UNIVERSITY OF CALIFORNIA SAN DIEGO

Scalable algorithms for detecting boundaries and relationships of species from phylogenetic data

A dissertation submitted in partial satisfaction of the
requirements for the degree
Doctor of Philosophy

in

Computer Science

by

Maryam Rabiee Hashemi

Committee in charge:

Professor Siavash Mirarab, Chair
Professor Vineet Bafna, Co-Chair
Professor Melissa Gymrek
Professor Rob Knight
Professor Greg Rouse

2022

The dissertation of Maryam Rabiee Hashemi is approved, and it is acceptable in quality and form for publication on microfilm and electronically.

University of California San Diego

2022

DEDICATION

To my beloved family and friends for years of love and support.

EPIGRAPH

Once there was only dark.

If you ask me, the light's winning

—Rust Cohle

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

This dissertation concludes my Ph.D. journey at UC San Diego. Reflecting back upon the winding road behind, I feel indebted to many people who supported and inspired me along the way. I would like to thank my friends and family for being by my side throughout my Ph.D. They are the greatest supporters in my life.

I am additionally thankful to my colleagues in the Computer Science department, and all my labmates and collaborators who have been great to work with.

I would not be where I am today without the mentorship of Prof. Siavash Mirarab. I am grateful for his guidance and support, and I cannot thank him enough for all the time and effort he has taken to make my Ph.D. experience productive and intellectually stimulating, especially during the long sessions that we spent on problem solving together. He couldn't have trained me better for the skepticism and punctiliousness necessary for the scientific pursuit, by his professionalism and competency.

I would like to further thank the rest of my doctoral committee members, Professor Vineet Bafna, Professor Rob Knight, Professor Greg Rouse, and Professor Melissa Gymrek for their valuable suggestions and advice during my doctoral study.

Chapter 2, in full, is a reprint of the material as it appears in "INSTRAL: Discordance-aware Phylogenetic Placement using Quartet Scores" (2020). Rabiee, Maryam; Mirarab, Siavash, Systematic Biology, 69(2), 384-391. The dissertation author was the primary investigator and first author of this paper.

Chapter 3, in full, is a reprint of the material as it appears in "Forcing external constraints on tree inference using ASTRAL" (2020). Rabiee, Maryam; Mirarab, Siavash, BMC genomics, 21(2), 1-13. The dissertation author was the primary investigator and first author of this paper.

Chapter 4, in full, is a reprint of the material as it may appear in "Multi-allele species reconstruction using ASTRAL" (2019) Rabiee, Maryam; Sayyari, Erfan; Mirarab, Siavash, Molecular Phylogenetics and Evolution, 130, 286-296. The dissertation author was the primary investigator and first author of this paper.

Chapter 5, in full, is a reprint of the material as it appears in “SODA: Multi-Locus Species Delimitation Using Quartet Frequencies” (2021). Rabiee, Maryam; Mirarab, Siavash, Bioinformatics, 36(24), 5623-5631. The dissertation author was the primary investigator and first author of this paper.

Chapter 6, in full, has been submitted for publication of the material as it may appear in “QuCo: Quartet-based Co-estimation of Species Trees and Gene Trees” (2021) Rabiee, Maryam; Mirarab, Siavash, ISMB, (2022). The dissertation author was the primary investigator and first author of this paper.

VITA

2015	B. S. in Computer Engineering, Software Engineering, Sharif University of Technology, Tehran
2019	M. Sc. in Computer Science, University of California San Diego
2017-2021	Graduate Teaching Assistant, University of California San Diego
2022	Ph. D. in Computer Science, University of California San Diego

PUBLICATIONS

M. Rabiee and S. Mirarab, “QuCo: Quartet-based Co-estimation of Species Trees and Gene Trees” (2021) Rabiee, Maryam; Mirarab, Siavash, ISMB (2022)

M. Rabiee and S. Mirarab, “SODA: Multi-Locus Species Delimitation Using Quartet Frequencies”, Bioinformatics 36, no. 24 (2020):5623-5631.

M. Rabiee and S. Mirarab, “Forcing external constraints on tree inference using ASTRAL”, BMC genomics 21, no. 2 (2020): 1-13.

Q. Zhu, et al. ”Phylogenomics of 10,575 genomes reveals evolutionary proximity between domains Bacteria and Archaea.” Nature communications 10.1 (2019): 1-14.

M. Rabiee and S. Mirarab, “INSTRAL: Discordance-aware phylogenetic placement using quartet scores”, Systematic Biology, (2019), syz045, https://doi.org/10.1093/sysbio/syz045.

M. Rabiee, E. Sayyari, and S. Mirarab, “Multi-allele species reconstruction using ASTRAL” , Molecular Phylogenetics and Evolution,(2018).

C. Zhang, **M. Rabiee**, E. Sayyari, and S. Mirarab. “ASTRAL-III: Polynomial Time Species Tree Reconstruction from Partially Resolved Gene Trees”, BMC Bioinformatics 19, no. S6 (2018): 153

ABSTRACT OF THE DISSERTATION

Scalable algorithms for detecting boundaries and relationships of species from phylogenetic data

by

Maryam Rabiee Hashemi

Doctor of Philosophy in Computer Science

University of California San Diego, 2022

Professor Siavash Mirarab, Chair
Professor Vineet Bafna, Co-Chair

Many disciplines in life sciences, directly or indirectly, utilize evolutionary relationships among species, and this has made species tree inference from sequence data one of the central problems in evolutionary biology. In the last decade, advances in sequencing technologies have drastically reduced the price for DNA sequencing and have led to the ubiquity of molecular sequence data. This data avalanche has impacted phylogenetics, and researchers are facing new computational challenges in handling the new data. The need for phylogeny inference and update methods that can be highly accurate on ultra-large datasets has increased. Researchers seek new

methods that can enforce constraints from precedenting studies when handling the new data.

Another challenge in phylogenetics that is more inherent than the first challenge, is the heterogeneity of evolutionary histories among different genes across species, populations, or even individuals. This discordance among histories has been modeled by multi-species coalescent model (MSC), which has been adopted by several tree inference tools. This dissertation will discuss methods developed for considering the inherent heterogeneity of phylogenetic data, as modeled by the MSC model. The focus of the dissertation is overcoming described challenges and introducing scalable algorithms that can detect boundaries and relationships of species from phylogenetic data.

In Chapter 2, I introduce a new method for updating a species tree with new sequences that aids in utilizing existing phylogenies when inferring new species trees. This method, called INSTRAL, can update a backbone species tree with one new species at a time or several in parallel. Thus, while scalability is achieved, the relationships of new species are not retrieved, and post-processing is needed to obtain fully resolved species trees with no ambiguity. Chapter 3 of the dissertation introduces another method for updating phylogenies with multiple new species, that also obtains the relationships among new species. This method in effect is a constrained species tree inference method, as it creates a constraint-compatible species tree from the input constraint tree and the set of gene trees.

Constraints can come in several forms, and one form of constraint is the monophyly of individuals of a species in the species tree. In chapter 4, a summary method for creating a species tree from multi-individual data is introduced that can infer species tree following the constraint of monophyly of individuals of each species. However, these species boundaries are not always known *a priori*. Species delimitation is a challenging task by itself, and in Chapter 5, I describe a method for species delimitation based on gene tree topologies that is scalable to large datasets with thousands of genes. Finally, in Chapter 6, I describe a scalable quartet-based method to co-estimate gene trees and the species tree simultaneously to infer species tree more accurately.

Chapter 1

Introduction

Many disciplines in life sciences, directly or indirectly, utilize evolutionary relationships among species, and this has made species tree inference from sequence data one of the central problems in evolutionary biology. In the last decade, advances in sequencing technologies have drastically reduced the price for DNA sequencing and have led to the ubiquity of molecular sequence data. This data avalanche has impacted phylogenetics, and researchers are facing new computational challenges in handling the new data. Thus the need for phylogeny inference and update methods that can be highly accurate on ultra-large datasets has increased. Researchers seek new methods that can enforce constraints from precedenting studies when handling the new data.

Another challenge in phylogenetics that is more inherent than the first challenge, is the heterogeneity of evolutionary histories among different genes across species, populations, or even individuals. This discordance among histories has been modeled by multi-species coalescent model (MSC), which has been adopted by several tree inference tools. This dissertation will discuss methods developed for considering the inherent heterogeneity of phylogenetic data, as modeled by the MSC model. The focus of the dissertation is overcoming described challenges and introducing scalable algorithms that can detect boundaries and relationships of species from phylogenetic data.

As the number of available genomes continues to increase, as a result of reduction in sequencing cost and hands-on time by advances in sequencing technologies, a new problem is facing researchers on computational side. With more species becoming available, their position on the existing phylogenies are often sought for. One possible approach is to repeat the whole process of species tree inference from scratch, each time new data is acquired. However, this process is computationally extensive and is not generally scalable to the available amount of data. An alternative to this is updating existing phylogenies with the new data. This process is called phylogenetic placement [90]: adding a new *query* species onto an existing phylogeny. Despite the progress for the *de novo* inference of species trees, updating trees under the MSC model has been understudied.

There has been a myriad of placement softwares developed for placement of a new sequence onto a single tree. There are maximum likelihood (ML) methods such as `pplacer` [90] and `EPA-ng` [9], recent distance-based methods such as `APPLES` [7], and divide-and-conquer methods such as `SEPP` [98]. These methods insert/place a new sequence onto a single tree that could be a gene tree or a species tree when concatenation of sequences is performed with downside of ignoring gene tree concordance. None of these methods can handle species tree placement considering gene tree concordance. In chapter 2, I will discuss INSTRAL (Insertion of New Species using asTRAL) which is a concordance-aware phylogenetic placement software that I developed.

INSTRAL works on an input set of unrooted gene trees that has been updated with the new sequence, using one of the existing method, and a backbone tree. The backbone tree enforces a constraint on the species tree to be inferred. INSTRAL solves an optimization problem similar to ASTRAL [169, 100]. It finds the species tree with maximum amount of shared information, measured by number of shared quartets, with respect to input gene trees, that is compatible with the given backbone tree. In other words, it completes the backbone tree with the new species, using the updated gene trees.

If our goal is to place several new species onto a backbone tree, we can run INSTRAL, several times, in parallel, and update the backbone tree with the set of new species. Following this scenario, we get a species tree that is compatible with the backbone tree, however, we do not obtain the relationships of the new species, as they are inserted one at a time, and possibly in parallel. If the relationship of the new species is sought for, we need to take another path. Constructing phylogenies input constraints is the alternative approach. In chapter 3, I will discuss a method for creating phylogenies following input constraints, or it can be seen as placement of multiple new species at the same time. Besides multiple placement, constrained tree inference has several other use cases;in a hypothesis-driven analyses, it can be used to select the best hypothesis among a set of available hypotheses that have been gathered from exiting studies [92, 117, 153, 5].

Constrained searches can also be used in model selection in phylogeny; e.g., for testing whether a polytomy [118] for a node can be rejected or how the monophyly of a group [21] dominates the alternative model or not. Likewise, the support of branches that are not recovered in the main analysis can be assessed by enforcing constraint. This support is called “hidden” support, and can provide useful information for comparing the outcomes of several methods and different hypotheses. Furthermore, constrained searches have been used to combine the results of multiple methods [33] and more recently in taxonomic profiling [107].

The necessity of constrained search has been confirmed also by its implementation by most phylogenetic inference tools. Although there are several implementations of constrained search available, the problem is inherently difficult. Phylogeny inference is typically formulated as an optimization problem over the exponential space of all possible trees. Searching through all trees is not possible, even for moderately sized datasets. Several heuristics are employed by tree inference methods including iterative search (e.g., hill-climbing). These methods can be categorized by maximum parsimony and maximum likelihood. There is a third category, dynamic programming (DP), for an optimization score of interest, which usually solves the problem on a restricted search space of trees.

Phylogenetic inference using this particular DP approach has been used for many optimization criteria, including duplication and loss [55, 24, 10], deep coalescence [154], Robinson Foulds (RF) distance [157], quartet score [20, 99], and others [158]. Among these, ASTRAL [99], has found increasing popularity [95]. To our knowledge, the DP paradigm has not been adopted to perform constrained searches with user-defined constraints. Performing constrained searches in the DP paradigm has its own challenges as we need to make sure the search space explored by DP are all consistent with the constraint tree. Finding a large enough search space satisfying user-provided constraints is not an easy task. In my other paper, covered by chapter 3, I designed a method to perform constrained searches for DP for the first time with a new tree completion algorithm, and it is implemented inside the ASTRAL software.

Constraints can come in several forms like monophyly of groups of species, or sister clades, and they can be specified by a tree containing polytomies or membership pairs. Another form of constraint is association of individuals captured from a species to that species. Beyond dramatically increasing the amount of data available [132], whole-genomes have enabled us to study individual genealogies, which can be discordant with each other and with the species tree [85]. ILS can arise when multiple alleles of a gene survive through consecutive speciation events, and for potential impacts of polymorphism in present-day species, several authors have suggested sampling multiple individuals per species (and/or phasing) to create multi-allele datasets where each species can have multiple alleles per locus [86, 57, 70, 91, 59, 26]. This imposes another kind of constraint on species tree that need to be accounted. Models of sequence evolution that directly account for polymorphisms have been also developed [36]. Alternatively, one can use summary methods to analyze multi-allele datasets. This requires that gene trees estimated from sequence data are either multi-labeled by species names (i.e., several nodes in a gene tree are labeled by the same species) or are labeled by the name of individuals and a mapping between individuals and species is known. Then, the summary method can estimate the best species tree labeled with names of species; this is equivalent to finding the best species tree labeled with individual names constrained to each species being monophyletic.

Predefining species boundaries side-steps difficulties of defining boundaries of recently diverged species [23, 113], leaving that question as another type of computational methods, species delimitation, which I will cover in chapter 5. In other words, in multi-allele inference we assume the species are correctly delimited, although delimitation by itself is a very difficult task and requires application phylogenetic methods in advance.

When species boundaries are known *a priori*, the MSC model easily extends to the multi-individual case [152, 2, 3]. While the evidence for the cost-effectiveness of sampling multiple individuals remains mixed [116], several methods exist that can use such data [57, 82, 83]. To our knowledge, NJst is the only summary method that can handle multi-individual *unrooted* gene trees [82], and after a fix in handling multi-individual data [1], the NJst method is now statistically consistent for multi-individual datasets.

In chapter 4, I discuss multi-allele version of the popular software ASTRAL. The published ASTRAL algorithm, until now, could only take single-labeled trees as input and had no way of handling multi-allele datasets. Incipient implementations of a feature in ASTRAL to handle multi-allele datasets were not rigorously tested; nor were they formally described. In this chapter, I will introduce a new algorithm for handling multi-labeled gene trees in ASTRAL (which is different from the previous untested method) and establish its accuracy on both simulated and empirical datasets. I show that the quartet optimization problem extends in a natural way to multi-labeled datasets, leaving us with only one difficulty: defining the constrained search space for ASTRAL. I propose and test heuristic approaches based on subsampling individuals to build a sufficiently large search space. I test if predefining species boundaries improves accuracy and compare the accuracy of ASTRAL to NJst.

I also discuss the trade-off between sampling more genes versus more individuals, and our finding that sampling more genes is more effective than sampling more individuals, even under conditions where trees are shallow (median length: $\approx 1N_e$) and ILS is extremely high.

As discussed earlier, defining species boundaries and species delimitation is a difficult task [23, 127]. Evolution results in diversity across species and diversity within the same species, in ways that can make it difficult to distinguish species. Definitions of what constitutes a species are also varied [32] and subject to debate. Among varied species concepts, the most commonly used for Eukaryotes is the notion that individuals within a species should be able to mate and reproduce viable off-springs.

A wide range of species delimitation methods exist. More traditional methods simply relied on the mean divergence between sequences [e.g., 56, 122] or patterns of phylogenetic branch length [171, 53, 48] in marker genes or concatenation of several markers [e.g., 119]. Due to limitations of marker genes [61], many approaches to species delimitation have moved to using multi-locus data that allow modeling coalescence within and across species [69, 166, 113], not to mention more complex processes such as gene flow [e.g., 80]. Species delimitation is often studied using the Multi-species Coalescent (MSC) model [115, 126]. In this model, individuals of the same species have no structure within the species, and thus their alleles coalesce completely at random. Coalescence is allowed to happen deeper than the first opportunity, producing gene tree discordance due to Incomplete Lineage Sorting (ILS). In this context, given a set of sampled individuals, delimitation essentially requires inferring gene trees, one per locus, and detecting which delimitation is most consistent with patterns of coalescence observed in the gene trees.

Existing methods for species delimitation under the MSC model tend to suffer from one of two limitations. The most accurate methods are based on Bayesian MCMC and infer gene trees, (optionally) species trees, and species boundaries (e.g., BPP [166, 167], ABC [22], and STACEY [66]). Other Bayesian methods use biallelic sites [78], incorporate morphological data [145], or use structure [62]. These methods, however, are typically slow and cannot handle even moderate numbers of samples [53, 164].

The second class of methods (e.g., SpedeSTEM [46]) rely on a three-step approach: first, infer gene trees, then, date gene trees so that they all become ultrametric (i.e., have a unique root to tip distance), then, use ML calculation of alternative delimitations under the MSC model to decide species boundaries. These methods have been less accurate than Bayesian methods, and their reliance on ultrametric trees makes them hard to use for datasets where rates of evolution change substantially across the tree [22]. Yet other methods [e.g., 113, 172] rely only on input gene tree topologies, as in my proposed method.

In chapter 5, I introduce SODA, Species bOundry Delimitation using Astral. It is a new species delimitation approach that builds on the success of our species tree inference tool ASTRAL [99, 100, 169]. In multi-individual ASTRAL paper (chapter 4), we observed that if species boundaries are ignored, ASTRAL most often recovers individuals of the same species as monophyletic [123]. This result suggests a species delimitation method: Infer an ASTRAL tree with all individuals and use patterns of quartet trees mapped onto that species tree to decide where coalescence is completely random and where it is not; these boundaries can define species. By relying on quartet frequencies and the ASTRAL machinery, SODA is able to handle very large datasets with short running times. I discuss evaluation of its accuracy and scalability in simulation and on empirical datasets, and also compare its performance to a method that co-estimates gene trees and species tree, BPP.

There are several co-estimation method available, but they are not scalable to even medium sized datasets. Chapter 6 is dedicated to a method that I developed for co-estimation of species tree and gene trees that could be scalable. I build on three insights to develop a likelihood-based approach to co-estimation that can scale. One insight is that a method can be counted as a co-estimation method even if it first estimates gene tree distributions independently and then combines them. When a tree distribution is inferred for each gene, a post-processing step may be able to amalgam the results from multiple genes and *adjust* their distribution in a way that would correspond to computing them jointly. This insight, which we further explain below, may seem counter-intuitive but is not new for species tree inference [4, 76] or improving gene trees [150]. Second, sampling parameters such as branch lengths and substitution rates can slow down co-estimation. In most applications, the vast majority of parameters of individual gene trees (e.g., their branch lengths) are nuisance parameters that biologists do not directly examine. Therefore, we focus on tree topologies, marginalizing over gene tree branch lengths and rates using approximations that correspond to assumptions about how branch lengths in substitution units are generated. As we will see, this topology-focused approach also eliminates a need for assuming a model of molecular clock rate. Finally, building on the success of quartet-based methods for handling ILS [e.g., 99, 27], we co-estimate quartet species trees. Lest the reader worries about impacts of lowered taxon sampling and increased long branch attraction when using quartets, we note that gene tree estimation can be performed on the full set of taxon; only the amalgamation step should focus on (induced) quartets.

In chapter 6, I introduce and explain Quco designed based on these insights, and benchmark its performance on several datasets.

Chapter 2

INSTRAL: Discordance-aware Phylogenetic Placement using Quartet Scores

Phylogenomic analyses have increasingly adopted species tree reconstruction using methods that account for gene tree discordance using pipelines that require both human effort and computational resources. As the number of available genomes continues to increase, a new problem is facing researchers. Once more species become available, they have to repeat the whole process from the beginning because updating species trees is currently not possible. However, the *de novo* inference can be prohibitively costly in human effort or machine time. In this paper, we introduce INSTRAL, a method that extends ASTRAL to enable phylogenetic placement. INSTRAL is designed to place a new species on an existing species tree after sequences from the new species have already been added to gene trees; thus, INSTRAL is complementary to existing placement methods that update gene trees.

2.1 Introduction

Gene trees and species trees can differ [85, 38], and methods for accounting for discordance are now widely available and are adopted by many [151, 45]. Discordance-aware methods come in many forms, such as co-estimation of gene trees and species trees [e.g., 81, 57, 15], and site-based methods [e.g., 19, 27, 36, 141]. The most scalable approach for species reconstruction has remained what has been called a summary approach: first gene trees are inferred independently for all loci, and then they are combined to build a species tree. Many methods are available for combining gene trees [e.g., 71, 84, 105, 83, 25, 82, 163, 12, 135], and many of them are statistically consistent under various models of genome evolution. In particular, many methods have been designed to be consistent under the multi-species coalescent model [115, 126], which seeks to capture incomplete lineage sorting (ILS). Several statistically consistent summary methods, including ASTRAL [99], NJst/ASTRID [82, 156], and MP-EST [83] are in wide use.

Despite the progress for the *de novo* inference of species trees, updating trees under the MSC model has received little attention. As new genomes become available, researchers often need to know their position on an existing phylogeny. One solution is to reconstruct the species tree from scratch each time new data becomes available. This process can require excessive computation and may not scale to groups with tens of thousands of genomes (more than a hundred thousand bacterial genomes are currently available).

A more efficient alternative is what has been called phylogenetic placement [90]: adding a new *query* species onto an existing phylogeny. For placing a new sequence onto a single tree, we have maximum likelihood (ML) methods such as `pplacer` [90] and `EPA` [13, 9], distance-based methods such as `APPLES` [7], and divide-and-conquer methods such as `SEPP` [98]. Even earlier, sequential sequence insertion algorithms, which essentially solve the same computational problem, existed [e.g., 41, 50].

Existing placement algorithms place a new sequence onto a single tree, which is typically a gene tree. Current methods can be used to place new sequences on an estimate of the species tree using a concatenation of multiple genes, but this approach ignores gene tree discordance. We are not aware of any discordance-aware methods for placement onto species trees. Here, we present INSTRAL (Insertion of New Species using asTRAL) which extends ASTRAL to enable placing a new species onto an existing species tree.

2.2 Description

2.2.1 Background

ASTRAL estimates an unrooted species tree given a set of unrooted gene trees and is statistically consistent under the multi-species coalescent model given true gene trees [99]. ASTRAL seeks to maximize the quartet score: the total number of induced quartet trees in the gene trees that match the species tree. Similar to earlier work [20], ASTRAL uses dynamic programming to solve this NP-Hard problem [74]. However, to allow scalability, it constrains its search space so that the output draws its clusters from a predefined set X, which consists of clusters from gene trees and others that are heuristically selected (a cluster is one side of a bipartition). The most recent version, ASTRAL-III [169] guarantees polynomial running time and scales to datasets with many thousands of species.

2.2.2 Problem statement

Quartet Placement Problem. Given a set of k unrooted trees labeled with $n+1$ species and a backbone tree on n species, find the tree that includes all $n+1$ species and has the maximum quartet score with respect to the input trees.

Thus, one species, called the *query*, is not present in the backbone tree, and the goal is to insert the query species into the backbone. A typical use of this problem is placing a new species onto an existing species tree (Fig. 2.1). Imagine a previous analysis has already produced a set of k gene trees on n species and an ASTRAL tree (inferred from those k gene trees). Now, a new species with genome-wide data has become available. To insert the new species onto a given ASTRAL tree, we first add it to each of the k gene trees using tools such as `SEPP`, `pplacer`, or `EPA`. Then, we use the updated gene trees in addition to the existing ASTRAL tree as input to the quartet placement problem; the output will be a species tree with the new species included. Just like ASTRAL, the use of the quartet score ensures that the inferred position of the new species is a statistically consistent estimator of its true position under the MSC model given true gene trees.

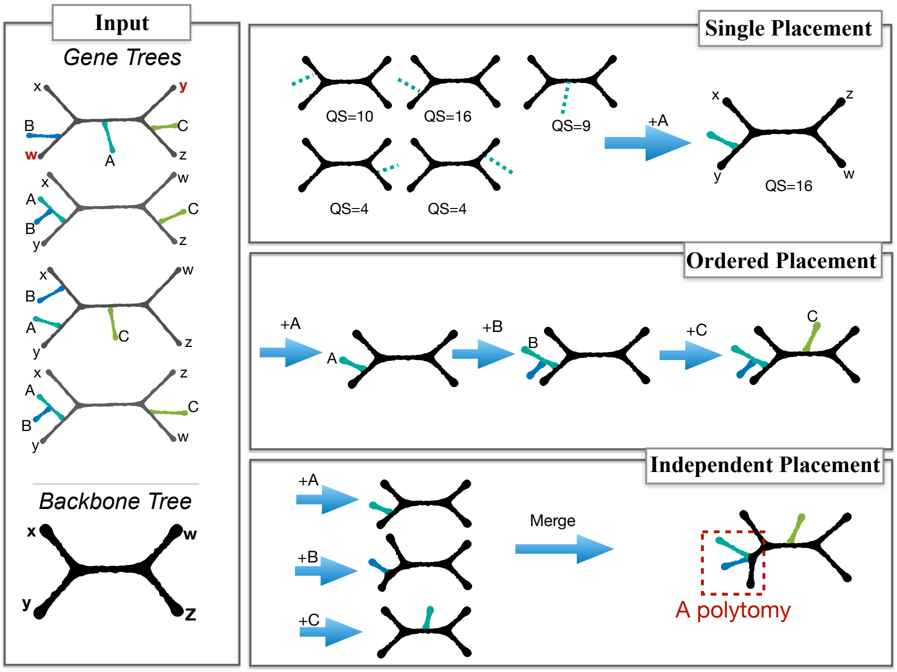


Figure 2.1: Left: The quartet placement problem. A backbone species tree with four leaves ($\{x, y, w, z\}$) and $k = 4$ gene trees are given; each gene tree also has new species (here, $\{A, B, C\}$). Note that the first gene tree is discordant with the species tree. Top right: placing a single new species (A) on the backbone tree requires computing the quartet score (QS) for each placement and finding the maximum. Here, the optimal placement is on the terminal branch of y, which matches 16 out of 20 quartets on $\{x, y, w, z, A\}$ in the gene trees. Middle right: placing multiple species can be done by ordering them and placing them one at a time. Bottom right: alternatively, all new species can be placed independently, and the results can be merged at the end (creating polytomies when multiple new species are placed on the same branch).

2.2.3 INSTRAL (single query)

INSTRAL finds the optimal solution to the quartet placement problem. Unlike ASTRAL, the number of possible solutions to the placement problem is small (grows linearly with *n*), and thus, INSTRAL can solve the problem exactly even for large trees. In principle, it is possible to develop algorithms that compute the quartet score for all possible branches, one at a time, and to select the optimal solution at the end. However, the ASTRAL dynamic programming allows for a more straight-forward algorithm.

The ASTRAL algorithm will solve the placement problem if we define the search space (set X) such that *all* trees that induce the backbone tree and *only those trees* are allowed. To achieve this, X should be the set of all clusters in the backbone tree both with and without the new species added. More precisely, let q be a set with the new species and let $\mathcal{B}(T)$ denote the set of all (including trivial) bipartitions of the backbone tree T on the leaf-set $\mathcal{L}$ with each bipartition represented as a tuple: $(A, \mathcal{L} \setminus A), A \subset \mathcal{L}$. Then

$$X = \{q, \mathcal{L} \cup q, \mathcal{L}\} \cup \bigcup_{(A,\mathcal{L}\setminus A) \in \mathcal{B}(T)} \{A, \mathcal{L} \setminus A, A \cup q, (\mathcal{L} \setminus A) \cup q\} \,. \tag{2.1}$$

With this set X, the search space will include all possible placement of the query on the backbone tree (due to $A \cup q$ and $(\mathcal{L} \setminus A) \cup q$). Moreover, every bipartition built from X is one that existed in the backbone tree once q is removed and thus only trees that induce the backbone are allowed. Since ASTRAL finds the optimal placement restricted to the search space, this algorithm is guaranteed to solve the quartet placement problem exactly. The number of clusters in this search space is $3 + 4(2n-3) = \Theta(n)$. Thus, its running time increases as $\Theta(nD) = O(n^2k)$ where D is the sum of degrees of all *unique* nodes in the input gene trees [see 169, for details].

2.2.4 Adding multiple new species

If multiple queries are available, we can still attempt to use the basic INSTRAL algorithm in one of two ways (Fig. 2.1). *i*) *Independent placement:* We add all the new queries independently without trying to find the relationship among the queries. This approach is reasonable if the goal is to detect the identity of some unknown species or if the set of new species are expected not to belong to the same branches of the backbone tree. If needed, we can merge separate placements into a single tree, introducing polytomies wherever multiple queries are placed on the same branch. *ii*) *Ordered placement:* We order the queries (e.g., arbitrarily) and then add them to the backbone one at a time, updating the backbone tree each time to include the latest query. This ordered placement approach gives us the relationships between queries. However, like similar greedy algorithms [41], it is not guaranteed to find the optimal tree at the end.

The advantage in using the independent insertion approach is that adding m queries requires time that increases linearly with m whereas the time needed for the ordered placement increases proportionally to m^3. The *de novo* execution of ASTRAL-III on $n+m$ species requires $O(((m+n)k)^{2.73})$ time in the worst case [169]. In contrast, INSTRAL-independent would run in $\Theta(m.D.n) = O(mn^2k)$ and INSTRAL-ordered would require $O(m^3k+(n+m)nmk)$. Thus, the relative running time of ASTRAL-III and INSTRAL-ordered depend on values of n, m, and k, while INSTRAL-independent is always faster than ASTRAL-III.

We can also ask a statistical consistency question: Starting from a correct backbone tree and placing several new species using INSTRAL, is the output tree guaranteed to be correct with high probability as the number of error-free gene trees drawn under MSC goes to infinity? In the independent placement scenario, the output is an unresolved tree and cannot be a statistically consistent estimate of the species tree. However, due to the consistency of each placement, with arbitrarily high probability, *all* placements are on the correct branch of the backbone given enough gene trees, and therefore, the output tree will not have wrong branches with high probability (but it will have missing branches). In the ordered placement scenario, since each placement is correct with an arbitrarily high probability given enough genes, we can make *all* placements be correct with an arbitrarily high probability. Thus, the ordered placement result is a statistically consistent estimate of the species tree (see supplementary material for proof). Note that this consistency is despite the fact that the ordered placement is *not* an optimal solution to the problem that ASTRAL seeks to solve.

2.3 Benchmark

Datasets. We first benchmark INSTRAL on a simulated dataset previously generated by Mirarab and Warnaw [100]. This dataset has 200 ingroup taxa and an outgroup species and is generated using SimPhy [88]. By setting the maximum tree heights to 10^7, 2×10^6, or 5×10^5 generations, this dataset has created three model conditions with respectively, moderate, high, or very high levels of ILS; the average normalized [128] distance (RF) between true gene trees and the true species tree are 15%, 34%, and 69%, respectively. In our experiments, we use gene trees inferred using FastTree-II [120] from sequence data. These inferred trees have relatively high levels of gene tree error ($25\%, 31\%$ and 47% for the three model conditions). For each replicate, we also have estimates of the species tree using both ASTRAL-II and concatenation with ML (CA-ML) performed using FastTree-II. We have 100 replicates per condition, and each replicate has 1000 gene trees, from which we have randomly sampled 200 and 50 gene trees to create three different input sets. Thus, in total, we have 9 model conditions (ILS level×# Gene). Following Mirarab and Warnaw [100], three replicates are removed because their gene trees are extremely unresolved; this leaves us with $9 \times 100 - 3 \times 3 = 891$ datasets in total.

2.3.1 Leave-one-out experiments.

Comparision to ASTRAL.

For each dataset, an ASTRAL species tree inferred from gene trees is available. For each of the 200 ingroup species in each dataset, we prune it from the ASTRAL tree and by INSTRAL we add it back onto the tree, using FastTree gene trees as input. Thus, overall, we have $891 \times 200 = 1.782 \times 10^5$ independent placements. When there are multiple placements with equal quartet scores (happens in only 63 cases), we break ties similarly to the full backbone ASTRAL tree.

Among all of these placements, in only 316 cases ($< 0.2\%$) the output trees have different quartet scores compared to the original ASTRAL tree. Note that INSTRAL is guaranteed to find the optimal placement, and therefore, its quartet score is always at least as good as the ASTRAL tree. Thus, these 316 cases are those where ASTRAL has failed to find the optimal placement for a species. We note that 178 out of 316 cases correspond to the model condition with very high ILS and only 50 genes. Increasing the number of genes and reducing the amount of ILS both decrease the number of cases where ASTRAL is sub-optimal (Table 2.1). For example, with moderate ILS/1000 genes, only 4 out of 20,000 placements using INSTRAL improved quartet scores compared to ASTRAL. Only 176 of 316 cases result in any change in the RF distance of the inferred tree compared to the true tree, and only in 59 out of 176 cases did INSTRAL reduce the RF distance compared to ASTRAL. Thus, removing and reinserting a species using INSTRAL is generally consistent with the ASTRAL tree but in rare cases improves quartet scores.

Comparison to concatenation using ML (CA-ML).

An alternative to INSTRAL is to simply concatenate all the genes and use ML to place the query on an existing tree. We compare INSTRAL to this CA-ML approach using EPA-ng [9] for ML placement. To avoid biasing results towards one method, we use the true species

Table 2.1: For each condition, we show the number of cases where (left) the INSTRAL tree has a different (i.e., higher) quartet score than the full ASTRAL tree, (middle) the [128] distance (RF) of the INSTRAL tree to the true tree is different than the RF distance of the full ASTRAL tree to the true tree, and (right) the INSTRAL tree has a *reduced* RF distance to the true tree compared to the full ASTRAL tree. All numbers are out of 20,000 insertions, except for very high ILS, which is out of 19,400.

	50 genes	200 genes	1000 genes
Moderate ILS	11; 8; 1	5; 3; 0	4; 4; 0
High ILS	41; 31; 13	12; 8; 5	5; 3; 2
Very high ILS	178; 140; 26	41; 33; 7	19; 12; 6

tree as the backbone, both for INSTRAL and CA-ML. For CA-ML, we use RAxML[146] to compute branch lengths of backbone and GTR+Γ model parameters based on true alignment. We test INSTRAL with two types of input. In one case, gene trees are computed *de novo* using FastTree-II. In leave-one-out experiments, we approximate this scenario by simply removing each species from the species tree but keeping it in all our estimated gene trees. In the second case, gene trees are updated using EPA-ng; thus, we first remove the query species from all gene trees and then place it on each gene tree using EPA-ng. Due to memory requirements of EPA-ng (up to 35GB), we could only run it for up to 200 genes and we restrict leave-one-out tests to only 50 randomly selected leaves.

In terms of accuracy, INSTRAL outperforms the CA-ML using EPA-ng regardless of the amount of ILS or the number of genes (Table 2.3 and Fig 2.2a). For example, with high ILS and 200 genes, CA-ML fails to find the correct placement in 17% of cases, while INSTRAL is incorrect in 5% and 8%, respectively, with *de-novo* and EPA-ng gene trees (Table 2.3). When methods are wrong, there are typically off by one edge and only rarely by two or more edges (Fig. 2.5). As the level of discordance goes up, the error increases for all methods, and contrary to our expectations, the relative performance of methods does not change. However, as the number of genes increases from 50 to 200, INSTRAL enjoys a substantial reduction in error but CA-ML benefits less from increased gene sampling for moderate to high levels of ILS (e.g., for moderate

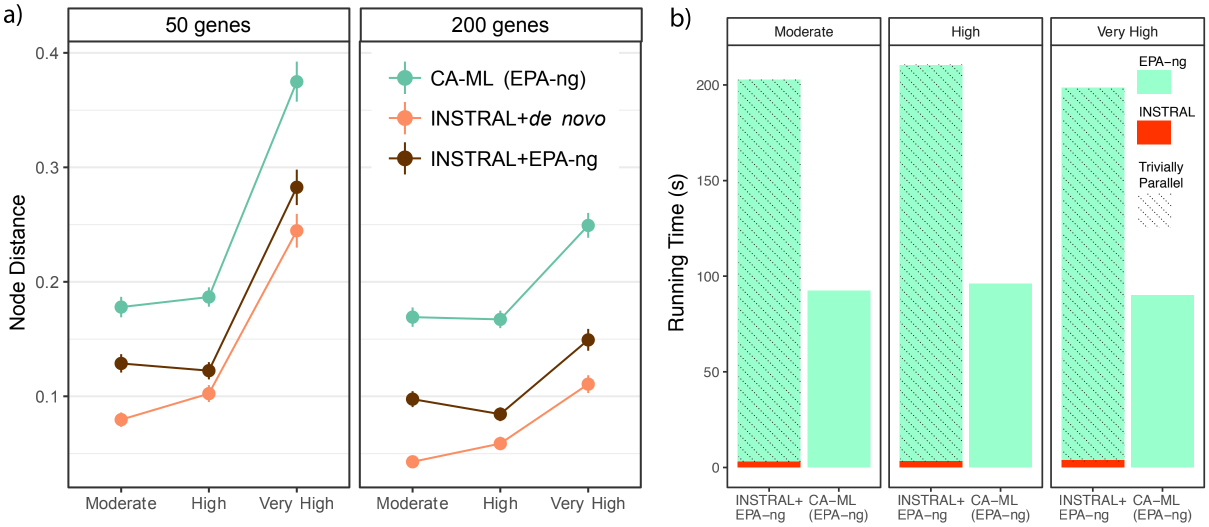


Figure 2.2: Comparison of concatenation using ML (CA-ML) and INSTRAL run on *de novo* gene trees or on gene trees updated using EPA-ng. Both method place on the true species tree in a leave-one-out experiment (50 species per replicate) with 200 or 50 genes. (a) Mean and standard error of placement error, measured as the number of nodes between the correct placement and placed edge. Results are over 2,500 placements for moderate and high and 2,350 placements for very high ILS. (b) Total running time measured in seconds, measured on the same machine, and all methods run with a single core both for EPA-ng and INSTRAL.

ILS, mean error drops from 0.08 edges to 0.04 for INSTAL+*de novo* but only from 0.18 to 0.17 for CA-ML). In all conditions, using *de novo* gene trees resulted in improved accuracy compared to using EPA-ng for updating gene trees; however, INSTRAL+EPA-ng is still substantially more accurate than CA-ML.

Comparing the total running time, INSTRAL+EPA-ng takes twice as much time as CA-ML using takes (Fig. 2.2b). INSTRAL+EPA-ng took on average about 200 seconds, of which, on average only 3 seconds were spend by INSTRAL and the rest was used up by EPA-ng on gene trees. However, note that gene tree updating using EPA-ng enjoys trivial parallelism (each gene tree can be assigned to a different CPU) whereas CA-ML does not enjoy trivial parallelism. Finally, using INSTRAL+EPA-ng requires a lot less memory than CA-ML using EPA-ng. CA-ML needed up to 35GB of memory (mean 19GB), while INSTRAL+EPA-ng runs with less than 0.5GB of memory in every case (Fig. 2.6).

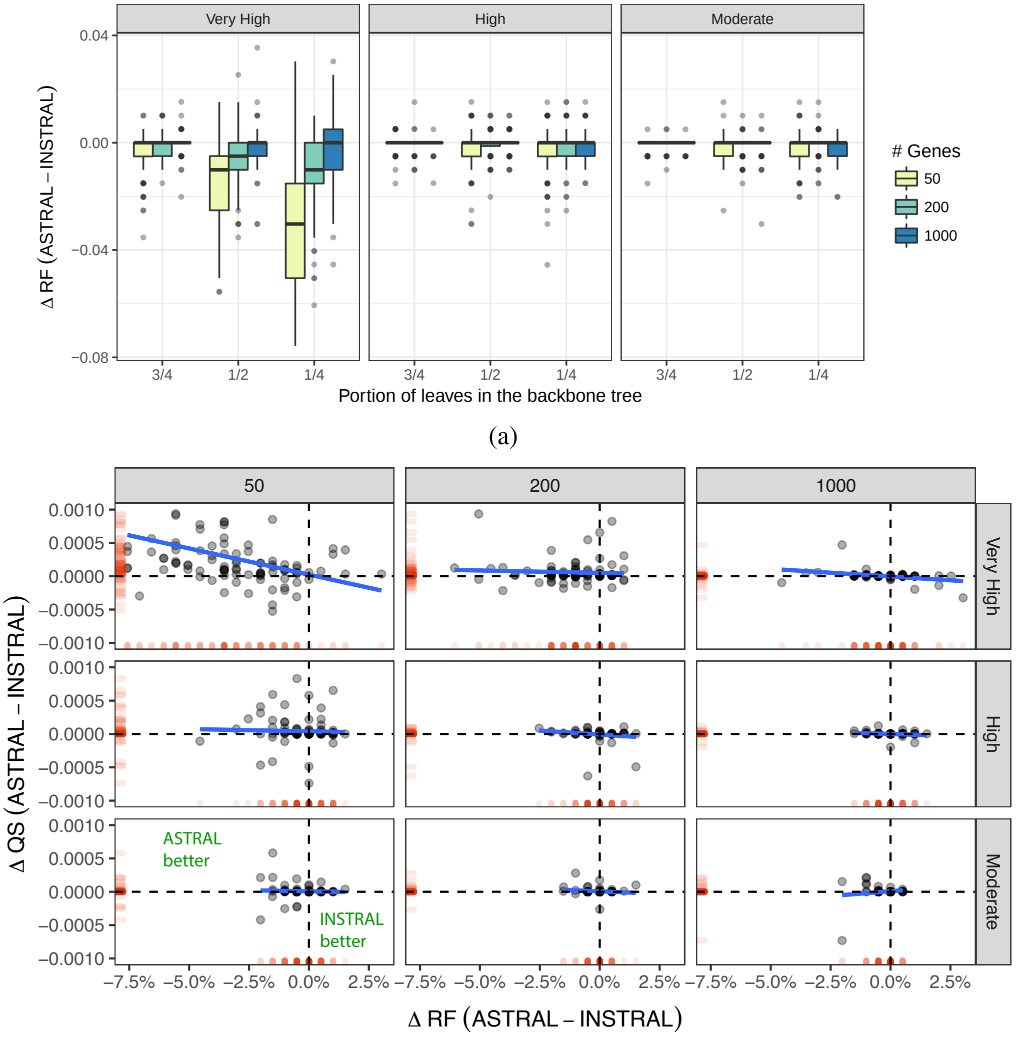
2.3.2 Ordered placement.

To see if the agreement with ASTRAL remains if more species are placed using INSTRAL, we perform a second experiment. Here, we first prune a portion ($\frac{1}{4}$, $\frac{1}{2}$, or $\frac{3}{4}$) of species from the ASTRAL species tree, order removed species randomly, and then place them one after another on the backbone tree, updating the backbone tree each time (Ordered Placement in Fig. 2.1). In the end, we have a tree on the full leaf-set; this tree, which we call the INSTRAL tree, can be thought of as a greedy solution to the same problem ASTRAL seeks to solve.

ASTRAL and INSTRAL trees have similar RF distances to the true tree, but ASTRAL is somewhat more accurate in the hardest conditions (Fig. 2.3a). Overall, the normalized RF error of ASTRAL is on average 0.3% lower than INSTRAL (corresponding to roughly half an edge), and these improvements are statistically significant ($p \ll 10^{-6}$ according to a paired t-test). Among all $891 \times 3 = 2,673$ INSTRAL trees that we have computed, 1,470 have RF distances to the true tree that are identical to the ASTRAL tree. Differences in the RF distance are seen more often among replicates with very high ILS (mean RF difference: 0.7%), 50 genes (mean RF difference: 0.7%), or starting trees with $1/4$ of the species (mean RF difference: 0.6%). Increasing the number of genes, increasing the size of the starting tree, and reducing the ILS reduce the number of mismatches between ASTRAL and INSTRAL (Fig. 2.3a).

Unlike the case of a single insertion, for multiple species, the quartet score of INSTRAL can be higher or lower than ASTRAL. Overall, when the two trees do not agree, ASTRAL tends to have higher quartet scores (Figs. 2.3b and 2.7). Out of 2,673 cases, ASTRAL has higher quartet scores in 1,210 cases while INSTRAL is better in 231 (they tie in the remaining 1,232). Reducing the number of genes and increasing the level of ILS both magnify the improvements of ASTRAL compared to INSTRAL.

Scalability. To test the scalability of INSTRAL, we started with a backbone tree of 10,000 species from a previous publication [169], and down-sampled it to smaller trees (down to 250). Each time, we placed 400 to 800 genomes on the backbone and computed the time INSTRAL



(a)

(b)

Figure 2.3: Comparison of ASTRAL and INSTRAL. (a) Δ RF: The Robinson Foulds (RF) distance of the ASTRAL tree to the true tree minus the RF distance of INSTRAL-ordered tree to the true tree (negative: INSTRAL is better). The size of the starting tree is set to $\frac{1}{4}$, $\frac{1}{2}$, or $\frac{3}{4}$ of species (51, 101, or 151). For three levels of ILS (boxes), each with three numbers of genes (colors), boxplots show distributions of ΔRF (100 points everywhere, except for very high ILS, where it is 97 points.) (b) Change in the quartet score (QS) versus the ΔRF for the starting tree with $\frac{1}{4}$ of species (see Fig. 2.7 for others). The marginal red bars show the projection of data on each axis.

took for the insertion (Fig. 2.4). On the backbone of 10,000 species, each placement took close to 16 minutes on average. As the backbone size decreased, the running time rapidly decreased and was close to 8 seconds on a backbone tree of 250 species. As expected, the running time grows

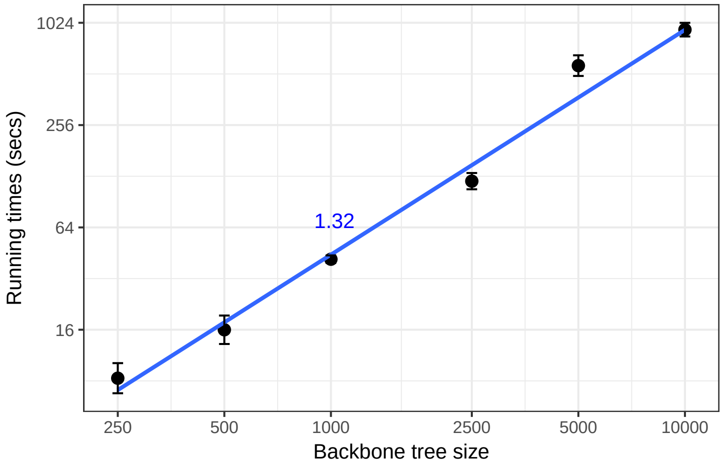


Figure 2.4: The running time scaling of INSTRAL versus backbone size n. Starting with a simulated dataset with 10,000 leaves, we prune random sets of leaves to create smaller trees. Dots and bars show the average and standard error of the running times of inserting a new genome to the backbone (800 insertions for $n < 5000$ and 400 insertions for n$\geq$ 5000). The slope (1.32) of the line fitted to this log-log plot gives an empirical estimate of the running time complexity being close to $n^{1.3}$, which is consistent with the theoretical running time complexity of $\Theta(n.D) = O(n^2)$.

faster than linearly with the size of the backbone (proportional to $n^{1.3}$ in this case).

2.4 Biological Examples

We have tested INSTRAL on three biological datasets: two transcriptomic datasets on insects by [101] and plants by [161], and an avian dataset by [64]. The insect dataset includes 1,478 protein-coding genes from 144 species spanning all of the insect diversity and has been recently re-analyzed using ASTRAL by [138]. The plant dataset includes 103 species and 424 genes, and the original study reported an ASTRAL tree. The avian dataset consists of 48 genomes representing all the orders of birds. For this dataset, statistical binning was used to build 2022 supergene trees [96] and [136] have published an ASTRAL tree on these supergene trees. Among these datasets, the avian dataset has extremely high levels of gene tree discordance.

For each of these datasets, we removed species one by one and placed them back onto the species tree using INSTRAL. In every case, INSTRAL found the same position for the new

Table 2.2: The average and standard deviation of RF distance between ASTRAL and INSTRAL trees as well as the change in the quartet score (INSTRAL-ASTRAL) on 20 random sets from each biological dataset. For each random set of leaves as a backbone tree, ordered placement has been done.

	Avian		Insects		1KP	
	mean	stdev	mean	stdev	mean	stdev
RF	0.0311	0.0592	0.0195	0.0126	0.0115	0.0131
QS	-0.0001098	0.0002973	-0.0000026	0.0000361	-0.0000039	0.0000057

species as the backbone ASTRAL tree. In contrast, EPA-ng on concatenated data of 1KP (the only dataset where we were able to test CA-ML) failed to find the same placement as the backbone for 35 out of 103 species and was on average away from the backbone position by 0.53 edges (Fig. 2.8).

We also tested the ordered placement, where we randomly selected half of the species (20 replicates), removed them, ordered them, and inserted them back on the remaining part of the tree using INSTRAL. The resulting INSTRAL-ordered trees were similar to the full ASTRAL tree (Table 2.2), recovering the same tree in one-third of cases and changing by one or two branches in a majority of the remaining cases (Fig. 2.9a). In several replicates, trees changed for five or more branches, including two replicates of the avian dataset, where the INSTRAL differed from ASTRAL in nine branches. In both cases, two or three unstable taxa had moved by several branches, causing the high incongruence (Fig. 2.9b). More broadly, changes are mostly among unstable taxa. For example, in the avian tree, Hoatzin, the most challenging taxon, moves by one branch in several replicates. The resulting INSTRAL trees have reduced quartet scores compared to ASTRAL trees (Table 2.2). Overall, these results indicate that for datasets with very high ILS, using INSTRAL instead of ASTRAL runs the risk of producing sub-optimal trees.

2.5 Availability

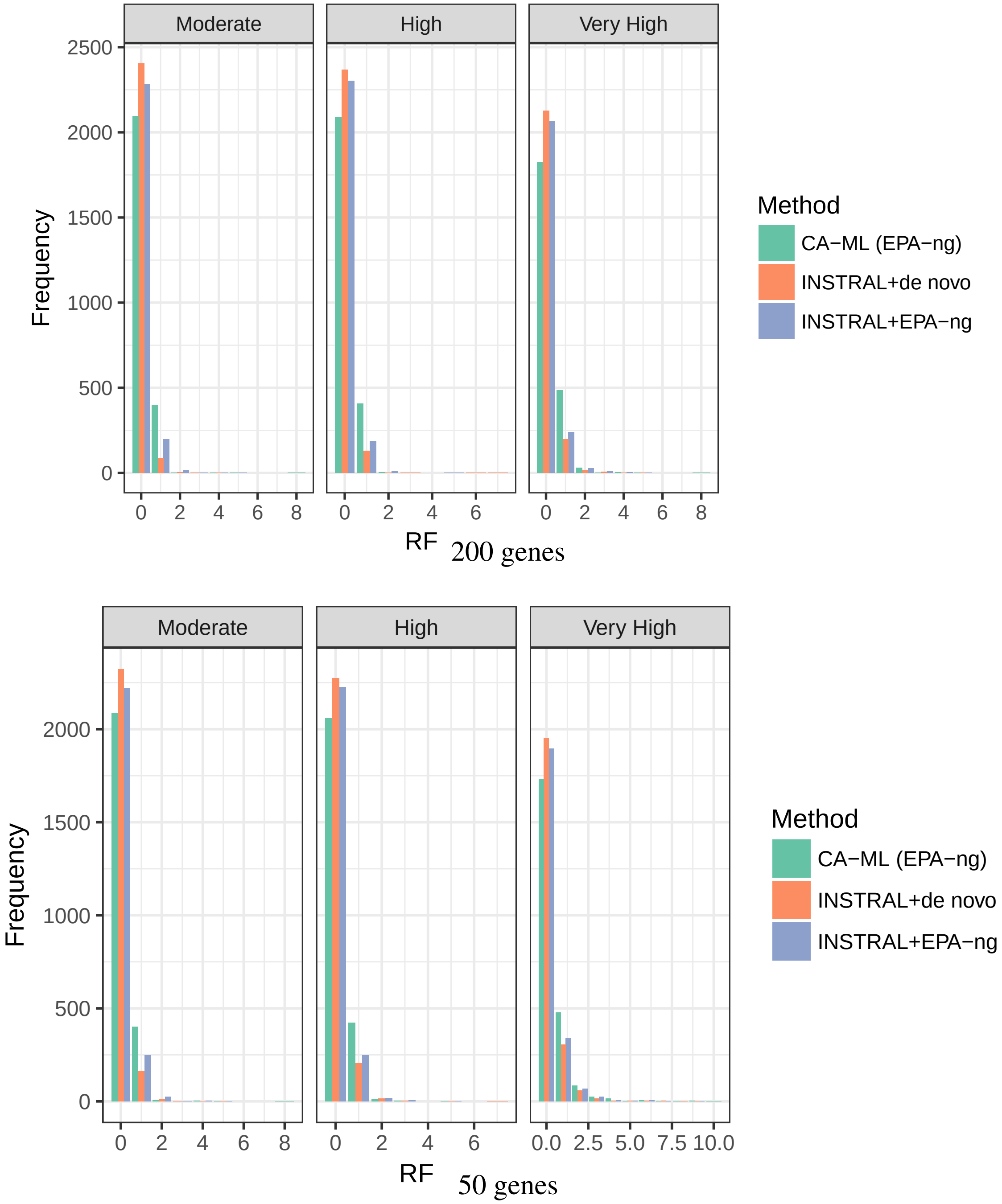
INSTRAL is available on GitHub (https://github.com/maryamrabiee/INSTRAL) in open-source. It is implemented in Java with straight-forward installation (the only dependency is Java 6+). A template tutorial and instructions to run INSTRAL is given there. The generated data, scripts to generate those data and results given in this paper are also available on GitHub (https://github.com/maryamrabiee/INSTRAL-results).

