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

Running Title: TREE-QMC

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

[1] *Department of Computer Science, University of Maryland, College Park, 20742, USA*

[2] *University of Maryland Institute for Advanced Computer Studies, College Park, MD 20740*

**Email: ekmolloy@umd.edu*

December 23, 2022

Abstract

Summary methods are one of the dominant approaches for estimating species trees from genome-scale data. However, they can fail to produce accurate species trees when the input gene trees are highly discordant due to gene tree estimation error as well as biological processes, like incomplete lineage sorting. Here, we introduce a new summary method TREE-QMC that offers improved accuracy and scalability under these challenging scenarios. TREE-QMC builds upon the algorithmic framework of QMC (Snir and Rao 2010) and its weighted version wQMC (Avni et al. 2014). Their approach takes weighted quartets (four-leaf trees) as input and builds a species tree in a divide-and-conquer fashion, at each step constructing a graph and seeking its max cut. We improve upon this methodology in two ways. First, we address scalability by providing an algorithm to construct the graph directly from the input gene trees. By skipping the quartet weighting step, TREE-QMC has a time complexity of $O(n^3k)$ with some assumptions on subproblem sizes, where n is the number of species and k is the number of gene trees. Second, we address accuracy by normalizing the quartet weights to account for "artificial taxa," which are introduced during the divide phase so that solutions on subproblems can be combined during the conquer phase. Together, these contributions enable TREE-QMC to outperform the leading methods (ASTRAL-III, FASTRAL, wQFM) in an extensive simulation study. We also present the application of these methods to an avian phylogenomics data set.

Introduction

Estimating the evolutionary history for a collection of species is a fundamental problem in evolutionary biology. Increasingly, species trees are estimated from multi-locus data sets,

with molecular sequences partitioned into (recombination-free) regions of the genome (referred to as loci or genes). A popular approach to species tree estimation involves concatenating the alignments for individual loci together and then estimating a phylogeny under some model of molecular sequence evolution, like the Generalized Time Reversible (GTR) model (Tavaré 1986).

Standard models assume the genes have a shared evolutionary history; however, this is not necessarily the case. The evolutionary histories of individual genes (referred to as gene trees) can differ from each other due to biological processes (Maddison 1997). Incomplete lineage sorting (ILS), one of the most well-studied sources of gene tree discordance, is an outcome of genes evolving within populations of individuals, as modeled by the multi-species coalescent (MSC) (Pamilo and Nei 1988; Rosenberg 2002; Degnan and Salter 2005). Concatenation-based approaches to species tree estimation can be statistically inconsistent under the MSC (Roch and Steel 2015). Moreover, simulation studies have shown concatenation can perform poorly when the amount of ILS is high (e.g. Kubatko and Degnan 2007). ILS is expected to impact many major groups, including birds (Jarvis et al. 2014), placental mammals (McCormack et al. 2012), and land plants (Wickett et al. 2014). Thus, species tree estimation methods that account for ILS, either explicitly or implicitly, are of interest.

An alternative to concatenation involves estimating gene trees (typically one per locus) and then applying a summary method. The most popular summary method to date, ASTRAL (Mirarab et al. 2014b), is a heuristic for the NP-hard Maximum Quartet Support Species Tree (MQSST) problem (Lafond and Scornavacca 2019), which can be framed as weighting quartets (four-leaf trees) by their frequencies in the input 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 the MSC model, which is why heuristics for this problem are widely used in the context of multi-locus species tree estimation. Proofs of consistency typically assume the input gene trees are error-free (Roch et al. 2018); however, this is rarely the case. Gene trees estimated in recent studies have had low bootstrap support on average (Table 1 in Molloy and Warnow 2018), suggesting that gene tree estimation error (GTEE) is pervasive

in modern phylogenomics data sets. GTEE has been shown to negatively impact the accuracy of summary methods in both simulation (e.g. Xi et al. 2015) and systematic studies (e.g. Meiklejohn et al. 2016). Together, GTEE and ILS present significant challenges to species tree estimation.

Scalability is also an issue when estimating species trees from large heterogeneous data sets. ASTRAL executes an exact (dynamic programming) algorithm for MQSST within a constrained version of the solution space constructed from the input gene trees. There have been many improvements to ASTRAL, with the latest version ASTRAL-III (Zhang et al. 2018) running in $O((nk)^{1.726}x)$ time, where n is the number of species (also called taxa), k is the number of gene trees, and $x = O(nk)$ is the size of the constrained solution space. In addition, a recent method FASTRAL (Dibaeinia et al. 2021) runs ASTRAL-III in an aggressively constrained solution space to speedup species tree estimation. Importantly, the ASTRAL operates directly on the input set of k gene trees instead of explicitly constructing a set of $\Theta(n^4)$ weighted quartets. This is in stark contrast to the other popular MQSST heuristics: weighted Quartet Max Cut (wQMC; Avni et al. 2014) and weighted Quartet Fiduccia-Mattheyses (wQFM; Mahbub et al. 2021).

The wQMC and wQFM methods take weighted quartets as input and thus require a preprocessing step, in which $\Theta(n^4)$ quartets are weighted by the number of gene trees that display them. Both implement divide-and-conquer approach to species tree estimation, which is quite different than the approach used by ASTRAL. Interestingly, a recent study found wQFM outperforms ASTRAL-III under model conditions characterized by high ILS and high GTEE (Mahbub et al. 2021); however, the scalability of wQFM is limited due to the required preprocessing. In this paper, we enable improved accuracy and scalability under these challenging scenarios by introducing TREE-QMC.

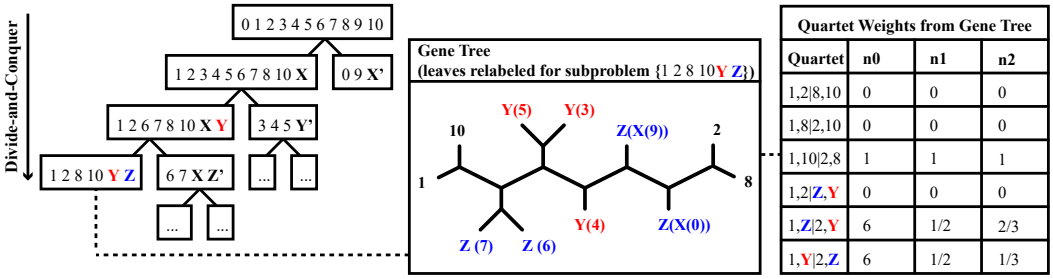
Results

Overview of TREE-QMC Method

TREE-QMC builds upon the first widely-used MQSST heuristic, wQMC, which reconstructs the species tree in a divide-and-conquer fashion. At each step in the divide phase, an internal branch in the output species tree is identified; this branch splits the taxa into two disjoint subsets (Figure 1). The algorithm continues by recursion on the subproblems implied by the two subsets of taxa. Importantly, "artificial taxa" are introduced to represent the species on the opposite of the branch so that solutions to subproblems can be combined during the conquer phase. The recursion terminates when the subproblem has three or fewer taxa, as there is only one possible tree that can be returned (Supplementary Figure S1). At each step in the conquer phase, trees for complementary subproblems are connected at their artificial taxa, until there is a single tree on the original set of species.

Central to wQMC's divide-and-conquer approach is a graph built from the (weighted) quartets. This graph is constructed in such a way that its max cut should correspond to a branch in the output species tree (Snir and Rao 2010, 2012; Avni et al. 2014). Our observation is that quartets with artificial taxa can have higher weights than quartets with only non-artificial taxa (called singletons) when looking at a single gene tree (Figure 1). As we will show, normalizing the quartet weights so that each gene tree gets one vote for every subset of four species greatly improves accuracy. The best performing normalization scheme (n2) weights quartets based on subproblem decomposition; specifically, quartets are upweighted if the species labeling their leaves are more closely related to the current subproblem (note: n1 denotes uniform normalization and n0 denotes no normalization). Moreover, we provide an algorithm to build the (normalized) quartet graph directly from the input gene trees, enabling TREE-QMC to have a time complexity of $O(n^3k)$ with some assumptions on subproblem sizes (see Methods section for details).

In the remainder of this section, we evaluate the performance of TREE-QMC (and its different normalization schemes) against the leading MQSST heuristics on simulated data. We then apply these methods to a real avian phylogenomics data set (Jarvis et al. 2014).



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

Figure 1: At each step in the divide phase, taxa are split into two disjoint subsets and then artificial taxa are introduced to represent the species on the other side of the split. To compute the quartet weights for a given subproblem, the leaves of each gene tree are relabeled by the artificial taxa. Without normalization (column n0), quartet $1, 2|Y, Z$ gets 0 votes and the alternative quartets get 6 votes each (note: quartet $1, Y|2, Z$ gets 6 votes by taking either species 5, 3, or 4 for label Y and either species 0 or 9 for label Z). With normalization, each gene tree gets one vote for each subset of four labels, although this vote can be split across the three possible quartets. In the uniform normalization scheme (column n1), we simply divide column n0 by the total number of votes cast in the unnormalized case. In the non-uniform normalization scheme (column n2), we leverage that structure implied by the divide phase of the algorithm; the idea is that species should have lesser importance each time they are re-labeled by artificial taxa.

Experimental Evaluation

We now give an overview of our simulation study; see Supplementary Materials for details.

Methods

TREE-QMC is compared against five leading MQSST heuristics: wQMC v1.3, wQFM v3.0, ASTRAL v5.5.7 (denoted ASTRAL-III or ASTRAL3), and FASTRAL. Two of these methods, wQMC and wQFM, which take weighted quartets instead of gene trees as input (the preprocessing step is performed using the script distributed on Github with wQFM). All methods are run in default mode. The current version of TREE-QMC requires binary gene trees as input so polytomies in the estimated gene trees are arbitrarily before running TREE-QMC (the same refinements are used in all runs of TREE-QMC to ensure a fair comparison across the normalization schemes).

Evaluation Metrics

All methods are compared in terms of species tree error, quartet score, and runtime. Species tree error is the percent Robinson-Foulds (RF) error (i.e., normalized RF distance between the true and estimated species trees multiple by 100). Because the true and estimated species tree are both binary, this quantity is equivalent to the percentage of false positive branches (i.e., internal branches in the estimated species tree that are incorrect and thus missing from the true species tree). Two-sided Wilcoxon signed-rank tests are used to evaluate differences between TREE-QMC-n2 versus FASTRAL as well as TREE-QMC-n2 versus ASTRAL3 (TREE-QMC-n2 is also compared against wQFM when possible). The quartet score is the number of quartets in the input gene trees that are displayed by the estimated species tree. All methods are run on the same data set on the same compute node, with a maximum wall clock time of 18 hours. The runtime of wQFM and wQMC includes the time to weight quartets based on the input gene trees (the fraction of time spent on this preprocessing phase is reported in the Supplementary Materials).

Simulated data sets

Our benchmarking study utilizes data simulated in prior studies, specifically the ASTRAL-II simulated data sets (Mirarab and Warnow 2015) as well as the avian and mammalian simulated data sets (Mirarab et al. 2014a). These data are generated by (1) taking a model species tree, (2) simulating gene trees within the species tree under the MSC, (3) simulating sequences down each gene tree under the GTR model, and (4) estimating a tree from set of gene sequences. Either the true gene trees from step 2 or the estimated gene trees from step 4 can be given as input to methods. This process is repeated for various parameter settings.

The avian and mammalian simulated data sets are generated from published species trees estimated for 48 birds (Jarvis et al. 2014) and 37 mammals (Song et al. 2012), respectively. The species tree branches are scaled to vary the amount of ILS, and the sequence length is changed to vary the amount of GTEE. There are 20 replicates for each model condition.

The ASTRAL-II data sets are generated from model species trees simulated under the

Yule model given three parameters: species tree height, speciation rate, and number of taxa. The speciation rate is set so that speciation events are clustered near the root (deep) or near the tips (shallow) of the species tree. There are 50 replicates for each model condition (note that a new model species tree is simulated each replicate data set)

The data properties (ILS and GTEE levels) are summarized in Supplementary Tables S1 and S2. The ILS level is the percent RF error (between the true species tree and the true gene tree) averaged across all gene trees, and GTEE level is the percent RF error (between the true and estimated gene trees) averaged across all gene trees. Overall, these data sets cover a range of important model conditions. The results are presented in four experiments looking at the impact of varying the number of taxa, the species tree scale/height (proxy for ILS), the sequence length (proxy for GTEE), and the number of genes.

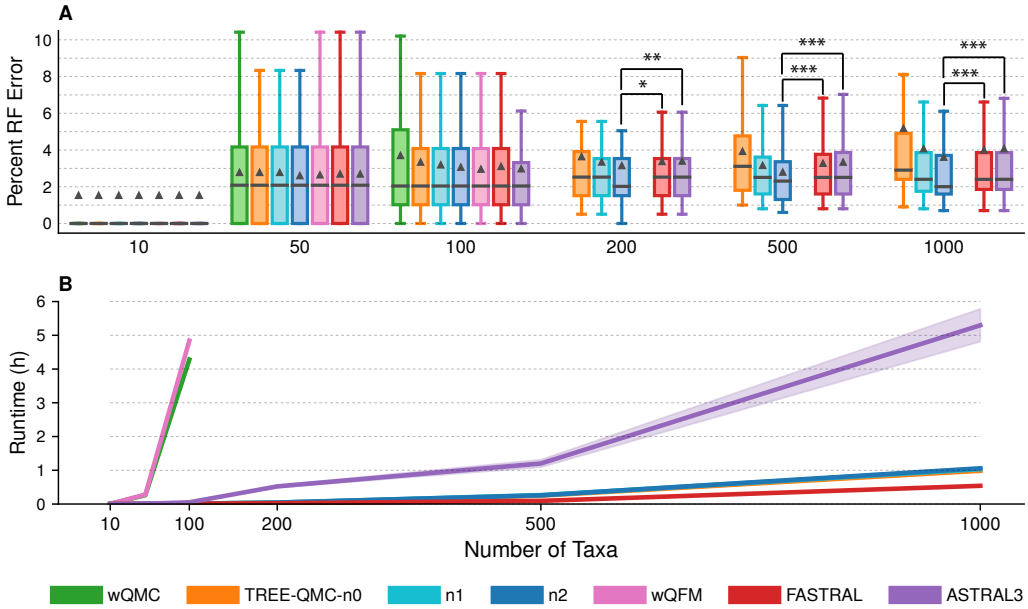


Figure 2: Impact of number of taxa. (A) Percent species tree error across replicates (bars represent medians; triangles represent means; outliers are not shown). The symbols *, **, and *** indicate significance at $p < 0.05$, 0.005, and 0.0005, respectively (all but * survive Bonferroni multiple comparison correction; see Supplementary Table S4 for details). (B) Mean runtime across replicates (shaded region indicates standard error). All data sets have species tree height 1X, shallow speciation, and 1000 estimated genes trees. The ILS level is 17–35% (ILS level), and GTEE level is 19–30%.

Number of Taxa

Figure 2 shows the impact of varying the number of taxa. The pipelines that need weighted quartets to be given as input (wQFM and wQMC) run on the order of seconds for 10 taxa, minutes for 50 taxa, and hours for 100 taxa, and did not complete within 18 hours (our maximum wallclock time) for the vast majority of data sets with 200 taxa. Importantly, the runtime of these pipelines is dominated by the time to weight $\Theta(n^4)$ quartets by their frequency in the input gene trees (Supplementary Table S3). In contrast, TREE-QMC implements the same approach as wQMC but bypasses this preprocessing step, scaling to 1000 taxa and 1000 genes. For these data sets, FASTRAL, TREE-QMC-n2, and ASTRAL-III complete on average in 32 minutes, 64 minutes, and 5.3 hours, respectively (although ASTRAL-III fails to complete on 3/50 replicates within 18 hours). Thus, TREE-QMC-n2 is much faster than ASTRAL-III and is not much slower than FASTRAL. More importantly, TREE-QMC-n2 is significantly more accurate than either FASTRAL or ASTRAL-III when the number of taxa is 200 or greater. For these same conditions, quartet weight normalization, and especially the non-uniform (n2) scheme, improves TREE-QMC's accuracy.

Incomplete Lineage Sorting (ILS)

ASTRAL-II data (200 taxa, 1000 estimated gene trees). Figure 3 shows the impact of varying the species tree height and thus the amount of ILS for the ASTRAL-II data sets. TREE-QMC-n2, FASTRAL, and ASTRAL-III produce highly accurate species trees, with median species tree error at or below 6% for all model conditions (note that wQMC and wQFM cannot be run on these 200-taxon data sets within the maximum wall clock time). For some conditions, TREE-QMC-n2 is significantly more accurate than FASTRAL or ASTRAL-III, and there is no significant difference between pairs of methods for the other conditions. Notably, quartet weight normalization improves the accuracy of TREE-QMC; this effect is most pronounced when the amount of ILS was very high (species tree height: 0.5X). On these same conditions, ASTRAL-III is much slower than the other methods, taking taking 73 minutes on average for the highest amount of ILS (species tree height: 0.5X) compared to 5 minutes on average for the lowest amount of ILS (species tree height:

5X). In contrast, both TREE-QMC-n2 and FASTRAL are quite fast, taking on average less than 3 minutes for model conditions with 200 or fewer taxa.

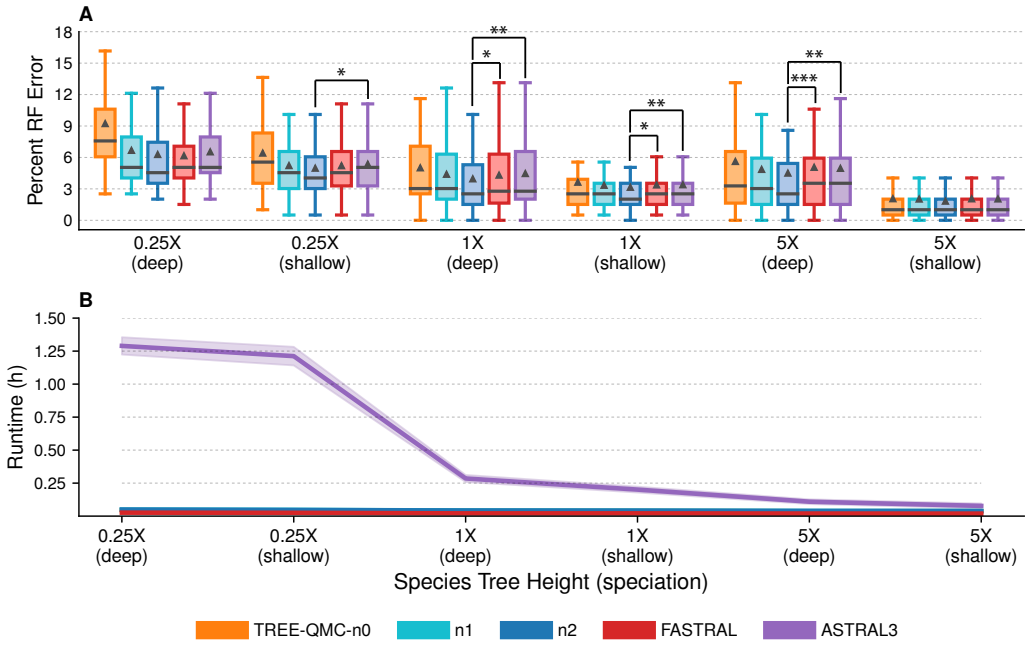


Figure 3: Impact of the amount of ILS on MQSST heuristics. (A) Percent species tree error across replicates (bars represent medians; triangles represent means; outliers are not shown). The symbols *, **, and *** indicate significance at $p < 0.05$, 0.005, and 0.0005, respectively (three tests survive multiple comparison corrections; see Supplementary Table S5 for details). (B) Mean runtime across replicates (shaded region indicates standard error). All data sets have 200 taxa and 1000 estimated gene trees. One model condition with species tree height 1X and shallow speciation is repeated from Figure 2. For species tree heights 0.5X, 1X, and 5X, the ILS level is 68–69% , 34%, and 9–21%, respectively, and the GTEE level is 44%, 27%–34%, and 21-28%, respectively.

