The Dissertation Committee for Siavash Mir arabbaygi
certifies that this is the approved version of the following dissertation:

Novel scalable approaches for multiple sequence alignment and phylogenomic reconstruction

Committee:

Keshav Pingali, Supervisor

Tandy Warnow, Co-Supervisor

David Hillis

Bonnie Berger

Joydeep Ghosh

Ray Mooney

Novel scalable approaches for multiple sequence alignment and phylogenomic reconstruction

by

Siavash Mir arabbaygi, B.S.; M. APPL S.

DISSERTATION

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT AUSTIN

August 2015

Dedicated to my mother, and the memory of my father.

Acknowledgments

I wish to thank the countless people who helped me throughout my PhD. I enjoyed working with many collaborators, most of whom I never met, and not all of whom I can name here. I had the good fortune of working with my labmates, especially Nam Nguyen, Shamsuzzoha Md. Bayzid, and Théo Zimmermann, and our interactions have always been smooth and fruitful. I benefited tremendously from my involvement in both the avian phylogenomics and the thousand transcripts projects, both of which included many top researchers from across the world. I want to especially thank Bastien Boussau, Edward Braun, Tom Gilbert, Erich Jarvis, Jim Leebens-Mack, Gane Wong, Norman Wickett, and Guojie Zhang. I also thank Kevin Liu, Mark Holder, and Jeet Sukumaran for allowing me to use their code throughout my work. I would also like to thank all members of my committee. My understanding of the biology was tremendously improved by participation in discussions lead by Prof. David Hillis. I am also grateful to Prof. Keshav Pingali, who accepted me in his group after my original supervisor officialy left UT; Keshav was a role model, and involvement with his group has greatly expanded my depth. I should also thank my sources of funding (Howard Hughes Medical Institute international student fellowship, Canadian NSERC Doctoral award, and NSF) and the generous computational support of Texas Advanced Computing Center (TACC) and the computing cluster (Condor) at the CS department.

I cannot start to thank enough my supervisor, Prof. Tandy Warnow, for all her guidance and her support. I enjoyed the chance to argue with her frequently on all matters related to our research, small or large, and even though I often found myself on the loosing side of these arguments, I always felt encouraged. Nothing like her openness and acceptance in rare occasions I won an argument could have better trained me for the skepticism and inquisitiveness necessary for the scientific pursuit. I can only inspire to match her enthusiasm for research, which was a constant boost of energy to me throughout. Her endless support as a research advisor and a career mentor has opened to me many new doors, which I did not believe accessible when I started my studies. I owe much of my past and future achievements to her.

For all the support I had at work, no progress was possible if it wasn't for the persistent emotional support my fiancee, now wife, showered me with. She raised my spirits with expressions of confidence in my abilities, warranted or not. Her confidence in me was a constant motivation to overcome obstacles. And that's to say nothing of all the day-to-day ways in which she has nudged me towards being a more organized and focused student. I would have never been in the office early in the morning every day of the week, if it wasn't for her. She was a major reason I started my PhD studies, and her encouragements are a major reason I feel confident continuing as a researchers.

Finally, everything I have ever achieved has been possible only because of my mother, who thought me starting is easy, persevering hard, and letting go the hardest. Her example is always with me.

Novel scalable approaches for multiple sequence alignment and phylogenomic reconstruction

Publication No. ____________________

Siavash Mir arabbaygi, Ph.D.
The University of Texas at Austin, 2015

Supervisors: Keshav Pingali
Tandy Warnow

The amount of biological sequence data is increasing rapidly, a promising development that would transform biology if we can develop methods that can analyze large-scale data efficiently and accurately. A fundamental question in evolutionary biology is building the tree of life: a reconstruction of relationships between organisms in evolutionary time. Reconstructing phylogenetic trees from molecular data is an optimization problem that involves many steps. In this dissertation, we argue that to answer long-standing phylogenetic questions with large-scale data, several challenges need to be addressed in various steps of the pipeline. One challenges is aligning large number of sequences so that evolutionarily related positions in all sequences are put in the same column. Constructing alignments is necessary for phylogenetic reconstruction, but also for many other types of evolutionary analyses. In response to this challenge, we introduce PASTA, a scalable and accurate algorithm that

can align datasets with up to a million sequences. A second challenge is related to the interesting fact that various parts of the genome can have different evolutionary histories. Reconstructing a species tree from genome-scale data needs to account for these differences. A main approach for species tree reconstruction is to first reconstruct a set of “gene trees” from different parts of the genome, and to then summarize these gene trees into a single species tree. We argue that this approach can suffer from two challenges: reconstruction of individual gene trees from limited data can be plagued by estimation error, which translates to errors in the species tree, and also, methods that summarize gene trees are not scalable or accurate enough under some conditions. To address the first challenge, we introduce statistical binning, a method that re-estimates gene trees by grouping them into bins. We show that binning improves gene tree accuracy, and consequently the species tree accuracy. To address the second challenge, we introduce ASTRAL, a new summary method that can run on a thousand genes and a thousand species in a day and has outstanding accuracy. We show that the development of these methods has enabled biological analyses that were otherwise not possible.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Evolution is the mechanism that has generated the diversity of life we observe today on earth [1, 2], likely starting from a single common ancestor billions of years ago and generating new species through a branching process [3, 4]. Evolutionary histories of organisms are studied using phylogenetic trees [5]. A phylogeny is a tree that traces the evolution of a set of organisms, or certain characters derived from those organisms, through evolutionary time. The nodes of a phylogeny can represent entire populations of a species, in which case the phylogeny is called a *species tree* and its branching structure shows how new species have evolved from now-extinct species. Species can split into two species for various reasons [6], but at the molecular level, the driving force behind the evolution of new species is the constant process of mutations accumulating in the DNA and across the genomes.

Since evolution happens at the genomic level, one can reconstruct the evolutionary history from molecular sequences [5]. For example, DNA can be represented as a sequence of `A, C, G,` and `T` letters for each species of interest. Given these sequences, the goal is to find the phylogenetic tree that best explains the observed DNA sequences. Phylogeny reconstruction from molecular data has been studied as an optimization problem and particularly as a statistical inference problem for decades [5, 7–9]. Most problems in phylogenetics are NP-hard, but nevertheless, heuristic and approximate approaches have been developed to solve these optimization problems, and these approaches have been extensively used to reconstruct various parts of the tree of life [10].

Recent drops in sequencing costs [11] have lead to a rapid growth in the size of the datasets that we are interested in analyzing. The datasets used for phylogenetic reconstruction are increasing in two dimensions: on the one hand, we are gathering molecular sequence data from more species, and on the other hand, we are sequencing larger parts of the genomes of these species. The increase in the dataset size would ideally result in an increased ability to resolve hard phylogenetic questions [12, 13], and for a large set of organisms. However, the sheer size of the datasets creates many computational challenges [14, 15] and prevents some types of analyses [16, 17]. More importantly, it is not clear that methods developed for smaller datasets have good accuracy on larger dataset, or that existing methods are able to use larger datasets effectively; we will argue throughout this dissertation that analyzing large datasets requires new methods.

The increase in the number of genomic regions analyzed is especially interesting. Phylogenies can be reconstructed from various parts of the genome. When a phylogenies reflects the evolutionary history of a particular part of the genome, it is called a *gene tree*, as opposed to the species tree that reflects the genome evolution as a whole. An interesting biological fact is that the evolutionary histories of different parts of the genome can be different from one another [18–20]. Thus, gene trees need not agree with each other, or with the species tree. As an example, the closest relatives of humans are chimpanzees and gorillas, which each share with us about 95% of their genomes [21, 22]. At the species level, chimpanzees and humans are closer to each other than either is to gorilla [23]. However, for about 20% of the genome, gorillas are closer to either human or chimpanzees than those two are to each other [23–25]. This interesting opportunity for gene trees and species trees to be discordant can be due to various biological mechanisms, as we will discuss in Chapter 2 (which gives background information about various topics in this dissertation). However, it's important to note that gene tree discordance has implications for reconstructing the species tree. For situations where gene tree discordance is likely, to be able to reconstruct the species tree, we need to analyze large parts of the genome.

A phylogenetic analysis of sequence data from multiple genes requires a series of steps, shown in Figure 1.1. Samples are gathered from species of interest and various bioinformatic processing steps are used to generate the sequence data for a collection of regions in the genome, each of which we simply call a *gene*. Once these sequence data are gathered for all the genes of interest, two basic steps are necessary.

Step 1: Multiple Sequence Alignment (MSA): Sequences belonging to various species but the same gene can have different lengths, an issue we de-

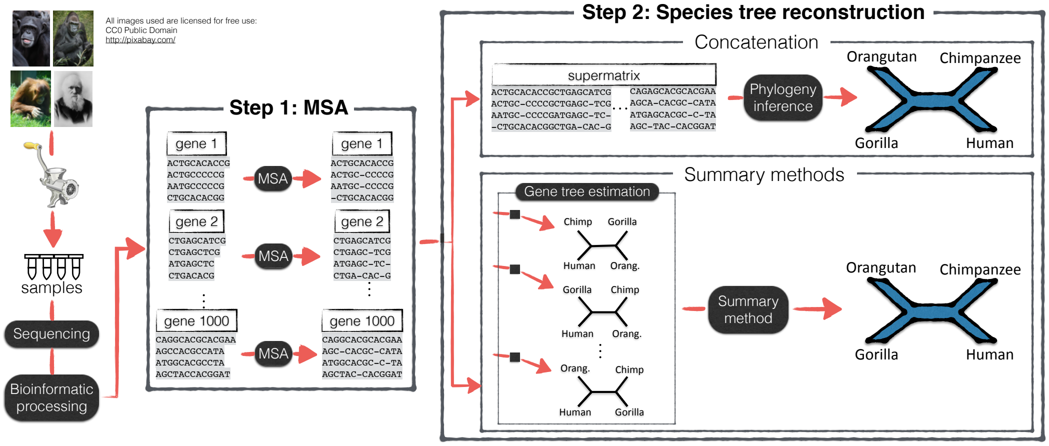


Figure 1.1: **Phylogenetic reconstruction from multiple genes**. The phylogenetic reconstruction pipeline starts by gathering samples from the species of interest, and sequencing samples. Various bioinformatics tools are used to assemble the sequence data and to extract sequence data for each gene. Then, two basic steps are needed: Step 1: Sequence data for each gene need to be aligned using a Multiple Sequence Alignment (MSA) tool. Step 2: A species phylogeny is reconstructed from the aligned gene sequence data. To build phylogenies from multiple genes, various approaches exist; two such approaches are shown here. In the concatenation approach, all gene data are concatenated into one supermatrix, which is then analyzed using a phylogenetic reconstruction method of choice. In the summary method approach, first, a separate tree is estimated for each gene, and then this set of estimated gene trees is used as input to a method called a summary method that produces a species tree. Phylogenies are shown as unrooted trees. See Chapter 2 for details of all the steps.

scribe in detail in Chapter 2. To be able to reconstruct phylogenies from these unaligned data using most methods, we first need to align them so that they all have the same length; the result is called a Multiple Sequence Alignment (MSA) of the data. Constructing an MSA is formulated as various optimization problems, and these problems are also NP-hard according to various formulations of the problem [26, 27]; however, various tools exist for computing MSAs heuristically [28, 29].

Step 2: Species tree reconstruction: Once alignments are obtained, a species tree can be obtained using various techniques, which we describe in detail in Chapter 2. Two of these approaches that can analyze large datasets are concatenation and summary methods. Concatenation puts all the gene data together in one "supermatrix" and infers a tree from this supermatrix using a phylogenetic reconstruction method of choice. Using summary methods involves two steps: first we need to estimate a gene tree for each gene separately using a phylogenetic reconstruction tool of choice, and then we need to estimate the species tree by summarizing the collection of gene trees. The summarization step requires a "summary method" that takes as input a set of gene trees and outputs a species trees that best explains that collection of gene trees. Thus, the input to summary methods is a set of estimated gene trees, and these gene trees can have estimation error.

In this dissertation, we show that with increased dataset size, methods that exist for both steps of the pipeline described above face substantial challenges. To address these challenges, we develop new methods for both steps of the pipeline, and we show that our methods enable accurate analyses of datasets with large numbers of species and large numbers of genes.

When the number of sequences being analyzed increases to many thousands or even a million sequences, MSA construction tools are either not able to run, or have reduced accuracy when they do run [15]. Yet, such analyses are being tried and analyzing many thousands of species is gradually becoming the norm. In Chapter 3, we introduce a new multiple sequence alignment algorithm called PASTA for co-estimation of alignments and trees using an iterative approach (PASTA is an extension of an earlier tool called SATé [30, 31]). PASTA can quickly and accurately analyze very large datasets, and achieves this using divide-and-conquer in conjunction with a new approach we introduce for merging a collection of sub-alignments. For a dataset with a million sequences, PASTA is able to generate highly accurate alignments in about two weeks using only 12 threads.

Once the alignments are obtained, concatenation or summary methods can be used for estimating the species tree, and the relative advantages of these two approaches are hotly debated [32–35]. Concatenation has the advantage of using all the data in one analysis, but it ignores gene tree discordance. Summary methods can take into account discordance, but since they have to go through two steps (gene tree estimation and summarization), they can be sensitive to errors introduced in the gene tree estimation step [36–38]. Moreover, summary methods are newer, and less research has focused on developing accurate and scalable summary methods. In this dissertation, we propose improvements to both steps of the summary method pipeline (i.e., gene tree estimation, and summarization) and show that comparison between concatenation and summary methods depend on the summary method pipeline used, in addition to other properties of the dataset being analyzed (e.g., its level of gene tree discordance).

In Chapter 4, we introduce the statistical binning pipeline for re-estimating gene trees with the goal of improving their accuracy. We first give evidence that error in the gene tree estimation step can translate into error in the species tree, and then present one potential solution: statistical binning. Our proposed approach divides the set of genes into non-overlapping groups, called bins, such that in each bin there is no strong evidence of discordance between pairs of gene trees. For each bin, we then concatenate sequences of all genes put in that bin and estimate a tree based on this "supergene" matrix; this produces a set of "supergene" trees, which we can then use as input to the summary method. We show in extensive simulation and biological studies that this approach can increase the accuracy of the summary method pipeline. Interestingly, binning can make summary methods more accurate than concatenation under some conditions where concatenation is more accurate than summary methods without binning. We introduce two variants of the statistical binning approach, with similar accuracy in experimental studies, but different theoretical statistical properties, which we will prove.

In Chapter 5, we target the second part of the two step summary method approach, and show that existing summary methods have important shortcomings in terms of analyzing large numbers of species. We propose a new summary method called ASTRAL, which has statistical guarantees of convergence to the correct species tree as the number of error-free input gene trees are increased, assuming specific causes of gene tree discordance. ASTRAL is based on a likely NP-hard optimization problem and uses dynamic programming to solve the problem exactly in exponential time, or for datasets of moderate to large size, to solve a constrained version of the problem in polynomial time. We introduce two version of ASTRAL, and show that the second version, ASTRAL-II, can run on datasets with a thousand species and a thousand genes in about 24 hours of running time, with only one thread. No other summary method we tested could analyze datasets of this size given reasonable running time limits. We show that ASTRAL has better accuracy than competing summary methods on smaller dataset where these methods can run.

The set of new methods we introduce in this dissertation, PASTA, statistical binning, and ASTRAL, enable accurate analyses of large datasets that could not be analyzed without these methods. We were motivated to develop these methods by our involvement in two large-scale phylogenetic projects that used sequence data from across the gnomes of two different sets of organisms (these studies are referred to as phylogenomics or phylotranscriptomics depending on the technology used for extracting data): birds and plants.

The avian phylogenomics project obtained sequences for the entire genomes of 48 different bird species [39] with the goal of estimating the bird phylogeny. Major challenges were presented to the team of scientists analyzing this data, among them, the fact that gene tree discordance was rampant. To account for gene tree discordance, it was clear that methods that take it into account need to be used to analyze the data. However, in addition to true biological gene tree discordance, gene tree estimation error was also rampant in the dataset, and the rampant gene tree error limited the ability of existing summary method pipelines for analyzing this dataset accurately. The traditional summary method pipeline did not produce highly supported or believable trees using the entire set of data, and could only accurately analyze subsets of the data. In response to this shortcoming, we developed the statistical binning pipeline, and were able to produce a highly resolved species tree that accounted for gene tree discordance; the tree produced by the statistical binning pipeline was presented as one of the two hypotheses of the bird phylogeny in the final paper published on this dataset [39].

The one thousand plant transcriptomes (1KP) project, has the goal of analyzing more than 1000 plant species, but focusing only on parts of the genome that code for proteins (transcriptome). The initial phase of the project included 103 plant species [40], which is still larger than similar multi-gene datasets. Gene tree estimation error was less rampant on this dataset; but the sheer size of the dataset and the evolutionary span (close to a billion year) made application of existing summary methods challenging. Existing methods could analyze only subsets of genes, and did not produce believable trees on the subsets of genes they could analyze. In response, we developed ASTRAL, which was able to analyze the dataset and produce a highly supported and believable tree. The ASTRAL tree is presented as one of the main hypotheses of plant evolution in the respective paper [40]. Our new version, ASTRAL-II, can handle the dataset that is currently being produced in the second phase of 1KP, which includes more than 1000 plant species and several hundred genes.

In summary, we show that phylogenetic analyses of large datasets requires new methods because existing methods tend to be either unable to run on large datasets, or when they run they do not show the accuracy that they could obtain on smaller datasets. In this dissertation, we identify three related areas were large datasets cannot be accurately analyzed with existing techniques and we develop new methods that enable analyzing these datasets accurately, and with reasonable running times. We prove theoretical guarantees of accuracy, give theoretical bounds for running time, and present extensive experimental studies that evaluate our new methods.

Chapter 2

Background

In this chapter, we first introduce phylogenies in more detail and describe their role in understanding evolutionary processes (Section 2.1). We then discuss how phylogenies can be used to represent two related but different concepts: gene trees and species trees (Section 2.2). Because of its relevance to the rest of this dissertation, we elaborate on one process that relates species trees and the gene trees (the so-called coalescent process). In doing so, we introduce the concept of Incomplete Lineage Sorting (ILS) and describe statistical a model of coalescence that describe how ILS arises. In Section 2.3, we describe a typical phylogenetic reconstruction pipeline, and go into details of two aspects of the pipeline that are most relevant to this dissertation: Multiple Sequence Alignment (MSA), and species tree estimation. Finally, we explain various procedures and measures used for evaluating the quality of reconstruction results in Section 2.4.

2.1 Phylogeny: an evolutionary tree

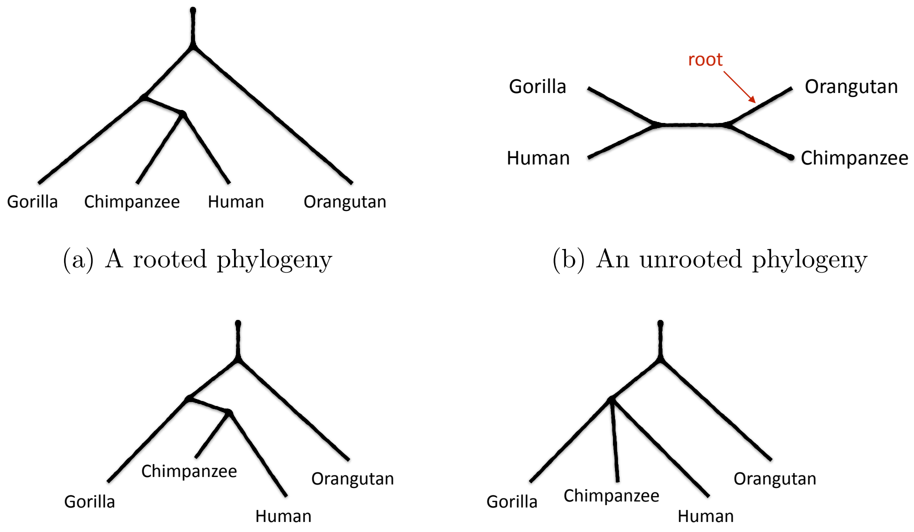
A phylogeny is a model of evolution represented most typically by a tree, but more generally as a network (i.e., a graph). In this dissertation, we almost exclusively focus on phylogenetic trees. In a phylogenetic tree, leaves represent present-day entities[1] (e.g., species) and the tree structure shows how various entities are related to each other through evolutionary time. Parent-child relationships in the tree represent evolutionary relationships: the child entity has evolved from the parent entity. Each internal node of this tree represents an entity that has in evolutionary time evolved to produce new entities. Thus, internal nodes typically represent entities that existed in the past but do not exist anymore (i.e., are extinct). All present-day entities share a common ancestor, namely the root of the tree. The branching structure of a tree is called its topology.

Figure 2.1a shows an example of a phylogenetic tree that depicts the evolutionary relationships between humans and our close relatives: chimpanzees, gorillas, and orangutans. This tree shows that humans and chimpanzees share a common ancestor that they don't share with the other species, and so are closer to each other than either is to gorilla or orangutan. The internal nodes represent species that existed in the past, but are extinct now.

2.1.1 Properties of a phylogenetic tree

Branch Length: The length of edges (also called branches) in an evolutionary tree can be drawn arbitrarily, or can reflect various quantities that can be measured for a branch. For example, the branch length could show the amount

[1] We use the term "entity" instead of a more specific term such as "species", because phylogenies can be used to describe various concepts, as described in Section 2.2.



(a) A rooted phylogeny

(b) An unrooted phylogeny

(c) Deviation from ultrametricity

(d) An unresolved phylogeny

Figure 2.1: **Examples of phylogenies**. A phylogenetic tree, representing the evolutionary relationships between four species of group Hominidae: human, chimpanzee, gorilla, and orangutan. The same phylogeny is shown as both (a) rooted and (b) unrooted trees. Rooting the unrooted tree at the branch labelled as root would produce the rooted tree. When all leaves have the same distance to the root, a tree is called ultrametric. Trees can be ultrametric (a) or can deviate from ultrametricity (c). Trees can be fully binary (c) or unresolved (d). Lack of resolution can signify lack of knowledge about the right relationships, or a true evolutionary multifurcation.

of time between two nodes. If nodes represent species, as they often do, such branch lengths would show the time between speciation events. Alternatively, branch lengths could show the amount of change or the expected amount of change in a character of interest between two nodes. Trees with branch lengths can be ultrametric, meaning that all their leaves have equal distances to the root, or can deviate from ultrametricity, as shown in Figure 2.1c.

Rooting: A phylogenetic tree can be rooted or unrooted (see Fig. 2.1b). Unrooted trees are used when the structure of the relationships between organisms can be inferred but the position of the root cannot. An unrooted tree with b branches can be rooted at any of those branches, producing b different rooted trees.

Multifurcations: Phylogenetic trees can be bifurcating, where all internal nodes have a degree of three, or they can be multifurcating, where at least one node has degree > 3; examples are shown in Figure 2.1d. An internal node with degree greater than three is called a polytomy. Multifurcation in a phylogenetic tree can signify two different scenarios: lack of knowledge about the evolutionary history for a particular part of the tree (a so-called “soft” polytomy), or the belief that the evolutionary history was in fact (close to) a true multifurcation (a “hard” polytomy). For example, the relationship between human, chimpanzee, and gorilla was for a long time represented as a polytomy shown in Figure 2.1d, and it was not clear whether this was a soft or a hard polytomy [41, 42]. More recently, molecular and genomic data has been used to show that humans and chimpanzees are closer to each other than either is to gorilla, and therefore the soft polytomy was resolved [23, 24, 43].

Bipartitions: Each branch in an unrooted tree defines a bipartition of taxa. For example, in Figure 2.1b, we have one internal branch, and that internal branch divides the set of taxa into the following bipartition:

`(human,chimpanzee | gorilla,orangutan)`. A bipartition that has a singleton (i.e. one leaf) as one of its two parts is *trivial* because such a bipartition has to be present in any tree that includes that leaf.

2.1.2 Character evolution

Phylogenetic analyses are based on studying how characters evolve on a tree. At its simplest form, a character is a quantity that can have one of multiple possible discrete states (i.e., values) for each organism. Any character has a particular state at the root, and through evolution, moves from one state to another. Thus, each node in the tree (internal nodes and leaves) has a value for that character. The values of the characters at the leaves are not independent from each other; they are related through the evolutionary history and therefore have information about the evolutionary past.

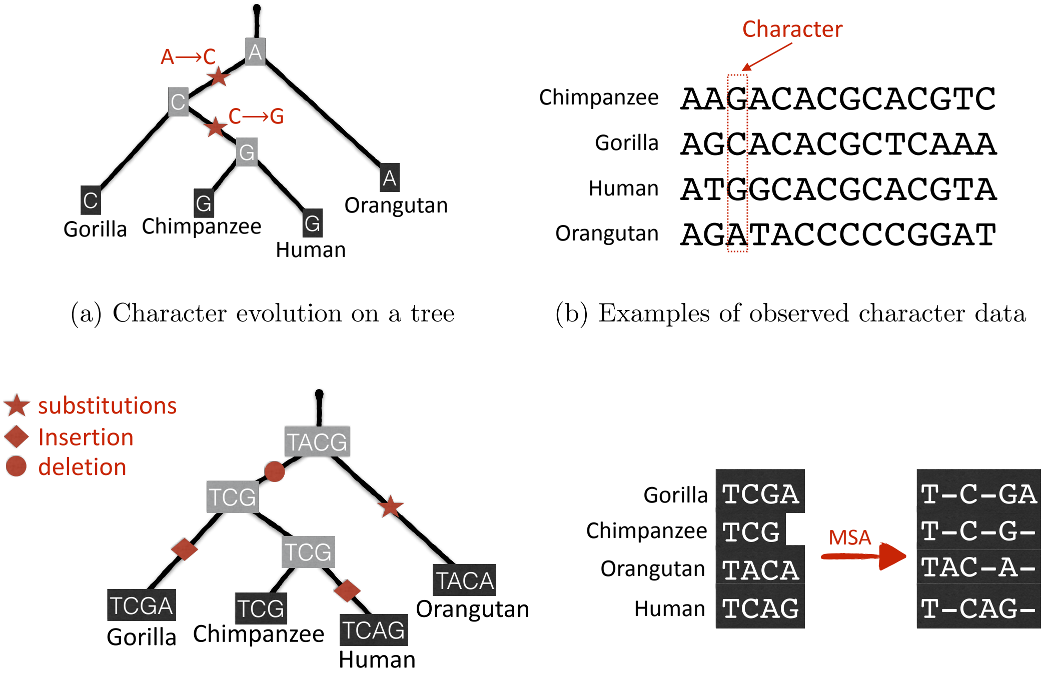
In a phylogenetic study, typically, values of characters are not known for internal nodes, but we can observe (or measure) their values for the leaves of the tree. Since these observed variables contain information about the evolutionary history, given a large enough number of characters, we can hope to recover the evolutionary past. For example, the fact that all birds fly and almost none of the mammals fly gives us some evidence that birds are all closer together than either is to (most) mammals. However, if we use only this character, we would be mislead to think that bats are also grouped with birds. Flight has developed multiple times through evolution, and the fact that bats and birds both fly is not through common decent, but rather through parallel evolutions of a character (sharing a character due to factors other than decent is called *homoplasy*). Only by looking at a larger set of characters (e.g., lactation, body hair, number of ear bones, etc.) can we infer that bats are closer to other mammals than to birds.

Traditionally, morphological characters were used in phylogenetic analyses. With the discovery of DNA, it became clear that at the most basic level, evolution operates on the genetic information encoded in DNA molecules. This opened up the possibility of using molecular data in phylogenetic analyses [5]. DNA, and other molecules derived from it (e.g., RNA and proteins), can be represented as sequences of letters, each corresponding to a unit in a long chained molecule. For example, DNA can be represented as strings of four different characters: `A`, `C`, `G`, and `T`, each corresponding to one of the four nucleotides that encode genetic information. Similarly, proteins can be modeled as sequences of 21 different amino acids residues. These discrete and well-defined strings of letters provide a natural source of phylogenetic characters. Now that we can read genomic data using various sequencing technologies [44], and we can do it cheaply [11, 45], large databases of molecular character data can be assembled for use in phylogenetics [9, 14, 46].

2.1.2.1 Substitutions

Evolution changes molecular sequences through mutations that can have various types. The simplest form of mutation is when a character is *substituted* with another character (e.g., a `A` could change to a `C`). Let's consider a regime of evolution where substitution is the only allowed mutation. In this regime, the root of the tree has a particular string of letters and through mutations and a branching pattern, this single string gives rise to various strings at the other nodes of the tree. The strings at the leaves are what we can sequence and observe; sequences at the other nodes, and indeed the structure of the tree is not known and needs to be inferred. Figure 2.2a depicts a hypothetical mutation process for one character. The particular character shown here is `A` at the root, but throughout the phylogeny, two mutations happen and this results in `A, C, G,` and `G` respectively for orangutan, gorilla, human, and chimpanzee. If we zoom out and look a string of characters, we can build data matrices that look like what is shown in Figure 2.2b. Each column in the matrix corresponds to a different character and all the characters evolve on the same tree. For simplicity, we can further assume that character evolution is independently and identically distributed (i.i.d). The process we just described creates the basic block of statistical models of evolution.

GTR: Let's consider one of the most commonly used models of evolution, called Generalized Time-Reversible (GTR) model [47]. The generative model is parameterized by a model rooted bifurcating tree, with branch lengths that are real numbers, as well as a transition rate matrix that gives the rate of transition between any two letters in the alphabet $\{A, C, G, T\}$. In addition, GTR assumes stationarity, meaning the probability of observing any character is the same for all nodes of the tree, and these *equilibrium* base frequencies are



(a) Character evolution on a tree

(b) Examples of observed character data

(c) Insertions and deletions (indels)

(d) Multiple Sequence Alignment (MSA)

Figure 2.2: **Character evolution**. (a) Characters evolve on a tree through a mutation process guided by the branching structure of the tree. (b) Strings of characters can be observed for leaves of the tree, creating matrices of character data. (c) indels can change the sequence length and blur the character homology. (d) Multiple sequence alignments can be used to build data matrices where each site consists of homologous characters.

also parameters of the model. In addition to stationarity, the GTR model assumes time-reversibility (that is, the rate of transition between any two letters is identical). Thus, there are only 6 transition rates and 4 equilibrium frequencies. The transition rates can be normalized by factoring out the overall rate of mutation, leaving 9 parameters plus this overall mutation rate parameter. Moreover, we typically express the branch lengths in the number of mutations instead of time, and hence overall mutation rate is simply 1, leaving us with 9 parameters and 8 degrees of freedom.

At the root, a random sequence is generated according to the base frequencies. Then, sequences evolve i.i.d down the branches of the tree within a Markovian process; thus, the sequence at the end of each branch depends only on the sequence at the beginning of that branch. The probability of observing any particular letter at the end of a branch is determined by the value at the beginning, the length of the branch, and the rate matrix.

Given generative models such as GTR, one can also try to estimate a phylogeny from sequence data, and we will discuss some of these approaches in Section 2.3. However, we note that since the model is time-reversible, sequence data cannot be used to find the direction of evolution. Thus, these models can be used only to infer unrooted phylogenies.

2.1.2.2 Alignments

Mutations are not restricted to substitutions. Many others types of mutations can also alter the molecular sequences in more complicated ways [48].

Perhaps the most prevalent and important of these other mutation types are insertions and deletions (*indels* for short), whereby genetic material are inserted or deleted throughout evolution. Indels have the effect that they blur what parts of sequences from various organism are related to each other. When two letters in a sequence (or more broadly two characters) are both derived from a common letter in an ancestor, they are called *homologous* and the relationship is called *homology*.

Characters used in a phylogenetic analysis have to be homologous. However, indels result in difficulty in deciding what characters are homologous. For example, in the scenario shown in Figure 2.2c, not all the leaves have the same sequence length, and it cannot be assumed that two letters at the same position are homologous. For example, the last `A` in orangutan is not homologous to the last `A` in gorilla. In order to find the homology relationships in sequence data we need to use sophisticated algorithms, further discussed in Section 2.3.1. The result of these algorithms [28, 29, 49, 50] is a Multiple Sequence Alignment (MSA): a matrix where each site contains only homologous letters. To produce a MSA, dashes are added to sequences so that each column consists entirely of homologous characters. These dashes therefore correspond to the indels. For example, in the multiple sequence alignment shown in Figure 2.2d, dashes in the second column correspond to the single deletion event on the left branch of the root node, dashes in the fourth column correspond to the insertion on the branch leading to humans, and dashes in the last column correspond to the insertion in branch leading to gorilla.

2.2 Gene trees and the species tree

2.2.1 Definitions and concepts

We mentioned that phylogenetic trees can be used to represent evolution for various types of entities. Two inter-related types of entities that can be modeled by phylogenies are species and genes.

Species tree: In a species phylogeny, each leaf represents the entire population of a particular species. The branching structure captures how speciation events split populations of species into subsequent species through a diverse host of mechanisms [6]. For example, in *allopatric* speciation, a population is split into two geographically isolated populations, each of which continue to evolve independently until they constitute two different species. The succession of these speciation events creates a tree, which we call the species tree. The speciation history leaves its mark all over entire genomes of extant species.

Gene trees: A gene phylogeny is a tree that describes the evolution of particular parts of the genome across various species. Genome evolution involves many processes that can result in differences between the evolutionary histories of various parts of the genome [18, 20]. These process include duplication and loss of genes, recombination and coalescence, horizontal gene transfer, and hybridization. The phylogenetic history of a particular part of the genome is broadly referred to as a *gene tree.* Importantly, gene trees can be different from each other and from the species tree. For example, in one part of the genome, human and chimpanzee might be closer to each other whereas in another part, human might be closer to gorilla [24]. These differences between gene phylogenies and their difference to the species phylogeny is referred to as *gene tree discordance* or *incongruence*.

The exact meaning of a “gene”, and the exact ways in which gene trees differ from one another depend on mechanisms that cause discordance. Some of these mechanisms cause individual parts of the genome to have phylogenies that do not agree with the species phylogeny, but do *not* contradict a tree-like species phylogeny. In contrast, other biological processes result in complex evolutionary histories that cannot be represented as trees at the species level. Representing species phylogenies as trees is based on the underlying assumption that each species has evolved from one other species. This assumption of *vertical* evolution is accurate in many cases, but at least two genome evolution mechanisms result in *reticulate* evolution and break the vertical structure: 1) a new species can occasionally evolve as a result of hybridization between two species [51], 2) organisms can pick up genetic material from their environment, a phenomenon known as Horizontal Gene Transfer (HGT) [52, 53]. In such cases, gene phylogenies are still trees, but the species phylogeny is best modeled as a network [54, 55].

For the rest of this section and the rest of this dissertation, we operate under the assumption that the species phylogeny is a tree, and that only a particular cause of discordance called Incomplete Lineage Sorting (ILS) causes true biological gene tree discordance. We elaborate on ILS next, but it’s important to note that ILS is not the only source of true biological discordance even when the species phylogeny is in fact tree-like.

With a tree-like species phylogeny, the major source of gene tree discordance other than ILS is duplication and loss [43, 56, 57]. Gene duplication can create two copies of a gene and copies of a gene that have diverged from each other through a duplication event are called *paralogous*. In contrast, two genes in different species that diverged from each other during a speciation event are called *orthologous*. When some pairs of genes analyzed in a phylogenetic study are paralogous, the resulting gene tree can be discordant from the species tree even in the absence of other sources of discordance such as ILS. In phylogenetic analyses that seek to reconstruct the species tree, researchers try to find orthologous genes, and to the extent that they succeed in this potentially difficult task, they eliminate duplications as a source of discordance. However, undetected paralogy should always remains a possibility in practice. Detection of orthology is an active field of research [58–61], and one that we do not address here. Thus, we only focus on orthologous sets of genes and ignore error in orthology detection.

2.2.2 Coalescence and Incomplete Lineage Sorting (ILS)

A major reason for discordance between gene trees and the species tree is Incomplete Lineage Sorting (ILS), a population level processes that can spill into species level phylogenies, as we will describe. Before describing ILS, we need to briefly introduce some related concepts.

Recombination: A major force in evolution of genomes is recombination [62, 63]. Let's start by considering the population of a single diploid species (but note that asexually reproducing organisms also have mechanisms for recombination [64]). New generations of individuals in a diploid population have genomes that recombine genomes of individuals in the previous generations. Thus, moving across a particular chromosome of an individual, genetic material are initially inherited from one ancestor but can switch in the middle of the chromosome to be inherited from the other ancestors. If you now consider the entire history of evolution for the chromosome since a particular common ancestor (so consider parents, grandparents, and so on), it is easy to see that these recombination events accumulate and divide the chromosome into multiple regions such that the history is shared for all the sites in the same region, but can change from one region to another.

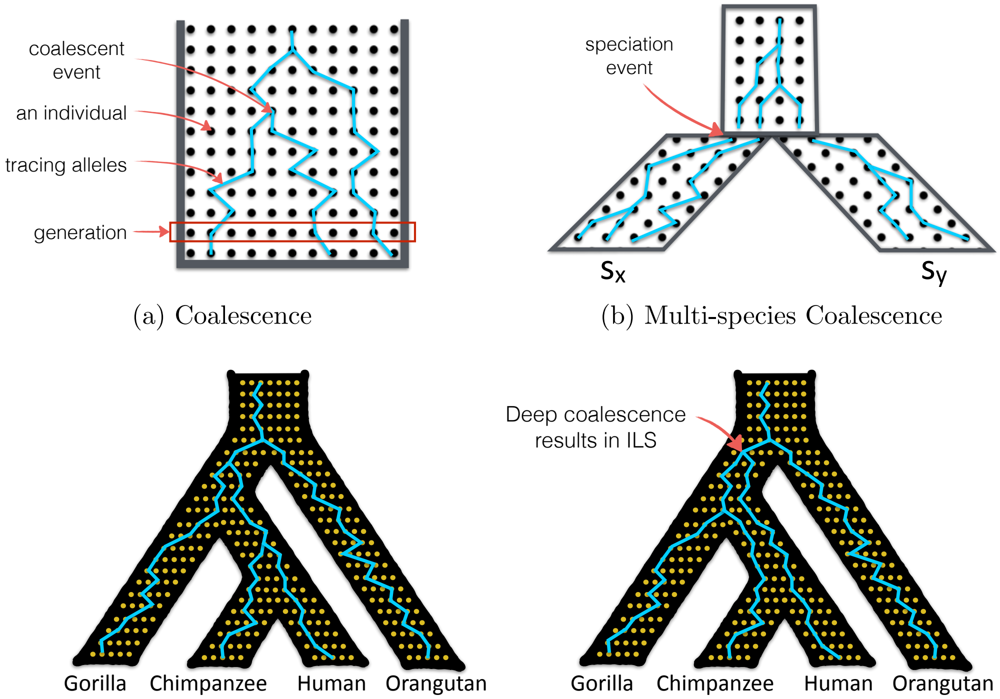
Coalescent genes: A consecutive part of a genome that has been transmitted as a single unit without going through recombination across our organisms of interest is called a coalescent gene, or a "c-gene" [34, 65, 66]. These c-genes constitute the smallest part of the genome that can be analyzed phylogenetically as a unit without worrying about the possibility of having multiple histories embedded in the data. Note that the term "gene" is commonly used to refer something different: stretches inside the genome that code for proteins and perform a certain function. A gene in this functional sense might span multiple c-genes, and multiple c-genes might be present in a single gene [65].

For the purposes of understanding evolutionary histories what matters is a c-gene. For simplicity of terminology, in the rest of this dissertation, unless otherwise stated, we use the term "gene" to refer to coalescent genes[2]. We also use the term locus (loci) and gene(s) interchangeably.

2.2.2.1 Coalescence

Different c-genes can have different evolutionary histories, and understanding this at the population level is simple. Recombination creates divergent evolutionary histories and each history corresponds to a different tree. This phenomenon is mathematically modeled in what is called the coalescent process [67]. The coalescent process starts with present day variants of a gene (called *alleles*) and traces them back in time across successive generations by following which alleles in the previous generation produced a given allele in the current generation. As we move back in the time, we eventually reach a point where the two alleles share a common ancestor. This point is where the two alleles *coalesce*. The coalescent history creates a lineage tree, as shown for example in Figure 2.3a. Kingman's coalescent model makes assumptions of non-overlapping generations, constant population size, random mating, and a sufficiently large population size; given these assumptions, the time (measured as the number of generations) to coalescence for two randomly selected alleles can be shown to be exponentially distributed [67]. Thus, if T_i is the

[2] Note that a "gene" is technically defined as a unit of heredity of a living organism, and so calling a c-gene simply a gene is justified. However, since the term gene is more commonly used to refer to functional genes, the distinction is clarified here.



(a) Coalescence

(b) Multi-species Coalescence

(c) Concordant gene tree

(d) Discordant gene tree due to ILS

Figure 2.3: **Coalescence and multi-species coalescence**. (a) The coalescent process for a single population. Each row is a generation and alleles are traced back in time through generations; coalescence is when two lineages find a common ancestor. Coalescent history creates a lineage tree, here shown for three samples drawn from a population of 11 individuals. (b) Multi-species coalescent for two leaf species s_x and s_y and their parent population. Here, $k_x = 3$ and $k_y = 2$ individuals are sampled for s_x and s_y respectively; at the speciation point, $r_x = 2$ and $r_y = 1$ lineages exist; these start a new coalescent process in the parent population with three lineages. (c) Multi-species coalescence results in a gene tree inside a species tree; here, the gene tree is concordant with the species tree in terms of the topology. (d) When lineages coalesce deeper than their first ancestral population, they have a chance to create gene trees that are different from the species tree, as shown here.

waiting time for the coalescence of any two alleles from i sampled alleles, and the population size is N_e, then:

$$T_2(N_e) \sim \exp(t; \lambda = \frac{1}{N_e}) = \frac{1}{N_e} e^{-\frac{t}{N_e}}$$

And more broadly:

$$T_i(N_e) \sim \exp(t; \lambda = \frac{\binom{i}{2}}{N_e}) = \frac{\binom{i}{2}}{N_e} e^{-t\frac{\binom{i}{2}}{N_e}}$$

To simplify these equations, and since N_e is in many cases unknown, one can simply measure time in N_e generations, and state the waiting time in what is known as “coalescent units”. Note that for diploid species (i.e., those with two versions of each chromosome), a population of size $2N_e$ is equivalent of a haploid (i.e., single chromosome) population of size N_e; therefore, coalescent units are measured in $2N_e$ generations for diploids. With this formulation, T_i simplifies to being an exponential random variable with rate $\binom{i}{2}$. To calculate the time to most recent common ancestors of a set of n samples, we simply need to sum up T_i random variables for $1 < i \leq n$. Similarly, we can compute the probability that starting from u lineages at the current time and tracing back, we have v lineages in t generations before present time [68]:

$$P_u v(t) = \sum_{j=v}^{u} e^{-\binom{j}{2}t} \frac{(2j-1)(-1)^{j-v}}{v!(j-v)!(v+j-1)} \prod_{y=o}^{j-1} \frac{(v+y)(u-v)}{u+y} \tag{2.1}$$

2.2.2.2 Multi-Species Coalescent (MSC)

All the previous discussions were related to a single randomly mating population. But the general framework can be extended to phylogenetic analyses where multiple populations corresponding to multiple species are present (see Fig. 2.3b). The extension of the coalescent process to multiple species is known as the Multi-Species Coalescent (MSC) model [69]. The model tree is a species tree with branch lengths in coalescent units and from each leaf species s_i we have sampled k_i different individuals. Each branch is modeled using one instance of the Kingman coalescent process with a fixed population size. At the speciation events (i.e., internal nodes), the lineages that have not coalesced yet in the child populations are moved to the parent population and a new identical process is initiated.

Tracing alleles back in time happens in the following way. Let's consider two sister species s_y and s_y and their parent population as shown in Figure 2.3b. At the terminal branch leading to species s_x, we start with k_x individuals at the bottom and trace back the Kingman coalescence for t_x generations (where t_x is the length of the branch); during this time, some but not necessarily all alleles coalesce. When we reach the start of the branch, assume $r_x \leq k_x$ branches remain (thus $k_x - r_x$ coalescent events happened on this branch). The probability of this scenario can be calculated using Equation 2.1. A similar process also happens at s_y and let's assume r_y alleles remain once we reach the ancestral node (i.e., r_y alleles have not coalesced). These remaining alleles go back to the ancestral population. Therefore, on the branch above s_x

and s_y, we start a new Kingman coalescent process with $r_x + r_y$ alleles at the bottom. We repeat this process on all branches until all the alleles coalesce in the root branch. This model, therefore, assumes independence between coalescent histories in different branches of the species tree, given the number of lineages that go in an out of a branch. The coalescent history represents a gene tree that evolves *inside* the species tree. Since the coalescent process is random, it can lead to various gene trees, some of which can be different from the species tree, as we next show.

Incomplete Lineage Sorting (ILS): Tracing a lineage through the multispecies coalescent process can result in various gene trees, as depicted in Figures 2.3c and 2.3d. When two lineages from sister populations reach the parent population (in the backwards coalescence tracing) they may or may not coalesce in that first ancestral population. When they don't coalesce, the two lineages go further back in time to a deeper ancestral populations. At that more ancestral population, other lineages from other species are also present. Since mating is random, lineages from these other species have a chance of coalescing with lineages from one of the two sister species before those two lineages coalesce with each other. When this happens, gene trees become discordant with the species tree, and this scenario is called Incomplete Lineage Sorting (ILS).

Figure 2.3d shows one example of ILS. Here, the lineages from human and chimpanzee do not coalesce in their ancestral population and go further back to the population ancestral to human, chimpanzee, and gorilla. In that deep ancestral population, the gorilla lineage and the chimpanzee lineage happen to coalesce first and only then this lineage coalesces with the linage from human. This creates a gene tree where gorilla and chimpanzee are sister species, unlike the species tree where human and chimpanzee are sisters.

Gene trees inside the species tree: Under the MSC model, each species tree defines a unique distribution on the gene trees [70]. Thus, for each gene tree topology, one can compute the probability of observing that topology among a random sample of gene trees. Moreover, the species tree is uniquely identifiable from the true distribution of gene trees [70, 71]. Thus, despite the possibility of having high levels of gene tree discordance, one can still hope to recover the true species tree given a large enough number of true gene trees. This task however is not trivial. For example, it has been shown that for certain shapes and lengths of branches, the most likely gene tree can be different from the species tree [72] (the so-called *anomaly zone*). Thus, simply taking the most frequent gene tree as an estimate of the species tree is not sufficient. We come back to the question of estimating a species tree given gene trees in Section 2.3.3.

Species radiations: It is constructive to think about the scenarios that result in high levels ILS. To produce discordance due to deep coalescence, it is required that two lineages fail to coalesce in their ancestral population. Recall that the time to coalescence is exponentially distributed, with an expected time equal to the size of the population. Consequently, for two branches to be likely to not coalesce, we either need to have branches that are short or population sizes that are large. Population sizes depend on biological organisms of interest and are in fact in many cases hard to estimate for extinct species. The time between speciation events depends on the tempo of evolution [73]. Sometimes speciation events happen quickly and in succession and other times long times are passed between successive speciation. When many new species evolve in short spans of time, the chance of ILS increases [74]. For example, such *radiation* scenarios have been postulated to have happened for birds [39, 75, 76] and mammals [77], among other organisms.

Summary: To summarize, gene trees can be discordant with the species tree for various biological reason. A major biological process that creates discordance is incomplete lineage sorting, modeled by multi-species coalescent model. The MSC model defines a unique distribution on gene trees, and a true gene tree distribution defines a unique species tree. ILS is most likely for short successive branches in the species tree, commonly found in rapid radiations.

2.3 Phylogenetic reconstruction

Building phylogenies from sequence data has been the subject of much research in the past few decades [5, 8, 9]. Methods of reconstructing phylogenies are varied in many aspects, including the sources of character data they use (morphology [78], sequence data [5], sequence repeat abundance [79], etc.), biological processes they seek to model (substitutions, indels, ILS, duplication and loss, etc.), and the methods they use (maximum parsimony, distance-based reconstruction, Bayesian or maximum likelihood statistical inference, etc.). While we cannot hope to cover all these diverse methodologies, we cover the most standard pipeline and we elaborate on parts of the pipeline that are most closely related to the rest of this dissertation.

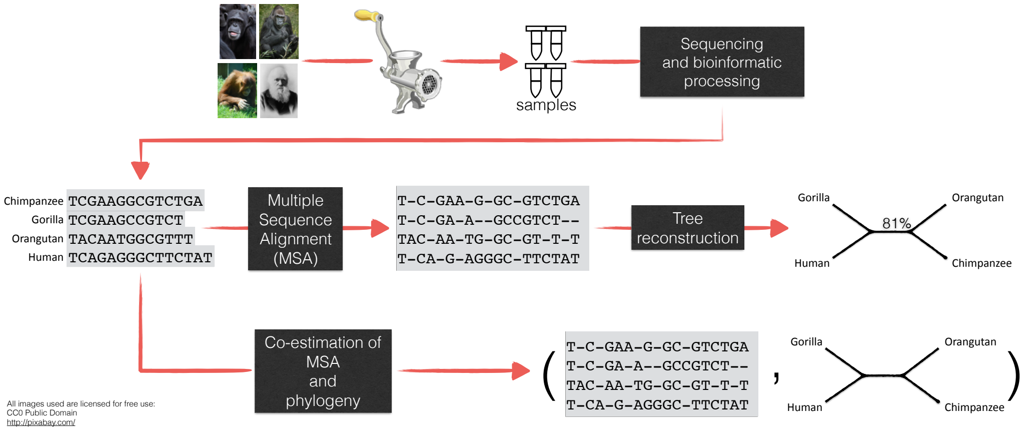


Figure 2.4: **Single gene phylogenetic reconstruction pipeline**. The traditional pipeline used for phylogenetic reconstruction is shown. After samples are gathered from organisms of interest, DNA, RNA, or protein of gene(s) of interest are sequenced. Results of the sequencing technologies go through bioinformatic post-processing and sequence data are obtained for genes of interest. The traditional pipeline for phylogenetic reconstruction consists of first aligning the sequences and then estimating a phylogeny based on the alignments. More recently, a new approach has emerged where sequence alignments and trees are co-estimated.

Gathering data: The process starts by gathering samples from organisms of interest, a potentially difficult process that we can willfully ignore as computer scientists. With samples at hand, we need to gather character data. Modern phylogenetics is mostly based on molecular sequence data, but sequencing technologies are varied and sequence data can be gathered in many different ways. Most commonly, DNA or RNA molecules are sequenced. When sequencing RNA, we can only gather data from the protein sequencing portions of the genome, and it needs to be understood that protein coding genes are only a small portion of the genome. To sequence data, we might use technologies that target specific "marker" genes believed apriori to be particularly useful for phylogenetic reconstruction. Or, as is becoming more common, we can try to sequence the entire genome using various next generation sequencing technologies [44, 45]. These high throughput technologies target the entire genome or transcriptome (protein-coding regions of the genome) and produce short error-prone fragments of data that are then bioinformatically assembled into longer sequences [80]. Error and incompleteness should be expected in data produced by these technologies.

Pipeline for analyzing a single gene: We describe the pipeline for analyzing a single gene here, and then in Section 2.3.3, discuss how this pipeline can be extended for analyzing multiple genes. A traditional pipeline has two steps (see Fig. 2.4): sequences are first aligned to produce a MSA, and then the MSA is analyzed using a phylogenetic reconstruction tool to create a tree. As we will discuss in the next section, this traditional two-step pipeline, like all pipelines of data analysis, has an important drawback: the quality of alignment can impact the tree building [15, 81–84]. To address this issue, and a fundamental dependency between alignment and tree estimation problems [30, 85, 86], researchers have also developed an alternative approach where alignments and trees are co-estimated in a single analysis.

We next provide some background information about MSA methods and tree reconstruction methods, and then move on to describe how multiple genes can be used for phylogenetic reconstruction.

2.3.1 Multiple Sequence Alignment (MSA)

Reconstructing multiple sequence alignments is one of the most basic tasks in computational biology, with application to predicting the structure and function of RNAs and proteins and estimating phylogenies. Many methods have been developed for aligning multiple sequences (see [28, 29, 50]), and some such as ClustalW [87], Muscle [88], and MAFFT [89, 90] are in widespread use. The goal of MSA is to add gaps to sequences such that all letters in the same column are homologous; i.e., have evolved from a common ancestor through substitution processes.

MSA tools are generalizations of the simpler problem of pairwise alignment [91, 92]. Pairwise alignment algorithms use dynamic programming to optimize a score that rewards sequence similarity and penalizes gaps. Optimizing such a score for more than two sequences becomes NP-hard [27], even when a fixed phylogenetic tree is used to guide the alignment [26], and so heuristic approaches are required. The tools mentioned before are all heuristics for solving the MSA problem. Many other heuristics also exist, including OPAL [93], T-Coffee [94], FSA [95], and ProbCons [96].

Multiple sequence alignment tools typically use a guide tree that they somehow compute from the sequence data, and use the guide tree to compute an alignment of all sequences. For example, progressive alignment methods use the tree as a guide to progressively add sequences to a growing alignment, each time using pairwise alignments for adding sequences. Iterative alignment tools use a similar approach, but they also update the alignments of already added sequences as they progress.

Alignment and tree co-estimation: A main concern in multiple sequence alignment is the interdependency between alignments and trees. Methods of reconstructing phylogenetic trees require alignments. But at the same time, an alignment is an evolutionary statement of homology, and therefore can be done well only when the phylogeny is known [84]. Considering this dependency, a method called PRANK [97] and its newer version, PAGAN [98], assume that the phylogeny is known and make sure that the insertions and deletions added to sequences are compatible with the phylogeny. For example, they do not allow two phylogenetically independent insertions at the same site. These methods, however, assume the correct phylogeny is known and they can be sensitive to the quality of the tree used. This limits their applicability to phylogenetic reconstruction.

More recently, co-estimation methods have been developed to estimate alignments and trees simultaneously. Some of these methods use statistical models of evolution that incorporate both indels and substitutions (e.g., [99–102]). Most of these methods use Bayesian Markov Chain Monte-Carlo (MCMC) [103] to find the probability distribution of alignment/tree pairs according to those models [85, 101, 104]. Some of these methods, such as Bali-Phy [105], are implemented in available software programs that have been optimized to be able to handle datasets with moderate size (tens of sequences, and perhaps even more).