2.6 Acknowledgement

Chapter 2, in full, is a reprint of the material as it appears in “INSTRAL: Discordance-aware Phylogenetic Placement using Quartet Scores” (2020). Rabiee, Maryam; Mirarab, Siavash, Systematic Biology, 69(2), 384-391. The dissertation author was the primary investigator and first author of this paper.

Table 2.3: Comparison of INSTRAL with *de novo* or EPA-ng gene trees and CA-ML using EPA-ng. For two numbers of genes (*k*) and all three model conditions (ILS level), we show % cases with placement on an incorrect branch and mean (and standard deviation) of the node distances between each placement and the correct placement.

k	ILS	% incorrect placements			mean (std) of node distance		
		CA-ML	INSTRAL+		CA-ML	INSTRAL+	
		(EPA-ng)	*de novo*	EPA-ng	(EPA-ng)	*de novo*	EPA-ng
50	Moderate	17%	**7%**	11%	0.18 (0.45)	**0.08 (0.31)**	0.13 (0.40)
50	High	18%	**9%**	11%	0.19 (0.43)	**0.10 (0.37)**	0.12 (0.38)
50	Very High	26%	**17%**	19%	0.37 (0.84)	**0.24 (0.72)**	0.28 (0.75)
200	Moderate	16%	**4%**	9%	0.17 (0.43)	**0.04 (0.23)**	0.10 (0.35)
200	High	17%	**5%**	8%	0.17 (0.38)	**0.06 (0.30)**	0.08 (0.31)
200	Very High	22%	**9%**	12%	0.25 (0.53)	**0.11 (0.38)**	0.15 (0.46)



200 genes

50 genes

Figure 2.5: Comparison of concatenation using ML (CA-ML) and INSTRAL (run on both *de novo* and EPA-ng gene trees) when placing on the true species tree in a leave-one-out experiment (50 species per replicate). We show the distribution of the placement error, measured as the node distance between the correct placement and the chosen placement, which is equal to the RF distance of the placement tree from the true tree. Results are shown for 200 genes (top) and 50 genes (bottom).

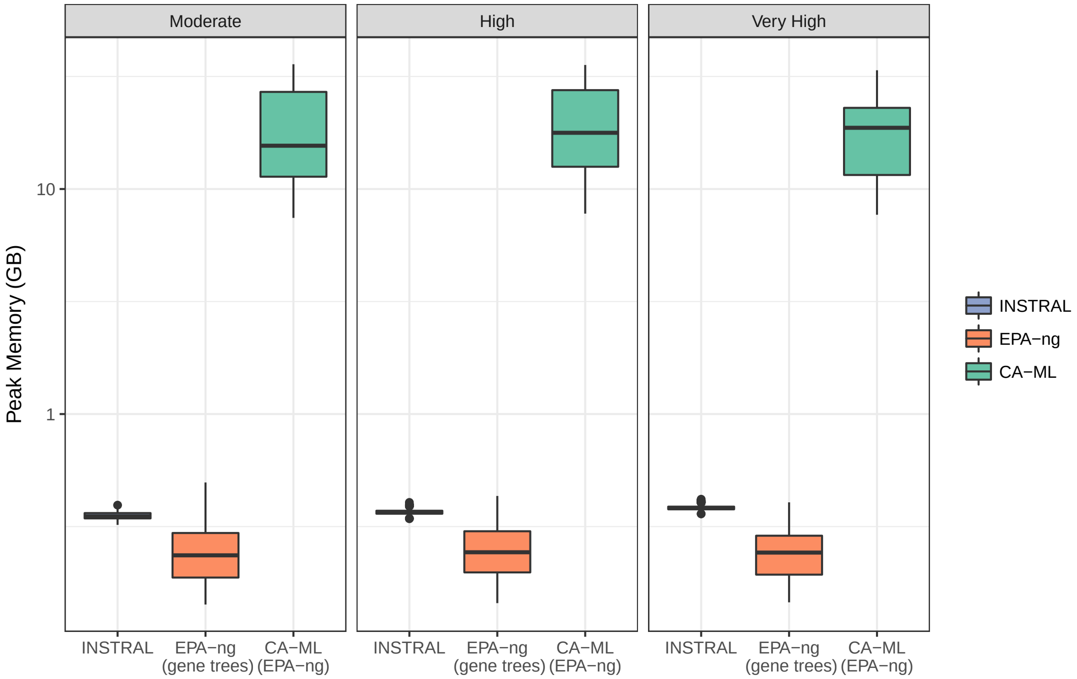


Figure 2.6: Comparison of peak memory usage of concatenation using ML (CA-ML) and INSTRAL run on gene trees updated using EPA-ng. For INSTRAL+EPA-ng, we break the peak memory usage to two steps: EPA-ng on gene trees and INSTRAL. Both method place on the true species tree in a leave-one-out experiment (50 species per replicate) with 200 genes. Results are over 2,500 placements for moderate and high and 2,350 placements for very high ILS.

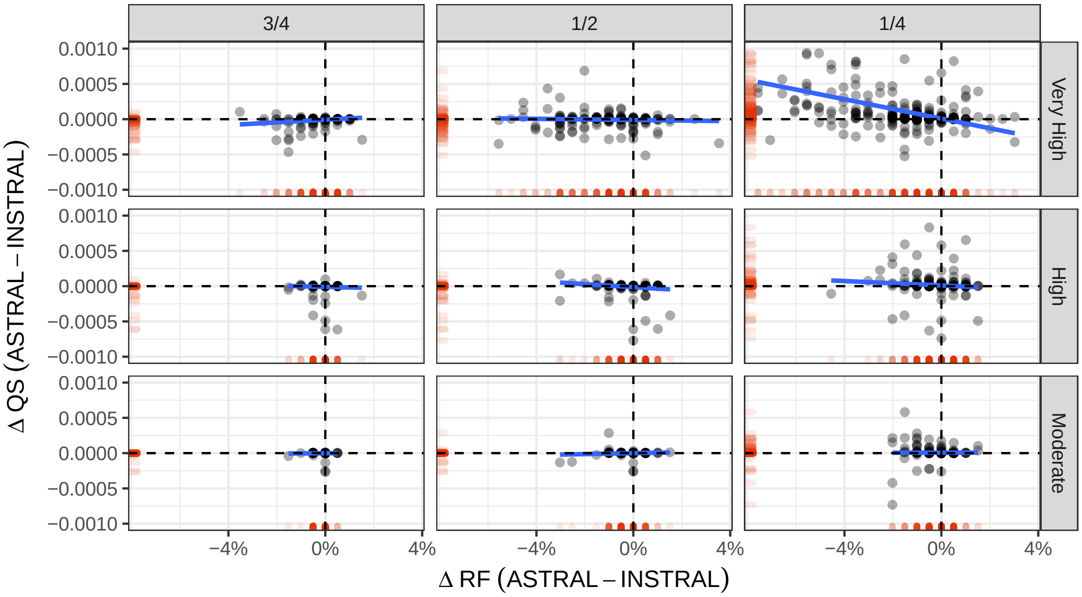


Figure 2.7: ΔQS of the output trees of the two methods based on ΔRF in all model conditions and with different backbone trees containing $1/4$, $1/2$ and $3/4$ of the species. The red tone on the x and y axes is the projection of data on both axes and represents the distribution of data on each axis.

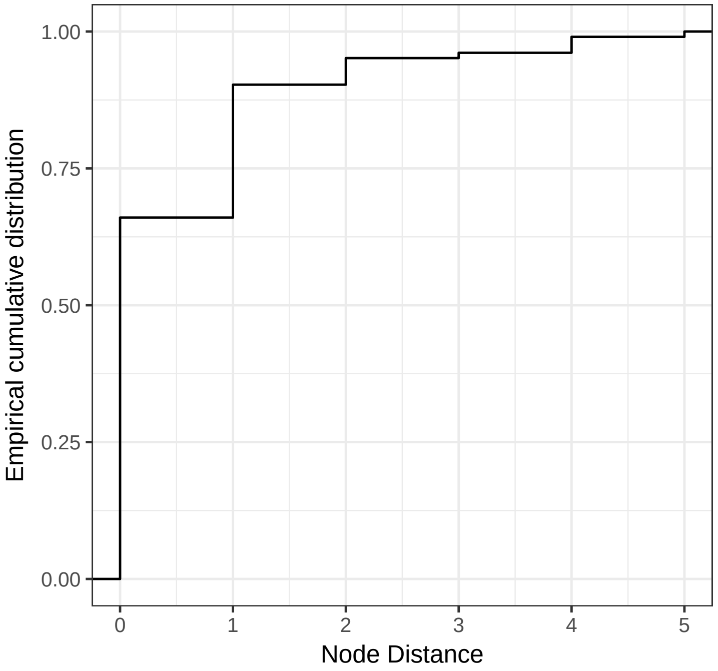
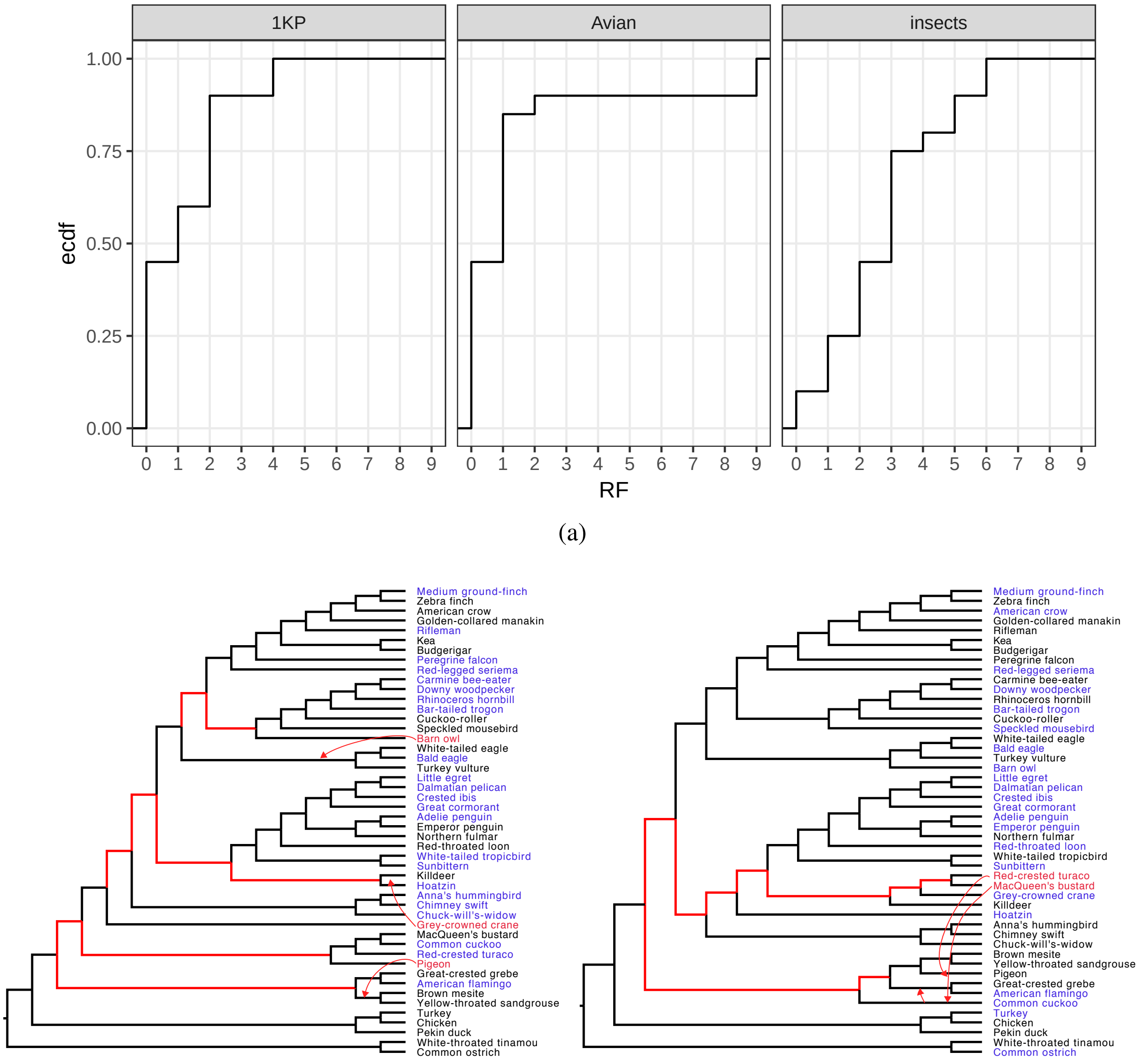


Figure 2.8: The empirical cumulative distribution function (ecdf) of the number of branch differences between concatenation using ML (CA-ML) from the original publication (file: FNA2AA.trim50genes33taxa.no3rd.unpartitioned.final.tre) and the EPA-ng (CA-ML) output on the 1KP alignment (file: FNA2AA.trim50genes33taxa.no3rd.unpartitioned.phylip).



(a)

(b)

Figure 2.9: **Results on the biological dataset.** (a) The empirical cumulative distribution function (ecdf) of the number of branch differences between running ASTRAL and INSTRAL-ordered (x-axis) on the three biological datasets. The distributions are over 20 replicates, each using half of the species, selected randomly, as backbone and placing the rest in some random order. (b) Two INSTRAL avian trees had high levels of difference from ASTRAL. Blue: backbone species. Red edges: different from ASTRAL. Red tips: The entire differences can be described by three (top) or two (bottom) rouge taxa that have moved far away from their position in the ASTRAL tree (red arrows).

Chapter 3

Forcing external constraints on tree inference using ASTRAL

Tree inference using dynamic programming is gaining increased popularity through wide-spread adoption of tools such as ASTRAL. The dynamic programming paradigm provides natural ways to restrict the search space, and practical applications of dynamic programming have required heuristic methods to define a restricted search space. However, enforcing arbitrary constraints provided by the user on the output tree is not trivially incorporated into such restrictions and requires algorithmic care. The ability to infer trees with user-defined constraints is needed for many phylogenetic analyses, but no solution currently exists for constraining the output of dynamic programming tools like ASTRAL. In this paper, we introduce methods that enable the ASTRAL dynamic programming to infer constrained trees in an effective and scalable manner. Our techniques extend recent tree completion algorithms to multifurcating input and output trees, and is both effective and fast.

3.1 Introduction

Phylogeny inference is typically formulated as an optimization problem over the space of all possible trees. The super-exponential growth of the tree topology space makes examining all topologies impossible, even for moderately large datasets. As a result, tree inference algorithms have adopted several heuristics strategies, including iterative search (e.g., hill-climbing), used by most maximum parsimony and maximum likelihood methods. However, other approaches exist.

An increasingly popular alternative is dynamic programming (DP). For an optimization score of interest, we need a recursive equation formulating how the optimal tree on a subset of leaves (or similar constructs) can be computed from the optimal trees on smaller subsets. With a recursive formulation, DP can be used to compute the optimal solution in the classic fashion (detailed below), typically implemented using memoization. Since the powerset grows exponentially with the set cardinality, this DP requires exponential running time. However, a restricted version of DP can be designed where each set is divided into only some of its subsets; the restricted DP can have polynomial running time with respect to the number of the leaves.

Phylogenetic inference using this particular DP approach has been known at least as early as 1996 [18] and has been used for many optimization criteria, including duplication and loss [55, 24, 10], deep coalescence [154], Robinson Foulds (RF) distance [157], quartet score [20, 99], and others [158]. Among these, ASTRAL [99], which estimates a species tree from a set of gene trees by minimizing the quartet distance, has found increasing popularity [95]. DP is mostly used for problems where the input is a set of trees, and the output is a tree with the minimum total distance to the input trees. The popularity of DP for these problems is perhaps because restricting the space explored by DP can be done in natural ways when the input is a set of trees. For example, the set of bipartitions observed in input trees can be used as the restriction set. More recently, ASTRAL suit of tools has introduced several heuristics to enrich the set of allowable bipartitions [100] while keeping the size of the search space polynomial [169].

The restrictions imposed on DP are not to be confused with the related concept of user-imposed constrained inference (we use “restricted” DP instead of “constrained” used in previous publications to avoid confusion). Systematists often would like to infer the best possible tree among trees that are compatible with a *constraint tree* of their choice. Finding such a tree can be considered completing and resolving the constraint tree. Constrained tree inference is needed for hypothesis-driven analyses where we are trying to choose the best among a set of hypotheses available by prior knowledge [92, 117, 153, 5]. Constrained searches can help in model selection; e.g., for testing whether a polytomy [118] or the monophyly of a group [21] can be rejected. Similarly, they can help gauge the “hidden” support for branches not recovered in the main analysis. Moreover, constrained searches have been successfully used to combine the results of multiple methods [33]. More recently, constrained trees were used in taxonomic profiling [107]. Finally, constrained searches enable updating existing trees without recomputing trees from scratch. For these reasons, most phylogenetic inference tools allow constraints.

To our knowledge, the DP paradigm has not been adopted to perform constrained searches with user-defined constraints. Performing constrained searches in the DP paradigm may appear easy: one needs to make sure the restricted set of bipartitions explored by DP are consistent with the constraints. As we show, there are roadblocks when the user-provided input is allowed to be arbitrary. The challenge is to find a large-enough search space that satisfies the user-provided constraints. Here, building on two recent advances [29, 8], we propose an algorithmic solution to this challenge. We implement our solution inside the ASTRAL software for species tree inference, thereby enabling it to perform constrained searches for the first time. In extensive tests, we show that the constrained searches remain as accurate as unconstrained searches while reducing the running time, can improve accuracy in the presence of external knowledge about individual relationships, and can reveal hidden support.

3.2 Method

Our goal is to extend ASTRAL so that it can honor a user-provided constraint tree. Before describing our algorithm, we review ASTRAL and an RF-based tree completion algorithm used in our method.

Notations. We are given a set of k (potentially multifurcating) input trees $\mathcal{T}$ on (subsets of) a leafset $\mathcal{L}$ of size n. We are also given a (multifurcating) constraint tree, $\bar{T}$ on a subset of $\mathcal{L}$. Let $l(t)$ be the set of leaves of a tree and let $l(u)$ be the set of leaves below a node u. We use $s(u)$ to denote the sister of a node (i.e., the set of all nodes sharing a parent with u). Let $\mathcal{L}' = \mathcal{L} \setminus \{o\}$ where $o \in \mathcal{L}$ is an arbitrarily chosen species. Denote $A \subset \mathcal{L}'$ as a cluster. Each edge in an unrooted tree corresponds to a bipartition of leaves, which corresponds to a cluster (the side missing o). A cluster A (i.e. the bipartition $A|\mathcal{L} \setminus A$) is called compatible with a tree T iff a tree exists that includes the bipartition and induces a resolution of T when restricted to same leaves as T. Two clusters are compatible iff they can be in the same tree [160].

3.2.1 Background: Tree Completion

Completing and resolving a tree based on a reference tree is a well-studied problem and is often formulated as minimizing the distance to the reference tree while maintaining compatibility with the original tree [10, 29, 8, 28]. A natural objective is to find a complete tree with the minimum RF distance (i.e., the total number of branches that differ between the two trees) to the reference tree [73]. OCTAL was the first quadratic time optimal solution to this RF completion problem [29]. Bansal later introduced a linear time solution [8], which we call B-RF(+) algorithm. Both methods take as input an incomplete backbone tree, T_b, and a complete and binary reference tree, T_r, and output a binary and complete tree compatible with T_b such that the RF distance to T_r is minimized among all allowable trees.

The B-RF(+) algorithm [8] achieves linear time solutions using constant time least-

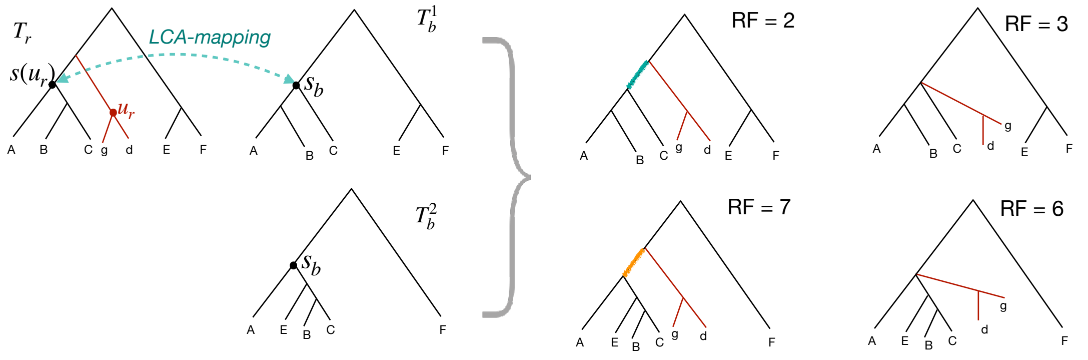


Figure 3.1: Updates need for the tree completion algorithm. T_b^1 and T_b^2 are both completed based on T_r, generating either a binary tree or a multifurcating tree. In case 1 (T_b^1), subtree under u_r should be added as sister to s_b to minimize RF; the green branch matches T_r, but creating a polytomy would result in a false negative. In case 2 (T_b^2), subtree under u_r should be added as a polytomy under s_b; otherwise, the new orange branch will be a false positive. In the second case, restricting the output to be binary (as in the B-RF(+) algorithm) leads to suboptimal RF distances.

common-ancestor (LCA) lookups made possible after a linear time preprocessing using the Schieber–Vishkin technique [139]. Both trees are rooted on an arbitrary shared leaf. For every fully-missing node u of T_r (i.e., a node where $l(u) \cap l(T_b) = \emptyset$), the subtree below u is added intact as the sister to the LCA of $l(s(u))$ inside T_b (Fig. 3.1). This placement of the subtree below u preserves all its bipartitions, the bipartition above it, and potentially the bipartition above the parent of u (we will come back to this point). The order of additions to T_b is determined by a pre-order traversal of T_r, adding each fully-missing node u when we visit the parent node of u. Note that the topology of the backbone tree will not change by the addition of new subtrees. Bansal proved this simple algorithm minimizes the RF distances between T_r and any possible binary output tree that is compatible with T_b.

Background: DP algorithm implemented in ASTRAL

ASTRAL estimates an unrooted (species) tree given a set of unrooted (gene) trees $\mathcal{T}$ and is statistically consistent under the multi-species coalescent model [115] of incomplete lineage sorting (ILS) given a sample of true gene trees. ASTRAL seeks the tree T with the maximum quartet score to $\mathcal{T}$ defined as $\sum_{t\in\mathcal{T}}|Q(T)\cap Q(t)|$, where $Q(.)$ is the set of quartet topologies of a tree. Let $S(A)$ be the score for an optimal subtree on the cluster A. Defining $S(\{x\}) = 0$ for $x \in \mathcal{L}$, the recursion is:

$$S(A) = \max_{A'\in X, A\setminus A'\in X} S(A') + S(A\setminus A') + w_{\mathcal{T}}(A'|A\setminus A'|\mathcal{L}\setminus A) \tag{3.1}$$

where X is a set of clusters and $w_{\mathcal{T}}$ is a function assigning weights to tripartitions of $\mathcal{L}$ such that the sum of all weights for any tree gives its quartet score. If X is set to $2^{\mathcal{L}'}$, the recursion tests all ways of dividing A into two smaller clusters and under this condition, $S(\mathcal{L}')$ (Eq. 3.1) gives the optimal quartet score [99] for $\mathcal{T}$ in time growing exponentially with n (expected, as the problem is NP-Hard [74]).

Forming set X: heuristics and restrictions

To handle large datasets, we need X to have a manageable size, preferably growing polynomially with n and k. At the same time, we ideally want X to have all clusters of the optimal tree. An obvious way of building X is to set it to all clusters in all trees in $\mathcal{T}$, hoping that all clusters in the optimal tree appear in at least one input. However, two difficulties emerge. Firstly, simulations under very high levels of gene tree discordance have shown this heuristic to be insufficient as bipartitions in the optimal tree can frequently be absent from gene trees [100]. To deal with this issue, starting from ASTRAL-II, set X is enhanced using a set of heuristic methods, and since ASTRAL-III, the size of X is restricted to grow linearly with n and k [169, 123]. These heuristics (among other techniques) build consensus trees from input trees and add resolutions of polytomies of consensus trees to X.

The second difficulty is having full resolutions. Equation 3.1 is well-defined only if for every non-singleton $A \in X$, there is $A' \subset A$ such that $A' \in X$ and $A\setminus A' \in X$. More generally, a cluster A in X is useful *only if* there exists a fully binary tree on $\mathcal{L}'$ that includes A and all of its clusters are in X. Including any other cluster in X is a waste of computation. Thus, set X (which needs to be non-empty) needs to satisfy a main property (recall $o \notin A \subset \mathcal{L}'$ and $2n-3$ is the number of clusters in a fully resolved tree):

P1: $\forall A_1 \in X, \exists \{A_1, A_2, \ldots, A_{2n-3}\} \subset X$ s.t. $\forall (i, j) : A_i$ is compatible with A_j.

Building X using bipartitions of input trees $\mathcal{T}$ can fail to satisfy both of these properties unless all trees are complete and binary. Thus, starting from ASTRAL-II, three steps are taken. i) Before adding bipartitions from $\mathcal{T}$ to X, it first completes each tree with respect to other trees using a distance matrix computed from quartet frequencies in $\mathcal{T}$ and an algorithm based on the four-point condition [94]. ii) Polytomies in input trees are resolved once [100] or more [169] using heuristic methods that sample leaves around polytomies and use the distance matrix mentioned earlier. iii) Heuristic enhancements of set X employed in ASTRAL-II and ASTRAL-III are all explicitly designed such that P1 is automatically satisfied, a feat that has been particularly challenging for multi-individual datasets [123]. Thus, in effect, the set X includes all clusters from each tree in a set of binary and complete trees (including modified input trees and others that ASTRAL heuristically selects.)

3.2.2 Algorithms for Enabling Constrained Search in ASTRAL

Given a constraint tree $\bar{T}$ and a set of gene trees, $\mathcal{T}$, our goal is to find the tree among all trees compatible with $\bar{T}$ that has the maximum quartet score with respect to $\mathcal{T}$. Compatibility with $\bar{T}$ is achieved if we enforce a third property on X.

P2: $\forall A \in X : A$ is compatible with $\bar{T}$.

Existing methods for forming X are not guaranteed to satisfy P2. One may think that we can follow standard methods of forming X and simply refuse to add clusters when they violate

P2. Unfortunately, that approach, in addition to being slow, can violate property P1 and is not viable. Thus, the main challenge in building set X is maintaining the P1, in addition to P2, and doing so in a scalable fashion.

Forming set X using tree completion

The main algorithmic trick we introduce depends on completing and resolving the tree $\bar{T}$ using every input tree $t \in \mathcal{T}$. We require a tree completion method $Comp(T_b, T_r)$ that adds to T_b leaves that are present in T_r but are absent from T_b. The algorithm should only add missing leaves to T_b and can also resolve (some of) its polytomies. In other words, the output restricted to leaves of T_b is a resolution of T_r; thus, $Comp(\bar{T}, t)$ will be compatible with $\bar{T}$.

Tree completion/resolution methods were traditionally proposed for completing a gene tree using the species tree [29, 10, 8]. However, in our algorithm, we turn the problem on its head and complete the constraint species tree $\bar{T}$ using individual gene trees and use this mixed tree to build the set X. This uncommon use of the completion method is the main algorithmic idea that enables us to satisfy P3.

Given a $Comp(T_b, T_r)$ method, we propose Algorithm 1 for forming set X. The first step is identical to ASTRAL-III, where, gene trees are completed in a reference-free fashion with respect to each other using a distance-based method (any method can be used). Step 2 is the primary new step and forces gene trees to be compatible with $\bar{T}$ (by definition of $Comp(T_b, T_r)$). Step 3, like ASTRAL-III, creates consensus trees $\mathcal{C}$ with polytomies from $\mathcal{T}'$; resolutions of these trees are used to enrich X. We use the compatible gene trees $\mathcal{T}'$ to compute consensus trees, and Step 4, like Step 2, forces consensus trees to be compatible with $\bar{T}$. Thus, all trees in $\bar{\mathcal{T}}'$ and $\bar{\mathcal{C}}$ are compatible with $\bar{T}$; any resolution of these trees (done in Step 5 using methods existing in ASTRAL-III) is compatible with $\bar{T}$. Thus,

Claim 1. *All bipartitions of X created by Algorithm 1 are part of a fully binary and complete tree (P1) and are compatible with the constraint tree $\bar{T}$ (P2).*

1. $\mathcal{T}' = \{$Completed t with reference to $\mathcal{T}$ $|t \in \mathcal{T}\}$
   (completing gene trees can use any method; e.g., distance-based method of ASTRAL [94])
2. $\bar{\mathcal{T}}' = \{Comp(\bar{T}, t) | t \in \mathcal{T}'\}$
3. $\mathcal{C} = \{$consensus trees computed from completed trees $\bar{\mathcal{T}}'\}$
   (consensus trees should only include bipartitions included in $\bar{\mathcal{T}}'$)
4. $\bar{\mathcal{C}} = \{Comp(\bar{T}, t) | t \in \mathcal{C}\}$
5. $\mathcal{T}^X = \bigcup_{\bar{t} \in (\bar{\mathcal{T}}' \cup \bar{\mathcal{C}})} \{$ one or more binary resolutions of $\bar{t}$ $\}$
   (resolving polytomies can use any algorithm; e.g., sampling methods of ASTRAL-III [169])
6. Root all trees from $\mathcal{T}^X$ at o and add all their clusters except $\{o\}$ to X.

Algorithm 1: **FormX($\mathcal{T}$,$\bar{T}$)** algorithm for computing set X from input gene trees $\mathcal{T}$ such that every cluster in X is compatible with $\bar{T}$.

After forming X (Algorithm 1), DP proceeds as before, computing $w_{\mathcal{T}}$ using the original gene trees $\mathcal{T}$. Since all bipartitions in X are compatible with $\bar{T}$, any tree formed by DP will be compatible with $\bar{T}$; thus, no other changes are needed.

Tree completion with non-binary input/output

We now describe our choice for the $Comp(T_b, T_r)$ method. We base our solution on the B-RF(+) algorithm [8] described earlier, which we re-implemented inside ASTRAL. However, several changes to the algorithm were necessary.

Multifurcating output. The B-RF(+) algorithm and OCTAL force the output to be binary. As a result, the output can include arbitrary branches that increase false positive edges (branches in the output missing from T_r) without reducing false negative edges (branches in T_r missing from the output) (Fig. 3.1). Thus, if the output tree is allowed to be multifurcating, neither algorithm is optimal (shown by a counter-example; Fig. 3.1). As mentioned earlier, ASTRAL has several heuristics to resolve polytomies in the input trees (Step 5 of Alg. 1), and these heuristics are preferable to an arbitrary resolution. Thus, we changed the B-RF(+) algorithm so that $Comp(T_b, T_r)$ avoids adding arbitrary resolutions in Steps 2 and 4, leaving resolving polytomies to heuristics of Step 5. Our experiments show that this change substantially reduced the RF of completed trees (Fig. 3.8).

Recall that the B-RF(+) algorithm adds each fully-missing node u_r in T_r as sister to the LCA of $l(s(u))$ in T_b; denote this LCA node as s_b. Let $\mathcal{M} = l(T_r) \setminus l(T_b)$ be the set of leaves missing from T_b. Two cases arise.

Case 1. $l(s(u_r)) \setminus \mathcal{M} = l(s_b)$. In this case, the optimal placement of the subtree below u_r in T_b is as sister to s_b, creating a new node above s_b. The reason is that this placement leads to the bipartition identified by $s(u_r)$ to be identical between T_r and the completed tree, thereby avoiding a false negative edge. **Case 2.** $l(s(u_r)) \setminus \mathcal{M} \neq l(s_b)$. Here, no placement of the subtree below u_r onto T_b can avoid the false negative penalty associated with missing the bipartition associated with $s(u_r)$ in T_b. However, placing the subtree as sister to s_b by creating a new internal node does lead to an unnecessary false positive edge in the completed tree, separating $l(s_b)$ from other leaves (Fig. 3.1). To avoid these false positive edges, we can simply create a polytomy in the completed tree by putting the new subtree as another child of s_b.

The change to the B-RF(+) algorithm is straight-forward. We compute the LCA mapping both ways. When inserting the u_r subtree into T_b at s_b, we check if the LCA of s_b in T_r matches $s(u_r)$. If it does, we create a new internal node above s_b; otherwise, we create a polytomy in T_b by adding the subtree as a child of s_b.

Note that every edge in the output of the new algorithm (T^n) will also be present in the output of the original B-RF(+) algorithm (T^o), which is proved to be optimal among all binary trees. By allowing multifurcating output, we can only hope to reduce FP edges. To see the justification for our algorithm, assume T^o is unique. Then, the optimal multifurcating tree should be a contraction of T^o where every FP edge is contracted unless contracting the FP edge prevents the output tree from inducing T_b. T^n is a contraction of T^o. From the description of Case 2, it is clear that every edge contracted in T^n compared to T^o is a FP edge. Also, every edge remaining in T^n is either shared with T_r (those from Case 1) or are otherwise those that existed (minus new taxa) on T_b and thus are needed to ensure the output induces T_b. Thus, it follows that T^n has the optimal score among multifurcating trees that are contractions of T^o. However, note that our analysis does not prove that when multiple optimal binary trees exist, results of the new algorithm are still optimal.

Multifurcating backbone. The B-RF(+) algorithm is defined for binary T_b, but we can have multifurcating T_b (here, $\bar{T}$). To adopt the algorithm to multifurcating backbones, prior to completion, we need to add to T_b those bipartitions in T_r that are compatible with T_b (or else we will have unnecessary false negatives). This can be done using the same LCA mapping of the B-RF(+) algorithm. In a post-order traversal of T_r, for every node u that maps to a polytomy v in T_b, we check whether all children of u have a LCA mapping to a child of v. If they do, we create a new node below v and move mapped children under v to be children of the new node. It's easy to see this method adds missing bipartitions from T_r to T_b.

Multifurcating reference. Changing the B-RF(+) algorithm to handle multifurcating T_r is simple. In the pre-order traversal, for any polytomy node u_r encountered in T_r, when there are multiple fully-missing nodes under u_r, we add them as a group to the same position (as a polytomy) in T_b. Other cases are naturally handled by the LCA mapping used by the B-RF(+) algorithm if the definition of a sister node is updated to mean the set of nodes sharing a parent with a node.

3.3 Results: simulations

We first test constrained ASTRAL on an existing [100] simulated dataset with 201 species. This SimPhy [89] dataset has three model conditions with moderate, high, or very high levels of ILS, controlled by setting the maximum species tree height to 10^7, 2×10^6, or 5×10^5 generations; mean RF distance (normalized by total number of branches in both trees) between the true species tree and true gene trees are 15%, 34%, and 69%, respectively. We use gene trees inferred using FastTree-II [120] from sequence data in our analysis. The estimated gene trees have relatively high levels of gene tree error(the average RF distance between estimated and true gene trees are 25%, 31%, and 47% for the three model conditions). Following previous publications [100], three replicates are removed from high ILS dataset due to lack of resolutions in gene trees.

We compare both constrained and unconstrained ASTRAL to the true tree. We measure the topological error using the normalized RF distance, and also report the change in quartet score and running time between constrained and unconstrained ASTRAL. Note that quartet score of the constrained tree can be higher than unconstrained ASTRAL, as default version of ASTRAL has a heuristic definition of X and is not guaranteed to find the optimal solution. We ask whether our method of forming the constrained and restricted set X is effective in providing the same level of accuracy as unconstrained (but restricted) searches while improving the running time. We then ask if the use of constraints can benefit accuracy.

3.3.1 Is our formation of X restricted to a constraint tree $\bar{T}$ sufficient?

Constraint tree $\bar{T}$ with missing leaves

We built constraint trees that include $\frac{1}{4}$, $\frac{1}{2}$, or $\frac{3}{4}$ of species by taking the ASTRAL-III tree on the full dataset and pruning leaves uniformly at random. Since the unconstrained tree induces the constraint tree, the relative accuracy of constrained and unconstrained search is entirely a function of the completeness of X.

The accuracy of the constrained ASTRAL in most condition matches that of the unconstrained ASTRAL (Fig. 3.2). For moderate (10^7) and high (2×10^6) levels of ILS, the drop in average accuracy for different numbers of genes never exceeds 0.4%. Only with very high ILS (5×10^5) and only if our constraint tree includes only 50 species, do we start to see small but noticeable drops in the accuracy of constrained ASTRAL. For example, with 50 genes and very

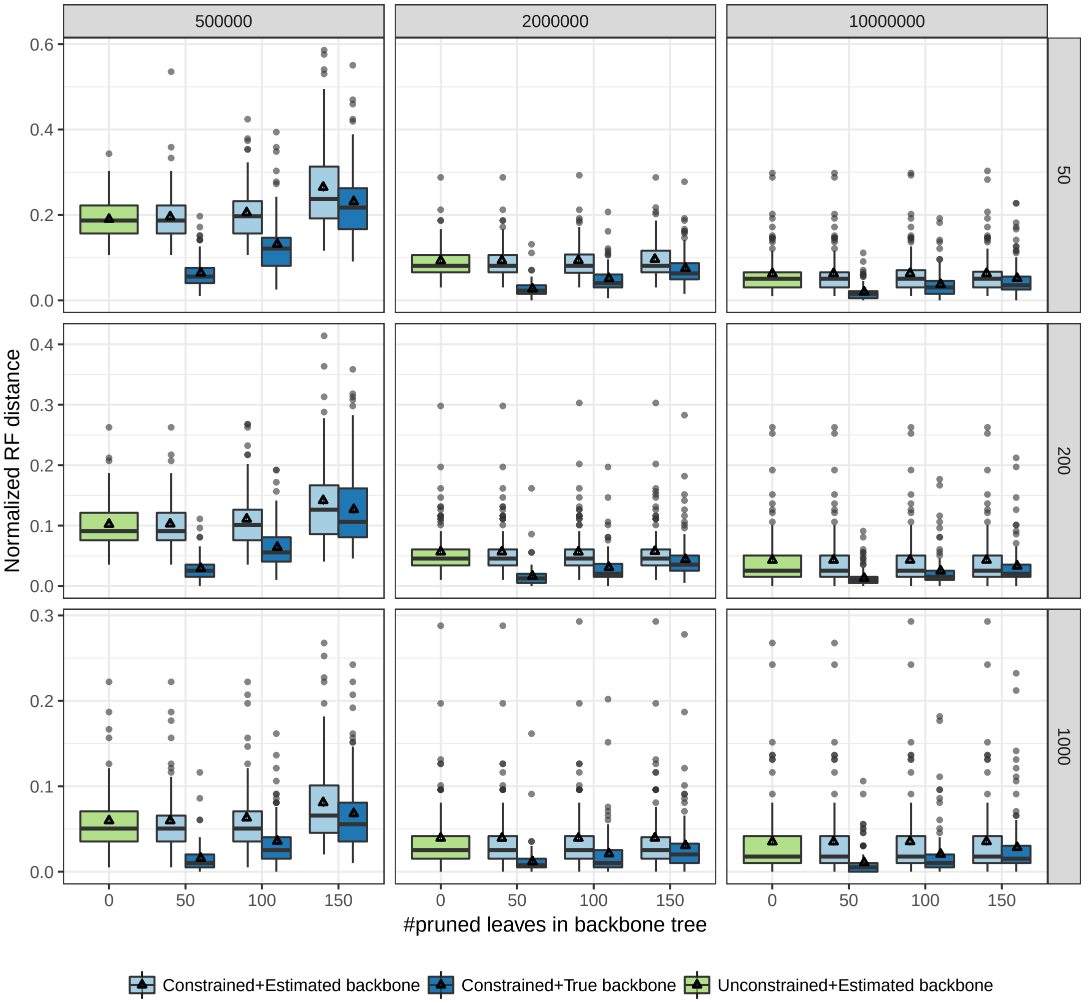


Figure 3.2: Effectiveness of constrained ASTRAL with constraint trees that have randomly distributed missing leaves. ASTRAL-III species trees are compared with and without constraints using the Normalized RF distance between inferred species tree and true species tree. Boxplots show distribution (over 100 replicates) and triangles show the mean. There are three different levels of ILS (500K, 1M and 2M generations, corresponding to very high, high, and moderate ILS, respectively) and varying number of genes (50, 200, 1000). The constraint trees are obtained by pruning 50, 100, or 150 (x-axis) randomly chosen leaves from the unconstrained ASTRAL tree or the true species tree.

high ILS, ASTRAL with no constraints has 20% error whereas with $\bar{T}$ of 50 leaves it has 26% error. Consistent with patterns of accuracy, the quartet score of the ASTRAL tree also remains largely unchanged in most cases, except, again, for the case of very high ILS and backbone trees that include only 50 leaves (Fig. 3.9).

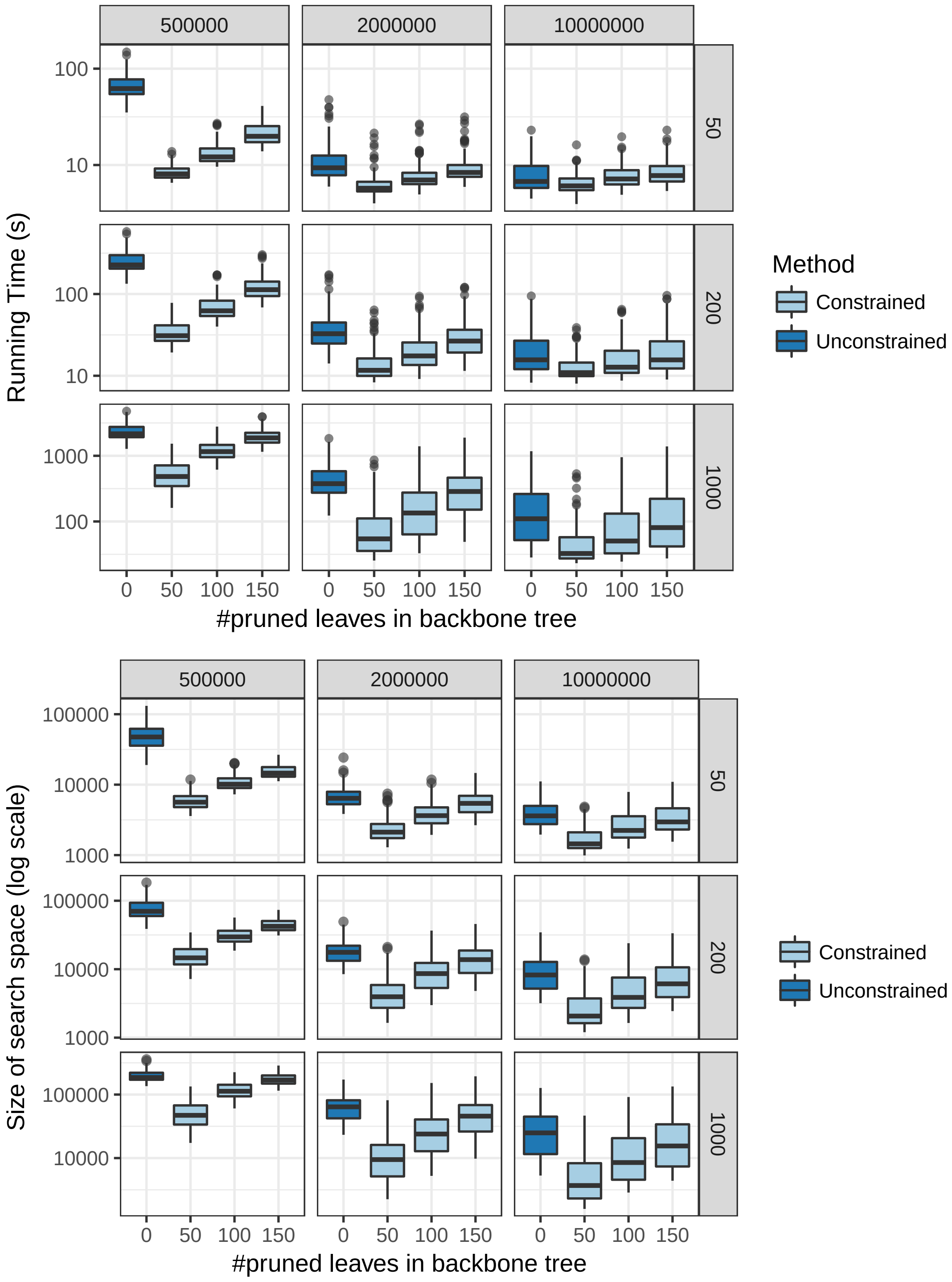


Figure 3.3: Impact of constrained search on the running time and search space. The running time (top) and search space size (bottom) are compared between constrained and unconstrained ASTRAL-III. Other settings of the figure are identical to Fig. 3.2.

It is instructive to see how constrained searches impact the running time and the search space X (Fig. 3.3). With very high ILS and $\bar{T}$ including 150 of leaves (pruning 50), we get from 4 to 8x improvement in running times and substantial reduction in the size of the search space. This reduction explains the small reduction in the accuracy of ASTRAL-III with constrained searches under these conditions. As backbone size becomes smaller, the running time converges to unconstrained ASTRAL; however, even when $\bar{T}$ includes a quarter of the leaves, we still have 1.2 to 3x improvement in the running time. With moderate and high levels of ILS, improvements are less pronounced but still substantial. Running time improvements mirror the change in the search space size, which is dramatically reduced (Fig. 3.3).

We also study a scenario where missing leaves in the constraint tree form clades instead of being uniformly distributed. Results of the clade-based removal did not substantially differ from the random removal (Fig. 3.10). With moderate or high ILS, random and clade-based removal were indistinguishable, and for very high ILS, only small differences were observed.

To summarize, our method of forming X for constraints with missing leaves retains accuracy and reduces running time, with only small reductions in the accuracy in the most extreme conditions (very high ILS and small constraint trees).

Constraint tree $\bar{T}$ with multifurcation

We next collapse randomly chosen branches from the unconstrained ASTRAL-III tree to create a complete but unresolved constraint tree. With these multifurcating constraint trees, constrained ASTRAL search is as accurate as the unconstrained ASTRAL even for very high levels of ILS (Fig. 3.4). Differences in mean accuracy are no more than 0.1% in any of the 27 conditions we tested. Remarkably, in the case of very high ILS, we even see a small but noticeable improvement in quartet score (but not accuracy) when the constraint tree includes only 50 branches (Fig. 3.11). Once again, the running time and the size of the search space both reduce dramatically in the constrained searches (Figure. 3.12). Thus, our method of forming X is

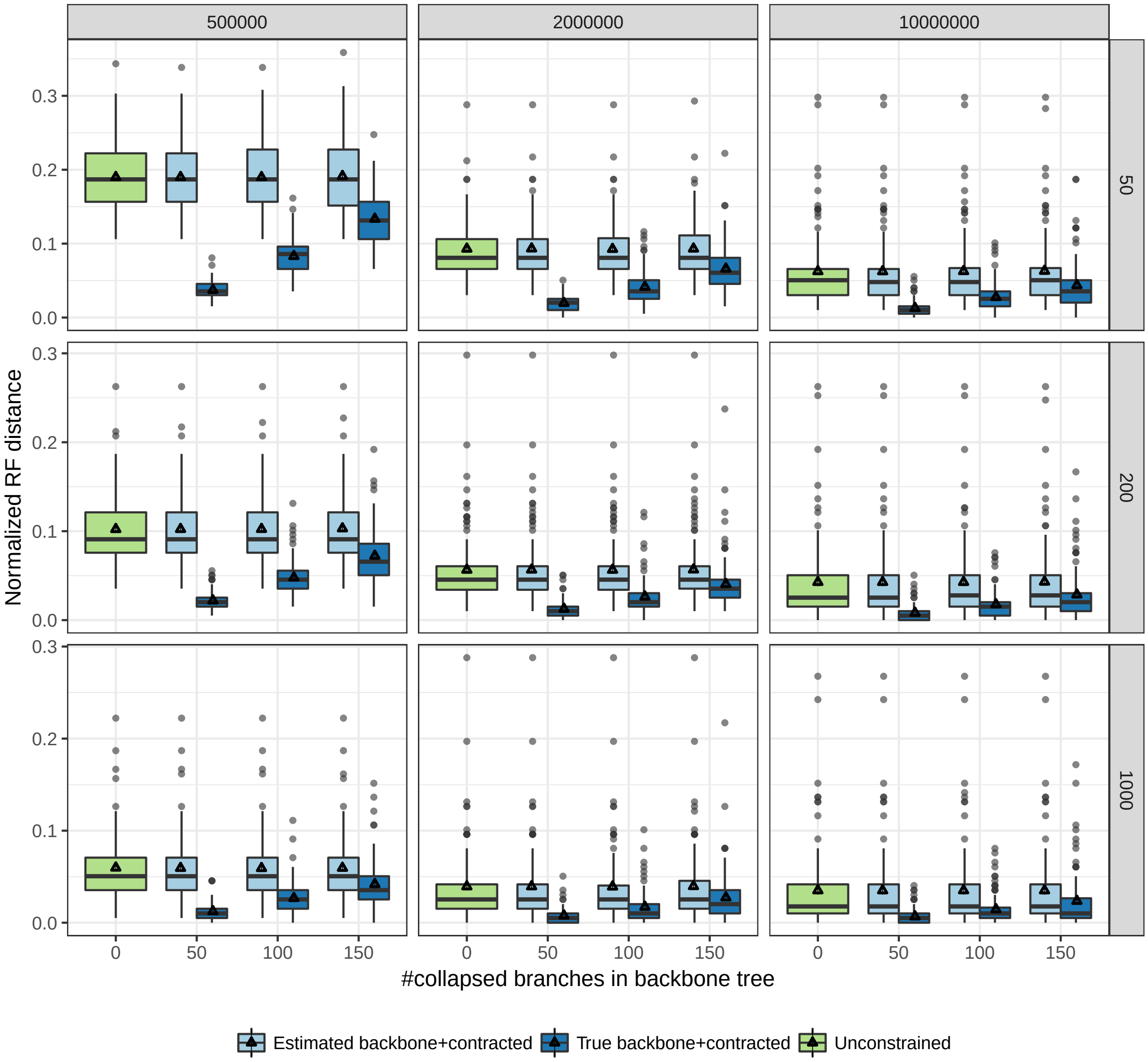


Figure 3.4: Effectiveness of constrained ASTRAL with constraint trees that have randomly distributed contracted branches. Settings are similar to Fig. 3.2. The constraint trees are obtained by collapsing 50, 100, or 150 (x-axis) randomly chosen branches from the unconstrained ASTRAL tree or the true species tree. Note that the set of branches contracted from the true and estimated species trees are not identical.

effective in the face of multifurcating constraint trees.

