TREE-QMC: Improving quartet graph construction for scalable and accurate species tree estimation from gene trees

Yunheng Han[1,2] and Erin K. Molloy[1,2,*]

[1] *Department of Computer Science, University of Maryland, College Park, College Park, 20742, USA*
[2] *Center for Bioinformatics and Computational Biology, University of Maryland, College Park, College Park, 20742, USA*

**Email: ekmolloy@umd.edu*

Abstract

Genome-scale data is increasingly available for species tree estimation, bringing attention to the fact that different regions of the genome can have evolutionary histories that differ from each other and from the species tree due to incomplete lineage sorting. This has led to the development of many new methods, some of which estimate the species tree from a collection of (estimated) gene trees. ASTRAL, the dominant method in this class, executes an exact algorithm for the Maximum Quartet Support Species Tree problem within a constrained solution space, constructed from the input gene trees. The other popular heuristics for this NP-hard problem, wQMC and wQFM, take a set of weighted quartets as input and construct the species tree in a divide-and-conquer fashion, for example via graph cuts. This approach has failed to compete with ASTRAL because it is less scalable when considering the time required to explicitly weight quartets based on the gene trees. Moreover, in prior studies, wQMC has been consistently less accurate than either wQFM or ASTRAL.

Here, we address that the poor accuracy of wQMC by showing that the quartet graph should be normalized to account for "artificial taxa," which are introduced during the divide phase so that solutions on subproblems can be combined during the conquer

phase. This led us to design efficient techniques for uniform and non-uniform graph normalization, with the goal of the latter being improved robustness on data sets with large numbers of taxa and gene tree heterogeneity. We provide an algorithm to construct the normalized quartet graph directly from the input gene trees, enabling our new method TREE-QMC to run in $O(n^3k)$ time where n is the number of species and k is the number of gene trees (provided some assumptions on subproblem sizes). In an extensive simulation study, graph normalization enabled TREE-QMC to be competitive in terms of species tree accuracy with ASTRAL-III and FASTRAL, a recent method that runs ASTRAL-III using an aggressively constrained solution space to speedup analyses. Notably, when the number of taxa was large (> 500), TREE-QMC (with non-uniform graph normalization) produced more accurate species trees than either FASTRAL and ASTRAL-III, at speeds closer to FASTRAL. We also re-analyzed an avian data set of 3,679 ultraconserved elements, finding that the estimated trees differed only on the short branches suggestive of ILS, with the TREE-QMC tree being closer to the concatenation tree than the ASTRAL-III/FASTRAL tree. Overall, our study shows that wQMC's divide-and-conquer framework can be competitive with the dynamic programming approach of ASTRAL.

Key words: species trees; coalescence; incomplete lineage sorting; quartets, supertrees; ASTRAL; wQMC

Introduction

Estimating the evolutionary history for a collection of species is a fundamental problem in evolutionary biology. There are two main approaches for species trees from multi-locus data sets, in which molecular sequences partitioned into orthologous,

recombination-free regions of the genome, referred to as loci or genes. The first approach, *concatenation*, involves concatenating the alignments (one for each locus) into a single alignment from which a phylogeny is estimated under a standard model of molecular evolution, such as the Generalized Time Reversible (GTR) model (Tavaré, 1986). The second approach, *gene tree summary*, involves estimating a phylogeny for each locus and then summarizing these gene trees into a species tree, with the goal of accounting for gene trees differing from each other and from the the species tree because of biological processes (Maddison, 1997). The popularity of this second class of methods has largely been driven by incomplete lineage sorting (also referred to as deep coalescence), which is expected to a major cause of gene tree discordance in placental mammals (McCormack et al., 2012), birds (Jarvis et al., 2014), and land plants (Wickett et al., 2014). Although concatenation does not account for gene tree discordance, both classes of methods have virtues as well as vices (Edwards, 2009; Springer and Gatesy, 2016; Molloy and Warnow, 2018) and are widely used.

The dominant gene tree summary method, ASTRAL (Mirarab et al., 2014; Mirarab and Warnow, 2015; Zhang et al., 2018), is based on the Maximum Quartet Support Species Tree (MQSST) problem, which can be cast as weighting quartets (i.e., four-leaf trees) by their frequencies in the gene trees and then seeking a species tree T that maximizes the total weight of the quartets displayed by T. The optimal solution to MQSST is a statistically consistent estimator of the (unrooted) species tree under a model of ILS (Mirarab et al., 2014), namely Multi-Species Coalescent (MSC) model (Pamilo and Nei, 1988; Rosenberg, 2002; Degnan and Salter, 2005). Unfortunately, MQSST is NP-hard (Lafond and Scornavacca, 2019), so ASTRAL executes an exact (dynamic programming) algorithm for MQSST within a constrained version of the solution space constructed from the input gene trees.

Importantly, ASTRAL operates directly on the input set of k gene trees instead of explicitly constructing a set of $\Omega(n^4)$ weighted quartets, where n is the number of species

(also referred to as taxa). This is in stark contrast to other popular heuristics for MQSST, namely Weighted Quartet Max Cut (wQMC; Avni et al., 2014) and Weighted Quartet Fiduccia-Mattheyses (wQFM; Mahbub et al., 2021), which take a set of weighted quartets as input and construct the species tree in a divide-and-conquer fashion, for example via graph cuts. When considering the time to weight quartets based on the gene trees, both wQFM and wQMC can be far more computationally intensive than ASTRAL (Mahbub et al., 2021). Additionally, wQMC was substantially less accurate than either wQFM or ASTRAL in a recent simulation study (Mahbub et al., 2021).

The comparatively poor performance of wQMC (or its unweighted version) has been observed in other prior simulation studies (Reaz et al., 2014; Davidson et al., 2015). Here, we resolve this issue showing that the quartet graph constructed by wQMC should be normalized to account for "artificial taxa," which are introduced so that solutions on subproblems can be combined within wQMC's divide-and-conquer framework. Overall, we offer two major methodological improvements:

1. We introduce two graph normalization techniques: one uniform and the other non-uniform, with the goal of improving robustness on data sets with large numbers of taxa and gene tree heterogeneity.

2. We provide an algorithm for constructing the normalized quartet graph directly from the input gene trees. This enables our new method TREE-QMC to run in $O(n^3k)$ time provided some assumptions on the subproblem sizes.

Together, these changes enable our new method, TREE-QMC, to be fast and accurate on data sets with large numbers of taxa in an extensive simulation study.

Specifically, we evaluated TREE-QMC in comparison to the other heuristics for MQSST, specifically the original wQMC method as well as wQFM, ASTRAL-III (Zhang et al., 2018), and FASTRAL (Dibaeinia et al., 2021), a new method that runs ASTRAL-III using an aggressively constrained solution space to speedup analyses. We found that methods requiring quartets to be explicitly weighted (i.e., wQMC and wQFM) were time

intensive on data sets with 100 taxa and 1000 genes and thus could be not run on larger data sets, unlike the other methods. In contrast, TREE-QMC achieved speeds closer to FASTRAL, even on data sets with 1000 taxa and 1000 (estimated) gene trees. Specifically, TREE-QMC and FASTRAL always ran in less than 2 hours, whereas ASTRAL-III ran in 5.3 hours on average and failed to finish running on three of these data sets within 18 hours.

We also evaluated TREE-QMC's graph normalization techniques, finding that they improved species tree accuracy in comparison to no normalization. When using graph normalization, TREE-QMC was competitive in terms of species tree accuracy with ASTRAL-III and FASTRAL. Moreover, when the number of taxa was large, the non-uniform normalization scheme enabled TREE-QMC to produce species trees that were more accurate than those produced by either FASTRAL and ASTRAL-III. Lastly, we compared methods on a biological data set of 3,679 ultraconserved elements from (Jarvis et al., 2014), finding that TREE-QMC returned a tree that was closer to the concatenation tree than the tree produced by ASTRAL/FASTRAL (although all three methods produce reasonable hypotheses based on quartet support and other analyses).

Overall, our study shows that a variety of heuristics for MQSST can lead to fast and accurate species tree estimation from gene trees. Looking beyond summary methods, SVDquartets (Chifman and Kubatko, 2014) depends on quartet amalgamation, leveraging wQMC. Graph normalization seems likely to be important in this context and could lead to improved accuracy.

Materials and Methods I. TREE-QMC

Terminology and Background

To present TREE-QMC, we begin with some terminology for phylogenetic trees. A *phylogenetic tree* T is a triplet $(g, \mathcal{L}, \phi)$, where g is a connected acyclic graph, $\mathcal{L}$ is a set of labels (typically denoting species), and ϕ maps leaves (i.e., vertices with degree 1) in g to

labels in $\mathcal{L}$. If ϕ is a bijection, we say that T is *singly-labeled*; otherwise, we say that T is *multi-labeled*. All trees are in this paper are assumed to be *binary*, meaning that all non-leaf, non-root vertices (referred to as *internal* vertices) have degree 3. For simplicity, we do not make an explicit distinction between a phylogenetic tree T and its underlying graph g, instead saying that T is a tree on species set $\mathcal{L}$ and denoting its edge set as $E(T)$, its internal vertex set as $V(T)$, and its leaf set as $L(T)$.

Trees may be either *unrooted* or *rooted*. Edges in an unrooted tree are undirected, whereas edges in a rooted tree are directed away from the root, a special vertex with in-degree 0 (all other vertices have in-degree 1). To transform an unrooted tree T into a rooted tree T_r, we select an edge in T, sub-divide it with a new vertex r (the root), and then orient the edges of T away from the root. Conversely, we transform a rooted tree T_r into an unrooted tree T by undirecting its edges and then suppressing any vertex with degree 2. Sometimes we consider a phylogenetic tree T *restricted* to a subset of its leaves $R \subseteq L(X)$. Such a tree, denoted $T|_R$, is created by deleting leaves in $L(T) \setminus R$ and suppressing any vertex with degree 2 (while updating branch lengths in the natural way).

We need two additional concepts, *bipartitions* and *quartets*, to present TREE-TMC. A bipartition splits a set $\mathcal{N}$ of labels into two disjoint sets: $\mathcal{E}$ and $\mathcal{F} = \mathcal{L} \setminus \mathcal{E}$. Each edge in a (singly-labeled, unrooted) tree T induces a bipartition because deleting an edge $e = (u, v)$ (but not its endpoints) creates two rooted subtrees whose leaf labels form the bipartition $\pi(e) = \mathcal{E}|\mathcal{F}$. A given bipartition is displayed by T if it is in the set $Bip(T) = \{\pi(e) : e \in E(T)\}$. We say this bipartition is trivial if $|\mathcal{E}| = 1$ or $|\mathcal{F}| = 1$; otherwise, we say that it is non-trivial. A quartet q is an unrooted, binary tree with four leaves a, b, c, d labeled by A, B, C, D, respectively. We use the notation $a, b|c, d$ to indicate that leaves a, b are siblings in q. It is easy to see that there are three possible topologies: $a, b|c, d$, $a, c|b, d$, and $a, d|b, c$. Note that we also refer to quartets by their labels, typically using lower case letters to denote leaves and capital letters to denote labels (this distinction is only important when trees are multi-labeled). A set of quartets can be

defined by a unrooted tree T by restricting T to every possible subset of four leaves in $L(T)$; the resulting set $Q(T)$ is referred to as the quartet encoding of T. If T is multi-labeled, then some of the quartets in $Q(T)$ will have multiple leaves labeled by the same label. Lastly, we say that T displays a quartet q if $q \in Q(T)$.