Avian simulated data (48 taxa, 1000 estimated gene trees). Figure 4A–C shows the impact of varying the species tree scale and thus ILS on the avian simulated data sets. The original wQMC method is the least accurate method and is even less accurate than TREE-QMC-n0 (no normalization). Normalization improves the performance of TREE-QMC for these data, enabling TREE-QMC-n2 to be among the most accurate methods when the amount of ILS is higher (species tree scales: 0.5X and 1X). Testing for differences between TREE-QMC-n2 versus wQFM, FASTRAL, and ASTRAL-III reveals that either

TREE-QMC-n2 is significantly better or there are no significant differences between the pairs of methods. All methods finish quickly: wQMC and wQFM completes in less than 13 minutes on average, ASTRAL-III completes in less than 4 minutes on average, and the other methods finish in less than 1 minute on average.

Figure 4D–F shows the difference between the quartet score of the estimated species tree minus the quartet score of the true species tree (species trees were scored with the same gene trees used to estimate them). Higher quartet scores do not necessarily correspond to greater accuracy. For example, TREE-QMC-n0 is always less accurate than TREE-QMC-n2 but the former a higher quartet score for the lowest ILS level (Figure 4D) and a lower quartet score for the middle ILS level (Figure 4E). In general, the best performing methods find species trees with higher quartet scores than the true species tree when gene trees have high estimation error.

Mammalian simulated data (37 taxa, 200 estimated gene trees). All methods have similar performance for the mammalian data, although these data sets represent easier model conditions in terms of ILS and GTEE levels (Supplementary Figure S3, Supplementary Table S5).

Gene Tree Estimation Error (GTEE)

Avian simulated data (48 taxa, 1000 gene trees). Figure 4A–C also shows the impact of GTEE for each species tree scale (ILS level). Across all ILS levels, methods are either given true gene trees or estimated gene trees with substantial error (60-62%). Without GTEE, there is no significant differences between TREE-QMC-n2 versus the other leading methods (wQFM, FASTRAL, and ASTRAL-III), and all versions of TREE-QMC perform similarly so the utility of normalization is diminished. In addition, methods find species trees with similar quartet scores to the true species tree when given true gene trees as input. Lastly, the performance of wQMC is inline with the other methods (Figure 4C) when there is very little gene tree heterogeneity due to ILS or GTEE.

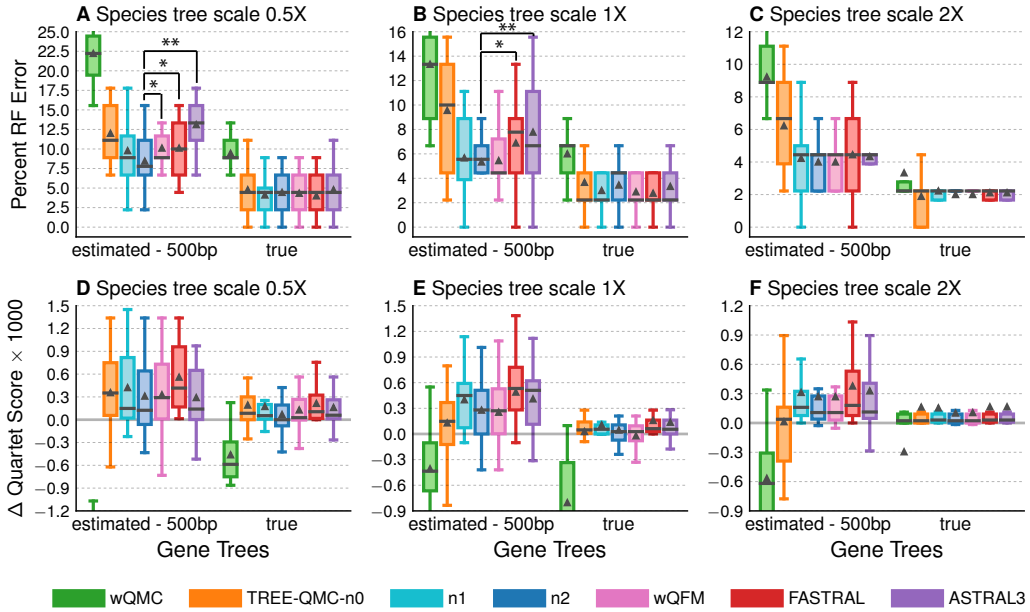


Figure 4: Impact of ILS and GTEE on MQSST heuristics. (A), (B, and (C) Percent species tree error for the avian data set with 1000 estimated or true gene trees and species tree scales 0.5X, 1X, and 2X, respectively. Two-sided Wilcoxon-signed ranked tests were used to evaluate differences between TREE-QMC-n2 versus wQFM, FASTRAL, and ASTRAL3 (9 tests per subfigure). The symbols *, **, and *** indicate significance at $p < 0.05$, 0.005, and 0.0005, respectively (for 0.5X species tree scale with estimated gene trees, the difference between TREE-QMC-n2 and ASTRAL-II survives Bonferroni multiple comparison correction; see Supplementary Table S6 for details). (D), (E), and (F) show the quartet score for the estimated species tree minus the quartet score for the true species tree times 1000 for species tree scales 0.5X, 1X, and 2X, respectively. For species tree heights 0.5X, 1X, and 2X, the ILS level is 60% , 47%, and 35%, respectively, and the GTEE level is 60%, 60%, and 62%, respectively. Results for wQMC are cut off because otherwise the trends cannot be observed (see Supplementary Figure S2 for full y-axes).

Mammalian simulated data (37 taxa, 200 gene trees). Similar trends between methods are observed for mammalian simulated data sets when varying the sequence lengths (Supplementary Figure S4). TREE-QMC is significantly more accurate than FASTRAL and ASTRAL-III for the shortest sequence length (250 bp; GTEE level 43%) and there are no differences between methods otherwise.

Number of Genes

Similar trends between methods are observed for varying the number of genes (Supplementary Figures S5). Overall, TREE-QMC-n2 is the best performing method, with error rates similar to wQFM (although, as shown in the first experiment, TREE-QMC-n2 scales to data sets with larger numbers of taxa).

Avian phylogenomics data set

We also re-analyze the avian data set from Jarvis et al. (2014) with 3,679 ultraconserved elements (UCEs). This 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 published 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 is 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). We repeat this analysis (except MLBS) on the TENT data (14,446 gene trees), which includes gene trees estimated on UCEs as well as exons and introns. In this case, methods are compared to the published TENT concatenation tree estimated by running ExaML (Kozlov et al. 2015).

UCE data

For the UCE data (48 taxa, 3679 gene trees), ASTRAL-III complete in 65 minutes, making it the most time consuming method. All other methods run in less than a minute; however, the preprocessing step to weight quartets for wQFM takes 41 minutes.

FASTRAL and ASTRAL-III produce the same species tree, and TREE-QMC-n2 and wQFM produce the species tree. We compare these two trees to the published concatenation tree for UCEs (Figure 5). There are many similarities between these three trees, as all

contain the magnificent seven clades. The TREE-QMC-n2 and FASTRAL trees differ from the concatenation tree by 7 and 9 branches, respectively, putting the TREE-QMC-n2 tree slightly closer to the concatenation tree than the FASTRAL 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). Overall, there are only five branches that differ between the TREE-QMC-n2 tree and the FASTRAL tree; all of these branches have nearly equal quartet support for their alternative resolutions so that both trees represent reasonable hypotheses.

TENT data

For the TENT data (48 taxa, 14446 gene trees), TREE-QMC-n2 and FASTRAL complete in less than 3 minutes, whereas it takes 2.35 hours to weight quartets. wQFM completes in less than a minute after this preprocessing phase. We do not run ASTRAL-III as this analysis was reported to take over 30 hours (Dibaeinia et al. 2021).

All three methods produce a different tree, which is compared to the published concatenation tree for TENT data (Supplementary Figure S6). None of the trees recover Passera, and only the concatenation and wQFM trees recover Afroaves, although this branch has very local support (local posterior probability of 0.0) in the wQFM tree. Once again, the TREE-QMC-n2 and wQFM trees are closest to the concatenation tree, with the TREE-QMC-n2, wQFM, and FASTRAL trees differing from it by 8, 8, and 10 branches, respectively. There are 5 branches that differ between the wQFM tree and the TREE-QMC-n2 tree (notably two of these branches in the wQFM have very low support: local posterior probability of 0.03 and 0.0). There are only 3 branches that differ between the TREE-QMC-n2 tree and the FASTRAL tree; as with the UCE data, these branches are reasonable based on quartet support for their alternative resolutions.

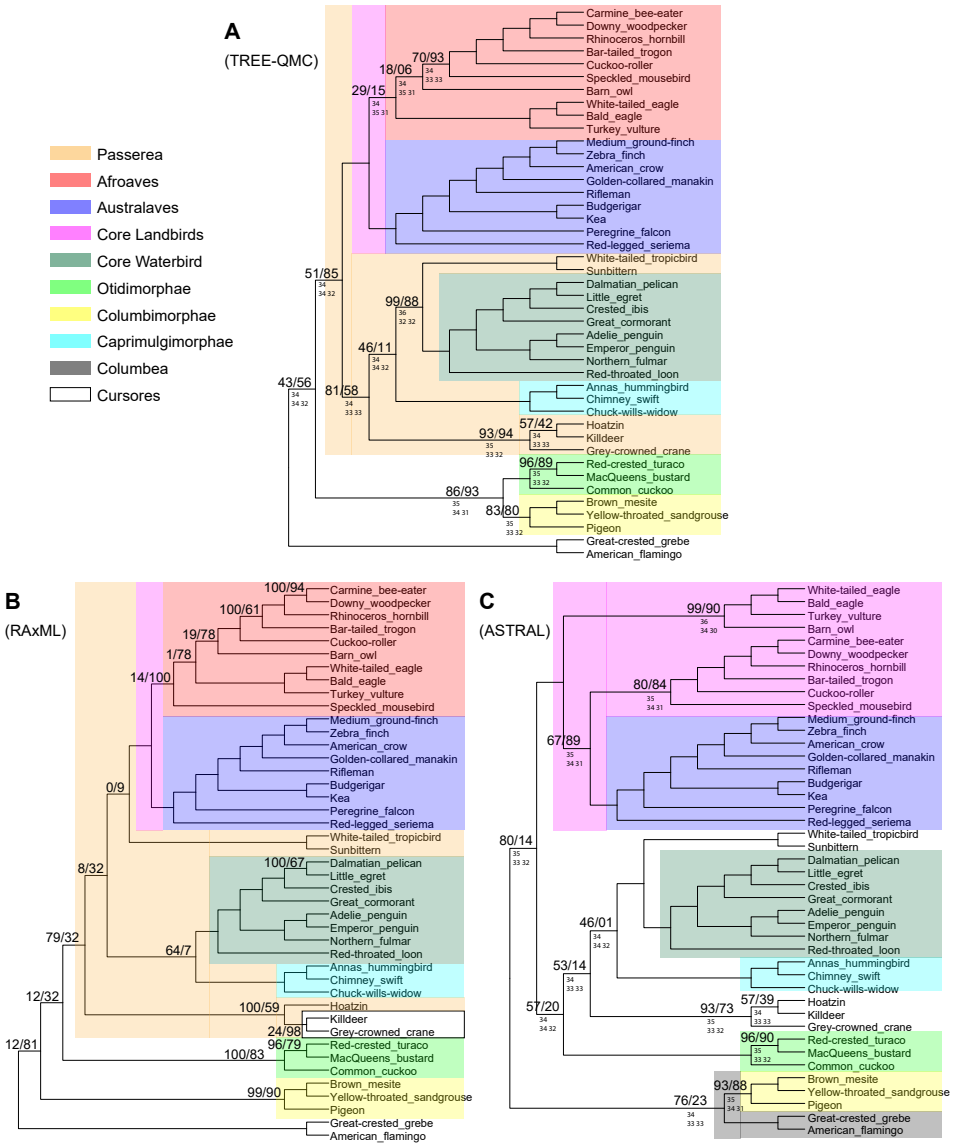


Figure 5: (A) Species tree estimated from UCE gene trees using TREE-QMC-n2 or wQFM. (B) Species tree estimated from concatenated UCE alignment using RAxML. (C) Species tree estimated from UCE gene tres using ASTRAL-III or FASTRAL. Above the branch, we show support values X/Y, where X is estimated using ASTRAL's local posterior probability (multiplied by 100) and Y is using MLBS for subfigures A and C and Y is the bootstrap support computed by RAxML for subfigure B. Support values are only shown when X is less than 100. 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). Taxa outside of Neoaves are not shown as all methods recovered the same topology outside of Neoaves.

Discussion

Our method TREE-QMC builds upon the algorithmic framework of wQMC (Avni et al. 2014) by introducing the *normalized* quartet graph and showing that it can be computed directly from gene trees. These contributions together enable our new method TREE-QMC to be highly competitive with the leading MQSST heuristics, even outperforming them. In our simulation study, TREE-QMC (with non-uniform normalization) is more accurate than other methods when the amount of gene tree heterogeneity due to ILS and/or GTEE is high and when the number of species is large. These scenarios are known challenges to species tree estimation and the issue of GTEE, in particular, has motivated a new version of ASTRAL, dubbed weighted ASTRAL (Zhang and Mirarab 2022), which was published during our study. The idea behind weighted ASTRAL is that quartets should be weighted based on the estimated gene trees, specifically branch support on the internal branch of the quartet and/or branch lengths on the terminal edges of the quartet. TREE-QMC's non-uniform normalization scheme also weights quartets but does so based subproblem division (i.e., quartets are upweighted if they are on species in more closely related subproblems, which ideally reflects closeness in the true species tree). In the future, it would be interesting to compare TREE-QMC to weighted ASTRAL as well as to implement other quartet weighting schemes within TREE-QMC.

There are several other opportunities for future work worth mentioning. First, the version of TREE-QMC presented here requires binary gene trees as input. Thus, TREE-QMC was given gene trees that are randomly refined in our experimental study, whereas all other methods were given gene trees with polytomies. This did not have a negative impact on TREE-QMC's performance relative to the other MQSST heuristics; however, it would be worth exploring this issue further. Ultimately, this inherent limitation of TREE-QMC could be addressed by devising an efficient algorithm for computing the " edges" in the quartet graph (see Methods section), although this would come at the cost of increased runtime. Second, the experimental study presented here only evaluates TREE-QMC in the context of multi-locus species tree estimation where gene tree can be discordant with the species tree

due to ILS and/or GTEE. Our study does not address the use of TREE-QMC as a more general quartet-based supertree method, and future work should explore whether quartet weight normalization is beneficial in this context. Lastly, TREE-QMC's algorithm operates on gene trees that are multi-labeled due to artificial taxa, so the algorithms presented here can be applied to gene trees that are multi-labeled due to other causes, such as multiple individuals being sampled per species (Rabiee et al. 2019) or genes evolving via duplications (Legried et al. 2021; Zhang et al. 2020; Yan et al. 2021; Smith et al. 2022). Future work should explore the effectiveness of TREE-QMC under these conditions as well those characterized by missing data due to gene loss or other causes (Nute et al. 2018).

Methods

We begin with some notation and 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 (species), and ϕ maps leaves in g to labels in $\mathcal{L}$. If ϕ is a bijection, we say that T is *singly-labeled*; otherwise, we say T is *multi-labeled*. Trees may be either *unrooted* or *rooted*. Henceforth, all trees are *binary*, meaning that non-leaf, non-root vertices (referred to as *internal* vertices) have degree 3. For a tree T, we denote its edge set as $E(T)$, its internal vertex set as $V(T)$, and its leaf set as $L(T)$. 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).

To present TREE-QMC, we need two additional concepts: *bipartitions* and *quartets*. A bipartition splits a set $\mathcal{L}$ 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 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 $\{\pi(e) : e \in E(T)\}$. The bipartition is trivial if $|\mathcal{E}|$ or $|\mathcal{F}|$ is 1; otherwise, 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. It is easy to see that there are three possible quartet trees given by their one non-trivial bipartition: $a, b|c, d$, $a, c|b, d$, and $a, d|b, c$ (note that we typically use lower case letters to denote leaf vertices and capital letters to denote leaf labels, although 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 $q \in Q(T)$.

Review of wQMC

As previously mentioned, our new MQSST heuristic, TREE-QMC, builds upon the divide-and-conquer method wQMC (Avni et al. 2014). To produce a bipartition on $\mathcal{X}$, wQMC constructs a graph from $\mathcal{Q}$, referred to as the **quartet graph**, and then seeks its 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 $\mathbb{G}$, where $w_{\mathcal{T}}(q)$ corresponds to the number of gene trees in the input set $\mathcal{T}$ that display q. 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 $\mathcal{C}$ to maximize $\sum_{(X,Y)\in\mathcal{C}}(\mathbb{G}[X, Y] - \alpha\mathbb{B}[X, Y])$, 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). The cut gives a bipartition

in the output species tree and the wQMC method proceeds by recursion on the two subsets of species on each side of the bipartition. Artificial taxa are introduced to represent the species on the other side of the bipartition.

Quartet Weight Normalization

Our key observation is that artifical taxa change the quartet weights so that a single gene tree will vote multiple times for quartets on artificial taxa and only once for quartets on only non-artificial taxa (called singletons). As shown in Figure 1, the weight of quartet $M, N|O, P$ is

$$f_0(M, N|O, P) = \sum_{m \in \mathbf{M}} \sum_{n \in \mathbf{N}} \sum_{o \in \mathbf{O}} \sum_{p \in \mathbf{P}} w_{\mathcal{T}}(m, n|o, p) \tag{1}$$

where $\mathbf{M} \subset \mathcal{L}$ denotes the set of leaves (i.e., species) in T associated with label M (and and similarly for $\mathbf{N}, \mathbf{O}, \mathbf{P}$). 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. In the simplest case, we simply divide by the number of votes cast so the weight of $M, N|O, P$ becomes

$$f_1(M, N|O, P) = \frac{f_0(M, N|O, P)}{|\mathbf{M}| \cdot |\mathbf{N}| \cdot |\mathbf{O}| \cdot |\mathbf{P}|} \tag{2}$$

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

$$f(M, N|O, P) = \sum_{m \in \mathbf{M}, n \in \mathbf{N}, o \in \mathbf{O}, p \in \mathbf{P}} I(m, n, o, p) \cdot w_{\mathcal{T}}(m, n|o, p) \tag{3}$$

where $I(m, n, o, p) = I(m) \cdot I(n) \cdot I(o) \cdot I(p)$. Specifically, Equation 3 reduces to Equation 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 (n1)*. More broadly, the quartet weights will be normalized whenever Equation 3 corresponds to a weighted average, meaning that

$$\sum_{m\in\mathbf{M}}\sum_{n\in\mathbf{N}}\sum_{o\in\mathbf{O}}\sum_{p\in\mathbf{P}} I(m,n,o,p) = \sum_{m\in\mathbf{M},n\in\mathbf{N},o\in\mathbf{O},p\in\mathbf{P}} I(m,n,o,p) = 1 \tag{4}$$

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 3 reduces to Equation 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. In Figure 1, artificial taxon Z represents species $\mathbf{Z} = \{0,6,7,9\}$ but species 0 and 9 were previously labeled by artificial taxon X. 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

$$I(z) = \prod_{v\in path(T_Z,z)} \frac{1}{outdegree(v)} \tag{5}$$

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 improved species tree accuracy for challenging model conditions.