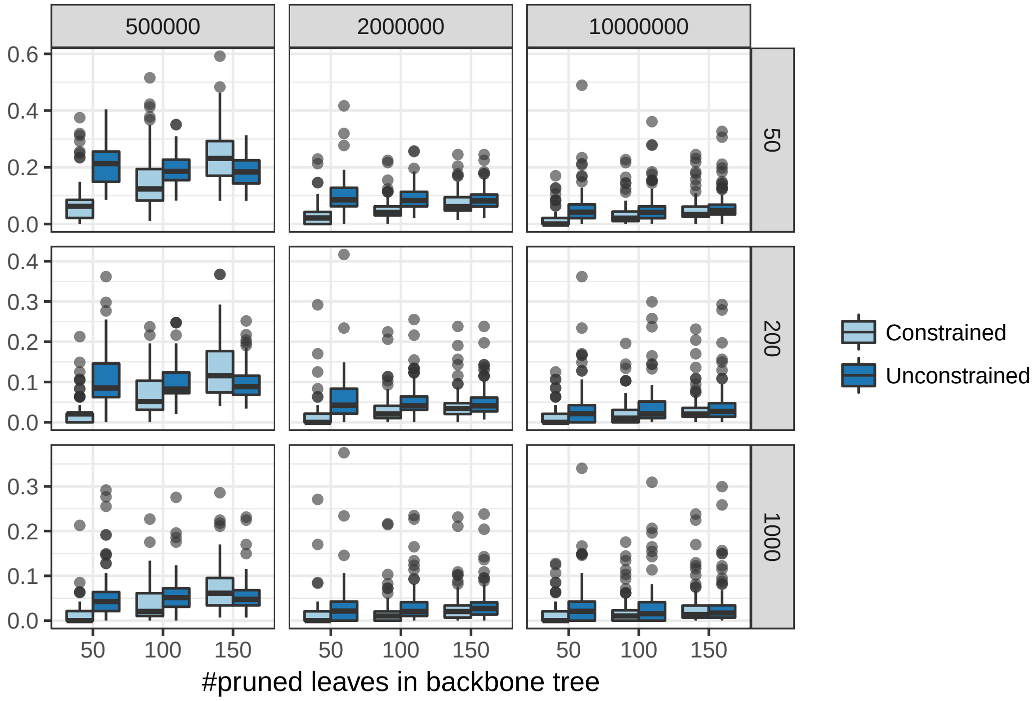


Figure 3.5: Constraints on some parts of the tree can improve other parts of the tree. The RF distance between the constrained or unconstrained ASTRAL trees and the true tree, all restricted to the set of leaves not present in the constraint tree. Results are for the dataset with constraint trees defined by pruning 50, 100, or 150 leaves from the true tree.

3.3.2 Can a constrained search help accuracy?

Constrained searches have the power to improve accuracy if prior knowledge of parts of the tree is available. To test this proposition in simulations, we study the accuracy of constrained ASTRAL when the constraint tree $\bar{T}$ is a subset or a contraction of the *true* species tree. In both cases (Figures 3.2 and 3.4), the accuracy of the ASTRAL tree improves, and changes are dramatic when the constraint tree is missing only 50 leaves or branches. The improvements are especially strong for the case of complete but multifurcating true species trees (Fig. 3.4) where a constraint tree with only 50/198 branches can reduce the error from 19% to 13% with 50 genes with very high ILS. If $\bar{T}$ includes 150/198 branches, the error reduces down to 4%.

The dramatic improvements in accuracy are perhaps not surprising given the fact that parts of the tree are fixed to match the true tree. More interesting is whether adding constraints to some

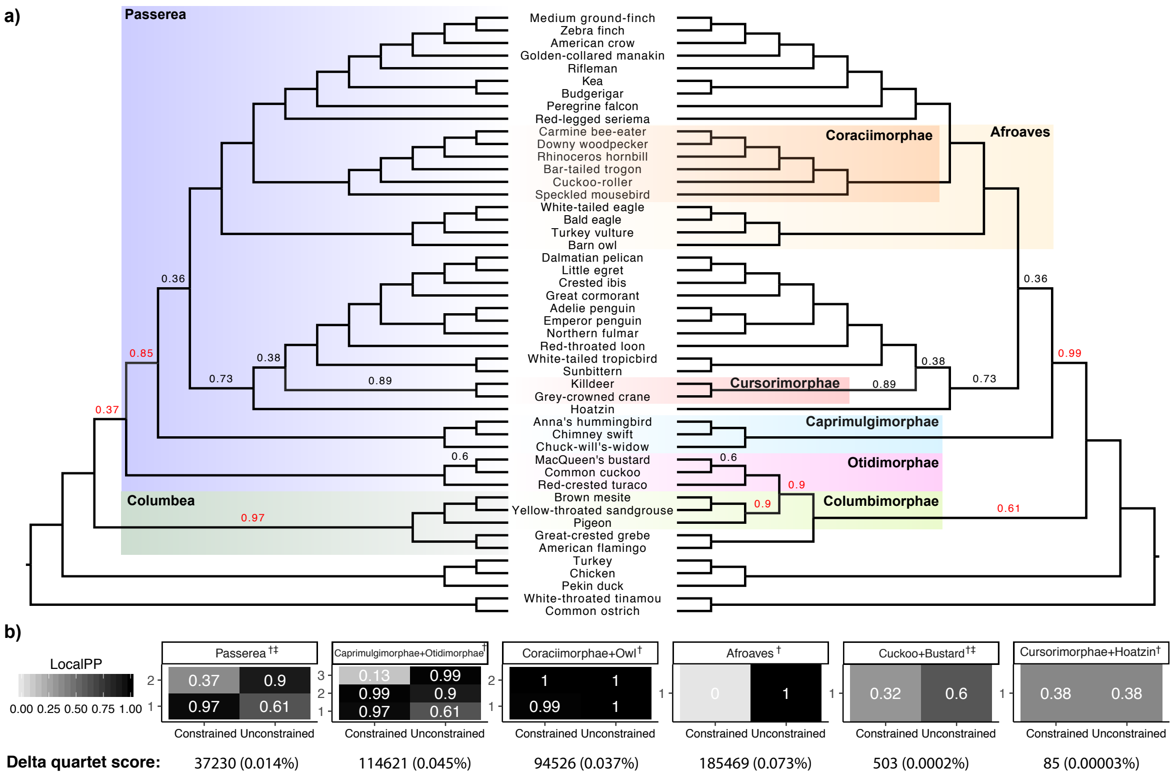


Figure 3.6: Constrained analyses reveal support for alternative clades. (a) On the avian genomics dataset [64], we estimated ASTRAL-III trees with no constraints (right) using 2022 binned (super)gene trees. The tree did not include the Passerea vs. Columbea division that most other analyses of this dataset reveal. Constraining the ASTRAL tree to include Passerea as a clade resulted in a tree (left) with two new branches and 0.97 localPP support for Columbea. LocalPP support values below 1.0 are shown on branches (red: change in support across the two trees). (b) Similar to (a), constrained analyses are performed to find support for five other clades found in the TENT (†) and (in some cases) in MP-EST* (‡) but not found in the unconstrained analyses. For each clade, we show localPP support for branches that differ between the constrained and unconstrained trees (one to three branches change in constrained searches). We also show the reduction in the quartet score in the constrained analyses as an abosulte number and percentage.

part of the tree improves the accuracy of the *remaining* parts of the tree. We thus evaluated the accuracy of trees only restricted to the leaves that are not part of the constraint tree. We observe that the accuracy of the remaining leaves has also increased dramatically as a result of having constraints (Figure 3.5). For example, in the very high ILS case with 50 genes, when 50 species are missing from the correct constraint tree, the error for the placement of these 50 species has reduced from 21% with no constraints to only 7% with constraints. Similarly strong levels of

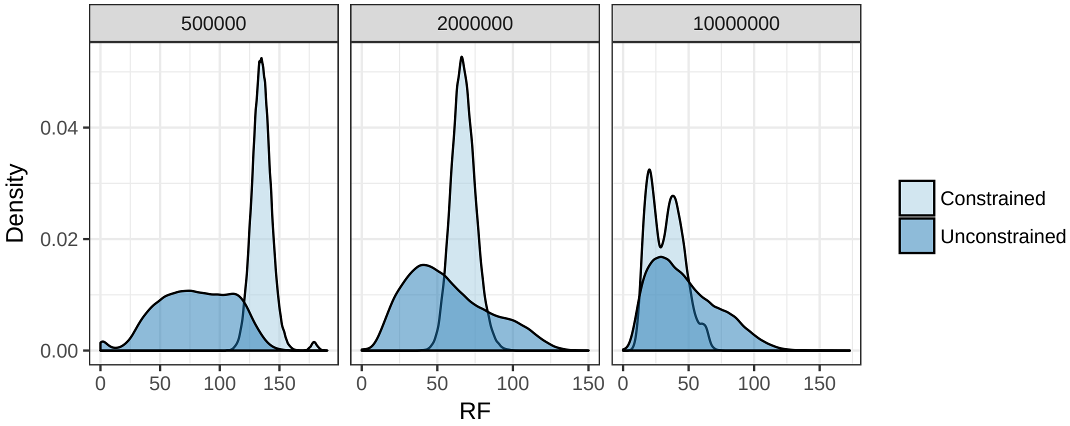


Figure 3.7: Accuracy of hybrid gene trees. We resolve the ASTRAL-III tree with branches that have support ≤ 0.99 contracted (as T_b) using each gene tree (as T_r) using our extended tree completion algorithm and call the resulting tree a hybrid gene tree. Density plots show the error of all 1000 hybrid (Constrained) and original (Unconstrained) gene trees, measured using RF distance to true gene trees (out of 396).

improvement are observed across all conditions, except when the constraint tree includes only 50 leaves. To summarize, the result demonstrates that given correct prior knowledge about parts of the phylogeny, a constraint ASTRAL search can improve the accuracy of the remaining parts.

3.4 Results: biological dataset

We reanalyze the avian phylogenomic dataset [64] with 48 bird species and more than 14,000 loci. The statistical binning method has been proposed to enable coalescent-based analyses of this dataset despite the low phylogenetic signal [96]. The main novel result found using this dataset is the division to Passerea and Columbea at the base of Neoaves, a relationship that was recovered in most analyses of the dataset, including concatenation (TENT), MP-EST [83] run on binned gene trees (MP-EST*), and ASTRAL run on unbinned gene trees with low support branches contracted [169]. However, running ASTRAL on 2022 binned gene trees failed to recover Passerea and Columbea and placed Otidimorphae (a clade within Passerea) within Columbea (Fig. 3.6). Nevertheless, the localPP support [136] for this alternative relationship is low.

Thus, using constrained searches, we now ask whether there is support for Columbea/Passerea in the binned gene trees.

Constraining the ASTRAL tree to include Passerea results in recovering Columbea and placing Otidimorphae as sister to other Passerea. The Columbea clade, absent from unconstrained ASTRAL, has high support (0.97 localPP) in the constrained tree. Moreover, the support of the Columbimorphae, a clade universally supported in modern analyses, increases from 0.9 in the unconstrained tree to 1.0. On the other hand, the localPP support for Passerea is only 0.37, which is barely above the expected support of a random resolution (0.33), and the total quartet score of the tree is reduced by 37230 quartets (0.015%). We then performed another constrained analysis forcing Otidimorphae to be with Caprimulgimorphae (as in TENT). This constraint leads to Passerea and Columbea both becoming monophyletic with 0.99 and 0.97 localPP (Fig. 3.6b). However, the Otidimorphae+Caprimulgimorphae clade itself has low localPP (0.13) and total quartet score reduces by 0.045%. Overall, while the unconstrained ASTRAL tree does not recover Columbea and Passerea, gene trees strongly support Columbea (if not Passerea).

We next tested four other clades recovered in TENT but absent from the unconstrained ASTRAL (Fig. 3.6b). Several patterns were observed. Forcing Afroaves to be monophyletic reveals a total lack of support for the monophyly of that clade (localPP= 0 and 0.07% reduction in quartet score). Forcing Cuckoo to be sister to Bustard or Hoatzin to be sister to Cursorimorphae shows a case where neither the constrained nor the unconstrained tree have strong support, and thus, results are inconclusive. Most interestingly, owl fits quite well with Coraciimorphae (localPP 0.99 in constrained analyses) as well as its unconstrained position as sister to birds of prey (localPP 1.0); this observation creates a suspicion of gene tree discordance due to processes other than ILS such as hybridization.

3.5 Acknowledgement

Chapter 3, in full, is a reprint of the material as it appears in “Forcing external constraints on tree inference using ASTRAL” (2020). Rabiee, Maryam; Mirarab, Siavash, BMC genomics, 21(2), 1-13. The dissertation author was the primary investigator and first author of this paper.

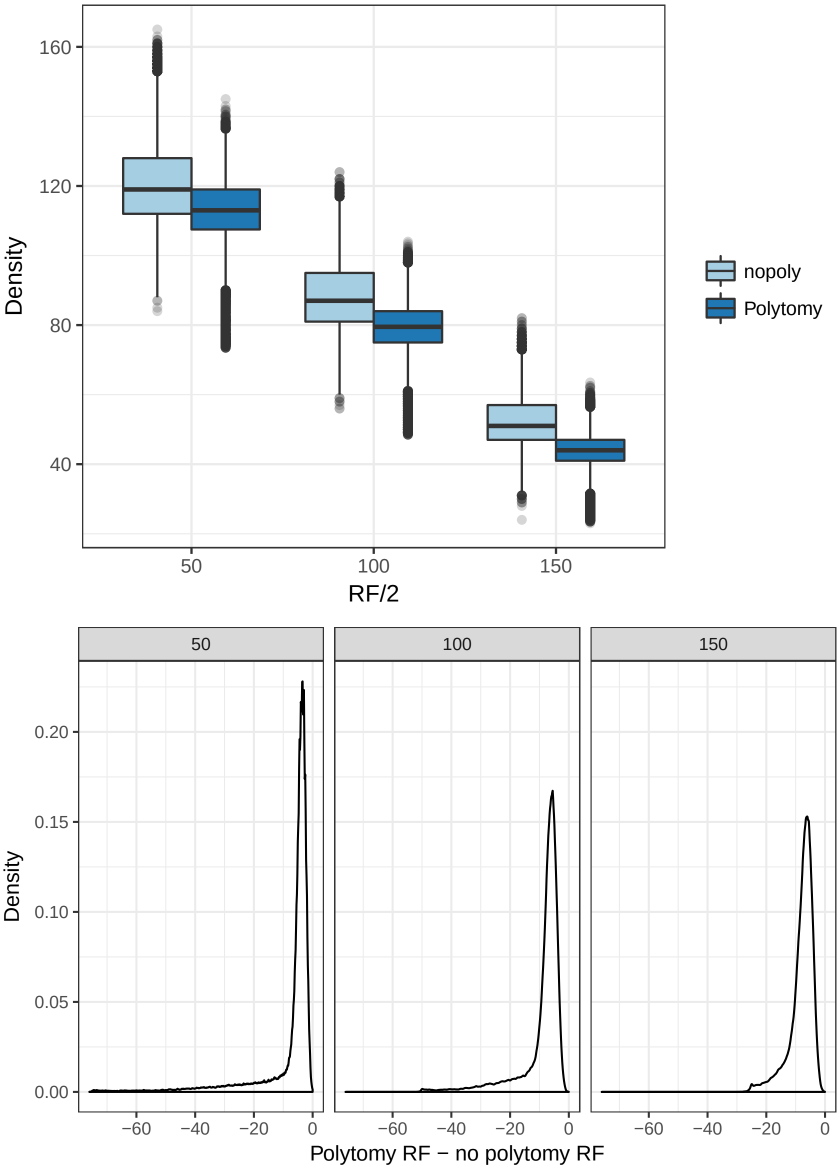


Figure 3.8: Impact of changes to the B-RF(+) algorithm. Our suggested changes to the B-RF(+) algorithm result in reduced RF distance between completed T_b and T_r. Top: The distribution of RF distance between results of tree completion and T_r with (Polytomy) and without (nopoly) our changes to allow multifurcating trees. Bottom: The distribution of the reduction in RF distance as a result of allowing polytomies in output trees. The drop in RF distance can be as high as 76 edges and is on average 8 branches.

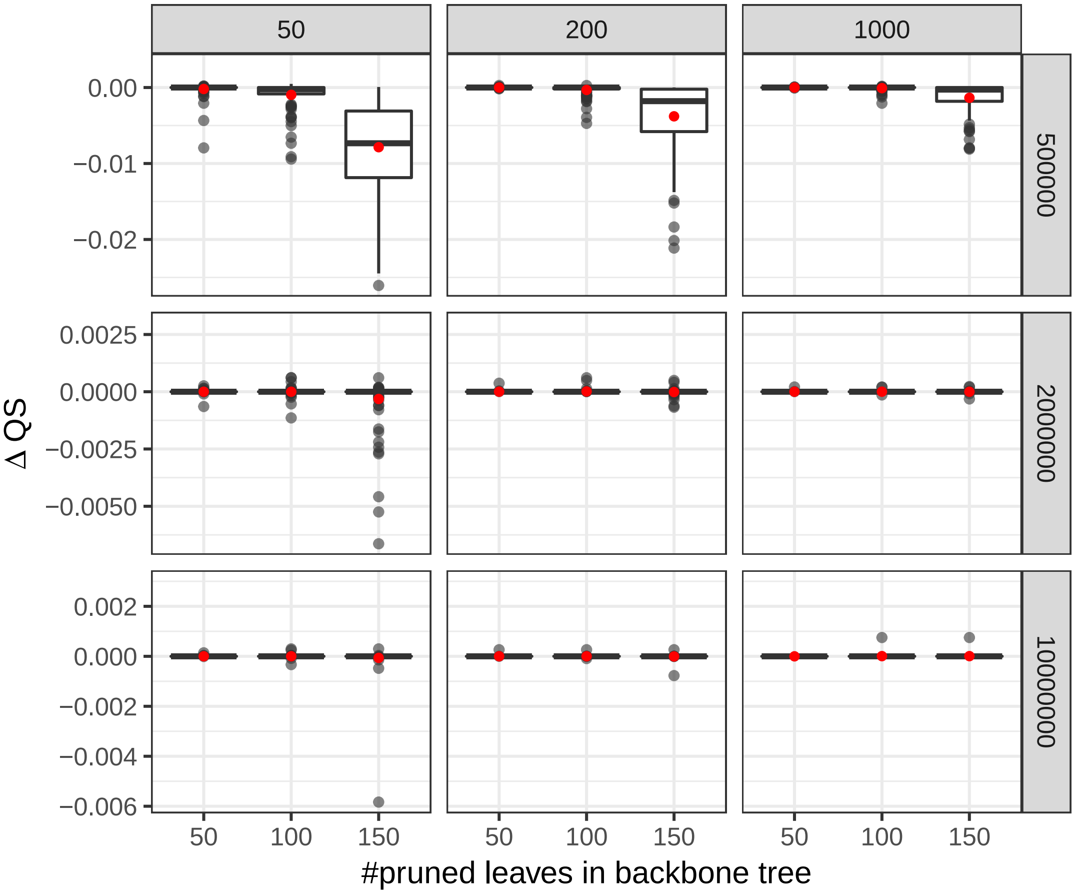


Figure 3.9: Change in quartet score due to constraints with missing leaves. The quartet score of the constrained search run on the backbone tree with a random set of leaves pruned is compared with unconstrained tree. Positive difference means improvement in the quartet score. There are three different levels of ILS (500K, 1M and 2M generations), varying number of genes (50, 200, 1000) and backbone trees with different sizes.

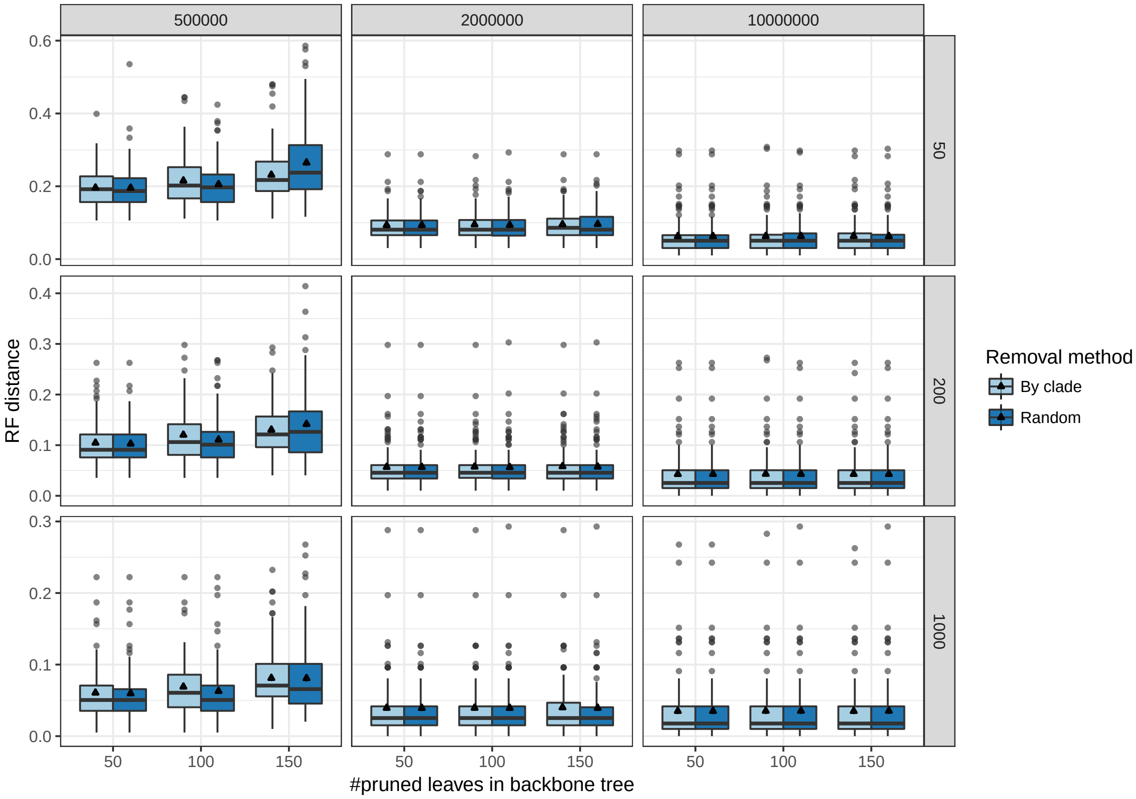


Figure 3.10: Impact of clade-based removal of species. The two methods for pruning leaves from backbone tree are compared here: random removal and clade-based removal. In the clade-based, for a desired number of pruned leaves, we repeatedly remove a randomly chosen remaining clade in the tree only if pruning the clade does not push us over the limit and continue until the desired size is achieved. Note that the selected nodes could be singletons (leaves) or internal nodes (clades) in the tree. We show Normalized RF distance between inferred constrained ASTRAL tree and the true species tree. There are three different levels of ILS (500K, 1M and 2M generations), varying number of genes (50, 200, 1000) and backbone trees with different sizes.

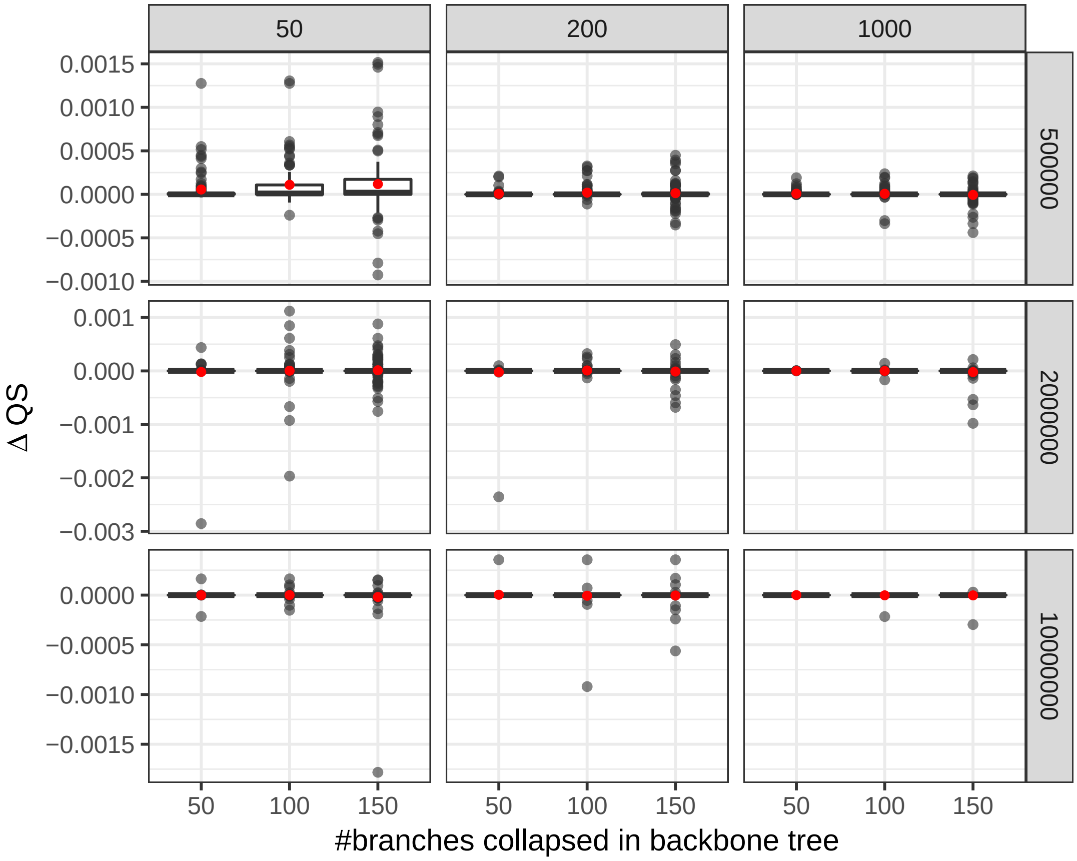


Figure 3.11: Change in quartet score due to constraints with missing branches. The quartet score of the constrained search run on the backbone tree with randomly collapsed branches is compared with unconstrained tree. Positive difference means improvement in the quartet score. There are three different levels of ILS (500K, 1M and 2M generations), varying number of genes (50, 200, 1000) and backbone trees with different number of branches collapsed. These branches are selected randomly among all $n-1$ internal branches.

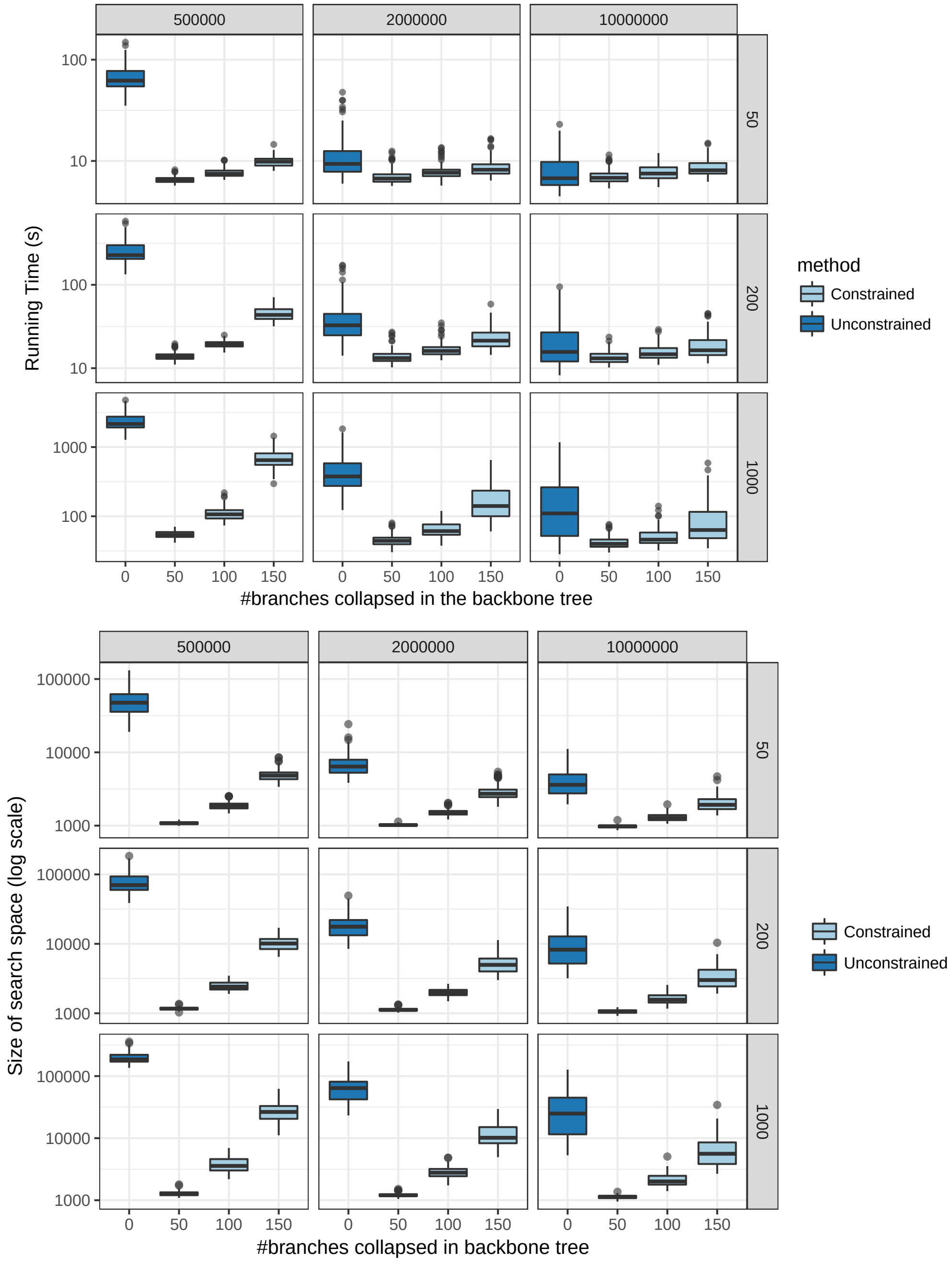


Figure 3.12: Impact of multifurcating constraint trees on running time and search space. The running time (top) and search space size $|X|$ (bottom) of constrained and unconstrained ASTRAL-III are compared here in log scale. There are three different levels of ILS (500K, 1M and 2M generations), varying number of genes (50, 200, 1000) and backbone trees with different number of branches collapsed. The search space size of constrained and unconstrained ASTRAL-III are compared here in log scale. There are three different levels of ILS (500K, 1M and 2M generations), varying number of genes (50, 200, 1000) and backbone trees with different sizes.

Chapter 4

Multi-allele species reconstruction using ASTRAL

Genome-wide phylogeny reconstruction is becoming increasingly common, and one driving factor behind these phylogenomic studies is the promise that the potential discordance between gene trees and the species tree can be modeled. Incomplete lineage sorting is one cause of discordance that bridges population genetic and phylogenetic processes. ASTRAL is a species tree reconstruction method that seeks to find the tree with minimum quartet distance to an input set of inferred gene trees. However, the published ASTRAL algorithm only works with one sample per species. To account for polymorphisms in present-day species, one can sample multiple individuals per species to create multi-allele datasets.

Here, we introduce how ASTRAL can handle multi-allele datasets. We show that the quartet-based optimization problem extends naturally, and we introduce heuristic methods for building the search space specifically for the case of multi-individual datasets. We study the accuracy and scalability of the multi-individual version of ASTRAL-III using extensive simulation studies and compare it to NJst, the only other scalable method that can handle these datasets. We do not find strong evidence that using multiple individuals dramatically improves accuracy.

When we study the trade-off between sampling more genes versus more individuals, we find that sampling more genes is more effective than sampling more individuals, even under conditions that we study where trees are shallow (median length: $\approx 1N_e$) and ILS is extremely high.

4.1 Introduction

Using a large number of loci to reconstruct the species phylogeny is becoming routine practice [40, 43, 64]. Beyond dramatically increasing the amount of data available [132], whole-genomes have enabled us to study individual genealogies, which can be discordant with each other and with the species tree [85]. Incomplete lineage sorting (ILS) is a major cause of such discordance [85, 38], and the multi-species coalescent (MSC) model [115, 126] has enabled a probabilistic study of ILS. In the presence of ILS, the traditional approach of concatenating data from multiple genes can become misleading [72, 44, 131], which has motivated researchers to develop alternative methods [86, 44]. One approach is to use a full Bayesian analysis under the MSC model [81, 57] to co-estimate gene trees and the species tree, but such analyses have severe computational limitations [70, 11, 143], and improving their scalability is a subject of active research [110, 173]. A more scalable approach first estimates gene trees individually, and then summarizes them to obtain the species tree, while accounting for the distribution on gene trees defined under the MSC model [39]; many such summary methods have been developed, including STAR/STEAC [84], NJst [82], STEM [71], GLASS [105], MP-EST [83], and more recently, ASTRAL [99]. Yet a third category of methods avoid gene trees altogether and estimate species trees from concatenated sequence data directly while accounting for gene tree discordance. Examples of this category of methods includes SNAPP [19], SVDQuartets [27], the PoMo model [36] and its implementation in IQ-Tree [141].

ILS can arise when multiple alleles of a gene survive through consecutive speciation events. For example, imagine an ancestral species (or population) R that gives birth to three present day species X, Y, and Z, with X and Y sharing a common ancestor P. Consider a locus with two alleles a and b in R. By random chance, only b survives in Z, whereas both alleles remain present throughout the life of P; however, only a survives in X and only b survives in Y. Such a scenario will create a linage tree (which we call a gene tree) that puts Z and Y as sisters to the exclusion of X, and this will be in conflict with the species tree that puts X and Y together. The probability of such a scenario will be much higher if P spans a relatively small number of generations or if it has a high effective population size [115]; the ratio of the number of generations to the haploid effective population size is called the coalescent unit (CU) and gives a measure of branch length that directly relates to the expected amount of ILS [37]. A similar scenario is that a and b both survive in all five populations, but the single individual chosen to represent X happens to be homozygous for a, whereas individuals representing Y and Z happen to be both homozygous for b; this will also lead to a lineage tree that is discordant with the species tree. The likelihood of this scenario, in addition to being affected by the internal branch lengths, also depends on terminal branches. This second scenario of discordance makes it clear that for short terminal branches, the choice of individuals representing a species may matter.

To account for potential impacts of polymorphism in present-day species, several authors have suggested sampling multiple individuals per species (and/or phasing) to create multi-allele datasets where each species can have multiple alleles per locus [86, 57, 70, 91, 59, 26]. Models of sequence evolution that directly account for polymorphisms have been also developed [36]. Alternatively, one can use summary methods to analyze multi-allele datasets. This requires that gene trees estimated from sequence data are either multi-labeled by species names (i.e., several nodes in a gene tree are labeled by the same species) or are labeled by the name of individuals and a mapping between individuals and species is known. Then, the summary method can estimate the best species tree labeled with names of species; this is equivalent to finding the best species tree labeled with individual names constrained to each species being monophyletic. Predefining species boundaries side-steps difficulties of defining boundaries of recently diverged species [23, 113], leaving that question to the analyst. In other words, the approach assumes the species are correctly delimited.

When species boundaries are known *a priori*, the MSC model easily extends to the multi-individual case [152, 2, 3]. While the evidence for the cost-effectiveness of sampling multiple individuals remains mixed [116], several methods exist that can use such data [57, 82, 83]. To our knowledge, NJst is the only summary method that can handle multi-individual *unrooted* gene trees [82], and after a fix in handling multi-individual data [1], the NJst method is now statistically consistent for multi-individual datasets.

One commonly used method of species tree reconstruction is ASTRAL [99, 100], which has been used on many phylogenomic datasets. Given a set of unrooted input gene trees, ASTRAL seeks to compute the species tree that shares the maximum total number of induced quartets with the input set of gene trees. A constrained version of this NP-hard problem [74] is solved by ASTRAL in polynomial time, but solutions to the constrained problem are proved statistically consistent under the MSC model [99].

The published ASTRAL algorithm, until now, could only take single-labeled trees as input and had no way of handling multi-allele datasets. Incipient implementations of a feature in ASTRAL to handle multi-allele datasets were not rigorously tested; nor were they formally described. In this paper, we will introduce a new algorithm for handling multi-labeled gene trees in ASTRAL (which is different from the previous untested method) and establish its accuracy on both simulated and empirical datasets. We show that the quartet optimization problem extends in a natural way to multi-labeled datasets, leaving us with only one difficulty: defining the constrained search space for ASTRAL. We propose and test heuristic approaches based on subsampling individuals to build a sufficiently large search space. We test the method in extensive simulations and ask whether using multiple individuals results in improved accuracy compared to having single individuals. We also test if predefining species boundaries improves accuracy and compare the accuracy of ASTRAL to NJst.

4.2 Theory

4.2.1 Background on ASTRAL

We are given a set $\mathcal{G}$ of k unrooted gene trees, singly-labeled by the leaf-set L of n taxa. There are $k\binom{4}{n}$ quartet trees induced by the input set $\mathcal{G}$. The Weighted Quartet (WQ) score of any candidate species tree is defined as the number of the $k\binom{4}{n}$ quartet trees that the candidate tree also induces. ASTRAL seeks to find the species tree that maximizes the WQ score [99, 100].

At the heart of the ASTRAL algorithm is the ability to score a tripartition of the species leaf-set in isolation from others tripartitions, which then enables the use of dynamic programming. The dynamic programming starts from the set L and recursively divides it into smaller subsets, each time choosing a division that maximizes the number of shared quartets. If we consider all ways of dividing a subset into smaller subsets, the problem is solved exactly but in exponential time. To obtain a polynomial time algorithm, we define a constraint set X of bipartitions and restrict the search to tripartitions derived from the set X. Let $X' = \{A : A|L-A \in X\}$. To constrain the search, we only consider divisions of a subset into two parts such that both parts appear in X'. We define $V(A)$ as the score for an optimal subtree on A, and set $V(A) = 0$ for $|A| = 1$. Then, the dynamic programming recursion is:

$$V(A) = \max_{A', A-A' \in X'} V(A') + V(A-A') + w(A'|A-A'|L-A) \tag{4.1}$$

where w computes the number of gene tree quartet topologies that match any species tree that includes the tripartition $A'|A-A'|L-A$ (see ASTRAL-II [100] – Eq. 2). ASTRAL-II, starts by including in X the set of bipartitions observed in the input gene trees and then supplements that set using a set of various heuristics [100]; ASTRAL-III slightly changes those heuristics such that the size of the set X is guaranteed to grow no more than linearly with both n and k [169].

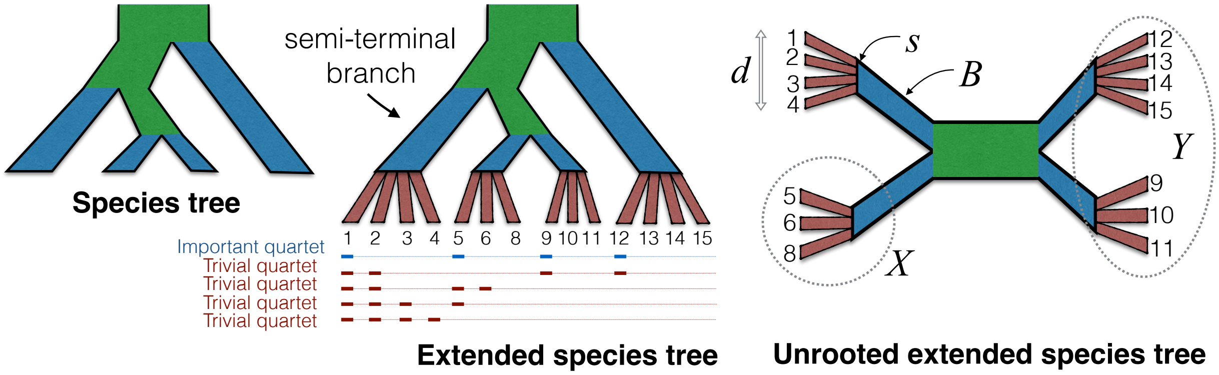


Figure 4.1: **The illustration of rooted and unrooted extended species tree.** Left: the species tree. Middle: the extended species tree with polytomies added for individuals of each species. Only quartets that have individuals from four species are important; the remaining quartets are trivial. For the top two trivial quartets, the extended species tree induces a resolved tree but all valid extended species trees define *the same* tree and therefore the quartet does not help in deciding among species trees. For the bottom two trivial quartets, all extended species trees give an unresolved quartet, and therefore, do not help in distinguishing species trees. Right: the unrooted extended species tree. The species s has d individuals; B is a semi-terminal branch in the extended species tree and corresponds to the terminal branch of s in the original species tree. Removing B and its two adjacent branches divides the tree into three groups, one corresponding to s and two opposite groups shown here as X and Y. Green: internal branches, Brown: branches corresponding to individuals, Blue: terminal/semi-terminal branches.

4.2.2 Quartet score for multiple individuals

In the presence of multiple individuals, the definition of the quartet score can be easily generalized. Let $S = \{1 \dots n\}$ be the set of species and let $R = \{1 \dots m\}$ be the set of individuals. The input is a mapping $m : R \to S$ from individuals to species and a set $\{t_1 \dots t_k\}$ of unrooted gene trees, each labeled by $R_i \subset R$. Following Allman *et al.* [3], for any species tree T labeled by S, we define an extended species tree T_{ext}, labeled by R. The extended tree is built by adding to each leaf of T all individuals corresponding to that species as a polytomy (Fig. 4.1); i.e., for leaf $s \in S$ of T, add a child $r \in R$ for every $r \in m^{-1}(s)$). We define the quartet score of an unrooted species tree T labeled by S with respect to the input gene trees as the quartet score of the extended species tree with respect to gene tree:

$$Q(T) = \sum_{i=1}^{k} \sum_{\{a,b,c,d\} \subset R_i} I(T_{ext} \upharpoonright \{a,b,c,d\}, t_i \upharpoonright \{a,b,c,d\}) \tag{4.2}$$

where $I(t_1, t_2)$ indicates if its arguments are topologically identical trees and $t \upharpoonright A$ denotes the tree t restricted to the set A. ASTRAL in multi-individual mode seeks to find the species tree that maximizes this quartet score.

To show the connection between the quartet score and ILS, we show that ASTRAL is a statistically consistent estimator of the species tree for multi-individual randomly-sampled error-free gene trees given a correct mapping between species and individuals. We do so using results by Allman *et al.* [3] who showed (Corollary 10 [3]) that a coalescent process on the extended tree with one sample per species leads to exactly the same distribution of gene trees as the the multiple-individual process on the original species tree.

The set of all $\binom{m}{4}$ quartets can be divided into two types: trivial and important quartets. A quartet $Q = \{a,b,c,d\}$ is trivial if any gene tree reduced to Q either contradicts every extended species tree or contradicts no extended species tree; in contrast, a gene tree resolution of an important quartet matches some extended species trees and contradicts others. Let $m^*(X) = \{m(r) | r \in X\}$ give the set of species for a set of individuals. It's easy to see that a quartet Q is important if $|m^*(Q)| = 4$; i.e., if it includes at most a single individual from any species. For example, in Figure 4.1, the quartet $\{1,5,9,12\}$ is important. All other quartets (i.e., those that include at least one species with multiple individuals) are trivial. To see this, first consider $|m^*(Q)| = 3$: two individuals (say, a and b) chosen from the same species and the other two chosen from two different species (e.g., $\{1,2,9,12\}$ in Fig 4.1). Any unrooted gene tree that puts a and b on one side of a quartet and c and d on the other side will match every possible extended species tree. Similarly, every gene tree that puts a and b on the opposite side of a quartet tree will contradict any possible extended species tree. Thus, these quartets are trivial. A similar argument carries if two individual are from one species and the other from a second species ($|m^*(Q)| = 2$; e.g., $\{1,2,5,6\}$ in Fig 4.1)). Quartets that include three (e.g., $\{1,2,3,5\}$ in Fig 4.1) or four (e.g., $\{1,2,3,4\}$ in Fig 4.1) individuals from the same species will always be unresolved in any extended species tree and thus cannot contradict any extended species tree. Thus, these are also trivial.

For all important quartets, since one individual is chosen per species, the unrooted species tree topology matches the most probable unrooted gene tree [3]. Thus, trivial quartets are inconsequential in the choice of the species tree and for important quartets, the central results required for proving ASTRAL statistically consistent carries through to multi-individual datasets. Therefore, the same argument used to prove ASTRAL statistically consistent for the single individual datasets [99] can be used here to argue that ASTRAL is statistically consistent.

4.2.3 Dynamic programming algorithm for multi-individual datasets

To find the species tree that maximizes the quartet score, we can continue to use the dynamic programming given in Equation (4.1) with an additional constraint: individuals of a species should not be separated into two parts. Thus, the new dynamic programming recursion is still given by equation (4.1) with the two changes: i) the set X' never includes a cluster (half a bipartition) that has some but not all individuals of a species; (ii) the boundary condition is changed to $|m^*(A)| = 1$. In other words, we stop as soon as A includes only individuals from a single species. With these constraints, for any $w(X|Y|Z)$ calculation, all individuals of any species will belong to only one of X, Y, or Z. The dynamic programming produces the extended species tree, which can be simply mapped to a S-labeled species tree by removing all terminal branches.

Because the dynamic programming stops short of resolving the relationship between individuals of the same species, some quartets will not be counted by the dynamic programming. Note that to find the species tree that maximizes the score, it does not matter if we count a given trivial quartet, as long as we either always count it or always ignore it. Important quartets, however, should always be counted. A gene tree quartet topology ab/cd is counted by the $w(X|Y|Z)$ function if there exists a permutation of (X,Y,Z) denoted as (U_1,U_2,U_3) so that $a \in U_1$, $b \in U_2$ and $c,d \in U_3$ or $c \in U_1$, $d \in U_2$ and $a,b \in U_3$. It's easy to see that an important quartet will be counted by the dynamic programming exactly twice, just like single individual ASTRAL, because the constraints have no bearing on quartets with one individual from four different species.

The case of trivial quartets is more complicated. A quartet Q with individuals from two or one species ($|m^*(Q)| \leq 2$) will always intersect with at most two of sides of $X|Y|Z$, and thus, can never be counted by any $W(X|Y|Z)$ calculation. This leaves us with one form of trivial quartets, namely, those that includes individuals from three species. Let $Q = \{a_1,a_2,b,c\}$ where $m(a_1) = m(a_2) \neq m(b_1) \neq m(c_1)$. For any species considered by the search space of the dynamic programming, there exist a tripartition $A'|A-A'|L-A$ scored in Equation 4.1 where a_1 and a_2 belong to one side, b belongs to a different side, and c belongs to the third side and thus $w(A'|A-A'|L-A)$ will count the number of genes where Q is resolved as a_1a_2/bc. Note that this is true for any tree that the dynamic programming could produce. Moreover, these quartets will be counted only once since there could never be a tripartition where b and c belong to one side, a_1 to another side, and a_2 to a third side. Thus these trivial quartets will contribute to the score of every species tree equally and will thus be inconsequential in the choice of the species tree.

To summarize, we showed that the dynamic programming, with two simple modifications will optimize the quartet score correctly. However, it will not compute the correct quartet score because it fails to count some trivial quartets. To address this, at the end, the score of the species tree is recomputed with a simple procedure that explicitly counts trivial quartets.

4.2.4 Defining the search space

The final and the main difficulty in using multi-labeled gene trees is defining the search space, which entails constructing the constrained set X. Recall that any $A \in X$ should include either all or none of the individuals of any species. Simply adding bipartitions from the gene trees, as ASTRAL-I, ASTRAL-II, and ASTRAL-III all do, will violate this constraint. In order to address this, for each gene tree t_i, we create 100 singly-labeled gene trees $t'_{i,0} \dots t'_{i,99}$ by randomly choosing one individual from each species, taking the induced tree, and renaming leaves from R to S using the mapping m. Then, we compute the greedy consensus of $t'_{i,1} \dots t'_{i,100}$, which will be labeled by S, and we extend it (just like building an extended species tree) to get the tree t''_i on leaf-set R. We proceed to build the set X using the usual ASTRAL-III approach with $t''_1 \dots t''_k$ as input. Thus, the set X will includes all bipartitions of $t''_1 \dots t''_k$ as well as selected resolutions of their polytomies; moreover, ASTRAL-III heuristics expand the set X using several methods [100, 169], many of which rely on a similarity matrix. To build the similarity matrix for pairs of species, we first build the ASTRAL similarity matrix [100] for pairs of individuals and then compute averages for all pairs of individuals corresponding to each pair of species as shown in Algorithm 2. To have a sufficiently large search space, we repeat the entire process for r rounds, setting r by default to $1 + \lceil \log_2 \max_{i \in S} |\{j \in R : m(j) = i\}| \rceil$. This default setting, which the user can adjust if desired, starts with 1 rounds if all species have a single species and slowly (i.e., logarithmically) increases the number of rounds as more individuals become available. For example, if the maximum number of individuals per species is 5, 10, 20, or 40, the number of rounds is set to 4, 5, 6, or 7, respectively. This slow pace of growth is based on our tests that shows the set X is sufficiently large with these settings.

4.2.5 Branch length and branch local posterior probability

One benefit of having multi-individual datasets is that the branch length for terminal branches can also be computed because terminal branches are internal branches in the extended species tree. We extend the approach currently used in ASTRAL [136], which estimate the ML branch length as $-\ln \frac{3}{2}(1 - \frac{\Sigma_1^k q_i}{k})$; here, q_i is the fraction of quartets defined "around" the branch of interest that agree with the species tree topology in gene tree t_i. We have defined a quartet to be around an internal branch b if each of its four leaves come from the subtree attached to one of the four branches adjacent to b. For multi-individual datasets, let a species s have $d = |m^{-1}(s)|$ individuals. For the terminal branch B corresponding to the species s in the unrooted extended species tree, let X and Y be the two sides of the opposite sides of the branch B (see Fig. 4.1). There are $\binom{d}{2}\binom{|X|}{1}\binom{|Y|}{1}$ quartets such that the length of the quartet tree correspond exactly to the length of B (e.g., $\{1,2,5,9\}$ in Fig 4.1). We define q_i as the proportion of these quartets that agree with the species tree in gene tree i. A naive enumeration of all possible quartets would result in slowed computation of branch lengths; however, an algorithmic trick can reduce the running time. For each individual i, we create a quadripartition of leaves: $\{i\}|m^{-1}(s)-\{i\}|X|Y$. For each quadripartition, we compute the branch length using the same fast technique introduced in a previous paper [136] for internal branches. This will require only d calculations per terminal branch; at the end, we have counted each quartet exactly d times, so we divide the sum by d. Also note that given q_i, not only we can compute the branch length, but we can also compute the local posterior probability (localPP) of the terminal branches [136]. This measure can be interpreted as the probability of the predefined boundaries between species being in fact correct.

4.3 Materials and Methods

4.3.1 Dataset

We simulate two new datasets (see Table 4.1): a heterogeneous dataset (D1) where many parameters are simultaneously changed and ILS levels are extremely high and a more homogeneous dataset (D2) where parameters are less varied and the amount of ILS is controlled to create three model conditions. For both datasets, we use SimPhy [89] to generate species trees according to the birth/death model and gene trees according to the MSC model (exact parameters described in C.1). Each replicate of the simulation has its own species tree and all replicates have 5 individuals per species.

D1. This dataset includes 330 replicates and for each replicate, the number of genes were uniformly sampled between 50 and 1000. The number of species were also uniformly sampled between 20 and 200. The birth rate parameter in simulating the species tree is randomly sampled from a log uniform distribution in $[10^{-7}, 10^{-6}]$, and the death rate is also sampled from a log uniform distribution, bounded from below by 10^{-7} and bounded by the birth rate parameter from above. The population size is sampled from a uniform distribution in $[10^5, 10^6]$. Since sampling multiple individuals is often done for shallow species trees, we sampled a maximum species tree height for each replicate from a log normal distribution with an expected value of 0.5M generations; the number of generations ranges between 0.19M and 1M in 90% of replicates (Fig 4.8). Moreover, internal branches are extremely short and 90% of them span between 4170 and $382K$ generations.

D2. This dataset has three model conditions, each with a fixed maximum number of generations: 0.5M, 1M, or 2M. We simulate 50 replicates of each model condition, all with 200 species and 1000 genes with species birth rate set to 10^{-6} and death rate set to zero (so, a birth-only model). The population size is fixed to 200,000. Each replicate of this dataset also has a single-individual outgroup. To test the effects of sampling, we also create two new versions of D2 where only one or two individuals per species are randomly sub-sampled.

The quartet scores of true species trees given true gene trees, used to measure gene tree discordance due to ILS, indicate that the D1 dataset has extremely high levels of ILS (Fig. 4.2a). The mean quartet score is 0.496 with standard deviation of 0.127. Note that quartet scores close to $1/3$ correspond to gene trees that are random with respect to the species tree. On the D2 dataset, quartet scores range from low (0.50 on average) to relatively high (0.78 on average), depending on the tree height (Fig. 4.2a).