Maximum Quartet Support Species Tree Problem. We can now define the MQSST problem:

- **Input:** A set $\mathcal{L}$ of n species and a set $\mathcal{T}$ of k (singly-labeled, unrooted, binary) gene trees, each on species set $\mathcal{L}$
- **Output:** A (singly-labeled, unrooted, binary) species tree T^* on species set $\mathcal{L}$ that maximizes $\sum_{q \in Q(T^*)} w_{\mathcal{T}}(q)$, where $w_{\mathcal{T}}(q)$ is the number of gene trees in $\mathcal{T}$ that display quartet q

As previously mentioned, the optimal solution to MQSST is a statistically consistent estimator of the unrooted species tree under the MSC model (Mirarab et al., 2014). The intuition here is that the model species tree defines a probability distribution on the space of gene trees, and when there are four taxa, the most probable unrooted gene tree identifies the unrooted species tree.

The methods evaluated in this study heuristics for MQSST, which is NP-hard (Lafond and Scornavacca, 2019). Our new method TREE-QMC is based on wQMC (Avni et al., 2014), which we now review. To run wQMC as a heuristic for MQSST, there is an $\Omega(kn^4)$ preprocessing step, in which each quartet is weighted by the number of gene trees that display it. Then, the species tree is reconstructed via the divide-and-conquer framework originally proposed by Snir and Rao (2010, 2012) (also see Supplementary Figure S1). Initially, the input set of weighted quartets with leaves labeled by species set $\mathcal{L}$. At each step in the *divide* phase, we (1) take as input a set $\mathcal{Q}_{\mathcal{X}}$ of weighted quartets on subsets of $\mathcal{X}$, (2) seek a bipartition $\mathcal{E}|\mathcal{F}$ on $\mathcal{X}$, which will appear in the final tree, and (3) recurse on the implied subproblems by introducing artificial taxa: E for $\mathcal{E}$ and F for $\mathcal{F}$.

Quartet Graph and MaxCut. To produce a bipartition on $\mathcal{X}$, wQMC constructs a graph from $\mathcal{Q}_{\mathcal{X}}$, referred to as the *quartet graph*, and then seeks a maximum cut (Snir and Rao, 2010, 2012; Avni et al., 2014). The quartet graph is formed from two complete graphs, $\mathbb{B}$ and $\mathbb{G}$, both on vertex set V (i.e., there exists a bijection between V and $\mathcal{X}$). All edges in $\mathbb{B}$ and $\mathbb{G}$ are initialized to weight zero. Then, each quartet $q = A, B|C, D \in \mathcal{Q}_{\mathcal{X}}$ contributes its weight $w_{\mathcal{T}}(q)$ to two "bad" edges in $\mathbb{B}$ and four "good" edges in $\mathbb{G}$. The bad edges are based on sibling pairs: (A, B) and (C, D). The good edges are based on non-sibling pairs: (A, C), (A, D), (B, C), and (B, D). We do not want to cut bad edges because siblings should be on the same side of the bipartition; conversely, we want to cut good edges because non-siblings should be on different sides of the bipartition. Ultimately, we seek a cut $\mathcal{C}$ that maximizes $\sum_{(X,Y)\notin\mathcal{C}} \mathbb{B}[X, Y] - \alpha \sum_{(X,Y)\in\mathcal{C}} \mathbb{G}[X, Y]$, where $\alpha > 0$ is a hyperparameter that can be optimized using binary search. Although MaxCut is NP-complete (Karp, 1972), fast and accurate heuristics have been developed (Dunning et al., 2018).

Subproblems and Artificial Taxa. After wQMC finds a bipartition on $\mathcal{X}$ from input $\mathcal{Q}_{\mathcal{X}}$, it recurses on the implied subproblems (Snir and Rao, 2010, 2012; Avni et al., 2014). Given bipartition $\mathcal{E}|\mathcal{F}$, quartet $q = A, B|C, D \in \mathcal{Q}_{\mathcal{X}}$ is assigned to one of four cases: (1) *satisfied* if $A, B \in \mathcal{E}$ and $c, d \in \mathcal{F}$ (or vice versa, meaning that $A, B \in \mathcal{F}$ and $C, D \in \mathcal{E}$), (2) *violated* if $A, C \in \mathcal{E}$ and $B, D \in \mathcal{F}$ (or vice versa) or $A, D \in \mathcal{E}$ and $B, C \in \mathcal{F}$ (or vice versa), (3) *untouched* if $|\{A, B, C, D\} \cap \mathcal{E}| = 4$ or $|\{A, B, C, D\} \cap \mathcal{F}| = 4$, and (4) *deferred* if $|\{A, B, C, D\} \cap \mathcal{E}| = 3$ or $|\{a, b, c, d\} \cap \mathcal{F}| = 3$. All satisfied and violated quartets are discarded because they are satisfied or violated regardless of later steps in the algorithm. The remaining quartets are split into two sets (subproblems): $\mathcal{Q}_{\mathcal{E}+}$ and $\mathcal{Q}_{\mathcal{F}+}$. The set $\mathcal{Q}_{\mathcal{E}+}$ is formed by first adding the untouched and deferred quartets with 4 or 3 leaves labeled by element in $\mathcal{E}$, respectively. To add a deferred quartet, its one leaf labeled by a element in $\mathcal{F}$ is relabeled with artificial taxon F, so that the quartets in $\mathcal{Q}_{\mathcal{E}+}$ have leaves labeled by $\mathcal{E} \cup \{F\}$. While the untouched quartets have the same weights as in the previous

subproblem, the deferred quartets are updated so that for all unique labels $i, j, k \in \mathcal{E}$, the weight of quartet $i, j|k, F$ equals $\sum_{f \in \mathcal{F}} w_{\mathcal{T}}(i, j|k, f)$ and similarly for the two other quartets: $i, k|j, F$ and $i, F|j, k$. The set $\mathcal{Q}_{\mathcal{F}+}$ is formed in the same fashion by introducing artificial taxon E.

The recursion terminates when the input is on three or fewer taxa, as there is only one possible tree that can be returned. At each step in the *conquer phase*, we (1) take as input a pair of trees, one on $\mathcal{E} \cup \{F\}$ and the other on $\mathcal{F} \cup \{E\}$, (2) connect the trees at artificial taxa E and F, and (3) return the resulting tree on $\mathcal{E} \cup \mathcal{F}$. The final tree is on species set $\mathcal{L}$.

In the remainder of this section, we present TREE-QMC, which is like wQMC except that the quartet graph is constructed directly from the gene trees while being normalized with respect to artificial taxa. The former is important for scalability, and the latter is important for species tree accuracy.

Normalization of Quartet Graph

In the divide-and-conquer framework, we introduce artificial taxa and re-label species (i.e. leaves) accordingly. For simplicity, we use the initial species labels when referring to the leaves of T (as there is a bijection between them). Therefore, the weight of quartet $M, N|O, P$ is

$$f_0(M, N|O, P) = \sum_{m \in \mathbf{M}} \sum_{n \in \mathbf{N}} \sum_{o \in \mathbf{O}} \sum_{p \in \mathbf{P}} w_{\mathcal{T}}(m, n|o, p) \tag{0.1}$$

where $\mathbf{M} \subset \mathcal{L}$ denotes the set of leaves (i.e., species) in T associated with label M. We say that label M is a singleton if $|\mathbf{M}| = 1$; otherwise, M is an artificial taxon. When labels M, N, O, P are all singletons, each gene tree casts exactly one vote for one of the three possible quartets: $M, N|O, P$ or $M, O|N, P$ or $M, P|N, O$ (assuming no missing data). Otherwise, each gene tree casts $|\mathbf{M}| \cdot |\mathbf{N}| \cdot |\mathbf{O}| \cdot |\mathbf{P}|$ votes (again assuming no missing data) and thus can vote for more than one topology. We propose to normalize the quartet

weights so that each gene tree casts one vote for each subset of four labels, although it may split its vote across the possible quartet topologies in the case of artificial taxa (Fig. 1).

In the simplest case, we simply divide by the number of votes cast so the weight of quartet $M, N|O, P$ becomes

$$f_1(M, N|O, P) = \frac{f_0(M, N|O, P)}{|\mathbf{M}| \cdot |\mathbf{N}| \cdot |\mathbf{O}| \cdot |\mathbf{P}|} \tag{0.2}$$

This an be implemented efficiently by assigning an importance value $I(x)$ to each species $x \in \mathcal{S}$ and then compute the weight as

$$f(M, N|O, P) = \sum_{\substack{m \in \mathbf{M}, n \in \mathbf{N}, \\ o \in \mathbf{O}, p \in \mathbf{P}}} I(m, n, o, p) \cdot w_{\mathcal{T}}(m, n|o, p) \tag{0.3}$$

where $I(m, n, o, p) = I(m) \cdot I(n) \cdot I(o) \cdot I(p)$. Specifically, Equation 0.3 reduces to Equation 0.2 when $I(m) = |\mathbf{M}|^{-1}$ for all $m \in \mathbf{M}$ (and similarly for $\mathbf{N}, \mathbf{O}, \mathbf{P}$). Because all species with the same label are assigned the same importance value, we refer to this approach as *uniform normalization (n0)*. More broadly, the quartet weights will be normalized whenever Equation 0.3 corresponds to a weighted average, meaning that

$$\sum_{m \in \mathbf{M}} \sum_{n \in \mathbf{N}} \sum_{o \in \mathbf{O}} \sum_{p \in \mathbf{P}} I(m, n, o, p) = \sum_{\substack{m \in \mathbf{M}, n \in \mathbf{N}, \\ o \in \mathbf{O}, p \in \mathbf{P}}} I(m, n, o, p) = 1 \tag{0.4}$$

It is easy to see that this will be the case whenever $\sum_{m \in \mathbf{M}} I(m) = 1$ (and similarly for $\mathbf{N}, \mathbf{O}, \mathbf{P}$). Note that in *unnormalized (n0)* case, we assign all species an importance value of 1 so that Equation 0.3 reduces to Equation 0.1.