SATé: The extremely large space of alignment/tree pairs makes Bayesian co-estimation methods limited in the size of the dataset they can analyze. More recently, a tree/alignment co-estimation method called SATé was developed with the goal of being able to accurately analyze very large datasets [30, 31]. SATé and its newer version SATé-II iterate between tree estimation using Maximum Likelihood (ML), described in Section 2.3.2, and alignment estimation using divide-and-conquer. The current tree is used in each iteration to divide the set of sequences into smaller subsets of sequences that are likely to be close together in the true tree. Each subset is then aligned using the best available method that can handle these smaller datasets with high accuracy (e.g., MAFFT). Alignments obtained on these subsets are then merged together using methods for aligning two alignments (e.g., OPAL [93]). The exact strategy for dividing the dataset into sub-alignments and for merging subsets is different between SATé and SATé-II. Since SATé is heavily used in our next chapter, we defer a more detailed description of its algorithm to Chapter 3.

2.3.2 Tree reconstruction

Tree reconstruction techniques can be divided into four categories: distance-based methods, Maximum Parsimony (MP), Maximum Likelihood (ML), and Bayesian methods. In their most standard form, these methods take into account the substitution process but ignore other biological processes such as indels. In other words, the standard forms of these methods treat gaps as missing data. However, it should be noted that most of these method can be potentially extended to consider more complex scenarios of evolution.

In the rest of this section, we provide more details about each of these classes of methods, but before doing that, it is constructive to first introduce the concept of statistical consistency.

Statistical consistency: A method designed to estimate a particular value from data is called statistically consistent when as the number of data points increases, the estimates converge to the "true" value. When data are generated using a statistical generative model, then statistical consistency of a method of inference means that its estimates of parameters of the model converge in probability to the true parameters as the amount of data is increased.

2.3.2.1 Maximum parsimony

MP seeks to find a tree that explains the observed data with the fewest possible substitutions. Given a tree and character data, one can in time that is polynomial in the number of species assign character states to ancestral nodes such that fewest number of substitutions are required along the branches of the tree. The problem of finding the MP tree is NP-hard [106], but various heuristic solutions for finding an approximate solution have been developed and implemented in available software [107–110]. The justification of finding the maximum parsimony criteria for reconstructing trees is that absent of any other information, we try to invoke fewest possible character changes along the branches of the tree. When characters of interest change very rapidly, one expects to see many character changes along the tree, and thus seeking the maximum parsimonious tree is not easily justifiable [111].

2.3.2.2 Maximum likelihood

As mentioned earlier, various models of sequence evolution have been developed through the past decades [47, 112–114] (see for example our discussion of GTR under Section 2.1.2). These models provide a way of calculating the probability of observing character data given a model tree (i.e., the likelihood of the tree) [7]. Assuming a model of sequence evolution has generated the data, a natural way to reconstruct phylogenetic trees is to find the tree that has the maximum likelihood (ML) (i.e., data have the highest probability if we assume that tree has generated them). Finding the ML tree is NP-hard [115], but if it could be solved, the result is statistically consistent assuming the data is generated under the model is used for ML. In contrast, MP is not statistically consistent data is generated under one of common models of sequence evolution such as GTR [111].

Solving ML has been the subject of extensive research and many heuristic approaches have been developed to find an approximate ML tree [7, 116–119]. ML tools perform a heuristic search, usually using hill-climbing methods: they start from an initial tree, calculate its likelihood by optimizing free parameters, perturb the tree in small ways and recalculate likelihood, and accept changes that improve the likelihood until they reach some local optima. Some of the most widely used tools for ML tree estimation include RAxML [120], GARLI [117], PhyML [121], and FastTree-II [119]. Some ML methods tools are optimized to be fast and scalable; in particular, RAxML can use parallelism efficiently, and FastTree has been successfully used to estimate very large trees with many thousands or even a million sequences [119, 122]. There is some evidence that FastTree might be as accurate as RAxML while running in a fraction of time [123].

2.3.2.3 Bayesian estimation

Bayesian methods, just like ML, use models of sequence evolution, but instead of finding a single tree, they estimate a probability distribution on trees. Bayesian methods are typically based on MCMC searches in the tree space, and some widely used Bayesian tools include MrBayes [124], RevBayes [125], and BEAST [126]. Bayesian methods tend to be slower than ML, as convergence to the correct probability distribution can require a long running time.

2.3.2.4 Distance-based

A fourth category of tree reconstruction techniques relies on first computing a distance between all pairs of leaves, and then using these distances to compute a tree. Many different methods (e.g., Neighbor-Joining [127]) and many ways of calculating distances have been developed. Some combinations of distance measurement and distance-based algorithms have been shown to be statistically consistent under statistical models of sequence evolution [128].

2.3.2.5 Branch support

Inferring evolutionary histories is not easy and except in experimental settings where the evolutionary history is known [129], we cannot ever have full confidence in a phylogenetic tree. Some level of error should be expected in the trees produced by any tree reconstruction method. It is therefore desirable to not only estimate a tree, but also compute a measure of confidence for the tree produced and individual branches of the tree. Bayesian methods readily provide such measures of support as they produce a distribution of trees. For other methods, the most commonly used technique for estimating support is bootstrapping [130, 131].

The most typical bootstrapping procedure (non-parametric bootstrapping) creates a large number of replicate datasets, each consisting of randomly resampled columns of the original character data, with as many columns as the original data (resampling is with replacement). Each bootstrap replicate is analyzed separately, and the idea is that these bootstrap replicates provide a sample of the possible universe of the data we could have seen. The frequency of seeing each branch of the estimated tree in the set of bootstrap replicates is then taken as a measure of support for that branch.

2.3.3 Analyzing multiple genes

Analyzing various genes enables us to infer potentially discordant histories specific to individual genes, which might be of biological interest for various reasons such as inferring gene function [132]. However, estimating the species tree also often requires analyzing multiple genes for two fundamental reasons. On the one hand, any particular gene would include a limited number of sites, and therefore can provide only limited phylogenetic signal. Using multiple genes increases the amount of data and increases statistical power. On the other hand, since gene trees can be different from the species tree, we cannot be confident that even a completely correct reconstruction of the gene tree matches the species tree. In fact, under conditions conducive to high levels of gene tree discordance, any random gene tree can be very different from the species tree. To be able to infer the species tree, one has to be able to analyze many genes and take into account their overall distribution.

Pipelines for multi-gene phylogenetic reconstruction are varied, but broadly fall into three categories: concatenation, summary methods, and co-estimation, as shown in Figure 2.5. As emphasized before, here we are concerned only with pipelines that treat ILS as a source of incongruence and we assume that our data consist only of orthologous genes.

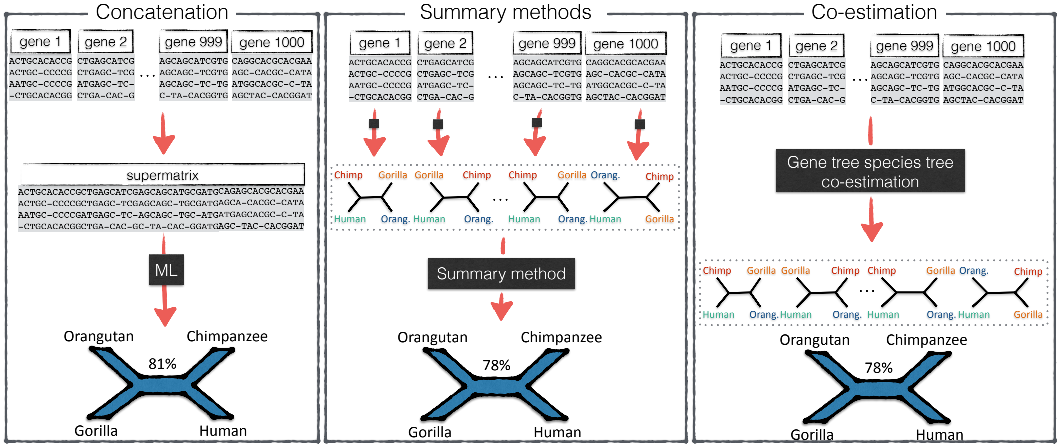


Figure 2.5: **Multi-gene phylogenetic reconstruction pipelines**. Concatenation: all gene data are concatenated into one supermatrix which is then analyzed using the phylogenetic reconstruction method of choice, such as ML. Summary methods: gene trees are estimated for all each genes separately (e.g., using ML) and then the set of these estimated gene trees is used as input to a summary method to produce the species tree. Co-estimation: gene trees and the species tree are all co-estimated in one statistical inference.

2.3.3.1 Concatenation

In the most basic pipeline, researchers simply concatenate all the data into one large supermatrix of data, and analyze the supermatrix in one inference. Such a *concatenation* approach takes full advantage of the statistical power that a larger dataset can provide, and some authors initially argued that it can be successfully utilized to solve longstanding difficult phylogenetic problems [12]. A major drawback of this approach is that it completely ignores gene tree discordance.

As noted before, here, we assume true gene trees (as opposed to inferred gene trees that are reconstructed with some error) can be different from the species tree only due to ILS. Under this assumption, it has been shown in simulation studies that concatenation can reconstruct the incorrect species tree (e.g., see [32, 133, 134]), even with high support [135]. An intuitive way to see why concatenation can result in wrong species trees is to recall that in anomaly zone, the most likely gene tree is different from the species tree. Thus, the idea that the dominant signal across various genes should give the species tree is not justified. Recently, it has been mathematically proved that concatenation using ML (CA-ML) is statistically inconsistent and in fact can be positively misleading [136]; thus, it can produce the wrong tree, and as the amount of data is increased, it can converge to the incorrect tree with high probability. It's worth noting that these proofs of inconsistency are for the case where a single set of branch lengths and model parameters are allowed to be inferred in the ML analysis (i.e., an unpartitioned analysis). The consistency of CA-ML is unknown when ML analysis is partitioned such that one topology is estimated but branch lengths and other model parameters are allowed to be estimated separately for each gene (i.e., a partitioned analysis)

2.3.3.2 Summary methods

Coalescent-based species tree estimation: Under the MSC model, a probability distribution on gene trees defines a unique species tree, as explained earlier. This basic observation opens up the path to a different approach to species estimation [137]. When a large enough number of genes have been sequenced, one can try to estimate the gene tree distribution and use this distribution to estimate the species tree under the MSC model. The true gene tree distribution gives the probability of observing each gene tree when the species tree is known. The empirical gene tree distribution is simply the percentage of times a particular gene tree has been observed. With a large number of genes, and if we are able to reconstruct the gene trees correctly, the law of large numbers can be invoked to argue that the empirical gene tree distribution converges in probability to the true species tree estimation. Besides the need to have a large number of genes, the two major challenges are estimating gene trees correctly and summarizing them accurately. Two coalescent-based pipelines have emerged for reconstructing species trees from gene tree distributions, and their main difference is their approach to gene tree estimation.

In the simpler pipeline, which we call *summary methods* (others have used other terms such as *shortcut coalescent methods* [34]), gene trees are first reconstructed independently from one another. This produces a collection of gene trees, which are then summarized to produce the species tree. The summarization step requires a technique that takes as input a set of gene trees and produces a species tree.

Many summary methods have been developed to estimate a species tree from gene trees. The oldest approach, minimizing deep coalescence (MDC) [18, 138–140], was based on the parsimony principal, but is not statistically consistent [141]. Various statistically consistent methods have also been developed. Some of these consistent methods use only gene tree topologies, and they mostly use simple distance-based techniques with distances computed from gene tree distributions (e.g., STAR [142], STELLS [143]). More recently, a more sophisticated approach called MP-EST has been developed that finds the maximum pseudo-likelihood species tree given the rooted gene trees [134]. Another class of methods (e.g., GLASS [144] and STEM [145]) uses both gene tree topologies and branch lengths to estimate the species tree. Most of these methods use rooted gene trees, but statistically consistent methods that can use unrooted gene trees have also been developed (e.g., NJst [146] and the population tree from BUCKy [133]).

All these summary methods (except for Bucky-pop) get as input one tree per gene, and they all provide statistical guarantees of consistency under the MSC model *only* when the input collection of gene trees converges in probability to the true gene tree distribution in limit. Since the gene trees provided to these methods are all inferred, these statistical guarantees cannot predict what happens in practice where gene tree estimation introduces error [147]. Moreover, the gene trees are all inferred independently, and this independent inference means that each gene tree inference is done using rather limited amounts of data. As a result, high levels of gene tree error are to be expected in many cases [34]. As we will argue in this dissertation, this is a major shortcoming that needs to be addressed.

2.3.3.3 Co-estimation of gene trees and species trees

basic limitation of summary methods is the independent estimation of gene trees. A more justified approach is to co-estimate gene trees and the species tree. To see this, it is helpful to think about the model that we are assuming generates the data. The data are generated in two steps. First, the species tree generates a set of gene trees according to the MSC model and then each gene tree independently generates sequence data according to some model of sequence evolution, such as GTR. In this generative process, all the gene trees are *conditionally* independent given the species tree but they are not completely independent. In other words, knowledge about one gene tree does have an impact on our belief about what other gene trees might look like. Thus, ideally gene trees have to be all co-estimated.

Co-estimation methods use statistical frameworks to infer gene trees and the species tree produced under the two-step model described above; thus, they co-estimate all gene trees and the species tree in one statistical inference. Co-estimating gene trees has the advantage that the estimation of each gene tree is affected by more than just limited data available from that gene. Simulation studies have demonstrated that co-estimation methods can estimate gene trees with much better accuracy than independent estimation of gene trees [148, 149]. However, this approach also faces limitations. Because all the gene trees, and the species tree, are inferred in one analysis, the parameter space that needs to be explored becomes extremely large and so co-estimation approaches are computationally demanding.

Two main co-estimation methods are BEST [150] and *BEAST [151]. These methods are both based on a Bayesian MCMC search that simultaneously samples the probability distributions of all gene trees. Despite their high accuracy [36, 152], these methods have serious limitations in terms of the dataset size they can analyze. For example, researchers have reported difficulty in running these methods on biological datasets (e.g., [16]) or to run them to convergence for relatively small simulated datasets (e.g., 11 species and 100 genes [17]). Thus, the application of these methods in practice remains limited, although some recent works (including some by us [153]) have tried to increase their applicability in practice.

BUCKy [133, 154] is a method that does not neatly fit into our two categories of summary methods or co-estimation. BUCKy takes as input a distribution for each gene tree, and these distributions are estimated independently. But instead of using these distributions directly, BUCKy estimates what it calls concordance factors for various genes and uses these to estimate the species tree. This process can be viewed as using all gene tree distributions collectively to "correct" gene tree distributions. In that sense, BUCKy can be viewed as co-estimating gene trees and the species tree. Like co-estimation methods, BUCKy has been shown to be robust to estimation error in individual genes [148, 155] but it also is computationally intensive [148]. Note that after concordance factors are computed BUCKy has two ways of summarizing them into the species tree, and only one of these two approaches (the population tree) is statistically consistent under MSC.

In addition to these two commonly used coalescent-based pipelines, various new pipelines have been developed in the past few years, but these are less commonly used. For example, some methods (e.g., SNAPP [156] and SVDquartets [157]) seek to estimate the species tree directly from the data, without computing gene trees. These methods are promising, but they are in their infancy, and are not the subject of our focus in this dissertation.

2.4 Method evaluation

Throughout this dissertation, we use experimental studies to evaluate various methods. Our reported results will be based on both simulated and biological datasets. In simulation experiments, we generate synthetic data using models of sequence evolution with various procedures that we will explain. The data are then analyzed using various methods that estimate the MSA and reconstruct the phylogeny. In these experiments, since the ground truth is known, we can easily measure the error in our reconstructions using the metrics we describe below. For biological datasets, the truth is usually not known, but we use various hand curated reference alignments and trees or knowledge from the literature to judge the quality of the results produced by various methods.

2.4.1 Comparing two phylogenies

Given two phylogenies, there are various ways to compare them to get a score of their similarity. Some of the most standard measures of tree similarity, used throughout this dissertation, are defined here. Not all measures of tree similarity are symmetric. In the definitions below, we compare a *reference* tree to an *estimated* tree, both on the same exact set of leaves.

False Negative (FN) rate: The FN rate is the proportion of bipartitions in the reference tree that are not found in the estimated tree. For example, two binary trees on 15 species each have 12 nontrivial bipartitions. If 9 of those 12 bipartitions are present in both trees, the FN rate is $\frac{3}{12} = 25\%$. This metric is also known as the *missing branch rate.*

False Positive (FP) rate: The FP rate is the proportion of bipartitions in the estimated tree that are not in the reference tree.

Robinson-Foulds (RF) distance: This is the total number of branches that are different between the two trees [158]. Normalized RF (or RF rate) is the proportion of branches that are different between the two trees, and is simply the mean of FN and FP rates. When both trees are fully bifurcating, FN rate, FP rate, and RF rate are all equal. RF is the most commonly used metric for comparing trees; however, this metric is not appropriate when the reference tree is not fully bifurcating.

When the two trees are not on the same exact set of leaves, other measures of similarity can be used (e.g., compatibility [159]) and we define these metrics where we use them.

2.4.2 Comparing two alignments

The following standard metrics are used for comparing two alignments. The comparisons are between a reference alignment and an estimated alignment.

The SP-score: the percentage of all pairwise homologies in the reference alignment recovered in the estimated alignment.

The modeler score: the percentage of all pairwise homologies in the estimated alignment that are found in the reference tree.

Pairs score: average of the SP-score and the modeler score (averaging these two scores amounts to penalizing false positive and false negative homologies equally).

TC score: the number of columns that are recovered entirely correctly in the estimated alignment.

Alignment accuracy is measured using a software we developed called FastSP [160].

Summary: To summarize, when data from many genes are available, one can concatenate the data and ignore gene tree discordance, or can try to reconstruct the species tree using the multi-species coalescent (MSC) model. Two coalescent-based pipelines are in common usage: summary methods and co-estimation methods. Summary-based methods first estimate gene trees separately and then summarize them, and co-estimation methods reconstruct all gene trees and the species tree in a single statistical inference.

Chapter 3

PASTA[1]

Multiple sequence alignment (MSA) is a basic part of bioinformatics, used in many analyses such as predicting the structure and function of RNAs and proteins and estimating phylogenies. As described in Section 2.3.1, an MSA is an evolutionary statement of homology (i.e., similarity due to common decent) and is created by adding gaps to sequences such that all the sequences have the same length. The goal is to add these gaps where insertions and deletions have happened, such that letters in each column are all homologous.

In this chapter, we introduce PASTA, a new multiple sequence alignment algorithm. PASTA is an extension of SATé [30, 31], but is more scalable and accurate. We evaluate PASTA on biological and simulated data with up

[1] Parts of this chapter have appeared in the following papers:

- Siavash Mirarab, Nam Nguyen, and Tandy Warnow. PASTA: ultra-large multiple sequence alignment. In *Proceedings of the International Conference on Research in Computational Molecular Biology*, pages 177–191. Springer International Publishing, 2014
- Siavash Mirarab, Nam Nguyen, Sheng Guo, Li-San Wang, Junhyong Kim, and Tandy Warnow. PASTA: Ultra-Large Multiple Sequence Alignment for Nucleotide and Amino-Acid Sequences. *Journal of Computational Biology*, 22(05):377–386, 2015

In both paper, SM and his supervisor, TM, designed the method, designed the studies, and wrote the papers (with comments from the other authors), and SM implemented the methods. SM and NN ran experiments. SG, LW, and JK produced the RNASim datasets and contributed to writing.

to a million sequences, and show that PASTA produces highly accurate alignments, improving on the accuracy and scalability of the leading alignment methods. Trees estimated on PASTA alignments are also highly accurate – slightly better than SATé trees, but with substantial improvements relative to other methods. Finally, PASTA is highly parallelizable and requires relatively little memory.

We start by giving motivations for developing a new method in Section 3.1 and then present background information about SATé in Section 3.2. We next describe the algorithmic details of PASTA in Section 3.3. We present the experimental setup in Section 3.4 and results in Section 3.5, and finish with summary and directions for future research in Section 3.6.

3.1 Motivation

Despite the large number of multiple sequence alignment tools (see Section 2.3.1), only a handful of the many MSA methods are able to analyze large datasets with 10,000 or more sequences [15]. Some of these scalable methods have focused on relatively slow evolving sequence datasets and have shown in performance studies that they can provide good accuracy, as measured by traditional alignment scoring criteria (sum-of-pairs or column scores). For example, Clustal-Omega has been recently developed and used for aligning large protein family databases [162]). Other methods that can analyze datasets with 10,000 sequences or more include Muscle [88, 163], Mafft-Parttree [164], Mafft-profile [165], and SATé-II [31]. SATé and SATé-II, in particular, focus on phylogeny estimation and datasets with high rates of evolution. SATé-II was able to produce sufficiently accurate alignments of sequence datasets that evolve under relatively high rates of evolution with up to 28,000 sequences [31].

Little is known about alignment accuracy and its impact on tree accuracy for datasets with more than few tens of thousands of sequences. Yet, phylogenetic analyses of very large datasets are necessary and important for answering many downstream biological questions. For example, new methods of studying co-evolution of sites in a protein alignment assume that accurate alignments of tens of thousands of sequences can be constructed [166].

Phylogenetic analyses of datasets containing more than 100,000 sequences are also being attempted. One important reason for reconstructing phylogenies of very large datasets is increasing taxon sampling, a well-established factor that impacts phylogenetic accuracy [13, 167–169]. For example, the iPTOL project [170] intends to produce trees with hundreds of thousands of plant species. A second situation where estimating trees with hundreds of thousands of leaves is necessary is analyzing genes that evolve with duplications (called “gene families”). Gene duplication is very common in some organisms, such as plants; genes with more than a hundred different copies per species are not uncommon among plants. For this reasons, studying the evolution of large “gene families”, as attempted by the Thousand Transcriptome project (1KP) [40], requires estimating very large gene trees, sometimes with more than 100,000 leaves even when only a thousand species are sequenced.

3.2 Background: SATé-II

Since PASTA uses many of the algorithmic ideas of SATé and SATé-II, we start by describing the SATé-II algorithm. SATé-II uses an iterative strategy, and each iteration involves many steps. The first iteration begins with a starting tree, and subsequent iterations begin with the tree estimated in the previous iteration. Each iteration has the following steps:

1. The guide tree is used to divide the set of sequences into smaller subsets. In SATé-II, this decomposition is based on one of two strategies, both operating on an unrooted tree, and both parameterized by a maximum subset size M:

   Centroid edge decomposition: The branch that breaks the tree in two equal halves (or comes closes to breaking the tree into two equal halves) is the centroid branch. The centroid edge decomposition finds the centroid branch, divides the tree into two subsets by removing that branch, and then recurses on each half. The recursion stops when a subset has fewer leaves than M. Centroid decompositions makes subsets that are all roughly equal in size, and each subset consist of sequences that are close in the phylogenetic tree.

   Longest edge decomposition: This decomposition strategy is similar to the centroid decomposition, except, in each step, the longest branch is removed from the tree. This decomposition can result in varied subset sizes, but makes sure that each subset consists entirely of sequences that are evolutionary close to each other.

One of these two decomposition strategies is used to divide the set of sequences into subsets, each with at most M sequences.

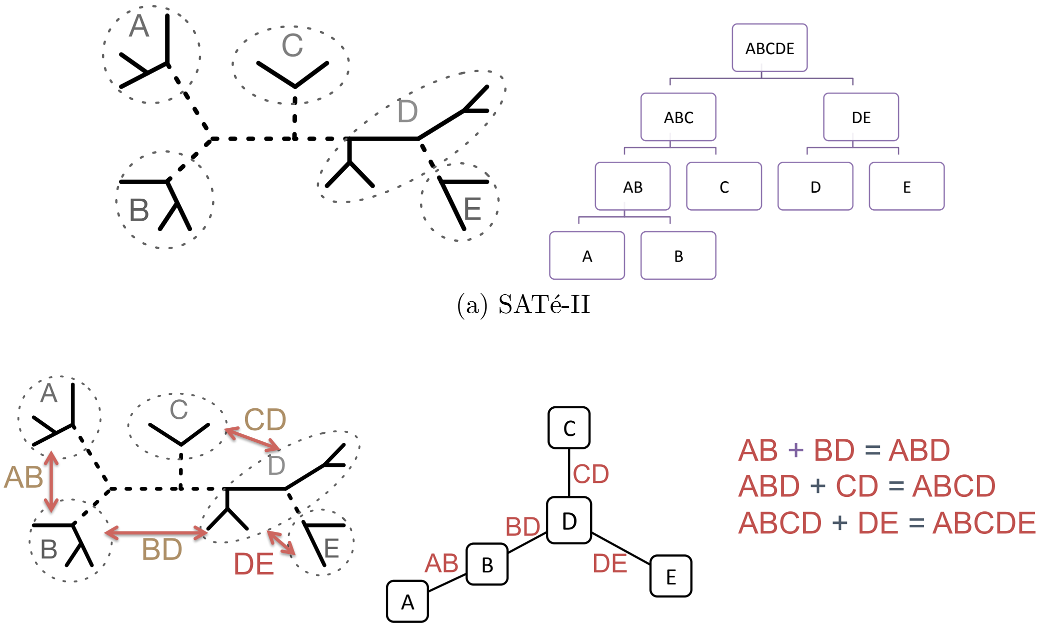
2. MSAs are independently estimated for every sequence subset using an external method of choice. By default, SATé-II uses Mafft [89] with the L-INS-i settings, which is based on the iterative refinement method incorporating local pairwise alignment information. This step produces an MSA for each subset of sequences.

3. Subset alignments are merged together hierarchically and according to the reverse order of edge removals in the decomposition step (see Fig. 3.1a). For aligning two alignments, various tools exist, and SATé-II uses OPAL [93] (or Muscle [88] if the dataset is very large).

4. Once the alignment is estimated, a new ML tree is estimated using RAxML, and this tree is used as the guide tree for the next step.

3.3 PASTA's algorithm

PASTA is an extension of SATé-II that uses the same iterative strategy. PASTA differs from SATé-II mainly in how it merges the subsets, but also in how the starting tree is computed, and some other minor design changes. As in SATé, PASTA uses the centroid edge dataset decomposition technique and



(a) SATé-II

(b) PASTA

Figure 3.1: **Merge step of SATé-II and PASTA**. (a) The centroid edge decomposition used in SATé-II first divides taxa to two subsets: $A \cup B \cup C$ and $D \cup E$, and then each subset is divided further until the dataset is divided into subsets A, B, C, D, and E. The order of centroid edge removals defines the hierarchy shown on the right, which is used for merging alignments. (b) In PASTA, a spanning tree is first computed from the guide tree such that each node of the spanning tree is a subset. On each branch of the spanning tree OPAL is used to merge two alignments and produce an alignment on the union of two subsets (we call these Type 2 sub-alignments). These type 2 sub-alignments are then merged together using transitivity to produce an MSA on all sequences.

computes MAFFT-L-INS-i [89] alignments on the subsets. While SATé uses Opal to hierarchically merge all the subset alignments into a single alignment, PASTA uses Opal only to merge pairs of adjacent subset alignments, producing overlapping alignments. The resulting collection of MSAs overlap with each other and have also other properties (described below) that enable us to merge these overlapping MSAs using transitivity. Thus, PASTA treats each resultant alignment as an equivalence relation and uses transitivity to merge these overlapping alignments (see Fig. 3.1b). We start by describing what we mean by transitivity and how it can be used to merge two alignments. We then describe the algorithmic steps of each iteration of PASTA, and finally show running time analyses of PASTA's merging step.

3.3.1 Transitivity merge of two alignments

Each MSA defines an equivalence relation on the letters within its sequences, so that two letters are in the same equivalence class *if and only if* they are in the same column [160]. For example, in Figure 3.2 (middle box in the bottom row), letters `A, A, T,` and `T` in the same column are considered equivalent, and the alignment creates an equivalence class. Given two overlapping alignments A_1 and A_2, we say they are compatible, if they induce identical equivalence classes on their shared sequences. For example, the two alignments shown in the last box of the bottom row of Figure 3.2 induce identical alignments for the shared sequences (in blue) and therefore are compatible.

Given two overlapping compatible alignments, we define their transitivity merge, as follows. We say that a and b are in the same equivalence class for the merged alignment if one of the following is true: (1) they are in the same equivalence class in either A_1 or A_2, or (2) there is some letter c such that a and c are in the same equivalence class in one alignment, and b and c are in the same equivalence class in the other alignment. In other words, we use transitivity to define the merger of two alignments (Fig. 3.2; bottom right corner). The resulting equivalence relation defines an MSA on $A_1 \cup A_2$, and is by construction compatible with both original alignments. We call this the transitivity merger and we can show that:

Corollary 3.3.1. *If we extract two overlapping sub-alignments A_1 and A_2 from an alignment A, then the transitivity merger of A_1 and A_2 will not include any false positive homologies (i.e., homologies not found in A) but can include false negative homologies (i.e., homologies in A that are not in the merger).*

Proof. It's easy to see that transitivity merger does not produce false positives (all relationships in the merger are either in the two sub alignments and hence true, or are mathematically inferred through transitivity). To see the possibility of false negatives, imagine that two letters are homologous in A, but one of them is in A_1 *only* and the other in A_2 *only*, and the remaining letters in that column are gaps (e.g., see the second column of the alignment in Fig. 3.2, bottom row middle box). Since the shared sequences between A_1 and A_2 have only gaps, transitivity cannot infer this homology, and therefore, the merger will have a false negative (e.g., see Fig. 3.2; bottom right corner). □

Computing the transitivity merger of two alignments is easy. We sweep the columns of both alignments simultaneously from left to right. If the two columns in A_1 and A_2 share a common letter (e.g., the i^{th} character of the j^{th} sequence) we simply merge the two columns into one column in the output; otherwise, the two columns have to have only gaps for the shared sequences, and these columns are added to the output alignment separately as two different columns (adding gaps where necessary).

3.3.2 Steps of a PASTA iteration

In the remaining of this section, we use the following notation:

S**:** The input set of sequences

s_i**:** A sequence in S (i.e., $s_i \in S$)

S_i**:** A subset of sequences in S (i.e., $S_i \subset S$); $S_1, \ldots, S_m$ partition S.

M**:** Maximum subset size (user input); thus, $|S_i| \leq M$ for $1 \leq i \leq m$

A **or** A_i**:** an alignment on S or S_i, respectively

T**:** A tree on the input set of sequences S

T^***:** A spanning tree with nodes representing subsets (i.e., nodes labelled S_i)

3.3.2.1 Six PASTA Steps

In PASTA, each iteration involves six steps (see Fig. 3.2). We provide a description of these steps in their default settings, in addition to a description

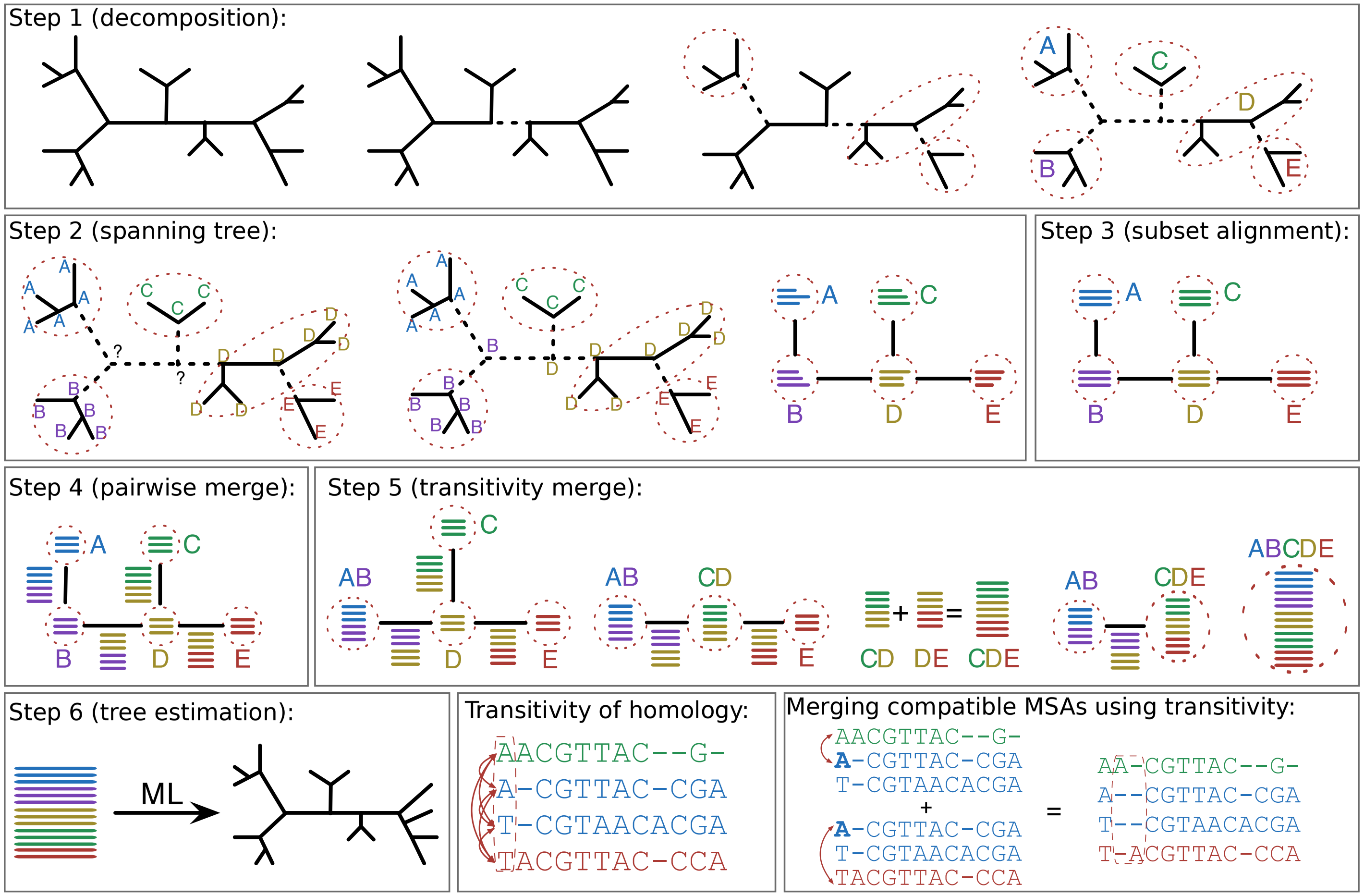


Figure 3.2: **Algorithmic design of PASTA**. The first six boxes show the steps involved in one iteration of PASTA. The last two boxes show the meaning of transitivity for homologies defined by a column of an MSA, and how the concept of transitivity can be used to merge two compatible overlapping alignments.

of how the starting alignment is built. All the numeric parameters mentioned below are just defaults and can be changed as desired by the user.

Step 0 - Default starting tree: First, we compute an alignment $A^{\mathcal{B}}$ of a random subset $S^{\mathcal{B}}$ of 100 sequences from S. We use HMMER [171, 172] to compute an Hidden Markov Model (HMM) that represents $A^{\mathcal{B}}$, and use this model and the hmmalign tool to align all sequences in $S - S^{\mathcal{B}}$ one by one to $A^{\mathcal{B}}$. This approach, which is the equivalent of a new alignment tool called UPP [173] with no decomposition, generates an alignment on S. We use this alignment and construct an ML tree using FastTree-II [119]. If the alignment step fails to produce an alignment on the full set of sequences (which can happen if HMMER considers some sequences unalignable), we randomly add the unaligned sequences into the tree.

Step 1 - Decompostion: We divide the set of sequences S into disjoint sets, $S_1, \ldots, S_m$, each with at most 200 sequences, using the current guide tree T and the centroid decomposition technique described above. The centroid decomposition procedure divides the set of leaves into subsets by a successive set of branch removal operations. Each time a branch is removed, the remaining leaves that go into the same set are connected to each other, but disconnected from the other leaves. When we stop dividing, each subset corresponds to a subtree of T that connects all leaves in that subset but includes no other leaves. The following corollary follows:

Corollary 3.3.2. *Let T be the guide tree and let $S_1, S_2, \ldots S_k$ be the subsets of taxa formed as a result of the centroid or longest edge decomposition. Then, if a node v in the guide tree is on the path between two nodes from the same subset (i.e., between $s_a \in S_i$ and $s_b \in S_i$), then it cannot also be on the path between any two nodes belonging to a different subset.*

Step 2 - Spanning tree: Given the current tree T, we compute a spanning tree T^* on the subsets $S_1, S_2, \ldots, S_m$ as follows. First we label every leaf s_x of T by the name of the subset it belongs to (i.e., s_x is labelled S_y iff $s_x \in S_y$). For every node v in T that is on a path between two leaves labelled S_y, we label v by S_y as well. Note that by Corollary 3.3.2, each node can be assigned only one label by this procedure because it can be only on one path between two leaves of the same label. However, it is possible for some nodes to be on no such path, and these will be left unlabelled. To label these remaining nodes, we propagate labels from nodes to unlabelled neighbors (breaking ties by using the closest neighbor according to branch lengths in the guide tree) until all nodes are labelled. We then collapse edges that have the same label at the endpoints. The result is a spanning tree on $S_1, S_2, \ldots, S_m$.

Step 3 - Subset alignment: We compute an MSA for each S_i using an existing MSA tool and refer to each such alignment as a *Type 1 sub-alignment*. By default, we use Mafft [89] with the L-INS-i algorithm to produce these alignments. L-INS-i is the most extensive version of Mafft and is based on an iterative refinement method incorporating local pairwise alignment information in its iterations. Experiments on SATé and SATé-II have found this version of Mafft to work better than alternative alignment methods for small datasets [30, 31].

Step 4 - Pairwise merge: Every node in T^* is labelled by an alignment subset for which we have a Type 1 sub-alignment from Step 3. For every edge (v, w) in T^*, we use OPAL [93] to align the Type 1 sub-alignments at v and w; this produces a new set of alignments, each containing at most 400 (more generally twice the maximum subset size) sequences, which are called *Type 2 sub-alignments*. We require that the merger technique used to compute Type 2 sub-alignments should not change the alignments on the Type 1 sub-alignments; therefore,

Corollary 3.3.3. *Type 2 sub-alignments induce the Type 1 sub-alignments computed in Step 2, and are all compatible with each other and with Type 1 sub-alignments.*

In other words, Type 2 sub-alignments retain all homologies in each of the two Type 1 sub-alignments and only add homologies between two Type 1 sub-alignments. More formally, when we merge two Type 1 sub-alignments A_i and A_j, we require that every homology in A_i and A_j be present in the Type 2 sub-alignment produced, and also require that every homology in the resulting Type 2 sub-alignment is either defined by A_i or by A_j, or is a homology between a letter $s_i \in A_i$ and a letter $s_j \in A_j$.

Step 5 - Transitivity Merge: Here we briefly describe how this step works, but complete details are given in Section 3.3.2.2. We use the spanning tree to merge all the Type 2 sub-alignments through a sequence of pairwise transitivity mergers into a multiple sequence alignment on the entire set of sequences. Note that each subset is part of at least one Type 2 alignment and each Type 2 alignment overlaps with at least one other Type 2 alignment (the adjacent edge in the spanning tree); thus, the final transitivity merger produces an alignment that includes all the sequences.

Step 6 - Tree estimation: If an additional iteration (or a tree on the alignment) is desired, we run FastTree-II to estimate a maximum likelihood tree on the MSA produced in the previous step. We remove all columns that have more than 99.9% gaps in the alignment obtained in Step 5; this filtering is used to improve the running time of the tree estimation step and has little impact on the eventual tree estimated from the data.

The six steps described above create one iteration of PASTA. The tree produced at the end is used as the guide tree for the next step. By default, PASTA runs for three iterations, but users can choose other stopping criteria.

3.3.2.2 Computing the transitivity merge

Recall that every node v in the spanning tree T^* computed in Step 2 is labelled by a subset S_v (i.e., a subset of the input sequence dataset on which we have a Type 1 sub-alignment A_v). In addition, during Step 4, we computed Type 2 sub-alignments for every pair of Type 1 sub-alignments whose alignment subsets are adjacent in the spanning tree T^*. We now define a label $L_{\mathcal{V}}(v)$ for every vertex v and $L_{\mathcal{E}}(e)$ for every edge e, as follows. For a node v in T^*, we define its label to be a set of subsets, and initially we set $L_{\mathcal{V}}(v) = \{S_v\}$ where S_v is the subset that node v corresponds to. For edge $e = (v, w)$ in T^*, we define its label to be a Type 2 sub-alignment and we set $L_{\mathcal{E}}(e) = A_{vw}$ where A_{vw} is the Type 2 sub-alignment we calculated in step 4 on $S_v \cup S_w$. For each node v, we also always keep an alignment A_v on the union of all the subsets in $L_{\mathcal{V}}(v)$.

We use T^* to guide a sequence of pairwise transitivity mergers, resulting finally in an MSA for the full set of sequences. The high-level algorithm is shown in Algorithm 3.1.

Algorithm 3.1 - Transitivity merge by spanning tree. *ContractTreeEdge* contracts and edge in the tree and return the new node created through edge contraction. *mergeAlignments* is defined below.

```
function MERGE($T^*$)
    while $|Nodes(T^*)| > 1$ do
        $e = (v, w) \leftarrow$ an arbitrary edge in $T^*$
        $u \leftarrow ContractTreeEdge(T^*, e)$
        $A_u \leftarrow mergeAlignments(A_v, A_w, L_{\mathcal{E}}(e))$
        $L_{\mathcal{V}}(u) \leftarrow L_{\mathcal{V}}(v) \cup L_{\mathcal{V}}(w)$
```

We contract branches of the spanning tree one by one until there is only one vertex left (see Fig. 3.2, step 5). Contracting an edge $e = (v, w)$ creates a new vertex u with a new label $L_{\mathcal{V}}(u) = L_{\mathcal{V}}(v) \cup L_{\mathcal{V}}(w)$, but does not modify the labels at the edges. The alignment associated with u is a merger of overlapping compatible alignments defined by e and its two endpoints (as computed by the *mergeAlignments* function defined below). Thus, the series of edge contractions corresponds to a series of transitivity merge operations. Since the results of applying multiple transitivity mergers does not depend on the order, the resulting alignment does not depend on the order in which branches are processed.

The following Invariants are maintained throughout Algorithm 3.1:

Invariant 1: Every A_v induces identical alignments as the Type 2 sub-alignments on pairs of subsets in $L_{\mathcal{V}}(v)$ and contains no homologies between sequences of two different subsets that cannot be inferred by transitivity

Invariant 2: For every edge $e = (v, w)$ with the label $L_{\mathcal{E}}(e) = (S_i, S_j)$, we can assert $S_i \in L_{\mathcal{V}}(v)$ and $S_j \in L_{\mathcal{V}}(w)$, and we have a Type 2 sub-alignment A_{ij} for $S_i \cup S_j$

Invariant 3: For every alignment subset S_i, there exists exactly one node v such that $S_i \in L_{\mathcal{V}}(v)$.

Note that initially these Invariants hold, since all vertices are labelled by only one alignment subset. Since the label of a new node u is set to the union of the labels of the two nodes removed, Invariant (3) always holds. Similarly, after we contract e, we ensure $L_{\mathcal{V}}(v) \subset L_{\mathcal{V}}(u)$, and therefore, if before contraction $S_i \in L_{\mathcal{V}}(v)$, then after contraction $S_i \in L_{\mathcal{V}}(u)$; it follows that Invariant (2) also always holds. To show that Invariant (1) always holds and to finish the description of the algorithm, we need to describe the *mergeAlignments* operation.

$mergeAlignments(A_v, A_w, L_{\mathcal{E}}(e))$**:** By Invariant (2), $L_{\mathcal{E}}(e)$ is a Type 2 sub-alignment A_{ij} on $S_i \cup S_j$, and by Invariant (3), $L_{\mathcal{V}}(v) \cap L_{\mathcal{V}}(w) = \emptyset$. Two scenarios are possible:

- $L_{\mathcal{V}}(v)$ and $L_{\mathcal{V}}(w)$ are singletons: In this case, *mergeAlignments* simply returns $L_{\mathcal{E}}(e) = A_{ij}$, which is a Type 2 sub-alignment on $S_v \cup S_w$. Invariant (1) follows from the requirement formalized in Corollary 3.3.3.

- $|L_{\mathcal{V}}(v)| > 1$ or $|L_{\mathcal{V}}(w)| > 1$: By Invariant (1) and (2), the alignments A_v and A_{ij} are overlapping compatible alignments, as are A_w and A_{ij}. Therefore, the three alignments A_{ij}, A_v, and A_w are all compatible, and so we can use transitivity merger described in Section 3.3.1 to merge them. To compute this transitivity merge, we first merge A_v and A_{ij}, and then we merge the resulting alignment with A_w (each step involves merging two overlapping compatible alignments using the approach described in Section 3.3.1, and the order of performing these two mergers does not matter). In each of these two steps, the two alignments overlap in a single alignment subset, and induce the same Type 1 sub-alignment on that subset. The result of each merger of these three MSAs creates a alignment on $L_{\mathcal{V}}(v) \cup L_{\mathcal{V}}(w)$, which *mergeAlignments* returns. Invariant

(3) still holds, as the only mechanisms used for merging alignments is transitivity.

Since the only mechanism used for merging Type 2 sub-alignments is the transitivity merge, from Corollary 3.3.1, we can infer that:

Corollary 3.3.4. *If all the Type 1 sub-alignments and Type 2 sub-alignments in PASTA are correct, then the final PASTA alignment has no false positive homologies, but can include false negative homologies.*

In other words, all the false positive homologies in the final PASTA alignment result either from running Mafft to produce Type 1 sub-alignments, or running OPAL to produce Type 2 sub-alignments, and not the transitivity merge steps. However, the false negatives can be introduced during the transitivity merge.

3.3.3 Running time

The order of traversing the spanning tree determines the order of a series of transitivity merge operations. The result of a series of transitivity mergers does not depend on the order of the operations. Thus, the final output of Step 5 (transitivity merge) does not depend on the order in which edges of the spanning tree are processed (i.e., an arbitrary order is shown in Algorithm 3.1). However, the order can impact the running time. An *arbitrary* order of edge contractions can result in a worst case $O(qm^2 + Lm)$ running time. However, if we merge sub-alignments using the reverse order of the centroid edge deletions, then the running time can be bounded, as follows.

Theorem 3.3.5. *Given m Type 1 alignments and $m-1$ Type 2 alignments, the algorithm to compute the transitivity merge of these alignments uses $O(qm \log m + Lm)$ time, where q is the maximum length of any sequence (not counting gaps) in any Type 1 alignment, and L is the length of the output alignment.*

Proof. We start by proving a lemma:

Lemma 3.3.6. *Let X, Y, and Z be disjoint sequence datasets, and alignments A and A' be compatible alignments on $X \cup Z$ and $Y \cup Z$, respectively (thus, A and A' induce identical alignments on Z). Let q be the length of the longest sequence in X, Y, and Z, and L be the total number of sites in A and A'. Then, we can merge alignments A and A' using transitivity in $O(L + q\ (|X| + |Y| + |Z|))$.*

Proof. To represent an alignment, we use a data structure with two elements: 1) the unaligned sequence and 2) a list of integers giving the position of each letter in the aligned sequences. Assume A has k columns, and A' has k' columns. We start by finding the sequences that belong to Z. For each shared sequence in Z, we find the columns that are non-gap in at least one shared sequence in A, and do the same thing for A' (we call these *shared columns*). Calculating shared columns can be done in $O(q|Z|)$, because our data structure for representing alignments has the list of column positions for each character of each sequence of Z.

Let k_s denote the number of shared columns. After computing shared columns we know that the final alignment will have $k + k' - k_s$ columns. We simultaneously iterate through the k columns in A and k' columns in A', and map these numbers to position numbers in the output alignment. We start at the leftmost position of both alignments, and keep a position in A (denoted by p), a position in A' (dented by p'), and a position in the output alignment (denoted by r). If both p and p' correspond to a shared column, we map both to r and increment all three. Otherwise, *w.l.o.g.* assume p is not a shared column; we map p to r and increment only p and r. At the end of this process, we have a mapping from columns of both input alignments to the columns of the output alignment, and this procedure takes $O(k + k' - k_s) = O(L)$ time. Finally, we build the output alignment by adding sequences from the original alignments, and by mapping their column indices using the mapping computed above. This step takes $O(q(|X| + |Y| + |Z|))$. Thus, the final running time is $O(L + q(|X| + |Y| + |Z|))$. Note that for a single gene alignment, typically $L << q(|X| + |Y| + |Z|)$, and therefore can be omitted from the analysis. □

We now continue with proof of Theorem 3.3.5. Let our dataset consist of N sequences, with each sequence of length at most q, and for the sake of simplicity, assume that our decomposition produces m subsets, all with equal sizes (note that centroid decomposition produces balanced subsets, so this assumption is justified). As described before, in Step 5, we chose an edge

$e = (v, w)$ from the spanning tree, contract that edge, and perform at most two transitivity merges: one between A_v and $L_{\mathcal{E}}(e)$, and another between the result of the first merger and A_w.

Based on the above lemma, the first transitivity merge will have a running time of $O(q(|L_{\mathcal{V}}(v)| + 2) + L)$, and the second merge will have a cost of $O(q(|L_{\mathcal{V}}(v)| + 2 + |L_{\mathcal{V}}(w)|) + L)$, and thus the cost of each edge contraction is $O(q(2 * |L_{\mathcal{V}}(v)| + |L_{\mathcal{V}}(w)|) + L)$. Note that each subset in the PASTA decomposition has at most M sequences and we don't increase M with the size of the dataset; therefore, the size of individual subsets can be replaced by a constant $O(1)$. Now, imagine the case where the spanning tree is a path. If we start merging from one end to the other end, we get the total running time of $O(q(3 + 4 + \ldots + m) + mL) = O(qm^2 + mL)$; however, we can improve on that. The important observation is that the spanning tree should be traversed such that transitivity mergers are between alignments with balanced number of sequences on each side.

The order in which edges are processed in PASTA is obtained by a recursive approach. Given the spanning tree, we divide it into two halves on the centroid edge, and thus obtain two roughly equal size subtrees. We process each half recursively using the same strategy, and thus get two single leaves at the endpoints of the centroid edge. Each leaf would represent the merger of all alignments in each half, and by construction they would have roughly equal size. We then contract the centroid edge, merge the two sides, and obtain the full alignment. If each half has roughly x sequences, the cost of the final edge contraction is $O(q(2x+x)+L) = O(3qx+L)$ (as shown before). If $f(x)$ denotes the cost of applying our transitivity merger on a spanning tree with x nodes, we have

$$f(2x) = 2f(x) + O(qx) + O(L)$$

which has an $O(x\log(x) + xL)$ solution. Therefore, our particular order of traversing the spanning tree results in a total cost of $O(qm\log(m) + mL)$. $\square$

Note that we fix maximum subset size q, and if we assume fixed alignment length (reasonable for a single gene dataset), then the running time of PASTA becomes $O(n\log n)$ for n sequences.

3.4 Experimental setup

We describe the datasets used, the methods that we compare to PASTA, our criteria of evaluation, and the computational resources used.

3.4.1 Datasets

We explore performance on both simulated and biological datasets and based on both nucleotide and amino acid sequences[2]. We explore performance on nucleotide datasets on both simulated and biological datasets. In simulations, we start by trees that are generated using a process (such as Yule process [174]) and then sequence data are simulated down each tree randomly but according to models of sequence evolution that include both indels and

[2] Datasets are available at `http://www.cs.utexas.edu/users/phylo/software/pasta`

substitutions. In simulations, the ground truth is known, and therefore true alignment and true (model) trees are used for evaluation.

3.4.1.1 Nucleotide

To explore performance on moderate-sized datasets, we used the 1000-sequence nucleotide datasets with average length 1000-1023 from the original paper studying SATé [30] that were generated using ROSE [175].

To explore performance on larger datasets, we simulated 10,000-sequence datasets using Indelible v. 1.03 [176] under three different rates of evolution (10 replicates each), with average sequence length 1000 (see Appendix A.1.2 for exact commands and parameters). These data are simulated with similar empirical statistics as the 1000-taxon 1000M2, 1000M3, and 1000M4 model conditions used in [30], and so we label these model conditions as 10000M2, 10000M3, and 10000M4. Empirical statistics of these model conditions are given in Table 3.1. 10000M2 has the highest rate of evolution (with a tree depth of 5 measured in the number of expected mutations per site); average hamming distance between pairs of sequences (p-distance) is 0.68, meaning that two sequences in average differed in 68% of their homologous sequences. 10000M3 and 10000M4 had lower rates of evolution (tree depth of 2.5 and 1, and average p-distance of 63% and 51%, respectively). The alignments had enough phylogenetic signal and trees estimated using true alignments had relatively high accuracy, as we will show.

To explore performance on ultra-large datasets (up to 1 million sequences), we used the million-sequence RNASim [177] dataset[3], with average sequence length 1556. RNASim is a simulator for RNA sequence evolution that was designed to simulate a complex molecular evolution process using a non-parametric population genetic model that generates long-range statistical dependence and heterogeneous rates. The simulated dataset using RNASim displays many of the properties of naturally observed RNA molecules in terms of both morphological variation and optimization difficulty. We randomly subsampled the million-sequence RNASim dataset to create datasets with 10k, 50k, 100k, and 200k sequences. For the 10k RNASim dataset we made 10 randomly subsampled replicates, but due to computational requirements, made only one replicate for the larger datasets.

In addition to the simulated data, we include three large biological datasets from the Comparative Ribosomal Website (CRW) [178]: the 16S.3 dataset (6,323 sequences of average length 1557, spanning three phylogenetic domains), the 16S.T dataset (7,350 sequences of average length 1492, spanning three phylogenetic domains), and the 16S.B.ALL dataset (27,643 sequences of average length 1371.9, spanning the bacteria domain). These datasets have curated reference alignments based on secondary and tertiary structures. Reference trees for the biological datasets were computed using RAxML [120] on the reference alignments and all edges with bootstrap support less than 75% were contracted; we also show results for other thresholds.

[3] available at `http://kim.bio.upenn.edu/software/csd.shtml`

3.4.1.2 Amino-acid

We used two different benchmark with real biological sequences for evaluation on amino acid sequences. One benchmark consisted of ten moderately large datasets (AA-10) which had curated MSAs (the eight largest BAliBASE datasets from [179] and IGADBL_100 and coli_epi_100 from [180]); These range in size between 320 to 807 sequences, and have average sequence lengths between 56.7 to 886.3. For these datasets, the curated alignment was used as the reference alignment.

We also used the 20 largest HomFam datasets that have between 10,099 to 93,681 sequences, but we omitted the "rhv" gene family due to the warning on the Pfam website[4] that the alignment is very weak[5] (thus retaining 19 datasets). The HomFam dataset were used previously to evaluate protein MSA methods on large datasets [181]. For this dataset, no global reference alignment was available; instead, Homstrad [182] reference alignments are available on very small subsets (5-20 sequences, median 7) of their sequences. These reference alignments are created based on structural properties, and are considered reliable. We used the alignment induced by these 5-20 sequences for evaluation.

