an edge e_C, with respective sizes of $\mathcal{L}(T_5)$ and $\mathcal{L}(T_6)$ being $\alpha\gamma n$ and $\alpha(1-\gamma)n$, and $1/3 \leq \gamma \leq 2/3$. Assume w.l.o.g. that e_C is incident with a vertex in T_5. This time, however, we will only consider pairs $u \in \mathcal{L}(T_4)$ and $v \in L(T_5)$ (we cannot allow u to be in $\mathcal{L}(T_3)$ since such pairs have already been accounted for in the previous case). We will obtain an analogous expression to expression (4.11) for expected cancellation of bad edges resulting from quartets of types q_3 (analogous to q_1) by good edges resulting from quartets of type q_4 (analogous to q_2):

$$(4.12) \alpha(1-\alpha)\gamma(1-\beta)n^2 \frac{24m}{n^2}\alpha^2(1-\gamma)^2 - o(m) \ .$$

We thus have that the expected cancellation is the sum of (4.11) and (4.12).

It now just remains to minimize (4.11) + (4.12) subject to $1/3 \leq \alpha \leq 1/2$ and $1/3 \leq \beta, \gamma \leq 2/3$, in order to get a lower bound on the expected negative edge cancellation. Clearly (4.11) is minimized when $\beta = 2/3$ and (4.12) is minimized when $\gamma = 2/3$ and $\beta = 2/3$. Their sum (up to the $o(m)$ term) now becomes only $m\frac{16}{27}(3\alpha(1-\alpha)^3 + \alpha^3(1-\alpha))$ whose minimum in $[1/3, 1/2]$ is attained at $\alpha = 1/2$, in which case it is $4m/27$. In any case, we see that when $m = \omega(n^2)$ we can replace $4m$ in Lemma 3.2 with $(4 - 4/27)m - o(m)$, and, together with the 3-cut method, this yields an improvement of the approximation algorithm to 0.472.

5 A randomized PTAS for dense instances

We present a randomized PTAS for MQC whenever the set of input quartets $\mathcal{Q}$ contains $\Theta(n^4)$ quartets. For every fixed $\epsilon > 0$, if the solution to MQC is $\mathcal{Q}_{opt}$, our algorithm will construct, with very high probability, a tree T so that $|Q_s(T)| \geq |\mathcal{Q}_{opt}|(1-\epsilon)$, in polynomial time.

Our proof is based upon a randomized reduction to *Complete MQC* (the case where $|\mathcal{Q}| = \binom{n}{4}$) for which a PTAS has been provided by Jiang at al. [10].

Theorem 5.1. *Let $\alpha > 0$ and $\epsilon' > 0$ be fixed. There is a polynomial time algorithm that, given an instance of $\mathcal{Q}$ of size at least $\alpha\binom{n}{4}$, constructs with very high probability a tree T such that $\mathcal{Q}_s(T) \geq (1-\epsilon')|\mathcal{Q}_{opt}|$.*

Proof. Given an input set $\mathcal{Q}$ of m quartets on the taxa set {1,. . . ,n}, we extend it into an instance of complete MQC as follows. For each 4-set $\{a,b,c,d\}$ of taxa that *does not* induce an element of $\mathcal{Q}$, we randomly construct a quartet on $\{a,b,c,d\}$ where each of the three choices is equally likely. We do this, independently, for all $\binom{n}{4} - m$ 4-sets. Let $\mathcal{Q}^*$ denote the set of $s = \binom{n}{4} - m$ randomly constructed quartets, and notice that $\mathcal{Q} \cup \mathcal{Q}^*$ is an instance of complete MQC.

Lemma 5.1. *With very high probability, there is no tree on labeled leaves $\{1,2,\ldots,n\}$ that satisfies more than $s(1/3+\epsilon/2)$ elements of $\mathcal{Q}^*$.*

Proof. Fix any tree T whose leaves are labeled with $\{1,\ldots,n\}$. The expectation of $|\mathcal{Q}^*_s(T)|$ is precisely $s/3$. We claim that, with very high probability, $|\mathcal{Q}^*_s(T)| < s/3 + s\epsilon/2$. Since each quartet in $\mathcal{Q}^*$ is chosen randomly and independently of all others, the random variable $|\mathcal{Q}^*_s(T)|$ is just a sum of s indicator random variables, each having probability 1/3. By a Chernoff large deviation inequality (see, e.g., [1], Appendix A),

$$\Pr[|\mathcal{Q}^*_s(T)| - s/3 > \epsilon s/2] < e^{-2(\epsilon s/2)^2/s} = e^{-\epsilon^2 s/2} \ .$$

It is well know (and easy) that the number of labeled trees with $2n-2$ vertices is at most $(2n)^{2n-4}$, and this is more than the number of possible trees T in our case. It follows from the union bound that the probability that some tree T has $|\mathcal{Q}^*_s(T)| \geq s/3 + s\epsilon/2$ is at most

$$(2n)^{2n-4}e^{-\epsilon^2 s/2} = o(1)$$

where we have used the fact that $s = \Theta(n^4)$ (in fact, the last claim works even if $s = \omega(n \log n)$). ■

LEMMA 5.2. *With very high probability, there exists a tree T for which $|(\mathcal{Q} \cup Q^*)_s(T)| > |Q_{opt}| + s(1/3 - \epsilon/2)$.*