Efficient Quartet Graph Construction

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. 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 with n leaves. In this case, we can compute the number of good edges between X, Y via $\mathbb{G}[X, Y] = \binom{n-2}{2} - \mathbb{B}[X, Y]$, 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 (Figure 6a). Deleting the edges on this path (and their end points) produces a forest of K rooted 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 $x, y|w, z$ 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 typically the case due to 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 quartet $0, 17|M, M$ has no topological information and should not be counted. The other quartets $0, 17|4, 6$ and $0, 17|5, 6$ correspond to $0, 17|M, 6$ and thus should be counted.

We now present an algorithm for computing $\mathbb{B}$ in $O(s^2 n)$ time, where n is the number

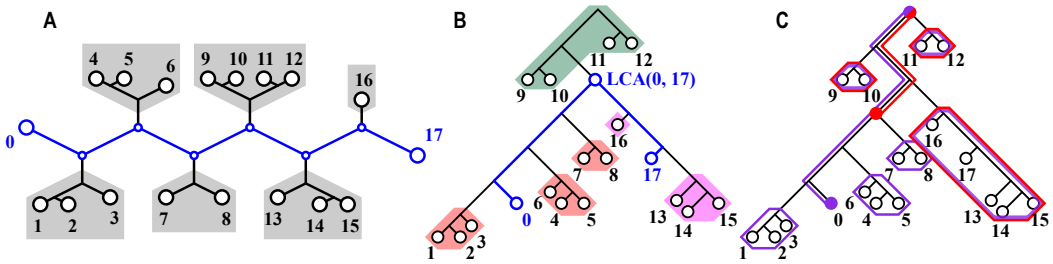


Figure 6: To count the quartets induced by T with 0 and 17 as siblings, we consider the path between them (shown in blue in (a)). The deletion of the path produces 6 rooted subtrees (highlighted in grey). Because 0 and 17 are siblings in a quartet if and only if the other two taxa are drawn from the same subtree, the number of bad edges can be computed as $\binom{3}{2} + \binom{3}{2} + \binom{2}{2} + \binom{4}{2} + \binom{3}{2} + \binom{1}{2} = 16$. Here we show how to compute the number of quartets induced by T with 0 and 17 as siblings after rooting T arbitrarily. Subfigure (b) 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, $\mathbb{A}[0, 17] = 6$ (Eq. 8). Cases 2 and 3 can also be computed in constant time as follows. Subfigure (c) 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 (b)) is $\mathbb{L}[0, 17] = p[0] - p[lca(0, 17).left] = 7$ (Eq. 9). Case 3 (not shown) can be computed as $\mathbb{R}[0, 17] = p[17] - p[lca(0, 17).right] = 3$ (Eq. 10). Putting this all together gives $\mathbb{B}[0, 17] = 16$ (Eq. 6).

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:

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

To summarize our 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 (Theorem 1 in the Supplementary Materials). 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 2 in the Supplementary Materials). Although we do not expect perfectly balanced subproblems in practice, we found TREE-QMC to be fast in our experiments.

Computing the number of bad edges given a singly-labeled gene tree

We first present an algorithm for computing the number of bad edges given a singly-labeled gene tree T. 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 6b). This allows us to break the computation into three parts

$$\mathbb{B}[X, Y] = \mathbb{A}[X, Y] + \mathbb{L}[X, Y] + \mathbb{R}[X, Y] \tag{6}$$

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, in which 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 (Figure 6c). 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

$$p[v] = p[v.parent] + \binom{c[v.sibling]}{2} \tag{7}$$

after initializing $p[root] = 0$. Now we can compute the quantities:

$$\mathbb{A}[X, Y] = \binom{n - c[lca(x, y)]}{2} \tag{8}$$

$$\mathbb{L}[X, Y] = p[x] - p[lca(x, y).left] \tag{9}$$

$$\mathbb{R}[X, Y] = p[y] - p[lca(x, y).right] \tag{10}$$

where $v.left$ denotes the left child of v and $v.right$ denotes the right child of v (see Figure 6c). It is possible to access $lca(x, y)$ in constant time after $O(n)$ preprocessing step (Gusfield 1997), although we implemented this implicitly by computing the entries of $\mathbb{B}$ during a post-order traversal of T. Thus, we can compute $\mathbb{B}$ in $O(n^2)$ time, provided that T is singly-labeled.

Computing the number of bad edges given a multi-labeled gene tree

We now present an algorithm for computing the number of bad edges $\mathbb{B}[X, Y]$ given a multi-labeled gene tree T. As previously mentioned, this breaks down into three cases. The first case (X, Y are both singletons) is below and the remaining two cases are presented in the Supplementary Materials.

Again, we focus 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 so that they are **uniquely labeled** $Z \neq W$ (note that we use capital letters W and Z to denote the current labels of leaves w and z, respectively). This modified binomial is computed by revising the preprocessing phase. We now 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. As before, we can compute c in $O(bn)$ time via a postorder traversal. However, the number of ways to select two leaves with different labels is now broken into three cases:

1. the number of ways to select two singletons, which equals $\binom{c_0[v]}{2}$,
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, and
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**:

$$g_0[v] = \binom{c_0[v]}{2} + c_0[v] \cdot G_1[v] + \frac{G_1[v]^2 - G_2[v]}{2} \tag{11}$$

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 our algorithm for singly-labeled trees by redefining prefix sum:

$$p_0[v] = p_0[v.parent] + g_0[v.sibling] \tag{12}$$

and then redefining the quantities from which we can compute $B[x, y]$ in constant time, that is, $\mathbb{A}[X, Y] = g_0[lca(x, y).above]$, and $\mathbb{L}[X, Y] = p_0[x] - p_0[lca(x, y).left]$, and $\mathbb{R}[X, Y] = p_0[y] - p_0[lca(x, y).right]$. As there are a^2 pairs of singletons in the subproblem, the total runtime is $O(a^2 + bn)$.

Normalizing quartet weights when computing bad edges

To normalize the quartet weights, $\mathbb{B}[X, Y]$ 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,

$$\mathbb{B}[X,Y] = I(x)I(y) \sum_{\substack{w,z \in L(T): Z \neq W \neq X \neq Y, \\ q(x,y,z,y) = x,y|z,y}} I(z)I(w) \tag{13}$$

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 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 $g_0[u]$.

Lemma 1. *The total weight of all taxon pairs in the subtree rooted at internal vertex u*

$$\sum_{\substack{z,w \in L(u): \\ Z \neq W}} I(z)I(w) = g_0[u] \tag{14}$$

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

See Supplementary Materials for proof.

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 2.

Lemma 2. *Let T be a multi-labeled gene tree, and let X, Y be singletons. Then,*

$$\mathbb{G}[X,Y] + \mathbb{B}[X,Y] = \binom{c_0[r]-2}{2} + (c_0[r]-2) \cdot G_1[r] + \frac{G_1[r]^2 - G_2[r]}{2} \tag{15}$$

where r is the root vertex of T.

See Supplementary Materials for proof.

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 for these cases are provided in the Supplementary Materials.

Software and Data Availability

TREE-QMC is available on Github: `https://github.com/molloy-lab/TREE-QMC`. The scripts used to run methods and analyze the results are also available on Github: `https://github.com/molloy-lab/tree-qmc-study`. The data (including true and estimated gene trees as well as true and estimated species trees) are available on Dryad: `https://doi.org/10.5061/dryad.m0cfxpp6g`.

Data Access

This research did not generate new data.

Competing Interest Statement

The authors have no competing interests.

Acknowledgments

This research was funded by the State of Maryland.

References

Avni E, Cohen R, Snir S. 2014. Weighted quartets phylogenetics. *Syst Biol* **64**: 233–242.

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

Degnan JH, Salter LA. 2005. Gene tree distributions under the coalescent process. *Evolution* **59**: 24–37.

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

Dunning I, Gupta S, Silberholz J. 2018. What works best when? a systematic evaluation of heuristics for Max-Cut and QUBO. *INFORMS J Comput* **30**: 421–624.

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

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

Karp RM. 1972. Reducibility among Combinatorial Problems. In *Complexity of Computer Computations: The IBM Research Symposia Series*. (eds. Miller RE, Thatcher JW, Bohlinger JD), pp. 85–103. Springer, Boston, MA.

Kozlov AM, Aberer AJ, Stamatakis A. 2015. ExaML version 3: a tool for phylogenomic analyses on supercomputers. *Bioinformatics* **31**: 2577–2579.

Kubatko LS, Degnan JH. 2007. Inconsistency of phylogenetic estimates from concatenated data under coalescence. *Syst Biol* **56**: 17–24.

Lafond M, Scornavacca C. 2019. On the weighted quartet consensus problem. *Theor Comput Sci* **769**: 1–17.

Legried B, Molloy EK, Warnow T, Roch S. 2021. Polynomial-time statistical estimation of species trees under gene duplication and loss. *J Comput Biol* **28**: 452–468.

Maddison W. 1997. Gene trees in species trees. *Syst Biol* **46**: 523–536.

Mahbub M, Wahab Z, Reaz R, Rahman MS, Bayzid MS. 2021. wQFM: highly accurate genome-scale species tree estimation from weighted quartets. *Bioinformatics* **37**: 3734–3743.

McCormack JE, Faircloth BC, Crawford NG, Gowaty PA, Brumfield RT, Glenn TC. 2012. Ultraconserved elements are novel phylogenomic markers that resolve placental mammal phylogeny when combined with species-tree analysis. *Genome Res* **22**: 746–754.

Meiklejohn KA, Faircloth BC, Glenn TC, Kimball RT, Braun EL. 2016. Analysis of a rapid evolutionary radiation using ultraconserved elements: Evidence for a bias in some multispecies coalescent methods. *Syst Biol* **65**: 612–627.

Mirarab S, Bayzid MS, Boussau B, Warnow T. 2014a. Statistical binning enables an accurate coalescent-based estimation of the avian tree. *Science* **346**: 1250463.

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

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

Molloy EK, Warnow T. 2018. To include or not to include: The impact of gene filtering on species tree estimation methods. *Syst Biol* **67**: 285–303.

Nute M, Chou J, Molloy EK, Warnow T. 2018. The performance of coalescent-based species tree estimation methods under models of missing data. *BMC Genom* **19(Suppl 5)**: 286.

Pamilo P, Nei M. 1988. Relationships between gene trees and species trees. *Mol Biol Evol* **5**: 568–583.

Rabiee M, Sayyari E, Mirarab S. 2019. Multi-allele species reconstruction using ASTRAL. *Mol Phylogenet Evol* **130**: 286–296.

Roch S, Nute M, Warnow T. 2018. Long-branch attraction in species tree estimation: Inconsistency of partitioned likelihood and topology-based summary methods. *Syst Biol* **68**: 281–297.

Roch S, Steel M. 2015. Likelihood-based tree reconstruction on a concatenation of aligned sequence data sets can be statistically inconsistent. *Theor Popul Biol* **100**: 56–62.

Rosenberg NA. 2002. The probability of topological concordance of gene trees and species trees. *Theor Popul Biol* **61**: 225–247.

Sayyari E, Mirarab S. 2016. Fast coalescent-based computation of local branch support from quartet frequencies. *Mol Biol Evol* **33**: 1654–1668.

Seo TK. 2008. Calculating bootstrap probabilities of phylogeny using multilocus sequence data. *Mol Biol Evol* **25**: 960–971.

Smith ML, Vanderpool D, Hahn MW. 2022. Using all gene families vastly expands data available for phylogenomic inference. *Mol Biol Evol* **39**: msac112.

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

Snir S, Rao S. 2012. Quartet MaxCut: A fast algorithm for amalgamating quartet trees. *Mol Phylogenet Evol* **62**: 1–8.

Song S, Liu L, Edwards SV, Wu S. 2012. Resolving conflict in eutherian mammal phylogeny using phylogenomics and the multispecies coalescent model. *Proc Natl Acad Sci USA* **109**: 14942–14947.

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. *Lect math life sci* **17**: 57–86.

Wickett NJ, Mirarab S, et al. 2014. Phylotranscriptomic analysis of the origin and early diversification of land plants. *Proc Natl Acad Sci USA* **111**: E4859–E4868.

Xi Z, Liu L, Davis CC. 2015. Genes with minimal phylogenetic information are problematic for coalescent analyses when gene tree estimation is biased. *Mol Phylogenet Evol* **92**: 63–71.

Yan Z, Smith ML, Du P, Hahn MW, Nakhleh L. 2021. Species tree inference methods intended to deal with incomplete lineage sorting are robust to the presence of paralogs. *Syst Biol* **71**: 367–381.

Zhang C, Mirarab S. 2022. Weighting by gene tree uncertainty improves accuracy of quartet-based species trees. *Mol Biol Evol* **39**: msac215.

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

Zhang C, Scornavacca C, Molloy EK, Mirarab S. 2020. ASTRAL-Pro: Quartet-based species-tree inference despite paralogy. *Mol Biol Evol* **37**: 3292–3307.

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

Yunheng Han and Erin K. Molloy

December 23, 2022

Contents

List of Tables 2

List of Figures 2

1 Divide-and-Conquer Framework 3

2 Details of Experimental Study 4
- 2.1 Replicates Excluded from ASTRAL-II Data . . . 4
- 2.2 Properties of Simulated Data . . . 4
- 2.3 Computational Resources and Empirical Runtime . . . 6
- 2.4 Species Tree Estimation Commands . . . 6
  - 2.4.1 ASTRAL, FASTRAL, and TREE-QMC. . . . 6
  - 2.4.2 wQMC and wQFM. . . . 6
- 2.5 Quartet Score and Branch Support Estimation Commands . . . 7

3 Additional Results on Simulated Data Sets 9
- 3.1 Number of taxa . . . 9
- 3.2 Species tree scale/height and thus ILS level . . . 10
- 3.3 Sequence length and thus GTEE level . . . 13
- 3.4 Number of Gene Trees . . . 14
- 3.5 Additional Results on Biological Data Sets . . . 17

4 Other Plots for ASTRAL-II Data Sets 17

5 TREE-QMC Algorithm 21
- 5.1 Quartet Graph Construction Case 1 cont. (Two Singletons) . . . 21
- 5.2 Quartet Graph Construction Case 2 (Singleton and Artificial Taxon) . . . 22
- 5.3 Quartet Graph Construction Case 3 (Two Artificial Taxa) . . . 23
- 5.4 Algorithms . . . 25
- 5.5 Time Complexity . . . 27
- 5.6 Correctness . . . 28

References 34

List of Tables

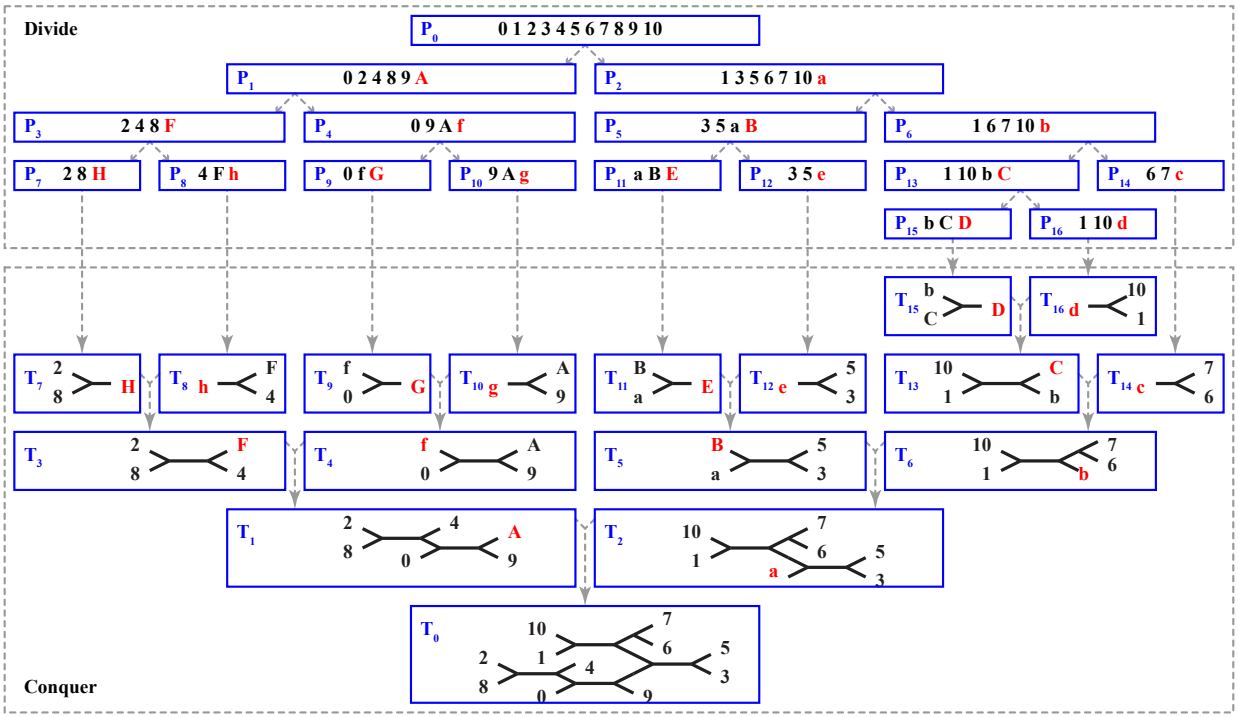
S1 Properties of ASTRAL-II data . . . 4
S2 Properties of avian and mammalian simulated data . . . 5
S3 Runtimes for ASTRAL-II data (1000 estimated gene trees) . . . 8
S4 Statistical testing for ASTRAL-II data varying number of taxa (1000 estimated gene trees) . . . 9
S5 Statistical testing for ASTRAL-II data varying ILS level (1000 estimated gene trees) . . . 10
S6 Statistical testing for avian simulated data with varying ILS level . . . 11
S7 Statistical testing for mammalian simulated data varying ILS level . . . 12
S8 Statistical testing for mammalian simulated data varying GTEE level . . . 13
S9 Statistical testing for avian simulated data varying number of gene trees . . . 15
S10 Statistical testing for avian and mammalian simulated data varying number of true and estimated gene trees, respectively . . . 16

List of Figures

S1 Divide-and-conquer framework utilized by wQMC, TREE-QMC, and wQFM . . . 3
S2 Reproduction of Figure 4 with different y-axis . . . 10
S3 Error and runtime for mammalian simulated data varying ILS level. . . . 12
S4 Error and runtime for mammalian simulated data varying GTEE level. . . . 13
S5 Error and runtime for avian and mammalian simulated data varying number of gene trees. . . . 14
S6 Avian TENT species trees . . . 17
S7 Error and runtime for ASTRAL-II data varying number of taxa (1000 true gene trees). . . . 18
S8 Error and runtime for ASTRAL-II data varying number of taxa (250 true gene trees). . . . 18
S9 Error and runtime for ASTRAL-II data varying number of taxa (250 estimated gene trees). . . . 19
S10 Error and runtime for ASTRAL-II data varying ILS level (1000 true gene trees). . . . 19
S11 Error and runtime for ASTRAL-II data varying ILS level (250 true gene trees). . . . 20
S12 Error and runtime for ASTRAL-II data varying ILS level (250 estimated gene trees). . . . 20
S13 TREE-QMC Algorithm Case 2 Example . . . 23
S14 TREE-QMC Algorithm Case 3 Example . . . 24

1 Divide-and-Conquer Framework

Figure S1: **Divide-and-conquer framework utilized by wQMC, TREE-QMC, and wQFM.** Initially, the input data (either gene trees or weighted quartets) are on species set $\mathcal{S} = \{0, 1, \ldots, 10\}$. At each step in the divide phase, we identify a bipartition and recurse on the implied subproblems. For problem P_0, we identify bipartition $\mathcal{E}|\mathcal{F}$, where $\mathcal{E} = \{0, 2, 4, 8, 9\}$ and $\mathcal{F} = \{1, 3, 5, 6, 7, 10\}$. We then introduce artifical taxa A (representing the taxa in $\mathcal{F}$) and a (representing the taxa in $\mathcal{E}$) and recurse on species sets $\{A\} \cup \mathcal{E}$ (subproblem P_1) and $\{a\} \cup \mathcal{F}$ (subproblem P_2), updating the input data (either gene trees or weighted quartets) accordingly. We continue in this fashion until there are fewer than four taxa (e.g., problem P_7) in which case we return the only possible tree (e.g., tree T_7). At each step in the conquer phase, we connect pairs of trees using the artifical taxa. On the final conquer step, we connect trees T_1 and T_2 at artificial taxa A and a, producing a tree on species set $\mathcal{S}$.