[4] `http://pfam.xfam.org/family/Rhv`

[5] The exact warning as of 5/17/2015: "CAUTION: This alignment is very weak. It can not be generated by clustalw. If a representative set is used for a seed, many so-called members are not recognised. The family should probably be split up into sub-families. Capsid proteins of picornaviruses. Picornaviruses are non-enveloped plus-strand ssRNA animal viruses with icosahedral capsids. They include rhinovirus (common cold) and poliovirus. Common structure is an 8-stranded beta sandwich. Variations (one or two extra strands) occur."

Table 3.1: **Empirical statistics of reference alignments**. Columns show number of sequences, number of sites, percentage of gap characters, maximum and average p-distance. For 10k RNASim and Indelible, average over 10 replicates is shown. For 100k and 200k RNASim, we approximate p-distances. For HomFam, we also show number of sequences in the seed alignment.

	Dataset	# Sequences	# Sites	% gap	Max p-dist	Avg p-dist
CRW	16S.B.ALL	27,643	6,857	80	0.769	0.210
CRW	16S.T	7,350	11,856	87	0.900	0.345
CRW	16S.3	6,323	8,716	82	0.832	0.315
RNASim	10,000	10,000	8,637	82	0.616	0.410
RNASim	50,000	50,000	12,400	87	0.620	0.410
RNASim	100,000	100,000	14,316	89	≈ 0.62	≈ 0.410
RNASim	200,000	200,000	16,365	91	≈ 0.62	≈ 0.410
Indelible	M2	10,000	5,109	80	0.75	0.68
Indelible	M3	10,000	3,088	68	0.70	0.63
Indelible	M4	10,000	1,831	45	0.59	0.51
AA (10)	1GADBL_100	561	490	34	0.71	0.46
AA (10)	coli_epi_100	320	150	11	0.87	0.58
AA (10)	RV100_BBA0039	807	2,696	85	1.00	0.42
AA (10)	RV100_BBA0067	410	1,092	58	0.92	0.78
AA (10)	RV100_BBA0081	353	1,693	65	1.00	0.86
AA (10)	RV100_BBA0101	509	4,214	88	1.00	0.78
AA (10)	RV100_BBA0117	460	110	48	1.00	0.75
AA (10)	RV100_BBA0134	717	3,186	85	1.00	0.73
AA (10)	RV100_BBA0154	303	1,275	59	0.85	0.66
AA (10)	RV100_BBA0190	397	2,547	65	1.00	0.69
HomFam	aat	10 (25,100)	476	15	0.87	0.71
HomFam	Acetyltransf	6 (46,285)	229	29	0.87	0.75
HomFam	adh	5 (21,331)	375	0	0.47	0.36
HomFam	aldosered	7 (13,277)	386	19	0.79	0.57
HomFam	biotin_lipoyl	7 (11,833)	112	26	0.84	0.71
HomFam	blmb	6 (17,200)	344	30	0.90	0.79
HomFam	ghf13	10 (12,607)	626	25	0.84	0.72
HomFam	gluts	14 (10,099)	235	8	0.81	0.60
HomFam	hla	5 (13,465)	178	0	0.33	0.24
HomFam	hom	8 (12,037)	98	35	0.84	0.64
HomFam	myb_DNA-binding	5 (10,398)	61	12	0.77	0.59
HomFam	p450	12 (21,013)	512	20	0.87	0.79
HomFam	PDZ	6 (14,950)	110	15	0.84	0.69
HomFam	Rhodanese	6 (14,049)	216	31	0.89	0.76
HomFam	rrm	20 (27,610)	157	45	0.91	0.77
HomFam	rvp	6 (93,681)	132	19	0.76	0.63
HomFam	sdr	13 (50,157)	361	28	0.89	0.77
HomFam	tRNA-synt_2b	5 (11,293)	467	34	0.88	0.81
HomFam	zf-CCHH	15 (88,345)	39	25	0.85	0.65

Table 3.1 gives number of sequences, number of sites, percentage of gap characters, and maximum and average p-distance for all the biological and simulated datasets.

3.4.2 Methods

We show PASTA results based on the default settings and with three iterations. We compare PASTA to:

1. SATé-II version 2.2.7: We ran SATé-II for three iterations and with identical starting trees as PASTA. Due to the high computational costs of running OPAL on large datasets, we used Muscle for merging alignments inside SATé-II for datasets with 5,000 sequences or more, and otherwise we used the default settings in SATé-II.
2. Muscle version 3.8.31: run in default settings
3. Mafft version 7.143b [89]: default settings wherever it could run, and otherwise Mafft-PartTree (for RNASim 100K dataset, three replicates from the Indelible 10K 10000M3 dataset, and the CRW 16S.B.ALL dataset).
4. Clustal-Omega version 1.2.0 [181]: default settings
5. Staring tree/alignments: we also included the starting alignment and tree of PASTA as a separate method

We used FastTree-II version 2.1.5 to compute ML trees on each alignment. See Appendix A.1 in supplementary material for the exact commands.

3.4.3 Criteria

We measure the alignment accuracy, tree error, and running time. Alignment accuracy is measured the pairs score, as defined in Section 2.4.2. As noted before, for HomFam datasets, we measure the alignment error with respect to a very small number of reference "seed" sequences for which a reliable alignment is provided. To measure tree error, we report the False Negative (FN) rate (see Section 2.4.1). For AA datasets, since the seed alignments include only a handful of sequences, we measure only alignment accuracy and not tree error.

3.4.4 Computational platform

We ran (almost) all analyses on the Lonestar Linux cluster at TACC [183], and each run was given one node with 12 cores and 24 GB of memory. Since running time on Lonestar is limited to 24 hours, we were only able to run techniques that could finish in 24 hours (see below). However, PASTA and SATé-II are iterative techniques, and we allowed them to perform as many iterations (but no more than three) as they could complete within 24 hours. We report the wall clock time in all cases. For experiments on the million-sequence RNASim dataset, as well as for running SATé-II on RNASim 50k, we ran the methods on a dedicated machine with 256 GB of main memory and 12 cores and ran until an alignment was generated or the method failed.

3.5 Results

We start by reporting which methods could finish analyzing our datasets in the allotted time. We then report tree and alignment accuracy on nucleotide and amino acid datasets produced in the restricted time. We next show results for two cases where we relaxed the 24 hour time constraint. Running time results are presented next. We then move on to provide a series of experiments on varying parameters of PASTA. We next show results where a threshold other than 75% is used for building the reference biological trees. We end by showing results comparing PASTA to a new method called UPP [173], developed after we published PASTA.

3.5.1 Ability to complete analyses

We report which methods completed analyses within 24 hrs using 12 cores and 24 GB of memory. The technique for producing an starting tree failed to produce a full alignment on the 16S.T dataset, because HMMER considered one of the sequences unalignable. We added the missing taxon randomly into the tree obtained on the partial alignment produced by HMMER for that dataset. All methods completed on all datasets with at most 30,000 sequences, with the exception of Clustal-Omega, which was not able to run on the Indelible 10,000 M2 dataset. However,

- Clustal-Omega, Muscle, and SATé-II failed to complete on the RNASim datasets with 50,000 sequences or more.

- Mafft failed to complete on the RNASim dataset with 200k sequences.
- On 100k RNASim, PASTA finished two iterations in 24 hours, and on 200k, PASTA was able to complete one iteration and was the only method that could run (besides its starting tree).
- On the RNASim dataset with one million sequences, PASTA and its starting tree were the only methods that could run (see Section 3.5.8).

3.5.2 Results on nucleotide datasets

Indelible 10K dataset Tree error for ML trees on reference and estimated alignments for the all the large nucleotide datasets are shown in Figure 3.3. Unsurprisingly, ML trees computed on the true or reference alignments had the best accuracy in all cases. On the Indelible dataset with low rates of evolution (M4), all methods had accuracy close to what could be achieved using the true alignment. As the rate increased, the error for Crustal-Omega, Muscle, and eventually for Mafft increased. However, the starting tree approach, SATé-II and PATA continued to have low error. PASTA had the lowest tree error and was in fact very close to the tree obtained on the reference alignment even at the highest rate of evolution.

Table 3.2 shows alignment accuracy according to both the TC and pairs scores. On the Indelible datasets, PASTA had the most accurate alignments according to both measures of accuracy, and the difference between PASTA and other methods increased as the rate of evolution increased.

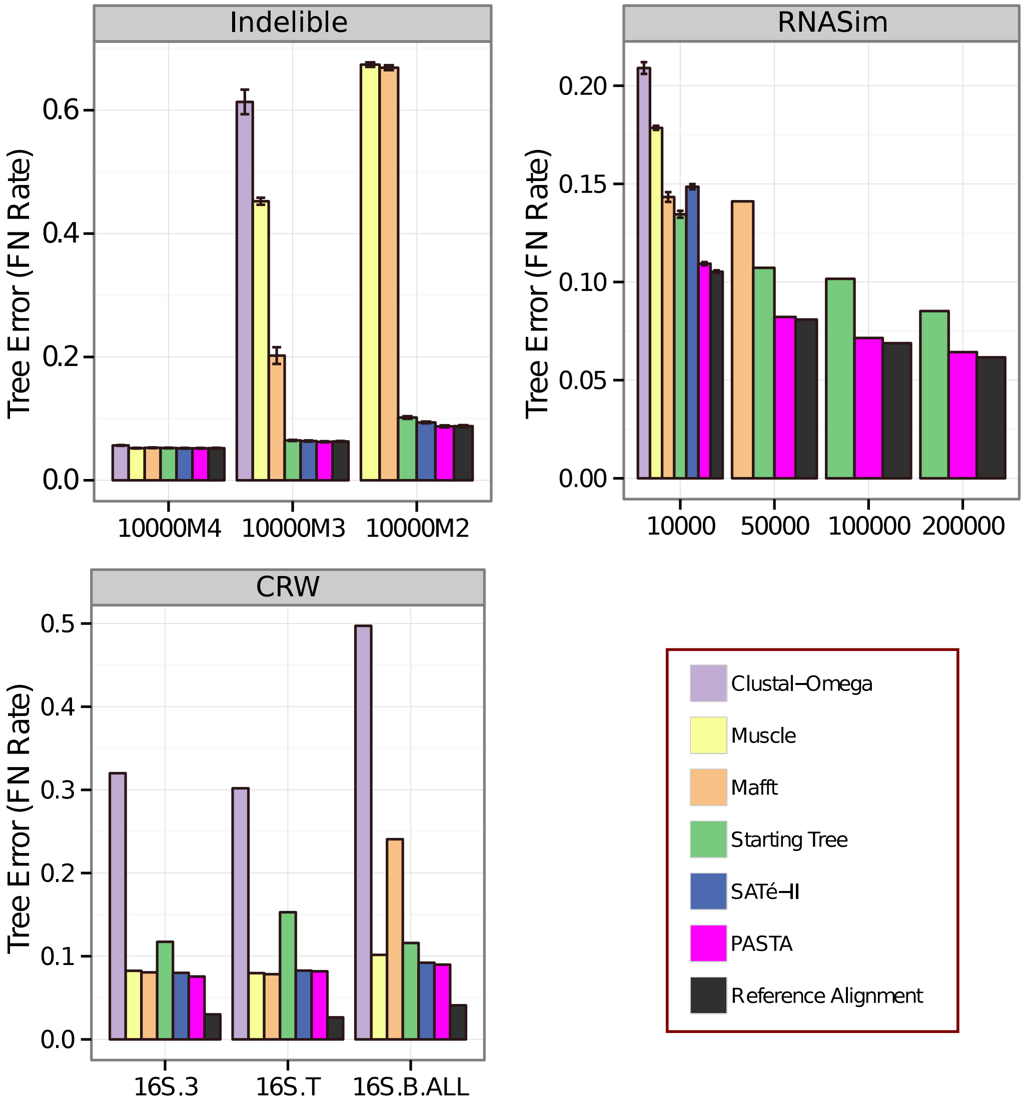


Figure 3.3: **Tree error rates on nucleotide datasets**. We show missing branch (also known as false negative or FN) rates for maximum likelihood trees estimated using FastTree-II, on the reference alignment as well as alignments computed using PASTA and other methods; results not shown indicate failure to complete within 24 hours using 12 cores on the datasets. Error bars show standard error over 10 replicates for all model conditions of the Indelible and the 10,000-sequence RNASim datasets.

Table 3.2: **Alignment accuracy on nucleotide datasets**. We show the number of correctly aligned sites (top) and the average of the SP-score and modeler score (bottom). X indicates that a method failed to run on a particular dataset given the computational constraints. "Initial" corresponds to the alignment approach used to obtain the starting tree of PASTA (HMMER failed to align one sequence in the 16S.T dataset) and Clustal-O stands for Clustal-Omega.

	Indelible - 10,000			RNASim				CRW (16S)		
	M4	M3	M2	10k	50k	100k	200k	.3	.T	.B.ALL
	Column (TC) score									
Clustal-O	160	10	X	13	X	X	X	12	0	1
Muscle	803	7	0	0	X	X	X	34	21	81
Mafft	337	13	0	28	30	26	X	75	85	15
Initial	422	106	18	11	15	5	4	33	X	24
SATé-II	977	758	792	35	X	X	X	**89**	60	87
PASTA	**987**	**920**	**1151**	**152**	**311**	**492**	**823**	71	**121**	**102**
	Pairs score (Mean of SP score and modeler score)									
Clustal-O	0.97	0.34	X	0.65	X	X	X	0.57	0.53	0.60
Muscle	**1.00**	0.12	0.01	0.35	X	X	X	0.74	0.67	0.66
Mafft	**1.00**	0.76	0.02	0.72	0.73	0.72	X	0.75	0.70	0.71
Initial	0.99	0.98	0.91	**0.87**	**0.88**	**0.87**	**0.88**	0.86	X	**0.95**
SATé-II	**1.00**	0.93	0.72	0.56	X	X	X	0.76	0.65	0.66
PASTA	**1.00**	**1.00**	**0.99**	0.85	0.85	**0.87**	0.86	**0.87**	**0.83**	0.94

RNASim datasets: PASTA trees had the best accuracy of all methods and had error that was really close to what could be achieved with the reference alignment (Fig 3.3). Interestingly, this experiment shows the effects of taxon sampling on tree estimation error. As the taxon sampling is increased, the tree error is reduced for all methods. For example, the starting tree had 14% error with 10K sequences, but as the taxon sampling improved, its error gradually dropped to 8% with 200K sequences. Similarly, PASTA started with 11% error for 10K sequences, but had only 6.4% error with 200K sequences. Thus, by producing highly accurate alignments of larger datasets, PASTA enables analyses that benefit from improved taxon sampling, which has been shown to be tremendously important in estimating accurate phylogenies [13, 167–169].

In terms of alignment accuracy (Table 3.2), for RNASim datasets, PASTA had by far the most accurate alignments of all methods tested according to TC, and its pairs scores were better than all other methods except for the starting alignment. The PASTA alignment had surprisingly high accuracy for datasets of this size: on the 200k dataset, its pairs score was 88% and more than 800 columns were recovered entirely correctly.

16S Biological data: On these biological datasets, Crustal-Omega had the highest tree error among all the methods, followed by the starting tree for two datasets and Mafft on the third dataset (Fig 3.3). On 16S.T and 16S.3, Mafft could be run in its default mode and had good accuracy; however, on 16S.B.ALL, the dataset size required that we use the PartTree command within Mafft and it had high error. Muscle, SATé-II, and PASTA had comparable accuracy on these data, with slight advantage for PASTA on 16S.B.ALL.

The gap between reference alignment and estimated alignments is not as small for these datasets as it was for the simulated datasets. However, here the reference tree *is* the ML tree estimated on the reference alignment; thus, to the extent that the tree on reference alignment has any error, it is simply due to the fact that FastTree is used here but RAxML is used for estimating the "reference tree" (and branches below 75% support are collapsed). While in simulated datasets the error of the RAXML ML tree on the true alignment was capturing the error introduced by ML reconstruction, here the error on reference alignments is simply measuring the difference between RAxML and FastTree. Thus, one expects very low error for the FastTree tree on the reference alignment.

On the 16S biological data, alignment accuracy generally favored PASTA (Table 3.2). Besides the 16S.T dataset, the starting alignment also had good alignment scores on these datasets but the other methods were generally less accurate. With respect to TC scores, on 16S.B.ALL and 16S.T, PASTA had the highest accuracy, but on 16S.3, SATé-II had the highest accuracy (followed by Mafft and PASTA).

1000-sequence datasets: Tree error and alignment accuracy on the 1000-sequence datasets are shown in Figure 3.4. The model conditions are labelled by the gap length distribution (M for medium length, S for short, and L for long), and increase in difficulty (higher rates of indels and substitutions) from left to right. Note that ML on the true alignment is the most accurate method in terms of tree accuracy, but that PASTA and SATé-II have almost indistinguishable accuracy on these data and both come very close to the ML tree on the true alignment. Both SATé-II and PASTA also have higher pairs score accuracy compared to other methods.

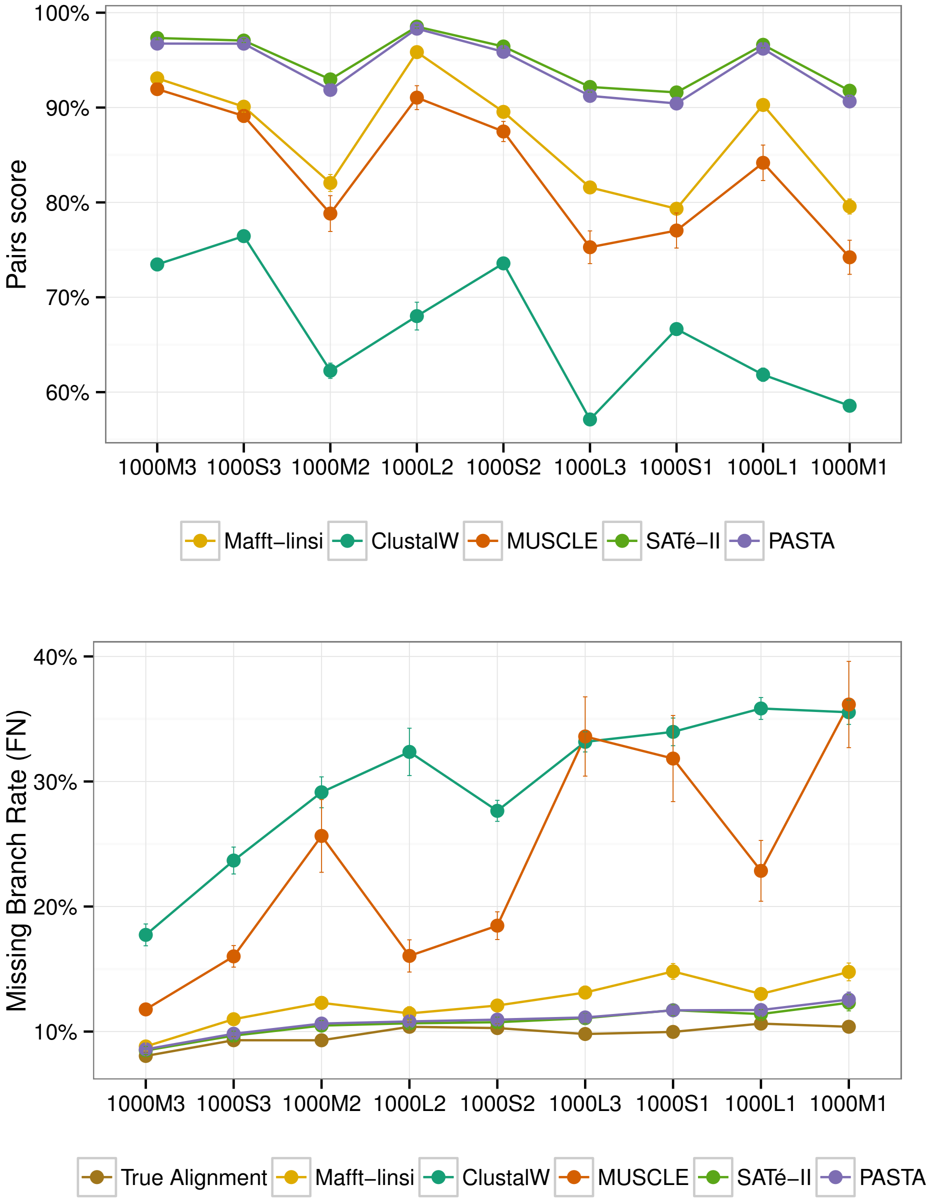


Figure 3.4: **Tree error and alignment accuracy on 1000-taxon datasets**. We report the pairs score (top) and missing branch rate (bottom) of alignments and trees estimated by FastTree-2 on the true alignment, and on alignments estimated using PASTA, SATé-II, and other alignment methods, on challenging 1000-taxon datasets from [30].

Table 3.3: **Alignment accuracy on AA datasets**. We show TC (the number of correctly aligned sites, left) and the pairs score (the average of the SP-score and modeler score, right). X indicates that a method failed to run on a particular dataset given the computational constraints. “Initial” corresponds to the alignment approach used to obtain the starting tree of PASTA (HMMER failed to align one sequence in the 16S.T dataset). All values shown are averages over all datasets in each category.

	Column (TC) score			Pairs score		
method	AA-10	HomFam-17	HomFam-2	AA-10	HomFam-17	HomFam-2
Clustal-O	78	88	29	**0.76**	0.72	0.71
Muscle	48	51	X	0.70	0.52	X
Mafft	81	**103**	32	**0.76**	0.75	0.79
Initial	54	95	16	0.75	0.71	0.81
SATé-II	**83**	73	X	0.75	0.64	X
PASTA	80	102	**36**	**0.76**	**0.78**	**0.83**

3.5.3 Alignment accuracy on AA datasets

Table 3.3 shows alignment accuracy on the AA datasets. Due to dataset sizes, Muscle and SATé-II failed to complete on two of the HomFam datasets, so we separate out the results for these two datasets from the remaining 17 HomFam datasets.

PASTA had the best pairs score or was tied for the best pairs score for both HomFam and AA-10 datasets. The difference between PASTA and other methods was more substantial for HomFam datasets. Mafft had the best TC score for HomFam(17), but PASTA was very close (103 versus 102 columns). For HomFam(2), PASTA had the best TC score and Mafft was a close second. On AA-10 datasets, SATé-II had the best TC score and was closely trailed by Mafft and PASTA.

3.5.4 Comparisons on larger datasets

In the previous section we reported the results on runs that could finish in 24 hours given 12 cores. Here, we report two extra analyses where we allowed methods to run on dedicated machines for longer times.

Comparison to SATé-II on 50,000-taxon dataset. SATé-II could not finish even one iteration on the RNASim with 50,000 sequences running for 24 hours and given 12 cores on TACC. However, we were able to run two iterations of SATé-II on a separate machine with no running time limits (12 Quad-Core AMD Opteron(tm) processors, 256GB of RAM memory). On this machine, two iterations of SATé-II took 137 hours, compared to 10 hours for PASTA. However, the resulting SATé-II alignment recovered only 30 columns entirely correctly while PASTA recovered 311 columns. The pairs score of SATé-II was extremely poor (38.2%), while PASTA was quite accurate (81.0%). The tree produced by SATé-II had higher error than PASTA (12.6% versus 8.2% FN).

Results on 1M sequences We also attempted to run PASTA on the full 1,000,000-sequence RNASim dataset on the same dedicated machine with 12 cores and 256 GB of memory. PASTA completed one iteration in 15 days, and produced an alignment with 81.5% pairs score error and a tree that had only 6.0% tree error; this was only 0.4% more than the tree error for FastTree-II run on the known true alignment. The PASTA starting tree, for comparison, had 8.4% tree error.

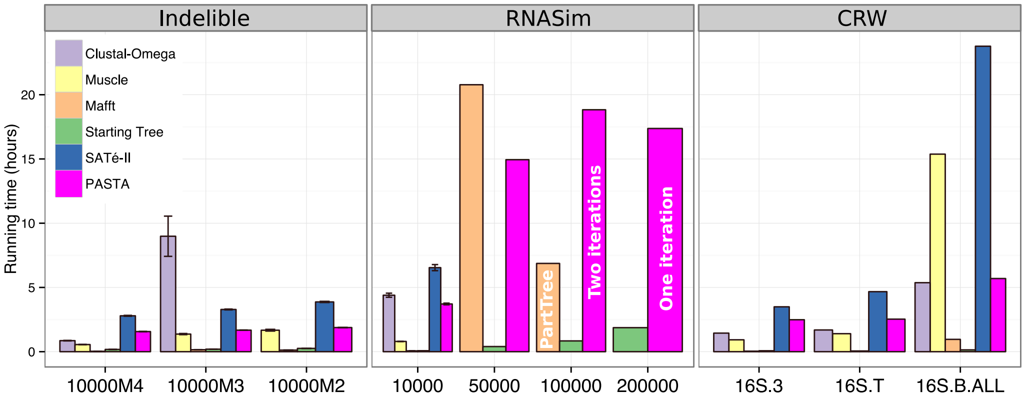


Figure 3.5: **Alignment running time (hours)**. Note that PASTA was run for three iterations everywhere, except on the 100,000-sequence RNASim dataset where it was run for two iterations, and on the 200,000-sequence RNASim dataset where it was run for one iteration. Mafft was run in default mode, except for the 100,000-sequences where PartTree was used.

3.5.5 Running Time

Figure 3.5 compares the running time (in hours) of different alignment methods. Note that PASTA was faster than SATé-II in all cases, and could analyze datasets that SATé-II could not (i.e., the RNASim datasets with 50k or more sequences). Comparisons to other methods show that PASTA was not always faster than other methods, but was able to complete its analyses of all datasets within the 24hr time limit, whereas other methods (except the starting tree) were unable to complete analyses on the largest datasets.

Figure 3.6a presents a detailed running time comparison of PASTA and SATé-II on two specific model conditions of RNASim dataset. Note that PASTA and SATé use similar iterative divide-and-conquer techniques, but differ in how the subset alignments (each on only 200 sequences) are merged

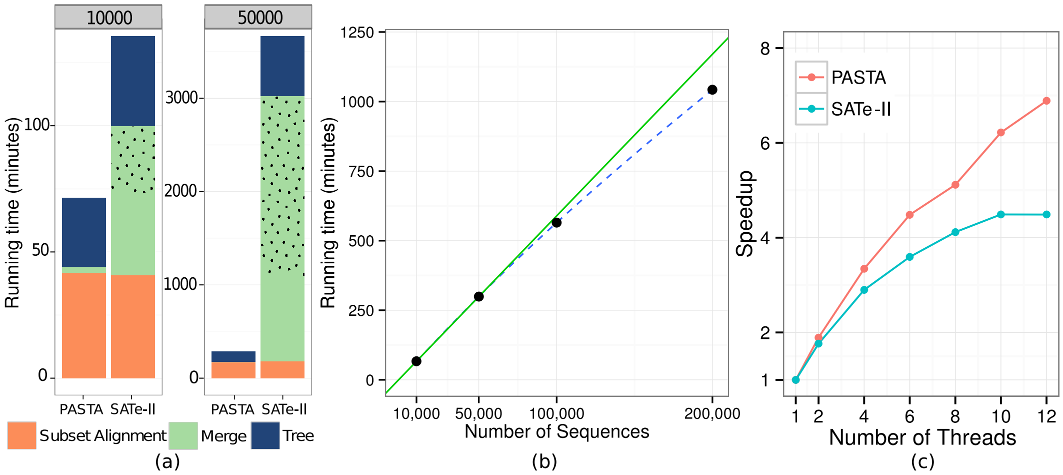


Figure 3.6: **Running time comparison of PASTA and SATé-II**. (a) Running time profiling on one iteration for RNASim datasets with 10k and 50k sequences (the dotted region indicates the last pairwise merge), (b) Running time for one iteration of PASTA with 12 CPUs as a function of the number of sequences (the solid line is fitted to the first two points), and (c) Scalability for PASTA and SATé-II with increased number of CPUs.

together into an alignment on the full set of sequences. Merging subset alignments (and the last pairwise merge, shown in the dotted area) was the majority of the time used by SATé-II to analyze the 50k RNASim dataset, but a very small fraction of the time used by PASTA. PASTA uses transitivity for all but the initial pairwise mergers, and therefore scales well with increased dataset size, as shown in Figure 3.6b (the sub-linear scaling is due to a better use of parallelism with increased number of sequences). Finally, Figure 3.6c shows that PASTA is highly parallelizable, and has a much better speed-up with increasing number of threads than SATé does. While PASTA has much improved parallelization, its parallelization does not quite scale up linearly, because FastTree-II does not scale up well beyond 3 threads.

Table 3.4: **Effect of the starting tree on final PASTA alignments and trees**. Alignment accuracy and tree error are shown for PASTA with various starting trees, after one iteration (top) and three iterations (bottom) for one replicate of the 10k RNASim dataset. The error for starting tree is also shown.

Initial Tree		Alignment Accuracy		Tree Error
method	Error (FN)	Pairs score	TC	FN
After One Iteration				
Random	100.0%	79.9%	2	52.3%
Mafft-PartTree	28.7%	**87.0%**	126	11.7%
Starting Tree	12.4%	86.8%	**138**	**10.5%**
True Tree	**0%**	86.1%	133	**10.5%**
After Three Iterations				
Random	100.0%	90.4%	138	11.0%
Mafft-PartTree	28.7%	83.9%	144	10.7%
Starting Tree	12.4%	88.8%	145	10.7%
True Tree	**0%**	**90.8%**	**150**	**10.5%**

3.5.6 Impact of varying algorithmic parameters.

3.5.6.1 Starting tree

We compared results obtained using four different starting trees: a random tree, the ML tree on the Mafft-PartTree alignment, PASTA's default starting tree, and the true (model) tree. Table 3.4 shows results of PASTA starting from one of these trees and after one or three iterations. After one iteration, PASTA alignments and trees based on our starting tree or true tree had roughly the same accuracy, and the starting tree based on Mafft-PartTree resulted in only a slightly worse tree (1% higher FN rate) despite the fact that the starting tree had substantial error (28.7%). However, using a random tree resulted in much higher tree error rates (52.3% error), and alignments that were also considerably less accurate (about 7% according to pairs score). Only two

Table 3.5: **Impact of alignment subset size**. We report tree error and alignment accuracy on one replicate of the 10k RNASim dataset and also on the 16S.T dataset, using three iterations of PASTA in which we explore the impact of changing the subset size from 200 (the default) to 100 and 50; other parameters use default values. Boldface indicates the best performance.

Dataset	Subset Size	Tree Error	Alignment Accuracy		Running Time
		FN	Pairs score	TC	(Seconds)
RNASim 10k	200	10.7%	**88.8 %**	145	13,478
RNASim 10k	100	**10.4%**	87.4 %	185	8,235
RNASim 10k	50	10.7%	88.6%	**210**	**6,015**
16S.T	200	8.2%	**82.7 %**	121	9,120
16S.T	100	8.1%	82.0 %	125	7,086
16S.T	50	**7.9%**	79.0%	**129**	**5,780**

columns were aligned correctly when starting from a random tree, whereas with estimated starting trees between 126 to 138 columns were recovered correctly.

Interestingly, after three iterations of PASTA, no noticeable difference could be detected between results from various starting trees. The tree error was only very slightly higher for the random tree (0.3%) compared to starting from the default starting tree, and alignment accuracy was identical according to pairs score and very close according to TC. Thus, PASTA is robust to the choice of the starting tree and even a random starting tree results in high accuracy in the final tree as long as enough iterations are used.

3.5.6.2 Subset size

We also evaluated the impact of changing the alignment subset size and using smaller subsets (50 or 100). Table 3.5 shows the results of these

Table 3.6: **Impact of using Muscle versus Opal as the alignment merger technique.**. We report results on one replicate of the 10K RNASim dataset, using three iteration of PASTA using all default settings for other algorithmic parameters. We report the missing branch rate for the tree error, and two accuracy measures for alignments: the Total Column (TC) score, and the pairs score. Boldface indicates the best performance on this data.

Parameters	**Tree Error**	**Pairs Score**	**TC score**	**Run Time (sec)**
Opal Type 2 merger	**10.7%**	**88.8%**	**145**	**13,478**
Muscle Type 2 merger	11.2%	73.9%	136	14,884

analyses for two datasets. Results are consistent across both datasets; these analyses showed that using alignment subsets of only 50 sequences improved the TC score and running time substantially, and only slightly changed the pairs score or tree error score. Although these analyses were performed only for two datasets, they suggest the possibility that improved results might be obtained through smaller alignment subsets. A more thorough study of this factor is left for future research.

3.5.6.3 Choice of tool for merging alignments

We also explored the difference between PASTA using Opal (the default merger) or Muscle for merging Type 1 alignments into Type 2 alignments; see Table 3.6. This comparison showed that OPAL can result in better final alignments and trees compared to Muscle. For example, on the 10,000 RNASim dataset, PASTA with OPAL and with MUSCLE had tree errors of 10.7% and 11.2%, a slight improvement, but that alignment accuracy changed substantially, especially when considering the average of the SP and modeler scores.

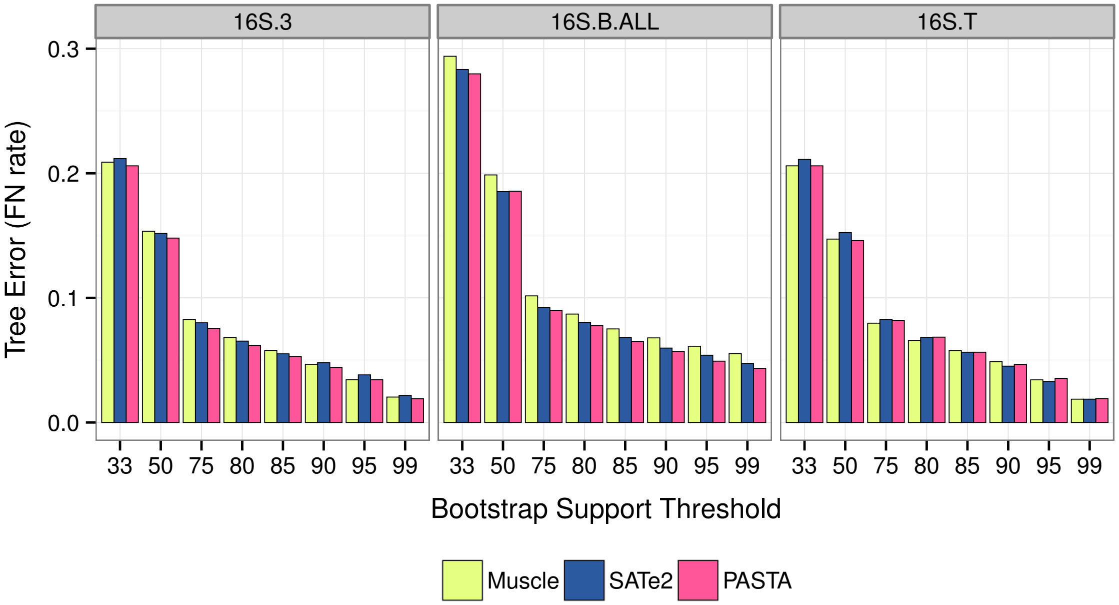


Figure 3.7: Tree error (FN) rates on biological datasets as a function of bootstrap threshold chosen to define the reference tree.

3.5.7 Varying the bootstrap threshold for reference tree

Performance on biological datasets is challenging to evaluate because the true phylogeny cannot be known with certainty. We used a set of reference alignments estimated based on secondary structures (available from the CRW website [178]), and estimated a ML tree on each dataset using RAxML with bootstrapping. We then contracted all branches with low support to obtain the reference tree. We have so far reported results based on contracting branches with less than 75% support; here, we show results with other thresholds. Note that the higher the thresholds, the more branches will be collapsed.

Figure 3.7 reports results only for Muscle, SATé-II, and PASTA, the three methods with the best performance on these biological datasets for thresholds ranging from 33% to 99%. Note that the difference in performance is small in most cases, and that relative performance generally does not change much as a function of threshold. Typically PASTA has slightly better tree accuracy than both SATé and Muscle, but there are a few cases where the relative performance changes. However, there are a few thresholds and datasets where the relative ordering between SATé, PASTA, and Muscle changes, so that Muscle is tied for best, or PASTA is less accurate than SATé.

3.5.8 Comparisons to UPP

A new method called UPP that was developed after we performed these experiments was also able to analyze the datasets analyzed here with high accuracy [173]. UPP, which algorithmically has some similarities to the PASTA starting tree, tends to estimate alignments that have better pairs score compared to PASTA, but PASTA has much better TC score and lower tree error [173]. For example, PASTA had 86% pairs score on 200K RNASim dataset whereas UPP had 87.5% accuracy. However, PASTA recovered 823 columns correctly, whereas UPP recovered only 5 columns correctly. The PASTA tree had 6.4% error whereas UPP had 8.5% tree error. Like PASTA, UPP was also able to complete on the 1M RNASim dataset, and it took 12 days (compared to 15 days for PASTA). On this dataset, UPP resulted in a better alignment pairs score compared to PASTA (87.2% versus 81.5% pairs score) but a tree that had 7.6% tree error (PASTA had 6.0% error).

3.6 Summary and discussion

The key algorithmic contribution in PASTA is the use of transitivity to align sequences on a guide tree. PASTA uses the centroid edge decomposition strategy of SATé to produce non-overlapping subset alignments, but creates overlapping alignments using a spanning tree, and completes the alignment using transitivity applied to these overlapping alignments. The new merging technique addresses computational limitations in SATé and also improves the accuracy of the alignments generated. PASTA is fast and scales well with the number of threads, so that datasets with even 200,000 sequences can be analyzed in less than a day with 12 threads. PASTA was able to align a dataset with a million sequences in 15 days.

PASTA is implemented in Python and the code is publicly available in open source form at `https://github.com/smirarab/pasta`. Since its publication, PASTA has gained a user base, and currently we are using PASTA for aligning the new 1KP dataset [40] with more than 1000 species[6] and some gene families that have hundreds of thousands of sequences.

While PASTA has excellent accuracy in the experiments we performed, there are datasets that PASTA is not designed to handle. A new study has shown that the accuracy of PASTA degrades if the dataset includes fragmentary data [173]. This finding is not surprising, because alignment of fragmentary sequences requires the use of local alignment techniques, where the

[6] list of species available at `http://www.onekp.com/samples/list.php`

assumption is that a sequence will align only partially to other sequences. All the techniques used inside PASTA, and specifically the technique used for aligning subsets are global alignment techniques; i.e., they assume that all the sequences align to each other from beginning to end. For PASTA to work well with fragmentary sequences, we need to make sure that fragmentary sequences are dealt with differently from the remaining sequences. For example, PASTA could initially align only full-length sequences and only then add fragmentary sequences to this “backbone” alignment using local alignment; this strategy is similar to what has been used in some new alignment tools [173, 184].

Another shortcoming of PASTA is that it can produce long and gappy alignments. This is related to the results presented in Corollary 3.3.4. Thus, while typical alignment tools tend to over-align [15, 31, 160], PASTA tends to under-align. For practical purposes, it suffices to remove columns from PASTA that are extremely gappy (e.g., those with more than 99.9% gaps). Packaged with PASTA are scripts that help the user with these kinds of alignment post-processing tasks. PASTA also introduces a new format for saving alignments on disk, so that very long and gappy alignments only take a fraction of the space they would take using traditional formats.

Chapter 4

Statistical Binning[1]

Species trees are important tools for understanding evolution, and have applications to comparative genomics [185], orthology detection [58–61], studying biodiversity analysis [186], and many other areas of biological study. Gene trees can be different from the species tree and a major cause of such discordance is incomplete lineage sorting [18, 137] (see Section 2.2.2). Estimation of species trees taking into account ILS can require a large number of genes [20]. Analyses of whole genomes for estimating the species tree are becoming feasible [12, 187–189], and arguably are necessary for resolving phylogenies of rapid species radiations, where very high levels of ILS are expected [39, 77].

[1] Parts of this chapter have appeared in the following papers:

1. Siavash Mirarab, Md. Shamsuzzoha Bayzid, Bastien Boussau, and Tandy Warnow. Statistical binning enables an accurate coalescent-based estimation of the avian tree. *Science*, 346(6215), 2014
2. Md. Shamsuzzoha Bayzid, Siavash Mirarab, Bastien Boussau, and Tandy Warnow. Weighted Statistical Binning: enabling statistically consistent genome-scale phylogenetic analyses. *PLoS ONE*, 10(6):e0129183, 2015
3. Siavash Mirarab, Md. Shamsuzzoha Bayzid, and Tandy Warnow. Evaluating summary methods for multi-locus species tree estimation in the presence of incomplete lineage sorting. *Systematic Biology*, page syu063, 2014

In all three cases, SM and his supervisor, TM, designed the method, designed the studies, and wrote the papers (with comments from the other authors), and SM implemented the methods. SM and MSB ran experiments (SM lead 1 and 3, and MSB lead 2). BB generated the simulated data, and all authors helped analyze results.

ILS can reduce accuracy of concatenation-based estimations of species trees where all the data are put together in one supermatrix and analyzed without modeling discordance [32, 133–135]. Coalescent-based species tree estimation methods have been developed to estimate a species tree given data gathered from multiple genes (see Section 2.3.3). The most widely used coalescent-based methods, called summary methods (see Section 2.3.3.2), are based on a two step pipeline: first each gene tree is estimated separately from the gene sequence data, and then, the species tree is reconstructed by summarizing these estimated gene trees. The method used in the second step can be statistically consistent under the multi-species coalescent (MSC) model [69] (see Section 2.2.2.2), and this two step pipeline can have good accuracy when gene trees have good accuracy. However, as we demonstrate in this chapter, this pipeline is sensitive to gene tree estimation error in the first step.

In this chapter, we propose a new method for improving the quality of the estimated gene trees. Our proposed pipeline, called statistical binning [190, 191], uses bootstrapping (see Section 2.3.2.5) to evaluate whether two genes are likely to have the same true tree topology, then groups genes into bins using these pairwise comparisons and a minimum vertex coloring optimization problem. It then estimates a tree on each bin by concatenating the data in that bin, and uses these trees as input to the preferred coalescent-based summary method. We evaluate statistical binning on a large set of simulated and biological data and show that it improves the accuracy of gene trees and the species tree measured in various ways.

We start this chapter by discussing effects of gene tree error on species tree estimation, specifically in the context of the avian phylogenomics project (Section 4.1). We then describe the statistical binning pipeline and present some theoretical results about its statistical consistency in Section 4.2. In Section 4.3, we describe the experimental setup used for evaluating binning, and then show results of evaluating statistical binning using simulated (Section 4.4) and real biological data (Section 4.5). We then finish by a discussion of results and directions for future research in Section 4.6.

4.1 Sensitivity of summary methods to gene tree error

A phylogenomic pipeline that uses a coalescent-based summary method begins with sequence alignments on different loci, estimates gene trees on each locus, and then combines the estimated gene trees into an estimated species tree using the summary method. Summary methods are by far the most frequently used method for species tree estimation, and have been used to analyze various biological datasets [16, 77, 150, 189, 192–194]; however, for some datasets, the summary methods have not been able to produce highly supported trees [195], even with a large quantity of data [16]. Simulation studies show that species trees estimated with summary methods can be less accurate than species trees estimated with concatenation, even in the presence of substantial ILS [17, 38, 135, 196]. A main reason for this disparity in performance is poor phylogenetic signal in individual genes, which is a potential problem for coalescent-based summary methods [17, 37]. Moreover, many realistic biological conditions (including short branches in gene trees) make completely accurate gene tree estimation from limited sequence data highly unlikely [197].

Phylogenomic analyses can utilize very large numbers of genomic loci to estimate the species tree, but genome-scale datasets can contain loci that have reduced phylogenetic signal so that their estimated gene trees have reduced bootstrap support (BS) [39]. While it is not known how summary methods are impacted when only some of the loci have low signal, some studies have showed that coalescent-based summary methods have reduced accuracy on datasets where all the gene sequence alignments are short [17, 37]. This challenge confronted the avian phylogenomics project [39] (which included us), where a large number of genes (14,446) were available for coalescent-based analyses, but these genes did not have enough signal for accurate gene tree estimation. The challenges faced on the avian dataset, in addition to our observations on simulation studies, motivated us to develop the statistical binning pipeline.

In the rest of this chapter, we first present simulation results that highlight the impact of gene tree estimation error on species tree error. We then further motivate the problem of gene tree estimation error by describing challenges faced on the avian project.

4.1.1 Gene tree error in simulations

In Section 4.3, we will provide extensive simulation results that show the impact of gene tree error and how our proposed pipeline reduces it. Here, just to motivate the development of statistical binning pipeline, we report on a separate simulation study that we have performed and showed the impact of gene tree error on species tree estimation [38]. The simulation procedure used in this study is similar to what we will describe in detail in Section 4.3.1. To avoid repetition, we don't explain the same procedure here in detail, and only describe the high-level procedure.

In our simulations, we generated sequence data under a procedure analogous to the GTR+MSC model, defined in detail under Section 4.2.2. Gene trees were generated from a fixed species tree under the MSC model, and sequence data were simulated under the GTR model of sequence evolution (see Section 2.1.2.1). Sequence data were then either analyzed directly using concatenation, or were used to estimate the gene trees, which were then used as input to summary methods. We show results for two summary methods: MP-EST [134] and greedy (both described more under Section 4.3.2), but we see similar trends with other summary methods (see [38]). We vary the gene sequence length from 250bp to 1500bp to generate model conditions with varying levels of phylogenetic signal per gene; this is to control the level of gene tree estimation error, which we measure as the average RF distance between true gene trees and the estimated gene tree. Using this procedure, we obtain five model conditions, distinguished by their average gene tree error: 0% error (using true simulated gene trees), 12% error (1500bp sequences), 16% error (1000bp sequences), 27% error (500bp sequences), and finally 42% error (250bp sequences). For each model condition, we simulated 20 replicates.

In Figure 4.1(a) we show the species tree error as a function of the

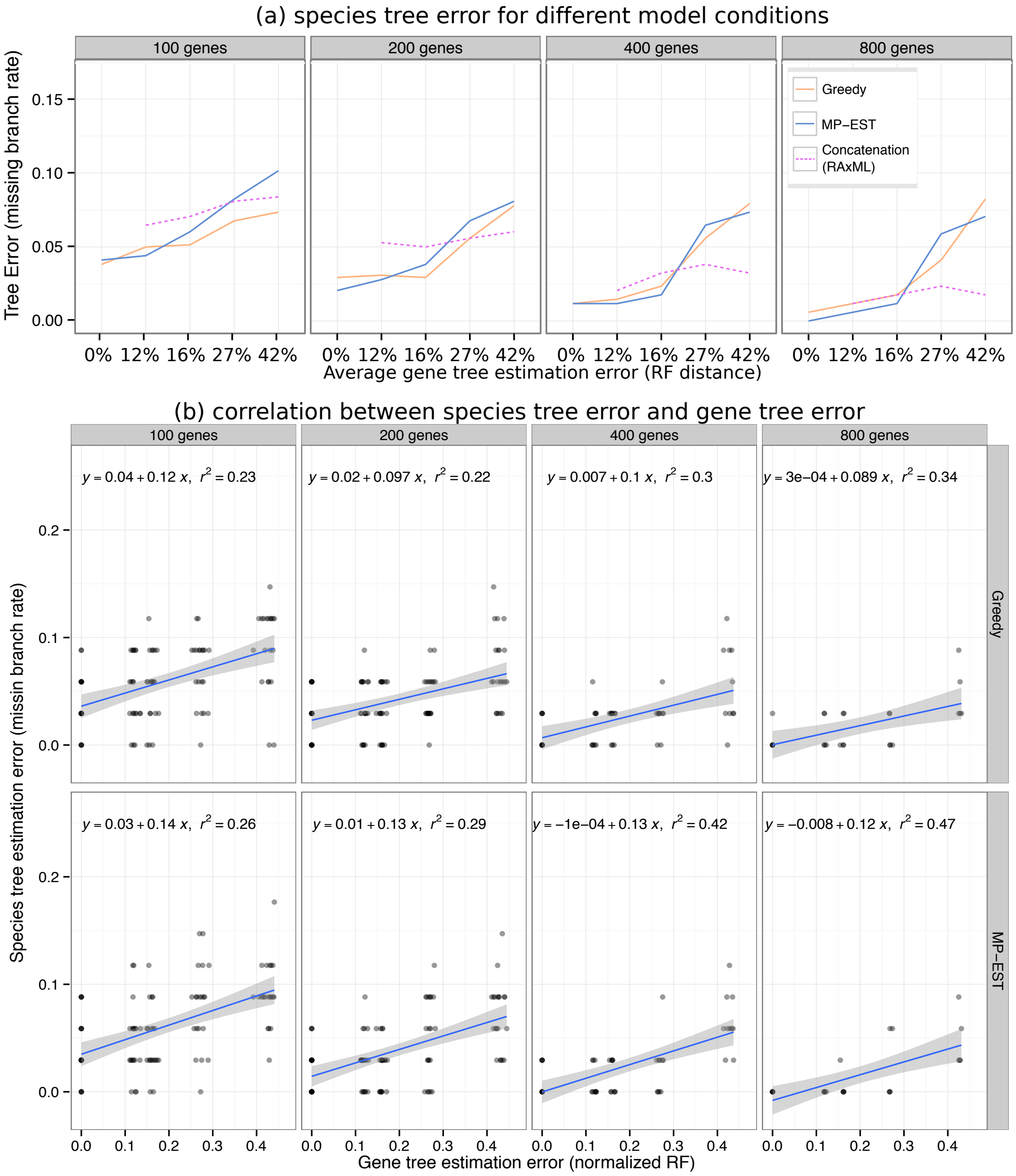


Figure 4.1: **Impact of the gene tree error on species tree estimation**. (a) Lines show average species tree error over 20 replicates for different number of genes (boxes) and for the five different model conditions characterized by average gene tree error (x-axis). (b) We show the correlation between gene tree error (x-axis) and species tree estimation error. Each box includes 100 dots, corresponding to 20 replicates of 5 model conditions. Linear correlations are shown over all 100 points.

average gene tree error, and the number of genes. The patterns are clear: for both summary methods, with true gene trees, or with gene trees that have low error, the species tree also has high accuracy; as the gene tree error increases, the species tree error also increases. For example, with 800 genes, true gene trees analyzed using MP-EST result in recovering the true species tree in all replicates. However, if gene trees have in average 42% error, the species tree has close to 7% error. Interestingly, concatenation analyses seem less sensitive to the variations in sequence length per gene (the only factor we use to vary gene tree error). Figure 4.1(b) shows the correlation between average gene tree error of each replicate and the species tree. There is clear correlation between the gene tree error and the species tree error, and interestingly, the correlation seems higher for large number of genes. Correlations seem also stronger for MP-EST, which is the only statistically consistent summary method studied here. Overall, these observations point to the vulnerability of species tree estimation methods to gene tree estimation error.

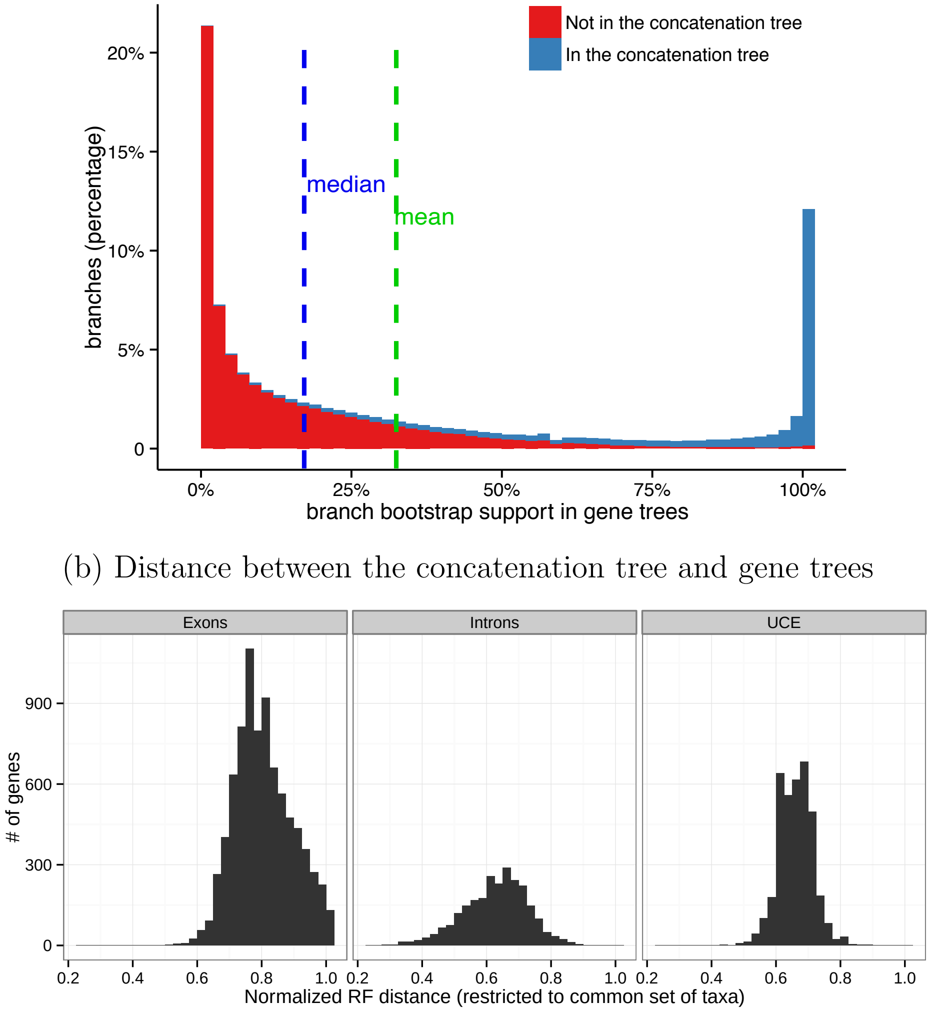
4.1.2 Genet tree estimation on the avian phylogenomics project

The avian phylogenomics consortium obtained whole genome sequences for 48 different bird species. A major goal was to estimate the species tree for the major lineages of birds (roughly corresponding to bird orders). Birds are believed to have gone through a rapid radiation [39, 75, 76] (see also Section 2.2.2.2), a condition that results in high levels of incomplete lineage sorting (we will provide more evidence for this in Section 4.5.1). Because of the high levels of ILS, a major goal of the project was to use ILS-aware methods to estimate a species tree that takes into account coalescence.