We now describe how to normalize quartet weights while leveraging the hierarchical structure implied by artificial taxa by assigning importance values to species with the same label. The idea is that species should have lesser importance each time they are *re-labeled* by an artificial taxon. Suppose that artificial taxon Z represents species $\mathbf{Z} = \{0, 6, 7, 9\}$ but species 0 and 9 were previously labeled by artificial taxon X (Fig. 1). This relationship can be represented as the rooted "phylogenetic" tree T_Z given by newick string:

$(6, 7, (0, 9)X)Z$. We use T_Z to assign importance values to all species $z \in \mathbf{Z}$, specifically

$$I(z) = \prod_{v \in path(T_Z, z)} \frac{1}{outdegree(v)} \tag{0.5}$$

where $outdegree(v)$ is the out-degree of vertex v and $path(T_Z, z)$ contains the vertices on the path in T_Z from the root to the leaf labeled z, excluding the leaf. Continuing the example, $I(6) = I(7) = \frac{1}{3}$ and $I(0) = I(9) = \frac{1}{3} \cdot \frac{1}{2} = \frac{1}{6}$. By construction, $\sum_{z \in \mathbf{Z}} I(z) = 1$ so this approach normalizes the quartet weights. Because different species with the same label can have different weights, we refer to this approach *non-uniform normalization (n2)*. In our experimental study, normalizing the quartet weights in this fashion can improve species tree accuracy, especially for data sets with large numbers of taxa.

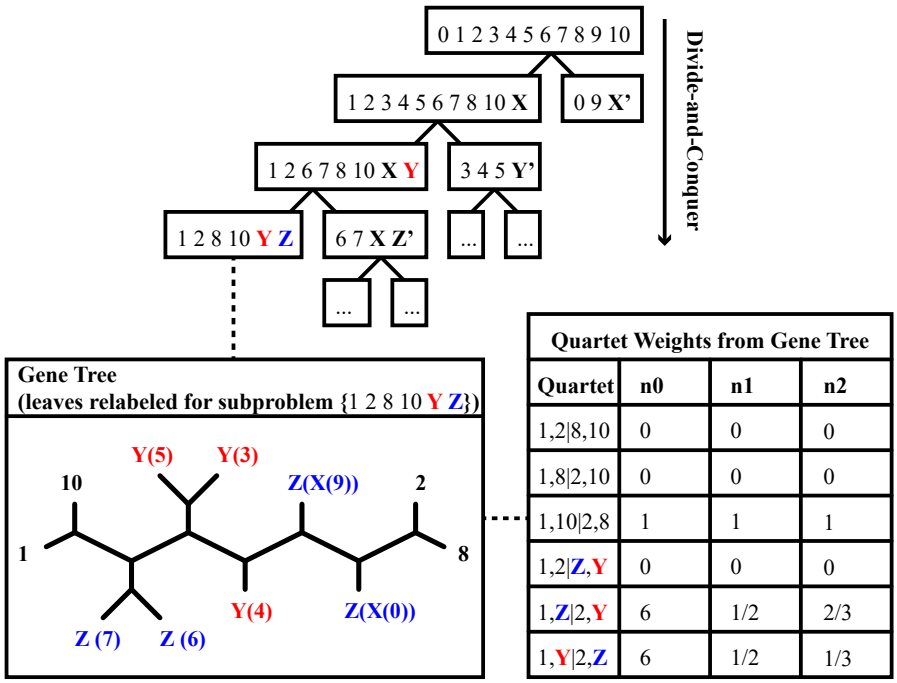
Efficient Construction of Normalized Quartet Graph

We now describe our approach for constructing the quartet graph directly from the input gene trees, which is implemented within our new method TREE-QMC. Recall that the quartet graph is built from two complete graphs $\mathbb{B}$ and $\mathbb{G}$ of bad edges and good edges, respectively. The total weight of bad edges between X and Y, denoted $\mathbb{B}[X, Y]$, is the number of quartets (displayed by the input gene trees) with X, Y as siblings (and similarly for $\mathbb{G}[X, Y]$ but non-siblings). Note that these quantities can be computed by summing over the number of bad and good edges contributed by each gene tree T. Henceforth, we consider how to compute $\mathbb{B}$ and $\mathbb{G}$ for a single gene tree.

We begin by considering a singly-labeled, binary gene tree T. In this case, we can compute the number of good edges between X, Y via

$$\mathbb{G}[X, Y] = \binom{n-2}{2} - \mathbb{B}[X, Y] \tag{0.6}$$

where n is the number of leaves in T. Because T is singly-labeled, there is exactly one leaf associated with label X, denoted x, and one leaf associated with label Y, denoted y. To compute $\mathbb{B}$ efficiently, we consider the unique path connecting leaves x and y in T (Fig. 2). Deleting the edges on this path (and their end points) produces a forest of K rooted



Quartet Weights from Gene Tree

Quartet	n0	n1	n2
1,2|8,10	0	0	0
1,8|2,10	0	0	0
1,10|2,8	1	1	1
1,2|Z,Y	0	0	0
1,Z|2,Y	6	1/2	2/3
1,Y|2,Z	6	1/2	1/3

Fig. 1. At each step in the divide-and-conquer pipeline, we seek a max cut of the quartet graph. We then recurse on the two subproblems defined by the cut (i.e., bipartition) after introducing artificial taxa. To compute quartet graph, we effectively re-label the leaves of each gene tree based on the taxa in the subproblem. In the example above, labels 1, 2, Y, and Z are associated with species sets $\mathbf{1} = \{1\}$, $\mathbf{2} = \{2\}$, $\mathbf{Y} = \{3, 4, 5\}$, and $\mathbf{Z} = \{0, 6, 7, 9\}$, respectively. After relabeling the leaves of the gene tree, we see that it casts $|\mathbf{1}| \cdot |\mathbf{Y}| \cdot |\mathbf{2}| \cdot |\mathbf{Z}| = 1 \cdot 3 \cdot 1 \cdot 4 = 12$ votes in total. Quartet $1, 2|Y, Z$ gets 0 votes (column n0). Quartet $1, Y|2, Z$ gets 6 votes (column n0) by taking either species 5, 3, or 4 for label Y and either species 0 or 9 for label Z. Quartet $1, Z|2, Y$ also gets 6 votes (column n0) by taking either species 5, 3, or 4 for label Y and taking either species 1 or 10 for label Z. We say that the votes are normalized (columns n1 and n2) when the gene tree votes once for each subset of four labels, although the vote can be split in the case of artificial taxa. In the uniform normalization scheme (column n1), we simply divide by the number of votes cast in the unnormalized case. The non-uniform normalization scheme (column n2) is based on the observation that some of the species now labeled by artificial taxon Z were previously labeled by artificial taxon X. The idea is that species should have lesser importance each time they are re-labeled by artificial taxa; see the main text for details.

subtrees, denoted $\{t_1, t_2, \ldots, t_K\}$. Let w and z be two leaves of subtrees t_i and t_j, respectively. Then, T displays quartet $x, w|z, y$ for $i < j$, quartet $z, y|w, z$ for $i = j$, and quartet $x, z|w, y$ for $i > j$. To summarize, x, y are siblings if and only if leaves w, z are in the same subtree off the path from x to y. It follows that $\mathbb{B}[X, Y]$ can be computed by considering all ways of selecting two other leaves from the same subtree for all subtrees on the path from x to y.

This observation can be used to count the quartets efficiently when gene trees are singly-labeled. However, we need to be more careful when T is multi-labeled, which is

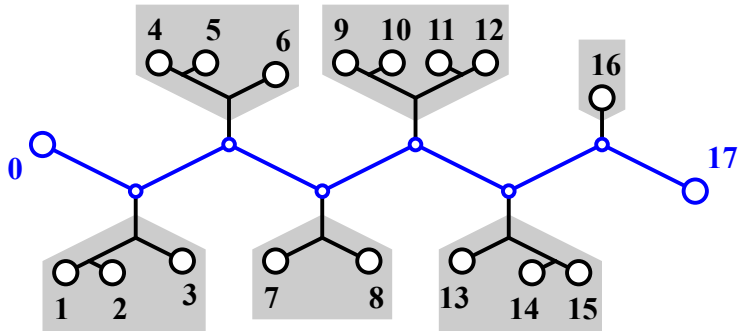


Fig. 2. To count the number of quartets induced by T with 0 and 17 as siblings, we consider the path between them (shown in blue). The deletion of these edges and their endpoints produces six rooted subtrees (highlighted in grey). Because 1 and 17 are siblings in a quartet if and only if the other two taxa are drawn from the same subtree, the number of bad edges can be computed as $\mathbb{B}_T[0,17] = \binom{3}{2} + \binom{3}{2} + \binom{2}{2} + \binom{4}{2} + \binom{3}{2} + \binom{1}{2} = 16$.

typically the case due to the introduction of artificial taxa. Following our example, suppose that we want to count the number of bad edges between 0 and 17 contributed by the subtree with leaves 4, 5, and 6. However, if leaves 4 and 5 are both re-labeled by artificial taxon M, the quartet on $0, 17|4, 5$ corresponds to satisfied quartet $0, 17|M, M$ should not be counted (as we do not want to count quartets that are already satisfied or violated). The other quartets $0, 17|4, 6$ and $0, 17|5, 6$ correspond to deferred $0, 17|M, 6$ and should be counted. Thus, we may wish to normalize the quartet graph, as described in the previous section.

In this paper, we present an algorithm for computing $\mathbb{B}$ in $O(s^2 n)$ time, where n is the number of leaves in gene tree T, and s is the number of labels in the subproblem (henceforth we let a denote the number of singletons and b denote the number of artificial taxa so the subproblem size is $s = a + b$). Our approach breaks down the calculation into three cases:

- Case 1: X, Y are both singletons
- Case 2: X is a singleton and Y is an artificial taxon (or vice versa)
- Case 3: X, Y are both artificial taxa

We present algorithms and proofs of correctness for case 1 in the Appendix; the other cases

are presented in the Supplementary Materials. To summarize the results, $B[X, Y]$ can be computed for all pairs X, Y in case 1, case 2, and case 3 in $O(a^2)$ time, $O(abn)$ and $O(b^2n)$ time, respectively. Thus, we can construct the quartet graph from k gene trees in $O(s^2nk)$ time. Afterwards, we seek a max cut using an $O(s^3)$ heuristic implemented in the open source library MQLib (Dunning et al., 2018). This gives us the final runtime of $O(s^2nk + s^3)$ for each subproblem. If the division into subproblems is perfectly balanced, the divide-and-conquer algorithm runs in $O(n^3k)$ time (Theorem 1 in the Supplementary Materials). Although we do not expect perfectly balanced subproblems in practice, we found method TREE-QMC to be fast in our experimental study.

Materials and Methods II. Experimental Study

We now give an overview of our simulation study to benchmark TREE-QMC against other heuristics for MQSST in the context of multi-locus species tree estimation.

Simulated data sets. We evaluated methods on a collection of data sets simulated by Mirarab and Warnow (2015). To summarize, 1000 gene trees were simulated down species trees under the MSC model using SimPhy (Mallo et al., 2015) and then sequences were simulated down each gene tree under the GTR model using INDELible (Fletcher and Yang, 2009). Lastly, gene trees were estimated from the molecular sequences using FastTree-2 (Price et al., 2010). It is worth noting that FastTree-2 represents identical sequences as a polytomy in the output gene tree.