Subproblems and Artificial Taxa. At each step in the divide phase, wQMC finds a bipartition on species set $\mathcal{X}$ using the input quartets $\mathcal{Q}_{\mathcal{X}}$. The the quartets are then updated based on the implied subproblems (Avni et al. 2014). Given bipartition $\mathcal{E}|\mathcal{F}$, a 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),
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 $|\{A, B, C, D\} \cap \mathcal{F}| = 3$.

If the quartet is satisfied or violated, it is discarded because it will be satisfied or violated regardless of the bipartitions produced at future steps in the divide phase of the algorithm. If the quartet is untoched or

deferred, it is assigned to one of the two subproblems: $\mathcal{Q}_{\mathcal{E}+}$ or $\mathcal{Q}_{\mathcal{F}+}$, where $\mathcal{Q}_{\mathcal{E}+}$ denotes the set of quartets formed by adding the untouched and deferred quartets with 4 or 3 leaves labeled by elements of $\mathcal{E}$ (and similarly for $\mathcal{Q}_{\mathcal{F}+}$). To add a deferred quartet to $\mathcal{Q}_{\mathcal{E}+}$, its one leaf labeled by an element of $\mathcal{F}$ is relabeled with artificial taxon F. After this relabeling process has been completed for all deferred quartets, there can be multiples of the same quartet (each with its own weight). These multiples are replaced by a single quartet whose weight is the sum.

2 Details of Experimental Study

2.1 Replicates Excluded from ASTRAL-II Data

All comparisons between methods are made on the same set of replicate data sets. For analyses of estimated gene trees, we excluded replicate data set from analyses whenever more than half of the 1000 gene trees had the majority of their branches unresolved. Specifically, we excluded

- 1 replicate (# 41) from the 10-taxon model condition
- 2 replicates (# 21, 41) from the 50-taxon model condition
- 2 replicates (# 8, 47) from the 100-taxon model condition
- 3 replicates (# 8, 15, 49) from the 200 taxon model condition with very high ILS and shallow speciation

We also excluded replicates on which ASTRAL-III failed to complete; this occurred only for 3 replicates (# 6, 8, 38) from the 1000-taxon, 1000-gene model condition.

2.2 Properties of Simulated Data

Table S1: **Properties of ASTRAL-II data sets.** ILS for each replicate is quantified as the normalized RF distance between the true species tree and the true gene tree average across all 1000 gene trees. GTEE for each replicate is quantified as the normalized RF distance between the true and estimated gene trees averaged across all 1000 gene trees. AD for each replicate is the normalized RF distance between the true species tree and estimated gene tree averaged across all 1000 gene trees. The values in the table is the average ($\pm$ standard deviation) across all replicates.

species tree height	speciation	# of taxa	ILS	GTEE	AD
0.5X	deep	200	0.68 ± 0.02	0.44 ± 0.14	0.74 ± 0.03
0.5X	shallow	200	0.69 ± 0.02	0.44 ± 0.12	0.74 ± 0.03
1X	shallow	10	0.17 ± 0.06	0.19 ± 0.09	0.28 ± 0.08
1X	shallow	50	0.31 ± 0.04	0.26 ± 0.11	0.42 ± 0.08
1X	shallow	100	0.33 ± 0.02	0.26 ± 0.09	0.44 ± 0.05
1X	deep	200	0.34 ± 0.02	0.34 ± 0.12	0.47 ± 0.08
1X	shallow	200	0.34 ± 0.02	0.27 ± 0.12	0.44 ± 0.07
1X	shallow	500	0.34 ± 0.01	0.28 ± 0.11	0.45 ± 0.07
1X	shallow	1000	0.35 ± 0.01	0.30 ± 0.11	0.47 ± 0.08
5X	deep	200	0.09 ± 0.01	0.28 ± 0.11	0.31 ± 0.10
5X	shallow	200	0.21 ± 0.02	0.21 ± 0.13	0.33 ± 0.09

Table S2: **Properties of avian and mammalian simulated data.** Downloaded data sets Mahbub et al. (2021), which did not include true gene trees for the 0.5X and 2X species tree scales for mammalian data set.

species tree scale	sequence length	ILS	GTEE	AD
Avian (48 taxa, 1000 estimated gene trees)				
0.5X	500	0.591 ± 0.002	0.597 ± 0.094	0.687 ± 0.001
1X	500	0.473 ± 0.002	0.597 ± 0.066	0.637 ± 0.002
2X	500	0.354 ± 0.002	0.619 ± 0.002	0.597 ± 0.002
Mammalian (37 taxa, 200 estimated gene trees)				
1X	250	0.330 ± 0.005	0.427 ± 0.009	0.538 ± 0.007
1X	500	0.330 ± 0.005	0.282 ± 0.058	0.438 ± 0.006
1X	1000	0.330 ± 0.005	0.161 ± 0.006	0.380 ± 0.006
1X	1500	0.330 ± 0.005	0.119 ± 0.005	0.362 ± 0.006

2.3 Computational Resources and Empirical Runtime

Running time is reported as the wall-clock time (i.e., the amount of time that elapses between the beginning and end of the computation). All computational experiments were performed on cluster nodes outfitted with 128 GB RAM and 32 cores (2 × 16-Core AMD Opteron(tm) Processor 6378 2400MHz) and running Red Hat Enterprise Linux 7 (x86_64) operating system. There are 12 nodes with these specifications, which can be accessed via a queue that allows exclusive access but limits the allowed memory to 36 GB. Therefore, we ran methods with exclusive access to the node with access to 36GB of memory, assigning all analyses of the same replicate data set to the same node. All five methods we evaluated are single-threaded.

2.4 Species Tree Estimation Commands

2.4.1 ASTRAL, FASTRAL, and TREE-QMC.

ASTRAL version 5.7.7 (Zhang et al. 2018) was run with the following command:

```
java -Xmx36G -D"java.library.path=<path to ASTRAL>/lib" -jar <path to ASTRAL>/astral.5.7.7.jar \
     -t0 -i [input gene trees] -o [output species tree] &> [output log file]
```

Note that `-t0` means that branch length and support estimation are not performed.
FASTRAL (Dibaeinia et al. 2021) was run with the following command:

```
fastral --ns 1,10,20,20 --nt 1000,500,250,100 --k 1000 \
        --it [input gene trees] --os [output directory]/samples \
        --aggregate [output directory]/[output constraints] \
        --o [output directory]/[output species tree] \
        --time [output directory]/[output timing file] \
        --path_ASTRAL [ASTRAL path] \
        --path_ASTRID [ASTRID path] &> [output log file]
```

Note that `-nt 250,125,63,25 -k 250` where used for data sets with 250 genes. We ran FASTRAL with the version of ASTRAL and ASTRID distributed with the code; however, we modified the command to call ASTRAL so that it could use 36 GB of RAM and so that it did NOT compute branch support for the final tree.

TREE-QMC version 1.0.0 was run with the following command:

```
TREEQMC -n 0 -i [input gene trees] -o [output species tree] &> [output log file]
```

Note that `-n 1` and `-n 2` where also run and in this case the input gene trees were the refined trees from running TREE-QMC with `-n 0`. This is so the only difference between run of TREE-QMC is the normalization scheme (and not the refinement of polytomies in the input gene trees).

2.4.2 wQMC and wQFM.

Unlike the previous methods, wQMC and wQFM take weighted quartets as input. We used the scripts provided with wQFM (https://github.com/Mahim1997/wQFM-2020) to extract weighted quartets from the gene trees. Specifically, we used the command:

```
quartet-controller.sh [input gene trees] [output weighted quartets] &> [output log file]
```

which calls by `triplets.soda2103` executable from (). We then selected the binary quartets

```
grep "),(" [input weighted quartets] >> [output weighted quartets for wQFM]
```

and re-formatted them for wQMC:

```
sed 's/),(/|/g' [input weighted quartets for wQFM] | \
     sed 's/(//g' | sed 's/)//g' | sed 's/; /:/g' \
     > [output weighted quartets for wQMC]
```

wQFM version 1.3 (Mahbub et al. 2021) was run with the following command:

```
java -Xmx36G -jar [path to wQFM]/wQFM-v1.3.jar \
     -i [input weighted quartets] -o [output species tree] &> [output log file]
```

wQMC version 3.0 (Avni et al. 2014) was run with the following command:

```
max-cut-tree qrtt=[input weighted quartets] \
             weights=on otre=[output species tree] &> [output log file]
```

2.5 Quartet Score and Branch Support Estimation Commands

Quartet score (and branch support) was computed for estimated species trees with the following command:

```
java -Xmx36G -D"java.library.path=<path to ASTRAL>/lib" -jar <path to ASTRAL>/astral.5.7.7.jar \
     -t2 -q [input species tree] -i [input gene trees] \
     -o [output scored species tree] &> [output log file]
```

Table S3: **Runtimes for ASTRAL-II data (1000 estimated gene trees).** Runtime (minutes) is reported for all methods (note: a runtime of zero minutes means it rounded down). Values shown are averages (± standard deviations) across across replicate data sets on which all methods completed (note: ASTRAL-III did not complete on 3 replicates with 1000 taxa and 1000 genes). Lowest median values are in bold. The value in parentheses for wQMC and wQFM is the fraction of the runtime spent weighting quartets to give to these methods as input.

Species tree height	# of taxa	spec.	ASTRAL-III	FASTRAL	TREE-QMC n0	TREE-QMC n1	TREE-QMC n2	wQMC	wQFM
1000 estimated gene trees									
0.5X	200	deep	77.4 ± 27.9	**1.6 ± 0.4**	2.7 ± 0.5	3.0 ± 0.6	3.0 ± 0.6	NA	NA
0.5X	200	shallow	72.7 ± 29.0	**1.5 ± 0.3**	2.6 ± 0.5	2.8 ± 0.5	2.8 ± 0.3	NA	NA
2X	10	shallow	**0.0 ± 0.0**	0.1 ± 0.0	**0.0 ± 0.0**	**0.0 ± 0.0**	**0.0 ± 0.0**	0.2 ± 0.0 (0.99)	0.2 ± 0.0 (0.94)
2X	50	shallow	0.6 ± 0.6	0.3 ± 0.0	**0.2 ± 0.0**	**0.2 ± 0.0**	**0.2 ± 0.0**	15.8 ± 3.0 (1.00)	16.8 ± 3.0 (0.94)
2X	100	shallow	3.0 ± 2.4	0.7 ± 0.1	**0.6 ± 0.1**	**0.6 ± 0.1**	**0.6 ± 0.1**	256.8 ± 34.7 (1.00)	290.2 ± 38.4 (0.88)
2X	200	deep	17.1 ± 11.4	**1.2 ± 0.1**	2.4 ± 0.1	2.5 ± 0.2	2.5 ± 0.2	NA	NA
2X	200	shallow	12.0 ± 9.5	**1.2 ± 0.2**	2.4 ± 0.4	2.5 ± 0.4	2.5 ± 0.4	NA	NA
2X	500	shallow	72.1 ± 46.2	**5.6 ± 1.1**	14.8 ± 2.7	15.8 ± 3.2	15.5 ± 2.5	NA	NA
2X	1000	shallow	317.8 ± 198.9	**32.4 ± 7.7**	59.2 ± 11.6	62.0 ± 9.2	63.5 ± 13.2	NA	NA
5X	200	deep	6.6 ± 7.7	**1.2 ± 0.0**	2.2 ± 0.1	2.3 ± 0.1	2.3 ± 0.1	NA	NA
5X	200	shallow	4.6 ± 7.9	**1.1 ± 0.0**	2.2 ± 0.0	2.3 ± 0.0	2.3 ± 0.0	NA	NA
250 estimated gene trees									
0.5X	200	deep	11.3 ± 4.8	0.8 ± 0.2	**0.7 ± 0.1**	**0.7 ± 0.1**	**0.7 ± 0.2**	NA	NA
0.5X	200	shallow	9.6 ± 3.4	0.7 ± 0.1	**0.6 ± 0.0**	0.7 ± 0.0	0.7 ± 0.0	NA	NA
2X	10	shallow	**0.0 ± 0.0**	0.1 ± 0.0	**0.0 ± 0.0**	**0.0 ± 0.0**	**0.0 ± 0.0**	0.1 ± 0.0 (0.98)	0.1 ± 0.0 (0.83)
2X	50	shallow	0.1 ± 0.0	0.1 ± 0.0	**0.0 ± 0.0**	**0.0 ± 0.0**	**0.0 ± 0.0**	4.1 ± 0.8 (0.99)	5.1 ± 1.0 (0.80)
2X	100	shallow	0.3 ± 0.2	**0.2 ± 0.0**	**0.2 ± 0.0**	**0.2 ± 0.0**	**0.2 ± 0.0**	65.4 ± 8.2 (1.00)	96.4 ± 12.5 (0.68)
2X	200	deep	1.7 ± 1.2	**0.5 ± 0.0**	0.6 ± 0.0	0.6 ± 0.0	0.6 ± 0.0	NA	NA
2X	200	shallow	1.1 ± 0.7	**0.5 ± 0.1**	0.6 ± 0.1	0.6 ± 0.1	0.6 ± 0.2	NA	NA
2X	500	shallow	7.1 ± 4.7	**3.0 ± 0.8**	3.7 ± 0.5	3.9 ± 0.7	3.9 ± 0.8	NA	NA
2X	1000	shallow	45.9 ± 50.4	22.4 ± 9.4	**15.2 ± 2.5**	15.8 ± 2.2	15.5 ± 1.5	NA	NA
5X	200	deep	0.6 ± 0.6	**0.4 ± 0.0**	0.6 ± 0.0	0.6 ± 0.0	0.6 ± 0.0	NA	NA
5X	200	shallow	0.5 ± 0.6	**0.4 ± 0.0**	0.6 ± 0.0	0.6 ± 0.0	0.6 ± 0.0	NA	NA

3 Additional Results on Simulated Data Sets

3.1 Number of taxa

Table S4: **Testing for differences between TREE-QMC-n2 and other methods on ASTRAL-II data sets with varying numbers of taxa and 1000 estimated gene trees (Figure 2).** All data sets were simulated with 1X species tree height and shallow speciation. The table shows the number of replicates for which species tree estimation error was better for TREE-QMC-n2, better for the other method, or a tie as well as other metrics. The Δ false negative (FN) error is the difference in the number of false negative branches between the other method and TREE-QMC-n2, averaged ($\pm$ standard deviation) across all replicates (including ties). Positive values indicate that TREE-QMC-n2 is better on average, whereas negative values indicate that the other method is better on average. Differences between methods were evaluated using two-sided Wilcoxon signed-rank tests, after removing tied values (McGee 2018) (note: the test was performed only if more than than 25 replicates were not ties). The symbol *, *, *** indicate significance at $p < 0.05$, 0.005, and 0.0005, respectively. Note that $p < 0.05/12 = 0.0041\bar{6}$ survives Bonferroni correction for the 12 comparisons made in this table (MC).