In the avian project, we used rigorous pipelines of orthology (defined in Section 2.2) detection, using collinearity of genomic regions (i.e., syntenic blocks [198]) and other criteria (see supplementary document S2 of [39] for details). This procedure identified 14,446 genomic regions (referred to as loci or genes henceforth) that could be used for phylogenetic reconstruction. These loci came in three types of genomic markers: exons (regions of the genome that code for proteins and are relatively slowly evolving) from 8251 genes[2], introns (regions in the genome that do not code for proteins and tend to be fast evolving) from 2516 genes, and 3679 UCEs (ultra-conserved elements that can be dispersed throughout the genome). We estimated gene trees using maximum likelihood (implemented in RAxML) from all 14,446 loci, and observed that no two gene trees had identical topologies [39].

Most loci had low phylogenetic signal, resulting in average bootstrap support (BS) of only 32% for the bifurcating ML trees estimated on these loci, and many branches that had extremely low bootstrap support in the gene trees (Fig. 4.2(a)). We estimated a species tree with a concatenated maximum likelihood analysis on these 14,446 loci. This tree (which we use here as an approximate estimate of the species tree) had a succession of short branches

[2] Here, we use the term gene in the function sense; the use of functional genes in the avian study as proxy for the c-genes (see Section 2.2.2) is based on the hope that a functional gene would include only one c-gene, but see for [34] for an in-depth discussion of this issue.



(a) Branch BS in gene trees of the avian datasets.

(b) Distance between the concatenation tree and gene trees

Figure 4.2: **Discordance and BS in the avian dataset**. (a) Branch BS in gene trees of the avian datasets. Histograms show the distribution of bootstrap support values across *all* branches of all 14,446 gene trees. Blue portions of a bar show the branches at a certain support level that have been recovered in the concatenation tree on the avian dataset; red portions show those that are not in the concatenation tree. Note that highly supported branches tend to be in the concatenation tree while moderately or poorly supported branches tend to be missing. (b) The normalized RF distance between the concatenation tree and all 14,446 gene trees, divided into three maker types.

suggestive of a radiation, and conflicted with all estimated gene trees. The most similar gene trees to this tree still missed more than 20% of its branches, as shown in Figure 4.2(b). However, while most of branches with low BS in the gene trees did not appear in the concatenation tree, branches with high BS were mostly present in that tree (Fig. 4.2(a)). This pattern suggests that while gene trees can have high levels of incongruence, most of the discordance may be due to gene tree estimation error (reflected in lack of support).

Among our data, exons had the least phylogenetic signal (average BS 24%), the introns had the most (average BS 48%), and the UCEs were intermediate in support (average BS 39%). All these markers had levels of BS that can be characterized as low or moderately low. The longest introns in terms of sequence length (defined as those with at least 10,000bp) have the highest average BS (59%), but these represent a very small fraction of the total set of gene trees examined (only 638 of more than 14k markers). Consistent with the observation that most gene tree conflict was due to gene tree error, introns, which had the highest average BS, also had the lowest distance to the concatenation tree, while exons, which had the lowest average BS, had the highest distance (Fig. 4.2(b)).

As we report in Section 4.5.1, our attempt to use summary methods on the collection of 144,446 loci produced trees that were not satisfactory because they had low support, and they failed to recover key clades that much smaller datasets had consistently recovered. Restricting the set of genes to only introns did produce trees that did not have obvious flaws. Several authors have suggested using loci with high support in phylogenomics analyses (e.g., [188]). However, restricting loci is problematic for statistically consistent coalescent-based summary methods, because the conditions under which they are guaranteed to be accurate (with high probability) require a large enough random sample of true gene trees; removing loci can violate this condition and potentially bias the analysis.

To summarize, the two step approach to species tree estimation was not able to analyze the complete avian phylogenomic dataset with high accuracy, and the size of the dataset prevented us from using more extensive co-estimation methods. This shortcoming was a major motivation for developing the statistical binning approach.

4.2 Statistical binning pipeline

The statistical binning pipeline is based on the idea that subsets of genes can be grouped together for the purpose of gene tree estimation. Even in the presence of high levels of ILS, some pairs of genes will have identical or very similar gene trees. If we could find those sets of similar genes, by putting their data together we could increase phylogenetic signal per unit of analysis. Finding combinable sets of data is what we strive for in statistical binning. Once we find combinable subsets of genes, we combine their data, and use these "bins" of data to produce a different set of estimated gene trees that can be used with the summary method. We call the use of binning with a given summary method the binned version of the summary method.

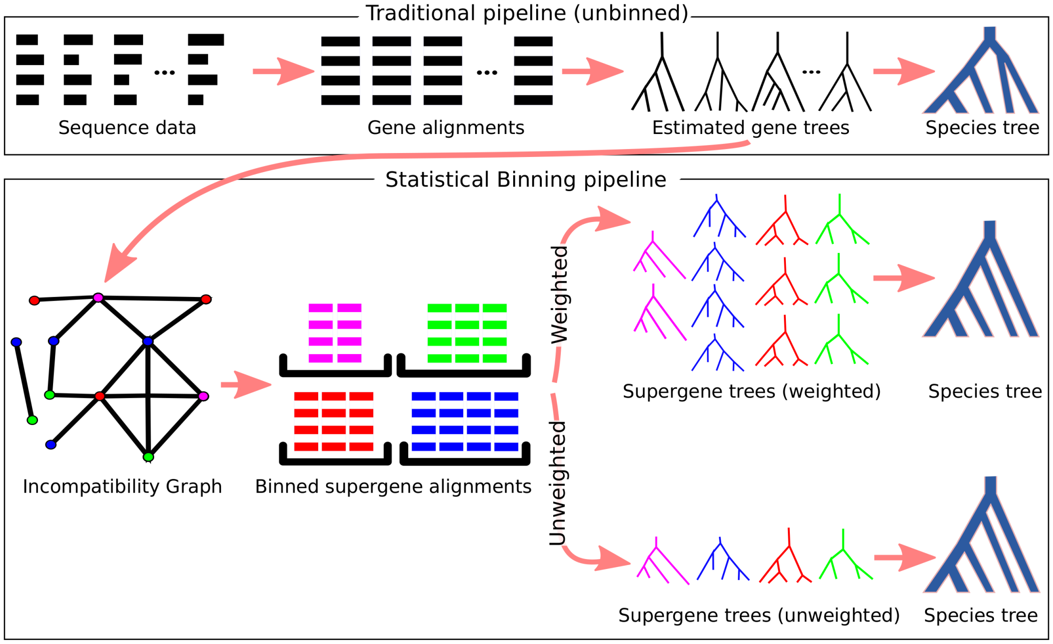

Traditional pipeline (unbinned)
Sequence data
Gene alignments
Estimated gene trees
Species tree
Statistical Binning pipeline
Incompatibility Graph
Binned supergene alignments
Weighted
Unweighted
Supergene trees (weighted)
Species tree
Supergene trees (unweighted)
Species tree

Figure 4.3: **Statistical binning pipeline**. In traditional two-step pipelines, gene trees are estimated from input sequence alignments separately, and then combined into a species tree using a coalescent-based summary method. Statistical binning takes estimated gene trees with branch support (e.g., from bootstrapping) and builds an incompatibility graph. In this graph, each node represents a gene and an edge between two genes represents a *detected* incompatibility between the estimated trees for those two genes at the specified statistical support threshold, or higher. We use an extension of Brélaz heuristic [199] to color the nodes of the graph so that no two adjacent vertices have the same color, and so that the color classes are of similar sizes. This coloring of the vertices defines a division of genes into bins and ensures that no two genes with strongly supported conflict are put in the same bin. We concatenate individual gene alignments of each bin to get a supergene alignment, and estimate a supergene tree from these supergene alignments using ML. A summary method of choice is run on supergene trees to produce an estimated species tree. Two versions of the pipeline are developed. In the unweighted version, each supergene appears once in the input to the species tree. In the weighted version, each supergene tree is repeated once for each gene put in that bin; thus, in the weighted version bins are weighted by their size and the number of trees analyzed by the summary method is the same as the number of genes.

Figure 4.3 shows how the statistical binning pipeline operates, given an input set of loci with their estimated sequence alignments and trees. We use bootstrap support values on branches of the estimated gene trees to divide the set of loci into bins of roughly equal sizes, so that each bin consists of a set of loci where differences in the estimated gene trees can be explained by gene tree estimation error. We concatenate the alignments of loci in each bin into a large alignment (called a supergene alignment) and compute trees on each supergene alignment using Maximum Likelihood (ML); this produces a set of trees (called supergene trees), with one supergene tree for each bin. We then construct a species tree from the set of supergene trees using the desired summary method. Thus, the difference between the unbinned and binned versions of a summary method is the set of trees it uses to compute the species tree: the unbinned summary method uses the original set of gene trees, and the binned summary method uses the set of supergene trees.

The pipeline has two versions: unweighted and weighted. In the unweighted statistical binning, each supergene tree is present only once in the input provided to the summary method. In the *weighted* statistical binning, each supergene tree is repeated as many times in the input to the summary method as the number of genes put in its corresponding bin. Thus, in the weighted approach, the number of supergene trees provided to the summary method is exactly the same as the number of genes. Weighing has no effect when bins are all the same size, but when bin sizes are different, it helps preserving the gene tree distribution, and enables some theoretical guarantees for statistical binning, as we show later in this section.

4.2.1 Details of statistical binning

The statistical binning technique is parameterized with a bootstrap support threshold, $\mathcal{S} < 1$. The input is a set of multiple sequence alignments, one for each of k given genes. Binning uses a simple statistical heuristic to determine which pairs of genes can be put into the same bin; this test is based on the BS of gene tree branches, and will prevent two genes from being in the same bin if their ML gene trees have conflicting branches, each with BS of at least $\mathcal{S}$. As we will discuss, supergene trees can be estimated using partitioned concatenated analyses, which would allow the branch lengths and other model parameters to be re-estimated for each gene within a bin. For this reason, we only need to consider topological incongruence and not branch length, thus, allowing genes whose estimated trees share the same topology but differ with respect to other model parameters to be placed in the same bin. Statistical binning pipeline includes the following steps.

Step 1 - initial gene trees: We use ML with bootstrapping to estimate gene trees with branch support values.

Step 2 - pairwise conflict: We compare all pairs of gene trees and note whether they conflict at the support threshold $\mathcal{S}$. This step requires $\binom{k}{2}$ comparisons, and therefore, it is important that each comparison is fast. We say that a given pair of trees exhibit conflict at threshold $\mathcal{S}$ if there is a pair of incompatible branches, one in each of the two gene trees and both with BS of at least $\mathcal{S}$. Two branches are incompatible when no tree can be constructed that has both of these branches [159]. More specifically, if two branches are incompatible, no tree exists with branches that induce the bipartitions defined by these two branches.

To test two trees for incompatibility at threshold $\mathcal{S}$ or higher, we first collapse all branches in both trees with support below $\mathcal{S}$ and also restrict them to their common set of leaves. We then ask whether a tree exists that is a common refinement of these two collapsed trees. This can be done by comparing each bipartition in the first tree against each bipartition in the second tree and asking whether the two bipartitions are compatible. Two bipartitions are compatible when (after restricting to their shared set of leaves) one part of one bipartition is a subset of one part of the other bipartition. Testing for compatibility of two trees can be performed in linear time [159]; hence, this calculation is fast.

Step 3 - bin formation: This step uses a graph-based optimization to divide the set of genes into bins. We build a graph in which each gene is represented by a node and an edge is present between two nodes (i.e., genes) if the estimated trees on that pair of genes exhibit conflict at threshold $\mathcal{S}$, as calculated in the previous step. By definition, the graph depends on the parameter $\mathcal{S}$: larger values for $\mathcal{S}$ will generally consider more genes to be combinable than smaller values.

The graph created in this step is called an incompatibility graph. To create bins from this graph, we color the vertices of the graph so that no two vertices with the same color are adjacent, and put all vertices with the same color into a common bin. This is the classic vertex coloring problem in graph theory [200] and each bin constructed using such a vertex coloring constitutes an independent set: a set where no two nodes are connected and therefore no pairwise incompatibility between genes has support of $\mathcal{S}$ or greater. Any vertex coloring would maintain our desired guarantee, but a natural optimization problem is to minimize the number of bins, so that bins have as many genes as possible given the constraints (more genes in a bin result in increased data and therefore increased signal). Moreover, among solutions that minimize (or come close to minimizing) the number of bins, we would like to choose a vertex coloring in which the different color classes have approximately the same size (i.e., are balanced). We seek this because we want to avoid some bins that are very large (and so are close to concatenation) and others that are very small (and do not benefit from binning).

Finding a minimum vertex coloring (regardless of whether bins are balanced) is NP-hard [199–201] but algorithms for solving the problem heuristically have been developed. One of the main heuristics for minimum vertex coloring is the Brélaz heuristic [199]. The Brélaz heuristic first finds a large clique in the graph and assigns a different color to each node in the clique. Then, in a greedy stage, nodes are processed in turn (according to an order described below), and each node is given the “first” color that can be legally assigned to that node (i.e., there is no edge from any of the nodes with that color to this new node). If no such color exists, a new color is created and the node is assigned this new color. The order of processing nodes is dynamically changing: the next selected node is always the one that conflicts with the largest number of existing colors (breaking ties arbitrarily).

When each node is being processed, the order of checking colors is fixed in the Brélaz heuristic (arbitrarily for the clique and then each color is added to the end of the order). This means that if a node can be assigned one of multiple colors, the first color is going to be chosen, and so, the arbitrary order of colors determines the size of bins. Our simple modification to the Brélaz heuristic is that in the greedy stage, each node is assigned to the smallest bin that is compatible with it (i.e., instead of ordering colors arbitrarily, we dynamically order them based on their size). When two or more bins have the same smallest size, the algorithm breaks the ties arbitrarily. This simple modification ensures that the bin sizes become as balanced as possible given the constraints of the graph and the limitations of the Brélaz greedy mechanism. Figure 4.4 shows an example of running unbalanced and balanced versions of the Brélaz heuristic.

Step 4 - supergene tree estimation: Once bins are formed, alignments of genes in the same bin are concatenated into a supergene alignment, and supergene trees are estimated on these alignments using ML. Concatenated

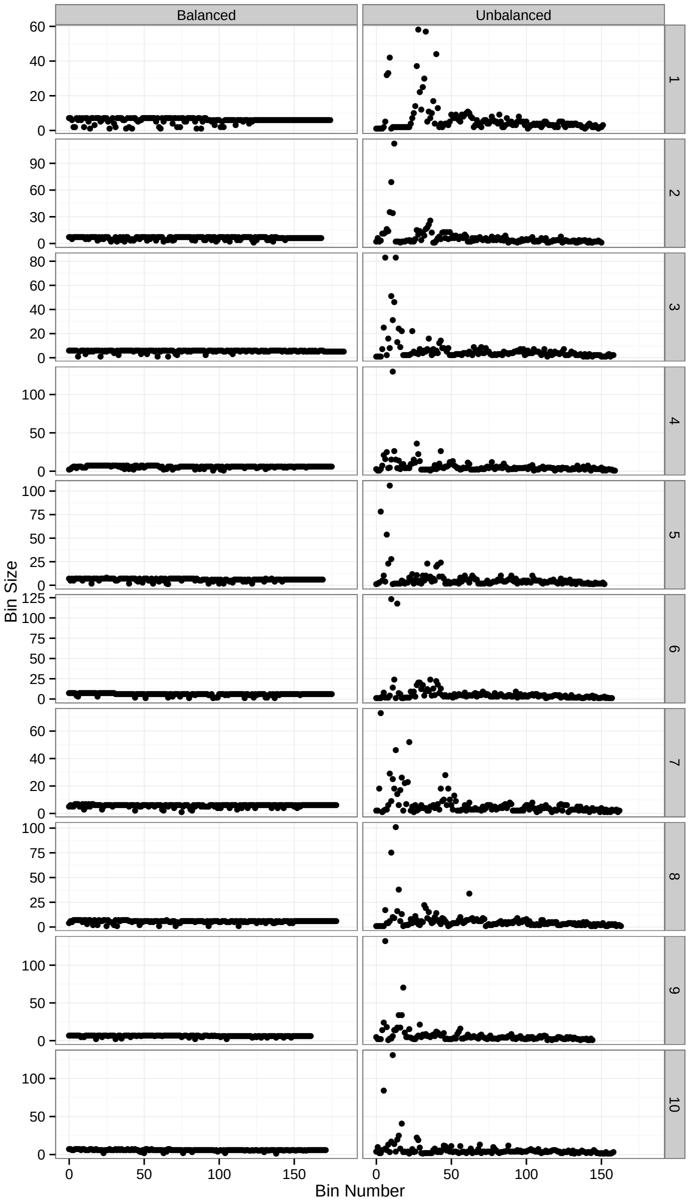


Figure 4.4: **Bin sizes using Brélaz heuristic and our balanced vertex coloring.**. We show bin sizes on the simulated dataset produced by the original Brélaz heuristic ("unbalanced") and our modification ("balanced"). Results are shown for the first 10 replicates of the avian simulated UCE-like dataset with 1000 genes and $\mathcal{S} = 50\%$. Each dot represents a bin, with vertical axis showing the bins size, and horizontal axis showing the bin index.

analyses of alignments from different loci can be performed in different ways. In an unpartitioned analysis, all the sites in the concatenated alignment are assumed to evolve down a single model tree with a fixed topology and fixed numeric parameters. In contrast, *fully partitioned* analyses of concatenated alignments assume that the different loci all evolve down the same tree topology, but allow the different parts within the concatenated alignment to have different values for all of the numeric parameters of the model, including the branch lengths. Fully partitioned and unpartitioned ML analyses can result in different trees, and these analyses have different theoretical properties as we discuss below. To accommodate differences between branch lengths in genes put in the same bin (and also other model parameters), we strongly recommend using a *fully partitioned* analysis where each gene is assigned a separate partition, and all model parameters are allowed to differ between partitions. We will show in Section 4.2.2 that weighted statistical binning has theoretical guarantees of statistical consistency under certain conditions, and unweighted binning is not consistent under those conditions.

Step 5 - species tree estimation: The supergene trees are used as input to the summary method of choice. In the unweighted pipeline, each supergene tree appears only once in the input to the species tree. If all the bins are fully balanced, this procedure is fine. However, if there are some remaining imbalances between the bin sizes, this can distort the distribution of the gene trees, as genes put inside larger bins contribute less to the overall gene tree distribution. Solving this issue is simple. For each bin, we weight its supergene tree by it size. This ensures that the distribution of gene trees is not distorted by patterns of bin size distribution. This *weighted* statistical binning pipeline has better theoretical guarantees as we show below.

4.2.2 Theoretical properties of statistical binning

Here we describe theoretical statistical properties of statistical binning. We use the following notation throughout:

$\mathcal{S}$: The input BS threshold

$\{g_1, \ldots, g_k\}$: the set of k input genes

s_i: sequence alignment for gene g_i

t_i: true tree for gene i

$\hat{t}_i$: tree estimated based on s_i using ML under the GTR model

T: the true species tree

$\hat{T}$: an estimated species tree

L: length of gene sequences. We clarify in context whether length of a single gene, the maximum length, or the minimum length is intended.

We will assume that the input sequences are generated under a two-step process:

GTR+MSC: gene trees $t_1, \ldots, t_k$ evolve within a species tree T under the MSC model (see Section 2.2.2.2), and then sequence alignments $s_1, \ldots, s_k$ evolve down each gene tree under the General Time Reversible (GTR) model (see Section 2.1.2.1). Each gene tree has its own GTR model parameters, and so the tree topologies, substitution matrices, base frequencies, and gene tree branch lengths can differ across genes.

Throughout this chapter, we discuss statistical consistency under conditions where the number of genes and the number of sites per gene are both allowed to go to infinity. Thus,

Statistical Consistency: We consider the statistical consistency under the situation where both L, the minimum sequence length of any gene, and k, the number of genes, is allowed to increase to infinity. Let ψ be a method of reconstructing a species tree $\hat{T}$ under the GTR+MSC model. We call ψ statistically consistent iff we can prove that as both $L \to \infty$ and $k \to \infty$, the estimated species tree $\hat{T}$ converges in probability to T.

The main results are given in Theorem 4.2.5 and Theorem 4.2.6, where we prove that using weighted statistical binning in a phylogenomic pipeline is statistically consistent under the GTR+MSC model, but that replacing weighted statistical binning with unweighted statistical binning is *not* statistically consistent under GTR+MSC. Both of our results are according to the definition of consistency where we allow both the number of genes and the sequence length of each gene to go to infinity. These conditions obviously make for weak statistical guarantees. Variations of the statistical consistency concept can be imagined where, for example, only k goes into infinity and L remains constant, and these make for stronger theoretical guarantees [202]. We have not been able to prove consistency or inconsistency of statistical binning under the stronger definition, but we note that traditional two-step pipelines have also not been proved consistent or inconsistent under those conditions (see [147] for an in depth discussion of this issue).

Recall that the statistical binning algorithm uses a heuristic to color the vertices. For our heuristic, we can prove:

Lemma 4.2.1. *Let $M = \{\hat{t}_1, \ldots, \hat{t}_k\}$ be the multi-set of estimated gene trees, and assume that all the branches in each $\hat{t}_i$ have BS above $\mathcal{S}$.Then, when statistical binning is run,*

1. *there will be one bin for each of the different estimated gene tree topologies in M, and*
2. *for every bin, every two genes in the bin will have the same estimated gene tree topology.*

Proof. Recall that the algorithm operates in two stages. In the first stage, our binning heuristic finds a clique in the graph and places the genes within that clique into different bins. After this stage, each bin will be a singleton, and therefore, the two conditions of the Lemma will hold for those genes that are binned so far: all the genes in the clique are pairwise incompatible and therefore they are all distinct, satisfying the first condition, and the second condition is irrelevant.

After processing the clique, which we assume has size c, the greedy phase starts. We prove by induction that as the remaining genes are processed in the greedy phase, the two conditions continue to hold. Thus, the inductive hypothesis is that after $i \geq c$ genes are processed, the two conditions of the lemma hold for the genes processed to that point. For $i = c$, we only have the clique and thus this inductive hypothesis is true.

Now suppose the inductive hypothesis holds for $i - 1 \geq c$ genes, and consider what happens when the i^{th} gene tree, $\hat{t}_i$, is processed in the greedy stage. When we process g_i, there are two cases, depending on whether its estimated gene tree $\hat{t}_i$ is a gene tree topology that has been seen before. If $\hat{t}_i = \hat{t}_j$ for some $1 \leq j \leq i - 1$, then there is a bin that contains all the genes with that topology (by the inductive hypothesis), and g_i can be added to that bin. Note that by the inductive hypothesis, all other bins contain genes with different estimated gene tree topologies than $\hat{t}_i$. Furthermore, by assumption, all edges of all gene trees have BS above $\mathcal{S}$. Hence, we cannot add g_i to any other bin. Therefore, if $\hat{t}_i$ has been seen before, there is only one bin we can add g_i to, and it is the bin for genes with the same tree topology as $\hat{t}_i$. The other case is where $\hat{t}_i$ has not been seen before. In this case, $\hat{t}_i$ is different from every previously seen estimated gene tree, and again since all BS values are above $\mathcal{S}$, it cannot be put in any other bin. Therefore, a new bin is created and g_i is placed in this new bin. As a result, the new set of bins satisfies the inductive hypothesis, so that there is one bin for every estimated gene tree topology, and no two genes in any bin have different estimated gene tree topologies. □

We can now prove the following theorem:

Theorem 4.2.2. *Let sequence alignments $s_1, \dots, s_k$ evolve under GTR+MSC on a species tree T. Let $t_1, t_2, \dots, t_k$ be the true gene trees, and let $\theta_1, \theta_2, \dots, \theta_k$ be the set of numeric GTR model parameters (gene tree branch lengths, base frequencies, and 4×4 substitution matrices) so that $m_i = (t_i, \theta_i)$ is a GTR model tree for each $i = 1, 2, \dots, k$. Let $\mathcal{M} = \{m_1, \dots, m_k\}$. Let $\epsilon < 1$ and BS threshold $\mathcal{S} < 1$ be given. Then, there is a sequence length L (that depends on $\mathcal{M}$, $\mathcal{S}$ and ϵ) such that if all gene sequences have at least L sites, then with probability at least $1 - \epsilon$, the following will be true:*

- *Each estimated gene tree $\hat{t}_i$ estimated using ML under GTR will have the same unrooted topology as t_i (the true gene tree for g_i), and will have BS greater than $\mathcal{S}$ for all its branches,*
- *When $\hat{t}_1, \dots, \hat{t}_k$ are used as input to the statistical binning, two genes g_i and g_j are put in the same bin, only if t_i and t_j have the same topology,*
- *All genes with the same true gene tree topology will be in the same bin.*

Proof. Since ML under GTR is statistically consistent for sequences generated by GTR model trees, then for any $\epsilon' > 0$, there is a sequence length L'_i such that given sequence alignment s_i with at least L'_i sites generated on t_i, we have $t_i = \hat{t}_i$ with probability at least $1-\epsilon'$ (thus, the ML tree topology under GTR is the true gene tree topology with high probability). The statistical consistency of ML also implies that there is a L_i such that for sequence alignment s_i with at least L_i sites, bootstrap support of all branches of $\hat{t}_i$ are greater than $\mathcal{S}$, with probability at least $1 - \epsilon'$ [203]. Letting $L = \max_i\{L_i, L'_i\}$, it follows that all estimated gene trees will be the true gene trees and have BS greater than $\mathcal{S}$ with probability at least $1 - k\epsilon'$. Therefore, when $\epsilon' = \frac{\epsilon}{k}$ and the sequences are all of length at least L, then the first condition of the theorem follows. Since under these conditions, estimated and true gene trees are identical with probability at least $1-\epsilon$, by Lemma 4.2.1, the other two conditions follow. □

Before providing our other proofs, we give a formal definition of a *fully partitioned* analysis.

Fully partitioned ML under GTR. In a fully partitioned ML under GTR analysis, the input is a set of k multiple sequence alignments, $\{s_1, s_2, \ldots, s_k\}$. These alignments are concatenated into a supermatrix, S, in which the locations where the different alignments begin and end are also noted. The ML score of a candidate tree t (note that t specifies only a topology and not also branch lengths) for input S is

$$score(t) = sup_{\Theta}\{\prod_{i}^{k} Pr(s_i|(t,\theta_i)) : \Theta = \{\theta_1, \theta_2, \ldots, \theta_k\}\} \tag{4.1}$$

Thus, Θ denotes a set of GTR model parameters (branch lengths and other GTR parameters) for each of the parts within the concatenated alignment S. We will refer to the tree topology that achieves the optimal score under this fully partitioned analysis as the solution to the fully partitioned ML analysis of the concatenated matrix, understanding that the numeric GTR parameters (branch lengths and substitution matrices) are estimated independently for each part of the alignment, and hence can differ between parts.

With that definition, we can now provide the following lemma:

Lemma 4.2.3. *Let $\{s_1, \ldots, s_k\}$ be a set of sequence alignments all on the same set of species. Suppose that tree topology t is an optimal solution for ML under GTR for each s_i (allowing various GTR parameters for different $i = 1, 2, \ldots, k$). Then t will be an optimal solution to a fully partitioned ML under GTR analysis on a concatenation of $s_1, s_2, \ldots, s_k$.*

Proof. In a fully partitioned ML under GTR analysis, the ML score of a given candidate tree t with respect to a matrix M under a fully partitioned ML analysis is given by Equation (4.1). Suppose that the tree topology t is an optimal solution to ML under GTR for each s_i but not an optimal solution to the fully concatenated ML under GTR analysis. Then, for some tree $t' \neq$

t, $score(t') > score(t)$. Therefore, for at least one i, $sup_\theta\{Pr(s_i|(t',\theta))\} > sup_\theta\{Pr(s_i|(t,\theta))\}$. But then t is not an ML tree topology for s_i, contradicting our assumption. Therefore, if the maximum likelihood analysis is performed in a fully partitioned manner, then tree topology t will be an optimal solution to the ML under GTR analysis. $\square$

We now consider the result of applying weighted statistical binning within a phylogenomic pipeline.

Corollary 4.2.4. *Let $\mathcal{G} = \{g_1, g_2, \ldots, g_k\}$ be a set of k genes, and $m_i = (t_i, \theta_i)$ be the true gene tree and GTR parameters (including branch length) for g_i, $i = 1, 2, \ldots, k$. Let $\mathcal{S} < 1$ be the user provided BS value. Assume that the gene sequence alignment s_i evolves down the GTR model tree $m_i = (t_i, \theta_i)$, for $i = 1, 2, \ldots, k$. As the sequence lengths for all the genes increase then with probability converging to 1, for each bin produced during a statistical binning analysis, all genes in any bin will have the same true gene tree topology, and the supergene tree topology produced for each bin using fully partitioned ML under GTR will converge in probability to the common true gene tree topology for the genes in the bin.*

Proof. By Theorem 4.2.2, as the sequence length increases, then with probability converging to 1, the genes in each bin will share a common true gene tree topology, their estimated gene trees will be topologically identical to each other and to the true gene tree, and will each have BS greater than $\mathcal{S}$. By

Lemma 4.2.3, under these conditions, a fully partitioned GTR maximum likelihood analysis of the concatenated alignment of the genes in a bin will produce the true gene tree topology for the genes in the bin. □

We now address the statistical consistency of phylogenomic pipelines that use weighted and unweighted statistical binning.

Theorem 4.2.5. *The phylogenomic pipeline that uses ML under GTR to estimate gene trees, uses weighted statistical binning to compute supergene trees, and then combines the supergene trees using a coalescent-based summary method, is statistically consistent under the GTR+MSC model.*

Proof. By Corollary 4.2.4, as the sequence length for each gene goes to infinity (minimum $L \to \infty$), when gene trees are estimated using ML, the estimated gene trees converge to the true gene trees and have BS that converges to 1.0, and so all genes put in any bin by statistical binning will have the same true gene tree with probability converging to 1, and the supergene trees produced for each bin will converge in probability to this common true gene tree. In weighted statistical binning, this common true gene tree topology is replicated as many times as the number of genes in the bin, and hence the distribution produced using weighted statistical binning is identical to the distribution of the true gene trees. Therefore, as both k and L increase to infinity, the gene tree distribution produced by weighted statistical binning converges in probability to the true gene tree distribution. The statistical consistency of the pipeline follows from the use of a coalescent-based summary method, since as $k \to \infty$, the species tree produced by the summary method given true gene trees converges in probability to the true species tree. □

We now consider the case where we use unweighted statistical binning instead of weighted statistical binning.

Theorem 4.2.6. *The phylogenomic pipeline that uses ML under GTR to estimate gene trees, uses unweighted statistical binning to compute supergene trees, and then combines the supergene trees using a coalescent-based summary method, is statistically inconsistent under the GTR+MSC model.*

Proof. The proof for Theorem 4.2.5 shows that as the sequence length L increases, the set of bins produced by statistical binning converges in probability to having one bin for each of the true gene trees, and the supergene tree for each bin converges to the common true gene tree for the bin. As $k \to \infty$, the set of all observed supergene trees converges in probability to the set of all possible gene trees (since all gene trees have strictly positive probability under the multi-species coalescent model). Hence, the multi-set of supergene trees produced by unweighted statistical binning will converge to the set that has each possible gene tree appearing exactly once. This is a flat distribution, and it is not possible to reconstruct the species tree from a flat distribution. Hence, the use of unweighted statistical binning in a phylogenomic pipeline is not statistically consistent. □

4.3 Experimental setup

We used biological and simulated datasets to evaluate species trees estimated using weighted and unweighted binning pipeline, and compared their results against the traditional (unbinned) summary method pipeline, as well as concatenation using ML under the GTR+Γ model, computed by RAxML [120].

4.3.1 Datasets

We used four biological and three sets of simulated datasets for evaluating binning.

4.3.1.1 Biological datasets

We studied the avian dataset [39] with 14k genes and 48 species, a mammalian dataset with 447 genes and 37 species [77], a yeast dataset with 23 species and 1070 genes, a vertebrate dataset with 15 species and 1087 genes, and a metazoan dataset with 21 species and 225 genes [188]. Species trees estimated using concatenation and gene trees were available [39, 188], except for the maximum likelihood gene trees from the mammalian dataset, which we recomputed using RAxML.

4.3.1.2 Simulated datasets

Two of the simulated datasets were simulated based on two biological datasets with the goal of emulating their properties. The third dataset was used to set up an artificial model condition, with few species and simulated under extreme conditions (which we describe below). The two biologically-inspired datasets were generated based the avian phylogenomics dataset with 48 species and 14k loci [39] and the mammalian dataset with 37 species and 447 loci [77].

In all three datasets, we simulate data according to GTR+MSC; thus, we use a species tree with branch lengths in coalescent units, from which we simulate gene trees according to MSC, and then use simulated (true) gene trees to simulate sequence data according to the GTR model. The details of this process are similar between our two biologically-inspired (avian and mammalian) datasets, but slightly different for the 10-taxon dataset. We first describe our simulation procedure for the two biological datasets, and then describe how 10-taxon data is generated.

On two biologically-inspired datasets, we choose the default parameters of our simulation procedure such that we produce levels of gene tree estimation error and ILS that resemble biological datasets. We then varied the model parameters to produce lower and higher ILS levels and simulated gene trees with varying levels of phylogenetic signal (and thus, gene tree estimation error).

Step 1 - choosing the model species tree: On the two biologically-inspired datasets, the default model species trees are themselves estimated from the two biological datasets using MP-EST, a statistical summary method that estimates both the species tree topology and branch lengths in coalescent units. By using the real data to estimate the model species tree, we try to explore parts of the parameter space that are close to parameters on biological data. MP-EST produces a tree with branch lengths in coalescent units, and therefore has all the information necessary to simulate gene trees under the MSC model. In the case of the mammalian dataset, the gene trees we estimated on the Song *et al.* data [77] had high average BS (mean 71%); therefore, we chose to use all the gene trees as input to MP-EST and use the resulting tree as our model species tree. For simulating the avian dataset, we used the MP-EST* tree from [39] as the reference topology, but we re-estimated the branch lengths on that model tree using only the longest genes with at least 10,000 sites. This was necessary because most gene trees had very low support values (and thus high estimation error); branch lengths estimated in coalescent units are directly impacted by observed gene tree discordance, and including gene trees with high estimation error (i.e., low support gene trees) inflates the amount of ILS. Resulting trees are shown in Figure 4.5(A,B).

To produce other model conditions with different amounts of ILS, we modified the branch lengths on the model species trees uniformly by dividing or multiplying them all by two. Thus, the 2X condition has doubled branch lengths and so reduces ILS, and the 0.5X condition has halved branch lengths and so increases ILS.

Step 2 - simulating gene trees: For each of the three avian and three mammalian model species trees (with coalescent unit branch lengths), we simulated true gene trees according to MSC using Dendropy [204]. For the avian

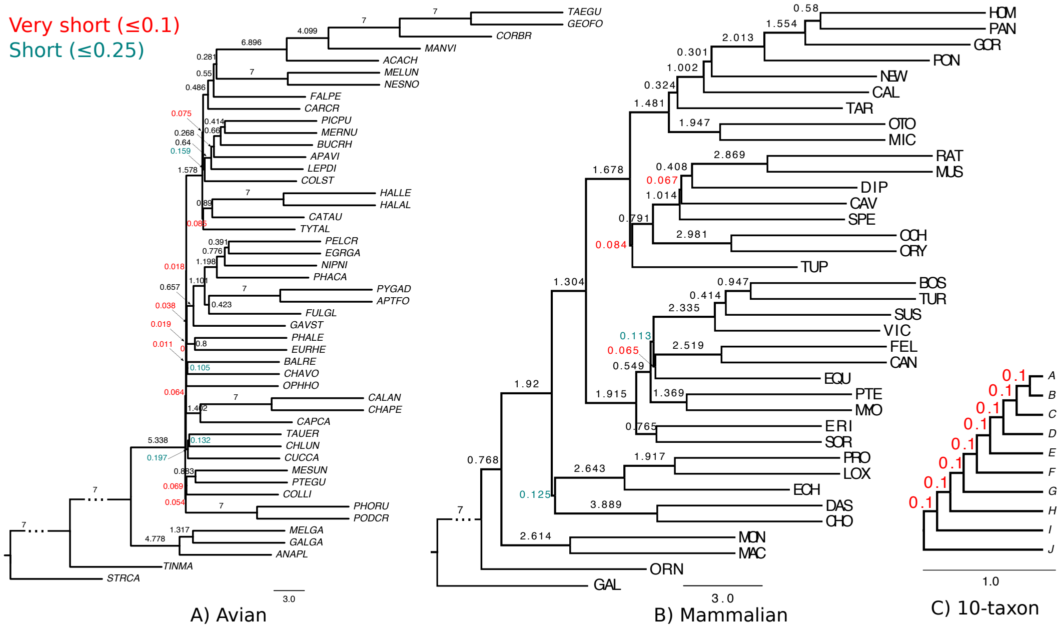


Figure 4.5: **Model (reference) species trees for simulated datasets**. We show model species trees used in our simulation studies with branch lengths in coalescent units. (A) and (B) Model species trees estimated using MP-EST from biological data; see text for details. (C) Caterpillar like model tree for the 10-taxon tree. The lengths of all internal branches and the two branches incident with leaves A and B are all set to 0.005 substitutions per site and the assumption of ultrametricity defines the remaining branch lengths; $\theta = 0.05$, and thus, all internal branch lengths are 0.1 in coalescent units.

dataset, we simulated 20,000 gene trees and subsampled these to create 20 replicates of model conditions that had 200, 500, or 1000 genes, and 10 replicates with 2000 genes per replicate. For the mammalian dataset, we created 20 replicates of model conditions that had 200, 400, or 800 genes per replicate. The amount of true gene tree discordance in our simulated datasets ranges from relatively low (mammalian 2X condition) to very high levels (avian and

Table 4.1: **Statistics on levels of ILS in simulated datasets**. We show true gene tree incongruence for simulated datasets, with varying model conditions. 2X corresponds to the case where ILS is reduced by multiplying branch lengths by two and 0.5X corresponds to the case where ILS is increased by dividing branch lengths by two. The first three rows show average, minimum, and max normalized Robinson-Foulds (RF) distances between true gene trees and the model species tree. The next three rows show average, minimum, and maximum distances between all pairs of true gene trees. Maximum, minimum, and mean values shown are averages across all 20 replicates of 1000 genes for the avian dataset and 20 replicates of 200 genes for the mammals dataset.

	Mammals			**Avian**		
	2X	1X	0.5X	2X	1X	0.5X
Distance to Species Tree (mean)	18%	32%	54%	35%	47%	59%
Distance to Species Tree (min)	0%	3%	26%	16%	24%	36%
Distance to Species Tree (max)	42%	62%	82%	58%	69%	78%
Distance to other Gene Trees (mean)	26%	46%	71%	44%	57%	68%
Distance to other Gene Trees (min)	3%	14%	36%	16%	24%	38%
Distance to other Gene Trees (max)	51%	77%	96%	69%	77%	89%

mammalian 0.5X condition). See Table 4.1 for summary statistics on distance between true gene trees and the species tree, and between true gene trees.

Step 3 - branch length conversion: Branch lengths on the simulated gene trees are expressed in coalescent units, and have to be converted into expected numbers of substitutions for simulating sequence alignments. To do this conversion, we used branch lengths observed from trees reconstructed from real data. For the avian data set, we used the gene trees reconstructed from the 190 longest introns. For the mammalian dataset, we used all 447 gene trees from [77]. For branches leading to leaves, we sample the distribution of branch lengths for the same leaf in the biological data; thus, for each species, we select a random value from lengths of branches leading to the same leaf in the real data. For internal branches, we used a different approach. We ordered the internal branch lengths on the simulated true gene trees and on the reconstructed biological trees separately, and matched branch lengths from the reconstructed trees with branch lengths from the simulated trees by their rank percentile. As a result, we converted branch lengths on simulated gene trees such that their branch in each specific rank percentile had the same length as the reconstructed gene trees of the real dataset. This way both internal and external branches of the simulated gene trees had realistic branch lengths, as observed in the real data. Also, this produced model gene trees that are not ultrametric (i.e., do not exhibit the strong molecular clock).

Step 4 - simulating sequence data: For each of the resulting simulated gene trees, we simulated alignments under a GTR+G4 model using bppseq-gen [205] based on parameters estimated by bppml [205] on the subset of avian genes that had all the taxa (1185 genes). The same GTR parameters were used for the mammalian dataset and are shown in Appendix A.2.1. To vary the amount of phylogenetic signal in the genes, we controlled the sequence length.

We use average BS to quantify the amount of phylogenetic signal in the gene alignments. The avian biological dataset had estimated gene trees with very low average BS. As noted before, the avian dataset had three types of genomic markers: 8251 exons, 2516 introns, and 3679 UCEs. We simulated model conditions that resembled the avian exons-only, UCEs-only, introns-only, and long introns-only datasets, with respect to their average BS values (Figure 4.6). To achieve these average BS levels, we simulated sequence alignments with varying length: 250bp, 500bp, 1000bp, and 1500bp, respectively (to get shorter alignments, we simply trimmed our longest alignments of 1500bp by retaining the first 250bp, 500bp, or 1000bp sites and discarding the rest). These resulted in average BS of 27%, 37%, 51%, and 60% – values that are very close to those of the four partitions of the avian datasets (24%, 39%, 48%, and 59%). Figure 4.6 shows the distribution of the average BS for real and simulated avian datasets and demonstrate the similarity between phylogenetic signal in our biological and simulated datasets. In our discussion of our results, we refer to these different four different model conditions as exon-like (250bp), UCE-like (500bp), intron-like (1000bp), or long intron-like (1500bp), or when appropriate simply by referring to their sequence length.

For the mammals dataset, we used two values for sequence lengths: 500bp and 1000bp; these resulted in average BS values of 63% and 79%, effectively bracketing the average BS in the real dataset (71%). We refer to these conditions as 63%-BS (500bp) and 79%-BS (1000bp).

Model conditions: To summarize, the three parameters that we vary are 1) the amount of ILS, controlled by species tree branch length, 2) phylogenetic signal per gene, controlled by sequence length, and measured by average gene tree BS, and 3) the number of genes. Studying all combinations of all param-

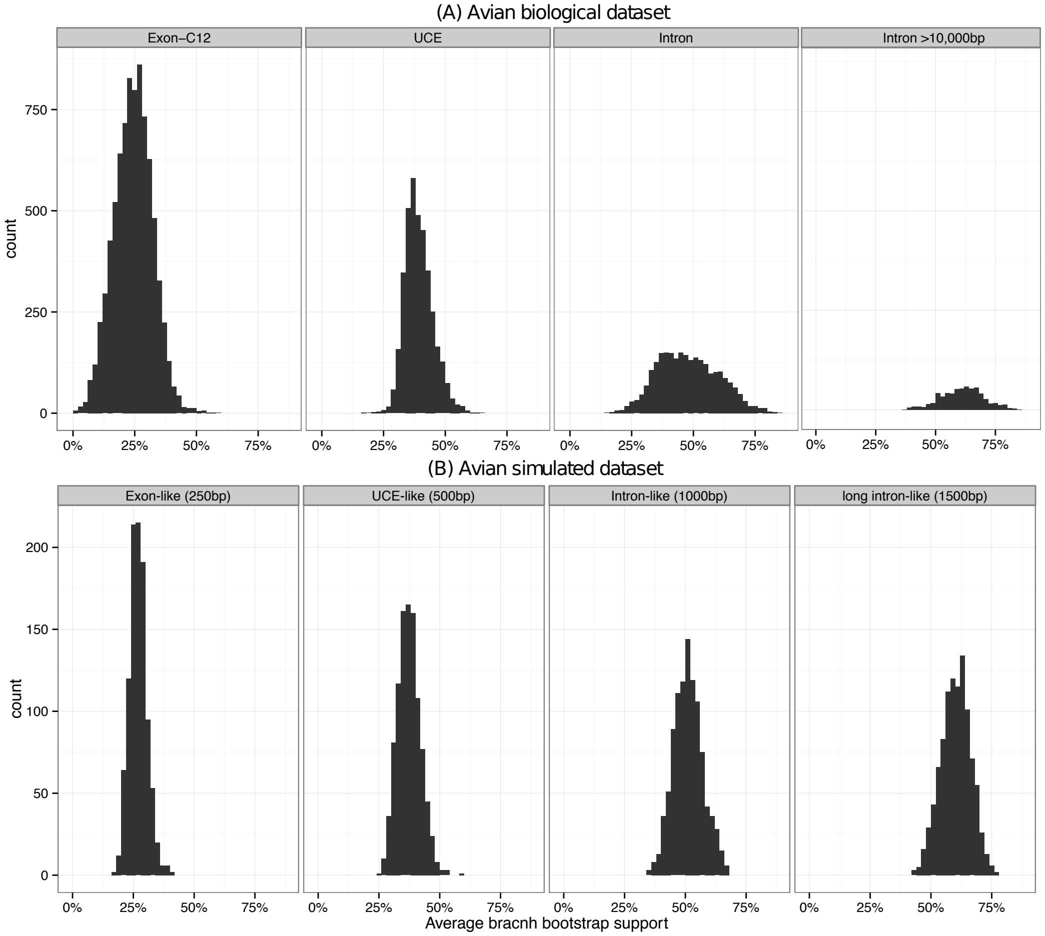


Figure 4.6: **Gene tree BS for avian biological and simulated datasets**. Histograms show the distribution of average bootstrap branch support across (A) four partitions of the avian dataset with a total of 14,446 loci [39], and (B) 1000 genes from each of the four simulated model conditions for the avian dataset with various target “support” levels. Note the extremely low support of most loci in the avian biological dataset. The simulation procedure adjusts alignment length (while fixing all other parameters such as rate of evolution, which also impact phylogenetic signal) so that the BS values obtained on estimated simulated gene trees resemble those of the real dataset. The Long intron-like model condition has BS values similar to a subset of introns that were all at least 10,000bp long.

eters would be infeasible. Instead, we start from default settings and change variables one at a time. Thus, we have the following data "collections":

Collection Avian-1 (1X, 1000 genes): we fix ILS to 1X and the number of genes to 1000, and vary the sequence length: Exon-like (250bp), UCE-like (500bp), Intron-like (1000bp), and Long intron-like (1500bp).

Collection Avian-2 (1X, UCE-like): we fix ILS to 1X and sequence length to 500bp (UCE-like), and vary the number of genes: 200, 500, 1000, and 2000.

Collection Avian-3 (1X, Intron-like): we fix ILS to 1X and sequence length to 1000bp (intron-like), and vary the number of genes: 200, 500, 1000, and 2000.

Collection Avian-4 (1000 genes, UCE-like): we fix number of genes to 1000, and gene length to 500bp (UCE-like), and vary the ILS level: 0.5X, 1X, and 2X.

Collection Mammalian-1 (1X, 63%-BS): we fix the ILS level to 1X and sequence length to 500bp (63%-BS) and vary the number of genes: 200, 400, and 800.

Collection Mammalian-2 (1X, 79%-BS): we fix the ILS level to 1X and sequence length to 1000bp (79%-BS) and vary the number of genes: 200, 400, and 800.

Collection Mammalian-3 (200 genes, 63%-BS): we fix the number of genes to 200 and the sequence length to 500bp (63%-BS), and vary the amount of ILS: 0.5X, 1X, and 2X.

Thus, the avian dataset has 12 model conditions, divided into four collections, with two conditions appearing in multiple collections. The mammalian dataset has 8 model conditions, divided into three collections, with one model condition appearing in two collections.

In addition to these collections, we built one replicate of a model condition with 14,350 genes for the avian simulation in order to closely approximate the actual avian dataset in terms of the number of loci and average BS for estimated gene trees; thus 8250 genes are exon-like in terms of average BS, 2500 are intron-like, and 3600 are UCE-like. Similarly, we built a mixed model condition for mammals, where 200 genes of 63% support level and 200 genes of 79% support level were combined to get 400 genes of 71% average support, resembling the real dataset. We refer to these two as mixed avian and mixed mammalian model conditions.

Overall, the avian simulated datasets have higher levels of ILS and lower BS values than the mammalian datasets, and so present a more challenging condition.

10-taxon simulated dataset: We simulated an artificial 10-taxon dataset to study the behavior of binning under conditions where the level of ILS was extremely high, and when few taxa are present. The conditions simulated here are not inspired by real biological datasets, but rather are meant to create a very challenging condition.

We used a 10-taxon model species tree with a caterpillar-like (also known as a pectinate, or ladder-like) topology, which has eight short internal branches in succession (see Fig 4.5(C)). The length of all internal branches is set to 0.005 substitutions per site and the population size parameter ($\theta = 4N\mu$) is set to 0.05 for all branches, and this results in seven very short internal branches (0.1 in coalescence units) in succession, a condition that gives rise to high levels of ILS [74, 135]. The average distance between true gene trees and the species tree is 79%. Ultrametric gene trees were simulated down this tree using McCoal [206] and using control files given in Appendix A.2.1. Unlike the biologically-inspired model conditions, no transformations of branch lengths were used, and therefore, gene trees follow a strict molecular clock. Sequence data were simulated down each gene tree using bppseqgen [205] and with the same parameters of sequence evolution as those used for the biologically-inspired datasets (see Step 4). We built 10 replicates for four model conditions by trimming gene data to 100 or 1000 sites, and by using 100 or 1000 genes.

4.3.2 Methods

Three summary methods – the greedy consensus, Matrix Representation with Parsimony (MRP) [207], and Maximum Pseudo-likelihood Estimation of Species Trees (MP-EST) [134] – were applied to simulated avian and mammalian datasets using the site-only multi-locus bootstrapping (MLBS) [208] procedure (see below for a description of MLBS). For the 10-taxon dataset, we did not run concatenation, MRP, or greedy consensus. The commands and method version numbers used in these analyses are given in Appendix A.2.2.

We chose MP-EST because it is statistically consistent under the multi-species coalescent model, has been used in several studies [77, 209–211], and had better accuracy than other summary methods in some studies [134]. The greedy consensus is inconsistent under the multi-species coalescent [212], and MRP and concatenation are also inconsistent [83, 136]. Since MP-EST is the only statistically consistent method among those mentioned above, we focus our discussions mostly on MP-EST but show some results using other methods and point out that the same patterns are observed with other methods as well.

MLBS: For each gene or supergene, 200 replicates of bootstrapping is performed using RAxML. Then, 200 different inputs to the summary method are built, where each of these 200 inputs consists of the i^{th} bootstrap replicate across all genes, with $1 \leq i \leq 200$. Next, the summary method (MRP, MP-EST, or Greedy) is run on each of the 200 inputs, and 200 “bootstrapped” species tree replicates are obtained. A greedy consensus of these 200 bootstrap species tree replicates is built, and support values are drawn on this greedy consensus by counting occurrences of each bipartition in the 200 replicates.

Gene and supergene tree estimation: All unbinned and supergene trees were estimated using RAxML under the GTR+Γ model, and with 200 bootstrap replicates of bootstrapping. Although it is recommended to use partitioning in the estimation of supergene trees, due to computational concerns, we did not perform partitioning on avian and mammalian datasets. For the 10-taxon datasets that are smaller, we were able to run fully partitioned analyses. Also, supergene trees on the biological datasets (avian, yeast, vertebrates, and metazoa) were estimated using a partitioned analysis, assigning one partition per gene.

Concatenation analyses: The concatenation analyses of the simulated datasets were performed using an unpartitioned RAxML GTR+Γ maximum likelihood analysis with 20 independent runs with varying random seed numbers, but without bootstrapping. Concatenation analyses of the biological datasets were obtained from the relevant publications, with the exception of the mammalian dataset analysis on the reduced gene dataset, which we re-estimated using an unpartitioned RAxML GTR+Γ maximum likelihood analysis.

Statistical Binning: We developed the statistical binning pipeline using various existing libraries and the resulting code is publicly available in open source[3]. We report results for both weighted and unweighted statistical binning in most cases, and in others we clarify which version is used. An important

[3] https://github.com/smirarab/binning

question is the choice of the bootstrap support threshold $\mathcal{S}$. We note that using 75% for the bootstrap support has been a standard threshold for branch reliability [213], and so 75% represents a reasonable setting for $\mathcal{S}$; however, when the datasets are large, we can afford to be more conservative and pick a smaller threshold. By default, we set two thresholds: a conservative threshold of $\mathcal{S} = 50\%$ that we use for all model conditions of the avian dataset that has more than 1000 genes, and a moderate threshold of $\mathcal{S} = 75\%$ for the mammalian dataset which had fewer than 1000 genes. We compare both thresholds on a subset of data, designed to show the effect of the support threshold, and also show both thresholds for the 10-taxon dataset.

4.3.3 Criteria

For the simulated datasets, we recorded the true species tree and true gene trees generated during the simulation process, which allows us to exactly quantify the topological error in the estimated trees. We measure gene tree error, gene tree distribution error, species tree topological error, and species tree branch length error. We also evaluate the reliability of bootstrap support values measured from MLBS procedure. We use the missing branch rate (also called the false negative rate) for measuring tree error (see Section 2.4.1).

Gene tree error: We measure gene tree error based on individual bootstrap replicates of gene trees, and note that bootstrap replicate gene trees estimated using RAxML are always fully resolved, and hence the missing branch rate is identical to the standard normalized Robinson-Foulds (RF) rate (see Section 2.4.1).

Gene tree distribution error: We also measure how well the entire distribution on gene trees is estimated by comparing triplet frequency distributions calculated from true gene trees and estimated gene trees. To compare gene tree distributions, we calculate how often each of the three possible topologies for every triplet of taxa appears in the set of true gene trees and the set of estimated bootstrap gene trees and supergene trees. Thus for every triplet, we get a distribution based on true gene trees and another one based on estimated gene trees, and we use the Kullback-Leibler [214] divergence statistic to measure how much the estimated distribution diverges from the distribution based on true gene trees. This measure of gene tree distribution error is used because MP-EST uses estimated triplet distributions to construct the species tree, and hence, finding correct triplet frequencies directly affects finding the correct species tree.

Species tree topological error: We measure species tree topological error using both the missing branch and false positive rates. In the vast majority of the cases the estimated species trees are fully resolved, and so the missing branch (false negative) and false positive rates are equal. In a few replicates of the avian simulated dataset, the species trees were incompletely resolved (so instead of 45, only 43 or 44 branches were present in the estimated species tree), and in those cases false positive rates are slightly smaller than the missing branch rates. The general pattern of performance does not change whether error is measured by missing branch (false negative) or false positive rates.