This process was repeated with different parameters to create 11 model conditions, each of the 50 replicate data sets. Specifically, species tree height used to control the ILS level (low-mid, high, or very high) and the speciation rate was used control whether speciation events were concentrated shallower or deeper in the species tree, producing 6 different model conditions with 200 taxa. The number of taxa also was varied (10, 50, 100, 500, and 1000) while keeping the species tree height and speciation rate fixed (mid-low ILS and shallow speciation). Summary statistics are reported for each model condition to

quantify the level of ILS and GTEE (Table 1).

Table 1. Summary statistics are reported for each model condition. The statistic AD refers to the normalized Robinson-Foulds (RF) distance (Robinson and Foulds, 1981) between the true species tree and the true gene tree, averaged across all 1000 gene trees. The statistic GTEE refers to the normalized RF distance between the true gene and the estimated gene tree, averaged across all 1000 gene trees.

# of taxa	speciation	AD	GTEE
Low-mid ILS			
200	recent	0.21 ± 0.02	0.21 ± 0.13
200	deep	0.09 ± 0.01	0.28 ± 0.11
High ILS			
10	recent	0.17 ± 0.06	0.19 ± 0.09
50	recent	0.31 ± 0.04	0.26 ± 0.11
100	recent	0.33 ± 0.02	0.26 ± 0.09
200	recent	0.34 ± 0.02	0.27 ± 0.12
200	deep	0.34 ± 0.02	0.33 ± 0.12
500	recent	0.34 ± 0.01	0.28 ± 0.11
1000	recent	0.35 ± 0.01	0.32 ± 0.13
Very high ILS			
200	deep	0.69 ± 0.02	0.44 ± 0.12
200	recent	0.68 ± 0.02	0.44 ± 0.14

For each replicate data set, species trees were estimated from the first 250 estimated gene trees as well as the first 1000 estimated gene trees (this process was repeated using true gene trees for all methods except wQMC and wQFM). For analyses of estimated gene trees, we excluded replicate data sets whenever 500 or more estimated gene trees (out of 1000) had the majority of their branches unresolved, like Mirarab and Warnow, 2015.

Methods. We evaluated five methods (wQMC v1.3, wQFM v3.0, ASTRAL v5.5.7, FASTRAL, and TREE-QMC), all of which are heuristics for MQSST. The first two methods take weighted quartets as input and the last three methods take gene tree as input. We performed this preprocessing step using the script distributed on Github with wQFM, which calls an executable from Brodal et al. (2013); see `https://github.com/Mahim1997/wQFM-2020`. All methods given their inputs were run in default mode; see Supplementary Materials for software commands. We ran our new method TREE-QMC using the three ways of graph normalization: none (n0), uniform

(n1), and non-uniform (n2). The current version of TREE-QMC requires binary gene trees as input so we refined polytomies in the estimated gene trees arbitrarily; the same refinements were used in all runs of TREE-QMC to facilitate a fair comparison of normalization schemes.

Evaluation. All methods were compared in terms of species tree error, quartet score, and running time. Species tree error is the number of false negative (FN) branches, that is, branches in the true species tree that are missing from the estimated species tree (sometimes we report the FN error rate, dividing FN by the number of internal branches in the true species tree). Quartet score is the number of quartets in the input gene trees that are induced by the estimated species tree. The runtime is the amount of time that elapses between the beginning and end of the computation (i.e., the wall clock time), as all methods evaluated in this study are single threaded. Computational experiments were performed on a cluster at the University of Maryland, College Park; see Supplementary Materials for details. It is worth noting that the runtime reported for wQFM and wQMC includes the time to weight quartets based on gene trees (the fraction of time spent during this preprocessing phase is also reported). Branch support estimation was turned off when running ASTRAL-III so that this calculation would not be included in it's runtime. We then used ASTRAL-III to compute the quartet score and branch support (Sayyari and Mirarab, 2016) for estimated species trees as well as the true (i.e., model) species tree.

Avian UCE data set. We also re-analyzed the avian data set from Jarvis et al. (2014) with 3,679 UCEs using the MQSST heuristics (note: data set includes the best maximum likelihood tree and the set of 100 bootstrapped trees for each UCE). Although the true species tree is unknown, we discuss the presence and absence of strongly corroborated clades, such as Passerea and six of the magnificent seven clades (excluding clade IV) (Braun and Kimball, 2021) . We also compare methods to the "concatenation tree" estimated by running RAxML (Stamatakis, 2014) on UCEs only (Jarvis et al., 2014); thus

the comparison between concatenation and the MQSST heuristics is on the same data set. Branch support was computed for the estimated species trees using ASTRAL-III's local posterior probability (Sayyari and Mirarab, 2016) as well as using multi-locus bootstrapping (MLBS; Seo, 2008).

Results

Overview

We present our results on simulated data sets in terms of three experiments. In our first experiment, we compare TREE-QMC (without graph normalization) to MQSST heuristics that leverage the same divide-and-conquer framework (i.e., wQMC and wQFM). In our second experiment, we evaluate the impact of graph normalization on TREE-QMC. In our third experiment, we benchmark the best versions of TREE-QMC against the leading MQSST heuristic, ASTRAL-III, as well as its recent improvement, FASTRAL. All figures and tables in the main paper show results for 1000 estimated gene trees. The figures and tables for 250 estimated gene trees and true gene trees (250 and 1000 genes) are provided in the Supplementary Materials. Lastly, we present a re-analysis of the Avian UCE data set.

Experiment 1: Comparison to other divide-and-conquer MQSST heuristics. Initially, our goal was to evaluate how TREE-QMC (without graph normalization) compared to wQMC and wQFM, as these methods leverage the same divide-and-conquer framework as TREE-QMC. We found that all three methods performed similarly with respect to species tree error (Fig. 3a, Supplementary Fig. S2a, Supplementary Table S1) but differed with respect to runtime (Fig. 3b, Supplementary Fig. S2b, Supplementary Table S2). Specifically, TREE-QMC was much faster than wQMC or wQFM, taking on average less than a minute for all model conditions with 100 or fewer taxa. In contrast, wQMC and wQFM ran on the order of seconds for 10 taxa, minutes for 50 taxa, and hours for 100 taxa

(therefore, we did run wQMC and wQFM on data sets with larger numbers of taxa and excluded them from prior experiments). Importantly, the runtime of both wQMC and wQFM was dominated by the time to construct the set of $\Omega(n^4)$ weighted quartets from the input gene trees (Supplementary Table S2). For data sets with 100 taxa, this preprocessing phase was nearly 100% of wQMC's running time and was 68% and 88% of wQFM's running time for data sets with 250 and 1000 estimated gene trees, respectively.

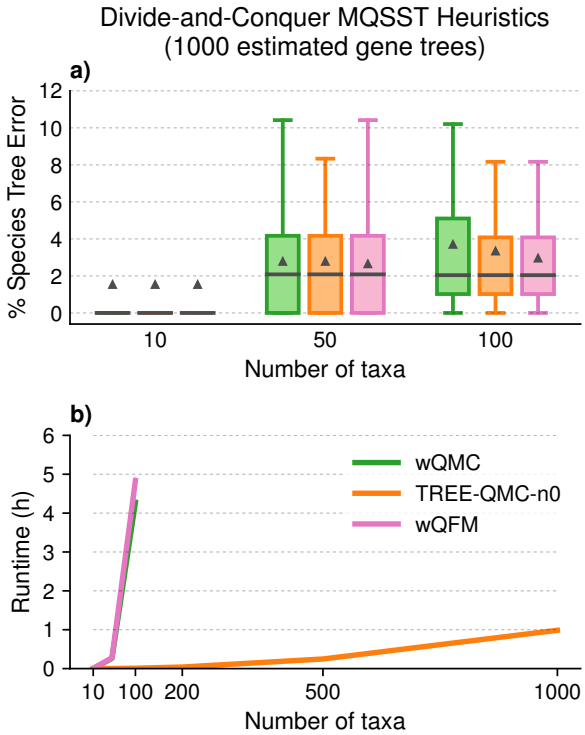


Fig. 3. Comparison of divide-and-conquer methods on data sets with 1000 estimated gene trees. Subplot (a) shows percent species tree error across replicates using box plots (bars represent medians, and triangles represent means; some outliers may not be shown due to the restriction of the y-axis). The order of box plots from left to right corresponds to methods: wQMC (green), TREE-QMC-n0 (orange), wQFM (pink). Subplot (b) shows mean runtime across replicates (standard error is too small to be seen). TREE-QMC-n0 is the only method shown that was run on data sets with up to 1000 taxa. The lines for wQMC and wQFM are on top of each other because their runtimes were dominated by the time to construct the set of $\Omega(n^4)$ weighted quartets from the input gene trees.

Experiment 2: Comparison of quartet graph normalization schemes. Our second goal was to evaluate the impact of quartet graph normalization on TREE-QMC, comparing no normalization (n0), uniform normalization (n1), and non-uniform normalization (n2).

Overall, we found that normalization improved species tree accuracy when the model condition was challenging, specifically larger numbers of taxa, higher levels of ILS, and deeper speciation (Fig. 4 and Supplementary Table S1). Moreover, non-uniform normalization was typically more effective than uniform normalization under these conditions. As an example, for the hardest model condition with 200 taxa (very high ILS and deep speciation), the mean (and median) species tree error was 9.2% (7.6%), 6.7% (5.0%) and 6.3% (4.6%) for the n0, n1, and n2 normalization schemes, respectively. In contrast, for the easiest model condition with 200 taxa (mid-low ILS and shallow speciation), the mean (and median) species tree error was 2.1% (1.0%), 2.0% (1.0%), and 1.8% (1.0%) for the n0, n1, and n2 normalization schemes, respectively. The results above are for data sets with 1000 estimated gene trees. Similar trends were observed for data sets with 250 gene trees (Supplementary Fig. S3) as well as true gene trees (Supplementary Figs. S4–S5 and Supplementary Tables S3); however, the benefit of normalization was greater for estimated (rather than true) gene trees.

We also considered the impact of normalization on quartet score, i.e., how well did TREE-QMC solved its optimization problem (Supplementary Tables S5 and S6). On data sets with 200 or more taxa, uniform normalization scheme achieved the highest median quartet score; however, the lowest median species tree error was achieved by non-uniform normalization. In other words, higher quartet score did not translate to higher species tree accuracy. This is not unexpected because the model species tree is only guaranteed to be the unique optimal solution to MQSST in the case of infinite error-free gene trees, which are not available in practice. Lastly, we note that normalizing the quartet graph did not impact runtime (Supplementary Table S2 and S4).