# of taxa	Other method	# of replicates TQMC-n2	OTHER	TIE	Δ FN error	p-value		MC
10	FASTRAL	0	0	49	0.00 ± 0.00	NA		
10	ASTRAL3	0	0	49	0.00 ± 0.00	NA		
50	FASTRAL	5	3	40	0.04 ± 0.41	NA		
50	ASTRAL3	5	3	40	0.04 ± 0.41	NA		
100	FASTRAL	9	8	31	0.04 ± 0.92	NA		
100	ASTRAL3	10	8	30	-0.08 ± 1.49	NA		
200	FASTRAL	20	6	24	0.44 ± 1.09	0.006217	*	N
200	ASTRAL3	20	6	24	0.48 ± 1.13	0.004217	**	Y
500	FASTRAL	34	9	7	2.40 ± 3.46	0.000009	***	Y
500	ASTRAL3	33	8	9	2.70 ± 3.89	0.000013	***	Y
1000	FASTRAL	32	10	5	3.94 ± 6.96	0.000123	***	Y
1000	ASTRAL3	35	8	4	4.64 ± 7.84	0.000008	***	Y

3.2 Species tree scale/height and thus ILS level

Table S5: **Testing for differences between TREE-QMC-n2 and other methods on ASTRAL-II data sets with varying ILS levels and 1000 estimated gene trees (Figure 3)**. All data sets have 200 taxa. Note: rows with parentheses are duplicated from Table S4.

species tree height	speciation	Other method	# of replicates TQMC-n2	OTHER	TIE	Δ FN error	p-value		MC
0.25X	deep	FASTRAL	22	21	7	-0.24 ± 3.46	0.70		
0.25X	deep	ASTRAL3	29	18	3	0.5 ± 3.52	0.10		
0.25X	shallow	FASTRAL	23	16	8	0.47 ± 2.24	0.16		
0.25X	shallow	ASTRAL3	23	15	9	0.77 ± 2.35	0.04414	*	N
1X	deep	FASTRAL	20	9	21	0.70 ± 2.15	0.04209	*	N
1X	deep	ASTRAL3	21	7	22	1.00 ± 2.25	0.00367	**	Y
(1X)	(shallow)	FASTRAL	20	6	24	0.44 ± 1.09	0.00622	*	N
(1X)	(shallow)	ASTRAL3	20	6	24	0.48 ± 1.13	0.00422	**	N
5X	deep	FASTRAL	27	6	17	1.10 ± 1.88	0.00032	***	Y
5X	deep	ASTRAL3	25	8	17	0.90 ± 2.26	0.00382	**	Y
5X	shallow	FASTRAL	13	8	29	0.44 ± 1.39	NA		
5X	shallow	ASTRAL3	13	9	28	0.42 ± 1.42	NA		

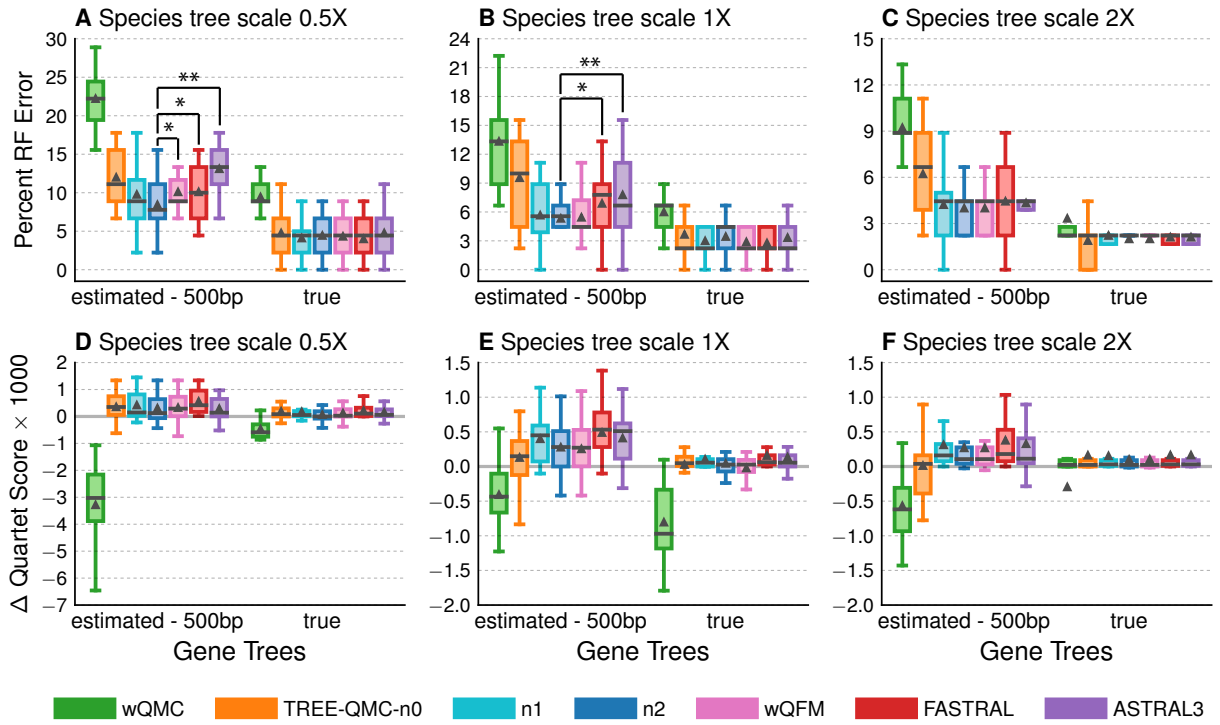


Figure S2: Reproduction of Figure 4 in main text with different y-axis.

Table S6: **Testing for differences between TREE-QMC-n2 and other methods on avian simulated data sets with species tree scales and thus ILS levels (Figure 4).** All data sets had 1000 gene trees. Differences between methods were evaluated using two-sided Wilcoxon signed-rank tests, after removing tied values (McGee 2018) (note: the test was only performed if more than than 10 replicates were not ties). The symbol *, **, and *** indicate significance at $p < 0.05$, 0.005, and 0.0005. Note: $p < 0.05/18 = 0.00278$ would be significant after Bonferroni correction for the 18 comparisons made in this table (MC).

Gene trees	Other method	# of replicates TQMC-n2	OTHER	TIE	Δ FN error	p-value		MC
Figure 4A – 0.5X species tree scale								
estimated	wQFM	11	2	7	0.75 ± 1.12	0.011006	*	N
estimated	FASTRAL	8	3	9	0.75 ± 1.48	0.0461	*	N
estimated	ASTRAL3	18	2	0	2.10 ± 1.86	0.00101	**	Y
true	wQFM	5	7	8	-0.05 ± 1.340	0.94		
true	FASTRAL	5	8	7	-0.20 ± 1.24	0.56		
true	ASTRAL3	9	7	4	0.15 ± 1.42	0.56		
Figure 4B – 1X species tree scale								
estimated	wQFM	5	4	11	0.05 ± 0.69	NA		
estimated	FASTRAL	9	3	8	0.70 ± 1.34	0.0333	*	N
estimated	ASTRAL3	12	1	7	1.1 ± 1.21	0.00293	**	N
true	wQFM	2	6	12	-0.25 ± 0.72	NA		
true	FASTRAL	2	7	11	-0.30 ± 0.92	NA		
true	ASTRAL3	4	5	11	-0.05 ± 0.89	NA		
Figure 4C – 2X species tree scale								
estimated	wQFM	4	5	11	0.00 ± 0.80	NA		
estimated	FASTRAL	7	5	8	0.20 ± 0.95	0.36		
estimated	ASTRAL3	7	5	8	0.15 ± 0.88	0.46		
true	wQFM	0	0	20	0.00 ± 0.00	NA		
true	FASTRAL	2	1	17	0.05 ± 0.39	NA		
true	ASTRAL3	2	1	17	0.05 ± 0.39	NA		

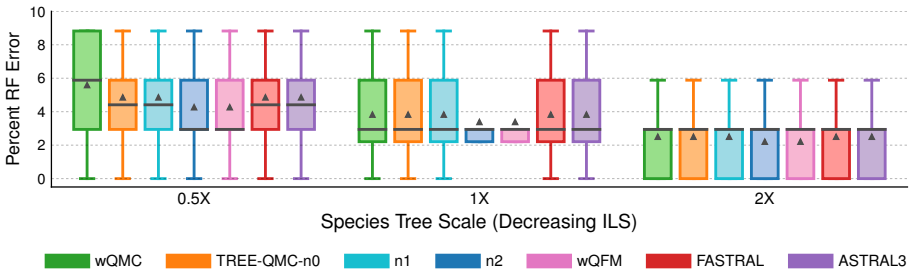


Figure S3: **Species tree error and runtime for mammalian simulated data set with varying ILS levels.** All data sets have 200 estimated gene trees (sequence length: 500). Percent species tree error across replicates is shown using box plots (bars represent medians; triangles represent means; some outliers are not shown).

Table S7: **Testing for differences between TREE-QMC-n2 and other methods on mammalian simulated data sets with varying species tree scales and thus ILS levels (Figure S3).** All data sets had 200 estimated gene trees (sequence length: 500).

Species tree scale	Other method	# of replicates TQMC-n2	OTHER	TIE	Δ FN error	p-value
0.5X	wQFM	0	0	20	0.00 ± 0.00	NA
0.5X	FASTRAL	5	2	13	0.20 ± 0.70	NA
0.5X	ASTRAL3	5	2	13	0.20 ± 0.70	NA
1X	wQFM	0	0	20	0.00 ± 0.00	NA
1X	FASTRAL	5	2	13	0.15 ± 0.59	NA
1X	ASTRAL3	5	2	13	0.15 ± 0.59	NA
2X	wQFM	0	0	20	0.00 ± 0.00	NA
1X	FASTRAL	3	1	16	0.10 ± 0.45	NA
2X	ASTRAL3	3	1	16	0.10 ± 0.45	NA

3.3 Sequence length and thus GTEE level

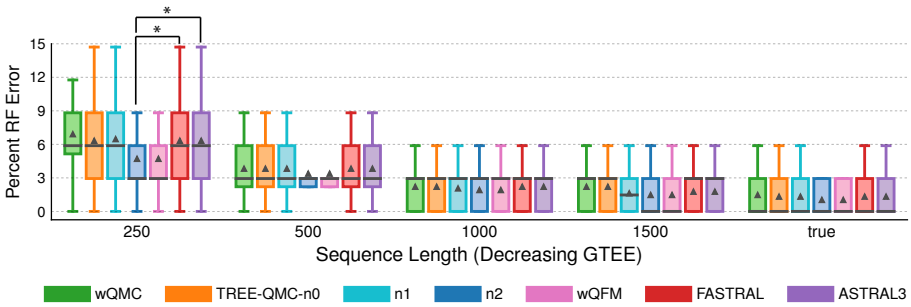


Figure S4: **Species tree error and runtime for mammalian simulated data set with varying sequence lengths and GTEE levels.** All data sets were simulated from unscaled species trees (1X) and have 200 gene trees (note: results for sequence length 500 are duplicated from Figure S3). Percent species tree error across replicates is shown using box plots (bars represent medians; triangles represent means; some outliers are not shown).

Table S8: **Testing for differences between TREE-QMC-n2 and other methods on mammalian simulated data sets with varying sequence lengths and thus gene tree estimation error (Figure S4).** All data sets were simulated with from unscaled species trees (1X) and have 200 gene trees. Note: rows with parentheses are duplicated from Table S7.

Sequence length	Other method	# of replicates TQMC-n2	OTHER	TIE	Δ FN error	p-value	
250	wQFM	0	0	20	0.00 ± 0.00	NA	
250	FASTRAL	11	4	5	0.55 ± 1.10	0.04	*
250	ASTRAL3	11	4	5	0.55 ± 1.10	0.04	*
(500)	wQFM	0	0	20	0.00 ± 0.00	NA	
(500)	FASTRAL	5	2	13	0.15 ± 0.59	NA	
(500)	ASTRAL3	5	2	13	0.15 ± 0.59	NA	
1000	wQFM	0	0	20	0.00 ± 0.00	NA	
1000	FASTRAL	2	0	18	0.10 ± 0.31	NA	
1000	ASTRAL3	2	0	18	0.10 ± 0.31	NA	
1500	wQFM	0	0	20	0.00 ± 0.00	NA	
1500	FASTRAL	3	1	16	0.10 ± 0.45	NA	
1500	ASTRAL3	3	1	16	0.10 ± 0.45	NA	
true	wQFM	0	0	20	0.00 ± 0.00	NA	
true	FASTRAL	3	1	16	0.10 ± 0.45	NA	
true	ASTRAL3	3	1	16	0.10 ± 0.45	NA	

3.4 Number of Gene Trees

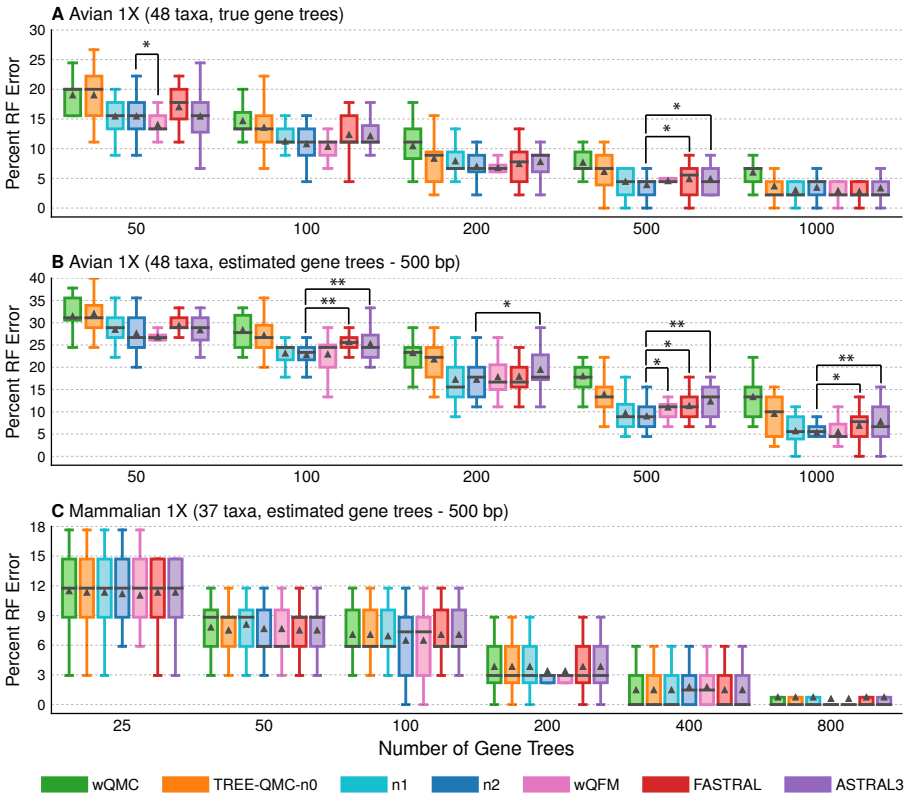


Figure S5: **Species tree error and runtime for avian and mammalian simulated data sets.** Bars represent medians, and triangles represent means (outliers are not shown). The symbols *, **, and *** indicate significance at $p < 0.05$, 0.005, and 0.0005, respectively (see Tables S9 and S10 for details). All data sets have unscaled species trees (1X). Note: the results for avian simulated data sets with 1000 true gene trees are duplicated from Figure 4A–C, and the results for mammalian simulated data sets with 200 estimated gene trees (sequence length: 500) are duplicated from Figures S3 and S4.

Table S9: **Testing for differences between TREE-QMC-n2 and other methods on avian simulated data sets with varying numbers of gene trees (Figure S5).** Note: rows with parentheses are duplicated from Table S6.

# of genes	Other method	# of replicates TQMC-n2	OTHER	TIE	Δ FN error	p-value	
Figure S5A – Avian 1X with true gene trees							
50	wQFM	2	10	8	-0.65 ± 1.18	0.03714	*
50	FASTRAL	11	5	4	0.70 ± 1.56	0.07448	
50	ASTRAL3	8	8	4	0.00 ± 1.41	1.00	
100	wQFM	7	6	7	-0.20 ± 1.24	0.47	
100	FASTRAL	9	3	8	0.70 ± 1.56	0.07745	
100	ASTRAL3	9	4	7	0.60 ± 1.31	0.06917	
200	wQFM	2	5	13	-0.10 ± 0.97	NA	
200	FASTRAL	8	5	7	0.20 ± 1.20	0.49	
200	ASTRAL3	9	4	7	0.35 ± 1.23	0.25	
500	wQFM	5	0	15	0.30 ± 0.57	NA	
500	FASTRAL	10	3	7	0.45 ± 0.89	0.04249	*
500	ASTRAL3	10	3	7	0.45 ± 0.89	0.04249	*
(1000)	wQFM	2	6	12	-0.25 ± 0.72	NA	
(1000)	FASTRAL	2	7	11	-0.30 ± 0.92	NA	
(1000)	ASTRAL3	4	5	11	-0.05 ± 0.89	NA	
Figure S5B – Avian 1X with estimated gene trees							
50	wQFM	3	8	9	-0.35 ± 1.66	0.47	
50	FASTRAL	11	5	4	0.80 ± 2.40	0.16	
50	ASTRAL3	8	6	6	0.35 ± 2.48	0.51	
100	wQFM	7	8	5	0.05 ± 1.67	0.93	
100	FASTRAL	13	1	6	1.30 ± 1.49	0.00202	**
100	ASTRAL3	13	2	5	1.10 ± 1.37	0.00276	**
200	wQFM	8	5	7	0.25 ± 1.41	0.39	
200	FASTRAL	9	6	5	0.30 ± 1.53	0.35	
200	ASTRAL3	11	4	5	1.00 ± 1.75	0.02670	*
500	wQFM	11	2	7	0.90 ± 1.45	0.01932	*
500	FASTRAL	11	3	6	1.05 ± 1.88	0.02633	*
500	ASTRAL3	15	2	3	1.50 ± 1.50	0.00112	**
(1000)	wQFM	5	4	11	0.05 ± 0.69	NA	
(1000)	FASTRAL	9	3	8	0.70 ± 1.34	0.03329	*
(1000)	ASTRAL3	12	1	7	1.10 ± 1.21	0.00293	**

Table S10: **Testing for differences between TREE-QMC-n2 and other methods on mammalian simulated data sets with varying numbers of estimated gene trees (Figure S5).** Note: rows with parentheses are duplicated from Tables S7 and S8.

# of genes	Other method	# of replicates TQMC-n2	OTHER	TIE	Δ FN error	p-value
Figure S5C – Mammalian 1X with estimated gene trees (500 bp)						
25	wQFM	0	1	19	-0.05 ± 0.22	NA
25	FASTRAL	4	5	11	0.05 ± 0.89	NA
25	ASTRAL3	4	5	11	0.05 ± 0.89	NA
50	wQFM	0	0	20	0.00 ± 0.00	NA
50	FASTRAL	6	5	9	-0.05 ± 0.95	0.81
50	ASTRAL3	6	5	9	-0.05 ± 0.95	0.81
100	wQFM	0	0	20	0.00 ± 0.00	NA
100	FASTRAL	7	3	10	0.20 ± 0.70	NA
100	ASTRAL3	7	3	10	0.20 ± 0.70	NA
(200)	wQFM	0	0	20	0.00 ± 0.00	NA
(200)	FASTRAL	5	2	13	0.15 ± 0.59	NA
(200)	ASTRAL3	5	2	13	0.15 ± 0.59	NA
400	wQFM	0	0	20	0.00 ± 0.00	NA
400	FASTRAL	0	2	18	-0.10 ± 0.31	NA
400	ASTRAL3	0	2	18	-0.10 ± 0.31	NA
800	wQFM	0	0	20	0.00 ± 0.00	NA
800	FASTRAL	1	0	19	0.05 ± 0.22	NA
800	ASTRAL3	1	0	19	0.05 ± 0.22	NA

3.5 Additional Results on Biological Data Sets

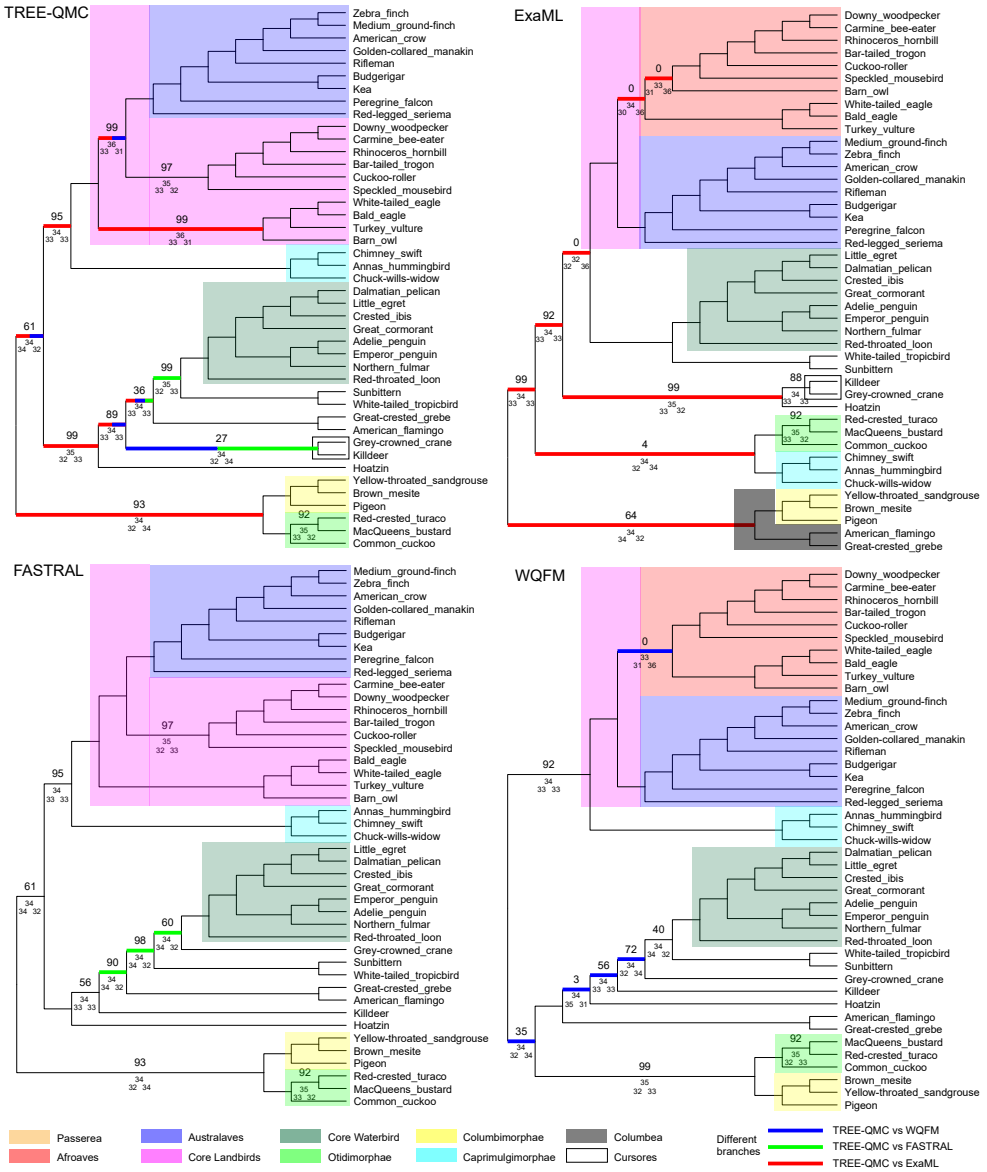


Figure S6: Species trees estimated on TENT data by TREE-QMC-n2, ExaML, FASTRAL, and wQFM. Above the branch, we show support values estimated using ASTRAL's local posterior probability (multiplied by 100). Only support values less than 100 are shown. Below the branch, we show the quartet support (the two values below it correspond to quartet support for the two alternative resolutions of the branch). Importantly, these support calculations are based on estimated gene trees. Taxa outside of Neoaves are not shown as all methods recovered the same topology outside of Neoaves.