After simulating true gene trees, we use Indelible [51] to evolve nucleotide sequences down the gene trees using the GTR+Γ model of sequence evolution with randomly sampled sequence lengths and mutation parameters. For each gene, we sample the sequence lengths from

Table 4.1: Summary of properties of datasets D1 and D2. For parameters drawn from distributions other than uniform and for quartet scores and gene tree error, we show summary statistics: (5% percentile, mean, 95% percentile). For sequence length, distribution parameters are drawn from another distribution as detailed in C.1. M indicates 10^6.

Dataset	**D1 (heterogeneous)**	**D2 (homogeneous)**
Number of replicates	330	150
Number of species	uniform(20,200)	200
Number of genes	uniform(50,1000)	1000
Population size	uniform(10^5, 10^6)	2×10^5
Birth-rate	log-uniform(10^{-7},10^{-6}) (1.1×10^{-7}, 3.8×10^{-7}, 8.3×10^{-7})	10^{-6}
Death-rate	log-uniform(10^{-7},birth-rate) (1.0×10^{-7}, 2.0×10^{-7}, 4.4×10^{-7})	0
Number of generations	log-normal(13,0.5) (1.9×10^5, 5.0×10^5 , 1.0×10^5)	$0.5M$, $1M$, or $2M$ (50 replicates each)
Sequence length	log-normal(–) (406bp, 1096bp, 1803bp)	log-normal(–) (322bp, 721bp, 1217bp)
Quartet score (ILS)	(0.37, 0.50, 0.75)	0.5M: (0.48, 0.50, 0.53) 1M: (0.56, 0.62, 0.66) 2M: (0.72, 0.78, 0.83)
Average gene tree error (AGTE); RF distance	(0.19, 0.4, 0.68)	0.5M: (0.13, 0.42, 0.84) 1M: (0.10, 0.31, 0.77) 2M: (0.06, 0.25, 0.70)
Sequence evolution model	GTR+Γ with parameters from Dirichlet (C.1)	
Estimated gene trees	FastTree ($GTR+\Gamma$)	

a log normal distribution with parameters specific to each replicate; the parameters are also drawn randomly from a distribution (described in C.1). The empirical average sequence length is 1096 for D1 and 721 for D2. The median gene sequence length is between 406bp and 1803bp in 90% of the 330 replicates of D1 and is between 322bp and 1317bp in 90% of the 150 replicates of D2. The $GTR+\Gamma$ parameters were drawn from Dirichlet distributions used in the ASTRAL-II paper (parameters are estimated from a collection of biological datasets [100]).

Given gene alignments, we then estimate gene trees using FastTree [120] with the GTR+Γ model. The average gene tree estimation error, as measured by the RF distance between true and estimated gene trees, is extremely varied (Fig. 4.2b). The gene tree error on the D1 dataset is on average 0.4 with standard deviation 0.13, and in 90% of replicates, it ranges between 0.19 and

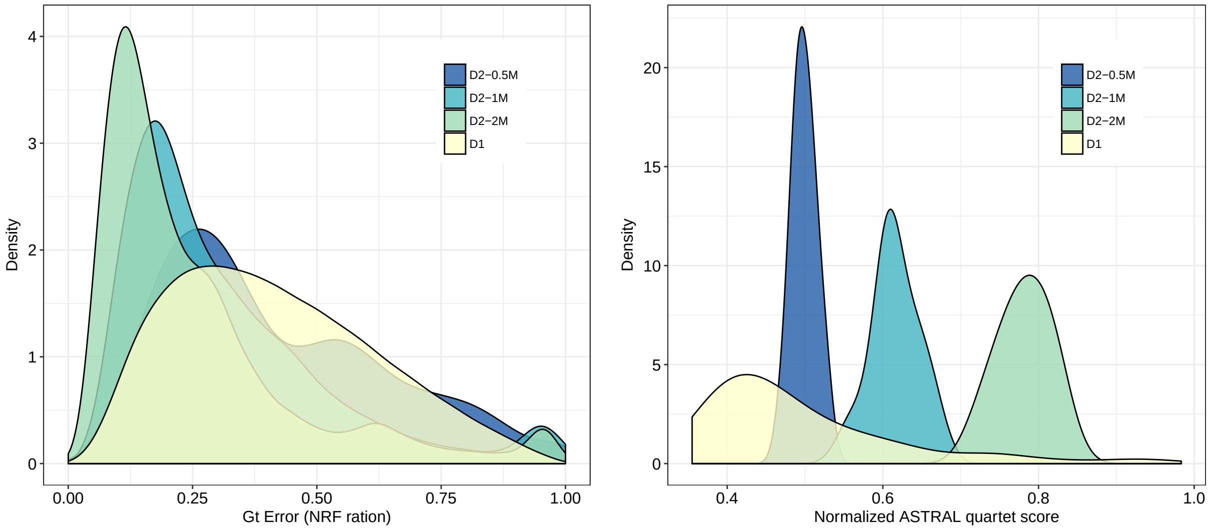


Figure 4.2: **Level of ILS and gene tree error in simulated data.** (a) The distributions of the quartet score of the true species tree versus the true gene trees shown as kernel density plots. Colors distinguish datasets D1 (326 replicates) and D2 (50 replicates per model condition defined by tree height set to 0.5M, 1M, or 2M generations). (b) Density plot of normalized Robinson Foulds (RF) distances between estimated and true gene trees, which quantifies gene tree error.

0.68. On the D2 dataset, the mean gene tree error was 0.42, 0.31, and 0.25 on average for 0.5M, 1M, and 2M model conditions. As expected, shallower trees not only have higher ILS, but also have increased gene tree error (Fig. 4.2). Both true (simulated) and the inferred gene trees are used as input to summary methods.

4.3.2 Compared Methods

We test several versions of ASTRAL and NJst. When we run ASTRAL-III (version 5.5.4 and version 5.5.9 for branch length and localPP) in the multi-individual mode (i.e., with a mapping file) we refer to the method as ASTRAL-multi. ASTRAL can handle polytomies and previous results indicate that removing very low support branches helps accuracy [169]. Thus, we also test a version of ASTRAL-multi where all the gene trees with branches with support below 5% are contracted and refer to this version as ASTRAL-multi-5%. For D1, we measure support using SH-like supports reported by FastTree. For D2, we compute gene tree bootstrap support with 100 bootstrap replicates. The default ASTRAL-III with each individual treated as a separate species (i.e., with no mapping file) is also tested (ASTRAL-ind). This allows us to investigate whether prespecifying species boundaries helps improving the accuracy of ASTRAL. We gave all ASTRAL runs a maximum of 48 hours of running time. In four out of 330 replicates of D1, either ASTRAL-ind or ASTRAL-multi failed to finish in the allotted time; we exclude these replicates.

NJst (i.e.;, USTAR [1]) is the main existing method capable of handling multiple individuals and *unrooted* gene trees. NJst estimates a distance matrix and uses neighbor joining to construct the species tree. It defines the distance between two species as the average gene tree internode distance or the average number of nodes between the gene copies sampled from the two species across all gene trees. The NJst method is also statistically consistent under the coalescent model after initial errors were fixed [1]. NJst cannot handle polytomies. To address this, we arbitrarily resolve polytomies in gene trees. Note that ML gene trees inferred by FastTree can include polytomies, for example, when multiple sequences are identical in a gene. We also tested removing genes with many polytomies, but this filtering did not help NJst (Fig 4.9).

4.3.3 Experiments

We study three research questions.

RQ1: Does the accuracy of species tree reconstruction improve when multiple individuals are included in a dataset? Does it still improve if the total sequencing effort is kept constant?

Using the three models of the D2 dataset with one, two and five individuals, we study effects of the number of individuals per species, either with variable total sequencing effort (i.e., keeping the number of genes fixed) or with fixed sequencing effort. To fix the sequencing effort, we reduce the number of genes such that the product of the number of genes and the total number of individuals is 1000. On the D1 dataset, some replicates have as few as 50 genes; thus, we could not perform these analyses as reducing the number of genes by a factor of 5 would leave us with only 10 genes.

RQ2: Is species tree accuracy improved by predefining species boundaries?

On the D1 dataset, We compare ASTRAL-ind and ASTRAL-multi. We compute the portion of species that are not monophyletic in ASTRAL-ind and also study the impact of predefined species on the rest of the tree.

RQ3: How does the accuracy of ASTRAL compare to alternative methods?

We compare the accuracy and running time of ASTRAL-multi and ASTRAL-multi-5% against NJst on both D1 and D2 datasets.

4.3.4 Evaluation Metric

Accuracy. To calculate the accuracy of a species tree constructed from input gene trees, we measure the False Negative (FN) rate, defined as the proportion of bipartitions in the true tree that are missing from the estimated tree. Note that since our trees are fully resolved, the FN rate is equal to the Normalized Robinson-Foulds (NRF) distance.

To compare ASTRAL-multi and ASTRAL-ind, we use the extended species tree. Recall that in an extended species tree, each species is replaced by a polytomy containing all individuals of the species. We use the extended tree as the reference tree, compute the FN rate of ASTRAL-ind, and break this FN rate into two components: the FN rate for branches that are terminal in the species tree but are internal in the extended species tree (we call these semi-terminal branches; see Fig 4.1) and the FN rate for the branches that are internal in both the species tree and the extended species tree (we call these internal branches). The FN rate for the semi-terminal branches gives the percentage of species that have not been recovered as monophyletic in ASTRAL-ind; the FN for semi-terminal branches is zero by construction for ASTRAL-multi.

Running time. Running time is measured on the Comet supercomuting cluster with 1 core out of 24 on Intel Xeon E5 CPUs and 5 GB of memory per job.

4.4 Results

4.4.1 RQ1: benefits of sampling multiple individuals

As expected, increasing the number of individuals from one to five gradually reduces the error (Fig. 4.3a). However, the improvements in accuracy tend to be small and are not statistically significant ($p = 0.24$ according to an ANOVA test with the number of individuals and the number of generations as independent variables); over all 150 replicates of all three conditions of D2, the error is reduced on average from 5.7% to 5.2% when going from a single individual to five. Contrary to our expectations,

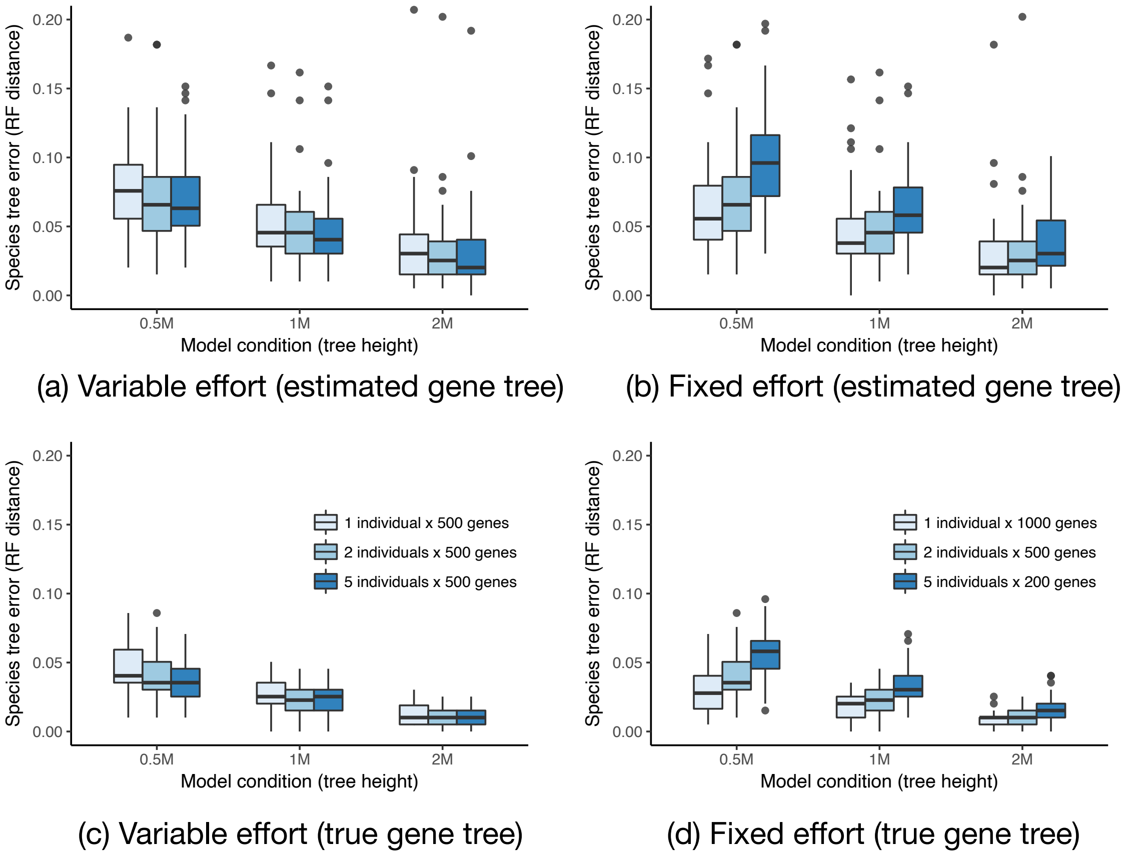
the impact of the number of generations (tree depth) on the effectiveness of increasing the number of individuals was also not significant ($p = 0.85$).

When we fix the total sequencing effort by reducing the number of genes as we increase the number of species, it becomes clear that sequencing more individuals is not nearly as effective as sequencing more genes (Fig. 4.3b). Thus, the error with 1000 genes and a single individual per species (4.7% on average overall) is less than the error with 200 genes and five individuals (7.0% on average overall). The same pattern is observed for all three model conditions.

Unlike estimated gene trees, when true gene trees are used, improvements in the accuracy are substantial (Fig. 4.3c), especially for the shallow model condition (0.5M generations). With true gene trees, improvements in accuracy with variable effort are indeed statistically significant ($p = 0.017$). Nevertheless, even with true gene trees, fixing the effort shows that having more genes is more effective than increasing the number of individuals (Fig. 4.3d).

4.4.2 RQ2: benefits of defining species boundaries

When we run ASTRAL-ind (i.e, without defining species) on the D1 dataset, we observe that the FN error of the resulting species tree is somewhat higher than when the mapping is predefined (Table 4.2). Focusing on the FN error rate of semi-terminal branches shows that



(a) Variable effort (estimated gene tree)

(b) Fixed effort (estimated gene tree)

(c) Variable effort (true gene tree)

(d) Fixed effort (true gene tree)

Figure 4.3: **Impacts of increased individual sampling.** The Robinson Foulds (RF) distance of the ASTRAL-multi species tree to the true tree is shown for three model conditions (x-axis) of the D2 dataset with (a,b) estimated gene trees and (c,d) true gene trees as input and with (a,c) variable sequencing effort or (b,d) fixed effort. Boxplots show the median (bars), the interquartile range (boxes), and outliers (points) defined as points above/below whiskers, which extend up to 1.5 times the height of the interquartile range on each side.

close to 9% of the species are not recovered as monophyletic when the mapping is not known in advance. Moreover, even focusing on internal branches, ASTRAL-ind has slightly higher error (8.3%) than ASTRAL-multi (7.8%), which indicates that providing species boundaries can also improve the accuracy of detecting the relationships among species. Nevertheless, we caution that the improvements, while statistically significant ($p < 10^{-5}$ according to a paired t-test) are not large.

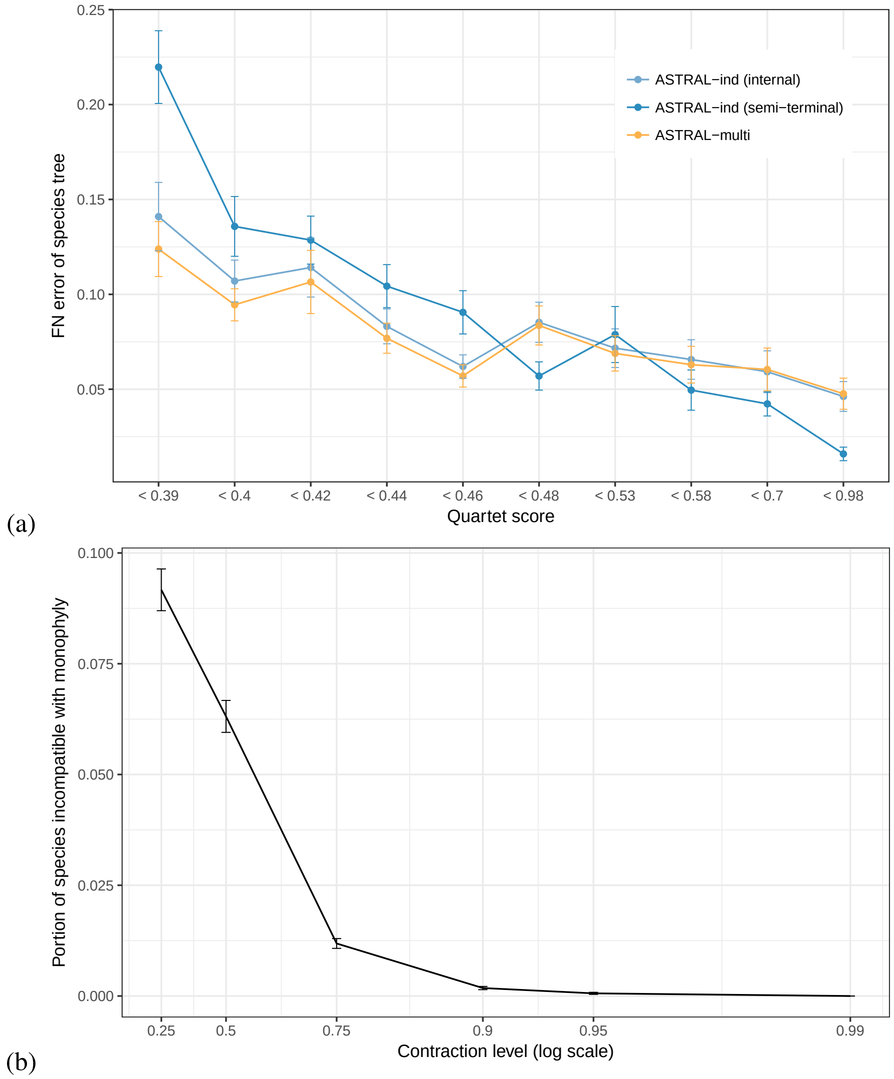


Figure 4.4: **Impact of predefining species.** (a) We show the proportion of branches of the true species tree missing in the trees inferred by ASTRAL-multi and ASTRAL-ind (without known species boundaries) on the D1 dataset (FN rate). For ASTRAL-ind, the reference tree is the extended species tree and the error is calculated separately for semi-terminal and internal branches. The 326 replicates are divided into deciles based on their level of ILS, measured by the quartet score of the true species tree (x-axis). The mean and standard error of FN are shown. (b) The percentage of species that are not compatible with monophyly in species trees constructed by ASTRAL-ind on D1 after branches below a level of localPP support (x-axis) are contracted.

Table 4.2: The species tree error of ASTRAL-multi and ASTRAL-ind on the D1 dataset. For ASTRAL-multi, we show the average (326 replicates) Robinson Foulds (RF) distance to the true species tree, which is identical to False Negative (FN) rate. For ASTRAL-ind, we compare the inferred tree to the extended species tree, and show the FN rate, divided into two categories: semi-terminal branches (i.e., those corresponding to species) and internal branches (i.e., all other branches).

	ASTRAL-multi	**ASTRAL-ind**	
	Internal	**Semi-terminal**	**Internal**
Mean	7.8%	9.2%	8.3%
Median	6.6%	6.8%	7.1%
STD	6.3%	8.6%	6.9%

The error of ASTRAL-ind in recovering species as monophyletic is greatly impacted by the amount of ILS (Fig. 4.4a). For the highest levels of ILS, the error for semi-terminal branches can be as high as 20%, but this gradually reduces to 2% as the quartet score increases and the amount of ILS decreases. Similarly, the error in the internal branches of ASTRAL-multi is lower than ASTRAL-ind mostly for high levels of ILS and less so as the amount of ILS decreases (Fig. 4.4a). Interestingly, the wrong semi-terminal branches have low localPP; removing branches with 0.95 localPP or lower make ASTRAL-ind trees compatible with the monophyly of almost all the species (Fig. 4.4b). Another benefit of prespecifying species boundaries is that ASTRAL-multi runs around 23% faster that ASTRAL-ind on the D1 dataset.

4.4.3 RQ3: method comparison

We now compare the error rate of ASTRAL-multi and ASTRAL-multi-5% to NJst. On the D1 dataset, methods performed similarly in terms of accuracy (Fig. 4.10). In terms of running time, ASTRAL-multi is about three times faster than NJst (Table 4.3). In the rest of this section, we will focus on on the D2 dataset.

Accuracy. On the D2 dataset with five individuals per species, ASTRAL-multi-5% is in most cases, but not always, better than ASTRAL-multi (Fig. 4.5). The comparison between ASTRAL-multi-5% and NJst depends on the dataset. With very shallow trees (0.5M condition),

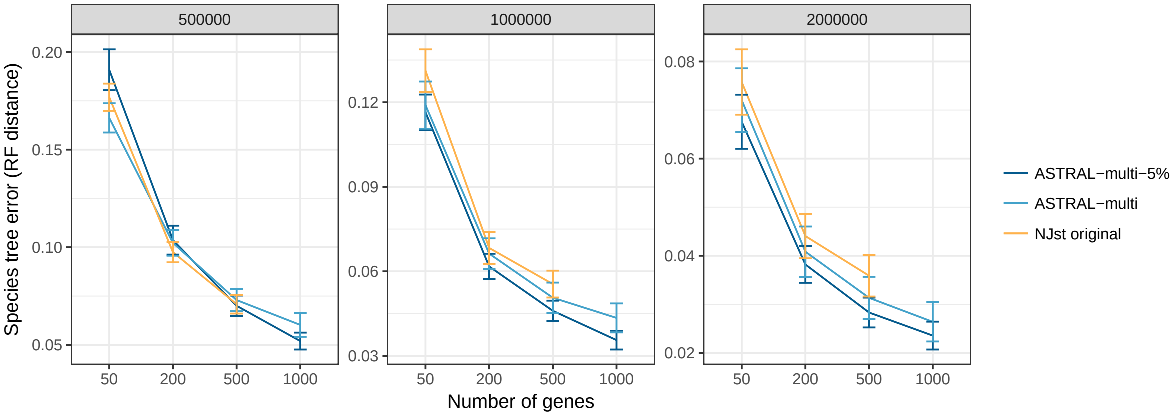


Figure 4.5: The error rate of ASTRAL-multi, ASTRAL-multi-5% and NJst on three models of dataset D2 with four levels of number of genes which are 50, 200, 500 and 1000. Mean and standard error of species tree error is shown over 50 replicates per condition.

ASTRAL-multi-5% and NJst are essentially tied, with a small advantage to NJst but the differences are not statistically significant ($p = 0.30$ according to a two-way ANOVA test with the choice of the method and the number of genes as independent variables). However, with the longer trees (1M and 2M conditions), ASTRAL-multi-5% is significantly better than NJst ($p = 0.012$), and the extent of the improvement is not significantly impacted by the number of genes ($p = 0.81$).

Overall patterns are as expected. Increasing the number of genes reduces the error while reducing the number of generations increases error. Not only shallow trees are harder to resolve, but also, smaller differences are observed between alternative methods for shallow trees. This could perhaps indicate that high gene tree error (Fig 2b) in shallow trees reduces the input quality and erases any potential differences between various summary methods.

Running Times. We compare the running time of ASTRAL-multi to NJst on D1 which has replicates with different input size (number of genes multiplied by number of leaves) from about 2000 to 16000. ASTRAL is substantially faster than NJst even running with contracted data (Fig 4.11). Given two days of running time, ASTRAL-multi and ASTRAL-multi-5% were able to finish on 327 and 324 replicates, respectively whereas NJst was able to finish on 313 replicates. A faster implementation of NJst called ASTRID [156] cannot currently handle multi-individual inputs.

4.5 Discussion

We introduced a multi-allele version of ASTRAL and on two large-scale simulated datasets we demonstrated its accuracy and running time efficiency. We saw that predefining the species boundaries can improve the accuracy. Sampling multiple individuals did not seem to help the accuracy.

Effectiveness of individual sampling. Unsurprisingly, increasing the number of individuals somewhat helped accuracy, but in the presence of gene tree estimation error, the improvements were marginal. Moreover, compared to increasing the number of genes, increasing the number of individuals did not seem an efficient use of resources. It can be argued that our way of fixing "effort" by controlling the product of the number of genes and the number of individuals is naive, as the cost of increasing loci versus individuals may differ. The exact relative costs will depend on the sequencing technology, sample collection, and many other factors beyond the scope of our work. Nevertheless, our results point to limited effectiveness of sampling more individuals.

It is perhaps surprising that even with variable sequencing effort, the reduction in the error is small as we increase the numbers of individuals. Our results are somewhat contrary to some previous simulation studies [86, 59, 91] that indicate that sampling more individuals is beneficial for shallow trees but agrees with others [116]. We stress that many of our simulations, especially D1, were on extremely shallow trees. For example, on the D1 dataset, our trees on average included 110 species generated in about half a million generations. Thus, the species tree branch lengths were extremely short; 70% of branches were a hundred thousand generations or shorter and 47% had a length ≤ 0.1 in coalescent units. Thus, our simulations were specifically designed to test conditions where multiple individuals may help according to previous reports, making it even more interesting that no strong pattern of improvement was observed. Our simulations differed from previous works in the level of gene tree error and also in the number of genes. Our gene trees have high levels of error (Fig. 2). It may be that gene tree estimation error for shallow trees reduces the value of having multiple individuals because the extra noise introduced by tree estimation weakens the signal. Consistent with this explanation, with true gene trees, using more individuals did result in improvements. Another difference between our study and previous studies is that we use hundreds of gene trees whereas previous studies include at most 50 gene trees [86, 59, 91, 116]. Another explanation is that with enough loci, the impact of having more individuals diminishes. We believe that with the current sequencing technology, testing methods in the presence of hundreds of loci is more relevant than tens of loci.

Errors in species delimitation. In all our analyses, the predefined species boundaries were perfectly correct. In practice, the boundaries defined *a priori* may or may not be correct. Introducing error in species identification may erase some of the benefits of using the multi-allele version of ASTRAL. Two solutions can be employed in practice. Helpfully, our simulations showed that contracting low support branches made all species compatible with monophyly (Fig. 4.4b). Thus, to find mistakes in the species mapping, the dataset can be analyzed in both modes: with and without species boundaries. When strong localPP is found for the lack of monophyly of some species, the analyst can reconsider the delimitation. Moreover, the multi-allele version of ASTRAL can produce branch length and localPP for the terminal branches of the species tree. In our simulations, terminal branches of the species tree generally had high support (Fig. 4.6c). For example, 75% of them had a localPP of 1.0 and close to 90% had a localPP of 0.75 or higher. When terminal branches have very low support, the species definition should be questioned.

Branch length accuracy. ASTRAL-multi can compute both terminal and internal branches. In our simulations, when true gene trees were used, terminal and internal branches lengths were both relatively accurate (Figs. 4.6ab and 4.12). However, the accuracy reduced substantially with estimated gene trees. Errors of close to even an order of magnitude were observed

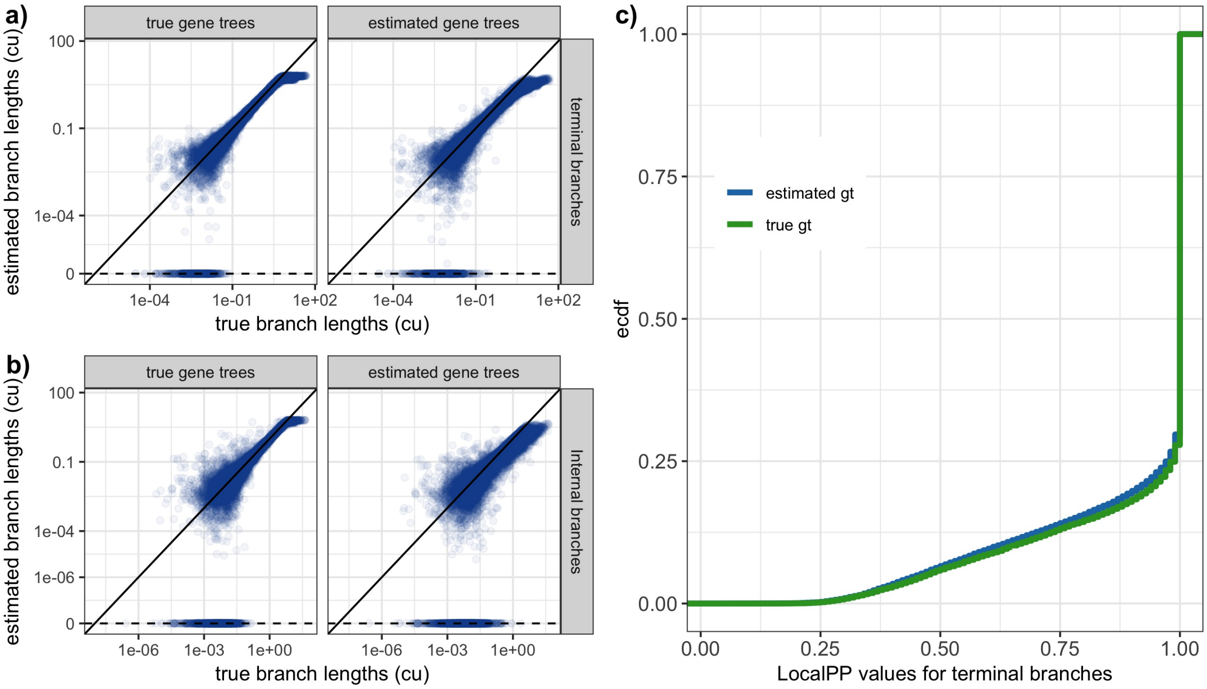


Figure 4.6: **The accuracy of branch length and localPP support**. (a,b) On the D1 dataset, we estimate the length of (a) terminal and (b) internal branches of the true species trees using both true and estimated gene trees in coalescent unit (cu). We show the true branch length (x-axis) versus the estimated branch length (y-axis), both in the log scale. (c) The empirical cumulative distribution function of the localPP support for terminal branches of the true species trees for D1 using estimated and true gene trees. Close to 75% of the terminal branches have the maximum localPP support.

in estimated branches. This pattern, which is consistent with older results [136], indicates that in the presence of gene tree error, branch lengths should be considered with caution. Interesting, we note that terminal branches seemed to have less error than internal branches (Fig. 4.6ab). Finally, using multiple individuals instead of one individual resulted in a small but consistent improvement in internal branch lengths, but once again, increasing the number of genes was more effective (Fig. 4.12).

Predictors of accuracy. The heterogeneity of the D1 dataset enables us to look for parameters that impact the accuracy of the ASTRAL species tree (Fig 4.7). We observed a strong dependence between the species tree error and the number of genes, the tree depth, the population size, the amount of true discordance measured by the quartet score of the true species tree versus true gene trees, and the average gene tree error. The only factor we studied that did not have a clear impact on the accuracy was the number of species. All factors other than gene tree error had a similar impact on the species tee accuracy if the true gene trees were used (Fig. 4.13). As expected, main predictors of the accuracy seemed to be the number of genes, the amount of true discordance, and the gene tree estimation error (Fig 4.7). These three factors combined in a simple linear model could explain 40% of the variation in the species tree error. The true discordance measured by the quartet score, when log-transformed, seemed to correlate close to linearly with the species tree error. The other two factors had more intricate patterns of impact.

Increasing the number of genes to ≈400 had a strong effect on the species tree accuracy (correlation coefficient: -0.44 in a linear fit). Beyond 400 genes, the impacts on accuracy were diminished (slope: ≈ 0; correlation coefficient: -0.06). When the number of genes was 800 or more, it was rare (5 replicates) that the species tree error was above 0.1 NRF, and when it did, the gene tree error was typically high. Gene tree error also had non-linear effects. When the average gene tree estimation error estimated by NRF was below $\approx \frac{1}{3}$, it had only a small impact on the species tree error, but beyond this threshold, it strongly impacted the species tree accuracy. However, it should be noted that even when the gene tree error is 0.6 NRF or higher, there are two replicates that have a large number of genes and a relatively low true discordance, and in both cases, ASTRAL has close to perfect accuracy. Conversely, when gene tree error is as low as 0.34, a replicate that has only 85 genes and very high discordance (quartet score= 0.37) had a relatively high species tree error (0.2 NRF error). Thus, to predict the accuracy, all three factors have to be considered.

Adequacy of the set X. To test whether the set X, the constrained set of bipartitions searched by ASTRAL, is sufficiently large, we ran ASTRAL with bipartitions of the true species tree forced to be in X. When this enforcement improves the quartet score or the accuracy, we can infer X is not sufficiently large; on the other hand, if making sure X includes all bipartitions of the true tree does not improve the score or the accuracy, then, further expanding X is unlikely to improve the accuracy. In our simulations, the median species tree error and quartet scores remained constant with the addition of true bipartitions and the average improved by only 0.0004 for accuracy and by 3×10^{-6} for the quartet score (Table 4.4). Thus, we conclude that the search space is sufficiently large. Moreover, as expected, $|X|$ is a function of m and k and just like single individual datasets, we observe that $|X|$ grows linearly with mk (Fig. 4.14).

Older versions of ASTRAL-multi. Even though we are describing the multi version of ASTRAL for the first time, previous versions of our algorithm have been made available publicly previously. Some published papers have already used those older versions to obtain their species trees [102, 140, 54]. Compared to previous versions ($< 5.0.0$), the new version is expected to be faster. It is not clear to us whether there are cases where the new version will give better quartet scores. We reanalyzed the 498-locus dataset of 112 species and 163 individuals from the genus Protea [102] using the latest version of ASTRAL-multi and compared the results to version 4.7.9 used by the original paper. The new version produces an identical tree, but the search space X has now reduced from 208505 clusters to 64183 cluster, resulting in a 4X reduction of the running time.

Table 4.3: Running times statistics of running ASTRAL and NJst on D1 in hours measured on the same clusters (Comet). 313 out of 330 replicates are reported here since NJst failed to finish on 17 of them in 48 hours. Also ASTRAL-multi could not finish 3 of them and ASTRAL-multi-5% fails on 6 of them.

	ASTRAL-multi	**ASTRAL-multi-5%**	**NJst**
Mean	3.745	4.124	10.658
Median	1.528	1.571	5.737
STD	5.227	5.995	11.400

Table 4.4: FN and quartet score difference of the species tree constructed from the set of gene trees of D1, with and without adding clusters of true species tree. The values shown are calculated such that positive FN diff means how much adding clusters of true species tree helps and negative quartet score diff shows how much quartet score has increased with adding true species tree bipartitions to search space.

	Quartet score diff	**FN diff**
Mean	-2.98673e-06	0.0004
Median	0.0	0.0
STD	2.00276e-05	0.0066

Algorithm 2: **- Computing similarity matrix.** *getSimilarity* is defined in [100] and gives the similarity matrix of the leaves of its input gene trees, labeled with individuals names.

```
function GETSPECIESSIMILARITY($\mathcal{G}$)
    $GS \leftarrow getSimilarity(\mathcal{G})$
    $S \leftarrow Zeros(n \times n)$
    for $i \in Rows(GS)$ do
        for $j \in Cols(GS)$ do
            $S[i, j] += GS[s(i), s(j)]$
    $D[i] = |\{j | s(j) \in i\}|$
    $S[i, j] = S[i, j]/D[i] \times D[j]$
```

4.6 Acknowledgement

Chapter 4, in full, is a reprint of the material as it may appear in “Multi-allele species reconstruction using ASTRAL” (2019) Rabiee, Maryam; Sayyari, Erfan; Mirarab, Siavash, Molecular Phylogenetics and Evolution, 130, 286-296. The dissertation author was the primary investigator and first author of this paper.

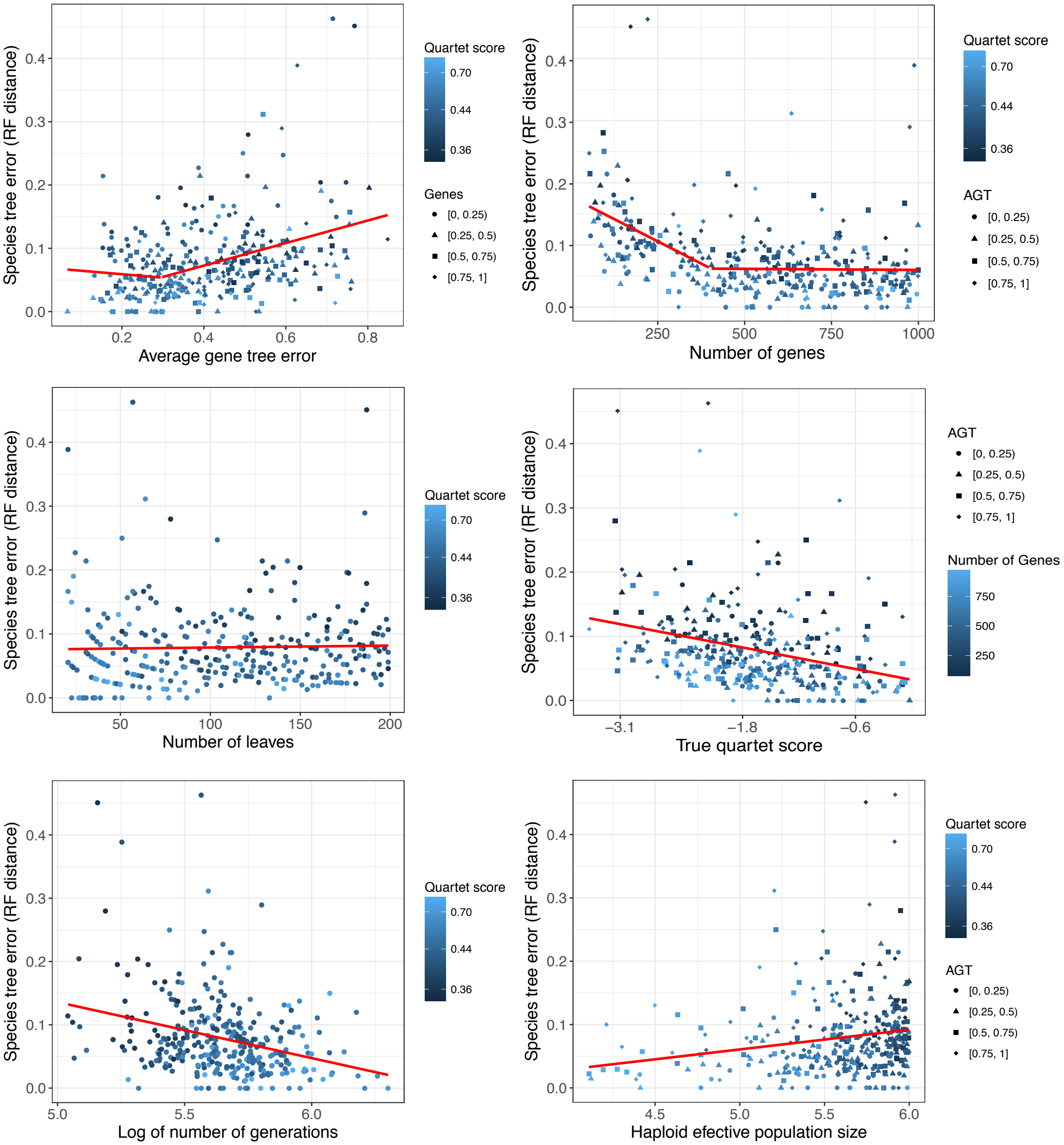


Figure 4.7: **The impact of simulation parameters on the ASTRAL-multi species trees.** We show Robinson Foulds (RF) distance between the ASTRAL-multi species tree and the true species tree (y-axis) versus (a) average gene tree error (AGTE), (b) the number of genes, (c) the number of species, (d) the quartet score of the true species tree based on true gene trees (showing the ILS level), (e) the number of generations in log scale, and (f) haploid effective population size. Shades of blue show the quartet score in all panels except (d) where they show the number of genes. Point shapes distinguish quartiles of AGTE (b, d, f) or the number of genes (a). A linear model (red line) is fitted to the data; in (a,b), a two-segment liner model is fitted, with breakpoints determined using the bootstrap fitting [162] algorithm.

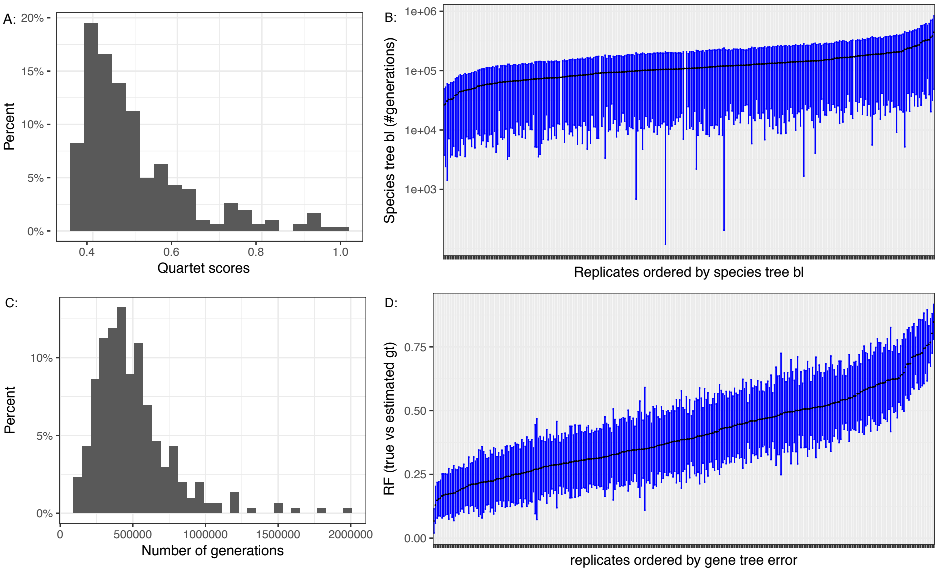


Figure 4.8: **Simulation properties for D1.** A) The empirical histogram of the quartet scores of ture species tree using true gene trees on D1. There are a few replicates with high ASTRAL quartet score. In 90% of replicates the quartet score is between 0.37 and 0.76 and the average ASTRAL quartet score is 0.5. B) The average (points) and standard deviation (bars) of species tree branch lengths measured in the number generations (log scale). The number of generations between speciation events ranges between 10,000 to 1000,000 in most cases. C) The empirical histogram of the tree height measured in the number of generations. D) The average (points) and standard deviation (bars) of the RF distance between true and estimated gene trees.

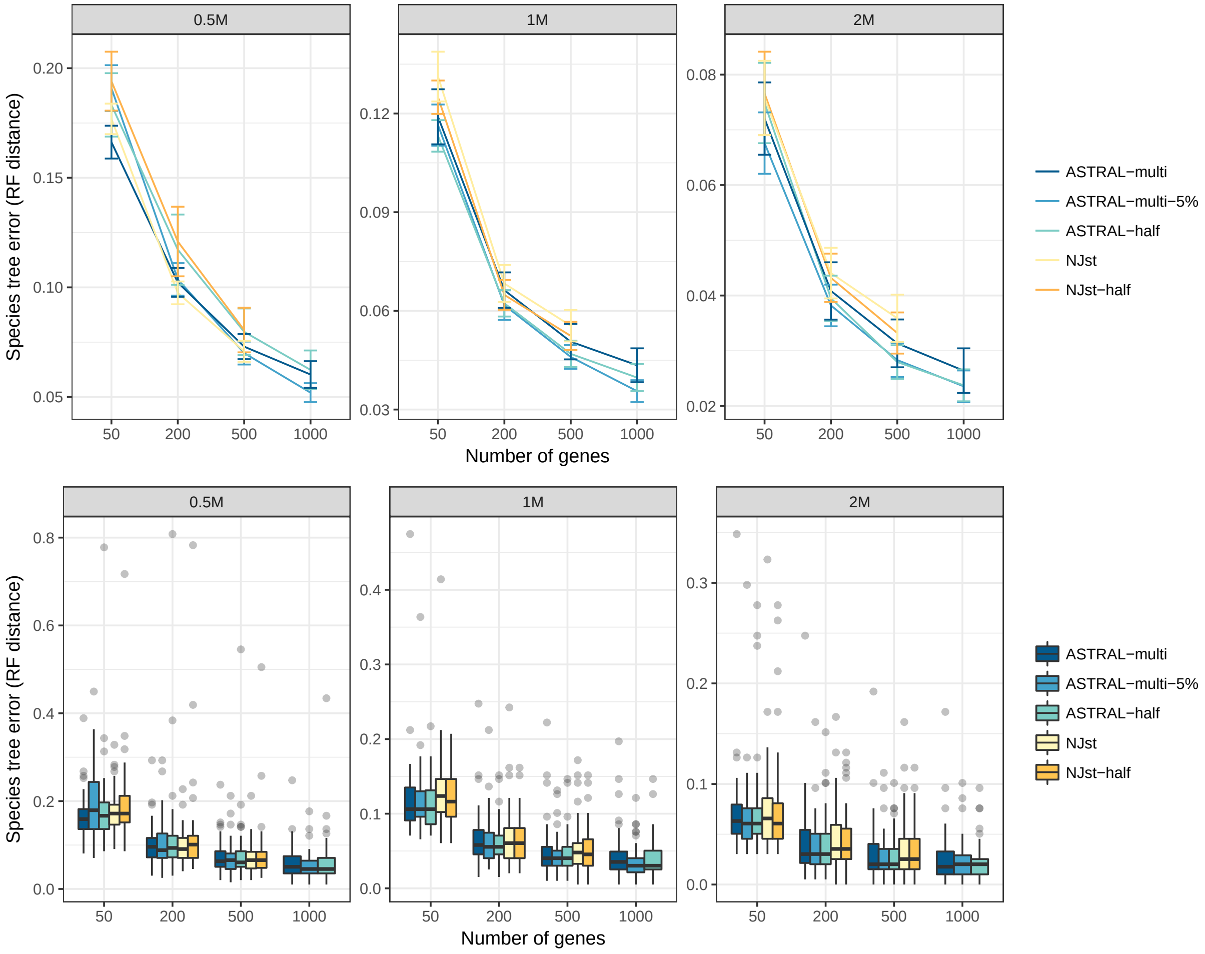


Figure 4.9: Species tree error (RF distance) of species tree constructed from estimated gene trees with different versions for ASTRAL and NJst on the D2 dataset. Top: mean and standard error over all 50 replicates for each model condition; Bottom: The bolxplots over the 50 replicates. ASTRAL-half is like ASTRAL-multi but genes with many polytomies are removed from the input set. Genes with many polytomies are defined to be those where the number of internal branches is at most half of the maximum possible number of internal branches. Similarly, for NJst-half, these genes are removed. This filtering does not produce a clear improvement in NJst or ASTRAL.

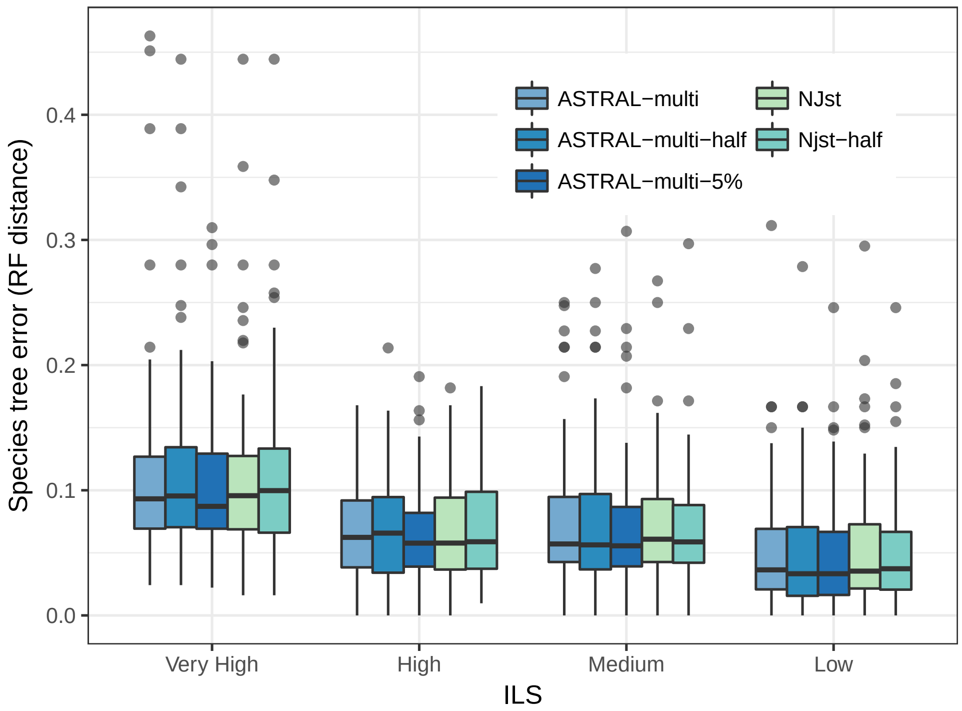


Figure 4.10: Species tree error of three version of ASTRAL versus two version of NJst run on the D1 dataset. The 326 replicates are divided into quantiles according to their ILS level, as measured by the quartet score (bottom). ASTRAL-multi-half is running ASTRAL-multi but genes with many polytomies are removed from the input set. Genes with many polytomies are defined to be those where the number of internal branches is at most half of the maximum possible number of internal branches. Similarly, for NJst-half, these genes are removed. On this dataset, all methods perform similarly.

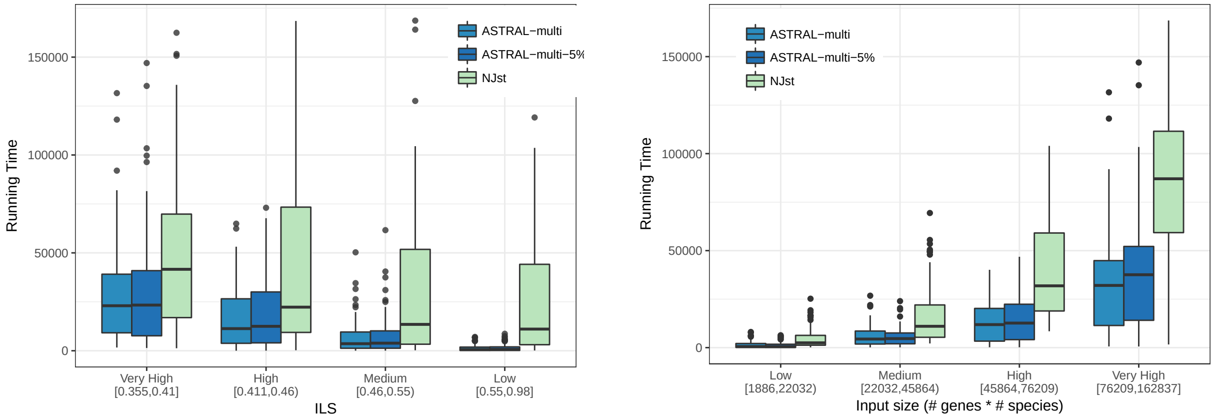


Figure 4.11: Running time comparison between ASTRAL-multi, ASTRAL-multi-5% and NJst in seconds with respect to ILS level and input size defined as number of genes multiplied by number of species. ASTRAL is faster than NJst in all conditions. The running time of the ASTRAL-Multi increases with increased ILS or with larger inputs.