Experiment 3: Comparison to leading MQSST heuristics. Our third goal was to compare the best versions of TREE-QMC (i.e., with graph normalization) to the leading MQSST heuristics: ASTRAL-III and FASTRAL. We found that all of these methods produced highly accurate species trees when given 1000 estimated gene trees as input; specifically,

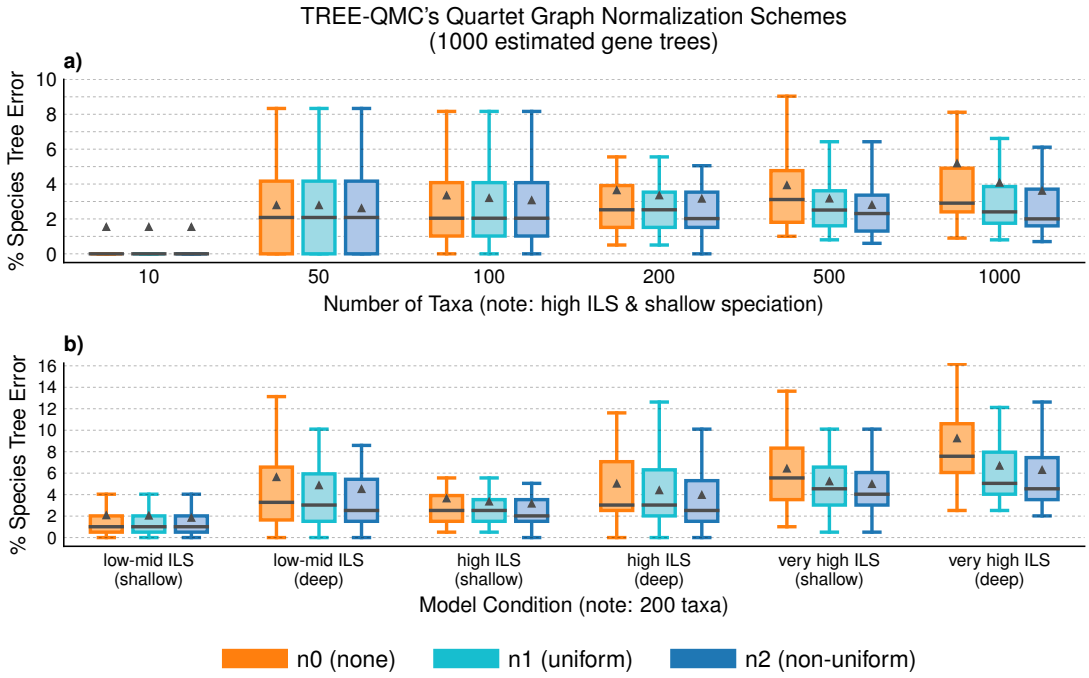


Fig. 4. Comparison of TREE-QMC's quartet graph normalization schemes. Both subplots show percent species tree error for replicates using box plots (bars represent medians; triangles represent means; outliers are not shown). Subplot (a) shows the impact of varying numbers of taxa (with high ILS and shallow speciation). Subplot (b) shows the impact of fixing the number of taxa at 200 and then varying the other model parameters; specifically species tree heights (which in turn varied the level of ILS) and speciation rates (which in turn varied whether speciation events tended to be shallower or deeper in the species tree). Note that one model condition (200 taxa, high ILS, shallow speciation) is repeated in both the subplot (a) and (b).

median species tree error was at or below 5% for all model conditions (Fig. 5a,c and Supplementary Table S1). Looking at the median error, FASTRAL was always at least as accurate as ASTRAL-III; similarly, TREE-QMC-n2 was always at least as accurate as TREE-QMC-n1. Therefore, we tested for significant differences between FASTRAL and TREE-QMC-n2 on data sets with estimated gene trees, finding that either TREE-QMC-n2 outperformed FASTRAL or else there was no significant differences between the two methods (Fig. 5a,c and Supplementary S5a,c). The model conditions for which there was a significant difference between methods were characterized by having 200 or more taxa (excluding the data sets with very high ILS). It is worth noting that FASTRAL typically had higher median quartet scores than TREE-QMC-n2, but, as previously observed, this does not necessarily translate to higher species tree accuracy (Supplementary Tables S5

and S6). Similar trends were observed for 250 estimated gene trees (Supplementary Fig. S7). All methods performed well on true gene trees (Supplementary Table S3 and Supplementary Figs. S8–S9) and had the same median species tree error across many model conditions.

Although TREE-QMC-n2 was more accurate than FASTRAL when the number of taxa was large, FASTRAL was faster than TREE-QMC-n2 in this case (Fig. 5b,d; Supplementary Table S2). For data sets with 1000 taxa and 1000 genes, FASTRAL TREE-QMC-n2, and ASTRAL-III completed on average in 32 minutes, 64 minutes, and 5.3 hours (it is also worth noting that ASTRAL-III failed to complete on 3/50 replicates within 18 hours). Thus, TREE-QMC-n2 was still much faster than ASTRAL-III.

Notably, runtimes for TREE-QMC-n2 and FASTRAL were not impacted by gene tree heterogeneity, unlike ASTRAL-III. For low-mid ILS model condition, ASTRAL-III took on average 5 minutes but this jumped to 73 minutes for the very high ILS (with 200 taxa, 1000 genes, and shallow speciation). Similarly, ASTRAL-III was much faster for true compared to estimated gene trees (Supplementary Fig. S8 and Table S4). In contrast, both TREE-QMC-n2 and FASTRAL were quite fast, taking on average less than 3 minutes for model conditions with 200 or fewer taxa. Similar trends were observed for data sets with 250 genes (Supplementary Figs. S6–S7).

Avian data set. Lastly, we present the re-analysis of the avian data set from Jarvis et al. (2014) with 3,679 UCEs (Figure 6. All methods, except for ASTRAL-III, ran in less than a minute; however, we needed to weight quartets for wQMC and wQFM, which took 41 minutes. ASTRAL-III completed in 65 minutes, making it the most time consuming method. We focused on the species tree produced by ASTRAL-III (as ASTRAL-III is widely used and FASTRAL produced the same species tree topology) as well as TREE-QMC-n2 (as this was the best method in our experimental study and wQFM produced the same species tree topology) We compare the ASTRAL and TREE-QMC trees to the concatenation tree for the UCE data set from (Jarvis et al., 2014). There were

many similarities between these three trees, which all contained the magnificent seven I. Core landbirds, II. Core waterbirds, III Phaethontimorphae (Sunbitten and Tropicbirds), IV Otidimorphae, V Caprimulgimorphae, VI Columbimorphae, VII Mirandornithes (Flamingos and Grebes) as well as Passerimorphae (Finches through Kea), Coraciimorphae (Bee-eaters through Mousebirds), Accipitrimorphae (Eagles and Vultures), Pelecanimorphae (Pelicans through Cormorants) Procellariimorphae (Fulmars and Penguins). The TREE-QMC-n2 and ASTRAL-III trees differed from the concatenation tree by 7 and 9 branches, respectively, putting the TREE-QMC-n2 tree slightly closer to the concatenation tree than the ASTRAL-III tree. Notably, the TREE-QMC-n2 tree recovers Passerea and Afroaves and fails to recover Columbea, like the concatenation tree and unlike the ASTRAL-III tree (note that Passerea was considered to be strongly corroborated, after accounting for data type effects, by Braun and Kimball, 2021).

There were only five branches that differed between the TREE-QMC-n2 tree and the ASTRAL-III tree. In the TREE-QMC-n2 tree, these branches are identified by ASTRAL (quartet) support values: 18 (34, 35, 31), 29 (34, 35, 31), 51 (34, 34, 32), 43 (34, 34, 32), 86 (35, 31, 24). In the ASTRAL-III, these branches are identified by ASTRAL (quartet) support values: 20 (34, 32, 34), 14 (35, 33, 32), 89 (35, 31, 34), 90 (36, 34, 30), and 23 (34, 33, 33). Thus, all of the branches for which TREE-QMC-n2 and ASTRAL-III differ are characterized by having nearly equal quartet support for the three resolutions, often with two of the resolutions having (nearly) equal quartet support. This suggests that alternate resolutions of these branches could be reasonably recovered by gene tree summary methods; thus, alternative analyses (and other methodological developments) may be required to resolve these recalcitrant branching orders.

Overall, TREE-QMC-n2 was very fast and was complementary to ASTRAL-III, in that it offered reasonable alternative hypothesis, which shared several notable features with the concatenation tree. The major challenge is that the differences between trees are not well supported, when looking at ASTRAL's branch support (local posterior

probability) However, it is difficult to have confidence in these alternative resolutions based on the quartet scores, which are derived from the estimated gene tree topologies.

Discussion and Conclusions

We have presented a new method TREE-QMC, which is based off the algorithmic framework of wQMC (Avni et al., 2014). Our algorithmic contributions were two-fold: first, we introduced the *normalized* quartet graph, and second we showed how to compute this graph directly from the gene trees. These developments enabled TREE-QMC to be highly competitive with the leading methods (ASTRAL-III and FASTRAL) in our simulation study. In particular, TREE-QMC (with non-uniform graph normalization) was the most accurate method on data sets with a large numbers of taxa (e.g. $\geqslant 500$) while being much faster than ASTRAL-III (and closer in speed to FASTRAL). This result is notable because the divide-and-conquer framework of TREE-QMC is quite different than the approach utilized by ASTRAL-III/FASTRAL (both solve MQSST exactly within a constrained search space defined from the input gene trees).

Both ASTRAL-III and FASTRAL are single-threaded, like TREE-QMC, and thus the comparison is fair. It would be interesting to compare TREE-QMC to the parallel method ASTRAL-MP (Yin et al., 2019), especially after parallelizing TREE-QMC. We leave this to future work but note that the divide-and-conquer framework implemented by TREE-QMC is amenable to parallelism, as subproblems can be processed in parallel. Notably, the time complexity of our method is best on the largest problems. The majority of species will be singletons in this case and thus the time complexity will be closer to $O(s^2)$ instead of $O(s^2n)$, where s is the subproblem size and n is the number of leaves in each gene tree. Finer grain parallelism can also be implemented as each gene trees contributions to the quartet graph can be computed in parallel.

Benchmarking methods on simulated data sets is an essential part of method development. However, these data sets do not necessarily capture or reflect the properties

of challenging biological data sets. To this end, we performed a re-analysis of the avian phylogenomics data sets from Jarvis et al. (2014). We found that the TREE-QMC tree recovered the clades Passerea and Afroaves, putting it closer to the concatenation tree (estimated on the same data set) than the tree estimated by ASTRAL-III. However, TREE-QMC only differed from the ASTRAL-III tree by a few branches, which were not highly supported in either tree, especially when looking at the quartet support values around the branch. It seems likely that other methodological developments (even beyond summary methods) will be necessary for resolving these recalcitrant nodes.

An example of an exciting development is the recently proposed weighted ASTRAL (Zhang and Mirarab, 2022) (this work was preprinted during our study). Their idea is weight quartets based on branch support or branch lengths in the estimated gene trees. This is more computationally intensive than the original version of ASTRAL, so Zhang and Mirarab (2022) also propose a divide-and-conquer framework (although it is different than the one leveraged by TREE-QMC). We can view the non-uniform version of TREE-QMC as weighting quartets except that the weights are based subproblem division, which ideally reflects the species tree topology (the idea is to down weight quartets that are least relevant to the subproblem). In the future, it would be interesting to compare TREE-QMC to weighted ASTRAL. Lastly, we note that TREE-QMC's algorithm already handles multi-labeled gene trees, and thus, it could be interesting to explore this approach on data sets with multi-labeled trees (Legried et al., 2021; Zhang et al., 2020; Yan et al., 2021; Smith et al., 2022).