Species tree branch length: We measured estimation error in the species tree branch lengths as follows: given a branch in an estimated species tree that is also present in the true species tree, we record the ratio of the branch length estimated for that branch by MP-EST to the true length of the branch (both in coalescent units) in the true (model) species tree. Since branches on MP-EST trees are in coalescent units, branch lengths directly reflect the predicted amount of ILS. Thus, our branch length evaluation also addresses how well the amount of ILS is estimated by the method.

Species tree bootstrap support: We explore bootstrap support of trees estimated on simulated avian datasets, as follows. We assign relative quality to each edge in an estimated tree, taking bootstrap support into account. The highest quality edges are the true positive branches with the highest bootstrap support, and the lowest quality edges are the false positive branches with the highest bootstrap support, and all other edges fall in between. We order all the edges by their quality, so that the true positive branches come first (with the high support branches before low support branches), followed by the false positive branches (with the low support branches before the high support branches). Given this ordering, we show empirical cumulative distribution functions to compare bootstrap support values of two methods. Thus, we create figures in which the x-axis indicates the edge quality (from very high to very low, as you move from left to right), and the y-axis indicates the fraction of the edges having at least the quality indicated by the x-axis. Thus, the higher the curve, the better the overall quality of the species tree.

Statistical tests: We evaluate the statistical significance of differences in species tree topology using an ANOVA test, with correction for multiple hypothesis using the Benjamini-Hochberg method (also known as False Discovery Rate) [215] ($n = 14$), and setting $\alpha = 0.05$. For each data collection, three two-sided ANOVA tests are performed to establish 1) whether weighted and unweighted binning used with MP-EST are any different, 2) whether binned MP-EST is better than unbinned MP-EST, and 3) whether binned MP-EST is better than concatenation. The two independent variables used in the ANOVA test are 1) the choice of the technique (e.g., weighted versus unweighted binned MP-EST), and 2) the variable parameter in the data collection (e.g. the number of genes for Avian-2 and Avian-3 collection). We report the p-values for the effect of the first independent variable (choice of the technique), and for the interaction between the second variable (varying parameter) and the first variable (choice of the technique).

4.3.4 Computational platform

While any single analysis can be performed in a reasonable time with moderate amount of parallelism, this study involved tens of model conditions, and for each model condition we have looked at 20 replicates (10 replicates for the model conditions with 2000 genes). Thus our total computational time was extremely large: we estimate that we used more than 1,000,000 hours (or more than 100 years) of CPU time overall. Running all these analyses was doable only because of the exceptional computational resources we had access to at TACC supercomputers and a Condor cluster at the University of Texas, Computer Science department.

4.4 Simulation results

We start by exploring two parameters of the binning approach: the binning threshold $\mathcal{S}$ and the use of weighting in statistical binning. We show that on our simulated datasets, weighting does not impact the accuracy of the binning pipeline. We then focus only on unweighted binning and a fixed $\mathcal{S}$ threshold, and present results from an extensive analysis of avian and mammalian datasets.

4.4.1 Binning parameters

We briefly explore the parameter $\mathcal{S}$ and the use of weighting, but note that the impact of algorithmic parameters are generally dependent on model conditions, and our findings in this section need to be interpreted with care.

4.4.1.1 Impact of support threshold

We tested the impact of support threshold $\mathcal{S}$ using two experiments.

Experiment 1- avian 1000 genes, UCE-like: Figure 4.7 compares results of unweighted statistical binning with various thresholds and also concatenation on 10 replicates of the avian dataset with 1000 UCE-like genes. All values of $\mathcal{S}$ resulted in improvements for MP-EST, MRP, and greedy compared to unbinned analyses on this dataset. However, $\mathcal{S} = 30\%$ resulted in lower improvements compared to other thresholds for all methods. Interestingly, very high values (e.g., $\mathcal{S} = 95\%$ here) also did not seem optimal, at least when MP-EST or MRP were used. Results using 50% and 75% thresholds were comparable on this model condition.

Experiment 2- 10-taxon: Figure 4.8 shows the impact of support threshold for both weighted and unweighted binning. On this dataset, increased $\mathcal{S}$ tends to result in improved accuracy in most cases. Here, the 25% threshold ranges from slightly helpful to slightly deleterious, whereas 75% threshold is always improving the accuracy. Thus, the comparison between binned and unbinned analyses depends on the threshold used (and also to a lesser extend whether weighting is used). With 1000 genes, all thresholds of binning result in improved accuracy; these improvements are statistically significant for $\mathcal{S} = 75\%$ with or without weighting ($p < 10^{-4}$) and $\mathcal{S} = 50\%$ without weighting ($p = 0.03$), and are close to significant for $\mathcal{S} = 50\%$ with weighting

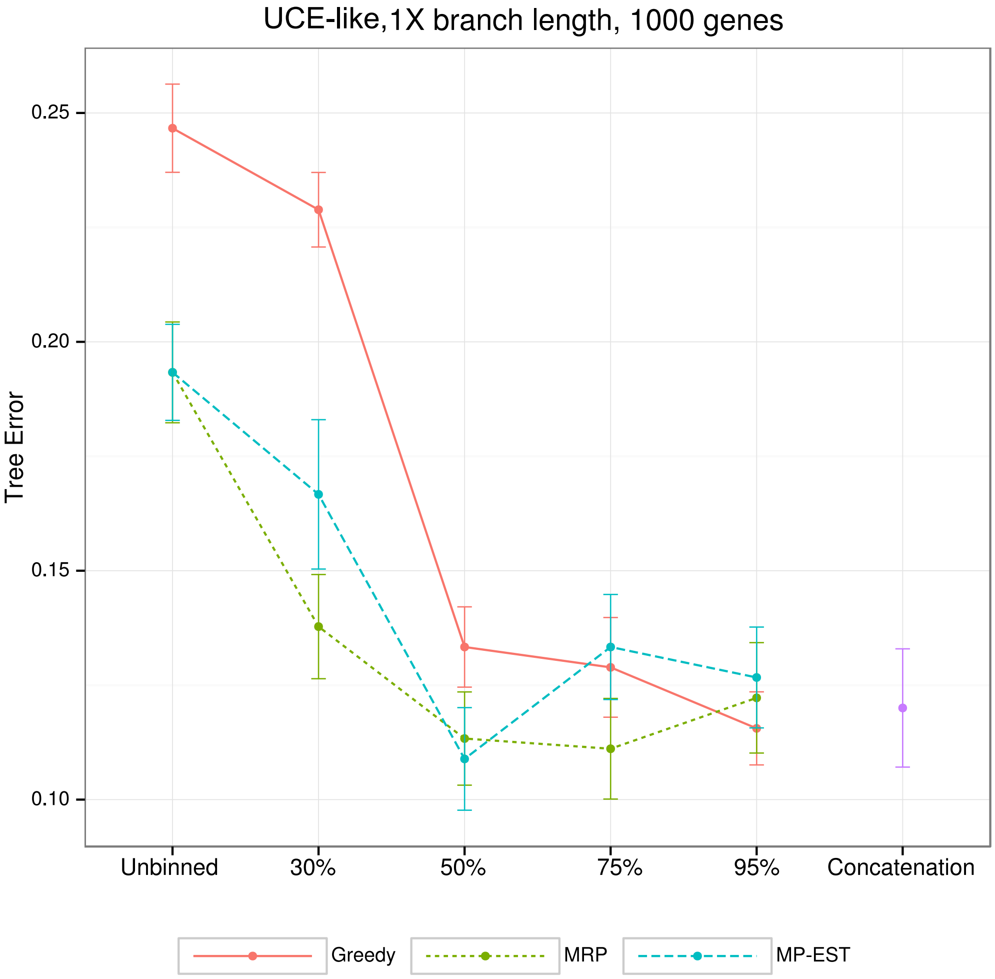


Figure 4.7: **Effects of $\mathcal{S}$ on the avian simulated dataset**. Results are shown for the simulated avian with 10 replicates, 1000 genes, and UCE-like gene tree support. Dots correspond to average tree error and error bars correspond to standard error. Results are shown for unbinned analyses, unweighted binned analyses with 30%, 50%, 75%, and 95% support threshold, and concatenation. Results are shown for three summary methods: Greedy, MRP, and MP-EST.

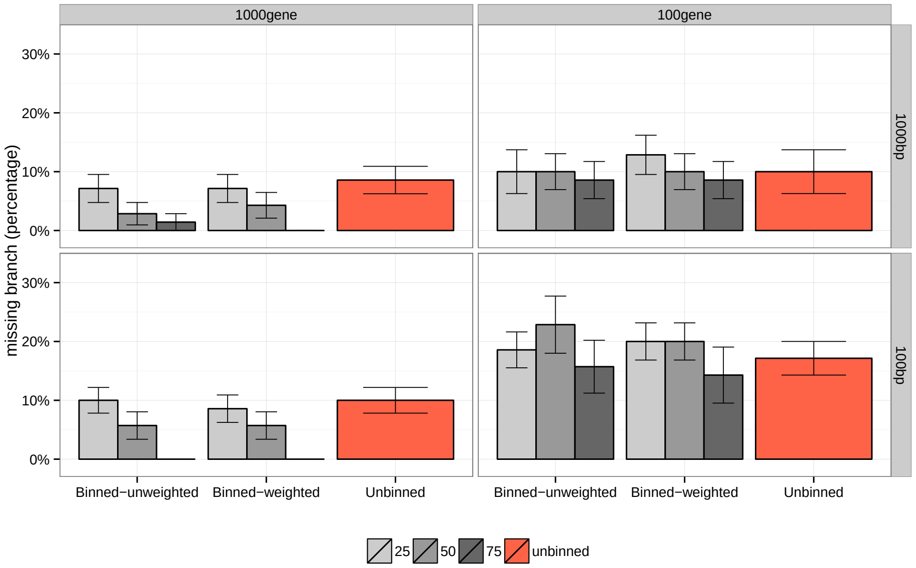


Figure 4.8: **Effects of $\mathcal{S}$ and weighting on the 10-taxon simulated dataset**. Results are shown for the simulated 10-taxon dataset with 10 replicates, 1000 or 100 genes, and either 1000bp or 100bp alignments. Bars show average tree error and error bars show standard error. Results are shown for unbinned analyses and both weighted and unweighted statistical binning with $\mathcal{S} = 25\%, 50\%$, or 75%.

($p = 0.066$), but are not significant for $\mathcal{S} = 25\%$. With 100 genes, however, binning tends to reduce accuracy with $\mathcal{S} = 25\%$ and $\mathcal{S} = 50\%$ (none of the differences are statistically significant for 100 genes). Thus, it is possible to get reductions in accuracy with the statistical binning pipeline, and the choice of the $\mathcal{S}$ parameter can have an impact.

4.4.1.2 Impact of weighting

We showed in Section 4.2.2 that theoretical guarantees of binning depend on the weighting. We now ask whether weighting makes any difference in terms of accuracy. We report results on three collections of the avian dataset, but trends were similar for other datasets that we have analyzed [191].

Figure 4.9 compares weighted and unweighted statistical binning on three collections of the avian dataset: Avian-1 (varying sequence length), Avian-2 (varying number of genes), and Avian-4 (varying level of ILS) and in terms of both species tree accuracy and branch length accuracy. The species tree accuracy was almost indistinguishable between the weighted and unweighted statistical binning. In particular, no statistically significant differences were observed according to a two-way ANOVA test between weighted and unweighted binning ($P > 0.5$ for all three collections).

In terms of branch lengths, there are some differences between weighted and unweighted binning, but the differences tend to be small. To the extent that the two methods produce different branch lengths, weighted binning seems to have slightly better branch length accuracy. While these differences are small, they are consistent across various datasets, and therefore are likely real, and not an artifact.

Summary of parameter exploration: We evaluated two parameters of the statistical binning pipeline: the choice of support threshold $\mathbb{S}$ and the effect of weighting. In the case of weighting, we observed no meaningful differences

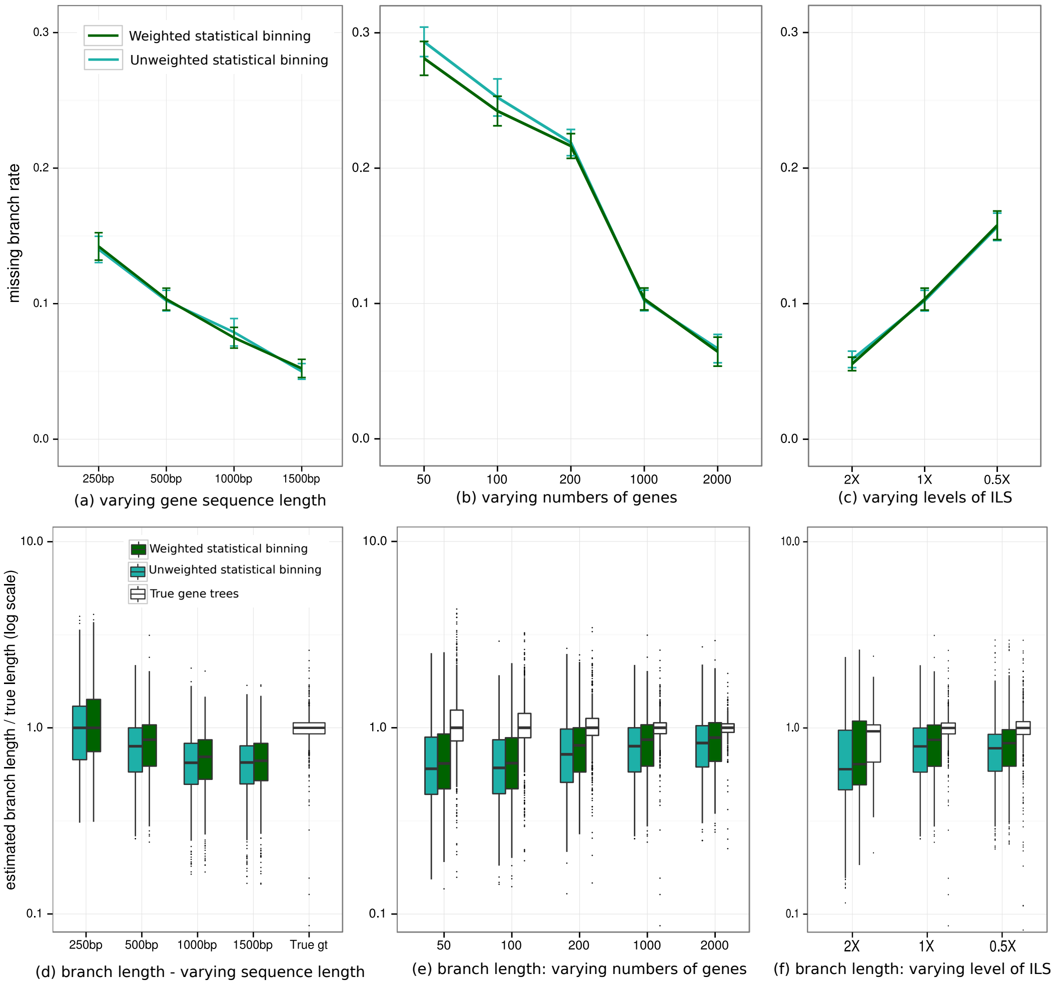


Figure 4.9: **Comparison of weighted and unweighted statistical binning on the avian simulated data**. Species tree topological error (a,b,c) and branch length accuracy (d,e,f) are shown for weighted and unweighted statistical binning ($\mathcal{S} = 50\%$) with MP-EST, and also with true gene trees, on three collections of the avian simulated dataset: (a,d) Avian-1 collection. (b,e) Avian-2 collection, and (c,f) Avian-4 collection. The species tree branch length error is measured as the ratio of estimated branch length to true branch length for branches of the true tree that appear in the estimated tree (1 indicates correct estimation).

between weighted and unweighted pipelines for the large datasets we explored. However, weighting is necessary for theoretical properties of the statistical binning approach and could make an empirical difference for other datasets (likely small datasets where the number of possible tree topologies is limited). It seems harder to generalize in terms of effects of $\mathcal{S}$; however, 50% and 75% seem to work well on various datasets, and are reasonable choices. It is not clear what threshold is ideal, or how one would tailor the threshold in practice for a dataset. As noted before, in our extensive analyses of the avian and mammalian datasets reported in the next sections, we only use unweighted statistical binning, and we set $\mathcal{S} = 50\%$ for avian and $\mathcal{S} = 75\%$ for mammalian datasets.

4.4.2 Avian simulations using unweighted binning

We now report results of extensive experiments on the avian dataset. In these experiments, we always use the unweighted pipeline, and we fix $\mathcal{S} = 50\%$. Thus, throughout this part, when we refer to statistical binning, we are referring to unweighted binning with $\mathcal{S} = 50\%$. We start by comparing unbinned gene trees and binned supergene trees in terms of their accuracy. We then compare species tree accuracy obtained by statistical binning using MP-EST against unbinned MP-EST and concatenation, and also compare branch lengths produced with and without binning. We finish by comparing the traditional and statistical binning pipelines in terms of the bootstrap support of the species trees they produce.

Table 4.2: **Gene tree estimation error, with and without binning for the simulated avian dataset**. Results are shown for Avian-1 data collection. Individual gene tree (GT) error is mean topological distance, measured using the missing branch rate between the true gene tree and all 200 bootstrap replicates of each estimated gene tree. Binned analyses are based on $\mathcal{S} = 50\%$. For the supergene trees, each bootstrap replicate of each supergene tree is compared separately against each true gene tree for the genes put in that bin. We also characterize gene tree distributions by calculating the triplet frequencies for all possible triplets, and we do this both for true and estimated gene trees (using all 200 bootstrap replicates of all genes/supergenes in the case of estimated trees). Thus, we obtain a true and an estimated triplet frequency distribution for each of the triplets. We report the mean Kullback-Leibler (KL) divergence of the estimated distribution from the true distribution. The triplet frequencies are calculated from unweighted supergene trees.

	Individual GT Error		GT Distribution Error (KL)	
	Unbinned	Binned	Unbinned	Binned
Exon-like (250bp)	79%	57%	0.234	0.025
UCE-like (500bp)	69%	57%	0.120	0.008
Intron-like (1000bp)	55%	51%	0.033	0.008
Long Int.-like (1500bp)	46%	45%	0.011	0.007

4.4.2.1 Gene tree (distribution) error

Table 4.2 shows the average gene tree estimation error and gene tree distribution error with and without binning for Avian-1 data collection (default ILS and varying sequence length). Statistical binning improved the estimation of gene tree topologies, with the largest reductions in gene tree estimation error for the exon-like genes, and decreasing impact as the gene sequences increased in length and gene trees increased in BS (Fig. 4.10). We also studied the gene tree error on Avian-4 collection, where we varied the amount of ILS, and observed that the reduction in gene tree error using binning is most pronounced when ILS levels are lower (Fig. 4.10).

The triplet frequencies measured from supergene trees were much more similar to the triplet frequencies measured from true gene trees compared to those measured from unbinned gene trees (Table 4.2). The reductions in triplet gene tree distribution error measured by KL divergence were larger than reductions observed for missing branch rate (Figs. 4.10), and these improvements were especially large for loci with shortest genes and hence the lowest BS.

4.4.2.2 Species tree error

We focus our discussion on the results obtained using MP-EST but also report results using other tools. Figure 4.11 shows topological species tree accuracy for all four collections of the avian simulated dataset, comparing unbinned MP-EST, binned MP-EST, and concatenation. Table 4.3 shows average and standard deviation, and Table 4.4 shows p-values resulting from our ANOVA statistical test (with FDR correction). Figure 4.12 shows results using MRP and greedy. We focus on false negative rates, but Figure 4.13 shows that the same trends hold for the false positive rate.

Avian-1 collection: In this collection, binned MP-EST was consistently and significantly more accurate than concatenation ($p < 10^{-5}$), and was also significantly more accurate than unbinned MP-EST ($p = 0.0001$). For gene trees with the highest BS values (i.e., long intron-like genes), both binned and

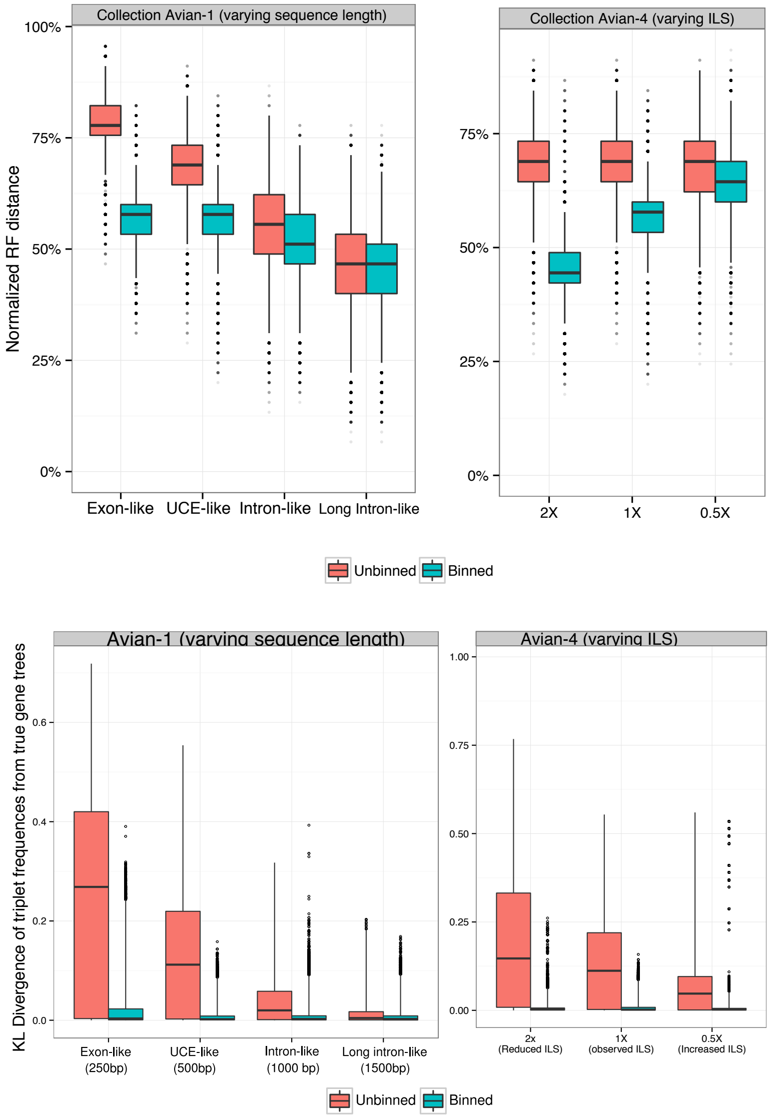


Figure 4.10: **Gene tree estimation error on Avian-1 and Avian-4 data collections**. Top: the distribution of RF distances between true gene trees and all 200 bootstrap replicates of each estimated gene tree. For binned supergene trees ($\mathcal{S} = 50\%$), each bootstrap replicate of each bin is compared separately against *each* true gene tree corresponding to genes put on that bin. Bottom: Divergence of estimated gene trees triplet distributions from triplet distributions of true gene trees. The boxplots show the distribution of the $\binom{48}{3}$ KL divergence measures over 10 replicates of each dataset. The triplet frequencies are calculated from unweighted supergene trees. The whiskers extend to 10 times the inter quartile range.

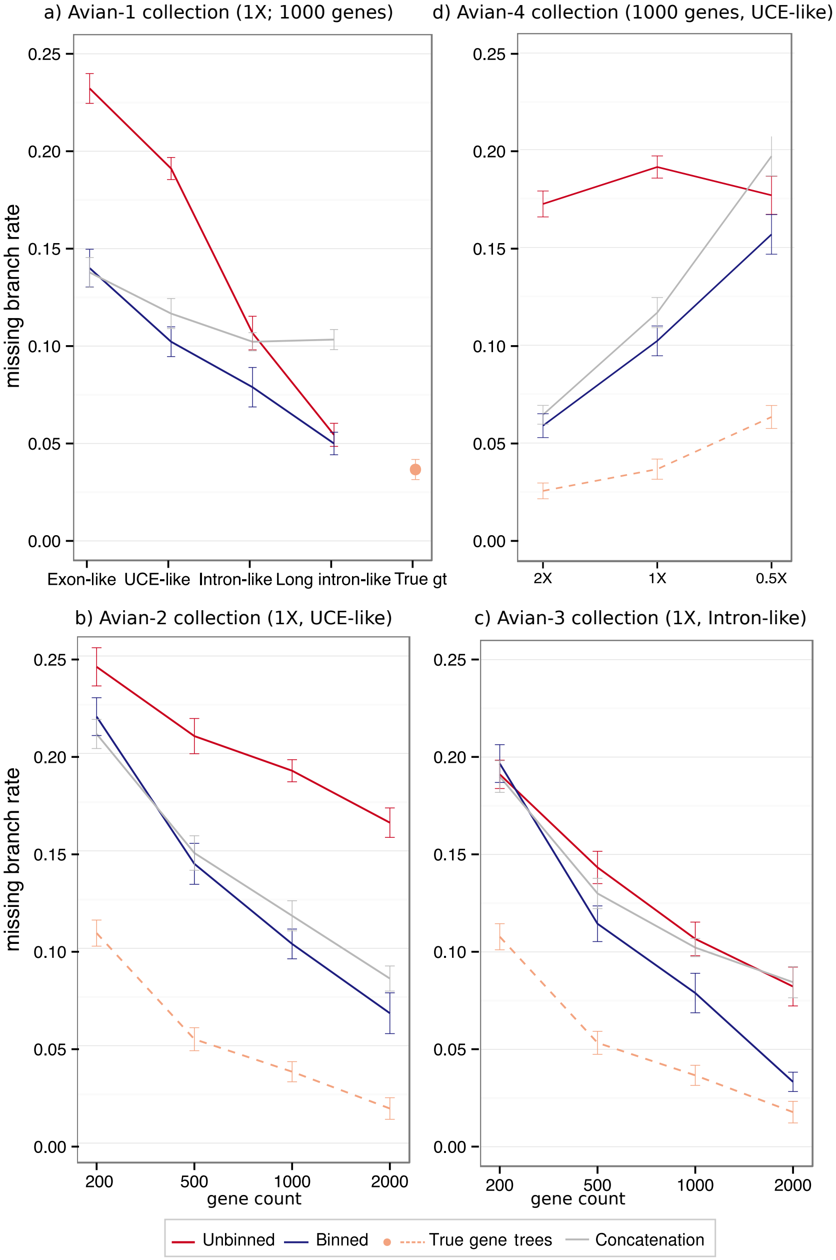


Figure 4.11: **Species tree topological error on the simulated avian datasets using MP-EST**. Results are shown for all collections of the avian dataset, and each line (dot) shows the mean species tree error over 10 replicates for the condition with 2000 genes, and 20 replicates for all other conditions. Error bars show standard error. Results are for unweighted statistical binning, with $\mathcal{S} = 50\%$.

Table 4.3: **Mean and standard deviation of missing branch rates on the avian simulated dataset**. We show average missing branch rates and standard deviation in parentheses. Results are shown for the four collections of the avian simulated dataset, in addition to the results on true gene trees. The best method is shown in bold. Binning results are for the unweighted version, with $\mathcal{S} = 50\%$.

Avian-1		Exon-like	UCE-like	Intron-like	Long intron-like
Greedy	Unbinned	0.288 (0.024)	0.243 (0.030)	0.190 (0.027)	0.154 (0.017)
Greedy	Binned	0.139 (0.026)	0.138 (0.031)	0.158 (0.035)	0.154 (0.020)
MRP	Unbinned	0.251 (0.030)	0.191 (0.026)	0.107 (0.043)	0.082 (0.015)
MRP	Binned	0.128 (0.038)	0.110 (0.034)	0.093 (0.032)	0.090 (0.017))
MP-EST	Unbinned	0.232 (0.034)	0.191 (0.025)	0.107 (0.039)	0.054 (0.026)
MP-EST	Binned	0.140 (0.043)	**0.102 (0.034)**	**0.079 (0.045)**	**0.050 (0.026)**
RAxML	Concatenation	**0.138 (0.034)**	0.117 (0.034)	0.102 (0.021)	0.103 (0.023)
Avian-2		200 genes	500 genes	1000 genes	2000 genes
Greedy	Unbinned	0.270 (0.040)	0.254 (0.032)	0.243 (0.030)	0.229 (0.030)
Greedy	Binned	**0.201 (0.031)**	0.163 (0.036)	0.138 (0.031)	0.127 (0.015)
MRP	Unbinned	0.238 (0.041)	0.199 (0.036)	0.191 (0.026)	0.183 (0.031)
MRP	Binned	**0.201 (0.037)**	**0.131 (0.040)**	0.110 (0.034)	0.082 (0.024)
MP-EST	Unbinned	0.244 (0.044)	0.209 (0.040)	0.191 (0.025)	0.164 (0.024)
MP-EST	Binned	0.219 (0.043)	0.143 (0.047)	**0.102 (0.034)**	**0.067 (0.033)**
RAxML	Concatenation	0.210 (0.033)	0.149 (0.040)	0.117 (0.034)	0.084 (0.020)
Avian-3		200 genes	500 genes	1000 genes	2000 genes
Greedy	Unbinned	0.220 (0.027)	0.208 (0.035)	0.190 (0.027)	0.180 (0.030)
Greedy	Binned	0.196 (0.035)	0.166 (0.022)	0.158 (0.035)	0.140 (0.018)
MRP	Unbinned	0.174 (0.038)	0.133 (0.037)	0.107 (0.043)	0.096 (0.033)
MRP	Binned	**0.170 (0.045)**	0.122 (0.026)	0.093 (0.032)	0.084 (0.014)
MP-EST	Unbinned	0.191 (0.033)	0.143 (0.037)	0.107 (0.039)	0.082 (0.032)
MP-EST	Binned	0.197 (0.043)	**0.114 (0.041)**	**0.079 (0.045)**	**0.033 (0.016)**
RAxML	Concatenation	0.190 (0.036)	0.130 (0.035)	0.102 (0.021)	0.084 (0.025)
Avian-4		2X	1X	0.5X	
Greedy	Unbinned	0.226 (0.028)	0.243 (0.030)	0.293 (0.026)	
Greedy	Binned	0.077 (0.022)	0.138 (0.031)	0.262 (0.026)	
MRP	Unbinned	0.163 (0.019)	0.191 (0.026)	0.222 (0.042)	
MRP	Binned	**0.058 (0.017)**	0.110 (0.034)	0.174 (0.029)	
MP-EST	Unbinned	0.172 (0.030)	0.191 (0.025)	0.177 (0.044)	
MP-EST	Binned	0.059 (0.027)	**0.102 (0.034)**	**0.157 (0.045)**	
RAxML	Concatenation	0.064 (0.022)	0.117 (0.034)	0.197 (0.045)	
(True gene trees, 1X ILS)		200 genes	500 genes	1000 genes	2000 genes
Greedy	True gene tree	0.143 (0.026)	0.131 (0.024)	0.123 (0.025)	0.120 (0.021)
MRP	True gene tree	0.117 (0.034)	0.096 (0.024)	0.076 (0.018)	0.058 (0.012)
MP-EST	True gene tree	**0.110 (0.030)**	**0.053 (0.026)**	**0.037 (0.023)**	**0.018 (0.018)**
(True gene trees, 1000 genes)		2X	1X	0.5X	
Greedy	True Gene Trees	0.066 (0.010)	0.123 (0.019)	0.181 (0.046)	
MRP	True Gene Trees	0.052 (0.011)	0.076 (0.019)	0.120 (0.024)	
MP-EST	True Gene Trees	**0.026 (0.018)**	**0.037 (0.019)**	**0.063 (0.030)**	

Table 4.4: **Statistical significance for simulated avian datasets**. Statistical significance of differences in species tree topology (dependent variable) are evaluated using a two-sided ANOVA test, with correction for multiple hypothesis using Benjamini Hochberg [215] ($n = 14$ including 8 tests performed here, and 6 tests performed for the mammalian dataset), and setting $\alpha = 0.05$. The two independent variables used in the ANOVA test are 1) the choice of the technique (Binned MP-EST vs. Unbinned MP-EST, and also Binned MP-EST vs. Concatenation), and 2) the variable model parameter (e.g. number of genes for Avian-2 collection). Binning results are for the unweighted version, with $\mathcal{S} = 50\%$. Top part shows p-values for impact of the choice of the technique. The bottom part shows p-values for the interaction between the varying parameter and choice of the technique. Thus p-values in the bottom part should be interpreted with regard to questions of the following form: "is the relative performance of binned MP-EST and unbinned MP-EST (or concatenation) affected by the choice of varying parameter." For example, for Avian-1 collection, the p-value shown under Binned vs. Unbinned indicates that the gene tree support has a statistically significant impact on the relative performance of binned and unbinned MP-EST.

Collection	2nd variable	Binned vs. Unbinned	Binned vs. Concat.
Significance of choice of technique			
Avian-1	support	$\mathbf{p < 10^{-5}}$	$\mathbf{p = 0.00012}$
Avian-2	# genes	$\mathbf{p < 10^{-5}}$	$p = 0.37100$
Avian-3	# genes	$\mathbf{p = 0.00212}$	$\mathbf{p = 0.01058}$
Avian-4	ILS	$\mathbf{p < 10^{-5}}$	$\mathbf{p = 0.00418}$
Impact of varying parameter on the choice of the technique			
Avian-1	support	$\mathbf{p < 10^{-5}}$	$\mathbf{p = 0.01162}$
Avian-2	# genes	$\mathbf{p = 0.00330}$	$p = 0.56569$
Avian-3	# genes	$p = 0.10614$	$p = 0.08705$
Avian-4	ILS	$\mathbf{p < 10^{-5}}$	$p = 0.15076$

unbinned MP-EST species trees had approximately the same error (Table 4.3). However, as gene tree BS values decreased, the improvements obtained by binned MP-EST compared to unbinned MP-EST increased, and this effect was statistically significant ($p = 0.003$). On this collection, concatenation was generally more accurate than unbinned MP-EST, except for gene trees with the highest BS. Results for MRP and Greedy showed similar trends (Fig. 4.12).

Avian-2 and Avian-3 collections: On Avian-2 and Avian-3 collections, where we varied the number genes with fixed sequence length, binned MP-EST was more accurate than unbinned MP-EST ($p < 10^{-5}$ for UCE-like and $p = 0.002$ for intron-like markers). Furthermore, the advantage provided by binning increased with the number of genes in Avian-2 collection (UCE-like), and the impact was statistically significant with $p = 0.003$; a similar pattern seems to also hold for Avian-3 collection (intron-like loci), but the impact was not statistically significant $p = 0.106$.

Binned MP-EST tended to be more accurate than concatenation on both UCE-like and intron-like loci, but the differences are statistically significant only for intron-like genes ($p = 0.011$). The improvement of binned MP-EST over concatenation appeared to increase with the number of intron-like loci, but the interaction effect is not statistically significant ($p = 0.087$).

Avian-4 Collection: When we varied the amount of ILS, regardless of the amount of ILS, binned MP-EST had lower average tree error than both un-

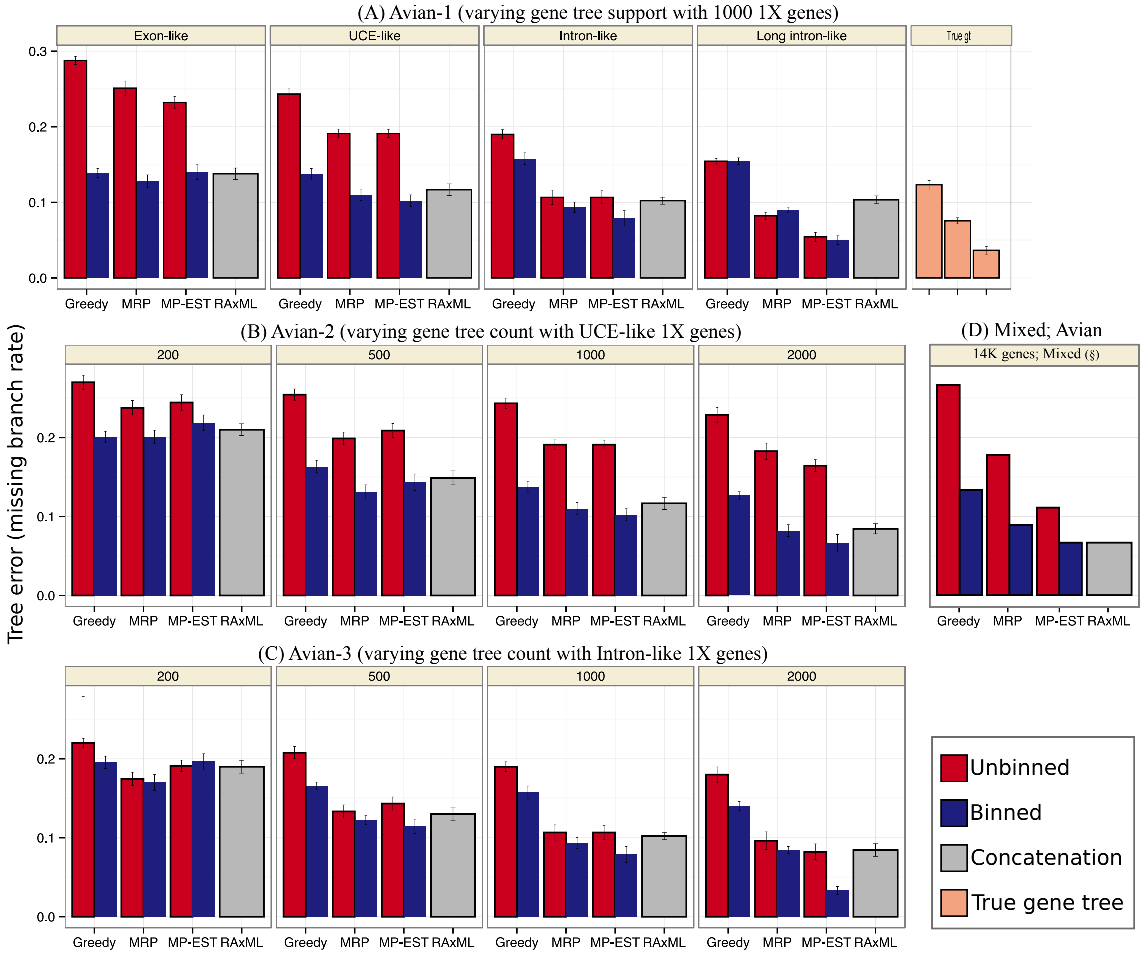


Figure 4.12: **Simulation results including MRP and Greedy on avian datasets**. Bars show average missing branch rates and error-bars show standard error. Results are over 20 replicates everywhere except 2000 genes model conditions, which is based on 10 replicates, and the mixed model condition, which is based on only 1 replicate. Binning results are for the unweighted version, with $\mathcal{S} = 50\%$.

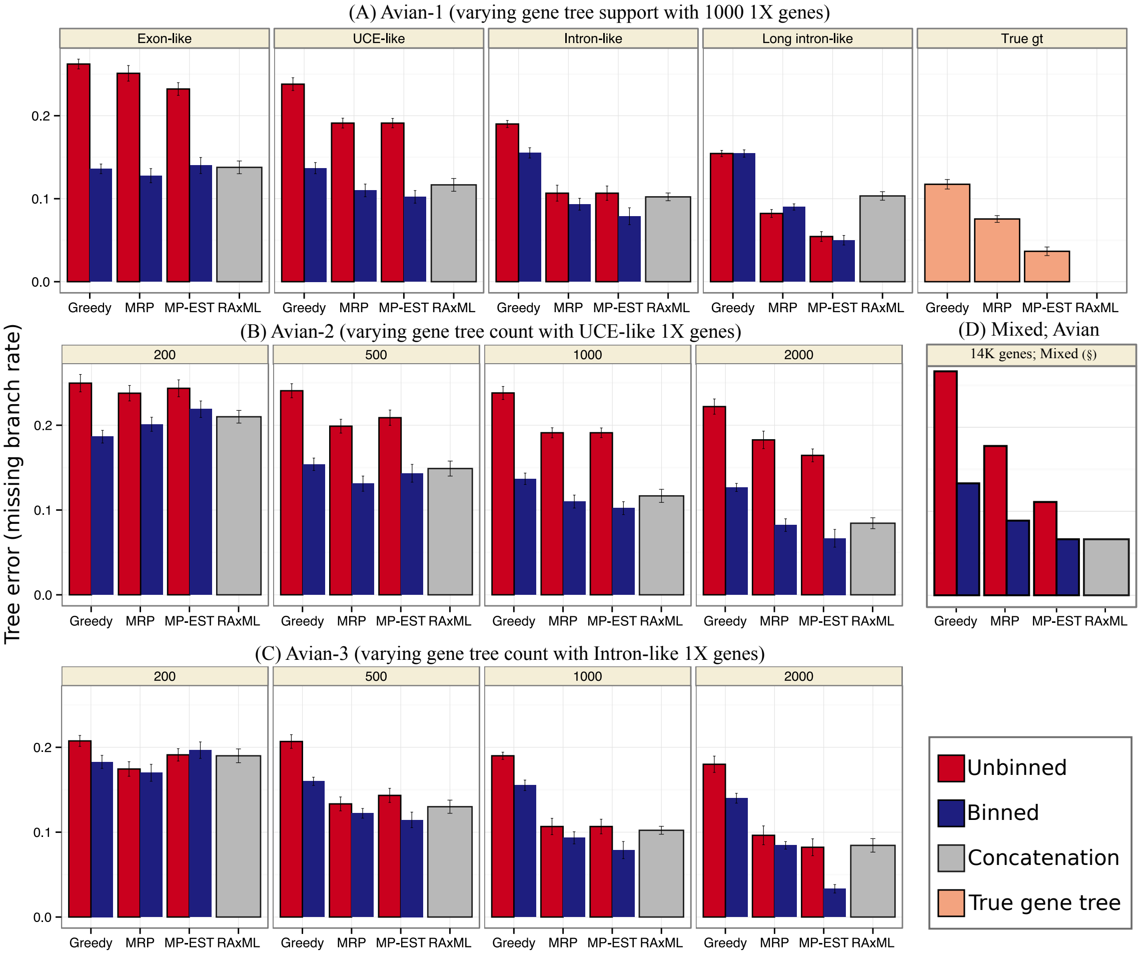


Figure 4.13: **False positive error rates on the avian simulation**. In some rare cases on the simulated avian datasets, the greedy consensus trees produced by the multi-locus bootstrapping procedure were missing one or two edges, and hence had small polytomies. In such cases, the missing branch (false negative) rate and false positive branch rates can be slightly different. For completeness, we show the false positive rates. Results are consistent with those observed on false negatives. Binning results are for the unweighted version, with $\mathcal{S} = 50\%$.

binned MP-EST and concatenation. The difference between binned and unbinned MP-EST was statistically significant ($p < 10^{-5}$), and so was the difference between binned MP-EST and concatenation ($p = 0.004$). Furthermore, reducing the ILS level (2X condition) increased the impact of binning, and increasing the ILS level (0.5X condition) decreased the impact ($p < 10^{-5}$ for the interaction effect).

Mixed condition: On the mixed model condition with 14k genes and BS resembling the avian dataset, concatenation and binned MP-EST each had 7% error, while all the other methods had at least 11% error (Fig. 4.12).

Summary of topological species tree error on avian simulations: Binned MP-EST had significantly lower topological error compared to unbinned MP-EST for all collections and binned MP-EST was also significantly more accurate than concatenation in Avian-1, Avian-3, and Avian-4 collections, but not in the Avian-2 collection. Gene BS (controlled by sequence length), number of genes, and the amount of ILS had a significant impact on the relative performance of binned and unbinned MP-EST, but only BS impacted the choice between binned MP-EST and concatenation.

4.4.2.3 Branch length accuracy

Figure 4.14 shows the branch length accuracy for the four collections of the avian dataset. MP-EST always underestimated species tree branch

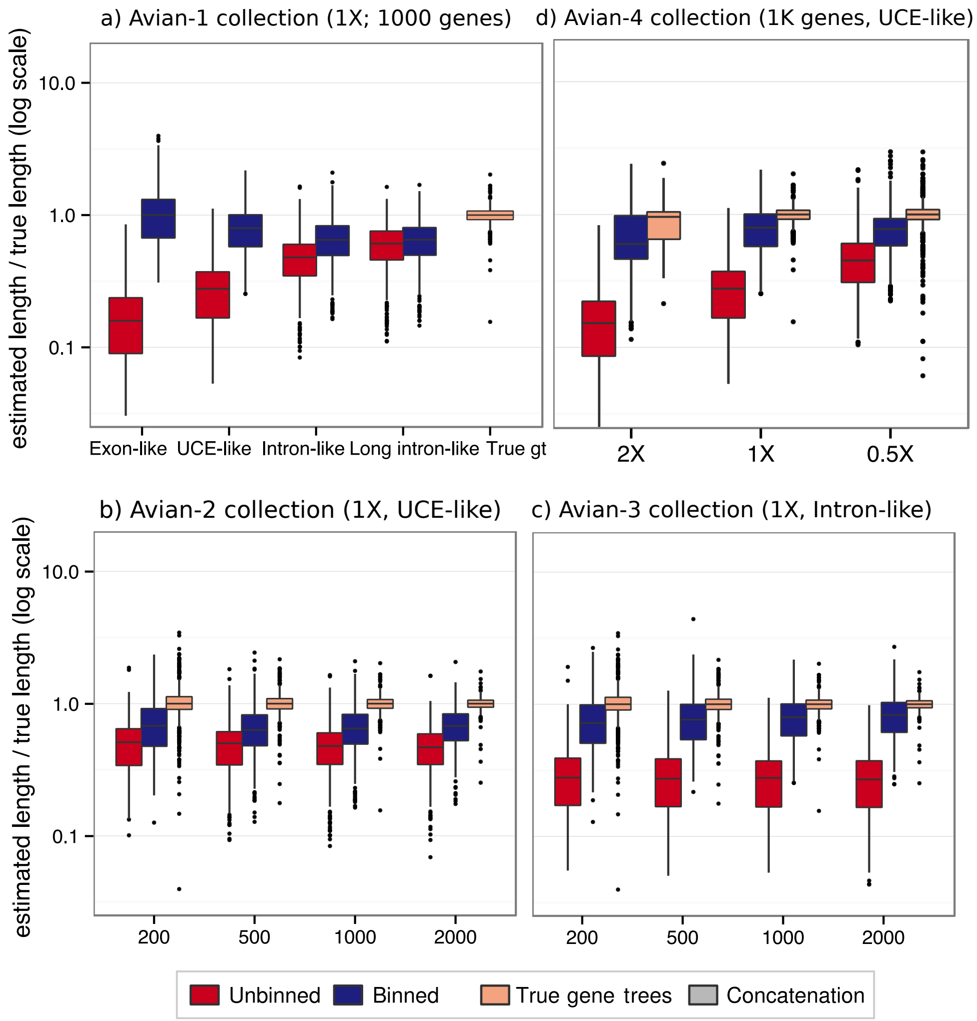


Figure 4.14: **Species tree branch length accuracy on the simulated avian datasets estimated using MP-EST**. Results are shown for all collections of the avian dataset. Boxplots show the distribution of the ratio of estimated branch length to true branch length for branches of the true species tree that appear in the estimated tree; thus, 1 indicates correct estimation. Results are over 10 replicates for the condition with 2000 genes, and 20 replicates for all other conditions. Binning results are for the unweighted version, with $\mathcal{S} = 50\%$. Note that y-axis is shown in logarithmic scale.

lengths in coalescent units when analyzing estimated gene trees. In contrast, when true gene trees are used, branch length are estimated very well, indicating that underestimation of branch lengths is a result of gene tree estimation error. When unbinned gene trees are used, the underestimation can be close to an order of magnitude, whereas the binned MP-EST trees had more accurate branch lengths. Improvements obtained by binning are largest for cases where gene trees have low BS (e.g., Exon-like), or when the amount of ILS is low. Because branch lengths determine the amount of ILS, underestimating branch lengths directly means overestimating ILS.

4.4.2.4 Species tree bootstrap support

Figure 4.15 shows the commutative bootstrap support distributions of both true and false positive branches for Avian-1 and Avian-4 collections (similar patterns observed on the other two collections). Binning improve bootstrap support in the sense that using binning increases the number of highly supported true positive branches and decreases the number of highly supported false positives. However, the sequence length (and hence gene tree BS) modulates the impact of binning on bootstrap support, so that the largest impact is for the Exon-like genes (250bp), and there is no discernible impact for the Long intron-like genes (1500bp). ILS levels also impact how binning affects the bootstrap support, so that the biggest improvement in bootstrap support is obtained for the lowest ILS level (2X branch lengths).

These results demonstrate that gene tree estimation error not only

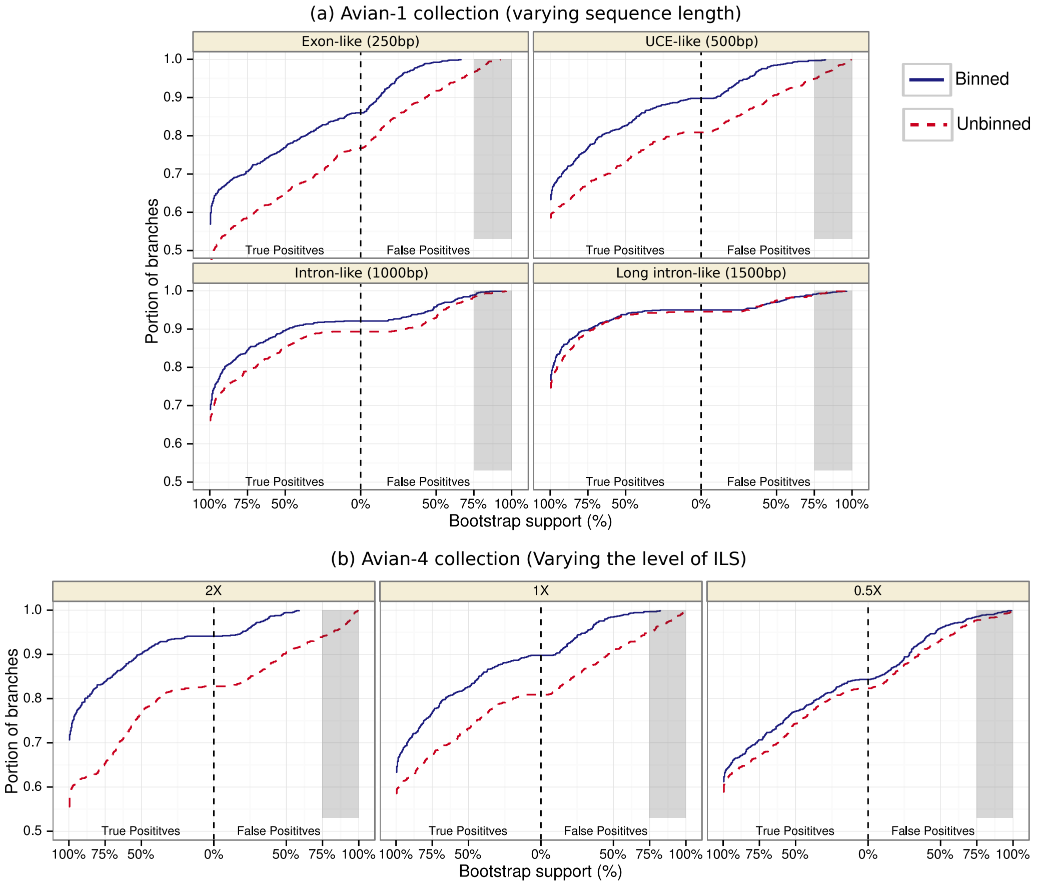


Figure 4.15: **Bootstrap support comparison on the avian simulated dataset**. We show cumulative distribution of the bootstrap support values of true positive and false positive edges estimated by binned and unbinned MP-EST on avian datasets. We show results for Avian-1 and Avian-4 collections, but similar patterns are observed elsewhere. To produce the graph, we order the branches in the estimated species tree by their quality, so that the true positives with high support come first, followed by lower support true positives, then by false positives with low support, and finally by false positives with high support. The false positive branches with support above 75% are the most troublesome, and the grey area indicates highly supported false positives. When the curve for a method lies above the curve for another method, then the first method has better bootstrap support. Binning results are for the unweighted version, with $\mathcal{S} = 50\%$.

can result in lack of resolution in the species tree (i.e., low support branches that are true or false), but can also result in highly supported false positive branches. Binning not only increases support, but also reduces highly supported false positives.

4.4.3 Mammalian simulations using unweighted binning

Similar to the experiments on the avian dataset, here, we always use the unweighted version of statistical binning. We also fix $\mathcal{S} = 75\%$. Thus, throughout this section, when we refer to statistical binning, we are referring to unweighted binning with $\mathcal{S} = 75\%$.

Patterns observed on the avian dataset were also seen on the mammalian dataset, but with less stark contrast between unbinned and binned pipelines. We briefly discuss gene tree error, species tree topological error, and species tree branch length accuracy on the mammalian datasets.

Gene tree error: Just like the avian dataset, on the mammalian dataset, binning reduced gene tree estimation error, and improvements were larger for the model condition with shorter sequences and lower BS (Table 4.5). The reductions in gene tree distribution error were also high, and were higher for the 500bp model condition compared to the 1000bp condition. Thus, when unbinned gene trees have high error, binning can improve gene tree accuracy.

Species tree topological error: Binned MP-EST generally either matched or improved upon both unbinned MP-EST and concatenation (Fig. 4.16).

Table 4.5: **Gene tree estimation error, with and without binning for the mammalian avian dataset**. Results are shown for Mammalian-1 and -2 data collections. See Table 4.2 for a description of the measures shown. Binned results are with $\mathcal{S} = 75\%$.

	Individual GT Error		GT Distribution Error (KL)	
	Unbinned	Binned	Unbinned	Binned
63% BS (500bp)	43%	35%	0.119	0.019
79% BS (1000bp)	27%	26%	0.038	0.027

On the moderate (63%) BS trees, binned MP-EST and concatenation had close accuracy (with no statistically significant differences; see Table 4.6), but unbinned MP-EST was significantly less accurate than binned MP-EST ($p < 10^{-5}$), and some conditions showed substantial differences (e.g., 800 loci). On higher BS (79%) loci, binned MP-EST was significantly more accurate than concatenation ($p = 0.003$), but there were no statistically significant differences between binned MP-EST and unbinned MP-EST.

On Mammalian-3 collection, the differences between binned and unbinned MP-EST were statistically significant($p = 0.0001$), but differences between binned MP-EST and concatenation were not significant (Table 4.6). The impact of ILS level on the mammalian datasets was as expected: more improvements were obtained for lower ILS; however, the impact of ILS level was not statistically significant.