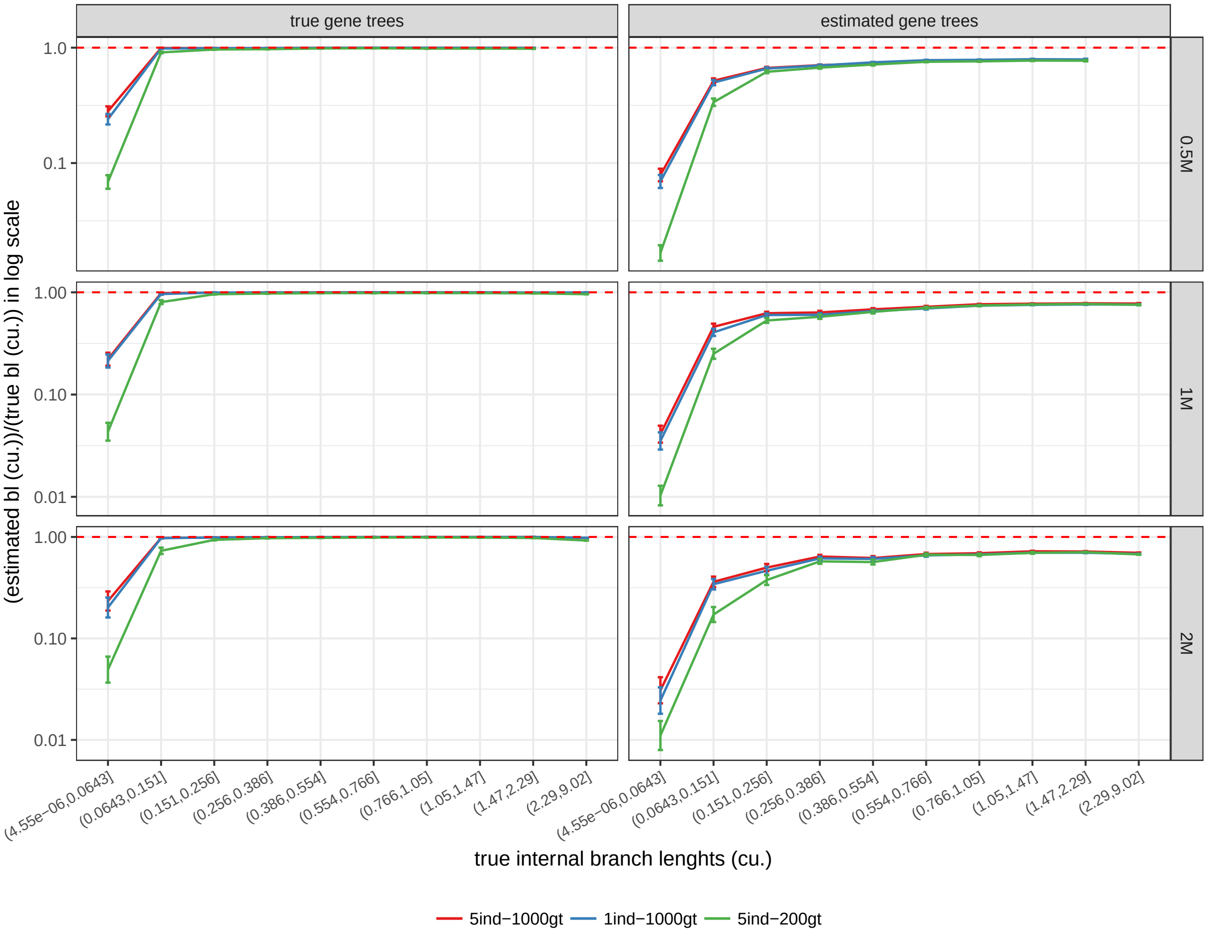


Figure 4.12: Accuracy of internal branch lengths for D2 using true species trees and estimated and true gene trees. x-axis: deciles of true branch lengths, y-axis: average ratio of estimated internal branch lengths to true internal branch lengths in log scales. Dotted red line indicates ratio of 1, which indicates perfect accuracy. solid red line: 5 individuals and 1000 gene trees, solid blue line: 1 individual and 1000 genes, solid green line: 5 individuals and 200 gene trees. Increasing the number of individuals at the expense of the number of genes reduces the accuracy of branch lengths.

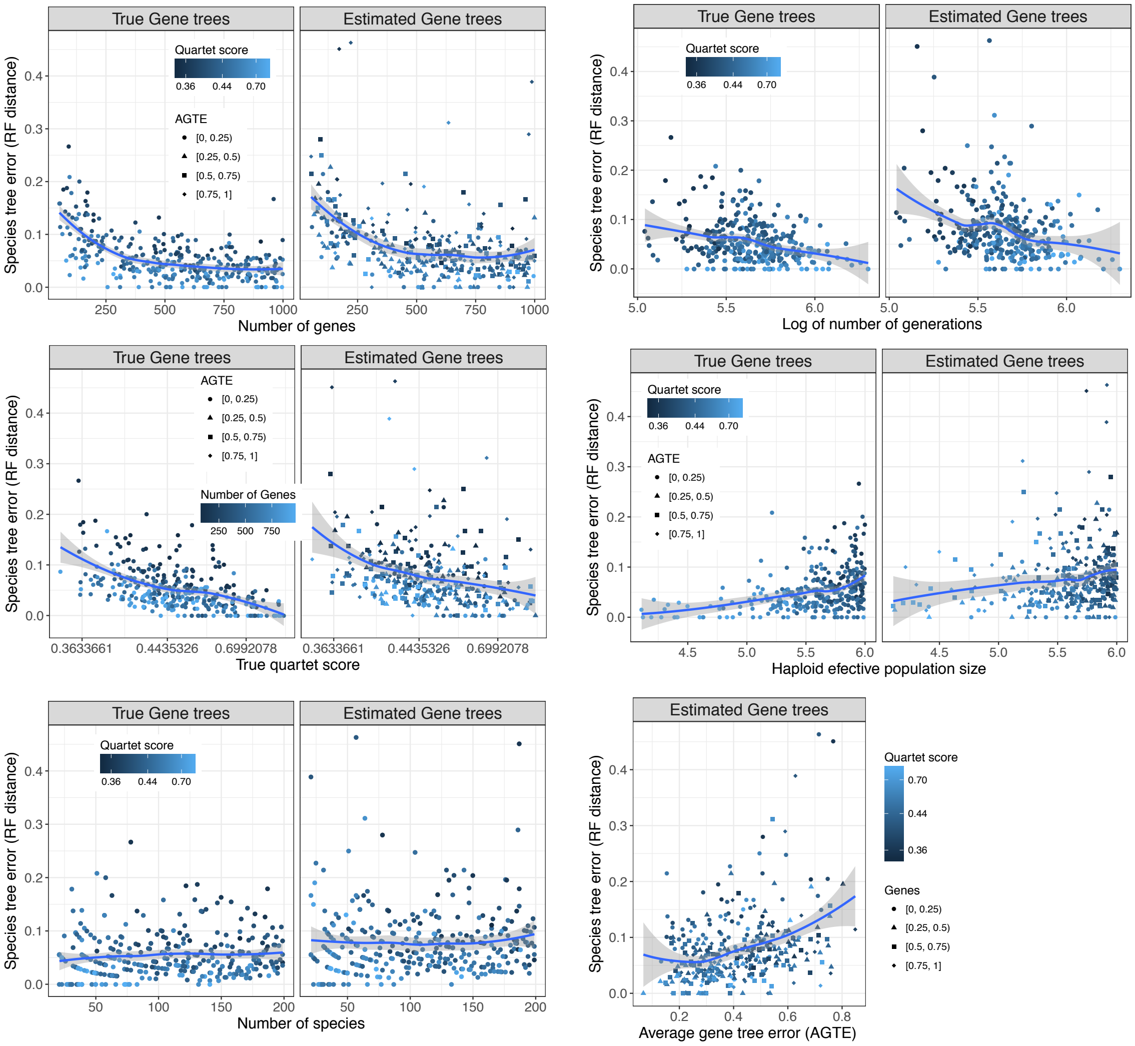


Figure 4.13: FN error of species trees constructed from estimated gene trees with respect to number of genes, log number of generations, true quartet score, haploid effective population size, number of leaves and average gene tree error (AGTE).

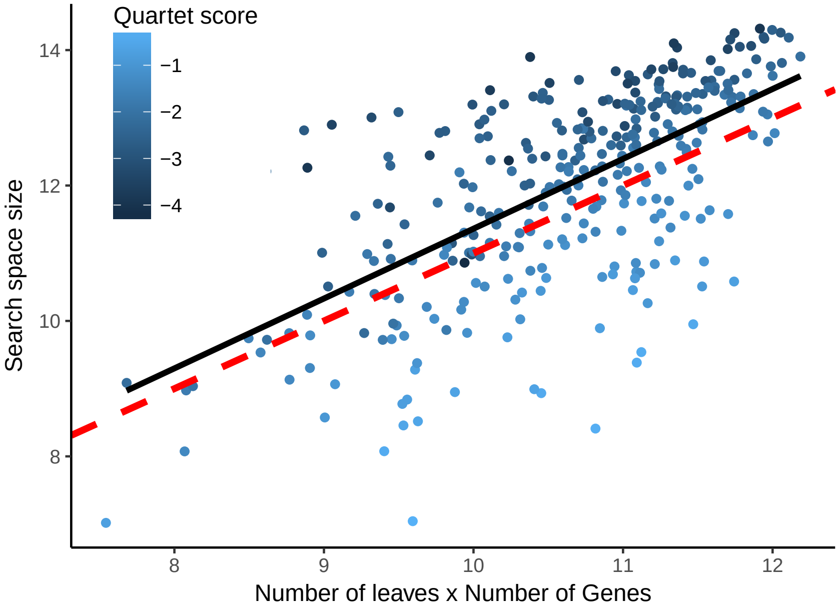


Figure 4.14: Number of clusters in the search space of ASTRAL-multi versus the size of the dataset on the D1 dataset. The search space size is quantified as $\ln(|X|)$ and the dataset set size is $\ln(mk)$ where m is the number of leaves and k is the number of genes. The red dotted line: $\ln|X| = \alpha + \ln mk$. The black line: a line fitted to the data. Colors: $\ln(\frac{3}{2}(q - \frac{1}{3}))$ where q is the quartet score; lighter values show higher quartet scores and thus, less ILS. Increasing the dataset size linearly increases the search space size (because the black line and the red dotted lines have similar slopes). The quartet scores is a large driver of the variation for each dataset size.

Chapter 5

SODA: Multi-locus species delimitation using quartet frequencies

Motivation: Species delimitation, the process of deciding how to group a set of organisms into units called species, is one of the most challenging problems in evolutionary computational biology. While many methods exist for species delimitation, most based on the coalescent theory, few are scalable to very large datasets, and methods that scale tend to be not accurate. Species delimitation is closely related to species tree inference from discordant gene trees, a problem that has enjoyed rapid advances in recent years.

Results: In this paper, we build on the accuracy and scalability of recent quartet-based methods for species tree estimation and propose a new method called SODA for species delimitation. SODA relies heavily on a recently developed method for testing zero branch length in species trees. In extensive simulations, we show that SODA can easily scale to very large datasets while maintaining high accuracy.

5.1 Introduction

Evolution results in diversity across species and diversity within the same species, in ways that can make it difficult to distinguish species. Definitions of what constitutes a species are varied [32] and subject to debate. Nevertheless, many biological analyses depend on our ability to define and detect species. Assigning groups of organisms into units called species, a process called species delimitation, is thus necessary but remains challenging [23, 127]. Among varied species concepts, the most commonly used for Eukaryotes is the notion that individuals within a species should be able to mate and reproduce viable off-springs.

A wide range of species delimitation methods exist. More traditional methods simply relied on the mean divergence between sequences [e.g., 56, 122] or patterns of phylogenetic branch length [171, 53, 48] in marker genes or concatenation of several markers [e.g., 119]. Due to limitations of marker genes [61], many approaches to species delimitation have moved to using multi-locus data that allow modeling coalescence within and across species [69, 166, 113], not to mention more complex processes such as gene flow [e.g., 80]. Modeling coalescence allows methods to account for the fact that across the genome, different loci can have different evolutionary histories, both in topology and branch length [85]. Species delimitation is often studied using the Multi-species Coalescent (MSC) model [115, 126]. In this model, individuals of the same species have no structure within the species, and thus their alleles coalesce completely at random. Coalescence is allowed to happen deeper than the first opportunity, producing gene tree discordance due to Incomplete Lineage Sorting (ILS). In this context, given a set of sampled individuals, delimitation essentially requires inferring gene trees, one per locus, and detecting which delimitation is most consistent with patterns of coalescence observed in the gene trees.

Existing methods for species delimitation under the MSC model tend to suffer from one of two limitations. The most accurate methods are based on Bayesian MCMC and infer gene trees, (optionally) species trees, and species boundaries (e.g., BPP [166, 167], ABC [22], and

STACEY [66]). Other Bayesian methods use biallelic sites [78], incorporate morphological data [145], or use structure [62]. These methods, however, are typically slow and cannot handle even moderate numbers of samples [53, 164]. For example, [106] had to divide their dataset of 62 individuals into six subsets, and [112] used a subsample of 20 out of 137 to use BPP to avoid mixing problems that usually happen with large datasets; running BPP on a dataset of 40 populations in our study needed 36 hours of running time. The second class of methods (e.g., SpedeSTEM [46]) rely on a three-step approach: first, infer gene trees, then, date gene trees so that they all become ultrametric (i.e., have a unique root to tip distance), then, use ML calculation of alternative delimitations under the MSC model to decide species boundaries. These methods have been less accurate than Bayesian methods, and their reliance on ultrametric trees makes them hard to use for datasets where rates of evolution change substantially across the tree [22]. Yet other methods [e.g., 113, 172] rely only on input gene tree topologies, as we do.

In this paper, we introduce a new species delimitation approach called SODA that builds on the success of our species tree inference tool ASTRAL [99, 100, 169]. A statistically consistent method, ASTRAL infers a species tree from a collection of gene tree topologies (ignoring branch lengths) based on the principal that the most frequent unrooted topology for each quartet of species is expected to match the species tree [3]. Thanks to its accuracy and scalability, ASTRAL has been widely adopted for species tree inference. In a recent paper that extended ASTRAL to multi-individual data, we observed that if species boundaries are ignored, ASTRAL most often recovers individuals of the same species as monophyletic [123]. This result suggests a species delimitation method: Infer an ASTRAL tree with all individuals and use patterns of quartet trees mapped onto that species tree to decide where coalescence is completely random and where it is not; these boundaries can define species. By relying on quartet frequencies and the ASTRAL machinery, SODA is able to handle very large datasets with short running times. We first describe SODA in detail and then evaluate its accuracy and scalability in simulation and on empirical datasets.

5.2 Methods

5.2.1 Coalescent-based topology-based delimitation

We take a two-step approach to species delimitation and assume unrooted gene tree *topologies* are already inferred from sequence data. Thus, we are given a set of unrooted gene trees $\mathcal{G}$ on individuals $\mathcal{L} = \{l_1 \dots l_m\}$. Ignoring errors in estimated gene trees, we assume these gene trees follow the MSC model. Optionally, we are given a partition of $\mathcal{L}$ into populations $\mathcal{P} = \{P_1 \dots P_p\}$; when $\mathcal{P}$ is not given, we define each individual as a singleton population. A partition of $\mathcal{P}$ into $\mathcal{S} = \{S_1 \dots S_n\}$ produces a mapping $r : \mathcal{P} \rightarrow \{1...n\}$ and by extension $q : \mathcal{L} \rightarrow \{1...n\}$ where $r(x)$ and $q(y)$ give a species index for $x \in \mathcal{P}$ and $y \in \mathcal{L}$.

We say a partition is coalescent-consistent if for any set $A \subset \mathcal{L}$ with at most one individual from each population and all individuals mapped to the same species ($\forall_{i,j} l_i, l_j \in A : \nexists_x l_i, l_j \in P_x, \exists_y l_i, l_j \in S_y$), the distribution of $\mathcal{G}$ restricted to S is consistent with the neutral [67] coalescence process. Because Kingman's process is robust to subsampling, if a partition is coalescent-consistent, further breaking each species into smaller species would remain coalescent-consistent. Having two species that could be combined without violating the coalescent model is not justified under the MSC because the model provides no support for the division. Thus, we formulate coalescent-based species delimitation as the problem of finding a coalescent-consistent partition that is not a refinement of any other coalescent-consistent partitions. Our method seeks to solve this problem within further restrictions stated below. This formulation follows the MSC model (free coalescent of lineages within species but constrained coalescence across the species) and shares its assumptions. The only population structure within the species that is modeled is the given structure $\mathcal{P}$, which is known *apriori*. Thus, it assumes that individuals selected from different populations of the same species evolve according to the neutral Wright-Fisher model, resulting in a distribution of gene trees within a species that follows the [67] coalescent process. This assumption is most defensible when the populations within a species do not further have *strong* differentiation. MSC also assumes lineages sampled from different species do not coalesce more recently than their separation event (Fig.5.10); thus, it ignores gene flow across species. While these assumptions can all be violated on real data, they provide a useful model that allows fast delimitation. We revisit these assumptions in the discussion session.

The branch lengths of gene trees can be modeled as a function of two processes: coalescent of lineages and changes in the mutation rate [126]. Simultaneously dealing with these two processes is challenging, motivating methods such as SpedeSTEM to take ultrametric (e.g., dated) gene trees as input and forcing Bayesian methods such as BPP to assume parametric rate models. In our work, we are after a fast delimitation method that can be applied to inferred non-ultrametric gene trees directly. To avoid complications of rate variations across lineages, we limit ourselves to gene tree topologies. To do so, we rely on the distribution of gene tree topologies under the MSC model, in particular for quartets of species [39].

Using gene tree topologies, however, has a limitation. Examining the distribution of tree topologies requires at least three lineages. Thus, two species, each with a single individual, and a single species with two individuals cannot be distinguished by topology alone. This forces us to assume that in the correct delimitation, each species has more than one individual sampled (i.e., $\forall_i |S_i| \geq 2$).

5.2.2 SODA Algorithm

A central concept in MSC is the "extended species tree," as defined by [3]. Let T^* be the true species tree on the leafset $\mathcal{S}$. The extended species tree $\mathcal{T}$ is a rooted tree labeled by $\mathcal{L}$, built by adding to each leaf of T^* all individuals corresponding to that species as a polytomy (Fig. 5.1); i.e., for leaf $s \in \mathcal{S}$ of T, add a child for every $r \in q^{-1}(s) \subset \mathcal{L}$. When populations are known *apriori*, we can similarly define the extended species tree $\mathcal{T}$ as a rooted tree labeled by $\mathcal{P}$, built by adding to each leaf of T^* all populations corresponding to that species as a polytomy.

Our species delimitation method, which we name Species bOundry Delimitation using

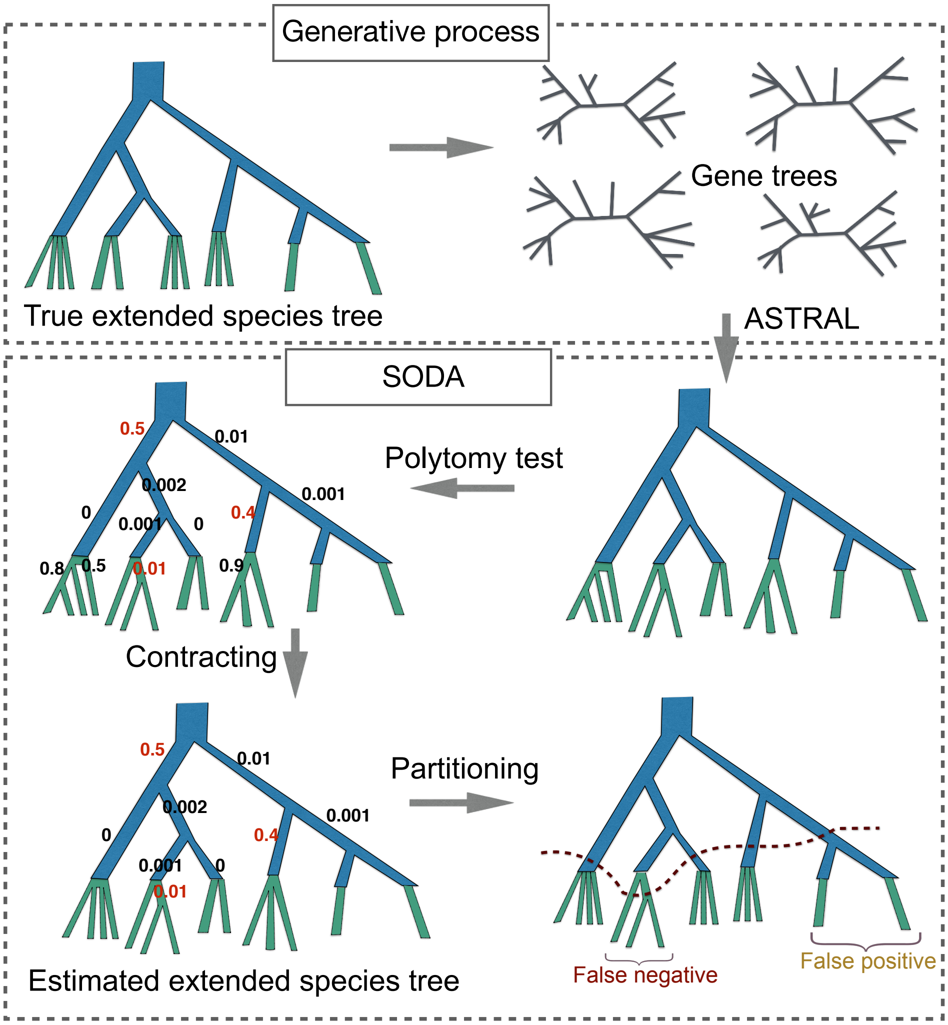


Figure 5.1: True extended species tree generates gene trees under the MSC model. From unrooted gene tree topologies (inferred from sequence data), SODA first estimates a guide tree using ASTRAL and then test the null hypothesis that each branch has length zero, obtaining a *p*-value (middle left), which may result in FP or FN rejection or retention of the null (red *p*-values). SODA then contracts branches where the null is retained as long as contracting them does not contradict with the monophyly of species defined by branches where the null hypothesis is rejected; thus, we keep some branches with high *p*-value (e.g., those with *p*-value 0.4 and 0.5) in a way that ensures the resulting tree can be an extended species tree (bottom left). The inferred extended species tree can be cut at branches above the terminal branches to define species. The result could include both false positives and false negative delimitations. However, we note that some errors in hypothesis testing (e.g., *p*-values 0.5 and 0.4) do not result in erroneous delimitation.

Algorithm 3: **SODA Algorithm**

```
function SODA($\mathcal{G}$, $T$,$\alpha$)
    if guide tree $T$ is not given then
        $T \leftarrow$ run ASTRAL on $\mathcal{G}$
    for internal branch $e$ of $T$ do
        $p(e) \leftarrow p$-value of the null hypothesis that length$(e) = 0$
    Root $T$ on the $\arg\min_e p(e)$
    CONTRACTTOEXTENDEDSPECIESTRE($T$,$\alpha$)
    partition $\mathcal{P}$ by cutting all “keep” internal edges of $T$
function CONTRACTTOEXTENDEDSPECIESTREE($T$,$\alpha$)
    for $e \in$ internal edges of $T$ do
        if $p(e) \leq \alpha$ then
            Mark $e$, sister of $e$, and all ancestors of $e$ as “keep”
    for $e \in$ internal edges of $T$ not marked “keep” do
        Contract $e$
```

Astral (SODA), is shown in Algorithm 3. Its inputs are a set of gene tree topologies and a significance level α, described below. SODA first infers (or takes as input) a guide tree T. The guide tree, a concept introduced by [166], is a phylogenetic tree with leaves set to known populations (i.e., the most divided possible delimitation). SODA assumes T is a *resolution* of the true extended species tree $\mathcal{T}$, meaning that it includes all branches of the species tree and has arbitrary relationships between populations of the same species. SODA then contracts *some of the* branches of T as long as it cannot reject the null hypothesis that they have length zero under the MSC model using α as the confidence level for statistical tests; the contracted tree is an estimated extended species tree $\hat{\mathcal{T}}$. Finally, SODA cuts internal branches of the $\hat{\mathcal{T}}$ to cluster species (Fig. 5.1).

Below, we describe each step in more detail.

Guide Tree: SODA needs a (potentially unrooted) guide tree T on the leafset $\mathcal{P}$. If not provided by the user, we infer the guide tree by running ASTRAL-III on $\mathcal{G}$; note that using the multi-individual version of ASTRAL [123], we can ensure that the tree generated by ASTRAL is labeled by $\mathcal{P}$, as opposed to original individuals (if different). We assume that T is a resolution of the extended species tree; thus, the accuracy of this guide tree is important.

Polytomy Test: For each branch of T, we next test the null hypothesis that it has zero length in coalescent units. If we cannot reject this null hypothesis, the branch can be collapsed to obtain a polytomy, helping us to obtain $\hat{\mathcal{T}}$. We use a recent test proposed by [137] that relies on a classic result: Under the MSC model, across gene trees, the frequencies of the three resolutions for each quartet around a given branch in the species tree are equal if and only if that branch has length zero [115, 3]. Moreover, gene trees are assumed independent. Thus, under the null hypothesis of a polytomy, the frequency of quartet topologies around each branch should follow a multinomial distribution with three categories, each with probability $\frac{1}{3}$. Whether observed frequencies of the three possible topologies for a quartet follow an equiprobable multinomial distribution can be tested using a Chi-Squared test, which is what the method of [137] uses. To achieve scalability, this method treats branches independently (based on the locality assumption of [136]) and takes the average of quartet frequencies for all quartets around each branch (which can be done easily in $O(n^2k)$ time). The test assumes the input the gene tree set is an error-free random sample generated by the MSC model from the true species tree. It produces one p-value per *internal* branch.

Rooting T: We need to root T (if not rooted) such that each species becomes monophyletic. We simply root T at the edge with the minimum p-value. Note that our goal is not to find the correct root because we do not need the correct rooting in the next steps. We only need the tree to be rooted on *any* internal branch of the extended species tree. The highest statistical confidence for having a positive length is achieved by the branch with the lowest p-value; thus, we can root here.

Infer extended species tree: To obtain $\hat{\mathcal{T}}$, we contract some of the branches of T where a zero-length null hypothesis cannot be rejected at a user-specified level α. When the null hypothesis is rejected for a branch e, we marked e as being part of $\hat{\mathcal{T}}$ (i.e., *keep* in Alg. 3). Parameter α can be adjusted for controlling how aggressively SODA divides species. Increasing

α results in rejecting more null hypotheses and hence, dividing individuals into more species. To ensure that we can get a valid extended species tree (with monophyletic species), we need polytomies to form only above the terminal branches. Thus, in addition, we mark the sister edge of e and all its ancestor edges as belonging to $\hat{\mathcal{T}}$.

Partitioning: Given $\hat{\mathcal{T}}$, the partition is obtained by cutting the remaining internal branches of $\hat{\mathcal{T}}$. The partition produced would be identical if we only cut internal branches that have at least one terminal branch as a child.

The accuracy of the algorithm, in addition to assumptions made by our problem formulation, depends on the accuracy of the statistical test. We formalize this notion in three claims (proofs in supplementary material).

Claim 1. *Assuming (i) gene trees $\mathcal{G}$ are generated under the MSC model on an extended species tree $\mathcal{T}$, (ii) the guide tree T (e.g., ASTRAL tree) is a resolution of $\mathcal{T}$, and (iii) the hypothesis testing has no false positive (FP) or false negative (FN) errors, the SODA algorithm returns the correct extended species tree ($\hat{\mathcal{T}}=\mathcal{T}$). Additionally, it will correctly delimitate species if all species are sampled more than once.*

This claim provides a reassuring result, but only under strong assumptions, most notably, that the test is perfect. However, errors in the test *can* lead to errors.

Claim 2. *Given a guide tree T that resolves the tree $\mathcal{T}$, SODA incorrectly divides a species S into multiple species (i.e., a false negative error) if and only if the zero-length hypothesis testing results in an FP error for one of the branches under the clade defined by S on T.*

Thus, FPs in the hypothesis test always result in the division of a species; however, an FP does not always divide S into exactly two parts. For example, if T has a caterpillar (a.k.a ladder-like) topology on S, an FP on the branch above the cherry (i.e., a node with two leaf children) leads to each remaining individual being marked as a species.

Claim 3. *Given a guide tree T that resolves $\mathcal{T}$, SODA incorrectly combines individuals from two species S_1 and S_2 into one species (a false positive error) under one of these two conditions. 1) S_1 and S_2 each have one sampled individuals and form a cherry. 2) The hypothesis testing has an FN error for all branches of $\mathcal{T}$ below the LCA of S_1 and S_2. This condition requires FN errors for two or more branches if neither species is a singleton.*

To combine two species incorrectly, we must fail to correctly mark all branches below their LCA; else, one of the branches below the LCA would be cut, which would prevent the FP. Thus, our approach is tolerant of some FN errors in the hypothesis test (e.g., p-value 0.4 in Fig. 5.1).

5.3 Experimental setup

5.3.1 Datasets

We used two simulated datasets, both generated using Simphy [89]. One dataset is large and allows us to evaluate SODA on conditions where other methods cannot run, whereas the other dataset is small and enables us to compare SODA to slower Bayesian methods.

Large dataset: We reuse a 201-species "homogeneous" dataset that we have previously simulated [123] using SimPhy to generate species trees based on the birth/death model and gene trees under the MSC model. Each species includes five individuals except the singleton outgroup (1001 in total). We have three model conditions (50 replicates each) with medium, high, or very high levels of ILS, with maximum tree height set to 2M, 1M, and 0.5M generations, respectively. The mean quartet score of true gene trees versus the species tree (i.e., the proportion of quartet trees that are shared between the two trees) is 0.78, 0.62, and 0.50 for these model conditions (Fig. 5.7). Note that a quartet score of $\frac{1}{3}$ indicates random trees, and even a value of 0.5 indicates very high levels of ILS. The proportions of branches in the true extended species tree missing from gene trees are 0.40, 0.57, and 0.77. Each replicate has 1000 genes, which we down-sample randomly to 500, 200, and 100. To evolve nucleotide sequences down the gene trees, we use INDELible [51] with the GTR+Γ model of sequence evolution with randomly sampled sequence lengths from a LogNormal distribution (empirical mean=721). The gene trees are estimated using FastTree [121] from the alignments; estimated gene trees are used throughout the experiments in this paper. The average gene tree error, measured as normalized [129] (RF) distance for the three model conditions is 0.25, 0.31, and 0.42 with large variance (Fig. 5.7).

Small dataset: We simulate a new dataset using SimPhy with 20 replicates, each only four species, ten individuals per species, and 1000 genes per replicate (commands shown in Appendix 1). The tree height is set to 200,000 generations, the population size is drawn uniformly between 10,000 and 500,000, and species trees are generated using the birth-only model with rate=0.00001. These settings lead to a high level of ILS, capturing a scenario where species delimitation is challenging. The quartet score of true gene trees versus the true species tree is 0.76, and 65% of extended species tree branches are on average missing from true gene trees. We deviate from ultrametricity by drawing rate multipliers for species and genes from Gamma distributions, with LogNormal priors on parameters of Gamma (Table S1). We simulate 1000bp alignments on each gene tree using INDELible [51] and estimate gene trees using FastTree based on the alignments [121]. The average gene tree error (normalized RF between true and estimated gene trees) is 43%. Gene tree distance between pairs of individuals of the same species has a wide range in our simulations, ranging between 10^{-5} and 10^{-2} mutations per site in most cases (Fig. 5.7).

Empirical dataset: We study three biological datasets. To show applicability on large data, we use the dataset of Protea L. with recent radiations [103], which has sampled multiple individuals from 59 species of Protea and six outgroup species (a total of 163 tips) and obtained 498 low-copy, orthologous nuclear loci. Due to its size, this dataset is only analyzed using SODA. To be able to compare to other methods, we use a smaller dataset of lizards of the Australian wet tropics (AWT). [142] analyzed genetic data for individuals from three species groups (*Carlia rubrigularis*, *Lampropholis coggeri*, and *Lampropholis robertsi*) that split into 13 putative lineages. In total, there are 25 individuals, and 3320 loci across all individuals are sequenced using an exome capture approach. Gene trees and the species tree are estimated using STARBEAST2 v0.13.5 [108], and delimitation results using STACEY [66] and BPP are available from the original study. We also study the human dataset analyzed by [63] comprising sequences from 50 loci (415 to 960 bp long) for four widely sampled groups of humans defined geographically (as the original study states: these four groups are ethnically diverse and are not "populations" in any biological sense). The dataset includes ten samples from each of Africa, Europe, and Asia, and 12 samples from South and Central America. Analyzing the human dataset, where we clearly know all individuals belong to one species, enables us to test the propensity of the method to lead to false positive delimitation.

5.3.2 Measures of accuracy

We evaluate accuracy using two measures.

ROC. Each pair of individuals is categorized depending on whether they are correctly grouped together (TP), correctly not grouped together (TN), incorrectly grouped together (FP), or incorrectly not grouped together (FN). We then show recall $= \frac{TP}{(TP+FN)}$ and $FPR = \frac{FP}{(TN+FP)}$ on a Receiver Operating Characteristic (ROC) curve as we change the α setting of SODA.

Adjusted Rand Index is a similarity measure between two partitions of a set, also based on pairwise comparisons. We report ARI between the true species partition and the partition estimated by each method. The [124] index is $\frac{TP+TN}{TP+TN+FP+FN}$. The adjusted rand index (ARI) adjusts RI for the expected similarity of pairs according to a generalized hypergeometric distribution that controls for the number of objects and classes [60]. ARI equals one only for the correct partition and is close to zero for a random partition.

5.3.3 Methods compared

BPP. We compare SODA to the widely used Bayesian Phylogenetics and Phylogeography (BPP) method. BPP uses MCMC for inferring species boundaries directly from sequence alignments by sampling gene trees and other model parameters (e.g., rates) under the MSC model. We use BPP 4.1.4 and take advantage of its multi-thread version to be able to run it with up to 1000 genes. Provided with all 40 individuals of the small dataset that we sampled as 40 separate populations, BPP could not run to completion in 36 hours, perhaps because the set of possible delimitations was too large. To be able to test BPP, we use two subsampled sets. First, we randomly sample 4 individuals per species and designate each as its own population, for a total of 16 populations. In the second scenario, from each species, we sample 7 individuals and randomly assigned them to 3 populations of sizes 2, 2 and, 3, comprising 12 populations in total. We use a uniform prior across all possible partitions on the resulting sets of populations. BPP calculates the posterior probability for each partition, and we use the delimitation with the highest posterior probability to measure the accuracy of the method.

We explore settings of BPP as follows. The total number of MCMC iterations is set to 208000, with the first 8000 discarded as burnin. We also run BPP with twice the number of iterations in one experiment with 500 genes to ensure convergence is not an issue. For the required species tree (similar to the guide tree for SODA), we run BPP in two ways. By default, we provide BPP with the ASTRAL species tree (just as we do for SODA). The guide trees are rooted to match the true species tree. However, BPP is able to jointly infer species trees and species delimitation based on sequence alignments; we also run BPP with this co-estimation setting. This setting makes BPP 2× slower (even though we only have four species), making it impractical for an extensive study. The priors for the inverse gamma distribution parameters, (α, β), were chosen to be (1.525,0.0001), (1.525,0.001) and (1.525,0.01) for population size θ_s and twice the β for τ_s. The mean of the distribution is set based on the true average of species pairwise distances in the gene trees. An example of a control file used for running BPP is given in Figure 5.4 for the full set of parameters.

SpedeSTEM. Given a set of rooted ultrametric gene trees, SpedeSTEM uses the STEM [71] algorithm to calculate the maximum likelihood species tree considering possible species tree and delimitation combinations and uses AIC to select among the models. SpedeSTEMv0.9 software includes a pipeline for inferring ultrametric rooted gene trees using paup* [149]. We used SpedeSTEMv0.9 to infer the rooted ultrametric input gene trees, which we then fed to the SpedeSTEMv2 software as input. We ran delimitation using theta values of 0.1 and 0.01, and a sampling ratio of 1 (no subsampling). SpedeSTEMv2 also requires a putative assignment of populations to species. For this, we tested assigning all populations to the same putative species or assigning all 12 populations to individual species. Results were similar, and we report the latter strategy.

SODA. We implemented SODA in python using Dendropy [147]. We vary α between 0.005 and 0.5 but designate $\alpha = 0.05$ as default. We infer the guide tree using ASTRAL-III on all 1000 genes. To study the impact of the guide tree, we also create the true extended species tree and resolve its polytomies randomly and use this guide tree as input to SODA. For tests with known populations, we inferred the guide tree using ASTRAL-multi, mapping leaves of the same population to a "species," assigning each population to a separate species.

5.4 Results

5.4.1 Large simulated dataset

On the large dataset with 1001 individuals, SODA takes no more than 35 minutes (Table S2) and is highly accurate (Fig. 5.2). Given 1000 genes, default SODA ($\alpha = 0.05$) is able to recover, on average, 183, 186, and 189 out of the 201 species entirely correctly for the three model conditions in decreasing order of ILS (Fig. 5.2a). The total number of species estimated by SODA ranges between 183 and 220 across all replicates with mean 208, which slightly over-estimates the correct number. The number of detected species increases as α increases; however, the number of correct species does not always increase. Reducing the number of genes reduces the number of correctly estimated species (down to 158 with 100 genes, very high ILS); however, it does not change the total number of species dramatically (for default α).

Some of the mistakes made by SODA are related to the guide tree. The ASTRAL tree failed to recover on average 5, 3, and 3 species as a monophyletic clade in these three model conditions, and these species could never be recovered correctly by SODA. On average, the estimated guide tree by ASTRAL (on 1000 genes) missed 4%, 3%, and 2% of branches in the true extended species tree for high, moderate, and low levels of ILS. The *estimated* extended species tree that SODA outputs has an RF distance of 6%, 5%, and 4% to the true extended species tree.

Table 5.1: ARI on small datasets. We show mean (standard deviation) across replicates. ARI of SODA has been measured with two thresholds (0.05 and 0.1).

	SODA			BPP				
Genes	0.01	0.02	0.05	A10(p1)	A10(p2)	A10(p3)	A10(p1)*2	A11(p1)
500	0.70 (0.24)	0.72 (0.25)	0.75 (0.26)	0.85 (0.20)	0.86 (0.16)	0.85 (0.19)	0.86 (0.19)	0.89 (0.15)
1000	0.78 (0.26)	0.79 (0.26)	0.80 (0.23)	0.90 (0.12)	-	-	-	-

Examining pairs of individuals, we observe very high accuracy. With $\alpha = 0.05$, the ARI ranged between 0.95 and 0.97 for our three conditions given 1000 genes and between 0.87 and 0.92 when given as few as 100 genes (Fig. 5.2b). Reducing α to 0.005 or increasing it to 0.1 can reduce or increase ARI slightly; however, increasing α beyond 0.1 can quickly lead to substantial reductions in ARI (Fig. 5.2b). The best choice of α is always between 0.01 and 0.1, but 0.05 is never far from optimal, motivating us to use it as default. Since our simulated replicates are very heterogeneous in terms of gene tree estimation error, we can also examine the impact of mean gene tree error on the accuracy of SODA. Except for an outlier replicate with very high gene tree error and low ARI < 0.9, we do not detect a strong correlation between gene tree error and accuracy (Fig. 5.8). However, the guide tree and the extended species tree *are* impacted by increased gene tree error (Fig.5.9). As the increased error does not impact delimitation, the

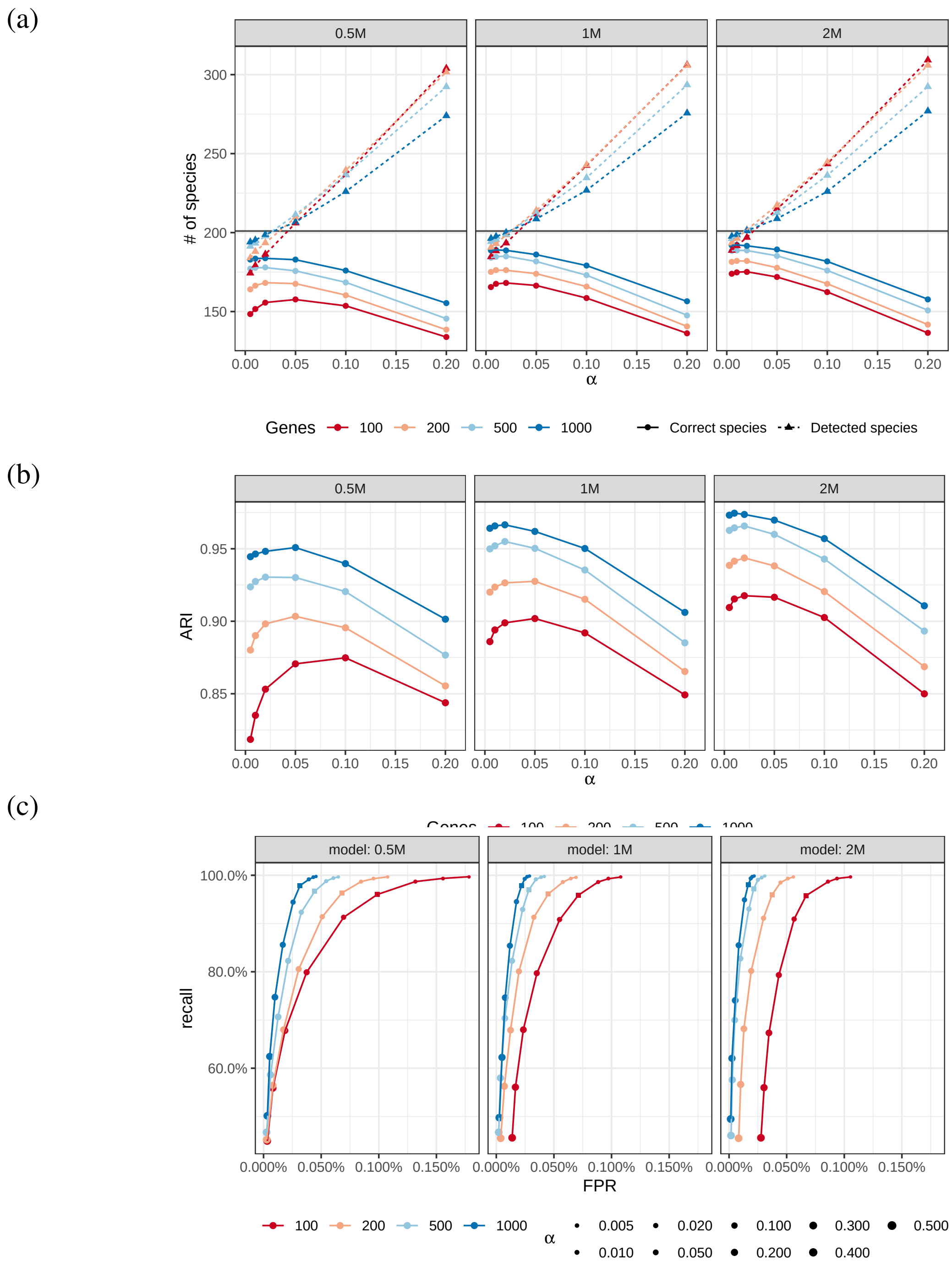


Figure 5.2: **Accuracy of SODA on the large dataset.** (a) We show the number of species that are completely correctly delimited (solid lines) and the total number of species found by SODA (dashed lines). Results divided into three model conditions with very high ILS (0.5M), high ILS (1M), and moderate ILS (2M) as we change α (x-axis) and the number of genes (colors). We clip α at 0.2 but show full results in Figure. 5.5. (b) ARI (y-axis) shows the accuracy of SODA. (c) ROC showing recall versus False Positive Rate (FPR) for all model conditions and different choices of α (dot size). The default value shown as a square.

increased error of the guide tree must be concentrated on deep branches that do not impact delimitation.

The trade-off between precision and recall with different choices of α can be examined using the ROC curve (Fig. 5.2c). With $\alpha \leq 0.05$, recall is always 96% or higher and is often close to 100% with $\alpha \leq 0.01$. The FPR, however, is strongly impacted by the number of genes. For example, with default α, FPR is never more than 0.03% with 1000 genes but increases to 0.98% with 100 genes. Increasing α reduces FPR; however, for $\alpha > 0.1$, we observe only small gains in FPR but precipitous declines in the recall. Thus, as observed earlier, $\alpha > 0.1$ does not seem advisable. Beyond the default value, a choice of $\alpha = 0.01$ seems desirable if more FP combinations can be tolerated. ROC curves also reveal interesting patterns in terms of the impact of ILS on the accuracy of SODA. For $\alpha = 0.05$ and a fixed number of genes, increasing ILS increases FPR (combining species) but does not substantially impact recall. Finally, using a random resolution of the true extended species tree as the guide tree has only a small positive impact on the accuracy (Fig. 5.6).

5.4.2 Small simulated dataset

On the small dataset with four species and 16 populations, SODA (default) has 98% recall with both 500 and 1000 genes (Fig. 5.3a). Increasing the number of genes mostly reduces FPR, from 14% with 500 genes to 11% with 1000 genes. Changing α trades off FPR and recall in expected ways; e.g., with $\alpha = 0.02$, recall is 100% but FPR increases to 12% for 1000 genes and 17% for 500 genes.

Compared to SODA, BPP has a lower FPR, ranging between 4% and 6% for 500 genes and 3% for 1000 genes. However, the recall of BPP is not better than the default SODA and ranges between 92% and 96%, depending on the setting used. Just like SODA, an increased number of genes improves the FPR of BPP but not its recall. Overall, SODA-default seems to err on the side of combining individuals, while BPP tends to over-split species. Judging by the

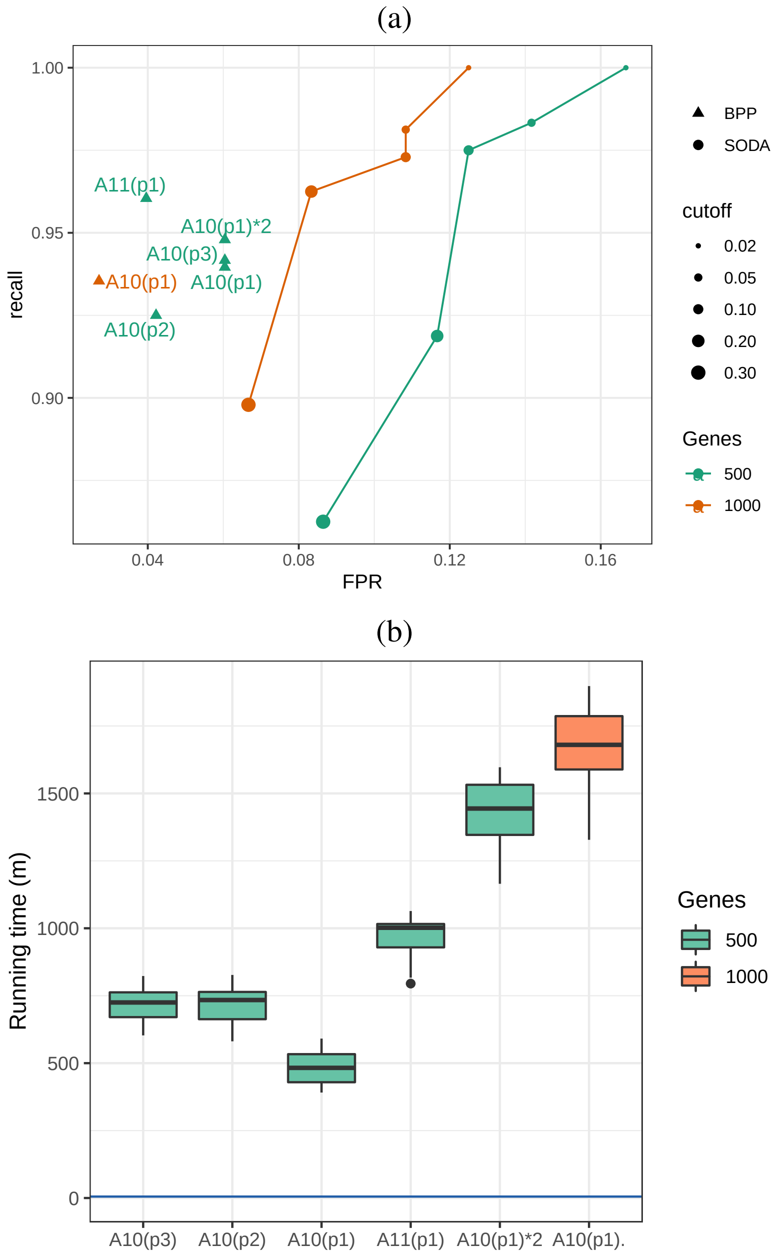


Figure 5.3: **Results on the small dataset.** (a) ROC curves for SODA and BPP on the small 4-taxon dataset with 500 or 1000 genes (colors) averaged over all replicates. BPP with 500 genes has several settings: three prior values for θ_s are p1=IG(1.525,0.0001), p2=IG(1.525,0.001) and p3=IG(1.525,0.01). A11 indicates species tree co-estimation while A10 indicates using ASTRAL as the guide tree; A10(p1)*2 indicates doubling the number of MCMC iterations. (b) The running time of BPP with various settings. The blue horizontal line shows the running time of SODA, including gene tree estimation. Both methods are run on Intel Xeon E5-2680v3 processors; however, SODA uses one core while we ran BPP with 4 threads and 4 cores.

ARI (Table 5.1), BPP has better accuracy overall; e.g., SODA-default has an ARI of 0.78 on 500 genes, while ARI of BPP ranges between 0.85 and 0.89. Overall, the parameter choices for BPP do impact accuracy but not in major ways. Doubling the number of iterations has limited to no impact, and the two choices for the prior were almost identical. A third setting resulted in lower FPR but also lower recall (Fig. 5.3a). The only parameter that increases accuracy substantially is species tree co-estimation, which improves recall by 3.5% and reduces FPR by 0.2%.

The slightly higher accuracy of BPP comes at a steep price in running time (Fig. 5.3b). BPP takes between 400 and 1900 minutes on these data, given four cores. In contrast, SODA never takes more than a minute, and the gene tree estimation takes a few minutes (≈ 5) for this dataset. Deviating from our default setting further increases the running time of BPP, with little impact on accuracy. For example, doubling the number of iterations results in a $3\times$ increase in running time, and asking BPP to co-estimate the species tree results in a $2\times$ increase.

We next compare SODA to BPP and SpedeSTEM on the setting where for each species, seven individuals divided into three populations known *apriori* are given (a total of 12 populations). In this setting, the FP rate of SODA is 8% on average, showing that species are sometimes combined together; in contrast, FN is 0, meaning that over-splitting does not occur (Table 5.2). BPP outperforms SODA in terms of FP (1 to 2%) but also occasionally over-splits (FN>0). Overall, according to ARI, BPP remains somewhat more accurate. SpedeSTEM, in comparison, has much lower accuracy; in almost all cases, it detects exactly two species (instead of four), leading to very high FP rates and low ARI. We note that in the STEM species tree inferred, the species are often non-monophyletic.

5.4.3 Empirical dataset

The ARI of SODA on the Protea dataset ranges from 0.61 to 0.66 with different values for α, given all 498 gene trees available (Table 5.3). This dataset includes many singleton species (12 out of 59 ingroups and 5 out of 6 outgroup species) along with many non-monophyletic clades; to

Table 5.2: ARI on small datasets with individuals assigned to populations. Three populations per species were defined randomly with 3, 2, and 2 individuals per population. We show mean (standard deviation) across replicates. A11 indicates species tree co-estimation while A10 indicates using ASTRAL as the guide tree.

	SODA	BPP		SpedeSTEM	
	0.05	A10	A11	$\theta = 0.1$	$\theta = 0.01$
FP	0.08 (0.12)	0.01 (0.02)	0.02 (0.03)	0.65 (0.11)	0.78 (0.11)
FN	0	0.01 (0.03)	0.01 (0.02)	0.03 (0.01)	0.0 (0.0)
ARI	0.82 (0.22)	0.93 (0.12)	0.89 (0.13)	0.03 (0.11)	0.05 (0.12)

Table 5.3: Delimitation accuracy measured using ARI for the Protea dataset with cutoff thresholds 0.01, 0.02, and 0.05. "Monophyletic species" means individuals of the species that make them non-monophyletic in the ASTRAL tree are pruned (19 in total). We run SODA on fully resolved gene trees and gene trees with branches with BS support below 5% or 10% collapsed.

Species	Gene trees	0.01	0.02	0.05
All species	fully resolved	0.662	0.655	0.611
	<= 5 BS contracted	0.653	0.654	0.637
	<= 10 BS contracted	0.654	0.654	0.628
Monophyletic species	fully resolved	0.776	0.767	0.717
	<= 5 BS contracted	0.736	0.775	0.765
	<= 10 BS contracted	0.736	0.785	0.755

test the accuracy of delimitation (as opposed to species tree inference), we removed individuals that form non-monophyletic clades (19 in total) and gained better delimitation with ARI ranging from 0.72 to 0.78 on the pruned species tree.

SODA-default detects 89 species and is thus over-splitting some species. Some of the split species seem to capture populations within species. For example, *Protea Acaulos* is divided into two groups that coincide with grouping based on the geographical locations of the samples (provided by the original study); these geographical separations may reflect two sub-population of this species.

We also tested collapsing low bootstrap support branches in gene trees to deal with the effects of high gene tree estimation error. Collapsing very low support branches improved the results slightly with the default $\alpha = 0.05$ (Table 5.3). The improvement is more clear when we prune non-monophyletic species.