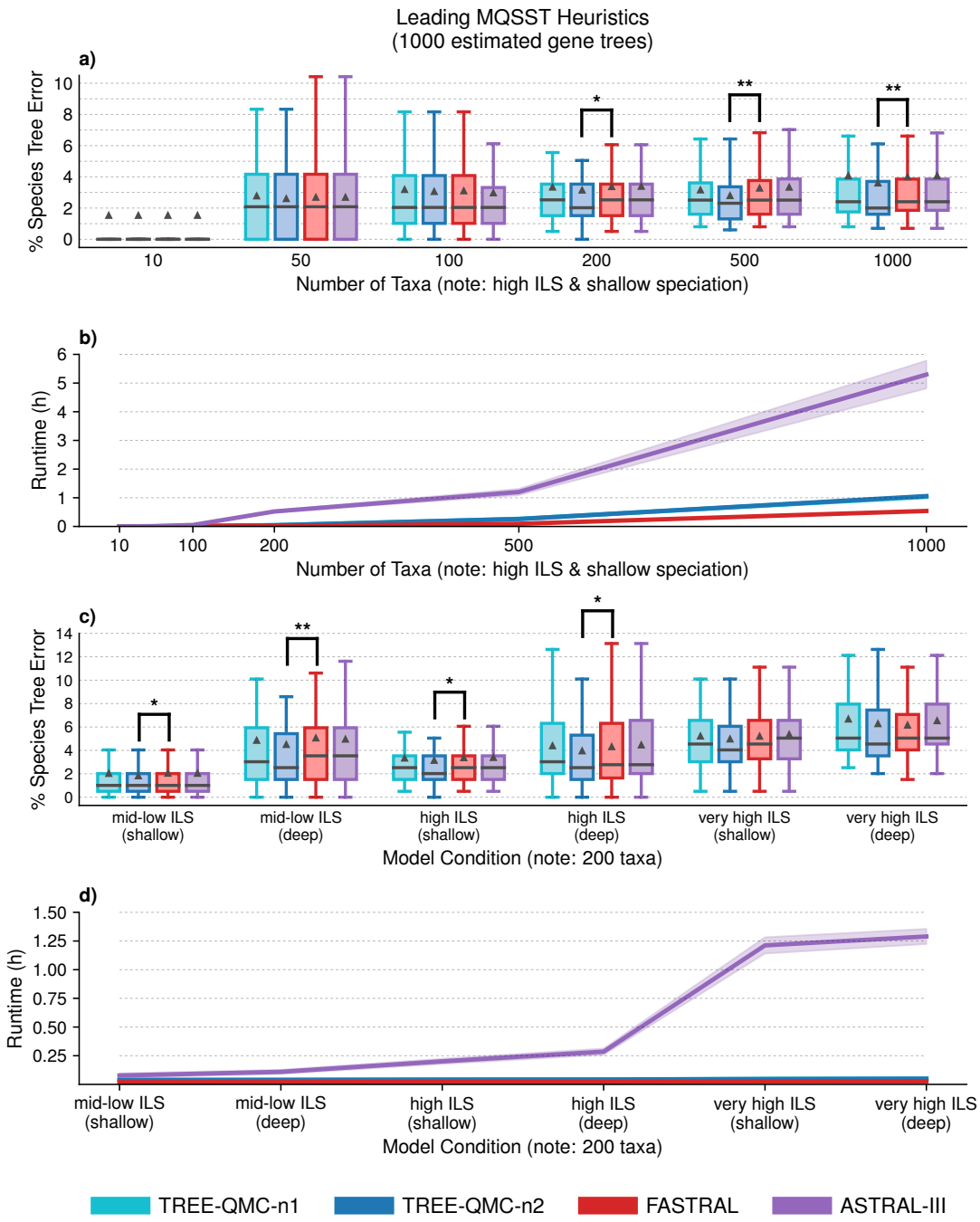


Fig. 5. Comparison of TREE-QMC to leading MQSST heuristics. Subplots (a) and (c) and show percent species tree error for replicates using box plots (bars represent medians; triangles represent means; some outliers are not shown). Two-sided Wilcoxon-signed ranked tests were used to evaluate differences between TREE-QMC-n2 and FASTRAL on estimated gene trees; we use * to indicate significance before Bonferroni correction ($\alpha < 0.05$) and ** to indicate significance after Bonferroni correction ($\alpha < 0.05/22 = 0.002$). Subplots (b) and (d) show mean runtime across replicates (shaded region indicates standard error). Subplots (a) and (b) show the impact of varying numbers of taxa (with high ILS and shallow speciation). Subplot (c) and (d) show the impact of fixing the number of taxa at 200 and then varying the other model parameters; specifically species tree heights (which in turn varied the level of ILS) and speciation rates (which in turn varied whether speciation events tended to be shallower or deeper in the species tree). Note that one model condition (200 taxa, high ILS, shallow speciation) is repeated in both the subplot (a) and (b).

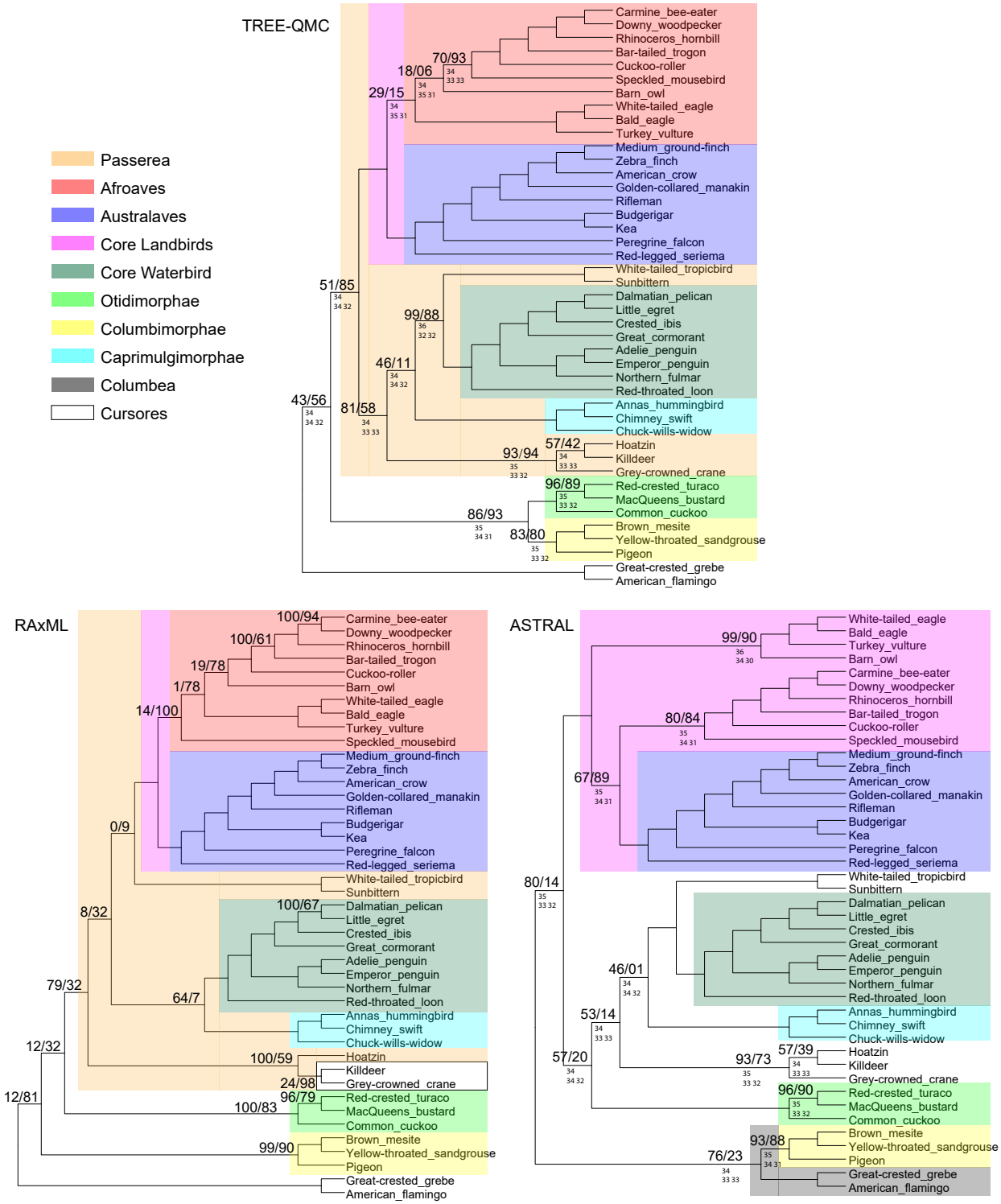


Fig. 6. Above are the trees returned by TREE-QMC-n2 (and wQFM) as well as ASTRAL-III (and FASTRAL). Above the branch, we show support values X/Y, where X is estimated using ASTRAL's local posterior probability (multiplied by 100) and Y is using MLBS. Below the branch, we show the quartet support (the two values below it correspond to quartet support for the two alternative resolutions of the branch). We also computed support values for the concatenation tree from Jarvis et al. (2014) using ASTRAL-III (but for this tree Y is the bootstrap support computed by RAxML). Support values are only shown when X is less than 100. Note that we do not show taxa outside of Neoaves as well methods recovered the same topology outside of Neoaves.

Appendix A: Efficient Construction of the Quartet Graph

Here, we present our algorithm for computing the bad edges $\mathbb{B}$ contributed by a single gene tree T. Recall that $\mathbb{B}[X, Y]$ is simply the number of quartets displayed by T in which X, Y are siblings. We first present an algorithm for singly-labeled gene trees and then describe how this algorithm can be modified (extensively) to accommodate multi-labeled gene trees.

Singly-Labeled Gene Trees

Let x, y to denote the leaf vertices currently labeled X, Y, respectively. Then, $\mathbb{B}[X, Y]$ can be computed by counting all ways to choose two leaves from each subtree off the path between x and y in T, as previously discussed (Fig. 2). After rooting T arbitrarily, we again consider the path between x and y, which now goes through their lowest common ancestor, denoted $lca(x, y)$ (Figure 7a). This allows us to break the computation into three parts

$$\mathbb{B}[X, Y] = \mathbb{A}[X, Y] + \mathbb{L}[X, Y] + \mathbb{R}[X, Y] \tag{0.7}$$

where $\mathbb{A}[X, Y]$ is the number of ways of selecting two leaves from the subtree above $lca(x, y)$, $\mathbb{L}[X, Y]$ the number of ways of selecting two leaves from the same subtree for all subtrees off the path from $lca(x, y)$ to leaf in it's *left* subtree (say x), and $\mathbb{R}[X, Y]$ the number of ways of selecting two leaves from the same subtree for all subtrees off the path from $lca(x, y)$ to the leaf in its *right* (say y). As we will show, each of these quantities can be computed in constant time, after an $O(n)$ preprocessing phase; thus, $\mathbb{B}$ can be computed in $O(n^2)$ time.