On the mixed model condition, which most closely resembles the real mammalian dataset in terms of the number of genes and gene tree support, binned MP-EST had only 1.8% error, concatenation had 3.7% error, and un-

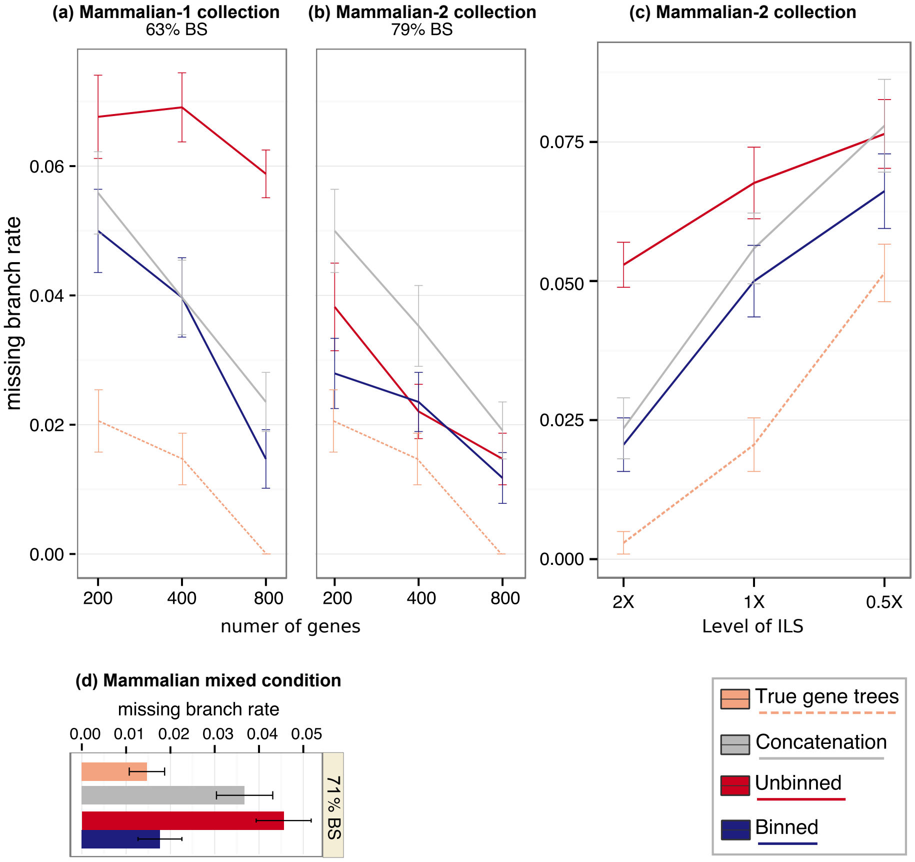


Figure 4.16: **Species tree topological error on the simulated mammalian datasets using MP-EST**. Results are shown for both collections of the mammalian dataset, and the mixed mammalian model condition. Each line or bar shows the mean species tree error over 20 replicates and error bars show standard error. Results are shown separately for gene trees with 63% and 79% bootstrap support. Panel C shows topological error for a mixed dataset with 200 genes of 63% BS level, and 200 genes of 79% BS level. Binning results are for the unweighted version with $\mathcal{S} = 75\%$.

Table 4.6: **Statistical significance for simulated mammalian datasets**. Statistical significance of differences in species tree topology (dependent variable) are evaluated using a two-sided ANOVA test, with correction for multiple hypothesis using Benjamini Hochberg [215] ($n = 14$ including 6 tests performed here, and 8 tests performed for the avian dataset), and setting $\alpha = 0.05$. The two independent variables used in the ANOVA test are 1) the choice of the technique (Binned MP-EST vs. Unbinned MP-EST, and also Binned MP-EST vs. Concatenation), and 2) the variable model parameter (e.g. ILS levels for Mammalian-3 collection). Binning results are for the unweighted version with $\mathcal{S} = 75\%$. Top part shows p-values for impact of the choice of the technique. The bottom part shows p-values for the interaction between the varying parameter and choice of the technique. Thus p-values in the bottom part should be interpreted with regard to questions of the following form: "is the relative performance of binned MP-EST and unbinned MP-EST (or concatenation) affected by the choice of varying parameter." For example, for Mammalian-3 collection, the p-value shown under Binned vs. Unbinned indicates that the level of ILS has no statistically significant impact on the relative performance of binned and unbinned MP-EST.

Collection	2nd variable	Binned vs. Unbinned	Binned vs. Concat.
Significance of choice of technique			
Mammalian-1	# genes	$\mathbf{p < 10^{-5}}$	$p = 0.34300$
Mammalian-2	# genes	$p = 0.35690$	$\mathbf{p = 0.00315}$
Mammalian-3	ILS	$\mathbf{p = 0.00012}$	$p = 0.24818$
Impact of varying parameter on the choice of the technique			
Mammalian-1	# genes	$p = 0.12071$	$p = 0.78500$
Mammalian-2	# genes	$p = 0.56569$	$p = 0.50470$
Mammalian-3	ILS	$p = 0.25511$	$p = 0.78500$

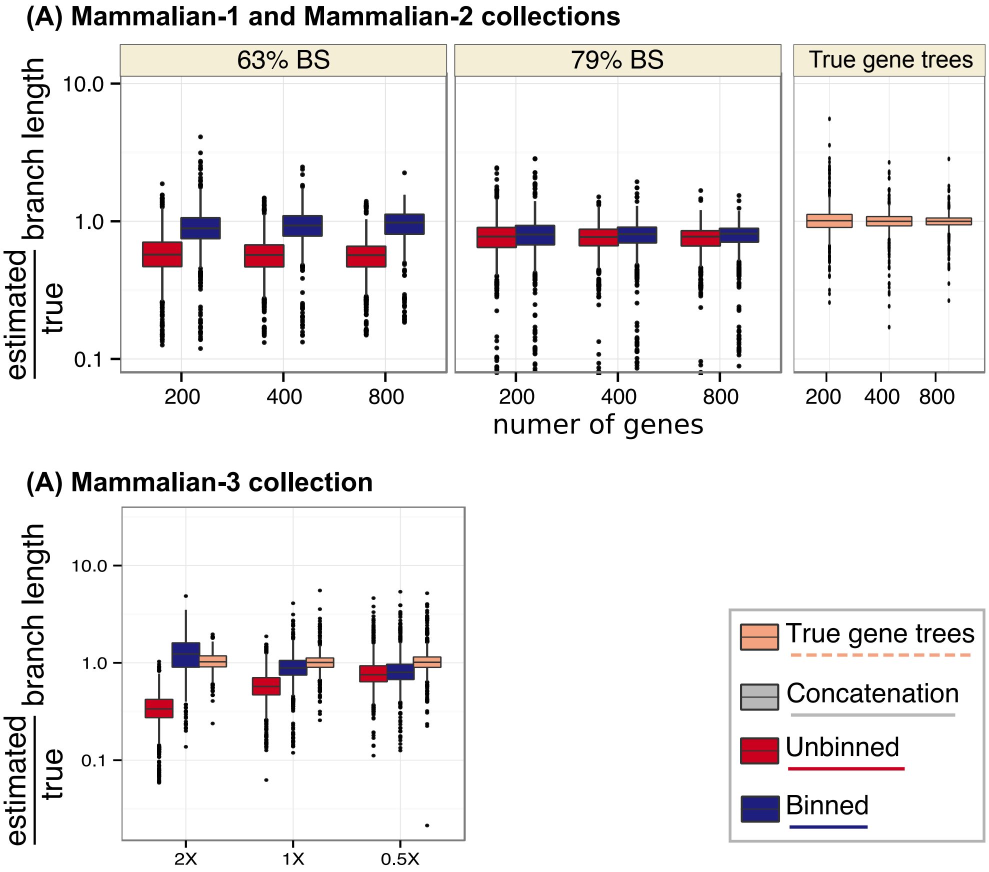


Figure 4.17: **Species tree branch length accuracy on the simulated mammalian datasets estimated using MP-EST**. Results are shown for both collections of the mammalian dataset. Boxplots show the distribution of the ratio of estimated branch length to true branch length for branches of the true species tree that appear in the estimated tree; thus, 1 indicates correct estimation. Results are over 20 replicates for all other conditions. Binning results are for the unweighted version with $\mathcal{S} = 75\%$. Note that y-axis is shown in logarithmic scale.

binned MP-EST had 4.6% error (Fig. 4.16).

Species tree branch length: Similar to avian dataset, MP-EST underestimated species tree branch lengths in coalescent units when given estimated gene trees but had good accuracy with true gene trees. The binned MP-EST trees had more accurate branch lengths (Fig. 4.17), especially for lower BS gene trees, and for lower levels of ILS.

4.5 Biological results

All of our biological datasets shows evidence of gene tree discord (see Fig. 4.2 for avian and Fig. 4.18 for other datasets), but they vary with respect to average BS (Fig. 4.19). For all datasets, the use of binning increased average gene tree support, and in many cases also reduced gene tree discordance. We discuss our findings on each of the four biological datasets in turn.

4.5.1 Avian

Gene trees: Evidence for ILS in the avian dataset is extensively reported in [39]. The avian dataset has very low average bootstrap support for almost all loci (Figs. 4.19 and Fig. 4.6) and large topological distances between estimated gene trees (Fig. 4.18). The average topological distance between estimated gene trees and the concatenation tree on the full set of 14K genes was very high (74%). However, most loci had low phylogenetic signal, with the result that the average BS for the estimated gene trees was very low, only 32%. Among

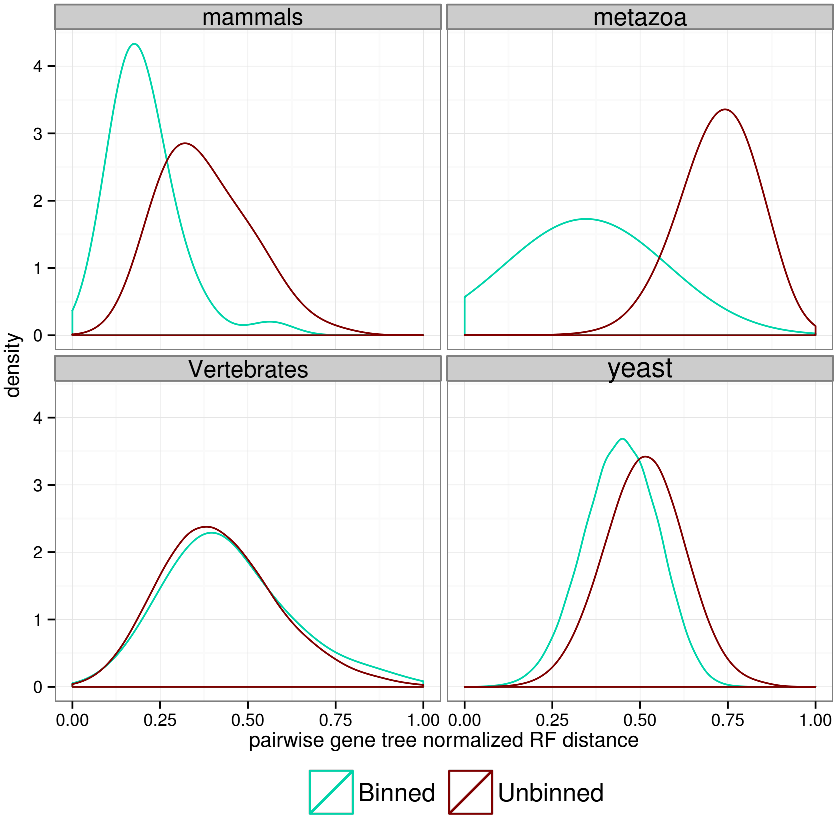


Figure 4.18: **Gene tree incongruence for biological datasets**. We measure gene tree incongruence using pairwise normalized RF distance between all pairs of estimated gene trees, with and without binning. For each of the four biological datasets, the distributions of pairwise gene tree distances are shown as kernel density plots [216] drawn using R [217].

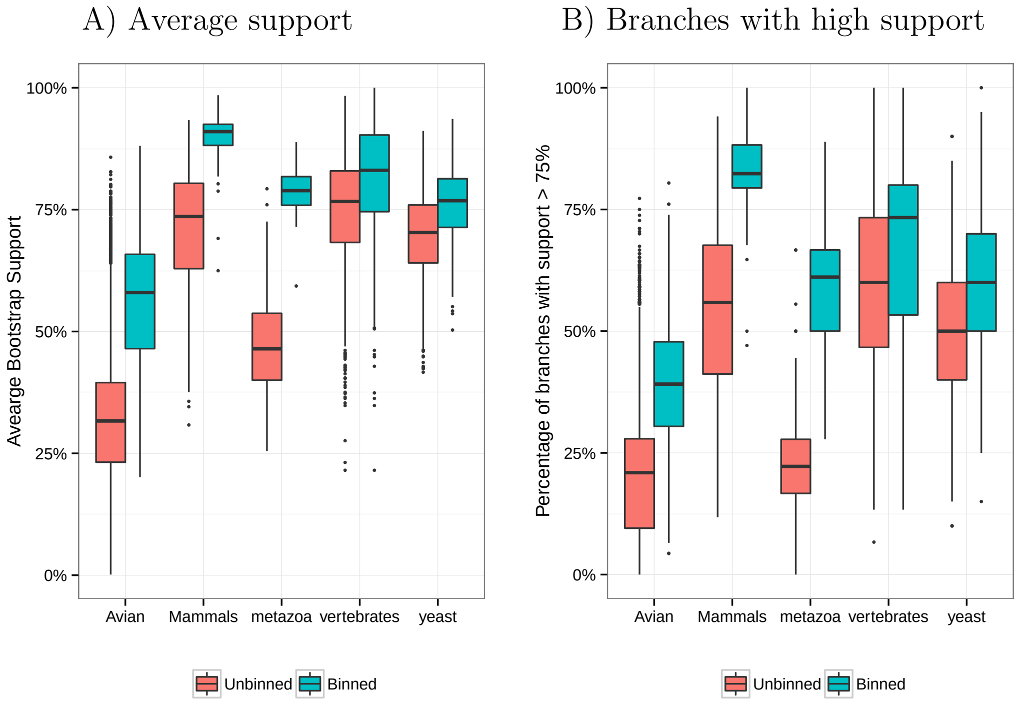


Figure 4.19: **Gene tree bootstrap support for biological datasets**. A) Boxplots showing distribution of average bootstrap support across all estimated gene trees. B) Boxplots showing distribution of the percentage of branches in each gene tree that have support above 75%. Distributions are shown for both original unbinned gene trees, and the supergene trees.

the 14,446 gene trees, introns had the highest BS values (48%), and also had a somewhat lower distance to the concatenation tree (63%). Therefore, the large topological distance between estimated trees is to some extent a result of poor phylogenetic signal in the gene sequences.

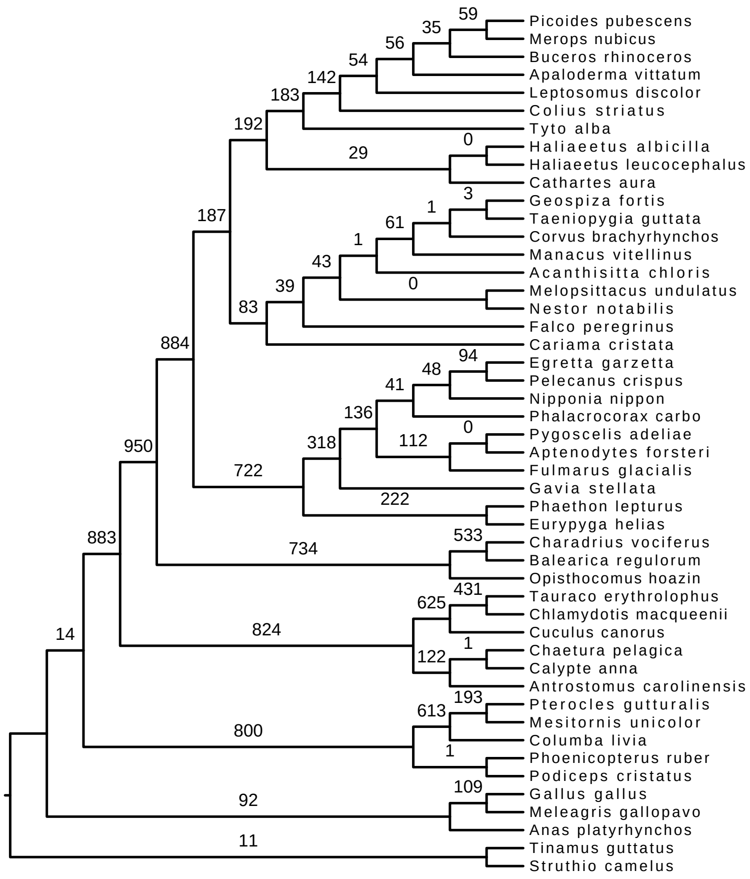


Figure 4.20: **Evidence for ILS in the avian dataset**. On each branch of the concatenation tree reported in [39], we show the number of intron gene trees (out of a total of 2516 loci) that rejected that branch with a BS of at least 75%. Edges with lots of highly supported conflict are closer to the base.

However, substantial evidence of strongly supported gene tree conflict remained even after taking these low bootstrap support values into account. First, as shown in Figure 4.20, many of the branches in the TENT are rejected by a large number of intron gene trees with high support (at least 75%); furthermore, there are many short branches adjacent to each other in the tree, as expected in a rapid radiation scenario. This is a condition that leads to high levels of ILS. Similarly, comparing gene trees to each other revealed substantial levels of discordance. On average, two estimated intron gene trees differed in 1.3 strongly supported edges (at least 75% support). Thus, a very high level of discordance is observed in the avian dataset, some of which is clearly due to lack of support. However, a lot of discordance is observed even among highly supported branches, providing evidence of real gene tree discord.

Species Trees: An unbinned MP-EST analysis of the full 14K loci produced a tree with low to moderate support for some branches (Fig. 4.21). Moreover, the unbinned tree failed to recover four key clades (Columbea, Cursores, Otidimorphae, and Australaves; all shown on Fig. 4.21). These clades are recovered consistently in other analyses on the full genome dataset [39], including all analyses that included introns and UCEs, and also unbinned MP-EST analyses restricted to non-coding data. Failure to recover Australaves is particularly surprising, as it has been recovered in the literature using various types of data [16, 195, 218, 219].

Weighted and unweighted binned MP-EST on all 14K loci ($\mathcal{S} = 50\%$

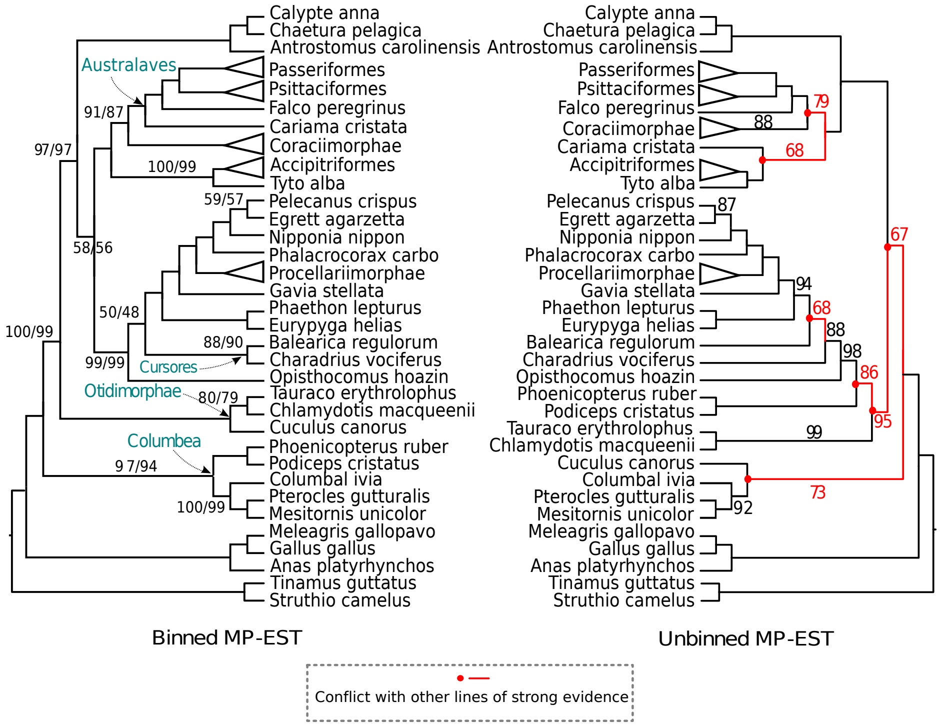


Figure 4.21: **Trees computed on the avian biological dataset using MP-EST**. We show results with weighted and unweighted binning (left), and unbinned analyses (right). We used 50% bootstrap support threshold for binning. Supergene trees were estimated using fully partitioned analyses. MP-EST with weighted and unweighted binning returned the same tree. The branches on the binned MP-EST tree are labeled with two support values side by side: the first is for unweighted binning and the second is for weighted binning; branches without designation have 100% support. Branches in red indicate contradictions to other sources of evidence from [39].

since we have more than 1000 genes) generated the same exact tree on this dataset with small variations in bootstrap support (Fig. 4.21). In contrast to unbinned analyses, binned MP-EST trees had highly supported branches throughout most of the tree, and recovered all key clades. The unweighted binned MP-EST tree was used by the Avian consortium as one of the two possible hypotheses of bird evolution [39].

An unbinned MP-EST tree generated on the introns-only dataset [39] had 31 out of 45 edges with 100% support and 34 edges with 95% or higher BS; it also recovered all the key clades missing from the unbinned MP-EST tree computed on the full set of 14K loci. However, the unweighted binned MP-EST analysis on the introns-only dataset also recovered all the key clades, and had higher support (33 edges with 100% support and 35 with 95% support or more), with increased support for some key novel clades [39]. Thus, these intron-only MP-EST trees are more congruent with other reliable analyses. This similarity is likely because intron gene trees have better support than other partitions, and (as shown in our simulation study) when gene trees have high support, even unbinned MP-EST can have high accuracy.

4.5.2 Mammalian

The mammalian dataset has gene trees with substantially higher average BS (71%), but also demonstrates substantial gene tree incongruence (see Figs. 4.19 and 4.18). Differences between MP-EST and concatenation (using ML) were observed for tree shrews and bats: the concatenated analysis put

Scandentia (tree shrews) as sister to Glires (Rodentia/Lagomorpha), while the MP-EST analysis put Scandentia as sister to primates [77].

We re-analyzed this dataset and identified 21 loci with mis-labeled sequences (subsequently confirmed by the authors) plus two outlier loci [190]. We removed these 23 loci, and re-analyzed the data using concatenation and both binned and unbinned MP-EST. We recovered a concatenation tree topologically identical to the concatenation tree in [77]. The unbinned MP-EST tree on this reduced gene set was similar to the unbinned MP-EST tree reported in [77], but had lower support for tree shrews as sister to primates (99% in [77], 64% with our analysis), and there was one topological difference among low support edges. These differences are most probably due to the fact that we have re-estimated our gene trees using RAxML, whereas the authors had used an inferior tree search tool (only NNI moves).

The weighted and unweighted binned MP-EST ($\mathcal{S} = 75\%$ since we have less than 1000 genes) were identical on the mammalian dataset, with small differences in support (less than 3%). The two binned trees were also similar to the unbinned MP-EST tree on the reduced gene set with one difference: the tree shrews were sister to Glires with 80% support in both binned MP-EST trees, just like their position in the concatenation tree. Thus, the placement of Scandentia, and whether it is sister to primates or to Glires, depends on the mode of analysis. This agreement between the binned MP-EST analysis and concatenated analysis of the reduced dataset may be an important finding, but contradicts [220] (which specifically addressed this question) and [221].

However, these two studies did not use coalescent-based methods to estimate species trees. The unbinned and binned MP-EST trees placed bats identically as sister to all other Laurasitheria (except for the basal Eulipotyphyla), and so both differed from the concatenation tree.

4.5.3 Metazoa

The Metazoan dataset also represents a challenging analysis, since the average bootstrap support is low (only 49%; see Fig. 4.19). On this dataset, we have only performed an unweighted statistical binning analysis. In this section, when we refer to binned trees, we are referring to an unbinned analysis with $\mathcal{S} = 75\%$ (since there are less than 1000 genes in this dataset).

The most important difference between the unbinned and unweighted binned MP-EST trees (Fig. 4.22) is among Chordates, where the unbinned MP-EST tree put Cephalochordates (represented by *B. floridae*) as sister to vertebrates (Craniates), and the binned MP-EST tree, (as in the concatenation analysis), put Urochordates (represented by *C. intestinalis*) as sister to vertebrates. While Cephalochordates were traditionally thought to be the sister to all the extant vertebrates [222], recent evidence supports Urochordates as the sister to all vertebrates [223–225], and hence the binned MP-EST tree is likely correct. There are also some differences between the two trees within Protostomia, but both MP-EST trees had low support for those relationships and neither was congruent with the literature.

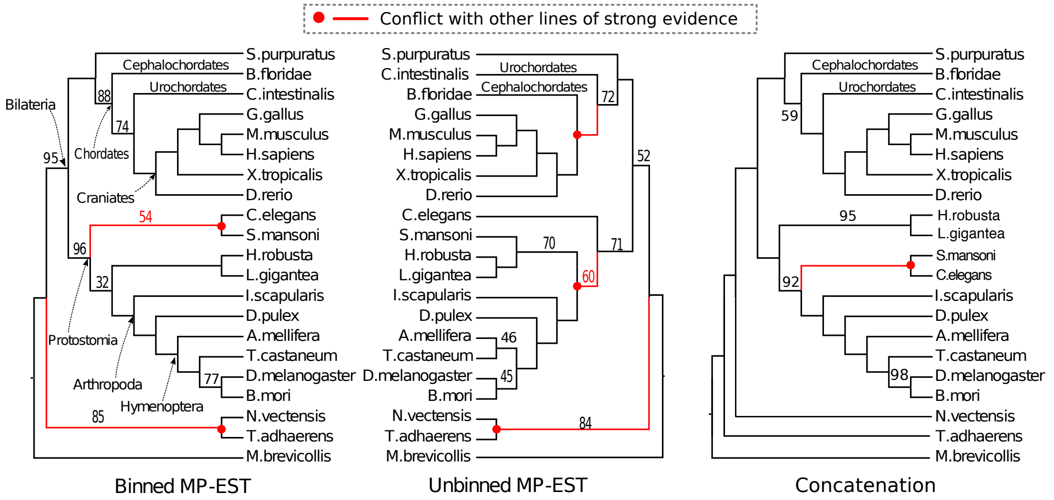


Figure 4.22: **Trees computed on the metazoan biological dataset using MP-EST**. We show results with unweighted binning (left), and unbinned analyses (right). We used 75% bootstrap support threshold for binning. Supergene trees were estimated using fully partitioned analyses. Branches without designation have 100% support. Branches in red indicate contradictions to other sources of evidence.

Sister to Bilateria: In both the binned and unbinned MP-EST trees, *N. vectensis* (representing Cnidaria) is grouped with *T. adhaerens* (representing Placozoa), and these two are sister to Bilateria. This relationship, which contradicts the monophyly of Eumetazoa, has some support in the literature [226], but the majority of recent molecular studies are congruent with the relationship recovered in the concatenation tree, where *N. vectensis* is sister to Bilateria [187, 227].

Protostomia: There are also some differences in the binned and unbinned MP-EST trees with respect to Protostomia, but these are hard to interpret because some relationships among major lineages of Protostomia are not well established [228]. Concatenation, binned and unbinned MP-EST analyses each results in a different resolution for Protostomia, and no topology is identical to some of the newer molecular studies [187, 228]. This is likely due to the poor taxon sampling of this dataset (only 20 metazoan taxa).

In all trees, Annelid (represented by *H. Robusta*) and *Mollusca* (represented by *L. gigantea*) are sisters with full support, as expected. However, Nematoda (represented by *C. elegans*), and Platyhelminthes (represented by *S. mansoni*) are put in different places. The likely correct relationship is that Nematoda should be sister to Arthropoda, and Platyhelminthes sister to Mollusca/Annelid [228]. The unbinned MP-EST analysis puts Platyhelminthes as sister to Mollusca/Annelid with 70% support, but fails to put Nematoda as sister to Arthropoda. Binned MP-EST recovers neither relationship, but is in fact essentially unresolved with regard to the relationship between Mollusca/Annelid, Platyhelminthes, and Arthropoda (only 32% support for an Arthropoda/Mollusca/Annelid clade, and 54% for Nematoda/Platyhelminthes). Concatenation puts Nematoda as sister to Platyhelminthes.

Among Arthropoda, binned and unbinned MP-EST trees differ in the position of Hymenoptera (represented by *A. mellifera*), where the binned MP-EST tree puts them as sister to other Holometabola, but the unbinned MP-EST tree puts them as sister to Coleoptera (represented by *T. castaneum*). While the exact position of Holometabola is debated, recent molecular analyses are consistent with the position in the binned MP-EST tree [229].

4.5.4 Vertebrates

The vertebrate dataset had the highest average BS (76%) of all datasets we examined (see Figs. 4.19). We performed only unweighted binning with $\mathcal{S} = 50\%$ (since this dataset has more than 1000 genes). Binned and unbinned MP-EST trees had the same topology, and both were topologically identical to the concatenation tree reported in [188]. The only difference between the two analyses is the bootstrap support for the clade containing horse (*E. caballus*) and dog (*C. familiaris*). The unbinned analysis has higher support (97%) for this clade, and the binned analysis has lower support (83%). All other branches have 100% support in both analyses. Whether horses (and more generally Perissodactyla) are closer to dogs (more generally Carnivora) or cows (more generally Cetartiodactyla) is an open question (see [230]).

4.5.5 Yeast

The yeast dataset has relatively high average BS (72%). We performed only unweighted binning with $\mathcal{S} = 50\%$ (since this dataset has more than 1000 genes). The binned and unbinned MP-EST topologies were identical, and both had 100% support for all but one branch, and both trees were also identical to the concatenation tree reported in [188] in all branches, except for the single branch that had less than 100% support. This particular branch unites *C. lusitaniae* with the *C. guiliermondii*/*D. hansenii* clade. While the exact position of *C. lusitaniae* is not known, the relationship recovered in the two MP-EST trees is closer to current belief about yeast evolution [231].

4.6 Discussion

Our simulation results demonstrate that both weighted and unweighted variants of binning reduces error in estimated species tree topologies, species tree branch lengths, gene tree topologies, and gene tree distributions under the conditions we studied. It also demonstrated that bootstrap support values in the species tree can be improved, in the sense that binned species trees tend to have higher support for true branches and lower support for false branches. These reductions in error result in estimations of ILS that are closer to correct ILS levels than unbinned MP-EST, which tends to over-estimate ILS levels. In our analyses, although unbinned methods are rarely more accurate than concatenation, binned MP-EST is almost always at least as accurate as concatenation, and there are many model conditions in which binned MP-EST is more accurate than concatenation while unbinned MP-EST is less accurate than concatenation. While empirical performances of weighted and unweighted binning were similar for the two datasets, theoretical guarantees of binning require weighting, and so we recommend using weighting on real biological datasets.

Below we discuss various aspects of the binning pipeline not discussed in detail before, and point out shortcoming of our study, in addition to directions for future research.

Imperfect binning: Binning can group genes together with different true topologies, despite its attempt to avoid such groupings. In such situations, binning can result in scenarios similar to concatenation analyses in the supergene tree estimation step. This could in principal lead to reduced accuracy for the estimated gene tree distributions. However, our simulations suggest that estimated gene tree distributions are more accurate after binning, under conditions we studied. We suggest that this is due to the fact that binning will never group genes with different topologies together unless the conflicting branches had low support, likely resulting from insufficient phylogenetic signal. As we have shown, the inclusion of poorly estimated gene trees distorts the estimated triplet gene tree distribution, and binning reduces this noise, suggesting that the overall impact of binning is beneficial. These results are also consistent with the observation that coalescent-based summary methods can be robust to recombination [232]. However, if levels of ILS are very high and bootstrap support in gene trees is lower than user-provided support threshold $\mathcal{S}$, we can get situations when binning hurts, as we saw on some model conditions of our 10-taxon dataset.

Variations of the binning algorithm: Our method can be seen as the logical extension of the "Naive Binning" technique: binning without attempting to evaluate whether genes have a common tree [17]. Unlike naive binning, we attempt to avoid putting genes together that have different true histories. In this dissertation, we explored only one algorithm for such a binning strategy, using bootstrap support values; however, alternative approaches can be imagined. For example, instead of using one support threshold, we could use a series of thresholds, and hierarchically divide bins into smaller subsets until some stopping criteria is met (e.g., no bin is larger than a certain size). Alternatively, we could use a measure of similarity between two gene trees and use clustering techniques for binning genes. Or, we could use more rigorous statistical tests for combinability (e.g., [233]) instead of bootstrap branch support values. Exploring these variations are topics of future research.

Bootstrap support $\mathcal{S}$: The results on the 10-taxon datasets stand out from the other analyses: statistical binning slightly increases species tree estimation error for some choice of $\mathcal{S}$. The difference in impact for statistical binning in this case is interesting, and points out the significance of how $\mathcal{S}$ is set. Even in cases where binning was universally helpful, the choice of the threshold $\mathcal{S}$ did impact the amount of improvement. We do not have a well-tested process for finding the best $\mathcal{S}$ threshold. The optimal choice likely depends on many factors, including the amount of ILS (thus with higher ILS one wants to be more conservative and pick lower thresholds), the number of genes (with more genes, one affords to be more conservative and pick lower thresholds), and the amount of gene tree error. One approach that can be used in practice is to simulate data under conditions similar to the biological dataset being analyzed, and to pick the threshold that performs the best in simulations.

Summary: In this chapter, we showed that gene tree estimation error impacts the accuracy of species tree estimation using summary methods, and described how the avian phylogenomics project suffered from lack of support in gene trees. We introduced the statistical binning pipeline to improve the quality of estimated gene trees, and described two variations of binning: weighted and unweighted. We showed that the weighted version has better theoretical guarantees than unweighted binning, but we did not observe any meaning differences between the two pipelines in our simulation studies. We showed in extensive simulation studies that under most conditions binning improves species tree accuracy. We showed that these improvements are largest for lower levels of ILS, lower levels of phylogenetic signal in genes, and for larger numbers of genes. We demonstrated the use of binning on several biological datasets, and were able to use binning on the avian phylogenomics project to produce the first coalescent-based highly supported avian tree of life.

Chapter 5

ASTRAL[1]

In the previous chapter, we described how a species tree can be estimated from a set of gene trees using either a traditional two step pipeline, or the statistical binning pipeline. Regardless of which pipeline is used, the final step requires a technique that produces a species tree given a collection of input (estimated) gene trees. Such a method is called a summary method, and a desirable attribute of the summary method is to be statistically consistent under the MSC model (see Section 2.2.2). Many statistically consistent methods have been developed through the years, and of these methods (e.g., MP-EST [134], which we used in the previous chapter) are now in widespread use. However, existing methods are too computationally intensive for use with

[1] Parts of this chapter have appeared in the following papers:

1. Siavash Mirarab, Rezwana Reaz, Md. Shamsuzzoha Bayzid, Théo Zimmermann, M Shel Swenson, and Tandy Warnow. ASTRAL: Genome-Scale Coalescent-Based Species Tree. *Bioinformatics*, 30(17):i541–i548, 2014
2. 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, 2015

In all three cases, SM and his supervisor, TM, designed the method, designed the studies, and wrote the papers (with comments from others), and SM implemented the methods. SM, MSB, and TZM ran experiments for (1) and SM ran all experiments for (2). MSS and RR worked on earlier versions of ASTRAL algorithmic ideas, and contributed to writing.

genome-scale analyses of large number of species or have poor accuracy under some realistic conditions, as we will show.

Some of these challenges were faced by the thousands plant transcriptomes (1KP) project [40]. The 1KP project has gathered sequences from across the genomes of a large number of plant species (103 plants in the initial phase and more than 1,100 in the ongoing second phase). The goal of the project was to estimate the species tree using various methods, including those that take gene tree incongruence due to incomplete lineage sorting into account. As we will show, these attempts had limited success, mostly due to limitations of existing summary methods.

In this chapter, we introduce a new summary method called ASTRAL (Accurate Species Tree Reconstruction ALgorithm). ASTRAL uses dynamic programming to solve a likely NP-hard optimization problem. ASTRAL can solve the optimization problem exactly in exponential time (doable only for up to 18 species), but more importantly, it can heuristically solve the problem in polynomial time by constraining the search space through a set of allowed bipartitions in the species tree (the constrained version of the problem is solved exactly). As we will show, ASTRAL is statistically consistent, even when run under the "constrained" mode. The constrained version can run on very large datasets, and has outstanding accuracy – improving upon various leading statistically consistent summary methods. ASTRAL is often more accurate than concatenation using maximum likelihood, except when ILS levels are low or there are too few gene trees.

We introduce two versions of ASTRAL: ASTRAL-I and ASTRAL-II. The second version is a direct improvement upon ASTRAL-I, with substantial advantages: ASTRAL-II is faster, can analyze much larger datasets (up to 1000 species and 1000 genes), and has substantially better accuracy under some conditions. ASTRAL-I's running time is $O(n^2k|\mathcal{X}|^2)$, and ASTRAL-II's running time is $O(nk|\mathcal{X}|^2)$, where n is the number of species, k is the number of loci, and $\mathcal{X}$ is the set of allowed bipartitions for the search space. ASTRAL is available in open source at `https://github.com/smirarab/ASTRAL/`.

In the rest of this chapter, we first motivate the development of a new summary method using simulation studies and some observations from the 1KP project. We then give the algorithmic details of ASTRAL-I and ASTRAL-II in Section 5.2 and discuss theoretical properties of both versions of ASTRAL. We then present a simulation study evaluating ASTRAL-I in Section 5.3 and a completely different simulation evaluating ASTRAL-II in Section 5.4. We then evaluate the use of ASTRAL on real biological data (Section 5.5) and finish by discussing results and pointing to directions for future research.

5.1 Motivation

Despite the availability of coalescent-based methods, many biological datasets are too large for the available methods. For example, MP-EST, easily scales to very large number of gene trees but cannot be used on datasets with large number of species due to computational reasons and degradation of accuracy (see [236], but we will show more results supporting this in our results section). BUCKy-pop [133], a method that tries to take into account gene tree uncertainty, is more computationally intensive and cannot run on datasets of moderate size. However, BUCKy tends to have very good accuracy where it can run, and can work with unrooted gene trees [148]. MP-EST has also been shown to have good accuracy under some conditions, but requires rooted gene trees [134]. A new distance-based method called NJst [146] can also handle unrooted gene trees, but NJst is new and its accuracy has not been tested extensively on various datasets.

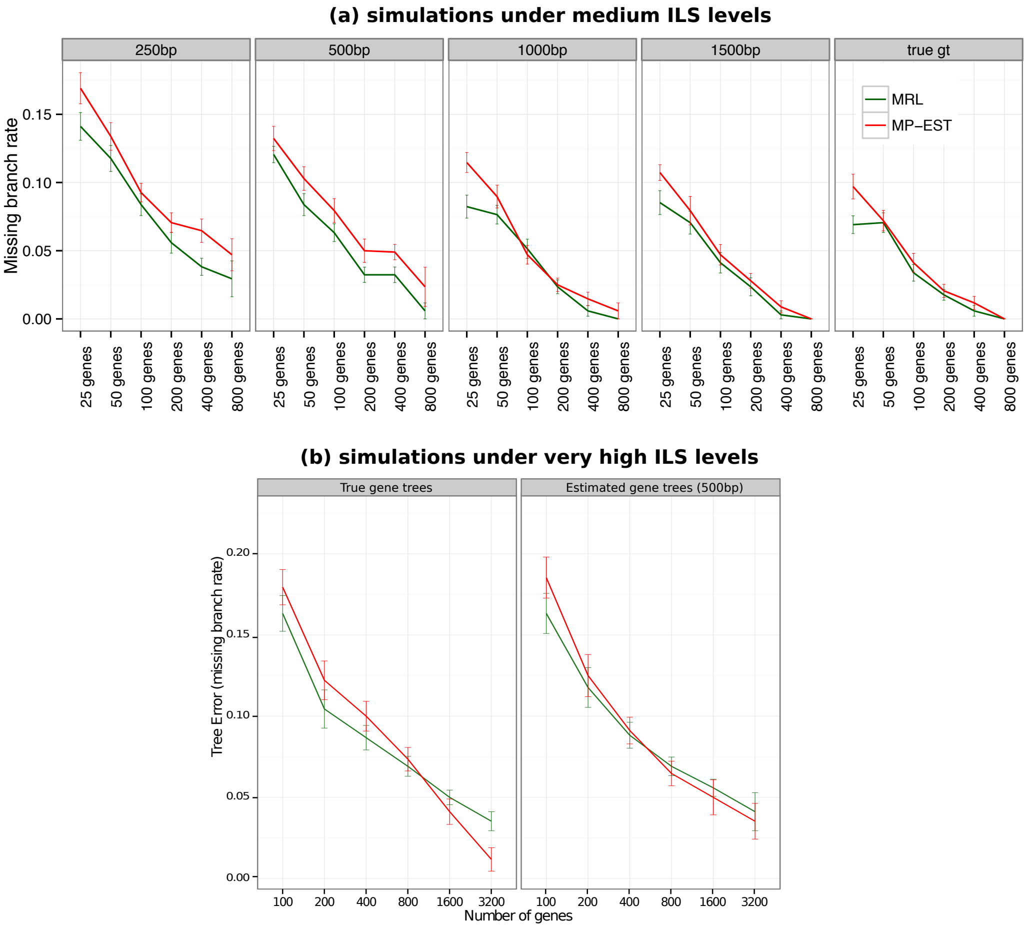
We were motivated to develop a new summary method by difficulties we were facing on a biological data analysis. The 1KP project [40] gathered sequence data across 103 plant species, with plans to go to more than 1,100 species in the next phase [2]. Our attempts to run MP-EST on this dataset had limited success. The pilot dataset that included 103 species was analyzed to extract 856 genes. We had difficulty in rooting many of these gene trees, since the common ancestor is believed to have existed close to a billion years ago, and our set of outgroups were missing from many of the genes. We built a restricted set of 669 gene trees that could be putatively rooted using outgroups. We attempted to analyze these 669 gene trees using MP-EST.

MP-EST took between 4 to 8 days to finish 5 random runs on each bootstrap replicate of this dataset. The results produced, however, were not

[2] see `http://www.onekp.com/samples/list.php` for the list of species

consistent among the 5 runs, and in some cases had log likelihoods scores that were many times larger than log likelihoods obtained from other runs (e.g., -69150891 in one run and -19540149 in a second run). These differences in log likelihood are not expected and show that the method is failing to search the tree space well in at least some of the random runs (this might be related to the fact that MP-EST uses only NNI moves). The species trees produced using MP-EST had low support, sometimes for easy-to-recover uncontroversial clades that we had recovered with 100% support using concatenation, and even simple statistically inconsistent summary methods such as MRP [237]. The shortcomings of MP-EST on the 1KP dataset could be the result of a combination of factors: rooting is challenging on this dataset, all gene trees are incomplete (are missing some species) and in some gene trees a large number of species are absent, and finally, the number of species being analyzed here is more than all the previous analyses that had tested or used MP-EST (typically below 50 species).

Beyond these challenges, it is possible that optimization scores other than pseudo-likelihood score optimized by MP-EST could simply correlate better with species tree accuracy. For example, a recent paper showed that a simple non-parametric quartet-based way of scoring species trees can predict species tree topological accuracy better than the pseudo-likelihood parametric score used by MP-EST [236]. Similarly, in a recent paper, we have shown that a simple statistically inconsistent method called MRL [238] outperforms MP-EST on large parts of the parameter space (see Fig. 5.1), suggesting that


(a) simulations under medium ILS levels

(b) simulations under very high ILS levels

Figure 5.1: **Comparison of MP-EST and MRL on a simulated mammalian dataset**. Species tree error is depicted for a simulated mammalian dataset reported in [38]. Simulation procedures are similar to those used in Chapter 4 and further described in Section 5.3.1.1. (a) We fix the level of ILS to medium and vary the number of genes and the gene alignment length, which controls gene tree estimation error. (b) We fix the level of ILS to very high, and vary the number of genes. We compare accuracy of MRL and MP-EST. On many conditions MRL has better accuracy; MP-EST, which has theoretical guarantees of statistical consistency, is better than MRL on these data only when levels of ILS are very high and very large number of genes are available.

better statistically consistent methods can be developed.

An accurate analysis of the 1KP dataset required a new method that could handle unrooted gene trees, could handle large number of species, and was robust to missing data. More generally, even the best coalescent-based summary methods have not been reliably more accurate than concatenation [17, 239], and analyses of biological datasets have in some cases resulted in species trees that were less well resolved and biologically feasible than concatenation [16, 195]. Hence, the choice between coalescent-based estimation and concatenation is highly controversial [33]. Improved accuracy and scalability for summary methods can help resolving this long-standing debate about the relative accuracy of concatenation and summary methods.

5.2 ASTRAL

Designing a statistically consistent summary method is complicated by the possibility that the most likely gene tree can be different from the species tree (the so-called anomaly zone [72], discussed in Section 2.2.2.2). However, it has been proved [20, 71, 240] that

Theorem 5.2.1. *There are no anomalous rooted 3-taxon species trees and no anomalous unrooted 4-taxon species trees.*

The complete proofs are given in [20] for rooted trees and [71, 240] for unrooted trees. Here, we provide a sketch for the rooted species tree on 3-taxa. Let's consider the case of the 3-taxon tree on human, chimp, and gorilla,

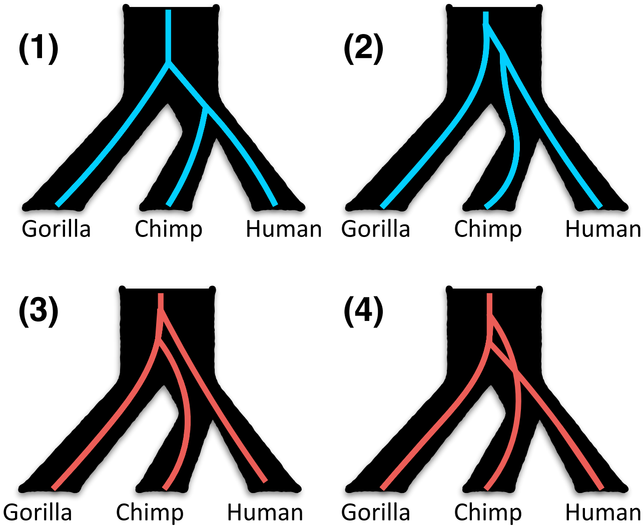


Figure 5.2: **Rooted gene trees and the species tree for 3 taxa**. Four coalescence scenarios can be imagine. (1) The two lineages from sister species chimp and human coalesce in their first ancestral population. The gene tree and the species tree will always be congruent under this scenario. (2-4) Lineages from chimp and human do not coalesce in the ancestral population and go further back into the common ancestor of all three populations. All three scenarios are equiprobable. Blue (1-2): concordance between species tree and gene tree. Red (3-4): discordance.

shown in Figure 5.2. There are three possible gene tree topologies (putting human with chimp or with gorilla, or putting chimp and gorilla together). The lineages from human and chimp have a *non-zero* probability p of coalescing in their most recent common ancestor (scenario 1); gene trees produced by this scenario will agree with the specie tree. If the two lineage fail to coalesce and go further back in time to the previous population, we have three lineages (human, chimp, gorilla) and the first coalescence event is equally likely to be between any pair of lineages (scenarios 2 – 4); thus, the three gene tree topologies are equiprobable in this case, and each topology has a probability of $\frac{1-p}{3}$. The probability of observing the species tree topology among the gene trees, therefore, is $p + \frac{1-p}{3} = \frac{1}{3} + \frac{2p}{3}$, which is strictly greater than $\frac{1}{3}$. Thus, the species tree topology has a higher probability than the two alternative trees. A similar argument can be made for 4-taxon unrooted species trees [20].

The fact that rooted 3-taxon and unrooted 4-taxon species trees do not have anomaly zones underlies the design of some summary methods and their proofs of statistical consistency. These methods decompose the gene trees into triplets or quartets of taxa (for the rooted or the unrooted case, respectively), find the species tree on the triplets or quartets, and then combine the triplet or quartet species trees. ASTRAL uses similar ideas in its design.

While some methods in the literature, such as MP-EST, use rooted triplets of taxa to speed up these analyses, we use unrooted quartet trees in ASTRAL. Rooting gene trees can be challenging, as it typically requires the use of an outgroup, but the given limited data in each gene, the position of the outgroup can be easily misconstructed [33]. For this reason, we believe that by using unrooted input gene trees, ASTRAL finds applicability for more datasets. As we will show, good running time can be achieved even with quartet trees, and ASTRAL has excellent accuracy.

We first start by giving some definitions and describing the notation. We next describe the first version of ASTRAL, and then describe how ASTRAL-II has improved upon ASTRAL-I.

5.2.1 Definitions and notations

We use the following notation throughout the rest of this chapter:

$\mathcal{S}$: a set of n species

$\mathcal{G} = \{t_1, \dots, t_k\}$: a set of k binary unrooted gene trees leaf-labelled by $\mathcal{S}$.

r: an arbitrary set of four species $\{a, b, c, d\} \subset \mathcal{S}$.

$\mathcal{Q}$: the set of all $\binom{n}{4}$ quartets of taxa selected from $\mathcal{S}$

q: an unrooted tree topology on quartet r. We use $ab|cd$ to indicate that a and b are sisters. Three topologies are possible: $ab|cd$, $ac|bd$, and $ad|bc$.

$t|r$: the quartet tree topology obtained by restricting tree t to the four species of r. When $q = t|r$, we say that t *agrees* or is compatible with q.

$Q(t)$: the set of quartet trees induced by tree t; thus, $Q(t) = \bigcup_{r \in \mathcal{Q}} \{t|r\}$

$w_{\mathcal{G}}(q)$: the number of trees in $\mathcal{G}$ that agree with q.

$\mathcal{X}$: a set of bipartitions (see Section 2.1.1) on leaf-set $\mathcal{S}$; all bipartitions in $\mathcal{X}$ are complete (include all taxa in $\mathcal{S}$). Each subset of $\mathcal{S}$ is called a *cluster*, and a bipartition defines two clusters. Since bipartitions in $\mathcal{X}$ are complete, we can represent $\mathcal{X}$ as a set of clusters instead of bipartitions, and when we do so, we refer to it as $\mathcal{X}'$.

For any quartet of taxa, the quartet tree topology that has higher $w_{\mathcal{G}}$ than the two alternative topologies is called the *dominant* topology (breaking ties arbitrarily).

5.2.2 ASTRAL-I

5.2.2.1 Optimization problem

Given a set $\mathcal{G}$ of k binary input gene trees on n taxa, there is a multi-set of $k\binom{n}{4}$ quartet trees induced by trees in $\mathcal{G}$. We define the Weighted Quartet (WQ) score of a tree t with respect to $\mathcal{G}$ to be the number of quartet trees from this multi-set that t also induces. Thus,

$$WQ_{\mathcal{G}}(t) = \sum_{1}^{k} |Q(t) \bigcap Q(t_i)| \tag{5.1}$$

An equivalent definition is

$$WQ_{\mathcal{G}}(t) = \sum_{r \in \mathcal{Q}} w_{\mathcal{G}}(t|r) = \sum_{q \in Q(t)} w_{\mathcal{G}}(q)$$

.

We now define an optimization problem for maximizing WQ.

Weighted Quartet Consensus (WQC) problem:

- Input: a set $\mathcal{G}$ of unrooted gene trees
- Output: the tree topology $\hat{T}$ on $\mathcal{S}$ that maximizes $WQ_{\mathcal{G}}$; i.e., return $\hat{T}$ such that $WQ_{\mathcal{G}}(\hat{T}) \geq WQ_{\mathcal{G}}(T')$ for $T' \neq \hat{T}$.

The WQC optimization problem, also called the quartet consensus [241] or Maximum Quartet Support Species Tree (MQSST) [234] problem, is a specific case of the general weighted quartet problem (where $w(q)$ is defined arbitrarily and not with respect to $\mathcal{G}$), which is an NP-hard [242] problem. The complexity of WQC has not been established. If the input trees are allowed to have missing data, then they could all include four leaves; in this case, WQC would be NP-hard [242]. When all the gene trees are restricted to be complete (i.e., contain all the species), the complexity of WQC is an open problem to our knowledge, but we suspect it is also NP-hard.

To be able to cope with the computational complexity of this likely NP-hard problem, we introduce a constrained version of WQC.

Constrained Weighted Quartet Consensus (CWQC) problem:

- Input: a set $\mathcal{G}$ of unrooted gene trees, and a set $\mathcal{X}$ of bipartitions on $\mathcal{S}$.
- Output: the tree topology $\hat{T}$ on species set $\mathcal{S}$ that maximizes $WQ_{\mathcal{G}}$ and all its bipartitions are in $\mathcal{X}$ (equivalently, all its clusters are in $\mathcal{X}'$).

CWQC is a generalization of WQC; setting $\mathcal{X}'$ in CWQC to the power set (set of all possible subsets) of $\mathcal{S}$ would solve WQC. As we show in Theorem 5.2.8, CWQC can be solved in time polynomial in the size of $\mathcal{X}'$, k, and n, and ASTRAL uses a dynamic programing algorithm to solve the problem. An exact solution to the constrained problem gives a heuristic solution to the unconstrained problem. Therefore, we refer to a solution to the constrained problem as the heuristic version of ASTRAL, and a solution to the unconstrained version as the exact version. Various settings of $\mathcal{X}$ would give different heuristics, and would each correspond to a specific constraint on the search space.

A natural way to define $\mathcal{X}$ is using the input gene trees and adding all their bipartitions to the set. The motivation for setting $\mathcal{X}$ in this manner is that we hope each bipartition in the species tree would appear in at least one of the gene trees. This definition of $\mathcal{X}$ is used by default in ASTRAL-I, but we allow the user to add extra bipartitions to this set if desired (in, ASTRAL-II, we expand this set automatically). Besides the intuitive reasons for setting $\mathcal{X}$ to bipartitions in the gene trees, this definition enables us to prove theoretical guarantees of statistical consistency.