Human. SODA results support grouping all individuals into one species, and all the p-values across the tree are above 0.34. Thus, on this dataset, reassuringly, SODA avoids a false positive breakup to multiple species. While the recovery of humans (a relatively recent species) as one species may seem an easy case of species delimitation, [63] showed that BPP supports a four-species model with high posterior in all their ten replicate runs, regardless of the prior used. The authors attributed this error to population structure and used this as a motivation to introduce the PHRAPL method, which unlike BPP, did unambiguously recover humans as a single species.

Lizards. Statistical species delimitation using both BPP and STACEY support a speciation event at every node of the guide tree (regardless of priors chosen). SODA, similar to BPP and STACEY, detects all lineages as separate species with p-value≈ 0 for all branches except one branch, which does not result in a false positive (Fig. 5.11). [142] report that the delimitation into 13 groups does not match a clear morphological separation between species, making this a potential case of a cryptic species. However, it should be noted that in the light of the results from humans, a false positive delimitation by all methods cannot be ruled out.

5.5 Discussion

We designed SODA, an ultra-fast and relatively accurate method for species tree delimitation. SODA relies on frequencies of quartet topologies to decide whether each branch in a guide tree inferred from gene trees is likely to have strictly positive length, using results to infer an extended species tree, which then defines species boundaries. SODA focuses exclusively on the MSC-based species delimitation, as applicable to Eukaryotes. It is not designed for defining viral quasispecies [42, 155].

Our method, like many of the existing methods, is based on several strong assumptions.

Most importantly, it ignores the population structure within species and does not consider gene flow. The presence of gene flow across species or population structure can lead to over-splitting for other methods like BPP, as several recent studies demonstrate [23, 63, 148, 80]. Our results on the Protea dataset indicate that SODA can suffer from a similar blind-spot. We did not directly test SODA under simulation conditions with gene flow and population structure; the fact that we simulate under MSC and test under MSC can describe why our errors tend to be of over-splitting nature on simulated data. On the real Protea dataset, species were split (often by geography), showing that SODA can be sensitive to population structure within species. SODA, unlike BPP, avoided breaking humans into multiple species; however, it is hard to know whether this result is due to the presence of only 50 gene trees in this dataset or some level of robustness to population structure. Note that SODA shares its sensitivity to structure and gene flow with methods purely based on MSC. Thus, we suggest that for datasets where high levels of gene flow after speciation is probable, the results of SODA should be used as a guide to enable more time-consuming delimitation using methods that do consider gene flow and are more robust to structure.

In addition to gene flow and population structure, several factors need to be kept in mind when using SODA. Like other methods relying on input gene trees, the accuracy of SODA may depend on the input gene trees [111]. It is helpful that we only rely on unrooted tree topologies, and thus, errors in branch length and rooting do not affect SODA. Plotting accuracy of SODA versus mean gene tree error across simulation conditions, we did not detect a strong impact from gene tree error except for an outlier (Fig. 5.8). Nevertheless, errors in gene tree topologies may bias SODA towards over-splitting because gene tree error tends to increase observed discordance [116, 96]. Moreover, the polytomy test used by SODA makes several assumptions, including the independent treatment of branches. These assumptions could, in theory, further impact the method in the presence of many adjacent short branches. To make sure this is not the case, our simulations used gene trees with high levels of error (Fig. 5.8) and included conditions with many adjacent short branches, and yet showed positive results. Nevertheless, simulations are by nature limited, and thus, further work may reveal other conditions where the method fails.

SODA also relies on a guide species tree. Luckily, given large numbers of genes, the accuracy of species trees tend to be much higher than gene trees, and [123] showed that individuals of the same species often group together in ASTRAL trees. In our simulations, switching to true guide trees resulted in small improvements in accuracy (Fig. 5.6). Nevertheless, if the guide tree includes substantial levels of error, SODA may suffer. For example, on the Protea dataset, several individuals were placed far from their presumed species. Assuming these individuals were correctly identified, we have to conclude the ASTRAL tree had several errors, a problem that SODA is not able to overcome. Finally, SODA requires that the analysis includes at least two individuals from each species, another factor that may limit its application to practice. Due to these caveats, applying SODA has to be done with care.

We were able to compare SODA against one of the most widely-used alternatives, BPP. Previous simulation studies [e.g., 170, 22, 168, 63] and empirical analyses [134, 68, 58] have established BPP as the most accurate and preferred MSC-based delimitation method. We do not expect other Bayesian methods to be substantially more accurate than BPP [22]. And they are not much faster either. For example, STACEY took seven days [66] on the [54] dataset with 19 individuals from 9 shrew species and 500 genes; SODA, on the same data, finished in a matter of seconds and produced identical results (Fig. 5.12). In the case of SpedeSTEM, it requires rooted ultrametric gene trees, which cannot be inferred using the standard models of sequence evolution. Using SpedeSTEM should be combined with rooting and a rate model, which can make the analyses sensitive to errors in those steps; moreover, SpedeSTEM has been less accurate than BPP in previous analyses [22]. In our analyses, SpedeSTEM was the least accurate method. Perhaps the lack of accuracy is due to divergences from a strict molecular clock used in our Simphy simulations, which perhaps the SpedeSTEMv0.9 default pipeline could not overcome. STEM, used in SpedeSTEM, has been shown to have low accuracy in computing the species tree [79], especially given variations in mutational processes [59].

We were not able to use other methods that take gene trees as input. For example, the algorithm of [113] infers species delimitation either using the full gene tree likelihood calculation, which is slow [39] or using the MDC cost [85]. However, this method (Brownie) does not seem to currently have stable software support. Similarly, the method of [172] relies on a species tree and partially labeled gene trees of individuals of several species; however, this method does not have a publicly available implementation. Older methods based on individual loci [e.g., GMYC by 119] were not relevant to our multi-locus datasets. Our attempts to run PHRAPL, which focuses on gene flow, were unsuccessful because the method did not finish after 48 hours of running time.

The advantage of SODA over BPP, in our tests, was two-fold: much better scalability and slightly better recall. BPP cannot handle more than tens of populations, while SODA can easily handle 1000 populations (used in our large simulations). However, overall, BPP was more accurate, especially when allowed to co-estimate the species tree. The relative strengths of the two methods suggest a natural way to combine them. We can first run SODA on the entire (large) dataset to obtain an initial delimitation. The results of SODA can be used to define populations and to divide the dataset into smaller subsets for a more extensive BPP analysis. This divide-and-conquer approach is what many analyses use in practice [e.g., 106] using a manual curation; SODA can help automate that process.

5.6 Proof of claims

Claim 1. *Assuming (i) gene trees $\mathcal{G}$ are generated under the MSC model on an extended species tree $\mathcal{T}$, (ii) the guide tree T (e.g., ASTRAL tree) is a resolution of $\mathcal{T}$, and (iii) the hypothesis testing has no false positive (FP) or false negative (FN) errors, the SODA algorithm returns the correct extended species tree ($\hat{\mathcal{T}}=\mathcal{T}$) and hence the correct delimitation under the MSC model. Additionally, it will correctly delimitate species if all species are sampled more than once.*

The proof is straightforward, and follows from the properties of the extended species tree. Under the MSC model, which is our first assumption, every branch in the guide tree that falls within a single species has zero branch length (because any resolution of the extended species tree is arbitrary). Also, internal branches of the extended species tree have non-zero length (Fig. 5.10). Since we assume the guide tree is a resolution of $\mathcal{T}$, the internal branches that have more than one species in their descendants should have non-zero branch lengths. If the null hypothesis of zero branch length always reject or accept the null hypothesis correctly, then, whenever we fail to reject the null hypothesis, we have a branch within a species (zero length) that needs to be contacted. And where the branch length is determined to be non-zero, the branch corresponds to a speciation event. Thus, with the assumptions of having correct gene trees, guide tree, and the hypothesis test, SODA outputs the correct extended species tree.

If each species has at least two individuals, then, for each species, there exists a bipartition such that one side of the bipartition include that species and only that species. Thus, given the correct extended species tree, and with at least two individuals per species, removing all remaining internal edges produces the correct species delimitation.

Claim 2. *Given a guide tree T that resolves the tree $\mathcal{T}$, SODA incorrectly divides a species S into multiple species (i.e., a false negative error) if and only if the zero length hypothesis testing results in an FP error for one of the branches under the clade defined by S on T.*

We first prove the "if" statement. If the hypothesis test for a branch e results in a false positive error, this means that e had zero branch length, but the test incorrectly rejected the null hypothesis for it. In this case, the algorithm marks e, sister of e and all ancestors of e as "keep" branches and will maintain them in the extended species tree. This is equivalent to assuming that e corresponds to an speciation event or descendants of e form separate species from the descendants of the sister branch since this branch belongs to an internal branch of the estimated extended species tree. A false negative error happens in this case, as leaf descendants of e and leaf descendants of sister of e belong to the same species (since e has zero branch length) and thus SODA has incorrectly divided them into two species.

Now we prove the "only if" statement. Under claim 1, we have proved that SODA returns the correct delimitation when the three conditions hold, so when guide tree T is a resolution of the $\mathcal{T}$, and assuming condition 1 holds, if a species is incorrectly divided into multiple species, then there must have been at least a branch that is marked by the algorithm as "keep", but in fact should get contacted since it is a within species branch. That is only possible when the hypothesis test makes an FP error.

Claim 3. *Given a guide tree T that resolves $\mathcal{T}$, SODA incorrectly combines individuals from two species S_1 and S_2 into one species (a false positive error) under one of these two conditions. 1) S_1 and S_2 each have one sampled individuals and form a cherry. 2) The hypothesis testing has an FN error for all branches of $\mathcal{T}$ below the LCA of S_1 and S_2. This condition requires FN errors for two or more branches if neither species is a singleton.*

We start with sufficiency statement. When a cherry of S_1 and S_2 each have one sampled individuals, the only branches separating S_1 and S_2 are terminal branches and we are not able to compute p-values of the null hypothesis test for terminal branches. Hence, SODA has no option but to merge the two (they won't get marked in any steps of the algorithm). For the latter case, let e_1 be the branch that S_1 individuals are pendant from and e_2 be the sister branch separating the subtree that S_2 belongs to. If the hypothesis test for a branch e_i results in a false negative error, this means that e_i had non-zero branch length but the test failed to reject the null hypothesis for it. In this case, the algorithm does not mark e and moves to the sister of e_i (e1 or e2 in this instance). If the same condition holds for the sister branch and all branches of T below the $LCA(S_1,S_2)$, then the whole subtree will not be marked and eventually the algorithm will contract those branches. This is equivalent to combining S_1 and S_2 since the only branches that will remain in the estimated extended species tree from this subtree are terminal branches forming a polytomy. A false positive error happens in this case, as S_1 and S_2 belong to the separate species and SODA has incorrectly merges them into one species. addtoresetclaimproofcount

We next prove with necessity statement. Under claim 1, we have proved that SODA returns the correct delimitation when the three conditions hold, so when guide tree T is a resolution of the $\mathcal{T}$, and assuming condition 1 holds, the error could be either from a cherry of two species with one sample or an error in hypothesis test. When two or more species are merged together, this means some branches should have been marked “keep” by the algorithm but they incorrectly have been contracted. This could be the case where the test is unable to generate a p-value for the null hypothesis (terminal branches) which is only problematic when species are sampled just once, or it could be FN in the hypothesis test for the branches above those species that has resulted in the contraction of them.

Table 5.4: Average running time of SODA (seconds) on Large dataset in different model conditions. These numbers do not include the time for inferring species tree and gene trees

genes	0.5M	1M	2M
100	68.7	74.6	78.1
200	163.4	185.2	194.2
500	723.5	856.4	842.5
1000	1715.1	2071.9	1849.9

5.7 Acknowledgement

Chapter 5, in full, is a reprint of the material as it appears in “SODA: Multi-Locus Species Delimitation Using Quartet Frequencies” (2021). Rabiee, Maryam; Mirarab, Siavash, Bioinformatics, 36(24), 5623-5631. The dissertation author was the primary investigator and first author of this paper.

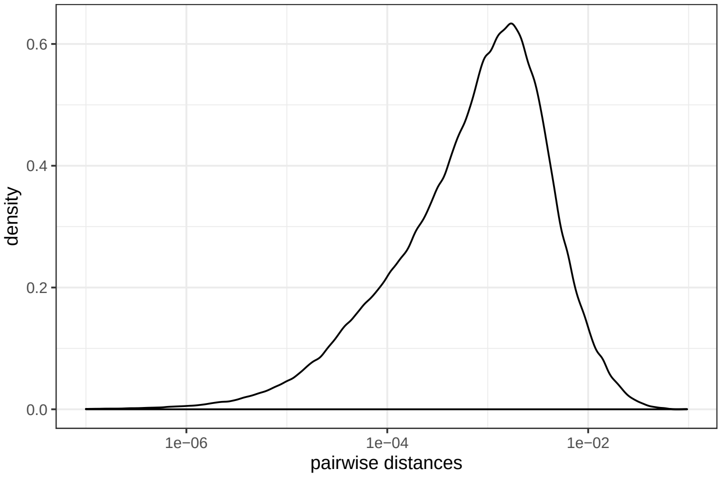


Figure 5.4: Density plot of the pairwise distances of individuals within species in small simulated dataset computed from true gene trees in substitution units.

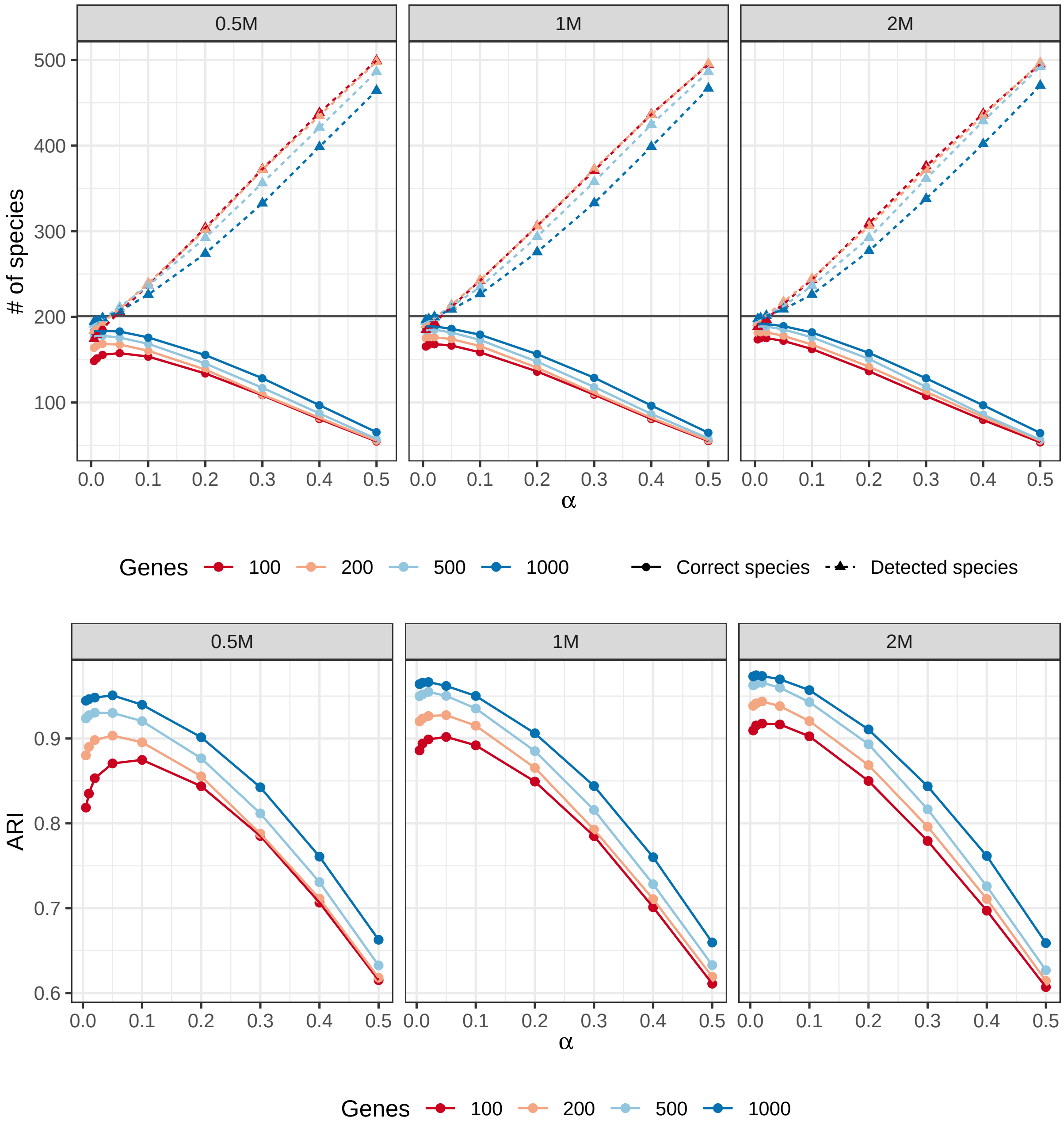


Figure 5.5: **Accuracy of SODA on the large dataset.** a) We show the number of species that are completely correctly delimited (solid lines) and the total number of species found by SODA (dashed lines). Results divided into three model conditions with very high ILS (0.5M), high ILS (1M) and moderate ILS (2M) as we change α (x-axis) and the number of genes (colors). b) ARI (y-axis) shows the accuracy of SODA on the three model conditions with very high ILS (0.5M), high ILS (1M) and moderate ILS (2M) as we change α (x-axis) and the number of genes (colors).

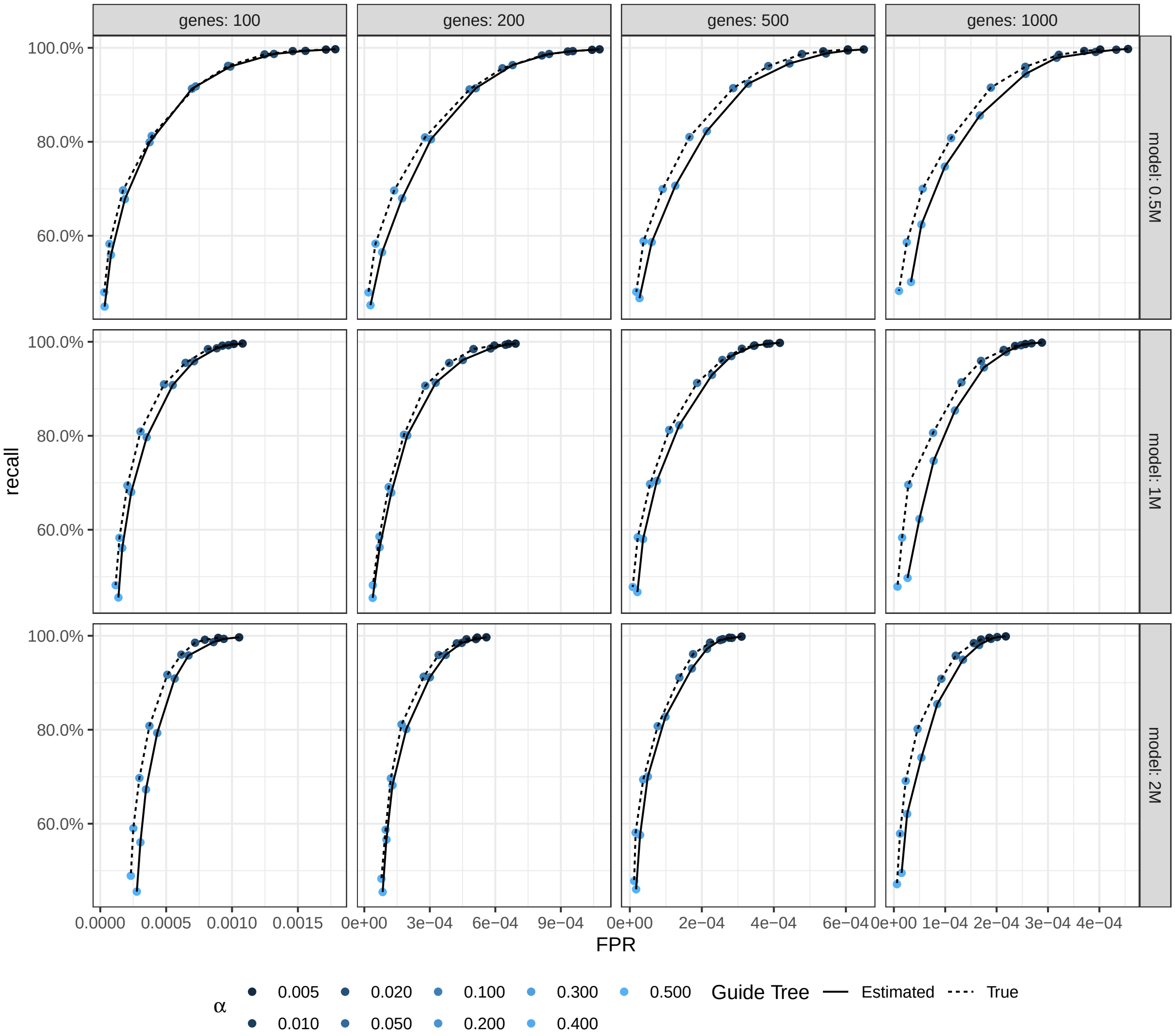


Figure 5.6: **Accuracy of SODA on the large dataset.** ROC showing recall versus False Positive Rate (FPR) for all model conditions and different choices of α (dot size). The dotted line shows the results for true guide tree and the solid line is for estimated.

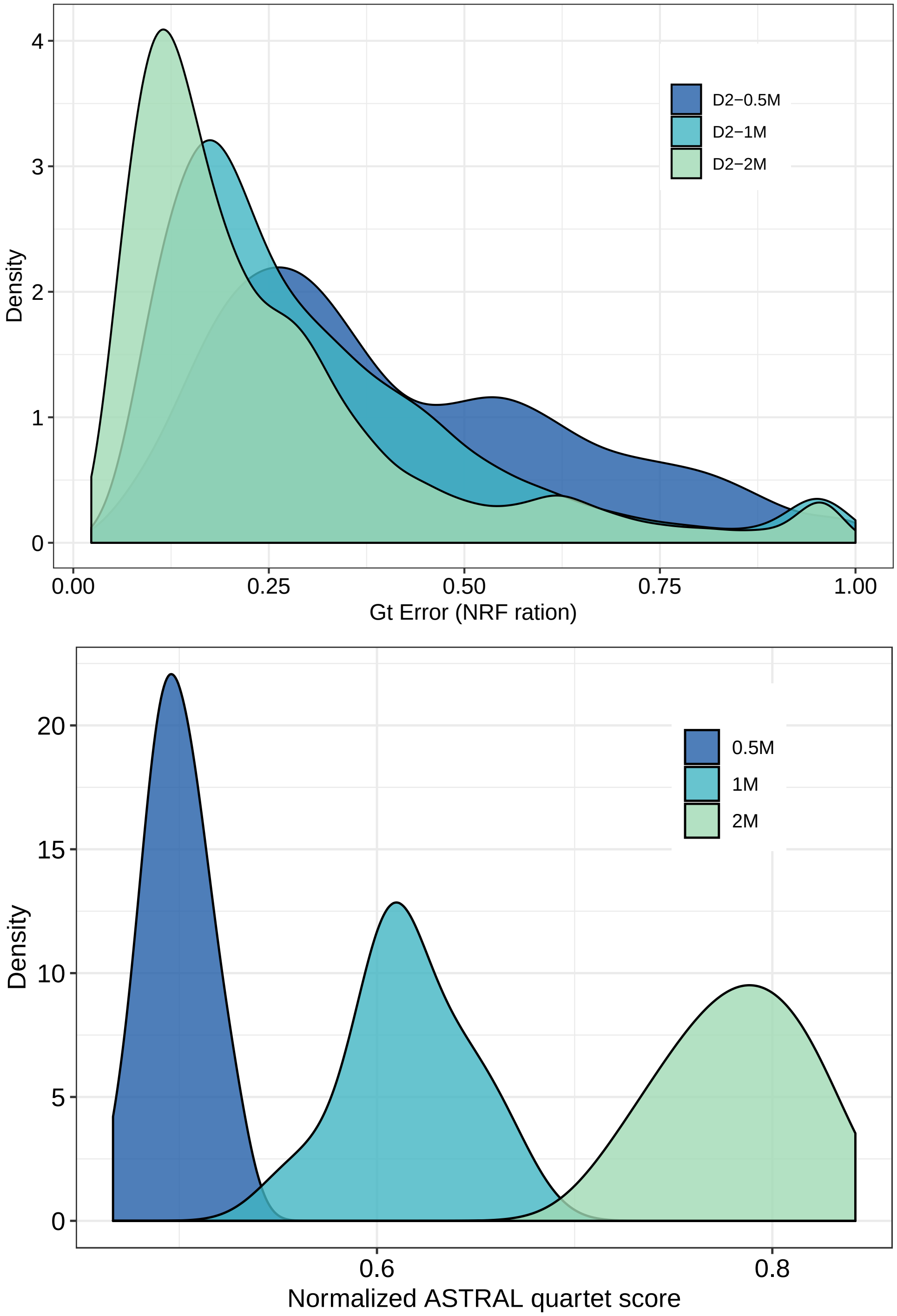


Figure 5.7: Properties of the large simulation dataset. We show gene tree error (top) and ILS as measured by quartet score of the species tree versus true gene trees (bottom).

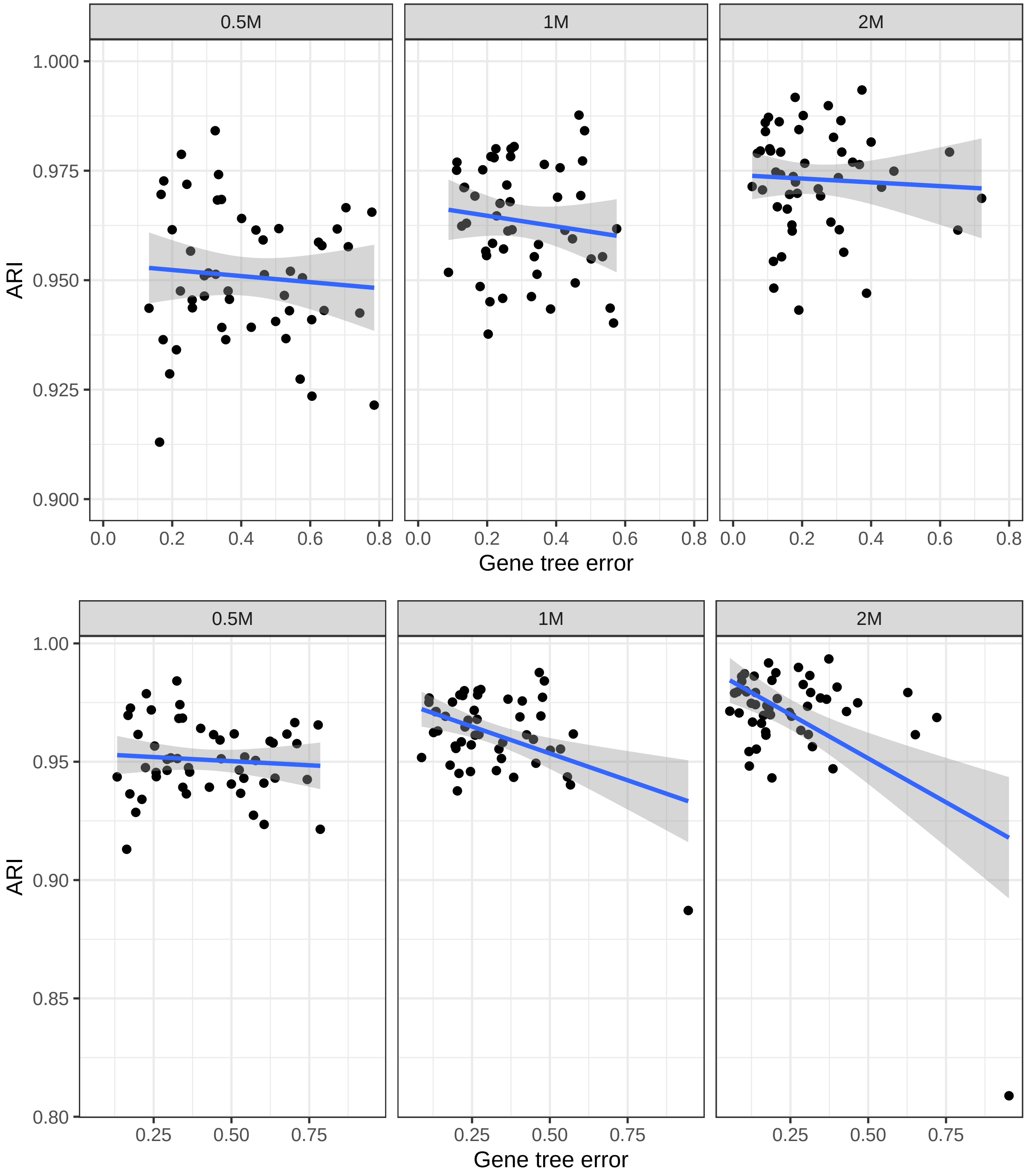


Figure 5.8: Correlation between gene tree error and accuracy of SODA delimitation. Bottom: all replicates. Top: one outlier replicate is removed from 1M and 2M conditions.

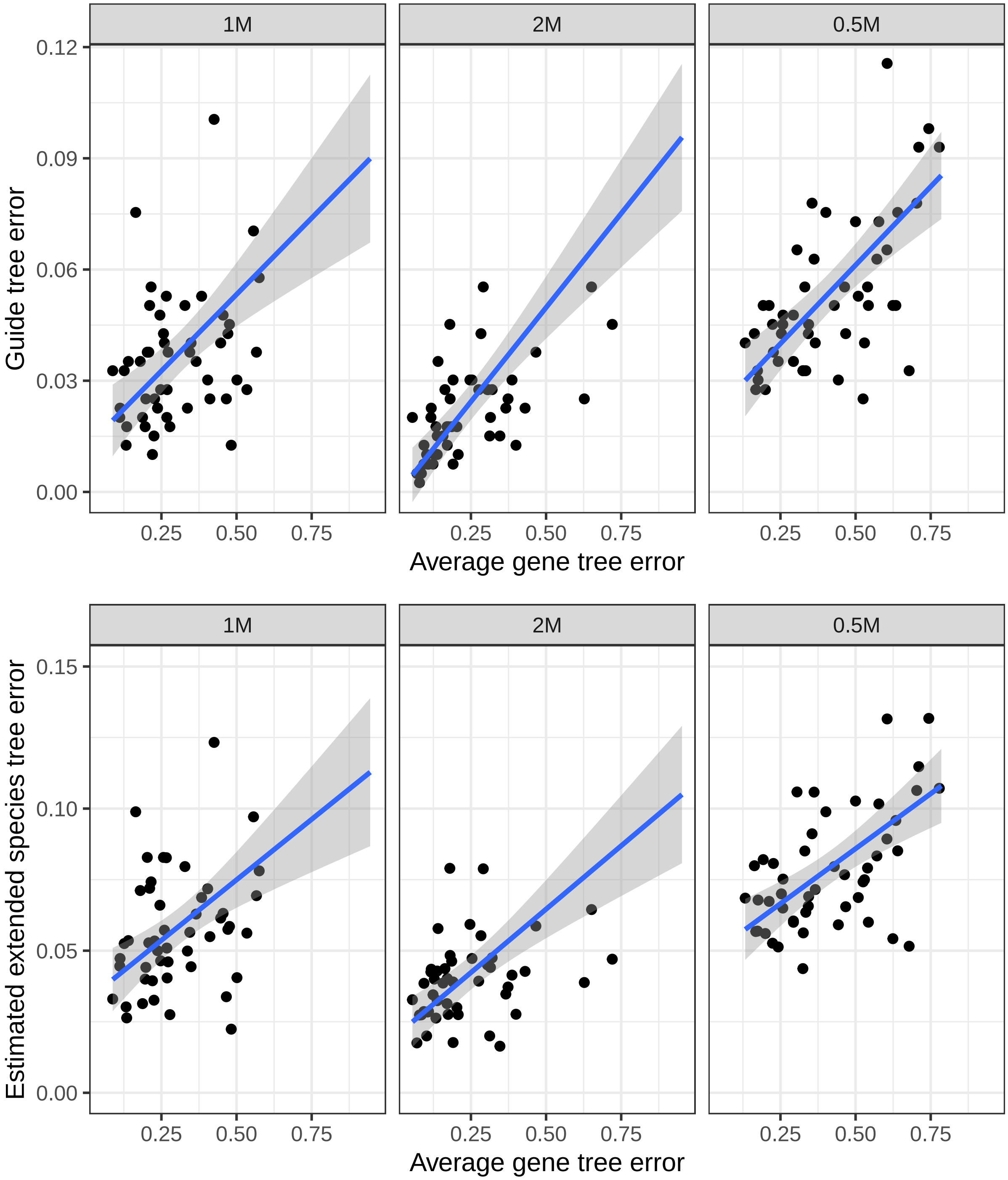


Figure 5.9: Correlation between gene tree error and accuracy of the estimated guide tree and extended species tree. The y-axis has been limited to (0,0.12) and (0,0.15), so some outliers have been removed.

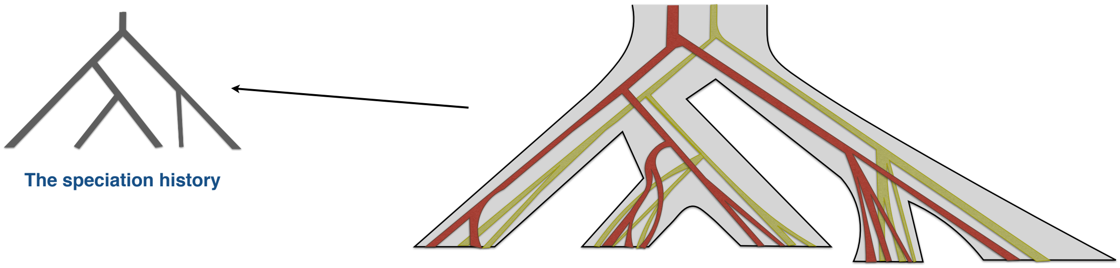


Figure 5.10: The species tree and gene trees with individuals as tips of the tree on the right. The tree on the left shows the true speciation history of the species that the individuals as leaves of the gene trees belong to.

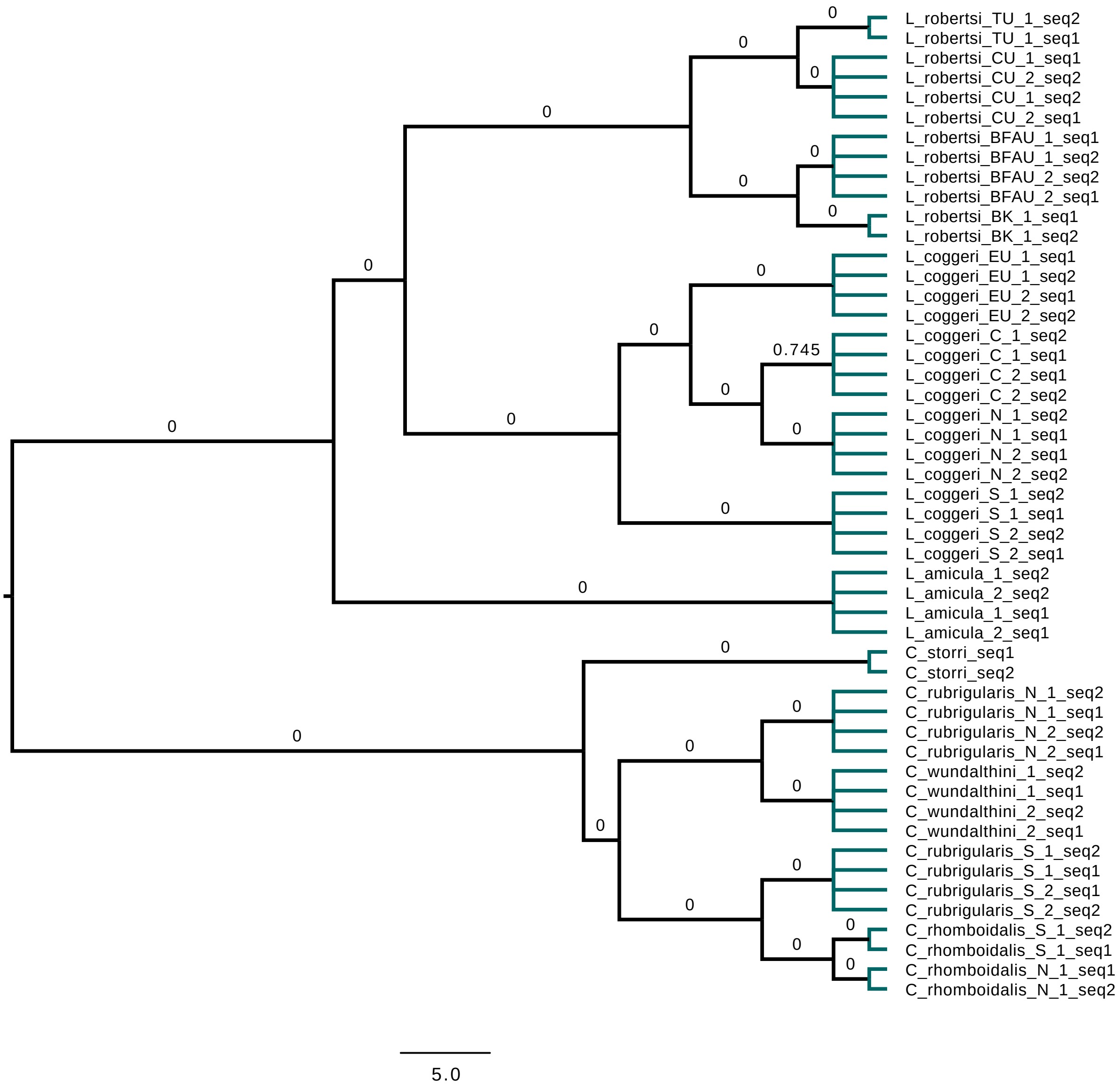


Figure 5.11: The species tree inferred from the gene trees using multi-individual version of astral given the pre-known mapping of the individuals to species as described in Singhal et al. (2018). The values on the branches shows the p-values generated by the polytomy test.

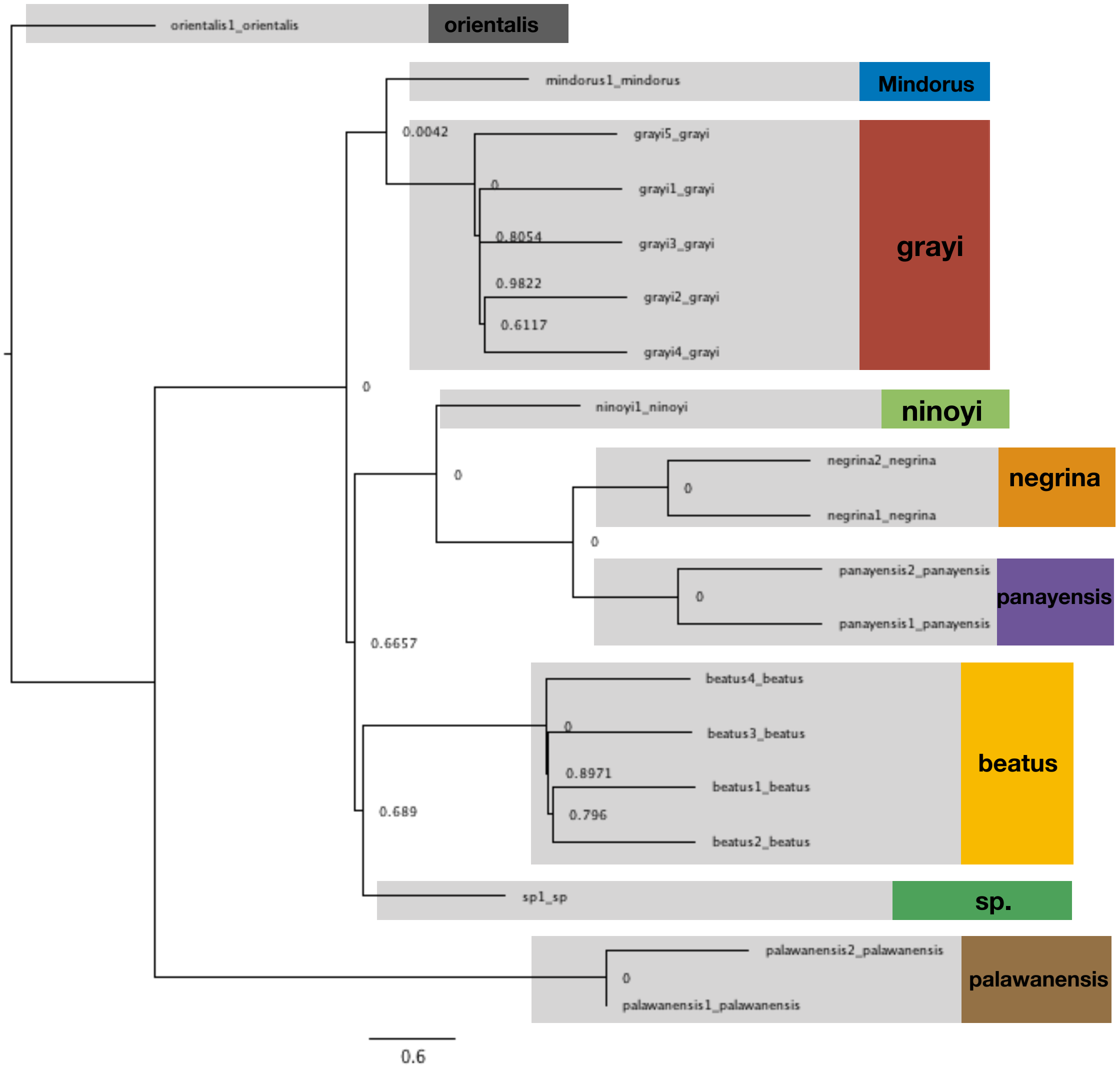


Figure 5.12: The species tree inferred from the simulated gene trees, Sim-Matching from (Giarla and Esselstyn, 2015) , with SODA p-values put on the nodes (each number representing the p-value of the branch above that node) and the delimitation results shown as color matching the original mapping of the individuals to species.

Chapter 6

QuCo: Quartet-based Co-estimation of Species Trees and Gene Trees

Phylogenomic analyses consistently face a dilemma: on the one hand, most accurate species and gene tree estimation methods are those that co-estimate them; on the other hand, these co-estimation methods do not scale to moderately large numbers of species. The summary-based methods, which first infer gene trees independently and then combine them, are much more scalable but are prone to gene tree estimation error, which is inevitable when inferring trees from limited-length data. Importantly, gene tree estimation error is not just random noise and can create biases, for example, through long branch attraction. Co-estimating gene trees and the species tree together is known to reduce the gene tree error. In this paper, we develop a likelihood-based approach to co-estimation that can scale. The method, called Quartet Coestimation (QuCo), takes as input independently inferred distributions over gene trees and computes the most likely species tree topology and internal branch length for each quartet, summed over all possibilities for each gene tree. In doing so, it updates the gene tree posterior probabilities based on the species tree (hence, the term co-estimation). By relying on quartets and focusing on gene tree topologies, not branch length, the method is able to provide fast calculations of the maximum likelihood gene trees and species tree. We benchmark our method with extensive simulations for quartet trees in zones known to produce biased species trees and further with larger trees. We also run QuCo on a biological dataset of bees. Our results show increased accuracy compared to the summary-based approach ASTRAL run on estimated gene trees.

6.1 Introduction

Species tree estimation from multi-locus genome-wide datasets while accounting for gene tree discordance is now routinely attempted. There has been much effort to develop methods that can infer species trees and gene trees accurately, and in particular, methods focused on handling incomplete lineage sorting (ILS) as modeled by multi-species coalescent (MSC) [125] abound [e.g., 83, 81, 99, 156, 27, 109]. Yet, a major challenge remains. The most accurate methods for species tree estimation have been Bayesian methods that co-estimate gene trees and species trees, while the most scalable methods have been summary methods that first estimate gene trees independently and then combine them to infer a species tree [97]. This dichotomy has left practitioners with a choice between using methods that can handle large data or more accurate methods that will have to be run on subsets of the data [143]. Neither solution is ideal.

Theoretical expectations and empirical evidence suggest that inferring all gene trees together should be more accurate than the two-step approach that independently computes gene trees first [79, 151, 34, 11, 110]. Gene tree estimation from limited-length locus data is error-prone, and gene tree estimation error impacts species trees [59, 116, 96, 75, 104]. Importantly, gene tree estimation error is not just random noise and can create biases. In particular, Roch et al. [130] proved that under challenging cases, long-branch attraction in gene trees could lead to long-branch attraction in species trees and statistically inconsistent estimates. Dealing with gene tree estimation error has motivated several remedies, including binning genes [96, 11], collapsing low support branches [169, 114], and sampling posterior distributions of gene trees [14]. However, these methods have their own drawbacks. Binning has the possibility of combining discordant genes, collapsing low support requires selecting a threshold, and simply combining samples from gene tree posteriors as input to summary methods shows mixed results in simulations [95]. Thus, co-estimation methods are still the ideal option. Yet, current co-estimation methods such as *BEAST [109] scale only to tens of species [91].

The scalability of co-estimation methods has remained limited because they address a fundamentally challenging problem using a the fundamentally slow Bayesian MCMC process. Developing theoretically justified scalable co-estimation methods requires simplifying the model or the partial use of heuristic methods. Wang and Nakhleh [159] attempted to speed up co-estimation by avoiding full sampling of the entire space using an EM-like algorithm that iteratively switches between species tree and gene tree estimation. However, while this iterative approach certainly seems to increase gene tree accuracy, it is not clear that it improves species tree accuracy, possibly due to its reliance on estimated species trees to improve gene trees.

We introduce an MSC-based likelihood-based approach to co-estimation designed to scale. The scalability is obtained using simplifying assumptions and heuristics while keeping a likelihood-based core. Our first insight is that while joint sampling of *continuous* parameters of gene trees, such as their branch lengths, slows down co-estimation, these are often nuisance parameters. Therefore, we focus on topology, marginalizing over gene tree branch lengths and other continuous parameters. However, this marginalization would still be intractable if done jointly. Instead, we make assumptions about substitution unit branch lengths, amounting to a type of no-common-mechanism model where continuous parameters across gene trees are fully unlinked. This admittedly strong assumption enables us to decouple genes and do the marginalization on a per gene basis.

Thus, we first estimate gene tree distributions independently using existing methods and then combine the distributions. This approach can still be called co-estimation because we combine results from multiple genes and *adjust* their distribution in a way that corresponds to computing them jointly. This insight is not new for species tree inference [where 4, 76, pioneered the idea in a method called BUCKy] or improving gene trees [150]. Finally, building on the success of quartet-based methods for handling ILS [e.g., 99, 27], we estimate quartet species trees using a likelihood-based approach but combine the quartet species trees heuristically using supertree methods. Lest the reader worries about impacts of lowered taxon sampling and increased long branch attraction when using quartets, we note that gene tree estimation is performed on the full set of taxa, but the amalgamation step focuses on *induced* quartets.

We introduce a method called Quartet-based co-estimation (QuCo for short) that takes as input a Bayesian posterior tree distribution per each of k genes, infers the distribution of quartet trees in that input, and summarizes the posteriors in a $3 \times k$ table per quartet. Next, for each quartet of species, it computes the maximum likelihood species tree topology and its single internal branch length in the coalescent unit, marginalizing over gene trees. It then improves the gene tree topologies using the species tree. Finally, it combines the inferred quartet species trees to obtain a final tree topology on the complete set of taxa. We evaluate the method on a set of simulations with four to 101 species and a real bee dataset and show that it increases accuracy while providing a path for scalable co-estimation.

6.2 The QuCo algorithm

6.2.1 QuCo: Maximum likelihood quartet species trees

We start with our theoretical results underpinning QuCo for inferring maximum likelihood quartet species trees. Throughout, we assume gene trees are inferred from single-copy orthologous gene trees that differ due to ILS only, as modeled by MSC. Given the posterior distributions of the gene trees, QuCo computes the maximum likelihood species tree for each quartet of species. Then, using the species tree topology and its branch length, it updates the gene tree posterior distributions. While our method for analyzing each quartet is based on likelihood calculations, to extend to more than four species, we rely on the heuristic method of examining all or a subset of quartets, a procedure we introduce at the end.

Marginalized Likelihood of the Quartet Species Trees

For a quartet of species $\{A,B,C,D\}$, we denote the three topologies $AB|CD$, $AD|BC$, $AC|BD$ by $j \in \{1,2,3\}$. Let $\mathcal{S} = \{\mathcal{S}_1, \cdots, \mathcal{S}_k\}$ be the set of sequences obtained for each available gene. Given $\mathcal{S}$, we seek to compute the likelihood of the species tree, parameterized by $\theta = (t,d)$ where $t \in \{1,2,3\}$ is the topology and d is the internal branch length in coalescent unit. This parameterization fully identifies the distribution of unrooted gene tree *topologies* [3]. The species tree likelihood is marginalized over all possibilities for the k gene trees, and we show the log-likelihood function as $l(\theta;\mathcal{S}) = \log\big(P(\mathcal{S};\theta)\big)$. Let $\mathbb{C} = \{1,2,3\}^k$. Any set of k quartet tree topologies, one per gene, can be indexed by a tuple $c = (c_1, \ldots, c_k) \in \mathbb{C}$. Let the true gene tree *topologies* be represented by $\mathcal{G}^* = (\mathcal{G}_1^*, \ldots, \mathcal{G}_k^*)$. Then:

$$\begin{aligned} P(\mathcal{S};\theta) = \sum_{c \in \mathbb{C}} P(\mathcal{S}|\mathcal{G}^* = c;\theta).P(\mathcal{G}^* = c;\theta) = \\ \sum_{c \in \mathbb{C}} \overbrace{P(\mathcal{S}|\mathcal{G}^* = c;\theta)}^{\text{sequence likelihood}} . \prod_{i=1}^{k} \overbrace{P(\mathcal{G}_i^* = c_i;\theta)}^{\text{gene tree likelihood}} \end{aligned} \tag{6.1}$$

where the last equation uses the conditional independence of gene trees for a fixed species tree. Working on quartet gene tree topologies makes the calculation of gene tree likelihood trivial. Under the MSC model [115, 3], for any $j \in \{1,2,3\}$

$$P(\mathcal{G}_i^* = j;\theta = (t,d)) = \begin{cases} 1 - 2/3e^{-d} & \text{if } j = t \\ 1/3e^{-d} & \text{o.w} \end{cases} . \tag{6.2}$$

However, working on gene tree topologies (c) makes sequence likelihood calculation challenging because we cannot readily write it as a product over genes. To do so, we need all continuous parameters, including gene tree branch lengths in substitution units and the rate matrix, which we jointly specify using r_i for each gene and $r = (r_1, \ldots, r_k)$. Then,

$$
\begin{aligned}
P(\mathcal{S}|\mathcal{G}^* = c;\theta) = \int_r f(r;\theta) P(\mathcal{S}|r, \mathcal{G}^* = c;\theta)\,\mathrm{d}r = \\
\int_r \prod_{i=1}^{k} f(r_i;\theta) P(\mathcal{S}_i|r_i, \mathcal{G}_i^* = c_i)\,\mathrm{d}r = \\
\int_r \prod_{i=1}^{k} f(r_i;\theta) \frac{P(r_i, \mathcal{G}_i^* = c_i|\mathcal{S}_i) P(\mathcal{S}_i)}{f(\mathcal{G}_i^* = c_i, r_i)}\,\mathrm{d}r
\end{aligned}
\tag{6.3}
$$

where the second equation uses the fact that given all gene tree parameters, gene sequence data are independent of each other and the species tree, and given the species tree, gene trees (thus r_i) are independent.