In the preprocessing phase, we compute two values for each vertex v in T. The first value $c[v]$ is the number taxa below vertex v. The second value $p[v]$, which we refer to as the "prefix" of v, is the number of ways to select two taxa from the same subtree for all subtrees off the path from the root to vertex v (Fig. 7b). It is easy to see that c can be

computed in $O(n)$ time via a post-order traversal. After which, p can be computed in $O(n)$ via a preorder traversal, setting

$$p[v] = p[v.parent] + \binom{c[v.sibling]}{2} \tag{0.8}$$

after initializing $p[root] = 0$. Putting this together gives us

$$\mathbb{A}[X, Y] = \binom{n - c[lca(x, y)]}{2} \tag{0.9}$$

$$\mathbb{L}[X, Y] = p[x] - p[lca(x, y).left] \tag{0.10}$$

$$\mathbb{R}[X, Y] = p[y] - p[lca(x, y).right] \tag{0.11}$$

where $v.left$ denotes the left child of v and $v.right$ denotes the right child of v (also see Figure 7b). Thus, we can compute $\mathbb{B}[X, Y]$ in constant time provided that we can access $lca(x, y)$ in constant time. This is possible via an $O(n)$ preprocessing step (Gusfield, 1997), although we implemented this implicitly by computing the entries of $\mathbb{B}$ during a post-order traversal of T. To summarize, we can compute $\mathbb{B}$ in $O(n^2)$ time, provided that T is singly-labeled.

We now wish to modify this approach for the case when T is a multi-labeled. This requires us to consider

- Case 1: X, Y are both singletons
- Case 2: X is a singleton and Y is an artificial taxon (or vice versa)
- Case 3: X, Y are both artificial taxa

Case 1 is handled in the next section, and cases 2 and 3 are handled in the Supplementary Materials.

Multi-Labeled Gene Tree Case 1: Two Singletons

In this section, we consider the computation of $\mathbb{B}[X, Y]$ when T is a multi-labeled gene tree and both X, Y are singletons (i.e., they are not artificial taxa). Again, we focus

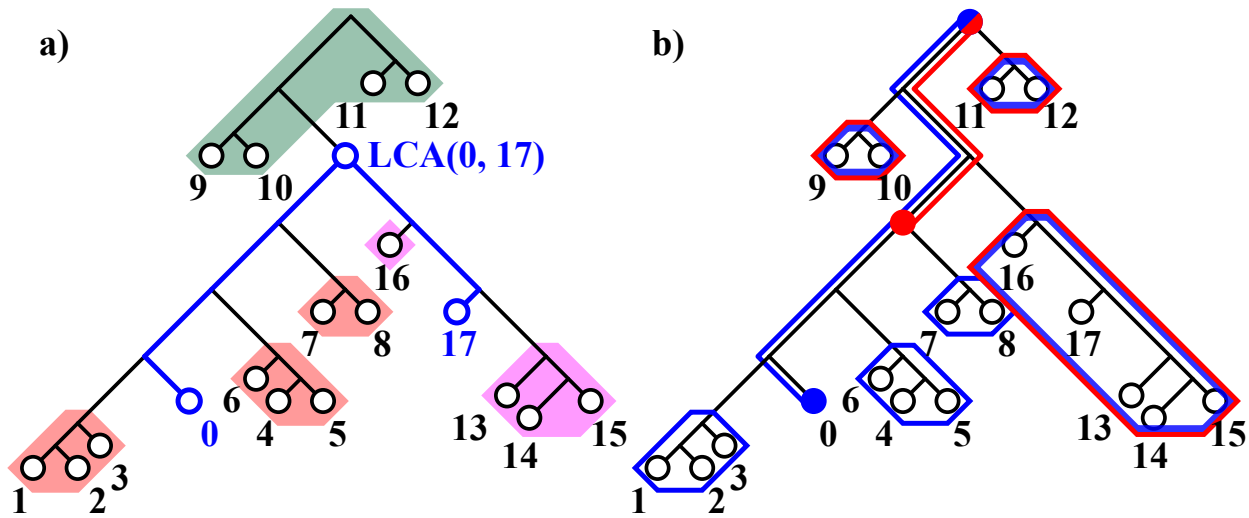


Fig. 7. **Rooting and prefix sum.** Here we show how to compute the number of quartets induced by T with 0 and 17 as siblings, denoted $\mathbb{B}[0, 17]$, after rooting T arbitrarily. Subfigure (a) shows that we need to consider the number of ways of selecting two taxa from the same subtree for three cases: (1) the subtree above the $lca(0, 17)$ (highlighted in green), (2) all subtrees off the path from the $lca(0, 17)$ to the left taxon 0 (highlighted in red), and (3) all subtrees off the path from the $lca(0, 17)$ to the right taxon 17 (highlighted in pink). Case 1 can be computed in constant time if we know the number of leaves below the LCA, that is, $\mathbb{A}[0, 17] = \binom{18-c[lca(0,17)]}{2} = 6$ (Equation 0.9). Cases 2 and 3 can also be computed in constant time as follows. Subfigure (b) shows the prefix of the left child of the $lca(0, 17)$, denoted $p[lca(0, 17).left]$ is the number of ways of selecting two taxa from the same subtree for all subtrees circled in red, which are off the path from the root to this vertex. Similarly, the the prefix of taxon 0, denoted $p[0]$, is the number of ways of selecting two taxa from the same subtree for all subtrees circled in blue, which are off the path from the root to 0. Therefore, the number of ways of selecting two taxa from all subtrees in case 2 (i.e., subtrees highlighted in red in subfigure a) is $\mathbb{L}[0, 17] = p[0] - p[lca(0, 17).left] = 19 - 12 = 7$ (Equation 0.10). Case 3 (not shown) can be computed as $\mathbb{R}[0, 17] = p[17] - p[lca(0, 17).right] = 41 - 38 = 3$ (Equation 0.11). Putting this all together gives $\mathbb{B}[x, y] = 6 + 7 + 3 = 16$ (Equation 0.7).

on the number of ways to select two leaves w, z from a collection of subtrees. When T is multi-labeled, it is possible for two leaves w, z to have the same label. Thus, we now need to count the number of ways to select two leaves z, w below vertex u such that $Z \neq W$, where we use capital letters W and Z to denote the current labels of leaves w and z, respectively.

We compute this quantity (referred to as the modified binomial) by modifying the preprocessing phase. Let $c_0[v]$ denote the number of leaves labeled by singletons below vertex v, and let $c_D[v]$ denote the number of leaves labeled by artificial taxon D below vertex v. Thus, for each vertex v, we store a vector $c[v]$ of length $b + 1$, where b is the number of artificial taxa in T. We can compute c in $O(bn)$ time via a postorder traversal. Then, the number of ways to select two leaves with different labels can (again) be broken into three cases.

- Case 1: the number of ways to select two singletons, which equals $\binom{c_0[v]}{2}$.
- Case 2: the number of ways to select one singleton and one artificial taxa, which equals $c_0[v] \cdot \sum_{D \in \mathcal{A}(v)} c_D[v]$, where $\mathcal{A}(v)$ is the set of artificial taxa below vertex v.
- Case 3: the number of ways to select two artificial taxa, which equals $\sum_{D \neq E \in \mathcal{A}(v)} c_D[v] \cdot c_E[v]$.

Putting this all together gives the modified binomial coefficient:

$$g_0[v] = \binom{c_0[v]}{2} + c_0[v] \cdot G_1[v] + \frac{G_1[v]^2 - G_2[v]}{2} \tag{0.12}$$

where $G_1[v] = \sum_{D \in \mathcal{A}(v)} c_D[v]$ and $G_2[v] = \sum_{D \in \mathcal{A}(v)} c_D[v]^2$. At each vertex, the calculation of $G_1[v]$ and $G_2[v]$ takes $O(b)$ time, after which we can compute $g_0[v]$ in constant time. Thus, g_0 can be computed in $O(bn)$ time. Note that we also need to compute modified binomial coefficient for the subtree "above" vertex v, denoted $g_0[v.above]$. This can be computed in a similar fashion by noting that number of singletons above v is $a - c_0[v]$ and that the number of leaves above v labeled by each artificial taxon D is $|\mathbf{D}| - c_D[v]$.

Using the modified binomial, we can apply the algorithm from the previous section by redefining the prefix of v

$$p_0[v] = p_0[v.parent] + g_0[v.sibling] \tag{0.13}$$

and then redefining the quantities:

$$\mathbb{A}[X, Y] = g_0[lca(x, y).above] \tag{0.14}$$

$$\mathbb{L}[X, Y] = p_0[x] - p_0[lca(x, y).left] \tag{0.15}$$

$$\mathbb{R}[X, Y] = p_0[y] - p_0[lca(x, y).right] \tag{0.16}$$

from which we can compute $B[x, y]$ in constant time. As there are a^2 pairs of singletons in the subproblem, the total runtime is $O(a^2 + bn)$.

The computation above is for the unnormalized quartet graph. To normalize the quartet graph, $\mathbb{B}[X, Y]$ becomes the *weighted* sum of quartets with X, Y are siblings, where

each quartet $x, y|z, w$ is weighted by $I(x, y, z, w) = I(x)I(y)I(z)I(w)$, where $I(x)$ is the importance value assigned to leaf x (which corresponds to a species in the singly-labeled gene tree). When X, Y are singletons,

$$\mathbb{B}[X, Y] = I(x)I(y) \sum_{\substack{w,z \in L(T): \\ Z \neq W \neq X \neq Y \\ q(x,y,z,y) = x,y|z,y}} I(z)I(w) \tag{0.17}$$

where the importance values of singletons are set to 1 so we know that $I(x) = I(y) = 1$. Note that all of the importance values are set to 1 in the unnormalized case.

To compute the normalized version of $\mathbb{B}[X, Y]$ using the previous algorithm, we set $c_D[v]$ to be the sum of the importance values of the leaves below v that are labeled by D (i.e., $c_D[v] = \sum_{m \in L(v), M=D} I(m)$ where $L(v)$ denotes the set of leaves below v). The proof of correctness follows from Lemma 0.1, in which we show that the total weight of selecting two uniquely labeled leaves below vertex u equals $g_0[u]$. Intuitively, this is because all other quantities $(p, \mathbb{A}, \mathbb{L}, \mathbb{R})$ are computed from $g_0[u]$ (see Supplementary Materials for details).

Lemma 0.1 The total weight of all taxon pairs in the subtree rooted at internal vertex u

$$\sum_{\substack{z,w \in L(u): \\ Z \neq W}} I(z)I(w) = g_0[u] \tag{0.18}$$

where $L(u)$ is the set of leaves below u.