4 Other Plots for ASTRAL-II Data Sets

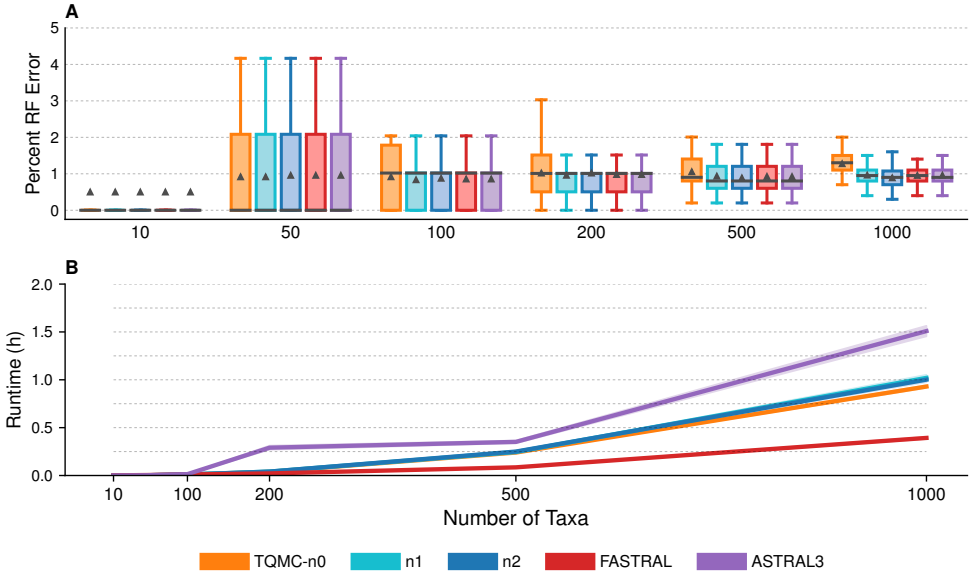


Figure S7: **Species tree error and runtime for ASTRAL-II data sets with varying number of taxa (1000 true gene trees).** (A) Percent species tree error across replicates (bars represent medians; triangles represent means; some outliers are not shown). No statistical tests performed. (B) Mean runtime across replicates (shaded region indicates standard error). All data sets were simulated with 1X species tree height and shallow speciation.

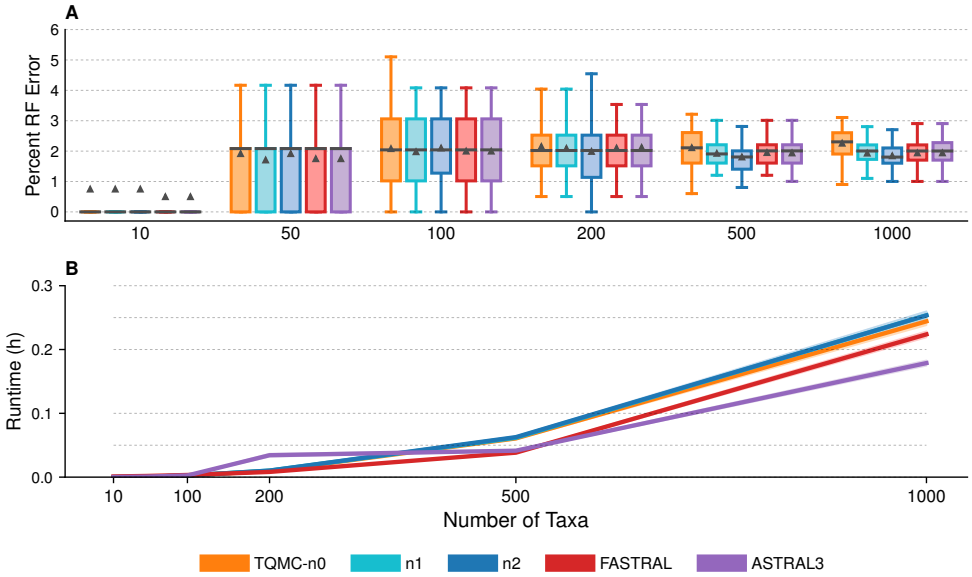


Figure S8: **Species tree error and runtime for ASTRAL-II data sets with varying numbers of taxa (250 true gene trees).** (A) Percent species tree error across replicates (bars represent medians; triangles represent means; some outliers are not shown). No statistical tests performed. (B) Mean runtime across replicates (shaded region indicates standard error). All data sets were simulated with 1X species tree height and shallow speciation.

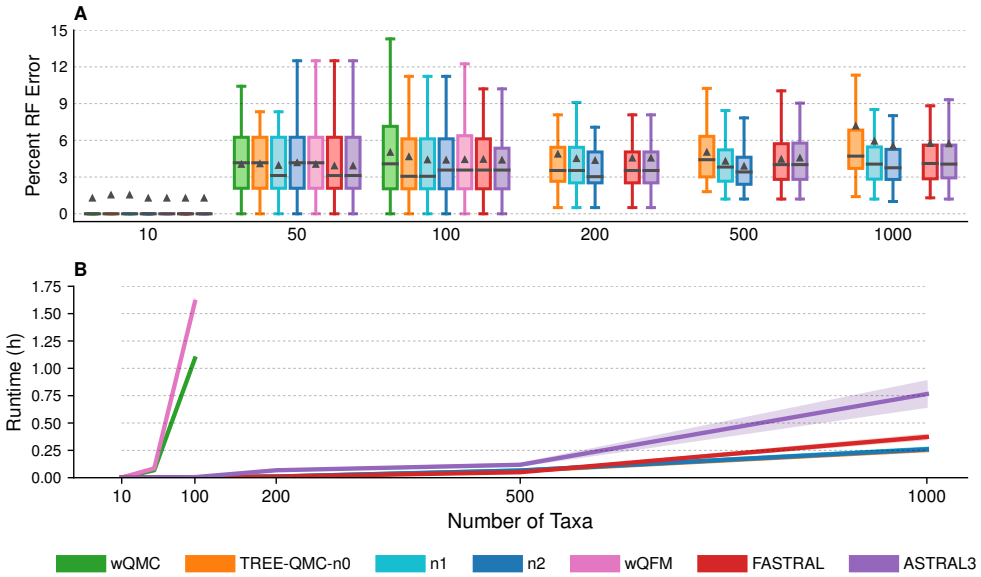


Figure S9: **Species tree error and runtime for ASTRAL-II data sets with varying numbers of taxa (250 estimated gene trees).** (A) Percent species tree error across replicates (bars represent medians; triangles represent means; some outliers are not shown). No statistical tests performed. (B) Mean runtime across replicates (shaded region indicates standard error). All data sets were simulated with 1X species tree height and shallow speciation.

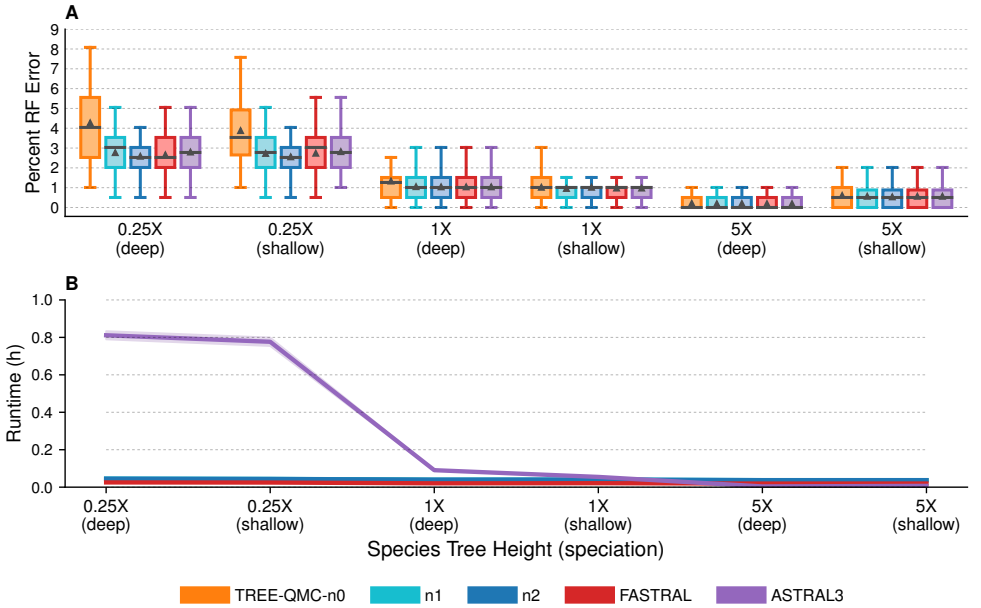


Figure S10: **Species tree error and runtime for ASTRAL-II data sets with varying ILS levels (1000 true gene trees).** (A) Percent species tree error across replicates (bars represent medians; triangles represent means; some outliers are not shown). No statistical tests performed. (B) Mean runtime across replicates (shaded region indicates standard error). All data sets have 200 taxa. One model condition with species tree height 1X and shallow speciation is repeated from Figure S7.

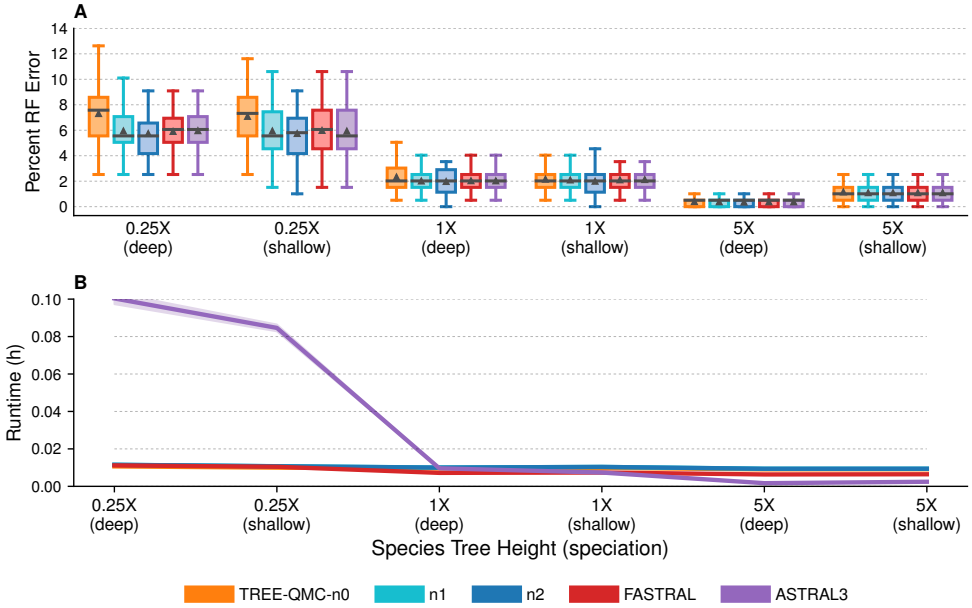


Figure S11: **Species tree error and runtime for ASTRAL-II data sets with varying ILS levels (250 true gene trees).** (A) Percent species tree error across replicates (bars represent medians; triangles represent means; some outliers are not shown). No statistical tests performed. (B) Mean runtime across replicates (shaded region indicates standard error). All data sets have 200 taxa. One model condition with species tree height 1X and shallow speciation is repeated from Figure S8.

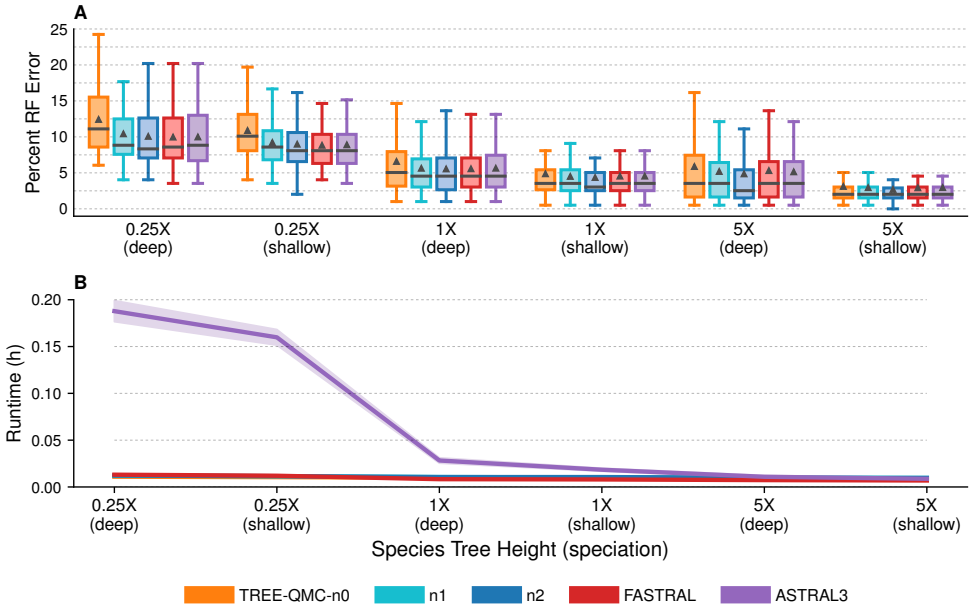


Figure S12: **Species tree error and runtime for ASTRAL-II data sets with varying ILS levels (250 estimated gene trees).** (A) Percent species tree error across replicates (bars represent medians; triangles represent means; some outliers are not shown). No statistical tests performed. (B) Mean runtime across replicates (shaded region indicates standard error). All data sets have 200 taxa. One model condition with species tree height 1X and shallow speciation is repeated from Figure S9.

5 TREE-QMC Algorithm

5.1 Quartet Graph Construction Case 1 cont. (Two Singletons)

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 vertex 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 1, in which we show that the total weight of selecting two uniquely labeled leaves below vertex u equals $g_0[u]$. All other quantities $(p, \mathbb{A}, \mathbb{L}, \mathbb{R})$ are computed from $g_0[u]$.

Lemma 1. *The total weight of all taxon pairs in the subtree rooted at internal vertex u*

$$\sum_{\substack{z,w \in L(u): \\ Z \neq W}} I(z)I(w) = g_0[u] \tag{1}$$

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

Proof. Let $\mathcal{S}(u)$ be the set of singletons below vertex 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 2.

Lemma 2. *Let T be a multi-labeled gene tree T, and let X, Y be singletons. Then,*

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

where r is the root of T.

Proof.

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

□

5.2 Quartet Graph Construction Case 2 (Singleton and Artificial Taxon)

In this section, we consider the computation of $\mathbb{B}[X,Y]$ when X is a singleton and Y is an artificial taxon (or vice versa). Our previous algorithm computed $\mathbb{B}[X,Y]$ when it reached the lowest common ancestor (LCA) X and Y during a postorder traversal. However, there are now multiple multiple leaves labeled Y and thus there can be multiple LCAs. Alternatively, we might count the number of quartets with X and Y as siblings that have vertex v as their LCA, meaning that X labels a leaf below vertex $v.left$ and Y labels a leaf below vertex $v.right$ or vice versa. We let $\mathbb{B}_v[X,Y]$ denote the first quantity and $\mathbb{B}_v[Y,X]$ denote the second quantity, defining $\mathbb{A}_v[X,Y]$, $\mathbb{L}_v[X,Y]$, and $\mathbb{R}_v[X,Y]$ in the natural way. Now the total number of quartets with X,Y as siblings can be computed as

$$\mathbb{B}[X,Y]=\mathbb{B}[Y,X]=\sum_{v\in V(T)}\big(\mathbb{B}_v[X,Y]+\mathbb{B}_v[Y,X]\big) \tag{3}$$

Without loss of generality, we show how to compute $\mathbb{B}_v[X,Y]$, where X is a singleton and Y is an artificial taxon. Let's assume this quantity is not zero so X labels *exactly one* leaf below $v.left$ and Y labels *at least one* leaf below $v.right$. In our previous algorithm, we utilized an approach for counting number of ways to select two z,w below vertex u such that $Z\neq W$. Now we must additionally require $Z\neq Y$ because Y already labels a leaf in the quartet. This gives us the modified binomial coefficient

$$g_Y[u]=\binom{c_0[u]}{2}+\Big(c_0[u]\cdot(G_1[u]-c_Y[u])\Big)+\left(\frac{(G_1[u]-c_Y[u])^2-(G_2[u]-c_Y[u]^2)}{2}\right) \tag{4}$$

from which can compute the modified prefix of v: $p_Y[v]=p_Y[v.parent]+g_Y[v.sibling]$. Now we compute a vector g of size $b+1$ and a vector p of size b but note that the time complexity of preprocessing phase is still $O(bn)$. Afterward, we can compute the quantities:

$$\mathbb{A}_v[X,Y]=c_Y[v.right]\cdot g_Y[v.above] \tag{5}$$

$$\mathbb{L}_v[X,Y]=c_Y[v.right]\cdot(p_Y[x]-p_Y[v.left]) \tag{6}$$

These are the same formulas given in the appendix, except that we now exclude Y when taking the modified binomial coefficient and then multiply by $c_Y[v.right]$ (because we can select any leaf labeled Y below $v.right$).