Assumptions. Even for a quartet, computing (6.3) is not easy. To move forward, we make two assumptions regarding branch lengths. *i*) We assume $f(\mathcal{G}_i^* = c_i, r_i) = \frac{1}{3} f(r_i)$, which is reasonable by symmetry when the species tree is *not* given. It requires assuming that *a priori* all three unrooted gene tree topologies are equiprobable, sequence evolution parameters are independent from gene tree topology, and substitution unit branch lengths are independent from *unrooted* gene tree topologies. *ii*) We assume $f(r_i;\theta) = f(r_i)$. The species tree clearly impacts the distribution of coalescent unit gene tree branch lengths. Typical ways of mapping branch lengths to substitution units assume distributions over population size and mutation rates. These two parameters are ideally drawn per branch, or else gene trees will be ultrametric. When drawn per branch, substitution unit branch lengths are still dependent on the species tree, though the dependence reduces as the variation of rates across branches increase. We assume an extreme case where the mutation rate branch lengths are drawn from distributions independent from the species tree parameter θ. In other words, each branch of the gene tree is assigned a substitution unit length that is independent of the coalescent units length of internal branch (d). We also assume that other continuous parameters (e.g., rate matrices) are either constant across the tree or drawn from distributions independent from θ. These assumptions are not entirely realistic but have the advantage of allowing arbitrary and unlimited deviations from the clock, eliminating the need to assume any clock models. Also, they make (6.3) tractable. Let **P** be the $3 \times k$ matrix where $\mathbf{P}_{j,i} = P(\mathcal{G}_i^* = j|\mathcal{S}_i)$. Then:

$$\begin{aligned}
P(\mathcal{S}|\mathcal{G}^* = c;\theta) = &\int_r \prod_{i=1}^{k} f(r_i) \frac{P(r_i, \mathcal{G}_i^* = c_i|\mathcal{S}_i)P(\mathcal{S}_i)}{\frac{1}{3}f(r_i)} \,\mathrm{d}r = \\
&A \int_r \prod_{1=1}^{k} P(r_i, \mathcal{G}_i^* = c_i|\mathcal{S}_i) \,\mathrm{d}r = \\
&A \prod_{i=1}^{k} \int_{r_i} P(r_i, \mathcal{G}_i^* = c_i|\mathcal{S}_i) \,\mathrm{d}r_i = \\
&A \prod_{i=1}^{k} P(\mathcal{G}_i^* = c_i|\mathcal{S}_i) = A \prod_{i=1}^{k} \mathbf{P}_{c_i,i}
\end{aligned}$$

where $A = \prod_1^k 3P(\mathcal{S}_i)$, and integral and product swap in the third line is possible because no term has two elements of r. Replacing RHS in (6.1):

$$P(\mathcal{S};\theta) = \sum_{c \in \mathbb{C}} A \prod_{i=1}^{k} \mathbf{P}_{c_i,i} P(\mathcal{G}_i^* = c_i;\theta) = A \prod_{i=1}^{k} \sum_{j=1}^{3} \mathbf{P}_{j,i}.P(\mathcal{G}_i^* = j;\theta) \tag{6.4}$$

where the second equation uses the fact that for any $3 \times k$ matrix $(x)_{j,i}$, we have $\sum_{c \in \mathbb{C}} \prod_{i=1}^{k} x_{c_i,i} = \prod_{i=1}^{k} \sum_{j=1}^{3} x_{j,i}$ (easy to confirm).

To compute matrix **P** (posterior gene tree topology probabilities marginalized over branch lengths and substitution parameters), we take advantage of Bayesian MCMC sampling implemented in standard methods such as MrBayes [133]. Thus, the input to QuCo is a set of k gene tree posterior distributions, each inferred separately on its full set of taxa without a species tree. The fraction of times any tree topology appears in the MCMC chain (after some burnout period) is a valid approximation of its posterior probability, marginalized over branch length and other continuous parameters, giving us all values of **P**. We can also approximate **P** using normalized quartet log-likelihood as implemented in IQ-TREE (`-wql`) [93]. Either way, recalling (6.2), note that

$$\sum_{j=1}^{3} \mathbf{P}_{j,i}.P(\mathcal{G}_i^* = j;\theta) = (1 - 2/3e^{-d})\mathbf{P}_{t,i} + 1/3e^{-d}(1 - \mathbf{P}_{t,i}) = \quad \mathbf{P}_{t,i} + e^{-d}(1/3 - \mathbf{P}_{t,i})$$

which, when replaced in (6.4), gives us the log-likelihood function:

$$l(\theta=(t,d);\mathbf{P}) = \log(A) + \log\Big(\prod_{i=1}^{k}\big(\mathbf{P}_{t,i}+e^{-d}(1/3-\mathbf{P}_{t,i})\big)\Big) = \\ A' + \sum_{i=1}^{k}\log\big(\mathbf{P}_{t,i}+e^{-d}(1/3-\mathbf{P}_{t,i})\big) \tag{6.5}$$

where A' is a constant term that is the same for all θ and can be ignored.

For each $t \in \{1,2,3\}$, we compute $l'(t) = \arg\max_d l((t,d);\mathbf{P})$ numerically; then, we simply select the topology t with the maximum $l'(t)$ value as the species tree. Maximizing the $l(\theta;\mathbf{P})$ function numerically is easy because it is twice differentiable and while it is not a convex function of d (the sign of its second derivative changes with different input parameters), we can prove (see Appendix 6.5):

Proposition 1. For a fixed $\tilde{\mathbf{t}}$, the $l((\tilde{\mathbf{t}},d);\mathbf{P})$ function (6.5) can have only one maximizer for $0 < d < \infty$.

Thus, we can seek the global maximum of $l((t,d);\mathbf{P})$ for each $t \in \{1,2,3\}$ by simple numerical search using any modern optimizer package. We use `scipy.optimize` package with the constraint $d > 0$ imposed using the trust-region constrained algorithm [31]. To help faster convergence, we provide the first and second derivatives of $l(.)$ to the optimizer, as shown in Appendix 6.5.1. Finally, we add a small pseudo-count of 10^{-8} to every element of $\mathbf{P}$ and normalize it appropriately.

Gene tree updates

Once a species tree $\theta = (t,d)$ is inferred, QuCo updates the gene tree posterior distribution to

$$P(\mathcal{G}_i^* = j|\mathcal{S}_i;\theta) = \frac{\mathbf{P}_{j,i}.P(\mathcal{G}_i^* = j;\theta)}{\sum_{a=1}^{3}\mathbf{P}_{a,i}.P(\mathcal{G}_i^* = a;\theta)} \tag{6.6}$$

where $P(\mathcal{G}_i^* = a;\theta)$ is computed using (6.2). This update is what makes the method a co-estimation. Note that this approach is not an iterative method switching between updating gene tree topologies and re-estimating species trees; if attempted, the gene tree updates can only increase the probability of the selected species tree compared to the alternative topologies, and will not lead to a change in the next iteration.

More than four species

To move beyond four species, QuCo uses a heuristic supertree approach that ignores the dependency between quartets and analyzes them independently. We first select a set of quartets such that the resolution of all these quartets (perhaps in addition to auxiliary information such as a guide tree) is sufficient to infer the species tree. The simplest choice is to select all $\binom{n}{4}$ quartets but we describe an alternative below. Once the set of quartets is selected, QuCo induces all trees in the MCMC samples of all k gene trees down to each selected quartet to compute the quartet posterior probabilities. Thus, for each quartet, a $3 \times k$ matrix will be obtained. Note that this step, while conceptually simple, needs to process a very large number of trees and thus needs to be implemented with care to obtain high efficiency. Next, for each quartet, we infer the maximum likelihood species tree as described earlier, obtaining a set of quartet species trees. The last step is to combine all the quartet species trees into a full tree using a quartet amalgamation method. While any such method can be used for this step, we will use ASTRAL (as a supertree method, not as a gene summary method) and will show that using wMaxCut [6] generates very similar results.

Sampling Quartets. For sufficiently small datasets (e.g., less than 50 species), we afford to examine all the quartets. For larger input, we use a two-step approach. We first run ASTRAL on input gene trees, defined for each gene as the majority-rule consensus (MRC) of the trees in the input distribution for that gene. Next, we contract all the branches in the ASTRAL tree with local-pp support [136] less than a threshold (default: 1.0). We then use an algorithm to sample quartets around polytomies of the resulting multifurcating guide tree, and this strategy focuses the quartet sampling on difficult parts of the tree.

The sampling works as follows. For a polytomy of degree d, we sample a single species from each side of the polytomy (or a uniformly sampled subsample of 12 sides when $d > 12$), and sample all $\binom{d}{4}$ quartets from that sample. To decide which species to choose from each side, we use a probabilistic method. The probability of sampling a leaf is set to $\frac{1}{2^p}$ where p is the number of nodes between the polytomy and the leaf. The closer the leaf is to the polytomy, the higher the chance we sample it. Trivially, the probabilities of all leaves on each side of a polytomy sum up to 1. We repeat the sampling procedure many times, and by default, reduce the rounds proportionally to the degree d (default number of rounds: $^{1200}/_{d}$). Note that since each round generates $\binom{d}{4}$ quartets for $d \leq 12$, we perform fewer rounds for larger d. In the end, in addition to the QuCo-resolved species tree quartets, we give the multifurcating guide tree to the subsequent supertree method (e.g., ASTRAL) as input. Thus, in effect, we use QuCo to resolve polytomies of the input guide tree.

6.2.2 Datasets

Felsenstein's zone.

Long branch attraction (LBA) is among the most challenging sources of systematic bias in phylogenomics [16, 65], and [130] have shown that both summary methods and concatenation are inconsistent under conditions that induce LBA. Thus, we perform simulation studies close to the Felsenstein zone [49] to assess the resiliency of our method to LBA. To do so, we designed a way of simulating gene trees that tend to be in Felsenstein's zone. First, gene trees in coalescent units are generated according to MSC on a fixed balanced species quartet tree (Fig. 6.1a). Each branch of the species tree has one of two mutation rates μ_s and μ_l assigned to it. Each gene tree branch length is multiplied by the rates of corresponding species branches (a gene tree branch may cover one to three species tree branches) to obtain their length in substitution units. We set μ_s and μ_l so that two non-sister terminal branches (B and D) and the internal branch in the unrooted gene trees share a short expected length s and the other two terminal branch lengths have expected length l. Setting μ_s and μ_l properly requires a lemma (proved in Appendix 6.5.2):

Lemma 1. *Under MSC, for a balanced quartet species tree with internal branch lengths $\frac{d}{2}$ (Fig. 6.1a), the expected length of terminal branch lengths in unrooted gene trees* above *the speciation nodes is $\tau_2 = 1 - \frac{1}{3}e^{-d}$.*

Let τ_1 be the fixed coalescent unit terminal branch lengths for all species, and let μ_l and μ_s be mutation rates assigned to the tree as shown in Figure 6.1. The expected substitution unit length of terminal branches of A and C (l) and terminal branches of B and D (s) are: $l = \mu_l \tau_1 + \mu_s \tau_2$ and $s = \mu_s(\tau_1 + \tau_2)$. Thus, we assign $\mu_s = \frac{s}{\tau_1+\tau_2}$ and $\mu_l = \frac{l - \tau_2 s/(\tau_1+\tau_2)}{\tau_1}$ so that the expected branch lengths are as desired. Finally, note that the expected length of the internal unrooted gene tree branch is 1 in coalescent units and μ_s in substitution units. To force the expected internal branch length in substitution units to be also s (as in Felsenstein's zone), we need to set set $\tau_1 = 1 - \tau_2 = \frac{e^{-d}}{3}$.

With this setting, each simulation is parameterized by the coalescent unit internal branch d (controlling amount of ILS) and expected length of long and short terminal branches, l, s, respectively. LBA is expected for high l/s. We used this simulator to create very hard conditions meant to break methods. We vary l, s, and d in 48 combinations, each with 20 replicate runs. We set $d \in \{0.1, 0.2, 0.3\}$, which corresponds to 40%, 45%, and 51% of gene trees matching the species tree. We use the Dendropy package [147] to simulate 500 true gene trees under neutral coalescent model conditioned on a species tree shown in Figure 6.1. For each d, we consider 16 combinations of short and long branch lengths: $s \in \{0.01, 0.02, 0.04, 0.08\}$ and $l \in \{0.1, 0.2, 0.3, 0.4\}$ and convert gene tree branch lengths to substitution units, as described earlier. Then, we use INDELible [52] to simulate sequences down these trees, setting the sequence length to 200, 400, 800, and 1600 bps. Thus, in total, we have $48 \times 4 = 192$ model conditions, 3840 replicates, and 1,920,000 gene trees. We infer gene trees using MrBayes and ensure convergence by checking the average standard deviations of split frequencies, which is less than

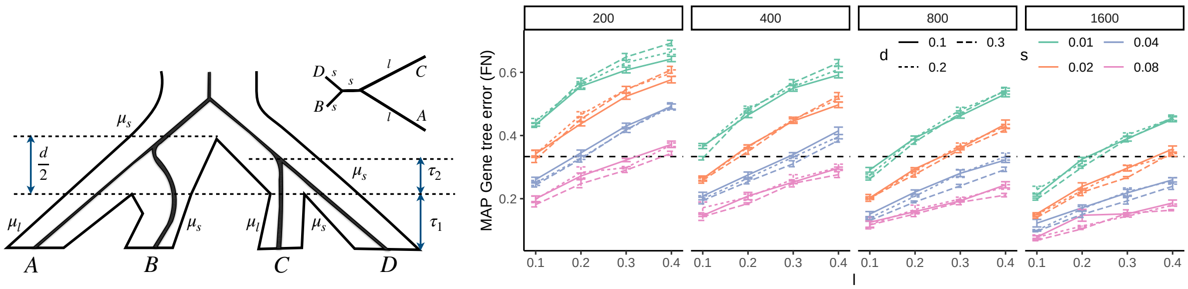


Figure 6.1: Felsenstein's zone simulation. Left: Each gene tree branch length is scaled by μ_s and/or μ_r; e.g., the length of the terminal branch of C becomes $\mu_l\tau_1 + \mu_s\tau_2$. Rates μ_s and μ_r are selected such that terminal branches of A and C in the unrooted gene tree have expected branch length l, and other branches have expected length s. Right: MAP gene trees estimated using MrBayes with simulations in Felsenstein's zone can have large estimation error, especially when $^l/_s$ is high and sequence lengths are short. A transformed gene tree is shown on top right corner of the species tree.

0.08 for all runs with 99 percentile equal to 0.025.

Gene trees estimated using MrBayes from alignments generated on the true gene trees can have high rates of error, depending on $^l/_s$ and sequence length (Fig. 6.1b). Note that a random selection of tree topology will still be correct ⅓ of times; thus, the MAP gene trees have more error (due to LBA bias) than randomly estimated trees in some conditions. Moreover, incorrect gene trees are not randomly distributed but are heavily biased towards putting long terminals (A and C) together. Thus, conditions with gene tree error above ⅓ are particularly difficult.

30-taxon datasets. We reuse a dataset simulated by [87] using Simphy [89] with three model conditions, and 500 genes, each with 50 replicates (sampled out of 100 original replicates). The three conditions are differentiated by their level of deviations from the molecular clock, as controlled by α, which is the inverse of the variance of the rate multipliers applied to gene tree branch lengths. Because of difficulties in running MrBayes to convergence for all of the $3 \times 500 \times 50 = 75000$ gene trees, we use IQ-TREE instead. We use IQ-TREE `-wql` option to compute the log-likelihood for all quartet topologies, which we then normalize and exponentiate to approximate posteriors and use as input to QuCo. See appendix E.1 for exact commands. We run QuCo on all $\binom{30}{4} = 27405$ quartets and combine these quartet trees using ASTRAL or wMaxCut.

101-taxon datasets. We use one model condition of a dataset by [169] with 101 taxa, 400 bp sequences, 200 genes, and 30 replicates sampled out of a total of 50 replicates, each with a distinct species tree. The species trees are simulated under the birth-only process with the birth rate 10^{-7}, fixed haploid N_e of $400K$, and the number of generations sampled from a log-normal distribution with the mean $2.5M$. The average normalized RF distance between true species trees and true gene trees was in most replicates in the [0.3, 0.6] range, with an average of 0.46. The simulation process is similar to the 30-taxon dataset and uses Simphy and INDELible. We run two chains of MrBayes MCMC for 600000 generations on each gene alignment. Here, we use the quartet sampling strategy described before.

Biological Datasets. We test QuCo on the dataset of *Pseudapis* genus of bees of [14] with 32 species and 1291 UCEs from the subfamily *Nomiinae* (Halictidae). We use the MrBayes posterior estimations from the original study and run QuCo on all $\binom{32}{4}$ quartets of the dataset. Then, we combine quartets using ASTRAL and enrich its search space with 853 IQ-TREE gene trees.

6.2.3 Evaluation Procedure and Metrics

We compare QuCo to ASTRAL and BUCKy-quartet [76]. As input to ASTRAL, we use maximum *a posteriori* (MAP) MrBayes or maximum likelihood (ML) IQ-TREE gene trees. On the 101-taxon dataset where MAP becomes impossible to estimate, we use the MRC summary for each gene. For four-taxon trees, the ASTRAL tree is equivalent to the most common topology among the MAP gene trees. On 4-taxon dataset, we also include BUCKy, which has been shown to have accuracy similar to MSC-based co-estimation methods [30].

We evaluate the species tree accuracy in terms of the topology and internal branch length. For quartet trees, we simply report how often the inferred topology is correct, and for larger trees, report the portion of true branches missing in the estimated tree (which is equal to the normalized

RF distance because all trees are fully resolved). To evaluate the accuracy of the branch length, we report the ratio between d estimated by QuCo to the true branch length, only considering cases where the species tree topology is correct. We compare accuracy to the ASTRAL branch lengths. Gene tree accuracy is evaluated by comparing how often the MAP estimate is correct before or after the co-estimation update step performed by QuCo using Equation (6.6).

6.3 Results

6.3.1 Simulation Results

Felsenstein's zone simulations.

Topological species tree accuracy. QuCo is at least as accurate as and in many conditions far more accurate than ASTRAL (Fig. 6.2a). Across all conditions, QuCo finds the correct tree in 1953 out of 3840 replicates, whereas ASTRAL is correct in 1572 cases. The improvements are most clear in model conditions where $^l\!/_s = 10$. For example, with $l = 10s = 0.2$ and 800bp sequences, QuCo has 100% and 60% accuracy respectively with $d = 0.3$ and $d = 0.2$ compared to 65% and 10% for ASTRAL. When s and l are close, both ASTRAL and QuCo work well. For example, both methods recover the true species tree in all replicates when $^l\!/_s \leq {}^5\!/_2$ (top right corner) with $d = 0.2$ or $d = 0.3$ and in most cases for $d = 0.1$. On the other hand, when $^l\!/_s > 20$ (bottom left corner), even with 1600 bp sequences, neither method recovers true topology in any replicate; with $^l\!/_s = 20$, QuCo recovers the true species tree between 5% to 70% of times if the sequence length is at least 800 bp, but ASTRAL continues to infer the wrong tree in every case.

Compared to BUCKy, QuCo shows improvements in many but not all conditions, and improvements are less substantial (Fig. 6.2b). When ILS is lower ($d = 0.3$), the two methods are identical or similar except in three l, s combinations where QuCo has a substantial advantage for 400bp or longer alignments and one case where BUCKy has a small advantage with 400bp

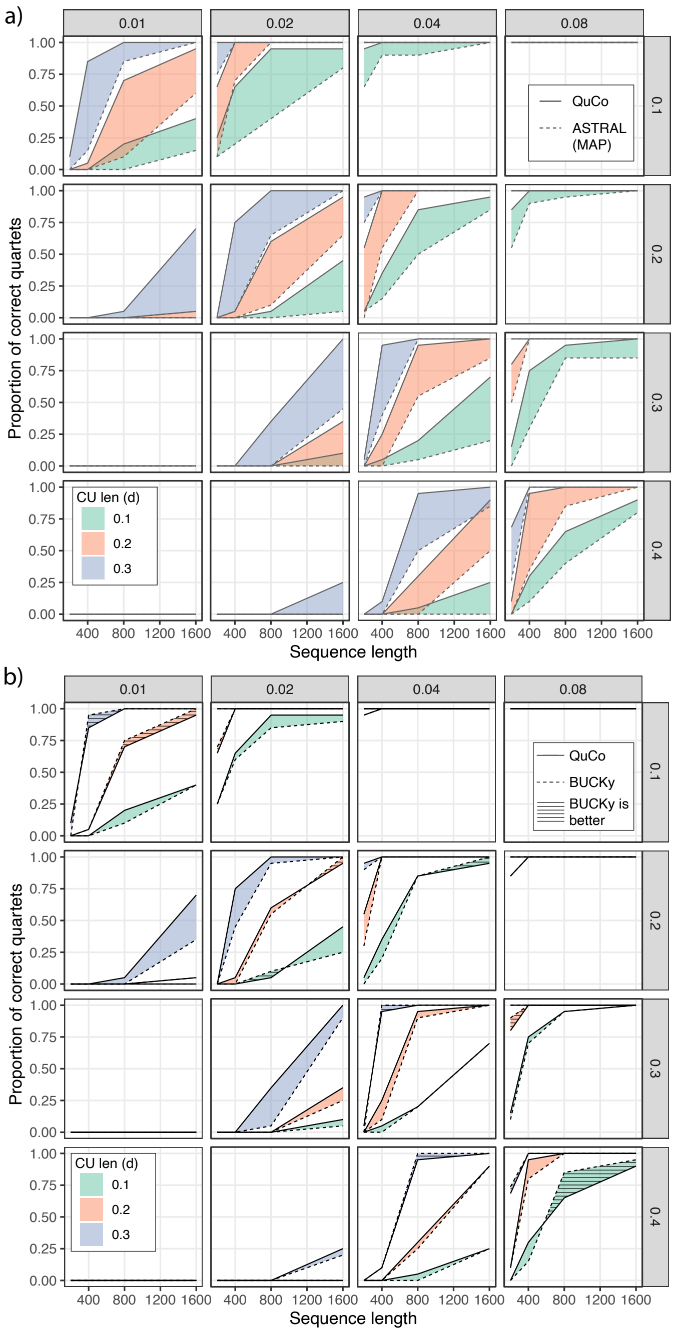


Figure 6.2: Felsenstein's zone quartet simulations comparing Quco to ASTRAL (a) or Bucky-Quartet (b). Each box shows a combination of long l (rows) and short s branch lengths (columns), and colors delineate ILS level controlled by d. Each ribbon shows the improvement of QuCo over ASTRAL or BUCKy, all run on MrBayes gene trees. When the ribbon is patterned, BUCKy is better than QuCo.

alignments. Across all conditions with $d = 0.3$, QuCo is correct in 811 out of 1280 replicates tested, which is 3% higher than BUCKy (788). With $d = 0.2$, the two methods are similar with small advantages for QuCo in nine conditions out of 64 and for BUCKy in four conditions. With the highest level of ILS, QuCo and BUCKy are each substantially better in some conditions. Among all species trees tested, the number of times QuCo is correct is 50 times more than BUCKy.

Consistently through all model conditions, longer sequences (hence more signal) in the gene trees result in more accurate species tree estimation, as expected. When sequence are as short as 200bp, the correct topology is almost never recovered when ${}^{l}/_{s} \geq 10$; with 400bp alignments, all methods fail in most cases when ${}^{l}/_{s} \geq 15$. Even some difficult cases such as $l = 0.3, s = 0.02$ or $l = 0.4, s = 0.04$ are rescued when using QuCo and to a lesser degree using BUCKy as long as sequences are sufficiently long; in these conditions, the accuracy can go from zero up to one with $d = 0.3$. The impact of longer sequences is also clearly observed in conditions with moderate ${}^{l}/_{s}$ (e.g., $l = 5 \times s = 0.1$ or $l = 5 \times s = 0.2$) where close to perfect accuracy is obtained by QuCo and BUCKy but not ASTRAL with 1600bp sequences even with $d = 0.1$.

As expected, higher levels of ILS (i.e., lower d) make inference harder for both methods. There are, however, conditions where QuCo is quite robust to the level of ILS while ASTRAL is not. For example, for $l = 0.3, s = 0.04$, with 1600bp sequences, QuCo has 70% accuracy for the highest ILS level and 100% in the other cases. In contrast, ASTRAL accuracy degrades with increased ILS (perfect for $d = 0.3$, 85% for $d = 0.2$, and 20% for $d = 0.1$).

Branch length accuracy. When the species tree is correct, QuCo branch lengths are much closer to true branch lengths than ASTRAL branch lengths with all sequence lengths (Fig. 6.3). The under-estimation bias of ASTRAL branch lengths as a result of inaccurate gene trees, as shown by [136], is vastly reduced by QuCo. With the most difficult model conditions, both methods under-estimate the internal branch length while QuCo produces far more accurate estimates. In most model conditions, longer sequences help QuCo to estimate more accurate branch lengths. However, when $s = 0.08$, QuCo can surprisingly *over*-estimate branch lengths by 12% with $\geq$800bp data.

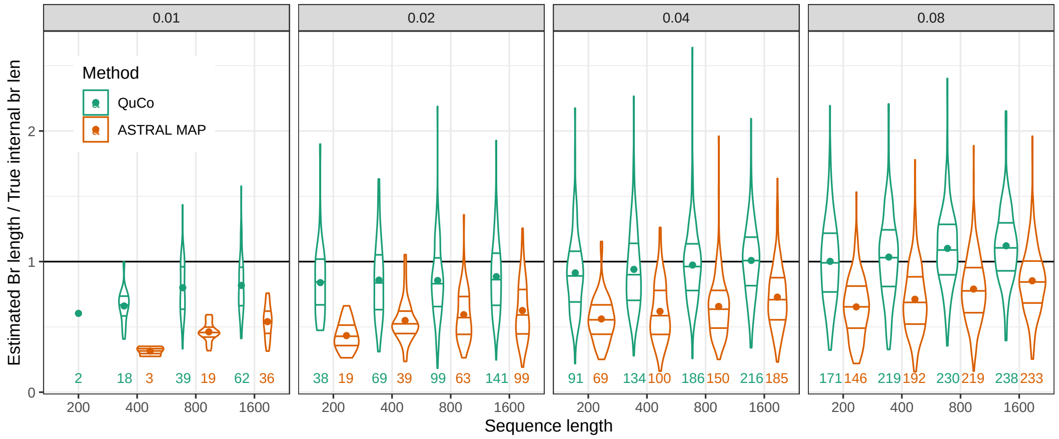


Figure 6.3: Branch length accuracy on Felsenstein's zone simulations. We show the distribution of estimated branch length divided by true branch length when the species tree is correctly estimated (number of such cases shown in each case). The lines delineate the four quartiles and the dot shows the mean. We combined all l values into the same figures, distinguishing only s (boxes).

Gene tree error. Unlike universal improvements in the species tree accuracy, the gene tree accuracy of QuCo is mixed (Fig. 6.4). The quartet gene trees produced by QuCo are better than the original MAP gene trees under most conditions where the species trees are improved compared to ASTRAL and under most "easy" conditions where both ASTRAL and QuCo find the correct tree. However, under the most challenging conditions where neither method can find the correct species tree (e.g., for $^{l}/_{s} > 20$), the QuCo gene trees are *less* accurate than the raw MAP trees. Note that co-estimation by QuCo first computes the species tree and then strictly increases the probability for gene tree topologies that match the species tree at the expense of those that disagree with it. Thus, a reduced gene tree accuracy with incorrect species trees can be expected.

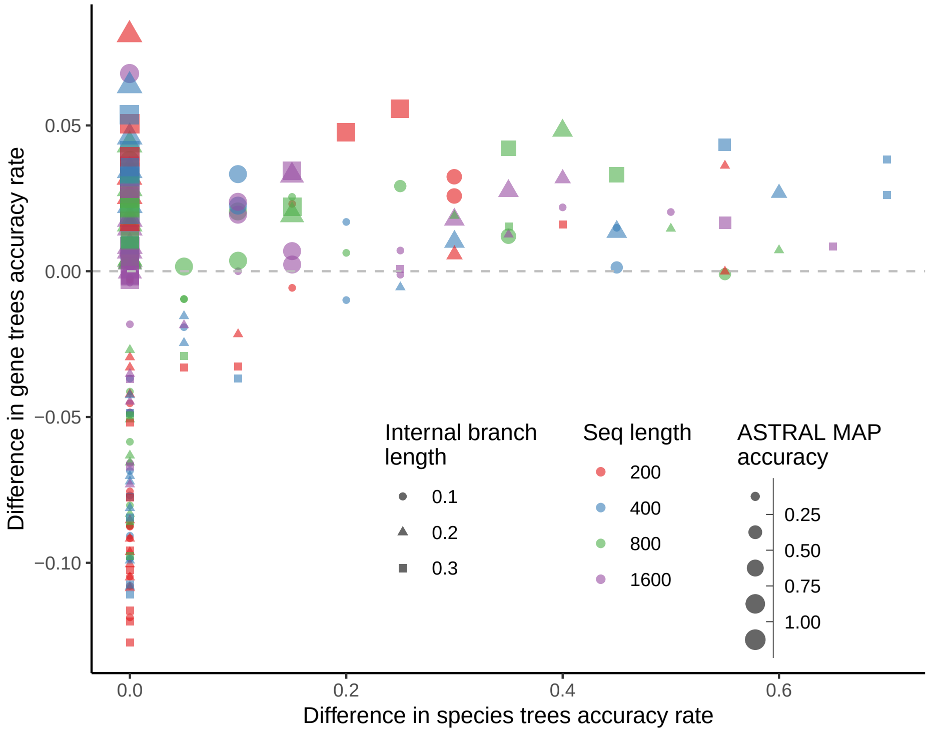


Figure 6.4: Gene tree estimation error on Felsenstein's zone simulations. Each dot corresponds to one model condition, with the x-axis showing the improvement in species tree accuracy by QuCo compared to ASTRAL and the y-axis showing the improvement in the average gene tree accuracy for all genes. The size of dots corresponds to the accuracy of ASTRAL species trees.

30-taxon datasets

On the larger 30-taxon dataset, depending on the model condition, QuCo+IQ-TREE either matches or improves on the accuracy of ASTRAL+IQ-TREE (Fig. 6.5). Note that testing BUCKy was not possible for these larger data. The improvements are obtained both for conditions with high and low deviations from the strict clock but are less clear for conditions with moderate deviations. When deviations are high, accuracy improves as a result of increasing the number of gene trees from 200 to 500 for both methods, but the improvements are larger for QuCo (from mean error of 11% to 7% versus from 12% to 10%). Note that the inputs to ASTRAL and QuCo are not identical in this experiment: The ML gene trees are inferred from the entire set of species, whereas quartet tree likelihoods are inferred per quartet. Thus, it is reasonable to expect the input

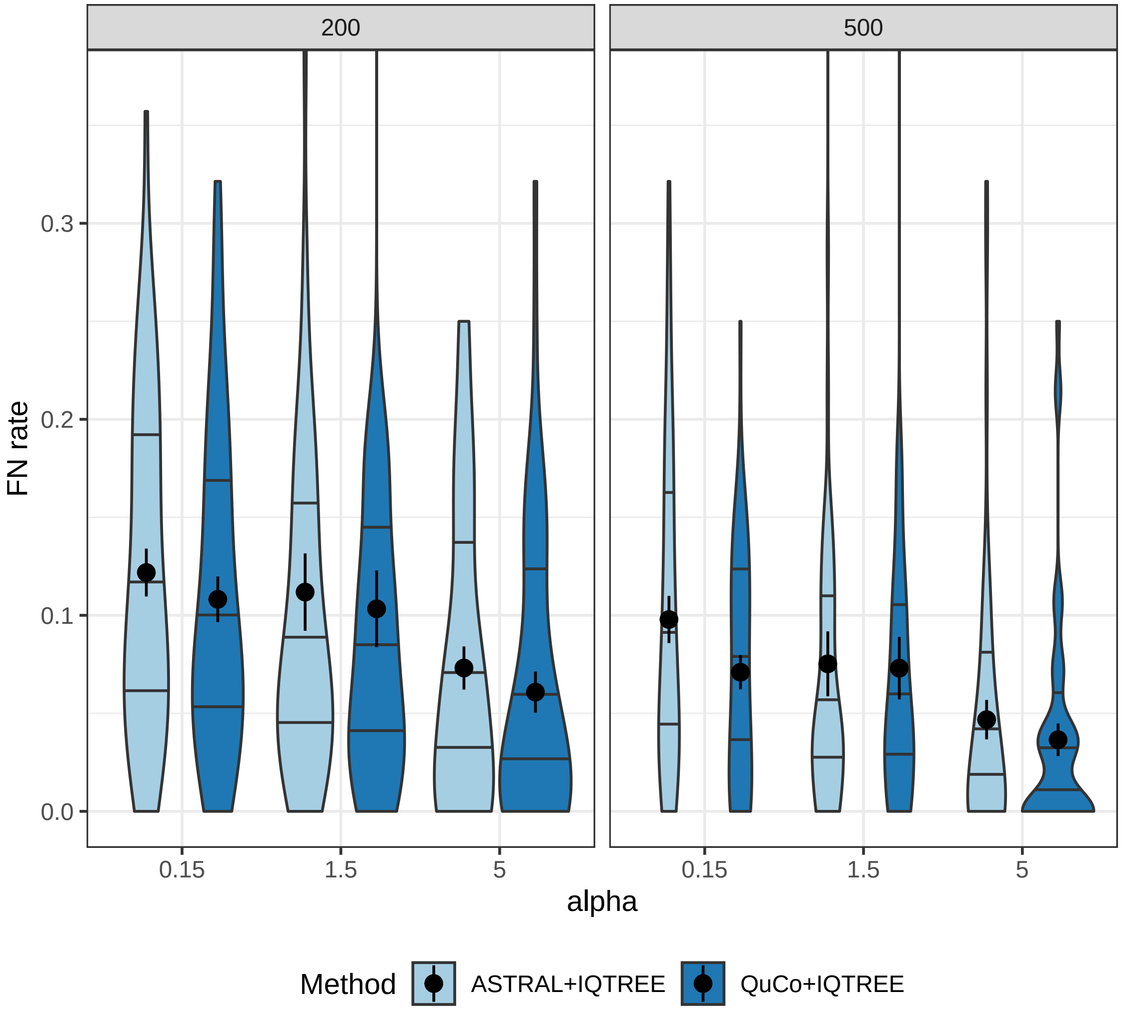


Figure 6.5: 30-taxon dataset. Left: Comparison of the error rate of the species tree generated by running ASTRAL on IQ-Tree ML gene trees and QuCo on IQ-Tree quartet likelihoods with 200 and 500 genes of 30-taxon dataset. The x-axis shows deviation from clock represented by parameter α (inverse of the variance of rate multipliers). Each box is over 50 replicates.

to QuCo to be more prone to LBA than ASTRAL, making it more remarkable that it has a lower error in its output.

On this dataset, we also compare ASTRAL and wMaxCut as supertree methods for the step where quartet species trees are combined (Fig. 6.7). The two ways of combining the quartets are competitive, with ASTRAL performing slightly better (worse) for low (high) deviations from the clock. Due to similar levels of accuracy, we use ASTRAL as the supertree method elsewhere.

The large 101-taxon dataset

On the larger dataset, where the sampling strategy is necessary, the number of internal nodes in the ASTRAL guide trees ranges from 69 to 84 (mean 78) compared to 99 nodes for a fully resolved tree. Our sampling strategy selects between 3127 and 51272 quartets (mean: 22508), which we resolve using QuCo. The polytomies range from small (degree 4) to a maximum degree of 19 (mean: 5), and the guide trees have no incorrect branches. After the polytomies of the guide tree are refined using QuCo, we observe a 12% decrease in the average topological error (Table 6.1) compared to the original ASTRAL tree run on MrBayes MRC gene trees and a 40% decrease compared to ASTRAL run on Fasttree gene trees with branches with bootstrap support below 10% contracted, which is the recommended setting used by [169].

6.3.2 Application on a Biological Dataset

The species tree that we recover by running QuCo matches the ASTRAL tree reported by [14] on gene trees created using Phylobayes, which is designed to reduce LBA [77]. [14] have reported 30 ASTRAL trees from gene trees created by ML and Bayesian methods, and these trees differ in five areas compared to the concatenation tree. The tree reported by QuCo differs from concatenation in two nodes and is identical for the other nodes (Fig. 6.6) and also differs from ASTRAL on MrBayes in one of those two nodes. These two nodes involve the two samples with the worst sequencing success, *Ruginomia rugiventris* and *Stictonomia schubotzi*. Both of these taxa have over 75% undetermined positions in the concatenated matrix and are present in less than half of all loci, making them hard to place.

6.4 Discussion

We introduced an algorithm for quartet co-estimation (QuCo for short) of species trees and gene trees. We showed that QuCo had better accuracy than ASTRAL in quartet simulations with LBA. By considering gene tree uncertainty, QuCo also outperformed ASTRAL under the anomaly zone simulations when the number of genes was limited. It can be easily proved that if all $\mathbf{P}_{t,i}$ values are either 0 or 1 (i.e., in the absence of gene tree estimation uncertainty), QuCo is equivalent to picking the most frequent quartet as the species tree, as is done in ASTRAL. The improvements, then, are a result of considering gene tree uncertainty. As the number of genes increased, better handling of uncertainty appeared to be less consequential as QuCo and ASTRAL converged in accuracy in the anomaly zone simulations. Compared to the alternative co-estimation method BUCKy, QuCo had a small advantage in accuracy; however, note that BUCKy has limited scalability [165].

Despite the fact that the method first infers gene trees independently, it is a co-estimation method because the species tree maximizes the joint likelihood marginalized over all possible gene tree topologies. This marginalization was computationally tractable because we consider quartets independently and can use a simple equation (6.2) for the likelihood. The likelihood of gene trees for more taxa is much harder to compute and requires exponential time [163]. Alternatively, one can assign branch lengths to gene trees in the same unit as the species tree to make likelihood calculation fast. However, this imposes a different challenge: the need to assume a distribution for mutation rates and population sizes, which further increases the number of parameters that need to be sampled. Many co-estimation methods side-step this challenge by assuming a strict molecular clock, an assumption that decades of research has proved problematic. By focusing on gene tree topologies and species tree branch length in the coalescent unit, we avoid using a strict gene tree clock model while making the problem *easier* to solve (i.e., requiring fewer parameters). Our approach did require assuming the independence of substitution branch lengths from the species tree topology and internal branch length. However, we note that none of our empirical tests made any such assumptions; thus, the high empirical performance of the method indicates these assumptions are not lethal. We also note that working with topologies comes with the caveat that our solution does not produce gene trees with branch lengths. Moreover, while a topology-centric view makes the likelihood calculations feasible for four species, going beyond four species requires the heuristic supertree approach. Thus, our simplifying assumptions have the benefits of 1) freeing us from assuming restrictive models of rate change across the tree, and 2) fast calculations of likelihood; however, they also render our method more heuristic than full Bayesian co-estimation methods.

Interestingly, while QuCo clearly increased species tree accuracy, it appeared less effective in increasing gene tree accuracy, especially when the species tree was not improved. This trend is in contrast to some of the existing co-estimation methods, such as the iterative method of Wang and Nakhleh [159], that are effective in increasing the gene tree accuracy but less so in terms of the species tree. While these patterns call for further study in the future, two points should be emphasized. By marginalizing over gene tree distribution, QuCo can improve the species tree, even when the maximum likelihood gene tree (given the species tree) is not improved. Moreover, when QuCo fails to improve the quartet species tree, it has no chance of improving the gene tree, and in fact, it likely degrades it. Finally, note that QuCo essentially generates a distribution over topologies for each quartet in each gene tree. When more than four species are available, a quartet amalgamation method such as wMaxCut needs to be used to compute the final updated gene trees.

The scalability of QuCo comes from the fact that the inference for each quartet is fast. Given the **P** matrix, the optimization step takes a fraction of a second per quartet. Even on the 30-taxon data, the optimization step takes close to one hour across all 27,405 quartets. Given MrBayes outputs, computing **P** is conceptually easy, and with appropriate implementation, can be fast (with I/O being the bottleneck). The entire running time, including the I/O heavy calculation of **P**, is still reasonably fast. For example, for the biological dataset with 32 species, QuCo took 12 hours to analyze all 35960 quartets across 1200 genes with no parallelization (mostly calculation of **P**). This can be run in parallel; using 80 cores, 13 minutes is enough to analyze all quartets. The final step of combining the quartet trees is also fast, taking 36 seconds using ASTRAL and only a couple of seconds using wMaxCut. The more time-consuming part of the pipeline, by far, is to run MrBayes on all gene trees. However, this step can be done in parallel and is much more manageable than co-estimation. For example, [14] reported that each MrBayes run on 32 taxa took 6.7 minutes on average. Running methods like MrBayes on thousands of genes with hundreds of species is doable. For even larger datasets where MrBayes may not scale, our results showed that using IQ-TREE quartet likelihoods, which are extremely fast to compute, can be very accurate. To summarize, 101-taxon is by no means the limit of the method.

Analyzing a large number of taxa (e.g., beyond 50) requires quartet selection strategies instead of using all $\binom{n}{4}$ quartets. Quartet subsampling is a problem that has been studied in the literature [144, 35] and solutions with quadratic [135] or even quasi-linear [17] numbers of quartets have been proposed. We left the exploration of such approaches to the future. Instead, we tried a simple method where a guide tree (here, ASTRAL) is estimated and uncertain branches are contracted. The polytomies left in the tree are the difficult parts of the tree, hence our desire to focus the quartet sampling around the polytomies. Our probabilistic leaf sampling strategy uses the well-established insight that short quartets (those with leaves closer to the polytomy) are easier to resolve correctly than long quartets [47, 144]. While our sampling strategy proved effective, we believe better methods may be possible, including those that would guarantee that the number of quartets increases quasi-linearly or quadratically with the number of species.

Method availability The method presented here is available on `https://github.com/maryamrabiee/quartet_coestimation`

6.5 Proofs

6.5.1 Likelihood maximization

Proof of Proposition 1. We can rewrite (6.5) without the logarithm as

$$P(\mathcal{S};\theta) = A\prod_{i=1}^{k}(\beta_i + x\alpha_i)$$

where $\beta_i = \mathbf{P}_{t,i}$, and $\alpha_i = 1/3 - \mathbf{P}_{t,i} = 1/3 - \beta_i$, and $x = e^{-d}$. Since α_is and β_is are constant, $P(\mathcal{S};\theta)$ is a polynomial in x and can have at most k roots of the form

$$-\frac{\beta_i}{\alpha_i} = \frac{\beta_i}{\beta_i - 1/3}$$

and $k-1$ local optima that must each be between two roots. Note that only valid values of x in our optimization are $0 < x < 1$ corresponding to $0 < d < \infty$. Thus, we are interested in local optima in that region. However, every root of the form shown is negative when $\beta < 1/3$ and is > 1 for $1/3 < \beta_i \leq 1$. Thus, none the roots are in the $0 < x < 1$ region we are interested in. Since the polynomial has no root in 0 to 1, it can have only one local optimum in that region. Note also that $x \to 0$ and $x \to 1$ both result in non-negative likelihood values, and thus, there must be one valid maximizer to the function.

□

Derivatives

The derivatives of the log likelihood function are given to the optimizer:

$$l'(t,d;\mathbf{P}) = \sum_{i=1}^{k} \frac{3\mathbf{P}_{t,i} - 1}{1 + 3\mathbf{P}_{t,i}(e^d - 1)})$$

$$l''(t,d;\mathbf{P}) = \sum_{i=1}^{k} \frac{3\mathbf{P}_{t,i}e^d(1 - 3\mathbf{P}_{t,i})}{(1 + 3\mathbf{P}_{t,i}(e^d - 1))^2}$$

6.5.2 Simulating Long branch attraction with MSC

Proof (sketch), Lemma 1. Recall the balanced quartet tree with length $\frac{d}{2}$ above the two speciation nodes as shown in Figure 6.1. By symmetry, all terminal branches have the same length in coalescent units. W.l.o.g., we take branch A. Recall that the probability density function for the coalescence of two lineages in time t before present is given by e^{-t} in coalescent time units. We compute the expectation by conditioning on three scenarios: (I) lineages A and B coalesce before the root, (II) lineage A and B do not coalesce before the root but lineage C and D do, and (III) neither lineage A and B nor lineages C and D coalesce before the root. Let c_3 and c_4 be the expected time to coalescence between A and another branch among 3 and 4 branches in total, respectively. Then, the expected length of terminal branch of A above its common ancestor with B in the species tree is:

$$\overbrace{\int_0^{\frac{d}{2}} te^{-t}\,\mathrm{d}t}^{I} + \overbrace{e^{-\frac{d}{2}}(1-e^{-\frac{d}{2}})\left(c_3+\frac{d}{2}\right)}^{II} + \overbrace{e^{-2\frac{d}{2}}\left(c_4+\frac{d}{2}\right)}^{III}$$

The first term simply follows from the definition of expectation. The second and third terms first compute the probability of lack of coalescence ($e^{-d/2}$) on one or both sides, and multiple by the expected length in each case. The c_3 and c_4 terms give the expected length in the root (by definition) and $^{d}/_{2}$ is added to account for the length on the branch above common ancestor of A and B. We compute c_3 again using conditional expectation:

$$c_3 = \frac{1}{3}\frac{2}{3} + \frac{1}{3}\left(\frac{1}{3}+1+1\right) = 1$$

The first term conditions on A being the first lineage to coalesce with another (probability $\frac{2}{3}$), and uses the fact that the expected length of the first coalescent among N lineages is $1/\binom{N}{2} = \frac{1}{3}$. The second term is conditioned on A not being the first lineage to coalesce with another and is computed similarly. In this scenario, A will continue for the first coalescent event (length$\frac{1}{3}$), up to the final coalescence $1/\binom{2}{2} = 1$; we also need to add the length of the branch to the other side of the deepest coalescence because we are dealing with unrooted trees. With similar logic, we can compute:

$$c_4 = \frac{1}{6}\frac{1}{2} + \frac{1}{3}\left(\frac{1}{6} + \frac{1}{3}\right) + \frac{1}{6}\left(\frac{1}{6} + \frac{1}{3} + 1 + 1\right) = \frac{2}{3}$$

where the three terms correspond to A being the first, the second, and the last branch to coalesce. Replacing these terms in the first equation we get the expectation equals:

$$1 - e^{-\frac{d}{2}}\left(1 + \frac{d}{2}\right) + e^{-\frac{d}{2}}(1 - e^{-\frac{d}{2}})\left(1 + \frac{d}{2}\right) + e^{-2\frac{d}{2}}\left(\frac{2}{3} + \frac{d}{2}\right) = 1 - \frac{e^{-d}}{3}$$

□

6.6 Acknowledgement

Chapter 6, in full, has been submitted for publication of the material as it may appear in “QuCo: Quartet-based Co-estimation of Species Trees and Gene Trees”, (2022) Rabiee, Maryam; Mirarab, Siavash, ISMB. The dissertation author was the primary investigator and first author of this paper.

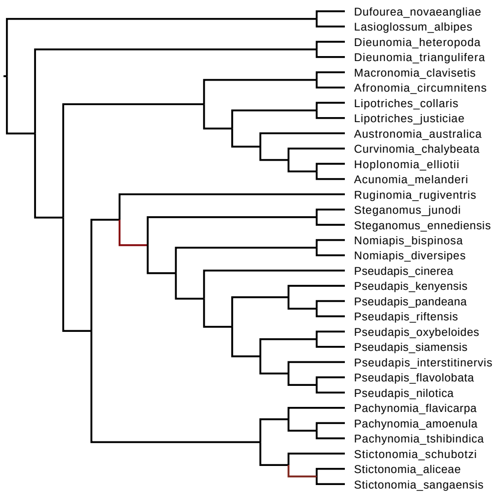


Figure 6.6: Species tree created by running ASTRAL on all quartets estimated by QuCo on bees dataset

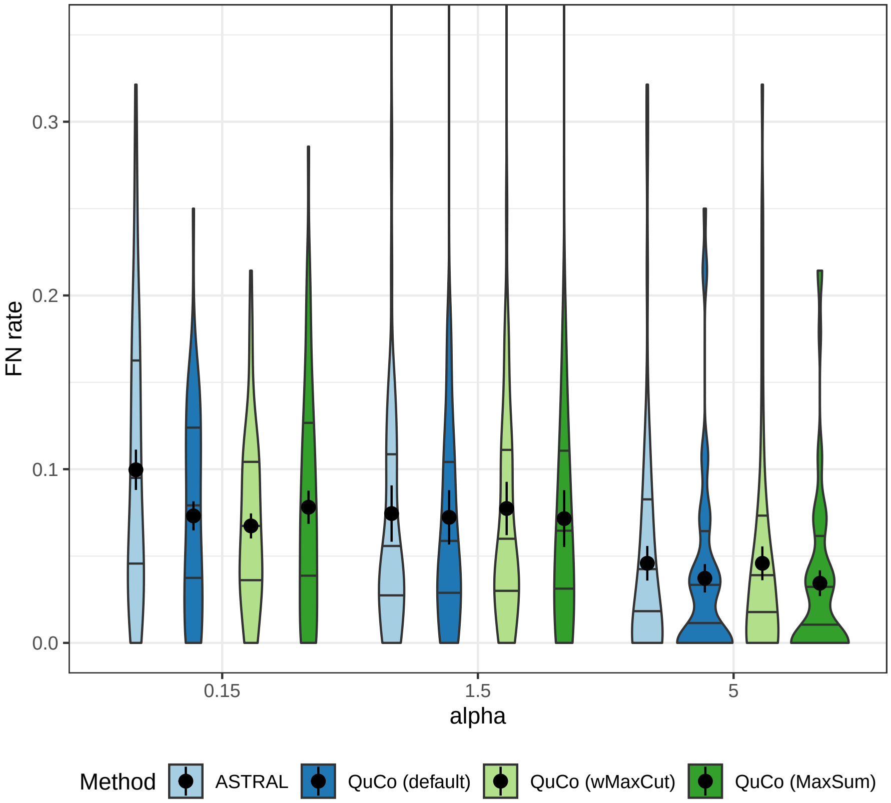


Figure 6.7: Comparison of the error rate of the species tree generated by running two versions of QuCo: the default version using ASTRAL for combing quartets, and the version combining quartets using wMaxCut. All methods are compared on 50 replicates with 500 genes.

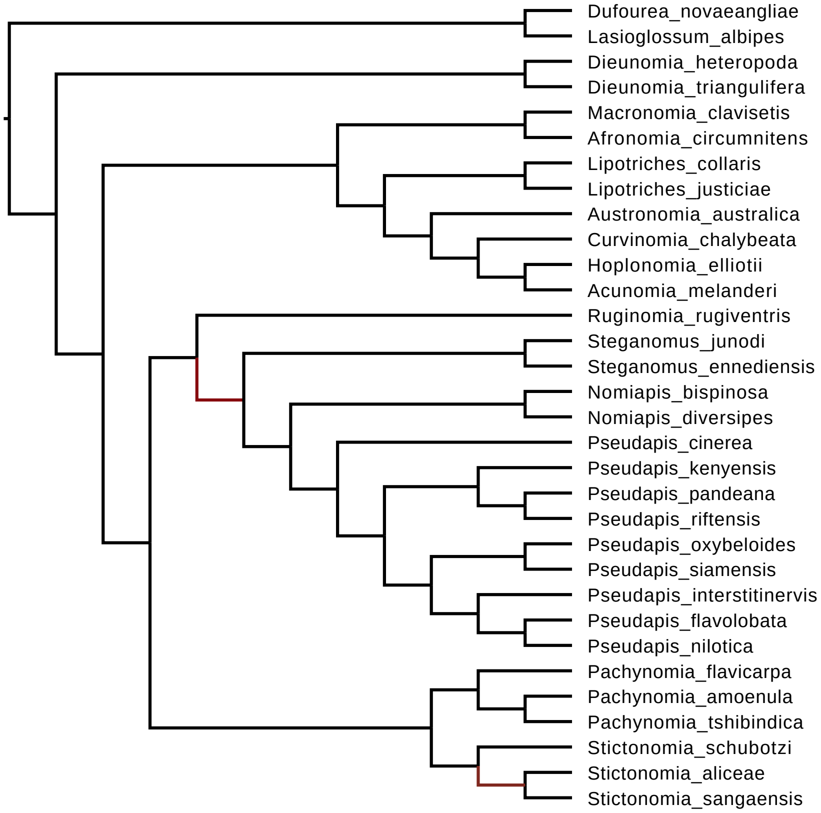


Figure 6.8: Species tree created by running ASTRAL on all quartets estimated by QuCo on bees dataset

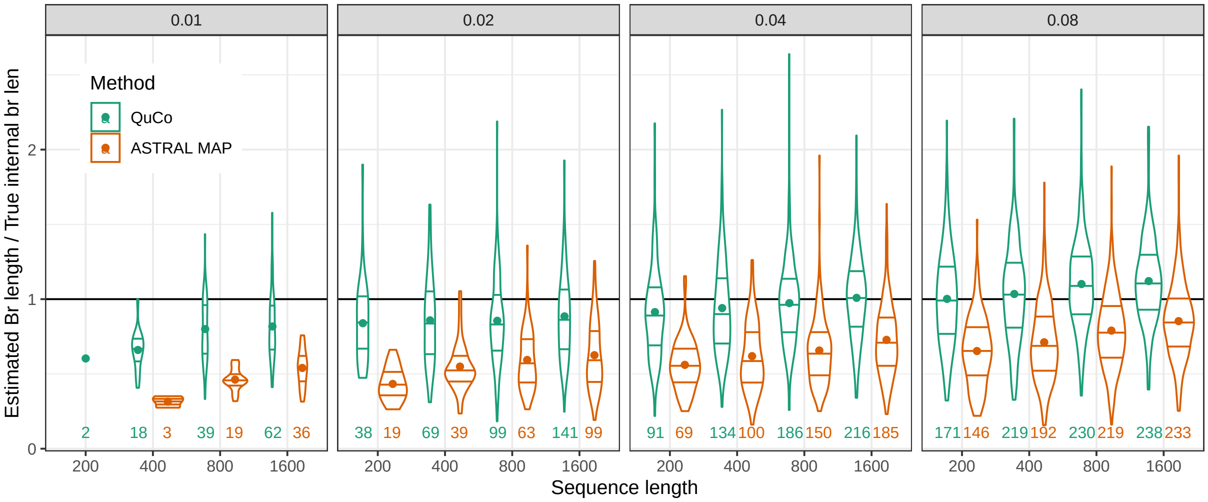


Figure 6.9: Branch length accuracy on Felsenstein's zone simulations, showing the distribution of estimated branch length divided by true branch length for correctly estimated species tree (the number of such cases shown in each case). Lines show the four quartiles and the dot shows the mean. Each box corresponds to a value of *s*, combining all *l* values.