Theorem 5.2.2. *An exact solution to CWQC problem is a statistically consistent estimator of the species tree topology under the MSC model when true gene trees are used as input, as long as $\mathcal{X}$ includes at least all bipartitions from all the input gene trees, but perhaps also more bipartitions.*

Proof. Let T be the true species tree. As stated in Theorem 5.2.1, unrooted quartet trees do not have anomaly zones [240]. Therefore, as the number of gene trees increases, with probability that approaches 1, each quartet topology induced by the species tree will appear more frequently in $\mathcal{G}$ than either of the two alternative topologies. Therefore, for every quartet of taxa r and every possible tree T', with probability that approaches 1 as we increase the number of genes, $w_{\mathcal{G}}(T|r) \geq w_{\mathcal{G}}(T'|r)$. By extension, if $\mathcal{Q}$ is the set of all possible quartets of taxa, then

$$\sum_{r \in \mathcal{Q}} w_{\mathcal{G}}(T|r) \geq \sum_{r \in \mathcal{Q}} w_{\mathcal{G}}(T'|r)$$

and thus:

$$WQ_{\mathcal{G}}(T) \geq WQ_{\mathcal{G}}(T')$$

Thus, the optimization criterion in WQC attains its maximum value with the true species tree with probability that approaches 1. The assumption of having a binary species tree ensures that the dominant quartet tree has a frequency that is strictly higher than two alternatives, and therefore, in the limit, the optimization problem has a *unique* maximum value (note that $Q(T_1) = Q(T_2)$ iff $T_1 = T_2$ [243]). Thus, an exact solution to the WQC problem is statistically consistent.

The species tree topology has a non-zero probability of being observed among gene trees. Therefore, as the number of gene trees increases, with probability converging to 1, at least one of the gene trees will be topologically identical to the species tree T. Therefore, in the limit, the set $\mathcal{X}$ will contain all the bipartitions from T with probability approaching 1. Thus, a solution to CWQC is also statistically consistent as long as $\mathcal{X}$ includes all bipartitions from all gene trees. Note also that $\mathcal{X}$ may contain all the bipartitions from T even without having T among its gene trees, but we invoked the probability of observing T in $\mathcal{X}$ for ease of proof. □

We note that CWQC takes into account the relative frequency of all three alternative quartet topologies for all quartets of taxa, and weights them accordingly. Thus, if the dominant quartet topology is much more frequent than the alternatives, trees that don't induce the dominant topology are penalized, but if the three alternative quartet topologies all have frequencies close to 1/3, that quartet will contribute little to the optimization problem. This approach is in contrast to some other quartet-based methods such as the population tree from BUCKy [133] that first try to find the dominant quartet topologies and then summarize them. Estimation of the dominant quartet tree is susceptible to error (due to insufficient gene sampling and estimation error) and the CWQC accounts for this.

The WQC optimization problem could be expressed as finding a *median tree*, where instead of finding a species tree that maximizes the total number of quartet trees that it satisfies, we would seek a fully binary species tree that has a minimum total distance to the input gene trees, where the distance is the number of gene tree quartet trees that it *violates*. Then, Theorem 5.2.2 asserts that the median tree (under this definition) is a statistically consistent estimator of the species tree.

5.2.2.2 Dynamic programming

ASTRAL uses a dynamic programming (DP) approach to solve the CWQC optimization problem. Moreover, the fact that weights of quartet trees are defined according to their frequency in the gene trees and not arbitrarily enables us to optimize the WQ score without explicitly enumerating the set of all possible quartet trees. Thus, we solve CWQC problem without ever explicitly calculating the $3\binom{n}{4}$ values of the $w_{\mathcal{G}}$ function.

For a given unrooted binary tree t and four leaves $r = \{a, b, c, d\}$ in the tree, the induced subtree of t connecting the four leaves will have exactly two nodes x and y with degree three (Fig. 5.3). We say that the quartet tree $q = ab|cd$ on four taxa r is associated (or mapped) to a pair of nodes $\{x, y\}$

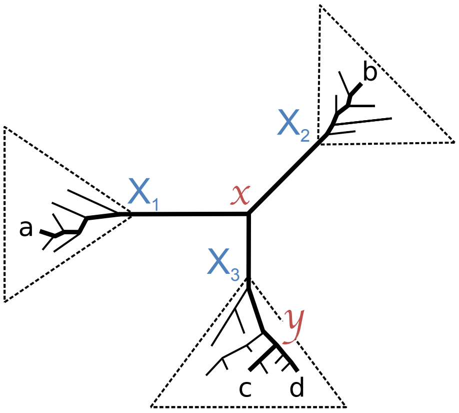


Figure 5.3: **Mapping a quartet tree to a tripartition**. Each node x in an unrooted tree defines a tripartition $(X_1|X_2|X_3)$ of the set of taxa and a tripartition defines a node. Each induced quartet tree $q = ab|cd$ maps to two nodes (x and y here). Node x is where the paths from a to c (or d) and b to c (or d) first join. Similarly, node y is where the paths from c to a and d to a first join.

in an unrooted binary tree t when q is compatible with t and x and y are the only two nodes that have a degree of three in $t|r$. We say that q is mapped to x from its ab side when a and b are on two different edges pending from x (similarly y is associated with the cd side of q).

Deleting x from a tree t separates it into three parts, X_1, X_2, and X_3, as shown in Figure 5.3; this is called a "tripartition", and is denoted $(X_1|X_2|X_3)$. Internal nodes of an unrooted tree and tripartitions are equivalent and we use them interchangeably. We call each part of a tripartition a "side" of the corresponding node.

For an internal node x, we can easily count the number of quartets that are associated with it. Recall that by definition, a quartet mapped to x

has two of its leaves pending from two different edges of x. Thus, to count the number of quartets mapped to x, we simply need to pick one of the three partitions of x (say X_1), and pick two leaves from it, and then pick one leaf from each of the remaining partitions, and do this for all ways of picking the first partition. Thus,

Corollary 5.2.3. *The number of quartet trees mapped to $x = (X_1|X_2|X_3)$, is*

$$F(x_1, x_2, x_3) = \binom{x_1}{2} x_2 x_3 + x_1 \binom{x_2}{2} x_3 + x_1 x_2 \binom{x_3}{2} = \frac{x_1 x_2 x_3 (x_1 + x_2 + x_3 - 3)}{2}$$

where x_1, x_2, and x_3 give the sizes of X_1, X_2, and X_3, respectively.

Recall that $q = ab|cd$ is mapped to x from the ab side when a and b belong to two different sides of x. Now, for two given tripartitions, x and y, we can derive how many quartets are mapped to both x and y *from the same side of the quartet.*

Lemma 5.2.4. *Let $x = (X_1|X_2|X_3)$ and $y = (Y_1|Y_2|Y_3)$ be two tripartitions on the same set of leaves $\mathcal{S}$. Let $\mathbf{C}$ be a 3×3 matrix with $\mathbf{C}_{ij} = |X_i \cap Y_j|$ for $i, j \in \{1, 2, 3\}$. The number of quartet trees mapped to both x and y from the same side of the quartet tree is:*

$$H(x, y) = H(\mathbf{C}) = \sum_{(a,b,c) \in G_3} F(\mathbf{C}_{1a}, \mathbf{C}_{2b}, \mathbf{C}_{3c}) \tag{5.2}$$

where G_3 gives the set of all permutations of $\{1, 2, 3\}$.

Proof. There are six bijections between the three parts of x and y. Take w.l.o.g. one of those bijections ($X_1 \to Y_1, X_2 \to Y_2, X_3 \to Y_3$). If we find the intersection between all three partitions paired with each other, we get a tripartition $z = (X_1 \cap Y_1, X_2 \cap Y_2, X_3 \cap Y_3)$ on a subset of $\mathcal{S}$. We can use the equation from Corollary 5.2.3 to count the number of quartet trees mapped to z. This is the term inside the sum in Equation 5.2 and note that we are summing over all possible bijections. The quartet trees mapped to z are clearly mapped also to both x and y. Moreover, any quartet tree mapped to z maps to x and y on its same exact side (the side that belonged to two sides of z). Furthermore, a quartet tree that maps to both x and y but from different sides won't be counted because z will not include it. To see this, consider $x = (a|b|cd)$ and $y = (ab|c|d)$; the quartet tree $ab|cd$ is mapped to both x and y, but is mapped from the ab side to x and from the cd side to y. All six ways of calculating z using bijections between partitions of x and y will have at least one empty part, and thus, H will be zero here. Therefore, H counts only quartets that are mapped to both x and y form their same side. We now need to show that all such quartet trees are counted exactly once.

Take any quartet tree $q = ab|cd$ that is mapped to both x and y w.l.o.g. from the ab side. By definition, a and b belong to two sides of x and w.l.o.g. let $a \in X_1$, $b \in X_2$, and $c, d \in X_3$ and similarly, w.l.o.g. let $a \in Y_1$, $b \in Y_2$, and $c, d \in Y_3$. The bijection that produces $z = (Z_1 = X_1 \cap Y_1, Z_2 = X_2 \cap Y_2, Z_3 = X_3 \cap Y_3)$ has $a \in Z_1$, $b \in Z_2$, and $cd \in Z_3$; therefore F applied to this bijection will count q. Tripartitions z produced by all five remaining bijections will miss one of the four taxa, and therefore will not count q. The lemma follows. □

We now count the number of quartet trees that a tripartition x shares with a collection of input trees. Let

$$s_{\mathcal{G}}(x) = \sum_{t \in \mathcal{G}} |Q(t) \cap Q(x)|$$

where $Q(x)$ is the set of quartet trees mapped to x. Then,

Lemma 5.2.5. *For a tripartition x and a set of unrooted binary trees $\mathcal{G}$,*

$$s_{\mathcal{G}}(x) = \sum_{t \in \mathcal{G}} \sum_{y \in \mathcal{N}(t)} H(x, y) \tag{5.3}$$

where $\mathcal{N}(t)$ is the set of internal nodes in t and $H(x, y)$ is given in Equation 5.2.

Proof. The proof follows from the fact that by Lemma 5.2.4, each $H(x, y)$ term counts all quartet trees that are mapped to x and y if and only if they are mapped from the same side. Each quartet tree q in a gene tree t that is mapped to x will therefore be counted, and will be counted only once: when y is the node in the gene tree that has q mapped to it, and has q mapped to it from the same side as x. □

We now present a major result.

Theorem 5.2.6. *The $WQ_{\mathcal{G}}$ score of a species tree $\hat{T}$ can be computed as*

$$WQ_{\mathcal{G}}(\hat{T}) = \frac{1}{2} \sum_{x \in \mathcal{N}(\hat{T})} s_{\mathcal{G}}(x) \tag{5.4}$$

Proof. Recall that $WQ_{\mathcal{G}}$ score defined in Equation 5.1 counts the number of quartet trees induced both by the species tree and the set of gene trees. Each quartet tree in the species tree maps to two of its internal nodes. Thus, if we simply count the number of quartet trees in all gene trees that are mapped to any internal nodes of $\hat{T}$ and sum up these values, we will count each quartet tree shared between the species tree and the gene trees exactly twice. The $s_{\mathcal{G}}(x)$ term, by Lemma 5.2.5, counts exactly this quantity for a given node. Thus, we just need to sum $s_{\mathcal{G}}(x)$ values for all the internal nodes of $\hat{T}$, and divide the sum by two. The theorem follows. □

The ability to score a tripartition of the species tree in isolation from other tripartitions using the $s_{\mathcal{G}}(x)$ function allows us to use dynamic programming to maximize the $WQ_{\mathcal{G}}$ score. The dynamic programming starts from the set $\mathcal{S}$ and recursively divides it into smaller subsets, each time finding the division that maximizes the $WQ_{\mathcal{G}}$ score. Backtracking defines the subtree that maximizes the score and at the top level returns the tree that maximizes $WQ_{\mathcal{G}}$.

Recall that $\mathcal{X}'$ is the set of clusters from bipartitions in $\mathcal{X}$ (i.e., $A \in \mathcal{X}'$ iff the bipartition $(A|\mathcal{S} - A) \in \mathcal{X}$). We compute $V(A)$, which gives the score for an optimal subtree on $A \subset \mathcal{S}$, using the following dynamic programming.

ASTRAL DP algorithm:

- $|A| = 1$: $V(A) = 0$
- $A = \mathcal{S}$: $V(A) = V(A - \{a\})$ for an arbitrary $a \in \mathcal{S}$
- otherwise:

$$V(A) = \max_{A', A-A' \in \mathcal{X}'} \{V(A') + V(A - A') + \frac{1}{2} s_{\mathcal{G}}((A'|A - A'|\mathcal{S} - A))\} \quad (5.5)$$

Note that $s_{\mathcal{G}}$ is defined in Equation 5.3 and $(A'|A - A'|\mathcal{S} - A)$ defines a tripartition, which can be scored using $s_{\mathcal{G}}$.

The recursion in the dynamic programming finds a way of dividing each set A into A' and $A - A'$ (each of which must be in $\mathcal{X}'$) such that the number of quartets satisfied by an optimal rooted tree on A' and $A - A'$, in addition to those satisfied by the tripartition $(A'|A - A'|\mathcal{S} - A)$, is maximized. The boundary cases are singleton clusters; for these, we set $V(A) = 0$. Also note that for $A = \mathcal{S}$, the tripartition $(A'|A - A'|\mathcal{S} - A)$ will have an empty set in its third part, regardless of the choice of A'; therefore $s_{\mathcal{G}}(A'|A - A'|\mathcal{S} - A)$ will be zero for $A = \mathcal{S}$. Since any trivial bipartitions (where one side has only one taxon) has to be in the final species tree, setting A' to any arbitrarily chosen leaf at the top level would work. Each division of A to two parts creates two new bipartitions in the species tree: $(A'|\mathcal{S} - A')$ and $(A - A'|\mathcal{S} - (A - A'))$; note that both of these bipartitions are restricted to those found in the set $\mathcal{X}$.

Theorem 5.2.7. *The ASTRAL DP algorithm finds an optimal solution to the CWQC optimization problem.*

Proof. Let tree $\hat{T}$ be the tree obtained by backtracking the sequence of set divisions in the DP algorithm. The $V(\mathcal{S})$ score computed by the DP algorithm equals the right hand side of Equation 5.4 and by Theorem 5.2.6, it equals $WQ_{\mathcal{G}}(\hat{T})$ (i.e. the optimization score of the tree). To see this, note that the recursive formula simply produces the sum of $s_{\mathcal{G}}$ scores for all the internal nodes of $\hat{T}$. We therefore need to only show the dynamic programming maximizes $V(\mathcal{S})$. For each A, the dynamic programming recursively finds the maximum possible V among all resolutions of A in addition to the score for the node resulting from that resolution; thus, by induction on A, the dynamic programming maximizes V. The theorem follows. □

5.2.2.3 Running time analysis

The score $s_{\mathcal{G}}(x)$ needs to be calculated for each tripartition of taxa visited in the dynamic programming. In ASTRAL-I, to compute $s_{\mathcal{G}}(x)$, we simply follow Equation 5.4. Thus, we sum over $O(nk)$ input gene tree nodes, and, for each node, we first calculate $\mathbf{C}$ and then compute $H(\mathbf{C})$ using Equation 5.2. We represent subsets of taxa as bitsets, which results in $O(n)$ running time for calculating $\mathbf{C}$; therefore, calculating each $s_{\mathcal{G}}(x)$ requires $O(n^2k)$ (we improve this in ASTRAL-II, as we will show). Note that our dynamic programming algorithm draws its clusters from the set $\mathcal{X}'$. Not all pairs of clusters in $\mathcal{X}$ can be put together, but for simplicity we assume they can; with this assumption, there are $O(|\mathcal{X}|^2)$ tripartitions that need to be scored. Thus,

Theorem 5.2.8. *ASTRAL-I runs in $O(n^2|\mathcal{X}|^2k)$ time, where n is the number of species and k is the number of gene trees.*

Note that this is a conservative running time analysis. The number of tripartitions scored is certainly lower than $|\mathcal{X}|^2$, and likely can be bounded with a lower exponent. Also, we do not need to calculate the score multiple times for the tripartitions that appear in multiple gene trees; we can compute the score once and simply multiply it by the number of times it appears. In practice, ASTRAL-I is really fast, as we will show.

We close by noting that our dynamic programming (DP) approach is similar to the algorithm used in [244] for constructing species trees from sets of gene trees, minimizing the total number of duplications and losses, and subsequently used to construct species trees minimizing deep coalescence [140]. We also note that Bryant and Steel give a dynamic programming for solving the general constrained weighted quartet problem (where weights are defined arbitrarily and not by the gene trees) [245]. Their dynamic programming also runs in polynomial time (with a n^4 term) and solves a constrained version of the problem where the bipartitions in the final tree are restricted to those coming from an input constraint set (analogous to $\mathcal{X}$). In our algorithm, we assume weights are the frequencies in the gene trees, and therefore, we can solve the problem without ever listing all $3\binom{n}{4}$ quartet topologies and their weights. Thus, we are able to achieve polynomial time running time with a lower exponent than n^4.

5.2.3 ASTRAL-II

We now describe how ASTRAL-II improves upon the older version. ASTRAL-II has three new features:

1. ASTRAL-II uses a faster algorithm to compute $s_{\mathcal{G}}(x)$.
2. ASTRAL-II searches a larger space by expanding the set $\mathcal{X}$ using heuristics.
3. ASTRAL-II can handle polytomies in its input gene trees.

Algorithm 5.1 - Weight calculation. Input is a gene tree set $\mathcal{G}$ and a tripartition $w = (X|Y|Z)$. Each part (e.g., X) is a bitset indexed by the species (thus, $X[i]$ is 1 if leaf i is in X and otherwise is 0). $H(\mathbf{C})$ is defined as in Eq. 5.2. Function WEIGHT computes $s_{\mathcal{G}}(x)$ defined in Eq. 5.3.

```
function WEIGHT($g, w = (X|Y|Z)$)
    for $t \in \mathcal{G}$ do
        $w \leftarrow 0$
        $S \leftarrow$ empty stack
        for $u \in postOrder(t)$ do
            if $u$ is a leaf then
                $(x, y, z) \leftarrow (X[u], Y[u], Z[u])$
            else
                $(\mathbf{C}_{11}, \mathbf{C}_{12}, \mathbf{C}_{13}) \leftarrow$ pull from $S$
                $(\mathbf{C}_{21}, \mathbf{C}_{22}, \mathbf{C}_{23}) \leftarrow$ pull from $S$
                $(x, y, z) \leftarrow (\mathbf{C}_{11} + \mathbf{C}_{21}, \mathbf{C}_{12} + \mathbf{C}_{22}, \mathbf{C}_{13} + \mathbf{C}_{23})$
                $(\mathbf{C}_{31}, \mathbf{C}_{32}, \mathbf{C}_{33}) \leftarrow (|X| - x, |Y| - y, |Z| - z)$
                $w \leftarrow w + H(\mathbf{C})$
            push $(x, y, z)$ to $S$
```

We motivate and discuss each feature in turn.

5.2.3.1 Running time improvement

Recall that ASTRAL-I computes $s_{\mathcal{G}}$ in $O(n^2k)$ time for each tripartition, by going over all $O(nk)$ input gene tree nodes, and, for each node, calculating H using Equation 5.2 in $O(n)$. In ASTRAL-II, instead of looking at all tripartitions in input gene trees, we use a post-order traverse of all gene trees (rooted arbitrarily) to calculate the score using Algorithm 5.1.

To score the input tripartition $w = (X|Y|Z)$, we traverse all the nodes of all gene trees. For each traversal node u, we compute a tuple (x, y, z), which gives the number of leaves under u that are shared with X, Y, and Z.

To do this for leaves, we simply need to find which side of w includes that leaf, which can be done in $O(1)$ if the tripartition is represented as three bitsets. For internal nodes, we can calculate (x, y, z) by simply summing up the same quantities already calculated for the two children of u, which also takes $O(1)$. The tuples from the two children of u in addition to $(|X| - x, |Y| - y, |Z| - z)$ give all the element of the 3×3 matrix $\mathbf{C}$ that gives the size of the intersection between all three sides of u and all three sides of w. Given $\mathbf{C}$, we simply need to calculate $H(\mathbf{C})$, which also takes $O(1)$. Thus, each inner-loop takes $O(1)$ and therefore, calculating $s_{\mathcal{G}}(w)$ for one tripartition requires $O(nk)$ running time. Thus,

Theorem 5.2.9. *ASTRAL-II runs in $O(nk|\mathcal{X}|^2)$ time, where n is the number of species, and k is the number of gene trees.*

5.2.3.2 Additions to $\mathcal{X}$

Theorem 5.2.2 established that the default way of setting the set $\mathcal{X}$ is statistically consistent. However, for a limited number of genes, as we will show in our results section, it is possible and sometimes likely that some of the bipartitions in the species tree do not appear in any of the gene trees. In ASTRAL-II, to account for this, we use a host of heuristic strategies to add extra bipartitions to the default set $\mathcal{X}$.

Similarity Matrix: For each pair of species a and b, we define

$$Q(\{a, b\}) = \{(ab|cd) : c, d \in \mathcal{S} - \{a, b\}\}$$

We now define a similarly measure between a pair of species:

$$s(a, b) = \sum_{t \in \mathcal{G}} |Q(\{a, b\}) \cap Q(t)|$$

Thus, the similarity between the two taxa is the number of quartet trees induced by gene trees where the pair appear on the same side of the quartet. This similarity matrix can be calculated using Algorithm 5.2. This algorithm traverses all nodes of all input gene trees (rooted arbitrarily), and for each node u, we look at all pairs of leaves chosen each from one of the children of u. For each such pair, we add $\binom{o}{2}$ to their similarity score, where o is the number of leaves *outside* the subtree below u. This will process each pair of nodes in each of the input k genes exactly once and would therefore require $O(n^2k)$ computations. The final score can be normalized by $|Q(\{a, b\})|$, the total number of quartet trees that include a and b on the same side. When input gene trees are complete, this normalization is not necessary and is not shown in Algorithm 5.2.

Once the similarity matrix is computed, we calculate an UPGMA tree and add all its bipartitions to the set $\mathcal{X}$. The UPGMA algorithm starts from

Algorithm 5.2 - Computing similarity matrix. *leafCount* gives the number of leaves under a node and is easily precomputed.

```
function getSimilarity($\mathcal{G}$)
    $S \leftarrow Zeros(n \times n)$
    for $g \in \mathcal{G}$ and $u \in postOrder(g)$ do
        for $l \in Left(u)$ do
            for $r \in Right(u)$ do
                $S[l, r] = s[r, l] = s[r, l] + \binom{n - leafCount(u)}{2}$
```

Algorithm 5.3 Additions to $\mathcal{X}$ using greedy consensus. See descriptions of functions in Table 5.1. Constants are by default set to $THS = \{0, \frac{1}{100}, \frac{1}{50}, \frac{1}{20}, \frac{1}{10}, \frac{1}{4}, \frac{1}{3}\}$; $ITERS = 10$; $RWD = 2$; and $FRQ = LTH = \frac{1}{100}$.

```
function addByGreedy($\mathcal{G}, S$)
    for $t \in THS$ do
        $gc \leftarrow greedy(\mathcal{G}, t, False)$
        for $p \in polytomies(gc)$ do
            $updateX(upgma(S, start = clusters(p)))$
            $c \leftarrow 0$
            $itercount \leftarrow ITERS$
            while $c < itercount$ do
                $c \leftarrow c + 1$
                $sample \leftarrow randSample(p)$
                $gr \leftarrow greedy(\mathcal{G}|sample, 0, True)$
                if $updateX(resolve(p, gr)) \geq FRQ$ then
                    $itercount \leftarrow itercount + RWD$
                $updateX(resolve(p, upgma(S|sample)))$
                if $t \leq LTH$ and $c < ITERS$ then
                    for $s \in sample$ do
                        $ld \leftarrow pectinate(sortBy(S, sample, s)$
                        $updateX(resolve(p, ld))$
```

n singleton clusters, one per taxa, and in each step, combines the two clusters with the highest similarity. The similarity of two clusters is the average similarity between all pairs of leaves chosen each from one of the two clusters.

Greedy: The greedy consensus of a set of trees is obtained by starting from a star tree and adding bipartitions from input trees in the decreasing order of their frequency if they don't conflict with previous bipartitions. This process ends when no remaining bipartition has frequency above a given threshold, or when the tree is fully resolved. We use greedy consensus of gene trees to

Table 5.1: **Functions used in Algorithm 5.3**.

Function	**Description**
$polytomies(t)$	For a given unrooted tree t, return all nodes with degree $d > 3$.
$greedy(\mathcal{G}, t, b)$	Finds bipartitions in all input trees in $\mathcal{G}$ and for each bipartitions notes its frequency. Sorts bipartitions by the descending order of frequency (with arbitrary tiebreakers) and discards those with frequency below t. Starts with a fully unresolved tree (i.e., the star tree), and adds bipartitions one at a time according to the order; if a bipartition conflicts with the tree, ignores it. At the end, if b is true, any remaining polytomies in the tree are randomly resolved. The branches (i.e., bipartitions) in the resulting tree are labelled by their bipartition frequency (i.e., their frequency in trees in $\mathcal{G}$).
$updateX(t)$	Adds all bipartition from t to the set $\mathcal{X}$ and notes which bipartitions are new. When edges in t have a frequency label (e.g., labels generated by the *greedy* function), *updateX* returns the maximum label of any *new* bipartition added to $\mathcal{X}$.
$clusters(p)$	An unrooted node p with degree d divides taxa into d subsets (Fig. 5.4). This function returns the partitions defined by p.
$upgma(S, C)$	Runs UPGMA using similarity matrix S on n taxa. By default, starts from n singleton clusters, one per taxa, and in each step, combines the two clusters with highest similarity. The similarity of two clusters is the average similarity between all pairs of leaves chosen each from one of the two clusters. When a set of clusters C is given, instead of starting with n singletons, starts by C.
$randSample(p)$	Selects a random leaf from each partition around node p.
$resolve(p, t)$	The input p is a node in an unrooted tree with leaf set L, and t is an unrooted tree on $L' \subset L$ such that L' contains exactly one leaf from each partition defined by p. Note that the tree t will be compatible with the tree that includes p. Every bipartition in t defines a further resolution of p. This function resolves p according to t and returns the results.
$pectinate(O)$	Given an ordered list of taxa O, it returns a pectinate tree based on O; e.g., $pectinate(a, d, e, c, b) = (a, (d, (e, (c, b))))$.
$sortBy(S, l, x)$	Sorts a list of taxa l based on their decreasing similarity to x and according to the similarity matrix S.

compute and add further bipartitions to $\mathcal{X}$, using Algorithm 5.3.

Algorithm 5.3 estimates the greedy consensus of the gene trees with various thresholds (THS). For each polytomy in each of these greedy consensus trees, it resolves the polytomy in multiple ways and adds bipartitions implied by those resolutions to the set $\mathcal{X}$ (if they don't already exist).

1. We resolve the polytomy by applying UPGMA to the similarity matrix; however, unlike the normal UPGMA algorithm that starts from singleton clusters, here, we start from clusters defined by each side of the polytomy.

2. We sample one leaf from each side of the polytomy randomly, and use the greedy consensus of the gene trees restricted to this subsample to find a resolution of the polytomy (randomly resolving remaining polytomies). We repeat this process at least 10 times, but if the subsampled greedy consensus trees include new bipartitions that are sufficiently frequent ($\geq 1\%$), we do more rounds of random sampling (we increase the number of iterations by two).

3. For each random subsample around a polytomy, we also resolve it by calculating an UPGMA tree on the similarity matrix restricted to the set of subsampled species.

4. For the two first greedy threshold values in THS and only for the first 10 random subsamples, we also use a third strategy that can potentially add a larger number of bipartitions: for each subsampled taxon a, we

resolve the polytomy as a pectinate tree (see Table 5.1) by sorting the remaining taxa according to their similarity with a (in decreasing order).

Gene tree polytomies: When gene trees include polytomies, we also add new bipartitions to $\mathcal{X}$. We first compute the greedy consensus of the input gene trees with threshold 0, and if the greedy consensus has polytomies, we resolve them using UPGMA; we repeat this process twice to account for random tie-breakers in the greedy consensus estimation. Then, for each gene tree polytomy, we use the two resolved greedy consensus trees to infer a resolution of the polytomy, and we add the implied bipartitions to $\mathcal{X}$.

Incomplete gene trees: The optimization problem used in ASTRAL can easily handle incomplete gene trees; i.e., gene trees where some of the leaves are not present. If $m < n$ quartets are present in a gene, it would contribute $\binom{m}{4}$ quartets to the WQ score defined in Equation 5.1. It is easy to show that if patterns of missing data are unbiased, the exact version of ASTRAL remains statistically consistent under gene trees that are incomplete. The challenging part of handling inputs with missing data is ensuring that the set $\mathcal{X}$ will include usable bipartitions.

When an input gene tree has missing data, at least one of its two parts (but possibly both parts) would not be in the complete gene tree, and therefore the inclusion of that part in $\mathcal{X}'$ is unlikely to be helpful (recall that $\mathcal{X}'$ is the set of all parts from all bipartitions in $\mathcal{X}$). When dealing with incomplete gene trees, we need to complete their bipartitions before adding them to $\mathcal{X}$. In ASTRAL-II, we use a heuristic approach to complete incomplete gene trees, and add bipartitions from the completed gene trees to $\mathcal{X}$. Note that this does not affect the scoring function, and only impacts the search space.

We use the similarity matrix computed in Algorithm 5.2 for adding missing taxa into incomplete trees. To ensure that the similarity matrix is not affected by arbitrary patterns of missing data in the gene trees, we need to also normalize the similarity values. As noted before, the normalization factor for each pair of leaves can simply be the number of quartets in all input gene trees that include the two taxa:

$$m(a,b) = \sum_{1}^{k} \binom{n_i - 2}{2} I_i(a,b)$$

where n_i is the number of leaves in gene tree g_i and $I_i(a,b) = 1$ if $\{a,b\} \subset g_i$ and otherwise $I_i(a,b) = 0$.

Given the similarity matrix, we add each missing taxon to each gene tree using an application of the four point condition [246]. When a distance matrix d is defined based on pairwise distances of leaves of a binary tree (i.e., with strictly positive branch lengths), for any quartet of taxa r, if the tree induces the quartet topology $q = ab|cd$, we have:

$$d(a,b) + d(c,d) < d(a,d) + d(b,c) = d(a,c) + d(b,d)$$

This inequality is called the four point condition.

We assume our similarity matrix (which can be converted to a distance matrix) uniquely defines a tree (i.e., is additive [247]). If all incomplete gene trees were identical topologically, our distance matrix would become additive as the number of genes increased. In the presence of discordance no such guarantees can be made, but we use this matrix anyway as a heuristic and note that our algorithm can be used with any similarity (or distance) matrix. We use Algorithm 5.4 to add missing leaves to the incomplete trees.

Algorithm 5.4 -Completing incomplete gene trees. Adds missing taxon m to tree t using similarity matrix S according to the four point condition. $arbLeaf(x)$ choses an arbitrary leaf under node x (by default, the left-most child). $addChild(x, y)$ adds y as a child of x.

```
function PLACE($t, S, m$)
    $closest \leftarrow \text{argmin}_{i \neq m} S[i, m]$
    $reroot(t, closest)$
    $u \leftarrow child(closest)$
    while $true$ do
        if $isLeaf(u)$ then
            $n \leftarrow Parent(u)$
            break
        $(l, r) \leftarrow (left(u), right(u))$
        $(lc, rc) \leftarrow (arbLeaf(lc), arbLeaf(rc))$
        $betterSide \leftarrow fourPoint(S, m, closest, lc, rc)$
        if $betterSide = closest$ then
            break
        else if $betterSide = lc$ then
            $u \leftarrow l$
        else if $betterSide = rc$ then
            $u \leftarrow r$
    $addAsChild(u, m)$
function FOURPOINT($S, m, a, b, c$)
    $as \leftarrow S[m, a] + S[c, b] - (S[m, c] + S[a, b])$
    $bs \leftarrow S[m, b] + S[a, c] - (S[m, a] + S[b, c])$
    $cs \leftarrow S[m, c] + S[b, a] - (S[m, b] + S[c, a])$
    $max \leftarrow \max(as, bs, cs)$
    return $c$ if $max = c$ else $b$ if $max = b$ else $a$
```

This heuristic algorithm first finds the taxon that has the highest similarity to the missing taxon m; it then roots the tree at this closest species c and traverses the nodes of the tree from root to the leaves. At each traversal point u, it decides whether it should move further down to the left (l) or the right (r) child of the current node u (we are assuming binary input genes, but extensions are straight forward), or if it should place the taxon at the branch above the current node. It arbitrarily chooses two leaves lc and rc under l and r (by default we choose the left most leaf). It places the taxon at the current branch iff m is closer to c than it is to either lc or rc according to the four point condition. If m is closer to one of the two arbitrarily chosen nodes, say lc, it choses that child of u, say l, as the next traversal nodes. Note that for each taxon x and any other three taxa, we can answer which of the three is closer to x by examining the four point conditions for all three possible topologies and finding which four point condition is closer to holding true (i.e., has a lower residual).

5.2.3.3 Multifurcating input gene trees

Although true gene trees are assumed to be binary, estimated gene trees can include polytomies. For example, some ML programs such as FastTree produce polytomies when several leaves have identical sequences. In maximum parsimony estimation of gene trees, if there are multiple trees with equal scores, a consensus of the trees is typically used, which can also result in polytomies. Most importantly, when bootstrapping (or other approach for obtaining branch support) are used, one can collapse low support branches in the gene trees, with the hope that impacts of gene tree estimation error are reduced [140, 149].

Extending ASTRAL to inputs that include polytomies requires solving the weighted quartet tree problem when each node of the input defines not a tripartition, but a multi-partition of the set of taxa. We start by a basic observation: every *resolved* quartet tree induced by a gene tree maps to two nodes in the gene tree *regardless* of whether the gene tree is binary or not (Fig. 5.4). In other words, induced quartet trees that map to only one node of the gene tree are *unresolved.*

When maximizing the quartet support, these unresolved gene tree quartet trees are inconsequential and need to be ignored. Now, consider a polytomy of degree d, which divides the set of taxa into d parts. There are $\binom{d}{3}$ ways to select three parts around the polytomy, and each of these defines a tripartition. Any selection of two taxa from one part of this tripartition and one taxon from each of the remaining two parts induces a resolved quartet tree, and each resolved quartet tree maps to exactly two nodes in our multifurcating tree. Thus, all the algorithmic assumptions of ASTRAL remain intact, as long as for each degree d node in an input gene tree, we treat it as a collection of $\binom{d}{3}$ tripartitions. Thus, to score a species tree tripartition $x = (X_1|X_2|X_3)$ with respect to a gene tree multi-partition $y = Y_1| \dots |Y_d$, we let $\mathbf{C}_{ij} = |X_i \cap Y_j|$ for all $i \in \{1, 2, 3\}$ and $j \in \{1, \dots, d\}$, and we generalize Equation 5.2 to:

$$H(x, y) = H(\mathbf{C}) = \sum_{(a,b,c) \in P_3} F(\mathbf{C}_{1a}, \mathbf{C}_{2b}, \mathbf{C}_{3c}) \tag{5.6}$$

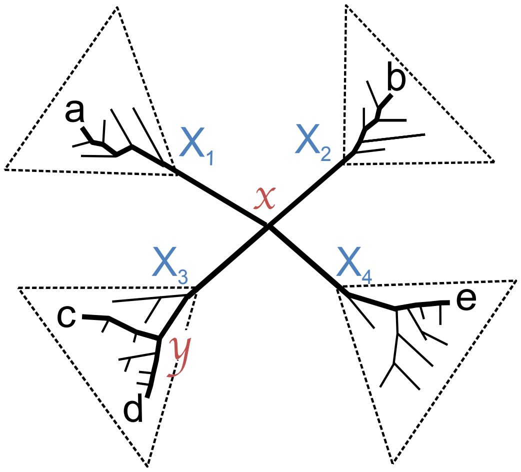


Figure 5.4: **Multipartitions in unrooted gene trees**. A polytomy divides the set of taxa into more than three parts (here, $d = 4$). A quartet tree mapped to two nodes (e.g., $ab|cd$) is a resolved quartet topology and needs to be counted towards WQ. A quartet tree mapped to only one node (e.g., $ab|ce$) is an unresolved quartet, and does not contribute to WQ; these need to be ignored. By treating the polytomy as a collection of $\binom{d}{3}$ tripartitions (in this case, $X_1|X_2|X_3$, $X_1|X_2|X_4$, $X_1|X_3|X_4$, and $X_2|X_3|X_4$), we ensure that all resolved quartet trees are counted and all unresolved quartet trees are left out. For example, here, $ab|ce$ would not be counted in our collection of $\binom{d}{3}$ tripartitions since each of its taxa are on a different part.

where P_3 is the set of all ordered subsets of size 3 from $\{1, \ldots, d\}$.

Extending Algorithm 5.1 to compute Equation 5.6 is straightforward. The leaves are treated the same. For internal nodes, instead of popping two values from the stack, $d - 1$ values are popped and are summed to calculate the tuple for the traversal node. All $\binom{d}{3}$ ways of choosing three subsets around that polytomy are then iterated over and H values are summed.

In the presence of polytomies, the running time analysis can change because analyzing each polytomy requires time cubic in its degree and the degree can increase with n. It is not hard to see that the worst case is when all gene trees have a polytomy with $d = \frac{n}{2}$ and each side of each polytomy has two leaves; in this case, Algorithm 5.1 would require $\binom{\frac{n}{2}}{3}$ calculations, which requires $O(n^3)$ running time; thus, the running time of ASTRAL-II is $O(n^3k|\mathcal{X}|^2)$ instead of $O(nk|\mathcal{X}|^2)$ in presence of polytomies.

5.3 Evaluation of ASTRAL-I on simulated data

5.3.1 Experimental setup

We evaluate ASTRAL-I on a collection of simulated datasets. Our simulation procedure is similar to what was used in Chapter 4. Simulated data are generated under the GTR+MSM model by first simulating gene trees down a species tree according to MSM and then simulating sequence data down each gene tree according to GTR. Gene trees are then estimated form the sequence data, and species trees are estimated from the gene trees using various summary methods. We also run concatenation under maximum likelihood (CA-ML) on the sequence data. The accuracy of the estimated species tree is evaluated against the model true species tree using the Robinson-Foulds (RF) [158] rate; because all species trees estimated here are completely bifurcating, this is the same as the missing branch rate (proportion of internal edges in the model tree missing in the estimated tree).

5.3.1.1 Datasets

100-taxon simulated datasets. These data were generated by Yang and Warnow [148]; we briefly describe the simulation process and direct the reader to the original publication [148] for details. The 100-taxon model species tree was created by a birth-death process, and 25 genes were evolved within the species tree under the MSC, producing ultrametric gene trees. Nucleotide sequences with 1000 sites were evolved down each gene tree under a process with GTR+Γ substitutions as well as insertions and deletions, using ROSE [175]. True alignments were used to generate estimated gene trees using RAxML.

37-taxon "mammalian" simulated datasets. We use the same mammalian simulated dataset used for evaluating statistical binning; Section 4.3.1.2 gives details of the simulation procedure, which we summarize here.

We simulated this collection of datasets based on a 37-taxon mammalian dataset with 447 genes studied in [77]. First, we used MP-EST to estimate a species tree on the biological dataset from [77], and used it as a model species tree, with branch lengths in coalescent units. We evolved gene trees down the model tree under the MSC model using Dendropy [204], and then rescaled the gene trees to deviate from the molecular clock and produce branch length patterns observed in the biological dataset. We then evolved sequences with 500 and 1000 sites down each gene tree under the GTR model of site evolution, using GTR parameters estimated on the biological dataset. This produces the "default" model condition that has the amount of ILS estimated for this dataset by MP-EST. We varied this protocol by scaling the model species tree branch lengths up (2X and 5X) or down (0.2X and 0.5X) to modify the amount of ILS; longer branch lengths reduces ILS, and shorter branch lengths increases ILS (note that in Chapter 4 we only multiplied or divided by two, but here we also multiply or divide by five). The default model tree conditions (including the number of genes, sequence length distribution, and amount of ILS) were set to produce a dataset called the "mixed condition" that most resembled the biological dataset.

The average bootstrap support (BS) in the biological data was 71%, and so we generated sequence lengths that produced estimated gene trees with BS values bracketing that value – 500bp alignments produced estimated gene trees with 63% average bootstrap support and 1000bp alignments produced estimated gene trees with 79% BS. The "mixed dataset" of 400 genes was produced using 200 genes with 63% BS and 200 genes with 79% BS, and had average BS of 71% - like the biological data.

We vary ILS levels, the number of genes, and sequence length. Unlike Chapter 4, where we went up to 800 genes, here we go up to 3,200 genes for the most challenging conditions with 0.2X branch lengths (thus, very high ILS). For each model condition (specified by the ILS level, the number of genes, and the sequence length), we created 20 replicates, except for the 1600- and 3200-gene model conditions where we created 10 and five replicates respectively. We used RAxML to estimate gene trees on the simulated sequence alignments, and we generated 200 ML bootstrap replicates for the mixed dataset.

5.3.1.2 Methods

We compare ASTRAL-I with MP-EST [134], BUCKy-pop (the population tree from BUCKy [133]), MRP (a supertree method [207]), the Greedy Consensus, and CA-ML computed by RAxML. Of these six methods, three are statistically consistent summary methods, two are inconsistent summary methods, and CA-ML is also inconsistent Note that BUCKy takes into account gene tree uncertainty and other methods don't [154].

For 100-taxon datasets and the mixed mammalian datasets, we ran summary methods using three different procedures: using maximum likelihood gene trees as input (bestML), using all bootstrap replicates of all genes as input (All BS), and using the site-only multi-locus bootstrapping (MLBS) procedure [208], described in Section 4.3.2. For MLBS, we used the greedy consensus of 200 replicate species trees, each computed on an input consisting of one bootstrap replicate tree per gene. BUCKy-pop takes as input distributions of gene trees, and its authors intended a Bayesian distribution to be the input; following results from Yang and Warnow [148], we approximate the distribution using bootstrap gene trees which are less computational intensive to generate and have resulted in the same accuracy as Bayesian trees in some analyses [148]; thus, BUCKy-pop is run with a procedure analogous to All BS. In subsequent analyses, where we study the impact of various model parameters, we only study the bestML approach. Exact commands and versions used are given in Appendix A.3.1.

5.3.2 Simulation results

5.3.2.1 Results on mammalian simulated datasets

We address the following three research questions on the mammalian simulated dataset, in three separate experiments.

RQ1: Given a choice of the gene tree input type (bestML, MLBS, or All BS), which of the six methods produces the best accuracy under the default mixed condition?

RQ2: How is relative performance of methods affected by the number of genes, levels of ILS, and gene tree error?

RQ3: How do summary methods compare under the highest levels of ILS if the number of genes is allowed to increase?

We now describe the results obtained for each question and finish by discussing the running time of ASTRAL-I in comparison to other methods.

RQ1: Figure 5.5 shows results on the mixed mammalian dataset, comparing all six methods and three types of inputs to summary methods (bestML, MLBS, and All BS). For MRP, MP-EST and ASTRAL-I, using bestML input trees produced more accurate species trees than using bootstrap replicates, either as one input (All BS) or using MLBS. The purpose of using bootstrap replicates is to take gene tree uncertainty (resulting from insufficient sequence length, for example) into account; the fact bestML gene trees had the best accuracy indicates that for this model condition, using bootstrapping does not alleviate the gene tree estimation problem. However, it is possible that other model conditions or other ways of addressing gene tree uncertainty might show some advantage over the bestML approach. For example, we have found in other studies that with few genes, the accuracy of the MLBS approach tends to be higher than the bestML approach, but as the number of genes increases, bestML becomes better [38]. Nevertheless, in this study we are not seeing any improvements form the use bootstrapped gene trees. Therefore, we use bestML input trees in the remaining experiments in this chapter (see [38] for more comparisons of using bestML or bootstrapped gene trees).

For the mixed model condition and using bestML trees, ASTRAL-I is the most accurate of these methods, MP-EST the next most accurate, followed by the other summary methods, and finally by CA-ML. ASTRAL-I with any of the three sets of inputs is also more accurate than BUCKy-pop; however, differences between ASTRAL-I on All BS and BUCKy-pop are relatively small.

RQ2: We now explore variants of the basic mammalian simulation, exploring the impact of changes to the number of genes, gene sequence length, and the ILS level (by scaling the species tree branch lengths) on the absolute and relative performance of various methods using bestML input. We first fix the ILS to the default 1X and vary both the number of genes and the sequence length. We then fix the number of genes to 200, and sequence length to 500bp, and vary the amount of ILS, in both cases also showing results on

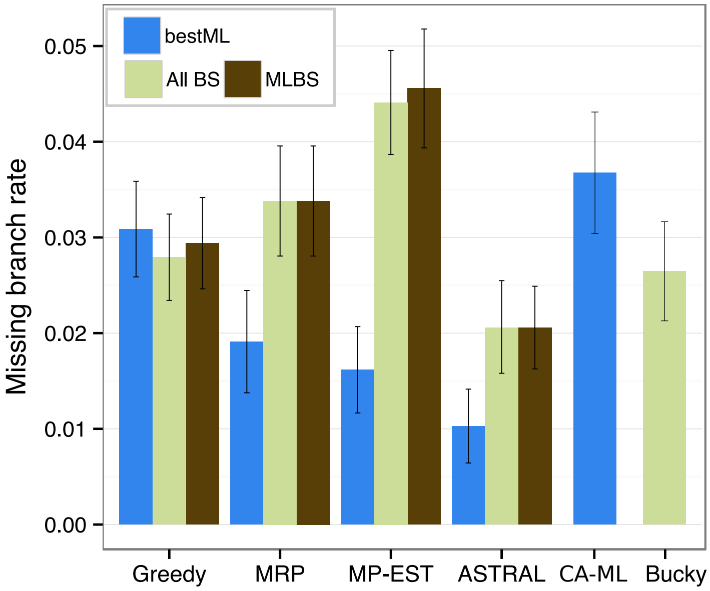


Figure 5.5: **Species tree estimation error on the default mixed mammalian datasets..** This dataset has 200 genes with 500bp and 200 genes with 1000bp, which results in 71% mean BS. We show the missing branch rates for estimated species trees computed using summary methods (MRP, MP-EST, greedy, BUCKy-pop, and ASTRAL-I) as well as concatenation using RAxML. Results are shown for running summary methods on maximum likelihood gene trees (bestML) and on the set of all bootstrap replicates from all genes (All BS), as well as the greedy consensus of running summary methods on individual bootstrap replicates from all genes (MLBS). CA-ML is run on the true alignment. Average and standard error shown based on 20 replicates.

true (simulated) gene trees. Figure 5.6 shows results for this experiment for all these model conditions. General trends as we changed parameters were as expected: all summary methods gave improved accuracy as the sequence length in each gene increased from 500bp to 1000bp; using true gene trees gave the best results; species tree error rates generally reduced as the number of genes increased; and species tree error rates increased as ILS levels increased.

ASTRAL-I was commonly more accurate than all the other summary methods we studied. ASTRAL-I was never outperformed by other summary methods; however, for a few cases, ASTRAL-I and one or more summary methods had identical accuracy. For example, on 800 true gene trees from default ILS levels, all summary methods (except for Greedy) produced the true species tree. We performed an ANOVA test comparing the species tree accuracy differences between ASTRAL and MP-EST, with the amount of ILS, number of genes, and the sequence length as independent variables. ASTRAL was significantly better than MP-EST ($p < 10^{-5}$) and the relative accuracy of ASTRAL and MP-EST depended only on the amount of ILS ($p = 0.008$), but not the number of genes ($p = 0.8$) or gene sequence length ($p = 0.3$).

Comparison of ASTRAL-I and CA-ML was interesting. ASTRAL-I was more accurate than CA-ML in general ($p < 10^{-5}$ according to an ANOVA test); however, the relative performance depended significantly on the level of ILS ($p < 10^{-5}$). With reduced ILS, CA-ML had better accuracy than all summary methods, including ASTRAL, but as the level of ILS increased, ASTRAL became more accurate.

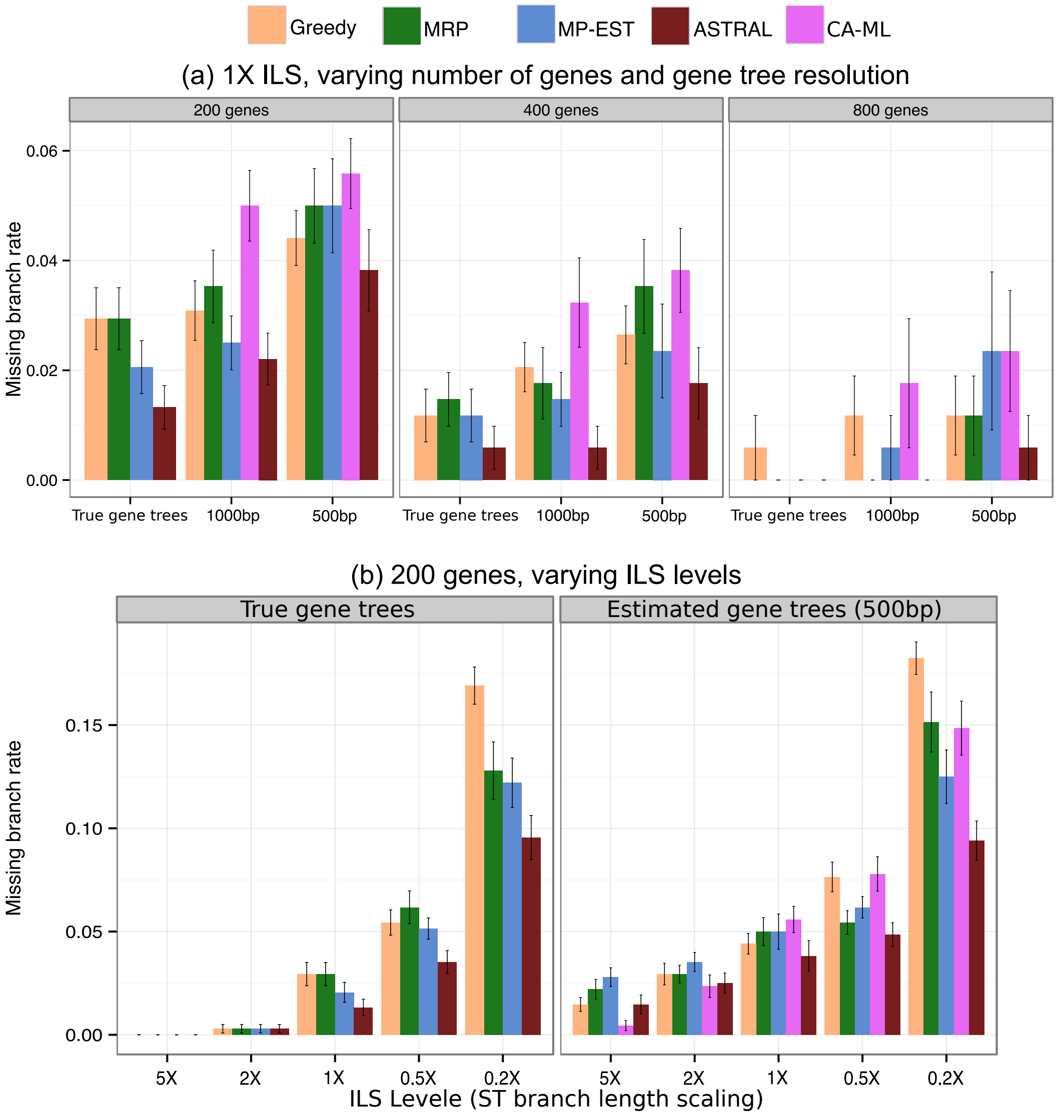


Figure 5.6: **Species tree estimation error on the simulated mammalian datasets, varying simulation parameters**. We show the missing branch rates for estimated species trees computed using summary methods (MRP, MP-EST, greedy, and ASTRAL-I) as well as CA-ML. Summary methods are run on RAxML bestML gene trees and true gene trees, and CA-ML is run using RAxML. (a) Default levels of ILS, varying the number of genes and gene tree resolution; (b) 200 genes, varying the amount of ILS from very low (5X species tree branch lengths) to very high (0.2X species tree branch lengths).

RQ3: For the most challenging ILS level, where with 200 genes the error was still high for all methods including ASTRAL-I, we asked whether increasing the number of genes reduces the error, as expected by the statistical consistency of ASTRAL-I. Figure 5.7 shows results for the case where fix the ILS level to 0.2X (very high) and increase the number of genes up to 3,200. As we increase the number of genes, the error reduces for all summary methods, except for the greedy consensus. With 3,200 gene trees, ASTRAL-I has 0.5% error, with true gene trees, and only 1.5% error with estimated trees. Thus, even with the most challenging ILS scenarios, with increased number of genes, high accuracy can be obtained. MP-EST also has reduced error with increased number of genes, but is always less accurate than ASTRAL-I. For example, the error of MP-EST with 1600 true gene trees is 4.1%, which is exactly the same as the error of ASTRAL-I with 800 genes, but with 1,600 true gene trees, ASTRAL-I has 2.0% error.

Running time. We examine running times under moderate ILS, gene sequences of length 500bp, and with 400 and 800 genes and with bestML input trees (except for BUCKy-pop). BUCKy-pop strictly runs in serial, using a Bayesian MCMC technique, which can take a long time and substantial memory to reach convergence. On the 37-taxon mammalian simulated datasets, BUCKy-pop ran to completion for datasets with up to 400 genes (where it took approximately 5 hours), but failed to complete (due to memory issues) on the 800-gene dataset.

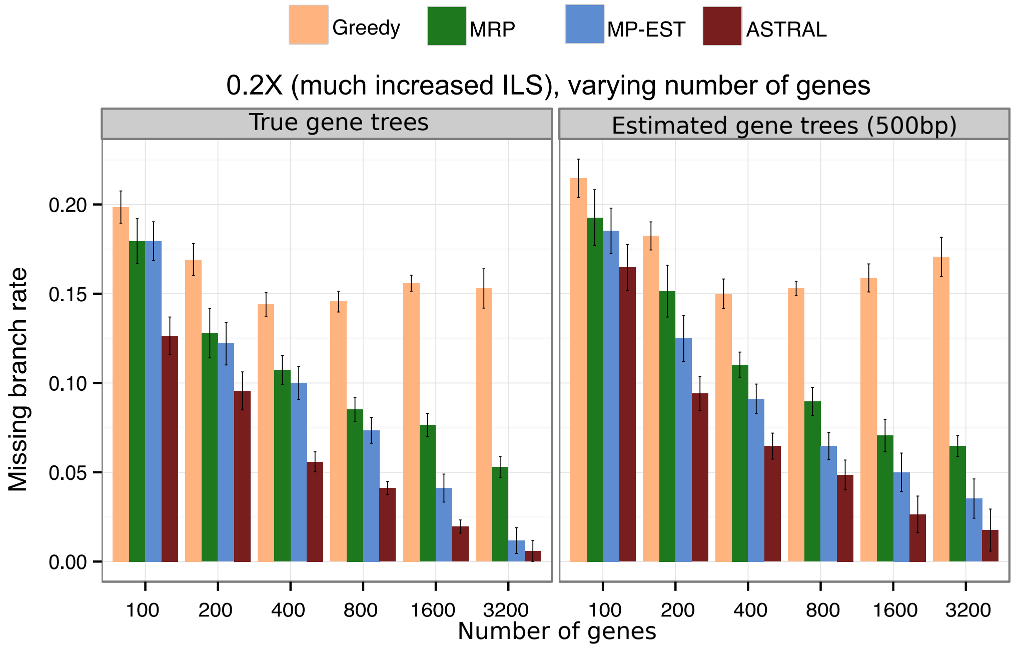


Figure 5.7: **Species tree estimation error on the simulated mammalian datasets with highest level of ILS**. We show the missing branch rates for estimated species trees computed using summary methods (MRP, MP-EST, greedy, and ASTRAL-I) run on RAxML bestML gene trees and true gene trees. ILS levels are fixed to 0.2X (very high) and the number of genes is increased to 3200.