Proof. Consider any tree T that is optimal for $\mathcal{Q}$. For such a tree we have $|\mathcal{Q}_s(T)| = |Q_{opt}|$. As in the previous lemma, the expectation of $|\mathcal{Q}^*_s(T)|$ is $s/3$. Again, using a Chernoff large deviation inequality (this time, however, we need to bound the deviation from the expectation from below):

$$\begin{aligned} & \Pr[|\mathcal{Q}^*_s(T)| - s/3 < -\epsilon s/2] \\ < \;& e^{-(\epsilon s/2)^2/(2s/3)} \\ = \;& e^{-(3/8)\epsilon^2 s} = o(1) \ . \end{aligned}$$

In particular, with probability $1 - o(1)$ for this tree we have $|(\mathcal{Q} \cup Q^*)_s(T)| > |Q_{opt}| + s(1/3 - \epsilon/2)$. ■

Completing the proof of Theorem 5.1: We construct a random $\mathcal{Q}^*$ as above, and feed $\mathcal{Q} \cup \mathcal{Q}^*$ an an input to complete MQC, for which the algorithm from [10] outputs, in polynomial time, a tree T_0 so that $|(\mathcal{Q} \cup \mathcal{Q}^*)_s(T_0)| \geq (1-\epsilon)|(\mathcal{Q} \cup \mathcal{Q}^*)_{opt}|$. By Lemma 5.1 we know that with very high probability, $|\mathcal{Q}^*_s(T_0)| \leq s/3 + \epsilon s/2$. By Lemma 5.2 we know that with very high probability, $|(\mathcal{Q} \cup \mathcal{Q}^*)_{opt}| > |Q_{opt}| + s(1/3 - \epsilon/2)$. It follows that, for ϵ sufficiently small (as a function of α and ϵ'), $|\mathcal{Q}_s(T_0)| \geq (1-\epsilon')|Q_{opt}|$, as required. ■

6 Concluding remarks

Although the MQC problem is open for almost twenty years, the only general result achieved so far is a PTAS for a complete input set of $\binom{n}{4}$ quartets. We extended this algorithm to handle arbitrary inputs of size $\Theta(n^4)$. We then introduced a new approximation algorithm for the MQC problem for arbitrary size inputs that originate from a uniform random distribution. The algorithm is based on solving a MaxCut problem in a certain weighted graph induced by the input quartets. In its most general form, the algorithm achieves an approximation ratio of 0.425 for consistent input. For mildly dense inputs, this ratio can be further improved to 0.468 using additional non-trivial techniques and structural properties of the induced graph.

An important observation is that the 3-cut method in Subsection 4.1 actually yields a PTAS for MQC for some tree topologies: it follows from that method that these are the tree topologies that have the property that there exists an internal vertex whose removal splits the tree into three parts where each part contains between a $1/13 + o(1)$ and a $6/13 - o(1)$ fraction of the leaves, and the same property holds recursively in each of the separated parts. For example, the complete binary tree in which all leaves are in the same level has this property.

The issue of random quartets deserves some treatment. Indeed in some settings it is possible to obtain all $\binom{n}{4}$ quartets and then employ the PTAS of Jiang *et al.* [9]. However, such an approach is very time costly (not to say prohibitive) even for not too large datasets (say 1000 taxa). The applicability of this approach has been demonstrated in [14] where a subset of the complete data set was used for the MaxCut approach and yielded better results than other reconstruction methods.

7 Acknowledgments

We thank Tandy Warnow for her very helpful comments and Benny Chor for insightful discussions.

References

[1] N. Alon and J. Spencer. *The Probabilistic Method.* John Wiley, 1992.

[2] Constantinos Daskalakis, Cameron Hill, Alexander Jaffe, Radu Mihaescu, Elchanan Mossel, and Satish Rao. Maximal accurate forests from distance matrices. In *RECOMB*, pages 281–295, 2006.

[3] Constantinos Daskalakis, Elchanan Mossel, and Sébastien Roch. Optimal phylogenetic reconstruction. In *STOC*, pages 159–168, 2006.

[4] P. Erdös, 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.

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

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

[7] M.X. Goemans and D.P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the Association for Computing Machinery*, 42(6):1115–1145, November 1995.

[8] I. Gronau, S. Moran, and S. Snir. Fast and reliable reconstruction of phylogenetic trees with very short branches. In *ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 379–388, 2008.

[9] T. Jiang, P. Kearney, and M. Li. Orchestrating quartets: approximation and data correction. In *IEEE Symp. Foundation of Computer Science (FOCS)*, pages 416–425, Palo Alto, California, November 1998.

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

[11] K. St. John, T. Warnow, B. M. E. Moret, and L. Vawter. Performance study of phylogenetic meth-

ods: (unweighted) quartet methods and neighbor-joining. In *Proceedings of the Sixth Annual ACM-SIAM Symposium on Discrete Algorithms*, 2001.

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

[13] S. Snir and S. Rao. Using max cut to enhance rooted trees consistency. *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*, 3(4):323–333, 2006. Preliminary version appeared in WABI 2005.

[14] S. Snir, T. Warnow, and S. Rao. Short quartet puzzling: A new quartet-based phylogeny reconstruction algorithm. *Journal of Computational Biology (JCB)*, 1(15):91–103, 2008.

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