Table 6.1: QuCo results with sampling on ASTRAL-III dataset of 101 species compared to ASTRAL run on MRC trees of MrBayes and Fasttree gene trees

	QuCo	ASTRAL+MRC	ASTRAL+Fasttree (10%)
Mean	4.6%	5.2%	7.7%
Median	41.%	5.1%	7.1%

Appendix A

Supplementary materials for ”INSTRAL: Discordance-aware Phylogenetic Placement using Quartet Scores”

A.1 Statistical consistency of ordered placement

The optimal solution to the Maximum Quartet Support Species Tree is a statistically consistent estimator of the species tree under the MSC model given error-free gene trees drawn randomly from the MSC model [99]. Thus, for any desired probability of recovering the correct species tree, there is a number of genes such that with those many genes or more, the species tree will be recovered with at least that probability.

Let the series of species placed in an ordered placement scenario be indexed by $1 \leq i \leq M$. Let E_i be the event that query indexed i is correctly placed and for ease of notation let E_0 be a trivial event with $P(E_0) = 1$. Recall that INSTRAL solves the placement problem optimally for each placement. Therefore, for any desired ε', assuming all previous placements are correct, there is a number of genes $k_i(\varepsilon')$ such that if we have at least those many genes, $P(E_i|E_{i-1} \dots E_0) \geq 1 - \varepsilon'$.

Let E be the event that *all* reads are placed correctly. We have to prove that for a desired ε, there is a k such that $P(E) \geq 1 - \varepsilon$. Let $\varepsilon' = \frac{\varepsilon}{M}$. Set

$$k = \max_i k_i(\varepsilon') = \max_i k_i(\frac{\varepsilon}{M})$$

. We claim that with this choice of k, we have $P(E) \geq 1 - \varepsilon$.

Note that since each placement is independent of future placements,

$$P(E) = \prod_1^M P(E_i | E_{i-1}, \ldots, E_0)$$

It is easy to see that $P(E_i | E_{i-1}, \ldots, E_0) \geq 1 - \varepsilon'$ for all $1 \leq i \leq M$ because by our choice, $k \geq k_i(\varepsilon')$ for all i. Thus,

$$P(E) \geq \prod_1^M (1 - \varepsilon') = (1 - \varepsilon')^M \geq 1 - M\varepsilon' = 1 - \varepsilon\ .$$

This completes the proof.

Appendix B

Supplementary materials for “Forcing external constraints on tree inference using ASTRAL“

B.1 Commands and versions used

B.1.1 Running Constrained ASTRAL

Constrained ASTRAL in this paper refers to version 5.6.7 of the code available on: https://github.com/maryamrabiee/Constrained-search

Unconstrained ASTRAL refers to version 5.6.3 of ASTRAL-III available on https://github.com/smirarab/ASTRAL

Constrained ASTRAL program were run with following command:

```
java −jar <program> −i <input> −o <output> −e <constrainttree> >
<completedtrees> 2> <log>
```

B.1.2 Contracting Low Support Branches

In order to contract gene tree branches with bootstrap up to a certain threshold we used this command:

```
nw_ed genetree ’i␣&␣(b<=$threshold)’ o
```

Appendix C

Supplementary materials for ”Multi-allele species reconstruction using ASTRAL ”

C.1 Simulation procedure

In order to generate D1 we used Simphy [89] with the following exact command

```
simphy -rs 330 -rl u:50,1000 -rg 1 -sb lu:0.0000001,0.000001
-sd lu:0.0000001,sb -st ln:13,0.5 -sl u:20,200 -si f:5
-sp u:10000,1000000 -su ln:-17.27461,0.6931472 -hs ln:1.5,1
-hl ln:1.551533,0.6931472 -hg ln:1.4,1
-cs 9644 -v 3 -o tre -ot 0 -op 1 -od 1
```

For D2 with 0.5M generations we used

```
simphy -rs 50 -rl U1000,1000 -rg 1 -st U1000000,1000000 -si U5,5
-sl U200,200 -sb U0.000001,0.000001 -p U200000,200000  -hs L1.5,1
-hl L1.2,1 -hg l1.4,1 -u E10000000 -so U1,1 -od 1 -or 0 -v 3
-cs 293745 -o model.200-5.1000000.0.000001
```

For D2 with 1M generations we used

```
simphy -rs 50 -rl U1000,1000 -rg 1 -st U1000000,1000000 -si U5,5
-sl U200,200 -sb U0.000001,0.000001 -p U200000,200000 -hs L1.5,1
-hl L1.2,1 -hg l1.4,1 -u E10000000 -so U1,1 -od 1 -or 0 -v 3
-cs 293745 -o model.200-5.1000000.0.000001
```

For D2 with 2M generations we used

```
simphy -rs 50 -rl U1000,1000 -rg 1 -st U2000000,2000000 -si U5,5
-sl U200,200 -sb U0.000001,0.000001 -p U200000,200000 -hs L1.5,1
-hl L1.2,1 -hg l1.4,1 -u E10000000 -so U1,1 -od 1 -or 0 -v 3
-cs 293745 -o model.200-5.2000000.0.000001
```

Gene length

D1 (heterogeneous) To draw the per-replicate μ and σ parameters of the log-normal distribution used for sequence length, we use the following approach. We first draw a number from a gamma distribution with shape of $k = 2\log(1000) - 1/8 = 209.3259$ and scale of $\theta = 0.033$. Then we subtract this number from $\log(1000)/0.033 = 13.69051$ to get the μ. The scale of the log normal distribution is also randomly drawn from a uniform distribution of $(0.3, 0.7)$.

D2 (homogeneous) The log mean is drawn uniformly between 5.7 and 7.3, which correspond to 300 sites to 1500 sites. Thus, the average alignment length for each replicate is a random value between 300 and 1500. The log standard deviation for the log normal distribution is also drawn uniformly between 0.0 and 0.3. Base

Appendix D

Supplementary materials for ”SODA: Multi-locus species delimitation using quartet frequencies”

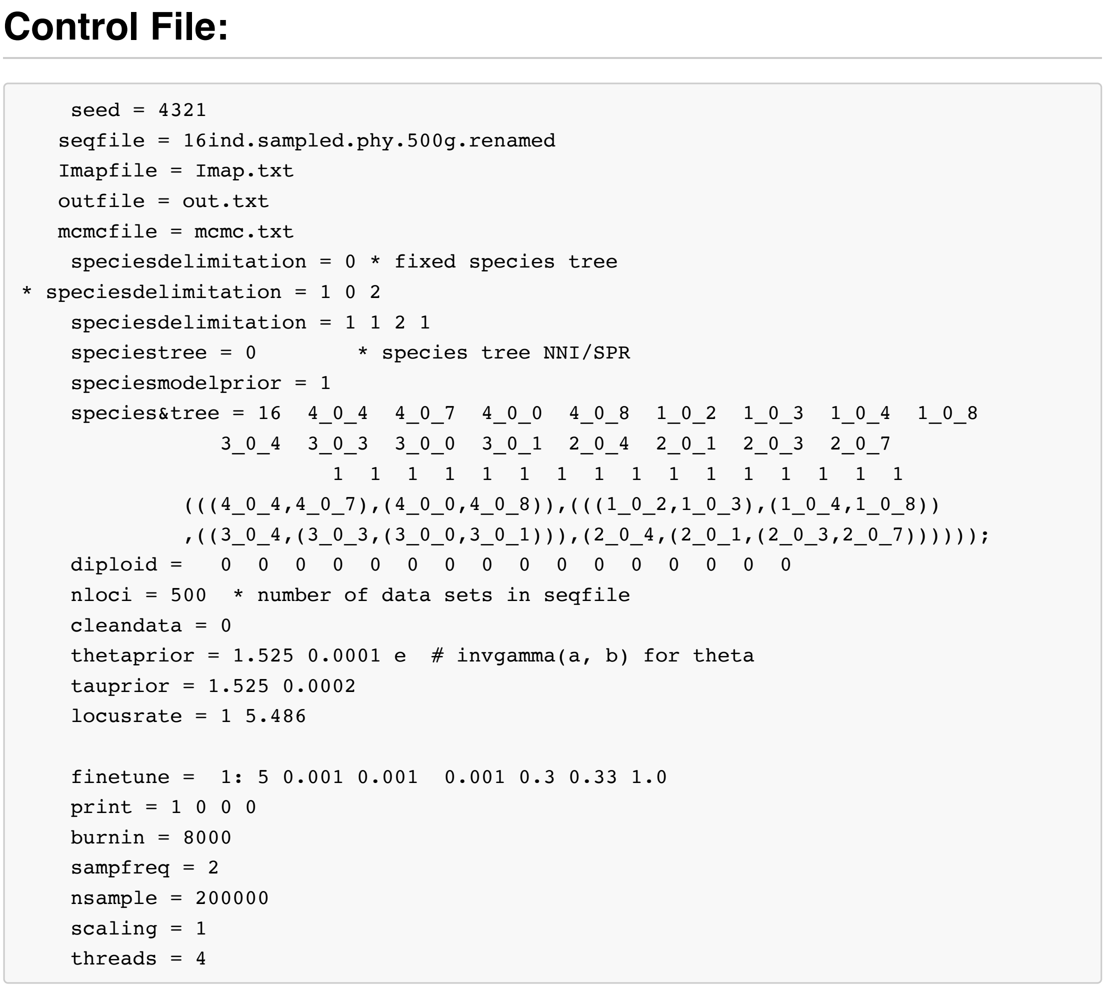
D.1 Supplement: commands and parameters

```
simphy -rs 20 -rg 1 -rl f:1000 -sb lu:0.00001,0.00001 -sd f:0 -st f:200000 -sl f:4 -si f:10
-sp u:10000,500000 -su ln:-19,0.6931472 -hs ln:1.5,1 -hl ln:1.551533,0.6931472 -hg ln:1.4,1 -cs
9644 -o yule4 -V 3
```

Table D.1: Parameters used in SimPhy simulation for small dataset

Arg	Description	value
RS	Number of replicates	20
RL	Number of loci	1000
RG	Number of genes	1
SB	Speciation rate	0.00001
SD	Extinction rate	0
ST	Maximum tree length	200000
SL	Number of taxa	4
SI	Number of individuals per species	10
SP	Global population size	Uniform(10000,500000)
SU	Global substitution rate	Log normal(-19,0.6931472)
HS	Species specific branch rate heterogeneity rates	Log normal(1.5,1)
HL	Gene family specific rate heterogeneity rates	Log normal(1.551533,0.693147)
HG	Gene by lineage specific rate heterogeneity rates	Log normal(1.4,1)
CS	Random number generator seed	9644

D.2 Supplementary figures and tables



Control File:

```
    seed = 4321
   seqfile = 16ind.sampled.phy.500g.renamed
   Imapfile = Imap.txt
   outfile = out.txt
   mcmcfile = mcmc.txt
    speciesdelimitation = 0 * fixed species tree
* speciesdelimitation = 1 0 2
    speciesdelimitation = 1 1 2 1
    speciestree = 0        * species tree NNI/SPR
    speciesmodelprior = 1
    species&tree = 16  4_0_4  4_0_7  4_0_0  4_0_8  1_0_2  1_0_3  1_0_4  1_0_8
                3_0_4  3_0_3  3_0_0  3_0_1  2_0_4  2_0_1  2_0_3  2_0_7
                         1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
             (((4_0_4,4_0_7),(4_0_0,4_0_8)),(((1_0_2,1_0_3),(1_0_4,1_0_8))
             ,((3_0_4,(3_0_3,(3_0_0,3_0_1))),(2_0_4,(2_0_1,(2_0_3,2_0_7))))));
    diploid =   0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
    nloci = 500  * number of data sets in seqfile
    cleandata = 0
    thetaprior = 1.525 0.0001 e  # invgamma(a, b) for theta
    tauprior = 1.525 0.0002
    locusrate = 1 5.486

    finetune =  1: 5 0.001 0.001  0.001 0.3 0.33 1.0
    print = 1 0 0 0
    burnin = 8000
    sampfreq = 2
    nsample = 200000
    scaling = 1
    threads = 4
```

Figure D.1: Sample control file used for running BPP with 500 genes

Appendix E

Supplementary materials for ”QuCo: Quartet-based Co-estimation of Species Trees and Gene Trees”

E.1 Supplement: commands and parameters

IQ-Tree

```
iqtree -s seqfile -st DNA -lmap ALL -wql -seed 1234567890 -m GTR -pre iqtree -n 0 -redo -keep-ident
```

MrBayes

```
begin mrbayes; set autoclose=yes nowarn=yes ;execute seq.nex; lset nst=6 rates=gamma; mcmc nruns=2 mcmcdiagn=yes samplefreq=500 stoprule=yes stopval=0.015 file=seq.nex; sumt; end;
```

Bibliography

[1] E. Allman, J. H. Degnan, and J. Rhodes. Species tree inference from gene splits by Unrooted STAR methods. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, PP(99):1–7, 2016.

[2] E. S. Allman, J. H. Degnan, and J. A. Rhodes. Determining species tree topologies from clade probabilities under the coalescent. *Journal of Theoretical Biology*, 289(1):96–106, 2011.

[3] E. S. Allman, J. H. Degnan, and J. A. Rhodes. Identifying the rooted species tree from the distribution of unrooted gene trees under the coalescent. *J. Math. Biol.*, 62:833–862, 2011.

[4] C. Ané, B. R. Larget, D. A. Baum, S. D. Smith, and A. Rokas. Bayesian estimation of concordance among gene trees. *Molecular Biology and Evolution*, 24(2):412–426, 2007.

[5] D. Arcila, G. Ortí, R. Vari, J. W. Armbruster, M. L. J. Stiassny, K. D. Ko, M. H. Sabaj, J. Lundberg, L. J. Revell, and R. Betancur-R. Genome-wide interrogation advances resolution of recalcitrant groups in the tree of life. *Nature Ecology & Evolution*, 1(January):0020, 1 2017.

[6] E. Avni, R. Cohen, and S. Snir. Weighted Quartets Phylogenetics. *Systematic Biology*, 64(2):233–242, 3 2015.

[7] M. Balaban, S. Sarmashghi, and S. Mirarab. APPLES: Fast Distance-based Phylogenetic Placement. *bioRxiv*, page 475566, 1 2018.

[8] M. S. Bansal. Linear-time algorithms for some phylogenetic tree completion problems under robinson-foulds distance. In *RECOMB International conference on Comparative Genomics*, pages 209–226. Springer, 2018.

[9] P. Barbera, A. M. Kozlov, L. Czech, B. Morel, D. Darriba, T. Flouri, and A. Stamatakis. EPA-ng: Massively Parallel Evolutionary Placement of Genetic Sequences. *Systematic Biology*, 68(2):365–369, 3 2019.

[10] M. Bayzid and T. Warnow. Gene tree parsimony for incomplete gene trees. In *17th International Workshop on Algorithms in Bioinformatics (WABI 2017)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2017.

[11] M. S. Bayzid and T. Warnow. Naive binning improves phylogenomic analyses. *Bioinformatics*, 29(18):2277–84, 2013.

[12] M. S. M. Bayzid, S. Mirarab, and T. Warnow. Inferring optimal species trees under gene duplication and loss. *Pacific Symposium on Biocomputing*, 18:250–261, 2013.

[13] S. A. Berger, D. Krompass, and A. Stamatakis. Performance, accuracy, and Web server for evolutionary placement of short sequence reads under maximum likelihood. *Systematic biology*, 60(3):291–302, 5 2011.

[14] S. Bossert, E. A. Murray, A. Pauly, K. Chernyshov, S. G. Brady, and B. N. Danforth. Gene Tree Estimation Error with Ultraconserved Elements: An Empirical Study on Pseudapis Bees . *Systematic Biology*, 0(0):1–19, 2020.

[15] B. Boussau, G. J. Szöllősi, and L. Duret. Genome-scale coestimation of species and gene trees. *Genome Research*, 23(2):323–330, 12 2013.

[16] H. Brinkmann, M. van der Giezen, Y. Zhou, G. P. de Raucourt, and H. Philippe. An Empirical Assessment of Long-Branch Attraction Artefacts in Deep Eukaryotic Phylogenomics. *Systematic Biology*, 54(5):743–757, 10 2005.

[17] D. G. Brown and J. Truszkowski. Towards a Practical O(n logn) Phylogeny Algorithm. pages 14–25. 2011.

[18] D. Bryant. Hunting for trees in binary character sets: efficient algorithms for extraction, enumeration, and optimization. *Journal of Computational Biology*, 3(2):275–88, 1996.

[19] D. Bryant, R. Bouckaert, J. Felsenstein, N. A. Rosenberg, and A. Roychoudhury. Inferring species trees directly from biallelic genetic markers: Bypassing gene trees in a full coalescent analysis. *Molecular Biology and Evolution*, 29(8):1917–1932, 2012.

[20] D. Bryant and M. Steel. Constructing Optimal Trees from Quartets. *Journal of Algorithms*, 38(1):237–259, 1 2001.

[21] J. G. Burleigh and S. Mathews. Phylogenetic signal in nucleotide data from seed plants: implications for resolving the seed plant tree of life. *American Journal of Botany*, 91(10):1599–1613, 10 2004.

[22] A. Camargo, M. Morando, L. J. Avila, and J. W. Sites. Species delimitation with abc and other coalescent-based methods: A test of accuracy with simulations and an empirical example with lizards of the liolaemus darwinii complex (Squamata: Liolaemidae). *Evolution*, 2012.

[23] B. C. Carstens, T. A. Pelletier, N. M. Reid, and J. D. Satler. How to fail at species delimitation, 2013.

[24] W.-C. CHANG, P. GÓRECKI, and O. EULENSTEIN. Exact solutions for species tree

inference from discordant gene trees. *Journal of Bioinformatics and Computational Biology*, 11(05):1342005, 10 2013.

[25] R. Chaudhary, M. S. Bansal, A. Wehe, D. Fernández-Baca, and O. Eulenstein. iGTP: a software package for large-scale gene tree parsimony analysis. *BMC bioinformatics*, 11(1):574, 1 2010.

[26] R. Chaudhary, D. Fernández-Baca, and J. G. Burleigh. Mulrf: a software package for phylogenetic analysis using multi-copy gene trees. *Bioinformatics*, 31(3):432–433, 2014.

[27] J. Chifman and L. S. Kubatko. Quartet Inference from SNP Data Under the Coalescent Model. *Bioinformatics*, 30(23):3317–3324, 8 2014.

[28] S. Christensen, E. Molloy, P. Vachaspati, and T. Warnow. Optimal completion of incomplete gene trees in polynomial time using OCTAL. In *Leibniz International Proceedings in Informatics, LIPIcs*, 2017.

[29] S. Christensen, E. K. Molloy, P. Vachaspati, and T. Warnow. OCTAL: Optimal Completion of gene trees in polynomial time. *Algorithms for Molecular Biology*, 13(1):6, 12 2018.

[30] Y. Chung and C. Ané. Comparing two Bayesian methods for gene tree/species tree reconstruction: simulations with incomplete lineage sorting and horizontal gene transfer. *Systematic biology*, 60(3):261–75, 5 2011.

[31] A. R. Conn, N. I. Gould, and P. L. Toint. *Trust region methods*. SIAM, 2000.

[32] J. A. Coyne and H. A. Orr. *Speciation*. Sinauer Associates Sunderland, MA, 2004.

[33] K. A. Crandall and J. F. Fitzpatrick. Crayfish Molecular Systematics: Using a Combination of Procedures to Estimate Phylogeny. *Systematic Biology*, 45(1):1–26, 3 1996.

[34] G. Dasarathy, R. Nowak, and S. Roch. Data requirement for phylogenetic inference from multiple loci: a new distance method. *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*, 12(2):422–432, 2015.

[35] R. Davidson, M. Lawhorn, J. Rusinko, and N. Weber. Efficient Quartet Representations of Trees and Applications to Supertree and Summary Methods. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 15(3):1010–1015, 5 2018.

[36] N. De Maio, C. Schlötterer, and C. Kosiol. Linking Great Apes Genome Evolution across Time Scales Using Polymorphism-Aware Phylogenetic Models. *Molecular Biology and Evolution*, 30(10):2249–2262, 10 2013.

[37] J. H. Degnan and N. A. Rosenberg. Discordance of Species Trees with Their Most Likely Gene Trees. *PLoS Genetics*, 2(5), 5 2006.

[38] J. H. Degnan and N. A. Rosenberg. Gene tree discordance, phylogenetic inference and the

multispecies coalescent. *Trends in Ecology and Evolution*, 24(6):332–340, 6 2009.

[39] J. H. Degnan and L. A. Salter. Gene tree distributions under the coalescent process. *Evolution*, 59(1):24–37, 1 2005.

[40] F. F. F. Delsuc, H. Brinkmann, and H. H. Philippe. Phylogenomics and the reconstruction of the tree of life. *Nature Reviews Genetics*, 6(5):361–375, 5 2005.

[41] R. Desper and O. Gascuel. Fast and accurate phylogeny reconstruction algorithms based on the minimum-evolution principle. *Journal of computational biology*, 9(5):687–705, 10 2002.

[42] E. Domingo, J. Sheldon, and C. Perales. Viral Quasispecies Evolution. *Microbiology and Molecular Biology Reviews*, 76(2):159–216, 6 2012.

[43] S. V. Edwards. Is a new and general theory of molecular systematics emerging? *Evolution*, 63(1):1–19, 2009.

[44] S. V. Edwards, L. Liu, and D. K. Pearl. High-resolution species trees without concatenation. *Proceedings of the National Academy of Sciences*, 104(14):5936–5941, 2007.

[45] S. V. Edwards, Z. Xi, A. Janke, B. C. Faircloth, J. E. McCormack, T. C. Glenn, B. Zhong, S. Wu, E. M. Lemmon, A. R. Lemmon, A. D. Leaché, L. Liu, and C. C. Davis. Implementing and testing the multispecies coalescent model: A valuable paradigm for phylogenomics. *Molecular Phylogenetics and Evolution*, 94:447–462, 2016.

[46] D. D. Ence and B. C. Carstens. SpedeSTEM: a rapid and accurate method for species delimitation. *Molecular Ecology Resources*, 11(3):473–480, 5 2011.

[47] P. Erdos, M. Steel, L. Szekely, and T. Warnow. A few logs suffice to build (almost) all trees: Part II. *Theoretical Computer Science*, 221(1-2):77–118, 1999.

[48] J. A. Esselstyn, B. J. Evans, J. L. Sedlock, F. A. A. Khan, and L. R. Heaney. Single-locus species delimitation: A test of the mixed yule-coalescent model, with an empirical application to Philippine round-leaf bats. *Proceedings of the Royal Society B: Biological Sciences*, 279(1743):3678–3686, 2012.

[49] J. Felsenstein. Cases in which Parsimony or Compatibility Methods will be Positively Misleading. *Systematic Zoology*, 27(4):401–410, 12 1978.

[50] J. Felsenstein. Evolutionary trees from DNA sequences: A maximum likelihood approach. *Journal of Molecular Evolution*, 17(6):368–376, 11 1981.

[51] W. Fletcher and Z. Yang. INDELible: A flexible simulator of biological sequence evolution. *Molecular Biology and Evolution*, 26(8):1879–1888, 2009.

[52] W. Fletcher and Z. Yang. INDELible: A Flexible Simulator of Biological Sequence

Evolution. *Molecular Biology and Evolution*, 26(8):1879–1888, 8 2009.

[53] T. Fujisawa and T. G. Barraclough. Delimiting species using single-locus data and the generalized mixed yule coalescent approach: a revised method and evaluation on simulated data sets. *Systematic biology*, 62(5):707–724, 2013.

[54] T. C. Giarla and J. A. Esselstyn. The Challenges of Resolving a Rapid, Recent Radiation: Empirical and Simulated Phylogenomics of Philippine Shrews. *Systematic Biology*, 64(5):727–740, 9 2015.

[55] M. T. Hallett and J. Lagergren. New algorithms for the duplication-loss model. In *Proceedings of the fourth annual international conference on Computational molecular biology - RECOMB '00*, pages 138–146, New York, New York, USA, 2000. ACM Press.

[56] P. D. N. Hebert, M. Y. Stoeckle, T. S. Zemlak, and C. M. Francis. Identification of Birds through DNA Barcodes. *PLoS Biology*, 2(10):e312, 9 2004.

[57] J. Heled and A. J. Drummond. Bayesian inference of species trees from multilocus data. *Molecular Biology and Evolution*, 27(3):570–580, 3 2010.

[58] S. Hotaling, M. E. Foley, N. M. Lawrence, J. Bocanegra, M. B. Blanco, R. Rasoloarison, P. M. Kappeler, M. A. Barrett, A. D. Yoder, and D. W. Weisrock. Species discovery and validation in a cryptic radiation of endangered primates: coalescent-based species delimitation in Madagascar's mouse lemurs. *Molecular Ecology*, 25(9):2029–2045, 5 2016.

[59] H. Huang, Q. He, L. S. Kubatko, and L. L. Knowles. Sources of Error Inherent in Species-Tree Estimation: Impact of Mutational and Coalescent Effects on Accuracy and Implications for Choosing among Different Methods. *Systematic Biology*, 59(5):573–583, 10 2010.

[60] L. Hubert and P. Arabie. Comparing partitions. *Journal of classification*, 2(1):193–218, 1985.

[61] R. R. Hudson and J. A. Coyne. Mathematical consequences of the genealogical species concept. *Evolution*, 56(8):1557–1565, 8 2002.

[62] J. P. Huelsenbeck, P. Andolfatto, and E. T. Huelsenbeck. Structurama: Bayesian inference of population structure. *Evolutionary Bioinformatics*, 7:EBO–S6761, 2011.

[63] N. D. Jackson, B. C. Carstens, A. E. Morales, and B. C. O'Meara. Species delimitation with gene flow. *Systematic Biology*, 66(5):799–812, 2017.

[64] E. D. Jarvis, S. Mirarab, A. J. Aberer, B. Li, P. Houde, C. Li, S. Y. W. Ho, B. C. Faircloth, B. Nabholz, J. T. Howard, A. Suh, C. C. Weber, R. R. da Fonseca, J. Li, F. Zhang, H. Li, L. Zhou, N. Narula, L. Liu, G. Ganapathy, B. Boussau, M. S. Bayzid, V. Zavi-

dovych, S. Subramanian, T. Gabaldon, S. Capella-Gutierrez, J. Huerta-Cepas, B. Rekepalli, K. Munch, M. Schierup, B. Lindow, W. C. Warren, D. Ray, R. E. Green, M. W. Bruford, X. Zhan, A. Dixon, S. Li, N. Li, Y. Huang, E. P. Derryberry, M. F. Bertelsen, F. H. Sheldon, R. T. Brumfield, C. V. Mello, P. V. Lovell, M. Wirthlin, M. P. C. Schneider, F. Prosdocimi, J. A. Samaniego, A. M. V. Velazquez, A. Alfaro-Nunez, P. F. Campos, B. Petersen, T. Sicheritz-Ponten, A. Pas, T. Bailey, P. Scofield, M. Bunce, D. M. Lambert, Q. Zhou, P. Perelman, A. C. Driskell, B. Shapiro, Z. Xiong, Y. Zeng, S. Liu, Z. Li, B. Liu, K. Wu, J. Xiao, X. Yinqi, Q. Zheng, Y. Zhang, H. Yang, J. Wang, L. Smeds, F. E. Rheindt, M. Braun, J. Fjeldsa, L. Orlando, F. K. Barker, K. A. Jonsson, W. Johnson, K.-P. Koepfli, S. O'Brien, D. Haussler, O. A. Ryder, C. Rahbek, E. Willerslev, G. R. Graves, T. C. Glenn, J. McCormack, D. Burt, H. Ellegren, P. Alstrom, S. V. Edwards, A. Stamatakis, D. P. Mindell, J. Cracraft, E. L. Braun, T. Warnow, W. Jun, M. T. P. Gilbert, and G. Zhang. Whole-genome analyses resolve early branches in the tree of life of modern birds. *Science*, 346(6215):1320–1331, 12 2014.

[65] O. Jeffroy, H. Brinkmann, F. Delsuc, and H. Philippe. Phylogenomics: the beginning of incongruence? *Trends in Genetics*, 22(4):225–231, 2006.

[66] G. Jones. Algorithmic improvements to species delimitation and phylogeny estimation under the multispecies coalescent. *Journal of Mathematical Biology*, 74(1-2):447–467, 1 2017.

[67] J. F. C. Kingman. On the genealogy of large populations. *Journal of Applied Probability*, 19(1982):27–43, 1982.

[68] E. R. Klein, R. B. Harris, R. N. Fisher, and T. W. Reeder. Biogeographical history and coalescent species delimitation of Pacific island skinks (Squamata: Scincidae: Emoia cyanura species group). *Journal of Biogeography*, 43(10):1917–1929, 10 2016.

[69] L. L. Knowles and B. C. Carstens. Delimiting Species without Monophyletic Gene Trees. *Systematic Biology*, 56(6):887–895, 12 2007.

[70] L. L. Knowles, H. C. Lanier, P. B. Klimov, and Q. He. Full modeling versus summarizing gene-tree uncertainty: Method choice and species-tree accuracy. *Molecular Phylogenetics and Evolution*, 65(2):501–509, 11 2012.

[71] L. S. Kubatko, B. C. Carstens, and L. L. Knowles. STEM: species tree estimation using maximum likelihood for gene trees under coalescence. *Bioinformatics*, 25(7):971–973, 4 2009.

[72] L. S. Kubatko and J. H. Degnan. Inconsistency of phylogenetic estimates from concatenated data under coalescence. *Systematic Biology*, 56:17–24, 2007.

[73] A. Kupczok. Split-based computation of majority-rule supertrees. *BMC Evolutionary Biology*, 11(1):205, 12 2011.

[74] M. Lafond and C. Scornavacca. On the Weighted Quartet Consensus problem. *Theoretical Computer Science*, 769:1–17, 5 2019.

[75] H. C. Lanier and L. L. Knowles. Applying species-tree analyses to deep phylogenetic histories: Challenges and potential suggested from a survey of empirical phylogenetic studies. *Molecular Phylogenetics and Evolution*, 83:191–199, 2 2015.

[76] B. R. Larget, S. K. Kotha, C. N. Dewey, and C. Ané. BUCKy: Gene tree/species tree reconciliation with Bayesian concordance analysis. *Bioinformatics*, 26(22):2910–2911, 11 2010.

[77] N. Lartillot, H. Brinkmann, and H. Philippe. Suppression of long-branch attraction artefacts in the animal phylogeny using a site-heterogeneous model. *BMC evolutionary biology*, 7(1):1–14, 2007.

[78] A. D. Leaché, M. K. Fujita, V. N. Minin, and R. R. Bouckaert. Species Delimitation using Genome-Wide SNP Data. *Systematic Biology*, 63(4):534–542, 7 2014.

[79] A. D. Leaché and B. Rannala. The accuracy of species tree estimation under simulation: A comparison of methods. *Systematic Biology*, 60(2):126–137, 3 2011.

[80] A. D. Leaché, T. Zhu, B. Rannala, and Z. Yang. The Spectre of Too Many Species. *Systematic Biology*, 68(1):168–181, 1 2019.

[81] L. Liu. BEST: Bayesian estimation of species trees under the coalescent model. *Bioinformatics*, 24(21):2542–2543, 11 2008.

[82] L. Liu and L. Yu. Estimating Species Trees from Unrooted Gene Trees. *Systematic Biology*, 60(5):661–667, 10 2011.

[83] L. Liu, L. Yu, and S. V. Edwards. A maximum pseudo-likelihood approach for estimating species trees under the coalescent model. *BMC Evolutionary Biology*, 10(1):302, 2010.

[84] L. Liu, L. Yu, D. K. Pearl, and S. V. Edwards. Estimating species phylogenies using coalescence times among sequences. *Systematic Biology*, 58(5):468–477, 10 2009.

[85] W. P. Maddison. Gene Trees in Species Trees. *Systematic Biology*, 46(3):523–536, 9 1997.

[86] W. P. Maddison and L. L. Knowles. Inferring Phylogeny Despite Incomplete Lineage Sorting. *Systematic Biology*, 55(1):21–30, 2 2006.

[87] U. Mai and S. Mirarab. TreeShrink: Efficient Detection of Outlier Tree Leaves. In J. Meidanis and L. Nakhleh, editors, *Comparative Genomics: 15th International Workshop, RECOMB CG 2017, Barcelona, Spain, October 4-6, 2017, Proceedings*, pages 116–140. Springer International Publishing, Cham, 2017.

[88] D. Mallo, L. de Oliveira Martins, and D. Posada. Simphy: phylogenomic simulation of

gene, locus, and species trees. *Systematic biology*, 65(2):334–344, 2015.

[89] D. Mallo, L. De Oliveira Martins, and D. Posada. SimPhy: Phylogenomic Simulation of Gene, Locus, and Species Trees. *Systematic biology*, 65(2):334–44, 3 2016.

[90] F. A. Matsen, R. B. Kodner, and E. V. Armbrust. pplacer: linear time maximum-likelihood and Bayesian phylogenetic placement of sequences onto a fixed reference tree. *BMC bioinformatics*, 11(1):538, 10 2010.

[91] J. E. McCormack, H. Huang, and L. L. Knowles. Maximum likelihood estimates of species trees: How accuracy of phylogenetic inference depends upon the divergence history and sampling design. *Systematic Biology*, 58(5):501–508, 2009.

[92] M. Medina, A. G. Collins, J. D. Silberman, and M. L. Sogin. Evaluating hypotheses of basal animal phylogeny using complete sequences of large and small subunit rRNA. *Proceedings of the National Academy of Sciences*, 98(17):9707–9712, 8 2001.

[93] B. Q. Minh, H. A. Schmidt, O. Chernomor, D. Schrempf, M. D. Woodhams, A. Von Haeseler, and R. Lanfear. Iq-tree 2: new models and efficient methods for phylogenetic inference in the genomic era. *Molecular biology and evolution*, 37(5):1530–1534, 2020.

[94] S. Mirarab. Novel scalable approaches for multiple sequence alignment and phylogenomic reconstruction, 2015.

[95] S. Mirarab. Species Tree Estimation Using ASTRAL: Practical Considerations. *Arxiv preprint*, 1904.03826, 4 2019.

[96] S. Mirarab, M. S. Bayzid, B. Boussau, and T. Warnow. Statistical binning enables an accurate coalescent-based estimation of the avian tree. *Science*, 346(6215):1250463–1250463, 12 2014.

[97] S. Mirarab, L. Nakhleh, and T. Warnow. Multispecies Coalescent: Theory and Applications in Phylogenetics. *Annual Review of Ecology, Evolution, and Systematics*, 52(1):247–268, 11 2021.

[98] S. Mirarab, N. Nguyen, and T. Warnow. SEPP: SATé-Enabled Phylogenetic Placement. In *Pacific Symposium On Biocomputing*, pages 247–58. WORLD SCIENTIFIC, 12 2012.

[99] S. Mirarab, R. Reaz, M. S. Bayzid, T. Zimmermann, M. S. Swenson, and T. Warnow. ASTRAL: genome-scale coalescent-based species tree estimation. *Bioinformatics*, 30(17):i541–i548, 9 2014.

[100] S. Mirarab and T. Warnow. ASTRAL-II: Coalescent-based species tree estimation with many hundreds of taxa and thousands of genes. *Bioinformatics*, 31(12):i44–i52, 6 2015.

[101] B. Misof, S. Liu, K. Meusemann, R. S. Peters, A. Donath, C. Mayer, P. B. Frandsen, J. Ware, T. Flouri, R. G. Beutel, O. Niehuis, M. Petersen, F. Izquierdo-Carrasco, T. Wappler, J. Rust,

Y. Li, X. Xu, Z. Yong, H. Yang, J. Wang, K. M. Kjer, and X. Zhou. Phylogenomics resolves the timing and pattern of insect evolution. *Science*, 346(6210):763–767, 2014.

[102] N. Mitchell, P. O. Lewis, E. M. Lemmon, A. R. Lemmon, and K. E. Holsinger. Anchored phylogenomics improves the resolution of evolutionary relationships in the rapid radiation of protea L. *American Journal of Botany*, 104(1):102–115, 2017.

[103] N. Mitchell, P. O. Lewis, E. M. Lemmon, A. R. Lemmon, and K. E. Holsinger. Anchored phylogenomics improves the resolution of evolutionary relationships in the rapid radiation of protea l. *American Journal of Botany*, 104(1):102–115, 2017.

[104] E. K. Molloy and T. Warnow. To Include or Not to Include: The Impact of Gene Filtering on Species Tree Estimation Methods. *Systematic Biology*, 67(2):285–303, 3 2018.

[105] E. Mossel and S. Roch. Incomplete lineage sorting: consistent phylogeny estimation from multiple loci. *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*, 7(1):166–171, 1 2010.

[106] L. J. Musher and J. Cracraft. Phylogenomics and species delimitation of a complex radiation of Neotropical suboscine birds (Pachyramphus). *Molecular Phylogenetics and Evolution*, 118(April 2017):204–221, 1 2018.

[107] N.-p. Nguyen, S. Mirarab, B. Liu, M. Pop, and T. Warnow. TIPP: taxonomic identification and phylogenetic profiling. *Bioinformatics*, 30(24):3548–3555, 12 2014.

[108] H. A. Ogilvie, R. R. Bouckaert, and A. J. Drummond. Starbeast2 brings faster species tree inference and accurate estimates of substitution rates. *Molecular biology and evolution*, 34(8):2101–2114, 2017.

[109] H. A. Ogilvie, R. R. Bouckaert, and A. J. Drummond. StarBEAST2 Brings Faster Species Tree Inference and Accurate Estimates of Substitution Rates. *Molecular Biology and Evolution*, 34(8):2101–2114, 8 2017.

[110] H. A. Ogilvie, J. Heled, D. Xie, and A. J. Drummond. Computational Performance and Statistical Accuracy of *BEAST and Comparisons with Other Methods. *Systematic Biology*, 65(3):381–396, 5 2016.

[111] M. Olave, E. Solà, and L. L. Knowles. Upstream Analyses Create Problems with DNA-Based Species Delimitation. *Systematic Biology*, 63(2):263–271, 3 2014.

[112] E. F. Oliveira, M. Gehara, V. A. São-Pedro, X. Chen, E. A. Myers, F. T. Burbrink, D. O. Mesquita, A. A. Garda, G. R. Colli, M. T. Rodrigues, F. J. Arias, H. Zaher, R. M. Santos, and G. C. Costa. Speciation with gene flow in whiptail lizards from a neotropical xeric biome. *Molecular Ecology*, 24(23):5957–5975, 2015.

[113] B. C. O'Meara. New Heuristic Methods for Joint Species Delimitation and Species Tree

Inference. *Systematic Biology*, 59(1):59–73, 1 2010.

[114] O. T. P. T. OneKP Initiative. One thousand plant transcriptomes and the phylogenomics of green plants. *Nature*, 574(7780):679–685, 10 2019.

[115] P. Pamilo and M. Nei. Relationships between gene trees and species trees. *Molecular biology and evolution*, 5(5):568–583, 1988.

[116] S. Patel. Error in Phylogenetic Estimation for Bushes in the Tree of Life. *Journal of Phylogenetics & Evolutionary Biology*, 01(02):110, 2013.

[117] P. J. Planet. Tree disagreement: measuring and testing incongruence in phylogenies. *Journal of biomedical informatics*, 39(1):86–102, 2 2006.

[118] S. Poe and A. L. Chubb. Birds in a Bush : Five Genes Indicate Explosive Evolution of Avian Orders. *Evolution*, 58(2):404–415, 2004.

[119] J. Pons, T. G. Barraclough, J. Gomez-Zurita, A. Cardoso, D. P. Duran, S. Hazell, S. Kamoun, W. D. Sumlin, and A. P. Vogler. Sequence-Based Species Delimitation for the DNA Taxonomy of Undescribed Insects. *Systematic Biology*, 55(4):595–609, 8 2006.

[120] M. N. Price, P. S. Dehal, and A. P. Arkin. FastTree-2 – Approximately Maximum-Likelihood Trees for Large Alignments. *PLoS ONE*, 5(3):e9490, 3 2010.

[121] M. N. Price, P. S. Dehal, and A. P. Arkin. FastTree 2 – Approximately Maximum-Likelihood Trees for Large Alignments. *PLoS ONE*, 5(3), 3 2010.

[122] N. Puillandre, A. Lambert, S. Brouillet, and G. ACHAZ. Abgd, automatic barcode gap discovery for primary species delimitation. *Molecular ecology*, 21(8):1864–1877, 2012.

[123] M. Rabiee, E. Sayyari, and S. Mirarab. Multi-allele species reconstruction using ASTRAL. *Molecular Phylogenetics and Evolution*, 130:286–296, 1 2019.

[124] W. M. Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical association*, 66(336):846–850, 1971.

[125] B. Rannala, S. V. S. V. Edwards, A. Leaché, and Z. Yang. The Multi-species Coalescent Model and Species Tree Inference. In C. Scornavacca, F. Delsuc, and N. Galtier, editors, *Phylogenetics in the Genomic Era*, pages 3.3:1–3.3:21. No commercial publisher — Authors open access book, 2020.

[126] B. Rannala and Z. Yang. Bayes estimation of species divergence times and ancestral population sizes using DNA sequences from multiple loci. *Genetics*, 164(4):1645–1656, 2003.

[127] B. Rannala and Z. Yang. Species delimitation, 2020.

[128] D. Robinson and L. Foulds. Comparison of phylogenetic trees. *Mathematical Biosciences*, 53(1-2):131–147, 1981.

[129] D. F. Robinson and L. R. Foulds. Comparison of phylogenetic trees. *Mathematical biosciences*, 53(1-2):131–147, 1981.

[130] S. Roch, M. Nute, and T. Warnow. Long-Branch Attraction in Species Tree Estimation: Inconsistency of Partitioned Likelihood and Topology-Based Summary Methods. *Systematic Biology*, 68(2):281–297, 3 2019.

[131] S. Roch and M. Steel. Likelihood-based tree reconstruction on a concatenation of aligned sequence data sets can be statistically inconsistent. *Theoretical Population Biology*, 100:56–62, 3 2015.

[132] A. Rokas, B. L. Williams, N. King, and S. B. Carroll. Genome-scale approaches to resolving incongruence in molecular phylogenies. *Nature*, 425(6960):798–804, 10 2003.

[133] F. Ronquist, S. Klopfstein, L. Vilhelmsen, S. Schulmeister, D. L. Murray, and A. P. Rasnitsyn. A total-evidence approach to dating with fossils, applied to the early radiation of the hymenoptera. *Systematic biology*, 61(6):973–99, 12 2012.

[134] S. Ruane, R. W. Bryson, R. A. Pyron, and F. T. Burbrink. Coalescent Species Delimitation in Milksnakes (Genus Lampropeltis) and Impacts on Phylogenetic Comparative Analyses. *Systematic Biology*, 63(2):231–250, 12 2013.

[135] E. Sayyari and S. Mirarab. Anchoring quartet-based phylogenetic distances and applications to species tree reconstruction. *BMC Genomics*, 17(S10):101–113, 11 2016.

[136] E. Sayyari and S. Mirarab. Fast Coalescent-Based Computation of Local Branch Support from Quartet Frequencies. *Molecular Biology and Evolution*, 33(7):1654–1668, 7 2016.

[137] E. Sayyari and S. Mirarab. Testing for Polytomies in Phylogenetic Species Trees Using Quartet Frequencies. *Genes*, 9(3):132, 2 2018.

[138] E. Sayyari, J. B. Whitfield, and S. Mirarab. Fragmentary Gene Sequences Negatively Impact Gene Tree and Species Tree Reconstruction. *Molecular Biology and Evolution*, 34(12):3279–3291, 12 2017.

[139] B. Schieber and U. Vishkin. On finding lowest common ancestors: Simplification and parallelization. In *VLSI Algorithms and Architectures*, pages 111–123. Springer-Verlag, Berlin/Heidelberg, 1988.

[140] A. N. Schmidt-Lebuhn and K. J. Smith. From the desert it came: evolution of the Australian paper daisy genus Leucochrysum (Asteraceae, Gnaphalieae). *Australian Systematic Botany*, 29(3):176–184, 2016.

[141] D. Schrempf, B. Q. Minh, N. De Maio, A. von Haeseler, and C. Kosiol. Reversible

polymorphism-aware phylogenetic models and their application to tree inference. *Journal of Theoretical Biology*, 407:362–370, 10 2016.

[142] S. Singhal, C. J. Hoskin, P. Couper, S. Potter, and C. Moritz. A framework for resolving cryptic species: a case study from the lizards of the australian wet tropics. *Systematic Biology*, 67(6):1061–1075, 2018.

[143] B. T. Smith, M. G. Harvey, B. C. Faircloth, T. C. Glenn, and R. T. Brumfield. Target capture and massively parallel sequencing of ultraconserved elements for comparative studies at shallow evolutionary time scales. *Systematic Biology*, 63(1):83–95, 2014.

[144] S. Snir, T. Warnow, and S. Rao. Short Quartet Puzzling: A New Quartet-Based Phylogeny Reconstruction Algorithm. *Journal of Computational Biology*, 15(1):91–103, 1 2008.

[145] C. Solís-Lemus, L. L. Knowles, and C. Ané. Bayesian species delimitation combining multiple genes and traits in a unified framework. *Evolution*, 69(2):492–507, 2 2015.

[146] A. Stamatakis. RAxML version 8: A tool for phylogenetic analysis and post-analysis of large phylogenies. *Bioinformatics*, 30(9):1312–1313, 2014.

[147] J. Sukumaran and M. T. Holder. DendroPy: a Python library for phylogenetic computing. *Bioinformatics*, 26(12):1569–1571, 2010.

[148] J. Sukumaran and L. L. Knowles. Multispecies coalescent delimits structure, not species. *Proceedings of the National Academy of Sciences*, 114(7):1607–1612, 2 2017.

[149] D. L. Swofford. Paup*: Phylogenetic analysis using parsimony (and other methods) 4.0. b5. 2001.

[150] G. J. Szöllõsi, W. Rosikiewicz, B. Boussau, E. Tannier, and V. Daubin. Efficient exploration of the space of reconciled gene trees. *Systematic biology*, 62(6):901–12, 11 2013.

[151] G. J. Szöllõsi, E. Tannier, V. Daubin, and B. Boussau. The inference of gene trees with species trees. *Systematic Biology*, 64(1):e42–e62, 7 2014.

[152] N. Takahata. Gene genealogy in three related populations: consistency probability between gene and population trees. *Genetics*, 122(4):957–966, 1989.

[153] J. E. Tarver, M. dos Reis, S. Mirarab, R. J. Moran, S. Parker, J. E. O'Reilly, B. L. King, M. J. O'Connell, R. J. Asher, T. Warnow, K. J. Peterson, P. C. Donoghue, and D. Pisani. The Interrelationships of Placental Mammals and the Limits of Phylogenetic Inference. *Genome Biology and Evolution*, 8(2):330–344, 2 2016.

[154] C. Than and L. Nakhleh. Species Tree Inference by Minimizing Deep Coalescences. *PLoS Computational Biology*, 5(9):e1000501, 9 2009.

[155] A. Töpfer, T. Marschall, R. A. Bull, F. Luciani, A. Schönhuth, and N. Beerenwinkel. Viral

Quasispecies Assembly via Maximal Clique Enumeration. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, pages 309–310. 2014.

[156] P. Vachaspati and T. Warnow. ASTRID: Accurate Species TRees from Internode Distances. *BMC Genomics*, 16(Suppl 10):S3, 2015.

[157] P. Vachaspati and T. Warnow. FastRFS: fast and accurate Robinson-Foulds Supertrees using constrained exact optimization. *Bioinformatics*, 33(5):631–639, 9 2017.

[158] P. Vachaspati and T. Warnow. SVDquest: Improving SVDquartets species tree estimation using exact optimization within a constrained search space. *Molecular Phylogenetics and Evolution*, 124:122–136, 7 2018.

[159] Y. Wang and L. Nakhleh. Towards an accurate and efficient heuristic for species/gene tree co-estimation. *Bioinformatics*, 34(17):i697–i705, 9 2018.

[160] T. Warnow. Textbook for 394C : Algorithms for Computational Biology.

[161] N. J. Wickett, S. Mirarab, N. Nguyen, T. Warnow, E. J. Carpenter, N. Matasci, S. Ayyampalayam, M. S. Barker, J. G. Burleigh, M. A. Gitzendanner, B. R. Ruhfel, E. Wafula, J. P. Der, S. W. Graham, S. Mathews, M. Melkonian, D. E. Soltis, P. S. Soltis, N. W. Miles, C. J. Rothfels, L. Pokorny, A. J. Shaw, L. DeGironimo, D. W. Stevenson, B. Surek, J. C. Villarreal, B. Roure, H. Philippe, C. W. DePamphilis, T. Chen, M. K. Deyholos, R. S. Baucom, T. M. Kutchan, M. M. Augustin, J. J. Wang, Y. Zhang, Z. Tian, Z. Yan, X. Wu, X. Sun, G. K.-S. Wong, and J. J. Leebens-Mack. Phylotranscriptomic analysis of the origin and early diversification of land plants. *Proceedings of the National Academy of Sciences*, 111(45):4859–4868, 10 2014.

[162] S. N. Wood. Minimizing model fitting objectives that contain spurious local minima by bootstrap restarting. *Biometrics*, 2001.

[163] Y. Wu. Coalescent-based species tree inference from gene tree topologies under incomplete lineage sorting by maximum likelihood. *Evolution*, 66(3):763–775, 2012.

[164] B. Xu and Z. Yang. Challenges in species tree estimation under the multispecies coalescent model. *Genetics*, 204(4):1353–1368, 2016.

[165] J. Yang and T. Warnow. Fast and accurate methods for phylogenomic analyses. *BMC Bioinformatics*, 12(Suppl 9):S4, 2011.

[166] Z. Yang and B. Rannala. Bayesian species delimitation using multilocus sequence data. *Proceedings of the National Academy of Sciences*, 107(20):9264–9269, 5 2010.

[167] Z. Yang and B. Rannala. Unguided species delimitation using dna sequence data from multiple loci. *Molecular Biology and Evolution*, 31(12):3125–3135, 2014.

[168] Z. Yang and B. Rannala. Unguided Species Delimitation Using DNA Sequence Data from Multiple Loci. *Molecular Biology and Evolution*, 31(12):3125–3135, 12 2014.

[169] C. Zhang, M. Rabiee, E. Sayyari, and S. Mirarab. ASTRAL-III: polynomial time species tree reconstruction from partially resolved gene trees. *BMC Bioinformatics*, 19(S6):153, 5 2018.

[170] C. Zhang, D.-X. Zhang, T. Zhu, and Z. Yang. Evaluation of a Bayesian Coalescent Method of Species Delimitation. *Systematic Biology*, 60(6):747–761, 12 2011.

[171] J. Zhang, P. Kapli, P. Pavlidis, and A. Stamatakis. A general species delimitation method with applications to phylogenetic placements. *Bioinformatics*, 29(22):2869–2876, 2013.

[172] L. Zhang and Y. Cui. An efficient method for DNA-based species assignment via gene tree and species tree reconciliation. In *International Workshop on Algorithms in Bioinformatics*, pages 300–311. Springer, 2010.

[173] T. Zimmermann, S. Mirarab, and T. Warnow. BBCA: Improving the scalability of *BEAST using random binning. *BMC genomics*, 15(Suppl 6):S11, 10 2014.