Computing $\mathbb{R}_v[X,Y]$ is more complicated, and thus, we define a *different* dynamic programming algorithm that counts triplets of the form $((W,Z),Y)$. Specifically, we define

$$\mathbb{R}_v[X,Y]=f_{Y,0}[v.right] \tag{7}$$

where $f_{Y,0}[u]$ is the number of triplets below vertex $u=v.right$ of the form $((W,Z),Y)$. This quantity can be computed in $O(n)$ time a preorder traversal, after initializing $f_{y,0}[leaf]=0$, as follows:

$$f_{Y,0}[u]=f_{Y,0}[u.left]+f_{Y,0}[u.right]+\Big(c_Y[u.left]\cdot g_Y[u.right]\Big)+\Big(c_Y[u.right]\cdot g_Y[u.left]\Big) \tag{8}$$

where the first and second term come from selecting three leaves below $u.left$ and $u.right$, respectively (these triplets have already been counted). The last two terms come from counting triplets whose LCA is u. We repeat this preprocessing to compute f for all b artificial taxa; thus, the total time complexity of the preprocessing phase is still $O(bn)$.

After the preprocessing phase, we can compute $\mathbb{B}_v[X, Y]$ in constant time (a similar argument can be made for $\mathbb{B}_v[Y, X]$). Both of these quantities will equal zero (and do not need to be computed) for vertices that are not on the path from the singleton X to the root; thus, we can compute $\mathbb{B}[X, Y]$ in $O(n)$ time. Repeating this calculation for all ways of selecting one singleton and one artificial taxon gives us the final time complexity: $O(abh + bn)$.

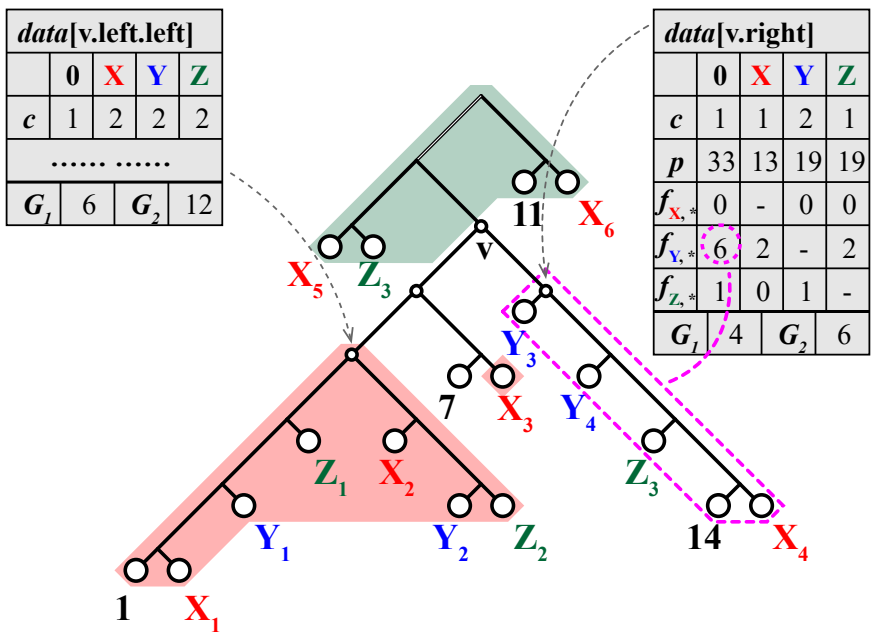


Figure S13: $\mathbb{B}_v$ **with one artificial taxon.** Above is an example of the data stored to compute $\mathbb{B}_v[7, Y]$, where 7 is a singleton in $v.left$ and Y is an artificial taxa in $v.right$. $\mathbb{B}_v[7, Y] = \mathbb{A}_v[7, Y] + \mathbb{L}_v[7, Y] + \mathbb{R}_v[7, Y] = 2 \cdot 5 + 2 \cdot 8 + 6 = 32$.

5.3 Quartet Graph Construction Case 3 (Two Artificial Taxa)

Lastly, we consider the computation of $\mathbb{B}[X, Y]$ when X and Y are both artificial taxa. This largely proceeds as previously described in the previous Section. First, we update the binomial coefficient to exclude both X and Y (instead of just Y) as follows:

$$g_{X,Y}[v] = \binom{c_0[v]}{2} + \left(c_0[v] \cdot (G_1[v] - c_Y[v] - c_X[v])\right) + \left(\frac{(G_1[v] - c_Y[v] - c_X[v])^2 - (G_2[v] - c_Y[v]^2 - c_X[v]^2)}{2}\right)$$

Second, we update the quantity for counting triplets

$$f_{Y,X}[v] = f_{Y,X}[v.left] + f_{Y,X}[v.right] + c_Y[v.left] \cdot g_{X,Y}[v.right] + c_Y[v.right] \cdot g_{X,Y}[v.left] \tag{9}$$

Third, we update quantities associated with $\mathbb{B}_v[X, Y]$, specifically

$$\mathbb{A}_v[X, Y] = c_X[v.left] \cdot c_Y[v.right] \cdot g_{X,Y}[v.above] \tag{10}$$
$$\mathbb{L}_v[X, Y] = c_Y[v.right] \cdot f_{X,Y}[v.left] \tag{11}$$
$$\mathbb{R}_v[X, Y] = c_X[v.left] \cdot f_{Y,X}[v.right] \tag{12}$$

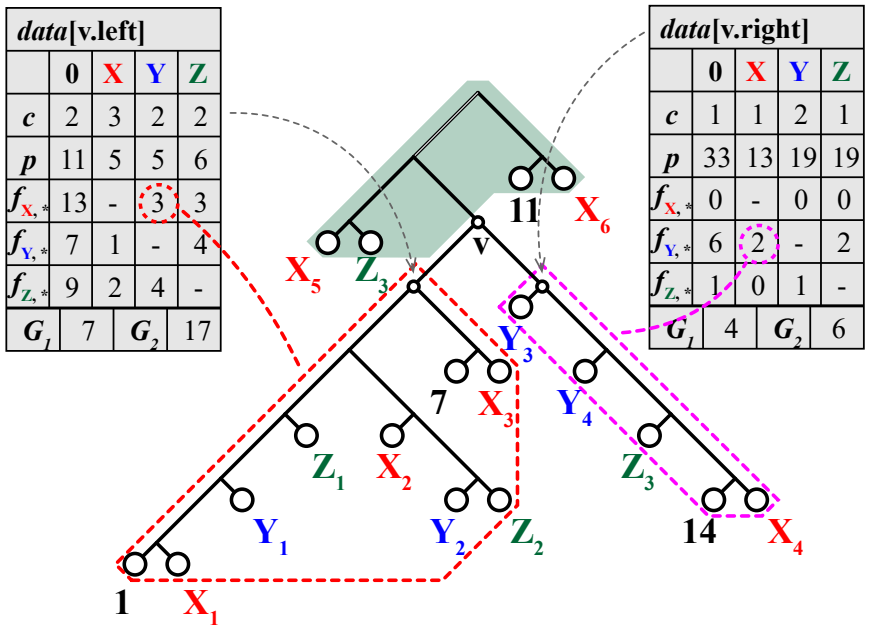


Figure S14: $\mathbb{B}_v$ **with two artificial taxa.** In subfigure b), we show how to compute $\mathbb{B}_v[X,Y]$ where both X and Y are artificial. Again, $\mathbb{B}_v[X,Y]$ is given by the sum $\mathbb{A}_v[X,Y]$, $\mathbb{L}_v[X,Y]$, and $\mathbb{R}_v[X,Y]$. $\mathbb{A}_v[X,Y]$ is still derived from the green subtree above v except that not only Y but also X is excluded. Consequently, $g_{X,Y}[v.above] = 1$ since there are only two taxa in the subtree except X, and $\mathbb{A}_v[X,Y]$ is therefore $c_X[v.right] \cdot c_Y[v.right] \cdot g_{X,Y}[v.above] = 6$. On the other hand, $\mathbb{L}_v[X,Y]$ is $c_Y[v.right] \cdot f_{X,Y}[v.left] = 2 \times 3 = 6$ while $\mathbb{R}_v[X,Y] = 3 \times 2 = 6$ because $c_X[v.left] = 3$ and $f_{Y,X}[v.right] = 2$. Finally, we have $\mathbb{B}_v[X,Y] = 6+6+6 = 18$. Similarly, $\mathbb{B}_v[Y,X] = 2+1+0 = 3$ because $c_Y[v.left] = 2$, $c_X[v.right] = 1$, $g_{X,Y}[v.above] = 1$, $f_{X,Y}[v.right] = 0$ and $f_{Y,X}[v.left] = 1$.

The key thing to note is that we are now storing a matrix of size b^2 at each vertex v. This brings the preprocessing phase to $O(b^2n)$, after which we can compute $\mathbb{B}_v[X,Y]$ in constant time. This quantity could be non-zero for any internal vertex of T so the total time to compute $\mathbb{B}[X,Y]$ is once again $O(n)$. We repeat this calculation for all ways of selecting one two artificial taxa to get the total time complexity is $O(b^2n)$.

5.4 Algorithms

Algorithm 1: Initialization

Data: the input gene tree T and pre-computed $c[v]$ and $g[v]$ values
Result: the $p[v]$ and $f[v]$ values for each node v in T

```
for each node v of T in the pre-order do
    if v is the root then
        p_0[v] ← 0
        for each artificial taxa X do
            p_X[v] ← 0
    else
        if v is not a leaf then
            p_0[v] ← p_0[v.parent] + g_0[v.sibling]
            for each artificial taxa X do
                p_X[v] ← p_X[v.parent] + g_X[v.sibling]
for each node v of T in the post-order do
    if v is a leaf then
        for each artificial taxa Y do
            f_{Y,0}[v] ← 0
        for each artificial taxa Y do
            for each artificial taxa X ≠ Y do
                f_{Y,X}[v] ← 0
    else
        for each artificial taxa Y do
            f_{Y,0}[v] ← f_{Y,0}[v.left] + f_{Y,0}[v.right] + c_Y[v.left] · g_Y[v.right] + c_Y[v.right] · g_Y[v.left]
        for each artificial taxa Y do
            for each artificial taxa X ≠ Y do
                f_{Y,X}[v] ←
                  f_{Y,X}[v.left] + f_{Y,X}[v.right] + c_Y[v.left] · g_{X,Y}[v.right] + c_Y[v.right] · g_{X,Y}[v.left]
```

Algorithm 2: Quartet Graph Construction – Bad Edges

Data: the input gene tree T with all pre-computed values
Result: $\mathbb{B}[X,Y]$: the weights of the bad edges between each pair of X and Y

```
for each internal node v in T do
    for each taxa X below v.left do
        for each taxa Y below v.right such that X ≠ Y do
            if X is an artificial taxa then
                if Y is an artificial taxa then
                    𝔸_v[X,Y] ← c_X[v.left] · c_Y[v.right] · g_{X,Y}[v.above]
                    𝕃_v[X,Y] ← c_Y[v.right] · f_{X,Y}[v.left]
                    ℝ_v[X,Y] ← c_X[v.left] · f_{Y,X}[v.right]
                else
                    𝔸_v[X,Y] ← c_Y[v.right] · g_Y[v.above]
                    𝕃_v[X,Y] ← c_Y[v.right] · (p_Y[x] − p_Y[v.left]) where y is the unique leaf labeled Y
                    ℝ_v[X,Y] ← f_{Y,0}[v.right]
            else
                if Y is an artificial taxa then
                    𝔸_v[X,Y] ← c_X[v.left] · g_X[v.above]
                    𝕃_v[X,Y] ← f_{X,0}[v.left]
                    ℝ_v[X,Y] ← c_X[v.left] · (p_X[y] − p_X[v.right])
                else
                    𝔸_v[X,Y] ← g_0[v.above]
                    𝕃_v[X,Y] ← p_0[x] − p_0[v.left]
                    ℝ_v[X,Y] ← p_0[y] − p_0[v.right]
            𝔹[X,Y] ← 𝔹[X,Y] + 𝔸_v[X,Y] + 𝕃_v[X,Y] + ℝ_v[X,Y]
            𝔹[X,Y] ← 𝔹[X,Y] + 𝔸_v[X,Y] + 𝕃_v[X,Y] + ℝ_v[X,Y]
```

Algorithm 3: Quartet Graph Construction – Good Edges

Data: the input gene tree T with root r, all pre-computed values for T, and $\mathbb{B}$
Result: $\mathbb{G}[X,Y]$: the weights of the good edges between each pair of X and Y

```
for each taxa X do
    for each taxa Y do
        if X is an artificial taxa then
            if Y is an artificial taxa then
                𝔾[X,Y] = g_{X,Y}[r] · c_X[r] · c_Y[r] − 𝔹[X,Y]
            else
                𝔾[X,Y] = ( C(c_0[r]−1, 2) + ( (c_0[r] − 1) · (G_1[v] − c_X[v]) ) +
                    ( ((G_1[r]−c_X[r])^2 − (G_2[r]−c_X[r]^2)) / 2 ) ) · c_X[r] − 𝔹[X,Y]
        else
            if Y is an artificial taxa then
                𝔾[X,Y] = ( C(c_0[r]−1, 2) + ( (c_0[r] − 1) · (G_1[r] − c_Y[r]) ) +
                    ( ((G_1[r]−c_Y[r])^2 − (G_2[r]−c_Y[r]^2)) / 2 ) ) · c_Y[r] − 𝔹[X,Y]
            else
                𝔾[X,Y] = C(c_0[r]−2, 2) + (c_0[r] − 2) · G_1[r] + (G_1[r]^2 − G_2[r]) / 2 − 𝔹[X,Y]
```

5.5 Time Complexity

Theorem 1. *Let n leaves in the gene tree T, and let s be the number of leaf labels for the current subproblem. Then, we compute $\mathbb{B}$ and $\mathbb{G}$ in $O(s^2n)$ time using Algorithms 1–3.*

Proof. Let a be the number of singletons in the subproblem, and let b be the number of artificial taxa (so $a + b = s$). To compute $\mathbb{B}$, we need to compute $\mathbb{A}, \mathbb{L}, \mathbb{R}$, which in turn depend on other quantities that are computed during a preprocessing phase.

Preprocessing phase: Computing the **number of taxa** below vertex v:

- $c_0[v]$ takes $O(n)$ time
- $c_Y[v]$ for each artificial taxon Y takes $O(bn)$ time
- all $G_1[v]$ and $G_2[v]$ takes $O(bn)$ time

via a post-order traversal on T. Computing the **modified binomial coefficients**:

- all $g_0[v]$ takes $O(n)$ time
- all $g_Y[v]$ for each artificial taxon Y takes $O(bn)$ time
- all $g_{X,Y}[v]$ for each pair X, Y of artificial taxa takes $O(b^2n)$ time

using $G_1[v]$ and $G_2[v]$. Computing the **prefixes**

- $p_0[v]$ takes $O(n)$ time
- $p_X[v]$ for each artificial taxon X takes $O(bn)$ time

see lines 1-10 of Algorithm 1. Computing the **triplet counters**

- $f_{Y,0}[v]$ for each artificial taxon Y takes $O(bn)$ time
- $f_{Y,X}[v]$ for each pair X, Y of artificial taxa takes $O(b^2n)$ time

see lines 11-23 of Algorithm 1. To sum up, the total initialization time is $O(b^2n)$ for each gene tree.

Now we can compute the weights of the **bad edges** $\mathbb{B}$ via Algorithm 2. Each iteration of Algorithm 2 (lines 4-23) costs $O(1)$ time. In other words, given any triple of v, X, and Y, the corresponding $\mathbb{B}_v[X, Y]$ is computed in constant time. Consequently, the total time complexity depends on the number of the distinct tripples (v, X, Y). If both X and Y are singletons, then there is only one unique v at which $\mathbb{B}[X, Y]$ or $\mathbb{B}[Y, X]$ are updated because v must be the LCA of X and Y. As a result, the triplet of (v, X, Y) is unique when X and Y are singleton, and the number of triplets for all singleton pairs of X and Y is $O(a^2)$. If X is a singleton and Y is an artificial taxon, the number of v is $O(h)$, where h is the height of T, because v must be one of the ancestors of X. In this case, the number of the triplets is $O(abh)$. The same holds in the case where Y is a singleton and X is an artificial taxa. If both X and Y are artificial taxa, then the number of v is $O(n)$ since v could be any inner node in the tree, and hence the number of triplets is $O(b^2n)$. To sum up, the total number of (v, X, Y), as well as the time complexity of Algorithm 2, is $O(a^2 + abh + b^2n) = O(s^2n)$. In the last step, we compute the weights of the **good edges** $\mathbb{G}$. We can update each $\mathbb{G}$ in $O(s^2)$ time using Algorithm 3. Thus, the time complexity is $O(s^2)$.