Proof. Let $\mathcal{S}(u)$ be the set of singletons below u, and let $\mathcal{A}(u)$ be the set of artificial taxa

below u. Then,

$$\begin{aligned}
\sum_{\substack{z,w\in L(u):\\ Z\neq W}} I(z)I(w) &= \sum_{\substack{z,w\in \mathcal{L}(u):\\ Z,W\in\mathcal{S}(u),Z\neq W}} I(z)I(w) + \sum_{\substack{z,w\in L(u):\\ Z\in\mathcal{S}(u),W\in\mathcal{A}(u)}} I(z)I(w) + \sum_{\substack{z,w\in L(u):\\ Z,W\in\mathcal{A}(u),Z\neq W}} I(z)I(w) \\
&= \sum_{\substack{z,w\in L(u):\\ Z,W\in\mathcal{S}(u),Z\neq W}} 1 \\
&\quad + \left(\sum_{\substack{z\in L(u):\\ Z\in\mathcal{S}(u)}} I(z)\right)\cdot\left(\sum_{\substack{w\in L(u):\\ W\in\mathcal{A}(u)}} I(w)\right) \\
&\quad + \frac{\left(\sum_{Z\in\mathcal{A}(u)}\sum_{m\in L(u):M=Z} I(m)\right)^2 - \left(\sum_{Z\in\mathcal{A}(u)}(\sum_{m\in L(u):m=Z} I(m))^2\right)}{2} \\
&= \binom{c_0[u]}{2} + c_0[u]\cdot G_1[u] + \frac{G_1[u]^2 - G_2[u]}{2} \\
&= g_0[u]
\end{aligned}$$

where $G_1[u] = \sum_{D\in\mathcal{A}(u)} c_D[u]$, and $G_2[u] = \sum_{D\in\mathcal{A}(u)} c_D[u]^2$. □

Lastly, we need to compute the good edges $\mathbb{G}[X,Y]$, which is the total weight of quartets in which X,Y are not siblings. This can be done in constant time, following Lemma 0.2.

LEMMA 0.2 Let T be a multi-labeled gene tree T, and let X,Y be singletons. Then,

$$\mathbb{G}[X,Y] + \mathbb{B}[X,Y] = \binom{c_0[r]-2}{2} + (c_0[r]-2)\cdot G_1[r] + \frac{G_1[r]^2 - G_2[r]}{2} \tag{0.19}$$

where r is the root of T.

Proof.

$$\begin{aligned}
\mathbb{G}[X,Y] + \mathbb{B}[X,Y] &= \sum_{\substack{z,w \in L(r): \\ Z \neq W \neq X \neq Y}} I(x)I(y)I(z)I(w) \\
&= \sum_{\substack{z,w \in L(r): \\ Z \neq W \neq X \neq Y}} I(z)I(w) \\
&= \sum_{\substack{z,w \in L(u): \\ Z,W \in \mathcal{S}(u) \setminus \{X,Y\}, \\ Z \neq W}} I(z)I(w) + \sum_{\substack{z,w \in L(u): \\ Z \in \mathcal{S}(u) \setminus \{X,Y\}, \\ W \in \mathcal{A}(u)}} I(z)I(w) + \sum_{\substack{z,w \in L(u): \\ Z,W \in \mathcal{A}(u), \\ Z \neq W}} I(z)I(w) \\
&= \binom{c_0[r]-2}{2} + (c_0[r]-2) \cdot G_1[r] + \frac{G_1[r]^2 - G_2[r]}{2}
\end{aligned}$$

□

This concludes our treatment of case 1, in which X, Y are both singletons. In order to compute all entries of $\mathbb{B}$ and $\mathbb{G}$, we also need to consider the other two cases. In case 2, X is a singleton and Y is an artificial taxon (or vice versa), and in case 3, both X and Y are artificial taxa. These cases are more complicated because the naive approach would consider all paths in the tree between a leaf labeled X and a leaf labeled Y, which is not efficient. The algorithms and proofs of correctness for these cases are provided in the Supplementary Materials.

References

Avni, E., R. Cohen, and S. Snir. 2014. Weighted quartets phylogenetics. Systematic Biology 64:233–242.

Braun, E. and R. Kimball. 2021. Data Types and the Phylogeny of Neoaves. Birds 2:1–22.

Brodal, G. S., R. Fagerberg, T. Mailund, C. N. S. Pedersen, and A. Sand. 2013. Efficient algorithms for computing the triplet and quartet distance between trees of arbitrary degree. Page 1814–1832 *in* Proceedings of the Twenty-Fourth Annual ACM-SIAM Symposium on Discrete Algorithms SODA '13 Society for Industrial and Applied Mathematics, USA.

Chifman, J. and L. Kubatko. 2014. Quartet inference from SNP data under the coalescent model. Bioinformatics 30:3317–3324.

Davidson, R., P. Vachaspati, S. Mirarab, and T. Warnow. 2015. Phylogenomic species tree estimation in the presence of incomplete lineage sorting and horizontal gene transfer. BMC Genomics 16:S1.

Degnan, J. H. and L. A. Salter. 2005. Gene tree distributions under the coalescent process. Evolution 59:24–37.

Dibaeinia, P., S. Tabe-Bordbar, and T. Warnow. 2021. FASTRAL: improving scalability of phylogenomic analysis. Bioinformatics 37:2317–2324.

Dunning, I., S. Gupta, and J. Silberholz. 2018. What works best when? a systematic evaluation of heuristics for Max-Cut and QUBO. INFORMS Journal on Computing 30.

Edwards, S. V. 2009. Is a new and general theory of molecular systematics emerging? Evolution 63:1–19.

Fletcher, W. and Z. Yang. 2009. INDELible: A flexible simulator of biological sequence evolution. Molecular Biology and Evolution 26:1879–1888.

Gusfield, D. 1997. Algorithms on Strings, Trees, and Sequences: Computer Science and Computational Biology. Cambridge University Press, Cambridge, United Kingdom.

Jarvis, E. D., S. Mirarab, et al. 2014. Whole-genome analyses resolve early branches in the tree of life of modern birds. Science 346:1320–1331.

Karp, R. M. 1972. Reducibility among Combinatorial Problems Pages 85–103. Springer US, Boston, MA.

Lafond, M. and C. Scornavacca. 2019. On the weighted quartet consensus problem. Theoretical Computer Science 769:1–17.

Legried, B., E. K. Molloy, T. Warnow, and S. Roch. 2021. Polynomial-time statistical estimation of species trees under gene duplication and loss. Journal of Computational Biology 28:452–468.

Maddison, W. 1997. Gene trees in species trees. Systematic Biology 46:523–536.

Mahbub, M., Z. Wahab, R. Reaz, M. S. Rahman, and M. S. Bayzid. 2021. wQFM: highly accurate genome-scale species tree estimation from weighted quartets. Bioinformatics Btab428.

Mallo, D., L. De Oliveira Martins, and D. Posada. 2015. SimPhy : Phylogenomic simulation of gene, locus, and species trees. Systematic Biology 65:334–344.

McCormack, J. E., B. C. Faircloth, N. G. Crawford, P. A. Gowaty, R. T. Brumfield, and T. C. Glenn. 2012. Ultraconserved elements are novel phylogenomic markers that resolve placental mammal phylogeny when combined with species-tree analysis. Genome Research 22:746–754.

Mirarab, S., R. Reaz, M. S. Bayzid, T. Zimmermann, M. S. Swenson, and T. Warnow. 2014. ASTRAL: genome-scale coalescent-based species tree estimation. Bioinformatics 30:i541–i548.

Mirarab, S. and T. Warnow. 2015. ASTRAL-II: coalescent-based species tree estimation with many hundreds of taxa and thousands of genes. Bioinformatics 31:i44–i52.

Molloy, E. K. and T. Warnow. 2018. To include or not to include: The impact of gene filtering on species tree estimation methods. Systematic Biology 67:285–303.

Pamilo, P. and M. Nei. 1988. Relationships between gene trees and species trees. Molecular Biology and Evolution 5:568–583.

Price, M. N., P. S. Dehal, and A. P. Arkin. 2010. FastTree 2 – approximately maximum-likelihood trees for large alignments. PLOS ONE 5:1–10.

Reaz, R., M. S. Bayzid, and M. S. Rahman. 2014. Accurate phylogenetic tree reconstruction from quartets: A heuristic approach. PLOS ONE 9:1–13.

Robinson, D. and L. Foulds. 1981. Comparison of phylogenetic trees. Mathematical Biosciences 53:131–147.

Rosenberg, N. A. 2002. The probability of topological concordance of gene trees and species trees. Theoretical Population Biology 61:225–247.

Sayyari, E. and S. Mirarab. 2016. Fast coalescent-based computation of local branch support from quartet frequencies. Molecular Biology and Evolution 33:1654–1668.

Seo, T.-K. 2008. Calculating bootstrap probabilities of phylogeny using multilocus sequence data. Molecular Biology and Evolution 25:960–971.

Smith, M. L., D. Vanderpool, and M. W. Hahn. 2022. Using all gene families vastly expands data available for phylogenomic inference. Molecular Biology and Evolution 39.

Snir, S. and S. Rao. 2010. Quartets MaxCut: A divide and conquer quartets algorithm. IEEE/ACM Trans. Comput. Biol. Bioinformatics 7:704–718.

Snir, S. and S. Rao. 2012. Quartet MaxCut: A fast algorithm for amalgamating quartet trees. Molecular Phylogenetics and Evolution 62:1–8.

Springer, M. S. and J. Gatesy. 2016. The gene tree delusion. Molecular Phylogenetics and Evolution 94:1–33.

Stamatakis, A. 2014. RAxML version 8: a tool for phylogenetic analysis and post-analysis of large phylogenies. Bioinformatics 30:1312–1313.

Tavaré, S. 1986. Some probabilistic and statistical problems in the analysis of DNA sequences. Lectures on mathematics in the life sciences 17:57–86.

Wickett, N. J., , et al. 2014. Phylotranscriptomic analysis of the origin and early diversification of land plants. Proceedings of the National Academy of Sciences of the United States of America 111:E4859–E4868.

Yan, Z., M. L. Smith, P. Du, M. W. Hahn, and L. Nakhleh. 2021. Species tree inference methods intended to deal with incomplete lineage sorting are robust to the presence of paralogs. Systematic Biology 71:367–381.

Yin, J., C. Zhang, and S. Mirarab. 2019. ASTRAL-MP: scaling ASTRAL to very large datasets using randomization and parallelization. Bioinformatics 35:3961–3969.

Zhang, C. and S. Mirarab. 2022. Weighting by gene tree uncertainty improves accuracy of quartet-based species trees. bioRxiv .

Zhang, C., M. Rabiee, E. Sayyari, and S. Mirarab. 2018. ASTRAL-III: Polynomial time species tree reconstruction from partially resolved gene trees. BMC Bioinformatics 19:153.

Zhang, C., C. Scornavacca, E. K. Molloy, and S. Mirarab. 2020. ASTRAL-Pro: Quartet-based species-tree inference despite paralogy. Molecular Biology and Evolution 37:3292–3307.