To conclude, the total time complexity is $O(s^2n)$. □

Theorem 2. *Let n be the number of species and let k be the number of gene trees. TREE-QMC has time complexity $O(n^3k)$ if subproblems are produced in a perfectly balanced fashion.*

Proof. At each step in the divide phase of the algorithm, we compute a bipartition and then recurse on two subproblems. Each subproblem is defined by taking the taxa on one side of the bipartition and adding an artificial taxon to represent the taxa on the other side of the bipartition. Thus, a subproblem of size $2^p + 2$ is divided into two subproblems of size $\frac{2^p+2}{2} + 1 = 2^{p-1} + 2$ if the division is balanced. The work f per subproblem has time complexity $O(s^3 + s^2nk)$ where the first term comes from applying the heuristic for

max cut and the second term comes from constructing the quartet graph by applying our algorithm to all gene trees (Theorem 1). Therefore,

$$
\begin{aligned}
T(2^p+2) =& \sum_{i=0}^{p} 2^i \cdot f(2^{p-i}+2) \quad \text{where } f \text{ is } O(s^3+s^2nk) \\
=& \sum_{i=0}^{p} 2^i \cdot [(2^{p-i}+2)^3 + (2^{p-i}+2)^2 \cdot (2^p+2) \cdot k] \\
=& \sum_{i=0}^{p} 2^i \cdot (2^{p-i}+2)^3 + \sum_{i=0}^{p} 2^i \cdot (2^{p-i}+2)^2 \cdot (2^p+2) \cdot k \\
=& \sum_{i=0}^{p} 2^i \cdot (2^{p-i}+2)^3 + (2^p+2) \cdot k \cdot \sum_{i=0}^{p} 2^i \cdot (2^{p-i}+2)^2 \\
=& \sum_{i=0}^{p} 2^i \cdot (2^{3(p-i)} + 3 \cdot 2^{2(p-i)+1} + 3 \cdot 2^{p-i+2} + 2^3) + (2^p+2) \cdot k \cdot \sum_{i=0}^{p} 2^i \cdot (2^{2(p-i)} + 2^{p-i+2} + 2^2) \\
=& \sum_{i=0}^{p} (2^{3(p-i)+i} + 3 \cdot 2^{2(p-i)+1+i} + 3 \cdot 2^{p-i+2+i} + 2^{3+i}) + (2^p+2) \cdot k \cdot \sum_{i=0}^{p} (2^{2(p-i)+i} + 2^{p-i+2+i} + 2^{2+i}) \\
=& \Big(\frac{1}{3} \cdot 2^p \cdot (4^p-1) + 3 \cdot 2^{p+1} \cdot (2^{p+1}-1) + 12 \cdot (2^{p+1}-1) + 2^{p+4} - 8\Big) \\
& + (2^p+2) \cdot k \cdot \big(2^p \cdot (2^{p+1}-1) + 2^{p+2} \cdot (p+1) + 2^{p+3} - 4\big) \\
=& O(2^{3p} + 2^p \cdot k \cdot 2^{2p}) \\
=& O(k \cdot 2^{3p}) \\
=& O(kn^3)
\end{aligned}
$$

□

5.6 Correctness

Lemma 3. *Let v be an internal node in T, let X be a taxon below $v.left$, and let Y, Z, and W be three taxa below $v.right$. The total weight of quartets $\{X, Y, Z, W\}$ in which X and Y are siblings, denoted by $\mathbb{R}_v[X, Y]$, is*

$$
\begin{cases}
c_X[v.left] \cdot f_{Y,X}[v.right] & \text{if both } X \text{ and } Y \text{ are artificial} \\
f_{Y,0}[v.right] & \text{if only } Y \text{ is artificial} \\
c_x[v.left] \cdot (p_X[y] - p_x[v.right]) & \text{if only } X \text{ is artificial} \\
p_0[y] - p_0[v.right] & \text{if both } X \text{ and } Y \text{ are singletons}
\end{cases}
\tag{13}
$$

where the values of $p_0[v]$, $p_X[v]$, $f_{Y,0}[v]$, and $f_{Y,X}[v]$ are given by Algorithm 1. Similarly, when Z and W are from $v.left$, the total weight, denoted by $\mathbb{L}_v[X, Y]$, is

$$
\begin{cases}
c_y[v.right] \cdot f_{X,Y}[v.left] & \text{if both } X \text{ and } Y \text{ are artificial} \\
f_{X,0}[v.left] & \text{if only } Y \text{ is artificial} \\
c_Y[v.right] \cdot (p_Y[x] - p_Y[v.left]) & \text{if only } X \text{ is artificial} \\
p_0[x] - p_0[v.left] & \text{if both } X \text{ and } Y \text{ are singletons}
\end{cases}
\tag{14}
$$

Finally, when Z and W are from $v.above$, the total weight, denoted by $\mathbb{A}_v[X, Y]$, is

$$
\begin{cases}
c_X[v.left] \cdot c_Y[v.right] \cdot g_{X,Y}[v.above] & \text{if both } X \text{ and } Y \text{ are artificial} \\
c_Y[v.left] \cdot g_Y[v.above] & \text{if only } Y \text{ is artificial} \\
c_X[v.right] \cdot g_X[v.above] & \text{if only } X \text{ is artificial} \\
g_0[v.above] & \text{if both } X \text{ and } Y \text{ are singletons}
\end{cases}
\tag{15}
$$

Proof. Note: capital letters are used to denote the label of a leaf. For example, leaf m would have label M in the current subproblem.

The proof for $\mathbb{A}_v[X,Y]$ is simple since there is only one subtree above v. We prove the case in which both X and X are artificial,

$$\begin{aligned}
\mathbb{A}_v[X,Y] &= \sum_{\substack{m\in L(v.left):\\ M=X}} \sum_{\substack{n\in L(v.right)\\ N=Y}} \sum_{\substack{z,w\in L(v.above):\\ Z\neq Y\neq X\neq W}} I(m)I(n)I(z)I(w) \\
&= \sum_{\substack{m\in L(v.left):\\ M=X}} I(m)\cdot\left(\sum_{\substack{n\in L(v.right):\\ N=Y}} I(n)\cdot\left(\sum_{\substack{z,w\in L(v.above):\\ Z\neq W\neq X\neq Y}} I(z)I(w)\right)\right) \\
&= c_X[v.left]\cdot c_Y[v.right]\cdot g_{X,Y}[v.above],
\end{aligned}$$

These equations are for artificial taxa X,Y but we can allow X to be a singleton by setting $c_X[u]=I(x)=1$ and decreasing $c_0[u]$ by $I(x)=1$ when computing $g_{X,Y}[u]$ (and similarly to allow Y to be a singleton). The other three cases of $\mathbb{A}_v[X,Y]$ are similar. In the following, we focus on $\mathbb{R}_v[X,Y]$ because the proof for $\mathbb{L}_v[x,y]$ is the same due to the symmetry. Recall that X is a taxon below $v.left$, and Y, Z, and W be three taxa below $v.right$.

If both X and Y are singletons, the prefix-sum technique is applied:

$$\begin{aligned}
\mathbb{R}_v[X,Y] &= \sum_{\substack{z,w\in L(v.right):\\ Z\neq W\neq Y\\ q(x,y,z,w)=x,y|z,w}} I(x)I(y)I(z)I(w) \\
&= \sum_{\substack{z,w\in L(v.right):\\ Z\neq W\neq Y\\ q(x,y,z,w)=x,y|z,w}} I(z)I(w) \\
&= \sum_{u\in\mathcal{P}(v.right,y)} \sum_{\substack{z,w\in L(u)\\ Z\neq W}} I(z)I(w) \\
&= \sum_{u\in\mathcal{P}(v.right,y)} g_0[u] \\
&= \sum_{u\in\mathcal{P}(root,y)} g_0[u] - \sum_{u\in\mathcal{P}(root,v.right)} g_0[u] \\
&= p_0[y]-p_0[v.right],
\end{aligned}$$

where $\mathcal{P}(i,j)$ is the set of the subtrees adjacent to the path from leaves i (included) to j (excluded). Note that $g_0[u]$ is used to compute $\sum_{z,w\in\mathcal{L}(u),Z\neq W} I(z)I(w)$ according to Lemma 1 in the main text.

If X is a singleton while Y is an artificial taxon, there could be multiple leaves labeled by Y in $L(v.right)$. Consequently, Y must be excluded when Z and W are selected, and thus $g_Y[u]$ is used instead of $g_0[u]$:

$$\begin{aligned}
\mathbb{R}_v[X,Y] &= \sum_{\substack{n\in L(v.right):\\ N=Y}} \sum_{\substack{z,w\in L(v.right):\\ Z\neq W\neq Y\\ q(x,n,z,w)=x,n|z,w}} I(x)I(n)I(z)I(w) \\
&= \sum_{\substack{n\in L(v.right):\\ N=Y}} I(n)\cdot\left(\sum_{u\in\mathcal{P}(v.right,n)} \sum_{\substack{z,w\in L(u):\\ Z\neq W\neq Y}} I(z)I(w)\right) \\
&= \sum_{\substack{n\in L(v.right):\\ N=Y}} I(n)\cdot \sum_{u\in\mathcal{P}(v.right,n)} g_Y[u] \\
&= f_{Y,0}[v.right]
\end{aligned}$$

Algorithm 1 computes $f_{Y,0}[v]$ according to the following recurrence:

$$f_{Y,0}[v] = \begin{cases} 0 & \text{if } v \text{ is a leaf} \\ f_{Y,0}[v.left] + f_{Y,0}[v.right] & \\ +c_Y[v.left] \cdot g_Y[v.right] + c_Y[v.right] \cdot g_Y[v.left] & \text{if } v \text{ is an inner node} \end{cases}.$$

The correctness of the recurrence is shown as follows.

$$\begin{aligned}
f_{Y,0}[v] &= \sum_{\substack{n \in L(v): \\ N=Y}} I(n) \cdot \sum_{u \in \mathcal{P}(v,n)} g_Y[u] \\
&= \sum_{\substack{n \in L(v.left): \\ N=Y}} I(n) \sum_{u \in \mathcal{P}(v,n)} g_y[u] + \sum_{\substack{n \in L(v.right): \\ N=Y}} I(n) \sum_{u \in \mathcal{P}(v,n)} g_Y[u] \\
&= \sum_{\substack{n \in L(v.left): \\ N=Y}} I(n) \cdot \left(g_Y[v.right] + \sum_{u \in \mathcal{P}(v.left,n)} g_y[u] \right) \\
&\quad + \sum_{\substack{n \in L(v.right): \\ N=Y}} I(n) \cdot \left(g_Y[v.left] + \sum_{u \in \mathcal{P}(v.right,n)} g_Y[u] \right) \\
&= \sum_{\substack{n \in L(v.left): \\ N=Y}} I(n) \cdot \sum_{u \in \mathcal{P}(v.left,n)} g_Y[u] + \sum_{\substack{n \in L(v.right): \\ N=Y}} I(n) \cdot \sum_{u \in \mathcal{P}(v.right,n)} g_Y[u] \\
&\quad + g_Y[v.right] \cdot \sum_{\substack{n \in L(v.left): \\ N=Y}} I(n) + g_Y[v.left] \cdot \sum_{\substack{n \in L(v.right): \\ N=Y}} I(n) \\
&= f_{Y,0}[v.left] + f_{Y,0}[v.right] + c_Y[v.left] \cdot g_Y[v.right] + c_Y[v.right] \cdot g_y[v.left].
\end{aligned}$$

If X is an artificial taxon and Y is a singleton, X could also appear in $L(v.right)$ so we must be excluded to select Z and W, and therefore $g_X[u]$ is used to compute the sum of $I(z)I(w)$ in a subtree.

$$\begin{aligned}
\mathbb{R}_v[x,y] &= \sum_{\substack{m \in L(v.left) \\ M=X}} \sum_{\substack{z,w \in L(v.right): \\ Z \neq W \neq X \\ q(m,y,z,w)=m,y|z,w}} I(m)I(y)I(z)I(w) \\
&= \sum_{\substack{m \in L(v.left): \\ M=X}} I(m) \sum_{u \in \mathcal{P}(v.right,y)} \sum_{\substack{z,w \in L(u): \\ Z \neq W \neq X}} I(z)I(w) \\
&= \sum_{\substack{m \in L(v.left): \\ M=X}} I(m) \cdot \left(\sum_{u \in \mathcal{P}(v.right,y)} \sum_{\substack{z,w \in L(u): \\ Z \neq W \neq X}} I(z)I(w) \right) \\
&= \sum_{\substack{m \in L(v.left): \\ M=X}} I(m) \cdot \left(\sum_{u \in \mathcal{P}(v.right,y)} g_x[u] \right) \\
&= \sum_{\substack{m \in L(v.left): \\ M=X}} I(m) \cdot \left(\sum_{u \in \mathcal{P}(root,y)} g_X[u] - \sum_{u \in \mathcal{P}(root,v.right)} g_X[u] \right) \\
&= \sum_{\substack{m \in L(v.left): \\ M=X}} I(m) \cdot (p_X[y] - p_X[v.right]) \\
&= c_X[v.left] \cdot (p_X[y] - p_X[v.right]).
\end{aligned}$$

Finally, if both X and Y are artificial taxa, $g_{X,Y}[u]$ is used.

$$\begin{aligned}
\mathbb{R}_v[X,Y] = & \sum_{\substack{m\in L(v.left):\\ M=X}} \sum_{\substack{N\in L(v.right):\\ n=Y}} \sum_{\substack{z,w\in L(v.right):\\ Z\neq W\neq X\neq Y\\ q(m,n,z,w)=m,n|z,w}} I(m)I(n)I(z)I(w) \\
= & \sum_{\substack{m\in L(v.left):\\ M=X}} I(m)\cdot\left(\sum_{\substack{n\in L(v.right):\\ N=Y}} I(n)\cdot\left(\sum_{u\in\mathcal{P}(v.right,n)} \sum_{\substack{z,w\in\mathcal{L}(u):\\ Z\neq W\neq X\neq Y}} I(z)I(w)\right)\right) \\
= & \sum_{\substack{m\in L(v.left):\\ M=X}} I(m)\cdot\left(\sum_{\substack{n\in L(v.right):\\ N=Y}} I(n)\cdot\left(\sum_{u\in\mathcal{P}(v.right,n)} g_{X,Y}[u]\right)\right) \\
= & \sum_{\substack{m\in L(v.left):\\ M=X}} I(m)\cdot f_{Y,X}[v.left] \\
= & c_X[v.left]\cdot f_{Y,X}[v.left]
\end{aligned}$$

Algorithm 1 computes $f_{Y,X}[v]$ according to the following recurrence:

$$f_{Y,X}[v] = \begin{cases} 0 & \text{if } v \text{ is a leaf} \\ f_{Y,X}[v.left] + f_{Y,X}[v.right] & \\ +c_Y[v.left]\cdot g_{X,Y}[v.right] + c_Y[v.right]\cdot g_{X,Y}[v.left] & \text{if } v \text{ is an inner node} \end{cases}$$

whose proof of correctness is similar to $f_{Y,0}[v]$, except that $g_{X,Y}[u]$ is used instead of $g_X[u]$. □

Lemma 4. *Let T be a binary tree, and let X and Y be taxa ($X \neq Y$). Then,*

$$\mathbb{B}[X,Y] = \sum_{v\in V(T)} \Big(\mathbb{A}_v[X,Y] + \mathbb{L}_v[X,Y] + \mathbb{R}_v[X,Y]\Big) + \Big(\mathbb{A}_v[Y,X] + \mathbb{L}_v[Y,X] + \mathbb{R}_v[Y,X]\Big)$$

where $V(T)$ is the set of internal nodes in gene tree T.

Proof. By definition, $\mathbb{B}[X,Y]$ is the total weight of the quartet in T whose topology is $X,Y|Z,W$, where Z

and W are two other taxa, so we have

$$
\begin{aligned}
\mathbb{B}[X,Y] &= \sum_{\substack{m\in L(T):\\ M=X}} \sum_{\substack{n\in L(T):\\ N=Y}} \sum_{\substack{z,w\in L(T):\\ Z\neq W\neq X\neq Y\\ q(n,m,z,w)=m,n|z,w}} I(m,n,z,w) \\
&= \sum_{v\in V(T)} \sum_{\substack{m\in L(v.left):\\ M=X}} \sum_{\substack{n\in L:(v.right)\\ N=Y}} \sum_{\substack{z,w\in L(T):\\ Z\neq W\neq X\neq Y\\ q(m,n,z,w)=m,n|z,w}} I(m,n,z,w) \\
&\quad + \sum_{v\in V(T)} \sum_{\substack{m\in L(v.right):\\ M=X}} \sum_{\substack{n\in L:(v.left)\\ N=Y}} \sum_{\substack{z,w\in \mathcal{L}(T):\\ Z\neq W\neq X\neq Y\\ q(m,n,z,w)=m,n|z,w}} I(m,n,z,w) \\
&= \sum_{v\in V(T)} \sum_{\substack{m\in L(v.left):\\ M=X}} \sum_{\substack{n\in L(v.right):\\ N=Y}} \Bigg(\sum_{\substack{z,w\in L(v.above):\\ Z\neq W\neq X\neq Y\\ q(n,m,z,w)=m,n|z,w}} I(m,n,z,w) \\
&\quad + \sum_{\substack{z,w\in L(v.left):\\ Z\neq W\neq X\neq Y\\ q(m,n,z,w)=m,n|z,w}} I(m,n,z,w) + \sum_{\substack{z,w\in L(v.right):\\ Z\neq W\neq X\neq Y\\ q(m,n,z,w)=m,n|z,w}} I(m,n,z,w) \Bigg) \\
&\quad + \sum_{v\in V(T)} \sum_{\substack{m\in L(v.right):\\ M=X}} \sum_{\substack{n\in L(v.left):\\ N=Y}} \Bigg(\sum_{\substack{z,w\in L(v.above):\\ Z\neq W\neq X\neq Y\\ q(m,n,z,w)=m,n|z,w}} I(m,n,z,w) \\
&\quad + \sum_{\substack{z,w\in L(v.left):\\ Z\neq W\neq X\neq Y\\ q(m,n,z,w)=m,n|z,w}} I(m,n,z,w) + \sum_{\substack{z,w\in L(v.right):\\ Z\neq W\neq X\neq Y\\ q(m,n,z,w)=m,n|z,w}} I(m,n,z,w) \Bigg)
\end{aligned}
$$

According to Lemma 3, this is

$$
\sum_{v\in V(T)} \Big(\mathbb{A}_v[X,Y] + \mathbb{L}_v[X,Y] + \mathbb{R}_v[X,Y] \Big) + \Big(\mathbb{A}_v[Y,X] + \mathbb{L}_v[Y,X] + \mathbb{R}_v[Y,X] \Big)
$$

□

The following theorem follows from the Lemma above.

Theorem 3. *Let T be a binary tree, and let X, Y be taxa ($X \neq Y$). Algorithm 2 computes $\mathbb{B}[X,Y]$ correctly for T.*

Lemma 5. *Let X and Y be taxa. Then,*

$$
\mathbb{G}[X,Y] + \mathbb{B}[X,Y] = g_{X,Y}[r] \cdot c_X[r] \cdot c_Y[r] \tag{16}
$$

where r is the root of gene tree T. These equations are for artificial taxa; however, they can be modified to accommodate singletons as follows (see Lemma 2 for details). If X is a singleton, set $c_X[r] = 1$ in the equation above. Then, compute $g_{X,Y}[r]$ by setting $c_X[r]$ to 0 and subtracting 1 from $c_0[r]$. If Y is a singleton, set $c_Y[r] = 1$ in the equation above. Then, compute $g_{X,Y}[r]$ by setting $c_Y[r]$ to 0 and subtracting 1 from $c_0[r]$.

Proof.

$$\begin{aligned}\mathbb{G}[X,Y]+\mathbb{B}[X,Y] &= \sum_{\substack{m,n,z,w\in L(r):\\ M=X,N=Y,\\ Z\neq W\neq X\neq Y}} I(m,n,z,w)\\ &= \sum_{\substack{m\in L(T):\\ M=X}} I(m)\cdot\left(\sum_{\substack{n\in L(T):\\ N=Y}} I(n)\cdot\left(\sum_{\substack{z,w\in L(T):\\ Z\neq W\neq X\neq Y}} I(z)I(w)\right)\right)\\ &= c_X[T]\cdot c_Y[T]\cdot g_{X,Y}[T].\end{aligned}$$

Note: capital letters are used to denote the label of a leaf. For example, leaf m would have label M in the current subproblem. □

The theorem below follows from the Theorem and Lemma above.

Theorem 4. *Let T be a binary tree, and let X, Y be taxa ($X \neq Y$). Algorithm 3 computes $\mathbb{G}[X,Y]$ correctly for T.*

References

Avni E, Cohen R, Snir S. 2014. Weighted quartets phylogenetics. *Syst Biol* **64**: 233–242.

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

Mahbub M, Wahab Z, Reaz R, Rahman MS, Bayzid MS. 2021. wQFM: highly accurate genome-scale species tree estimation from weighted quartets. *Bioinformatics* **37**: 3734–3743.

McGee M. 2018. Case for omitting tied observations in the two-sample t-test and the wilcoxon-mann-whitney test. *PLOS ONE* **13**: 1–19.

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