MP-EST completed relatively quickly - about 100 minutes - for both the 400-gene and 800-gene datasets. We ran MP-EST with 10 random starting points, so this time could be reduced by using just one starting point, but with a potential decrease in accuracy.

ASTRAL-I completed in 3.3 seconds on the 400-gene dataset, and in 5.3 seconds on the 800-gene dataset. Thus, ASTRAL-I is dramatically faster than the other methods, and able to run on these moderately large datasets in extremely short time frames. However, BUCKy is used with 200 bootstrapped gene trees for each gene, and outputs support values. Running ASTRAL-I and MP-EST using MLBS to obtains support values would increase their running times if run in serial, but ASTRAL-I would still be much faster than BUCKy (e.g., 11 minutes on the 400-gene dataset rather than 5 hours). In addition, parallelizing MLBS is trivial since each bootstrap replicate is independent.

Finally, Figure 5.8 shows how the running time of ASTRAL-I is impacted by the number of genes and the level of ILS. The running time of ASTRAL-I increases as the level of ILS is increased, because the set $\mathcal{X}$ is populated with more bipartitions when gene trees have high levels of ILS. As the number of genes are increased, the number of unique bipartitions in input gene trees increases, which increases the time required to calculate the score function w, and also the size of the set $\mathcal{X}$ is likely to increase. Thus, both factors impact the running time, but even under the most challenging conditions (3200 genes of 0.2X ILS level), ASTRAL-I finished in about two hours on the mammalian dataset.

5.3.2.2 100-taxon dataset

We evaluated the feasibility of using ASTRAL-I on datasets with large numbers of taxa using the 100-taxon simulated datasets, with 25 genes and 10 replicates. Because there is no single outgroup, the estimated trees are not rooted, and so we could not use MP-EST. ASTRAL-I had no difficulty analyzing these data (completing in under one second). ASTRAL-I had average

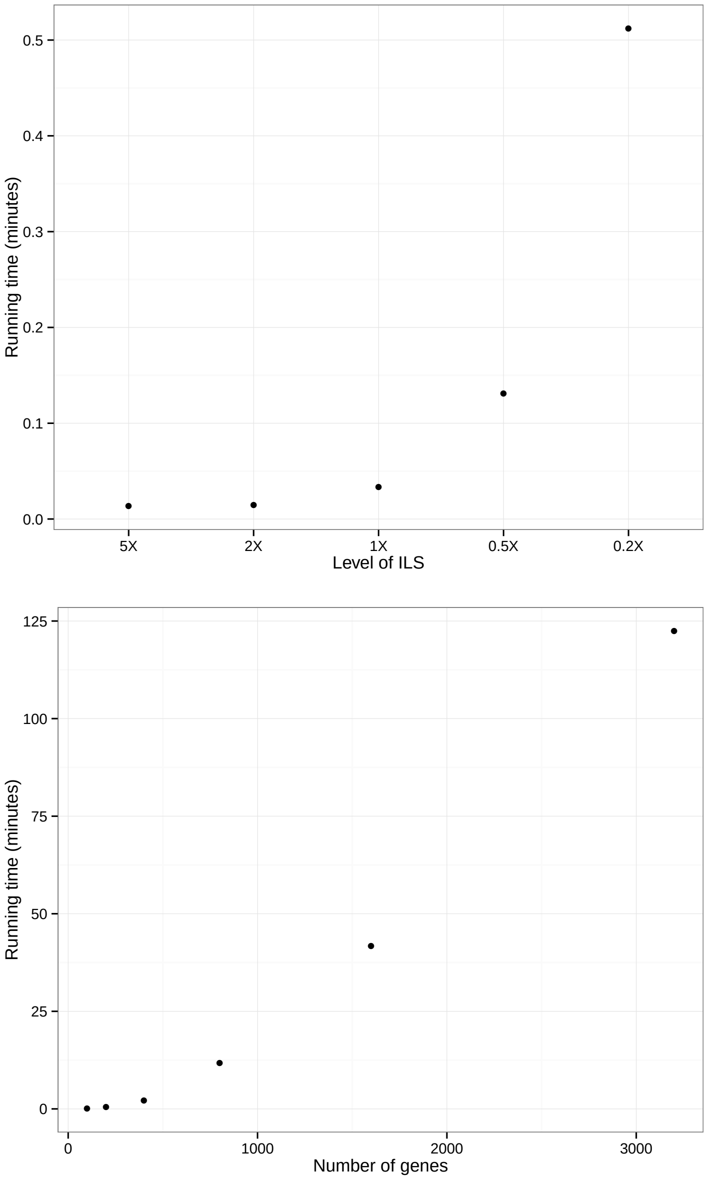


Figure 5.8: **Running time of ASTRAL**. We show the running time for default ASTRAL on the mammals simulated datasets with (top) varying levels of ILS with 200 genes of 500bp resolution, and (bottom) varying number of true gene trees with much increased ILS level (0.2X).

Table 5.2: **Results on 100-taxon dataset**. Average FN rates (over 20 replicates) of different methods on the 100-taxon 25-gene simulated datasets. The dataset does not have an outgroup, and therefore, we could not run MP-EST on it. Gene trees and CA-ML are estimated using RAxML.

Method	bestML	All BS
CA-ML	0.057	
ASTRAL	0.061	**0.052**
Greedy	0.064	0.056
MRP	0.064	0.055

missing branch rate of 6.1%, better than MRP and Greedy (6.4%), but not as good as CA-ML (5.7%); differences are not statistically significant ($p > 0.1$; paired Wilcoxon test).

5.3.3 Summary of results

In our study, ASTRAL-I was more accurate than MP-EST and BUCKy-pop, two leading coalescent-based methods, and improved or matched the accuracy of concatenation under maximum likelihood under many conditions, except when the amount of ILS was very low, where concatenation was more accurate. This study also showed that concatenation could be more accurate than coalescent-based estimation, provided that the amount of ILS is low enough. However, the best coalescent-based methods can be more accurate than concatenation under biologically realistic conditions.

Using bootstrap replicate gene trees instead of best ML gene trees did not improve species tree estimation accuracy on the simulated mixed mammalian dataset – and in fact made species tree estimations less accurate for MRP, MP-EST and ASTRAL-I. Similar results have been observed by others when taking gene tree estimation error into account [36]. This suggests the possibility that the topological error in bootstrap gene trees is large enough to offset any improvement in species tree estimation obtained by taking gene tree uncertainty into account. However, it is possible that an improvement might be obtained under other conditions, or that using a sample of gene trees estimated by a Bayesian MCMC analysis might be better suited to coalescent-based species tree estimation methods than maximum likelihood bootstrap trees, as suggested by [239] (although see [148]).

5.4 Evaluation of ASTRAL-II on simulated data

Our experiments on ASTRAL-I were all using relatively small datasets; we had either few species and large numbers of genes, or moderately large numbers of species and few gene trees. Here, we report the result of a more extensive simulation study that shows under certain conditions ASTRAL-I can have reduced accuracy because of the restrictions imposed by the default setting of the set $\mathcal{X}$. We show that ASTRAL-II addresses these problems, and we demonstrate that ASTRAL-II can run on datasets with up to 1000 genes and 1000 species in about a day.

5.4.1 Experimental setup

5.4.1.1 Dataset

We used SimPhy [248] to simulate species trees and gene trees under MSC and to generate gene trees in mutation units, and then used Indelible [176] to simulate nucleotide sequences down the gene trees according to GTR with varying length and model parameters. We estimated gene trees on these simulated gene alignments, which we then used as input to ASTRAL-I, ASTRAL-II, NJst [146], and MP-EST, in addition to concatenation.

We used SimPhy to simulate species trees according to the Yule process, characterized by the number of taxa, maximum tree length, and the speciation rate (this combination defines a model condition). We simulated 11 model conditions, which we divide into two datasets, with one model condition appearing in both datasets.

Dataset I: In 6 model conditions (forming Dataset I), we fixed the number of taxa to 200 and varied tree length (500K, 2M, and 10M generations), and speciation rates (1e-6, and 1e-7 per generation). The tree length impacts the amount of ILS, with lower length resulting in shorter branches, and therefore higher levels of ILS (Fig. 5.9). Speciation rate impacts whether speciation events tend to happen close to the tips (1e-06) or close to the base (1e-07). Different tree shapes (i.e., combinations of tree length and speciation rate) produce different levels of ILS starting from relatively low and going up to very high. The 10M/1e-06 condition had 0% to 20% distance between true gene trees and the species tree, measured by the RF distance, whereas 500K length (with 1e-06 or 1e-07 rate) had between 60% and 80% RF distance (Fig. 5.9). Thus, the 500K length has the highest ILS levels and 10M has the lowest, and 2M is in between.

Dataset II In six model conditions (forming Dataset II), we fixed the tree shape to 2M/1e-06 (medium ILS levels) and set the number of taxa to 10, 50, 100, 200, 500, and 1000. The amount of ILS only slightly increased as we increased the number of species (Fig. 5.9). Note that the model condition with 200 taxa and the 2M/1e-6 tree shape appears in both datasets.

For each model condition, we simulated 50 species trees, forming 50 replicates. On each species tree, 1000 gene trees were simulated according to the MSC model with the population size fixed to 200,000 (a reasonable value for vertebrates). SimPhy uses various rate parameters and rate heterogeneity modifiers to convert gene tree branch lengths to mutation units, introducing deviations from molecular clock and rate heterogeneity between genes. Parameters for these simulations are given in Appendix A.4.1.

We simulated indel-free gene alignments using Indelible [176] under the GTR+Γ model. First, for each replicate, two parameters, μ and σ, were drawn uniformly from $(5.7, 7.3)$ and $(0, 0.3)$ respectively. Then, the sequence length for each gene in that replicate was drawn from a log-normal distribution with μ and σ parameters (thus, average sequence length is uniformly distributed between 300bp and 1500bp). GTR+Γ parameters were drawn from a Dirich-

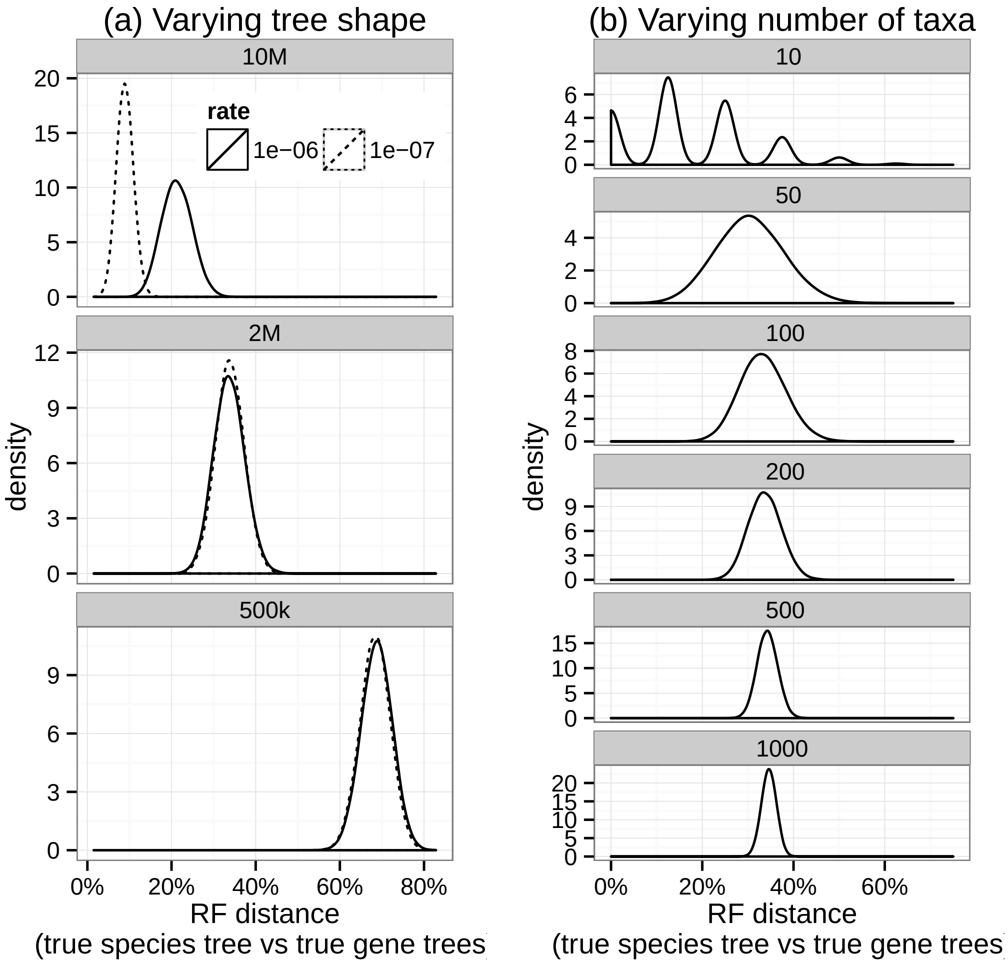


Figure 5.9: **ILS levels in ASTRAL-II simulation data**. RF distance between the true species tree and the true gene trees (50 replicates of 1000 genes) for (a) Dataset I and (b) Dataset II. Tree height directly affects the amount of true discordance; the speciation rate affects true gene tree discordance only with 10M tree length. The number of taxa has a modest effect on the amount of ILS.

let(36,26,28,32) distribution; we estimated the Dirichlet parameters from a collection of biological datasets using ML (see Appendix A.4.2 for details).

5.4.1.2 Methods

Gene tree estimation: Previous studies [123] have shown that FastTree-II [119] is generally as accurate at estimating the tree topology as more extensive ML heuristics such as RAxML [249], while being much faster. In our simulation studies, we used FastTree to estimate the 550,000 gene trees ranging from 10 to 1000 species. Our estimated gene trees had wide-ranging levels of gene tree estimation error (see Figure 5.10). The tree error was impacted by tree shape parameters; as expected, more ILS and deeper speciation lead to higher levels of gene tree error. Moreover, average gene tree estimation error varied across replicates, and gene tree error varied considerably among the 1000 genes in each replicate (Fig. 5.10). The number of taxa had only a small impact on gene tree estimation error.

FastTree outputs polytomies when sequence alignments cannot distinguish between competing tree resolutions. We removed any gene tree where more than 50% of the internal nodes were polytomies because they would not add much new information but would increase the running time of ASTRAL (and would be randomly resolved for other methods). This pruning left fewer than 500 genes for 9 out of 550 replicates in some model conditions: 200-taxon/500K/1e-06 (3 replicates), 50-taxon (3 replicates), 100-taxon (2 replicates), and 10-taxon (1 replicate). We removed these 9 replicates.

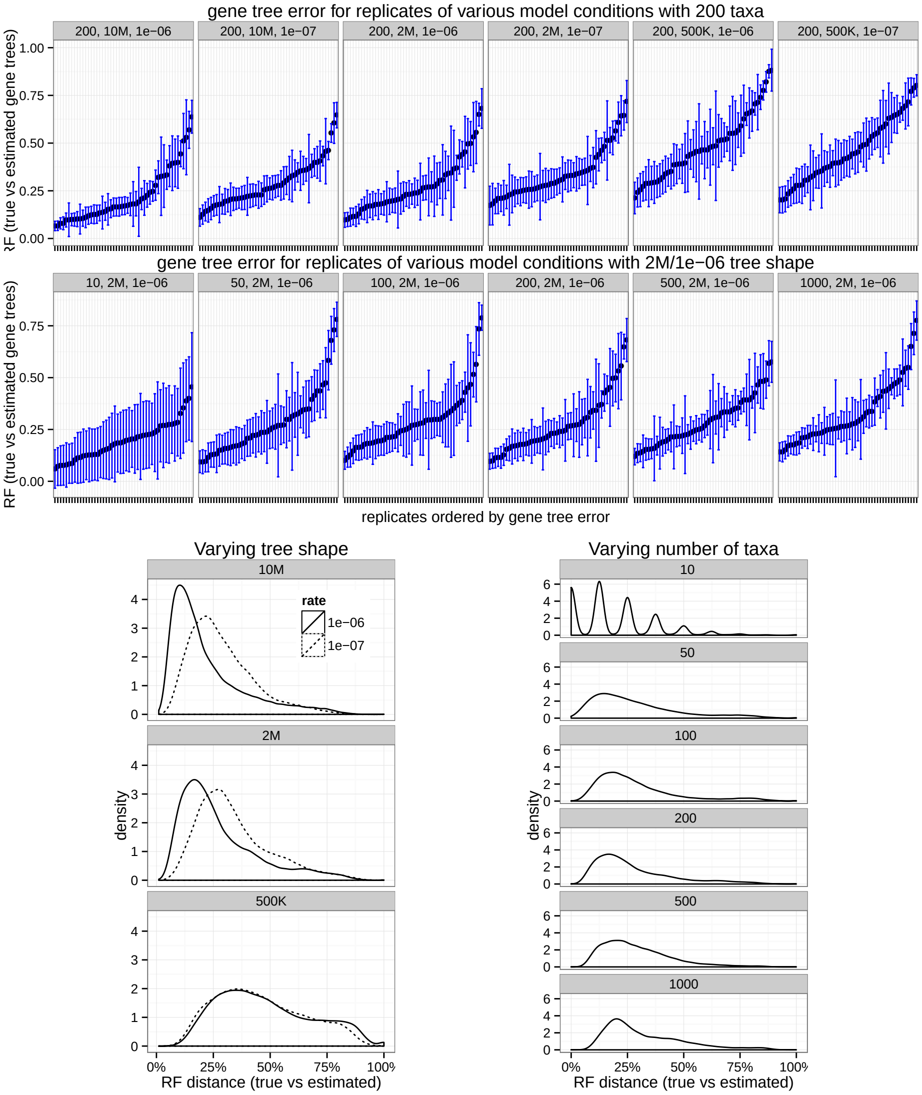


Figure 5.10: **Gene tree estimation error in simulated ASTRAL-II datasets**. Many parameters (e.g. alignment length, gene tree length, and substitution rates) were varied in a heterogeneous way to simulate 50 replicates per model condition with varying gene tree estimation error. Top: each box (box title: number of taxa, height, rate) shows averages and standard deviations of gene tree estimation error (across 1000 genes) for each replicate. Note wide variations in gene tree error across and within replicates. Bottom: both tree height and rate (left) affect gene tree estimation error; more ILS and deeper speciation result in higher error rates. With fixed tree shape (2M, 1e-06), changing the number of taxa (right) has little impact on the gene tree estimation error.

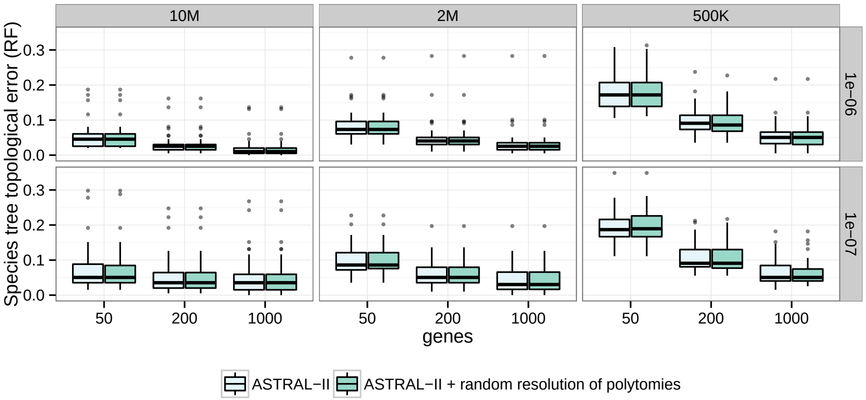


Figure 5.11: **Impact of polytomies**. Comparison of ASTRAL-II run on estimated gene trees with polytomies output by FastTree and with random resolutions of polytomies. Results are shown for dataset-I.

Species tree methods: We compared ASTRAL-I only to ASTRAL-II, and after establishing the improvements obtained in ASTRAL-II, we focused on the new version and compared it to MP-EST, NJst and CA-ML run using FastTree. We ran all methods given a maximum of 4 days of running time and 24GB of memory. MP-EST only finished for datasets with at most 100 taxa within time limits. Because of its running time, we ran MP-EST once (one random seed number) for each analysis. NJst, ASTRAL-I and MP-EST could not handle polytomies; therefore, we randomly resolved polytomies in inputs of these methods. We also ran ASTRAL-II on gene trees with randomly resolved polytomies and observed no differences with ASTRAL-II run on gene trees with polytomies (Fig. 5.11). Thus, differences between ASTRAL-II and other methods were not due to the random resolutions of polytomies.

5.4.1.3 Evaluation criteria

We evaluate methods in terms of species tree error and we also evaluate running time for coalescent-based methods. Species tree error is measured using the standard normalized RF distance. Running time of summary methods gives the wall clock running time and is measured on a heterogeneous Condor cluster at the University of Texas, Computer Science department.

5.4.2 Simulation results

We start by comparing ASTRAL-II with ASTRAL-I in terms of accuracy and running time (RQ1). We next focus on ASTRAL-II and compare it to other coalescent-based methods (RQ2) and then compare it to CA-ML (RQ3). This question leads us to a more in depth analysis of the effects of gene tree estimation error on the accuracy of various methods (RQ4). Finally, we evaluate the impact of collapsing low support branches in input gene trees on the accuracy of ASTRAL-II (RQ5).

5.4.2.1 RQ1: ASTRAL-I versus ASTRAL-II

Search space: ASTRAL-II adds extra bipartitions to the search space, which allows it to explore a larger search space; this tends to increase the accuracy of ASTRAL-II over ASTRAL-I. In our simulations, the extent of the improvement depended on the model condition. Table 5.3 shows the improvements obtained by ASTRAL-II compared to ASTRAL-I, and Figures 5.12 and 5.13 compare the two methods in terms of accuracy for Datasets I and II. In

Dataset I, with the lowest level of ILS or with the medium ILS level and recent speciation, ASTRAL-I and ASTRAL-II both had extremely low error (Fig. 5.12) and no substantial improvements were detected by the addition of extra bipartitions (Table 5.3). With 2M length and deep speciation, ASTRAL-II improved upon ASTRAL-I substantially, with improvements ranging from 3.5% with 1000 genes to 10.1% with 50 genes. Most dramatic differences were observed on the high ILS conditions, where ASTRAL-I performed extremely poorly, but ASTRAL-II reduced the error by about 40% (Table 5.3). Results on Dataset II showed that the effect of adding extra bipartitions also depended on the number of taxa in expected ways (Table 5.3): ASTRAL-I was as accurate as ASTRAL-II for up to 200 taxa, but with 500 taxa or more, ASTRAL-II had a substantial advantage (as large as 9%). As expected, the advantage of ASTRAL-II was larger with few genes and reduced with more genes.

The improvements obtained by ASTRAL-II are due to additions to the search space. We therefore asked whether the heuristic approaches used to add bipartitions to set $\mathcal{X}$ are sufficient, or improvements could be obtained by further expanding $\mathcal{X}$. To answer this question, we tested the impact of adding all the bipartitions from the species tree to the set $\mathcal{X}$, and compared ASTRAL-II with and without these extra bipartitions (see Figs. 5.12 and 5.13). We saw no significant differences between ASTRAL-II with and without these potentially new bipartitions (p=0.77 according to a two-way ANOVA test), indicating that the accuracy of ASTRAL-II is very unlikely to be improved further by expanding the search space.

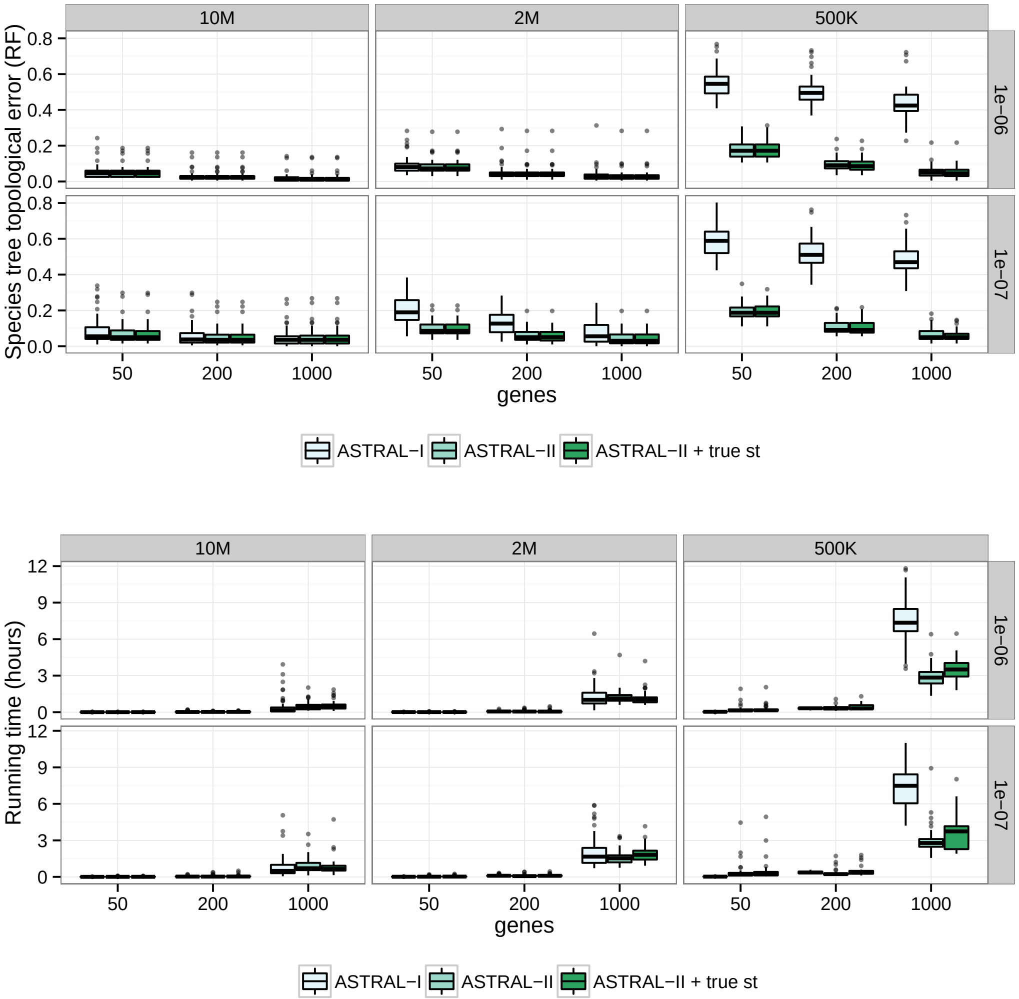


Figure 5.12: **Comparison of ASTRAL-I and ASTRAL-II on Dataset-I**. Species tree error (top) and running times (bottom) are shown. "ASTRAL-II + true st" shows the case where the true species tree is added to the search space; this is included to approximate an ideal solution (e.g. exact) where the set $\mathcal{X}$ includes all bipartitions that lead to the optimal score.

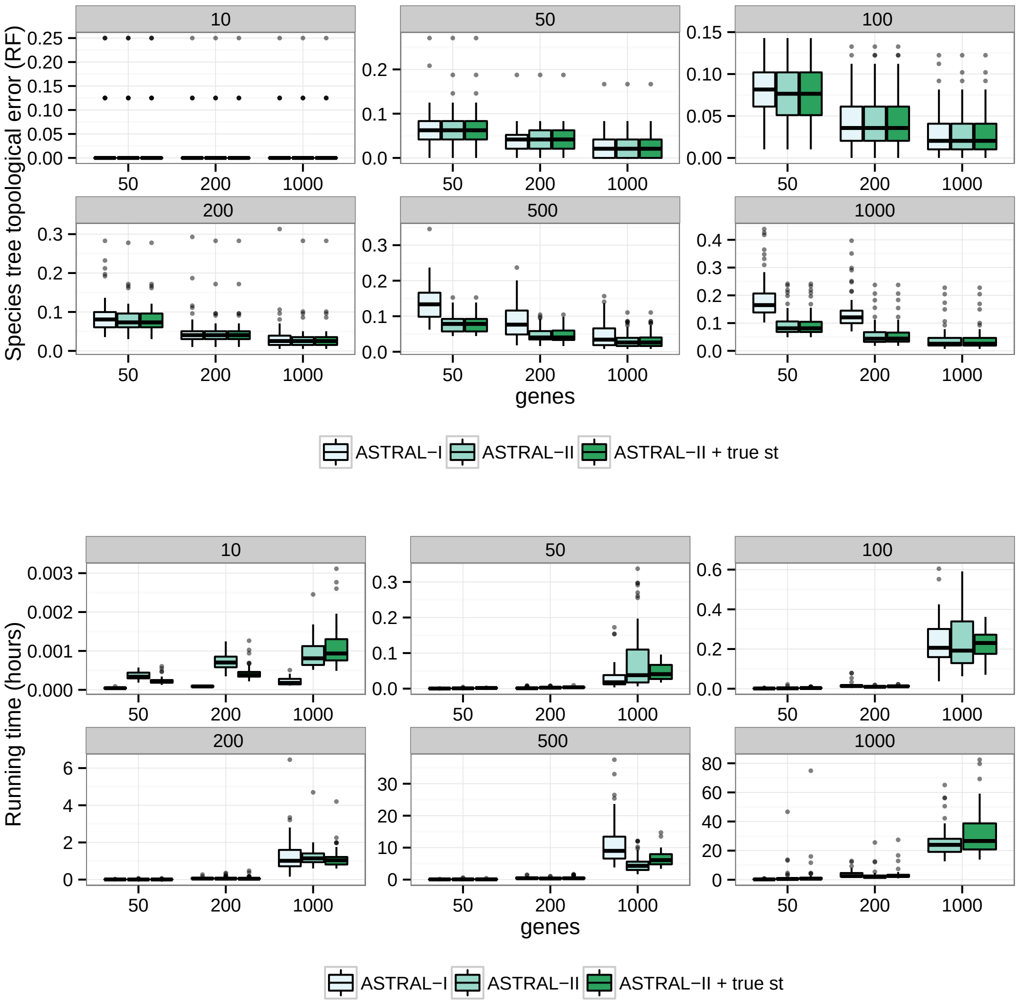


Figure 5.13: **Comparison of ASTRAL-I and ASTRAL-II on Dataset-II**. Species tree error (top) and running times (bottom) are shown. “ASTRAL-II + true st” shows the case where the true species tree is added to the search space; this is included to approximate an ideal solution (e.g. exact) where the set $\mathcal{X}$ includes all bipartitions that lead to the optimal score.

Table 5.3: **Reductions in species tree error obtained by ASTRAL-II compared to ASTRAL-I**. We report results using the difference in RF percentage; values above 0.0% indicate ASTRAL-II is more accurate.

Dataset I [200 taxa, varying tree shape (columns) and number of genes (rows)]

	10e-6 (recent)			10e-7 (deep)		
	10M	2M	500K	10M	2M	500K
50	0.2±0.2	0.7±0.3	37.9±1.0	1.7±0.6	10.1±0.9	38.7±0.9
200	0.0±0.1	0.2±0.1	41.0±1.1	0.7±0.3	7.4±0.7	41.4±1.0
1000	0.0±0.0	0.2±0.1	39.2±1.2	0.0±0.0	3.5±0.7	41.4±1.1

Dataset II [2M/1e-6 shape, varying the number of taxa (columns) and genes (rows)]

	10	50	100	200	500	1000
50	0.3±0.3	0.0±0.1	0.3±0.2	0.7±0.3	6.0±0.6	9.3±0.6
200	0.0±0.0	0.0±0.0	0.0±0.0	0.2±0.09	3.9±0.5	8.3±0.5
1000	0.0±0.0	0.1±0.1	0.0±0.0	0.2±0.08	1.7±0.4	

Running time: With 200 taxa and lower levels of ILS, ASTRAL-I and ASTRAL-II had similar running times (Fig. 5.12), but ASTRAL-II was faster with increased ILS (3 versus 7.5 hours of median run time). The improvement in speed is noteworthy, given that ASTRAL-II searches a larger tree space than ASTRAL-I. With small numbers of taxa, the two versions had close running times, but as the number of taxa increased, the running time of ASTRAL-II increased more slowly (Fig. 5.13). For 500 taxa, ASTRAL-II was twice as fast as ASTRAL-I (a median of 5 versus 10 hours), while ASTRAL-I did not complete on 1000 taxa and 1000 genes.

5.4.2.2 RQ2: ASTRAL-II vs. other summary methods

Completion within time constraints: ASTRAL-II completed on all model conditions, MP-EST completed only on datasets with at most 100 taxa, and NJst completed on all model conditions except for the condition with 1000 genes and 1000 taxa.

Dataset I: ASTRAL-II was more accurate than NJst in all model conditions, except 1e-07/500K where the two methods had identical error (Table 5.4, Fig. 5.14). Overall, the differences between ASTRAL-II and NJst were statistically significant ($p < 10^{-5}$), according to a two-way ANOVA test, and the relative performance of the methods was significantly impacted by the speciation rate ($p = 0.026$) but not by the number of genes or tree length. ASTRAL-II was faster than NJst, in some cases by an order of magnitude (Fig. 5.15).

Dataset II: On 10-taxon datasets all methods had high accuracy (Table 5.12). On 50- and 100-taxon datasets, MP-EST was able to finish, but it was the least accurate of all the methods. ASTRAL-II was more accurate than NJst for all conditions except for 50 taxa with 50 genes (Table 5.12); however, differences were generally small when the number of taxa was 200 or less, and more substantial with more taxa. Overall, differences between ASTRAL-II and NJst were significant ($p = 0.0007$) and were significantly impacted by the number of taxa ($p = 0.0004$) but not the number of genes. ASTRAL-II was also faster than NJst, especially with more genes and more taxa (Fig. 5.15).

Table 5.4: **Species tree error on Dataset I of ASTRAL-II analyses**. We show average and standard error of RF percentage. ASTRAL-II is always more accurate than NJst, but CA-ML (using FastTree) is sometimes more accurate than ASTRAL. For each row, the lowest average error and those error values that have an overlapping standard error with the lowest error value are in bold.

rate	height	genes	ASTRAL-II	NJst	CA-ML
1e-06	10M	50	**5.2±0.5**	**5.6±0.6**	**5.4±0.3**
1e-06	10M	200	**3.1±0.4**	**3.4±0.5**	**3.1±0.3**
1e-06	10M	1000	**2.0±0.4**	2.3±0.5	**1.4±0.2**
1e-06	2M	50	**8.4±0.6**	**9.1±0.7**	**9.2±0.4**
1e-06	2M	200	**5.0±0.6**	**5.6±0.6**	**5.5±0.5**
1e-06	2M	1000	**3.4±0.6**	3.9±0.6	**2.8±0.4**
1e-06	500K	50	**17.6±0.7**	20.9±0.7	27.9±0.7
1e-06	500K	200	**9.6±0.5**	11.0±0.5	16.2±0.7
1e-06	500K	1000	**5.3±0.5**	**5.7±0.4**	8.0±0.3
1e-07	10M	50	7.3±0.9	10.2±1.0	**4.0±0.4**
1e-07	10M	200	5.4±0.7	8.2±1.0	**2.2±0.3**
1e-07	10M	1000	5.0±0.8	8.0±1.0	**1.8±0.3**
1e-07	2M	50	**10.2±0.6**	11.7±0.7	**10.3±0.3**
1e-07	2M	200	**6.0±0.5**	7.5±0.7	**5.7±0.3**
1e-07	2M	1000	4.4±0.6	6.0±0.7	**2.8±0.2**
1e-07	500K	50	**19.3±0.7**	22.5±0.6	28.2±0.6
1e-07	500K	200	**10.7±0.6**	**11.4±0.5**	16.1±0.7
1e-07	500K	1000	**6.3±0.5**	**6.3±0.5**	8.0±0.4

Table 5.5: **Species tree error on Dataset II. of ASTRAL-II analyses**. We show average and standard error of RF percentage. Note that ASTRAL-II is always more accurate than MP-EST, and more accurate than NJst under all conditions except one (50 taxa and 50 genes), where NJst is slightly more accurate (7.2% vs. 7.3%). CA-ML (using FastTree) is also less accurate than ASTRAL, except for 100 taxon and 200 or 1000 genes, where the two methods differ in less than 0.5%. For each row, the lowest average error and those error values that have an overlapping standard error with the lowest error value are in bold.

taxa	genes	ASTRAL-II	NJst	CA-ML	MP-EST
10	50	**2.8±1.0**	**2.8±1.0**	3.8±0.9	**2.8±1.0**
10	200	**1.5±0.7**	**1.5±0.7**	**1.8±0.7**	**1.8±0.7**
10	1000	**1.5±0.7**	**1.8±0.7**	**2.1±0.8**	**1.5±0.7**
50	50	**7.3±0.7**	**7.2±0.6**	**7.8±0.6**	13.5±1.7
50	200	**4.2±0.5**	**4.4±0.5**	**4.5±0.4**	9.1±1.5
50	1000	**2.6±0.4**	**2.7±0.5**	**2.7±0.4**	8.2±1.5
100	50	**7.9±0.5**	**8.7±0.5**	9.1±0.4	16.9±1.3
100	200	**4.8±0.5**	**5.1±0.6**	**4.7±0.4**	13.7±1.5
100	1000	**3.0±0.4**	3.9±0.6	**2.5±0.3**	14.1±1.55
200	50	**8.4±0.6**	**9.1±0.7**	**9.2±0.4**	
200	200	**5.0±0.6**	**5.6±0.6**	**5.5±0.5**	
200	1000	**3.4±0.6**	3.9±0.6	**2.8±0.4**	
500	50	**8.0±0.4**	9.7±0.5	9.2±0.3	
500	200	**4.9±0.3**	6.1±0.5	**4.7±0.2**	
500	1000	3.3±0.4	4.7±0.5	**2.3±0.1**	
1000	50	**9.9±0.7**	12.1±0.9	**9.8±0.3**	
1000	200	**6.0±0.7**	7.9±0.9	**5.1±0.2**	
1000	1000	**4.5±0.7**			

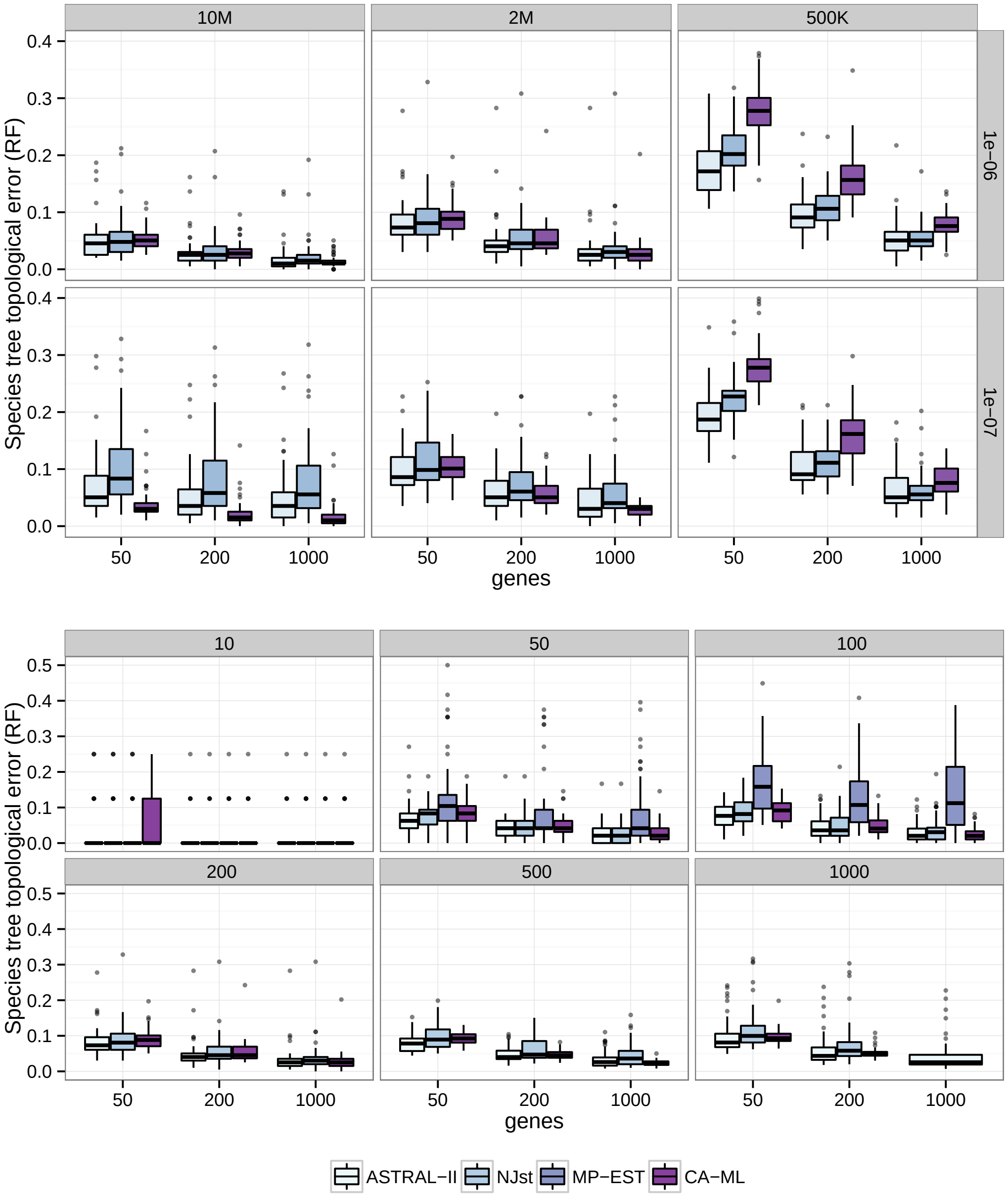


Figure 5.14: **Comparison of methods with respect to species tree topological error on ASTRAL-II simulated data**. Species tree error is shown for Dataset-I (top) and Dataset-II (bottom). ASTRAL-II is always at least as accurate as NJst and MP-EST, but CA-ML (using FastTree) is under some conditions more accurate.

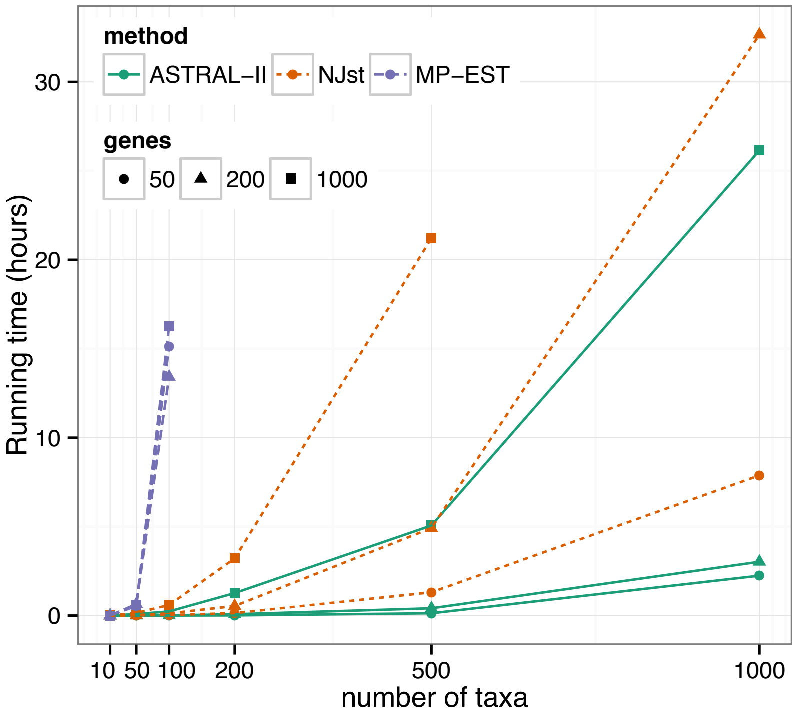


Figure 5.15: **Running time comparison with varying number of taxa and genes on Dataset II**. Average running time is shown for NJst and ASTRAL-II. Note that ASTRAL-II is much faster on large datasets.

For example, on 500 taxa and 1000 genes, ASTRAL-II typically finished in 2 to 10 hours, whereas NJst required 12 to 30 hours. MP-EST was the slowest method, but its running time was not impacted by the number of genes.

5.4.2.3 RQ3: ASTRAL-II vs. CA-ML

Dataset I: Interestingly, the relative accuracy of CA-ML and ASTRAL-II was significantly impacted by tree length ($p < 10^{-5}$), speciation rate ($p =$

0.00004), and the number of genes ($p < 10^{-5}$). With lower levels of ILS (10M and 2M) and recent speciation, CA-ML and ASTRAL-II had close accuracy, but CA-ML tended to be better with more genes and ASTRAL-II was better with fewer genes (Table 5.5, Fig. 5.14). With deep speciation and lower ILS, CA-ML was substantially more accurate than ASTRAL-II, but increasing the number of genes reduced the gap. At the high ILS levels, ASTRAL-II was much more accurate than CA-ML for all number of genes and for both recent and deep speciation.

Dataset II: Overall, differences between ASTRAL-II and CA-ML were not significant ($p = 0.2$), but the relative accuracy seemed to be impacted by the number of genes ($p = 0.06$). Regardless of the number of taxa, which did not impact relative accuracy ($p = 0.2$), CA-ML was slightly more accurate with 1000 genes, and ASTRAL-II was slightly often more accurate otherwise (Table 5.5, Fig. 5.14).

Running time: We ran CA-ML and ASTRAL-II on different platforms, and hence cannot make direct running time comparisons. Nevertheless, we provide our running time numbers to give a general idea. CA-ML using FastTree on 200-taxon model conditions with 1000 genes took roughly two hours, whereas ASTRAL-II took roughly one hour to estimate the species tree, and estimating gene trees also took about 1.5 hours. In general, therefore, the running times of ASTRAL-II and CA-ML are relatively close on this dataset.

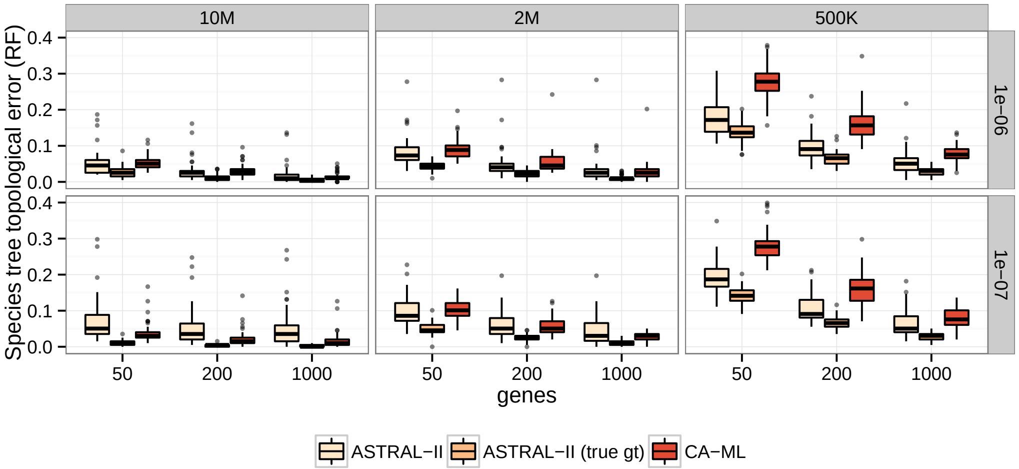


Figure 5.16: **Comparison of ASTRAL-II run on estimated and true gene trees and CA-ML on Dataset I**. The different between ASTRAL-II with true gene tree ("true gt") and ASTRAL-II with estimated gene trees indicates the impact of gene tree error. Note that with true gene trees, ASTRAL has excellent accuracy and is always better than CA-ML (using FastTree).

5.4.2.4 RQ4: Effect of gene tree error

In RQ3, we observed that under some conditions, CA-ML was more accurate than ASTRAL-II, a pattern that we attribute to high levels of gene tree error present in our simulations. When true (simulated) gene trees are used instead of the estimated gene trees, the accuracy of ASTRAL-II is outstanding, regardless of the model condition (see Fig. 5.16) and ASTRAL-II is always more accurate than CA-ML. Thus, the fact that CA-ML is occasionally more accurate than ASTRAL-II under lower levels of ILS is related to estimation error in the input provided to ASTRAL-II.

In our ASTRAL-II and NJst analyses, gene tree error had a positive correlation with species tree error (Fig. 5.17), with correlation coefficients that were similar for ASTRAL-II and NJst. The error of CA-ML also correlated with gene tree error (obviously the relationship is indirect as factors such as short alignments impact both CA-ML and gene tree error), but the correlation was weaker than the correlation observed for coalescent-based methods (Fig 5.18). Interestingly, the correlation between gene tree estimation error and species tree error was typically higher with fewer genes.

To further investigate the impact of the gene tree error, we divided replicates of each model condition into three categories: average gene tree estimation error below 0.25 is labelled low, between 0.25 and 0.4 is labelled medium, and above 0.4 is labelled high. We plotted the species tree error within each of these categories (see Figs. 5.19 and 5.20). The relative performance of ASTRAL-II and NJst is typically unchanged across various categories of gene tree error, but increasing gene tree error tends to increases in the magnitude of the difference between ASTRAL-II and NJst. Furthermore, MP-EST seemed to be more sensitive to gene tree error than either NJst or ASTRAL-II (Fig. 5.20).

The relative performance of ASTRAL-II and CA-ML depended on gene tree error. For those model conditions where CA-ML was generally more accurate than ASTRAL-II (e.g., 2M/1e-07), ASTRAL-II tended to outperform CA-ML on the replicates with low gene tree estimation error (Fig. 5.19). Consistent with this observation, we noted that ASTRAL-II was impacted by gene

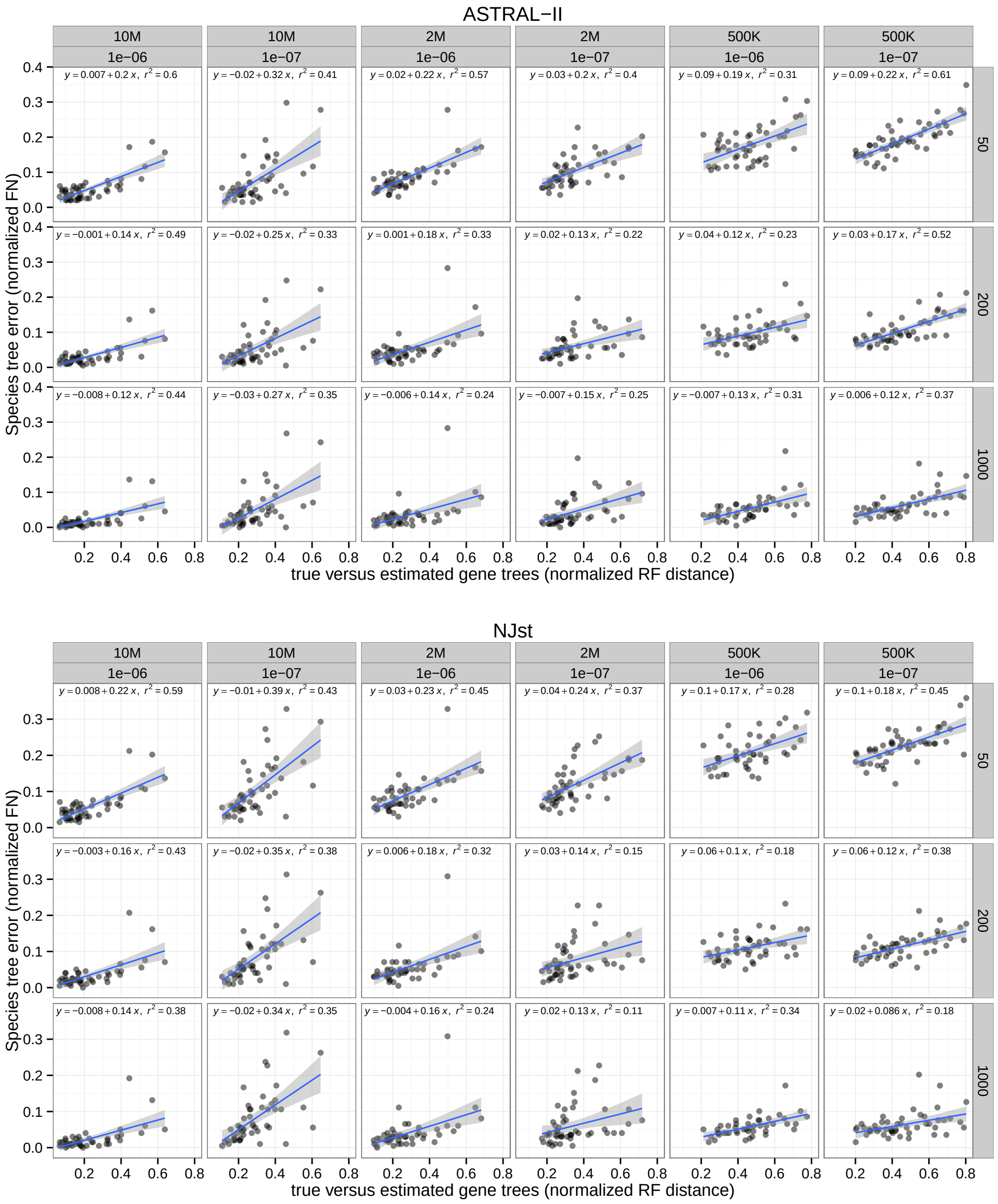


Figure 5.17: **Correlation between gene tree estimation error and species tree error for ASTRAL and NJst on Dataset-I**. Gene tree and species tree error correlate well, and the correlation is stronger for fewer genes and *lower* levels of ILS. Varying tree shapes are shown in columns and numbers of genes are showed in rows.

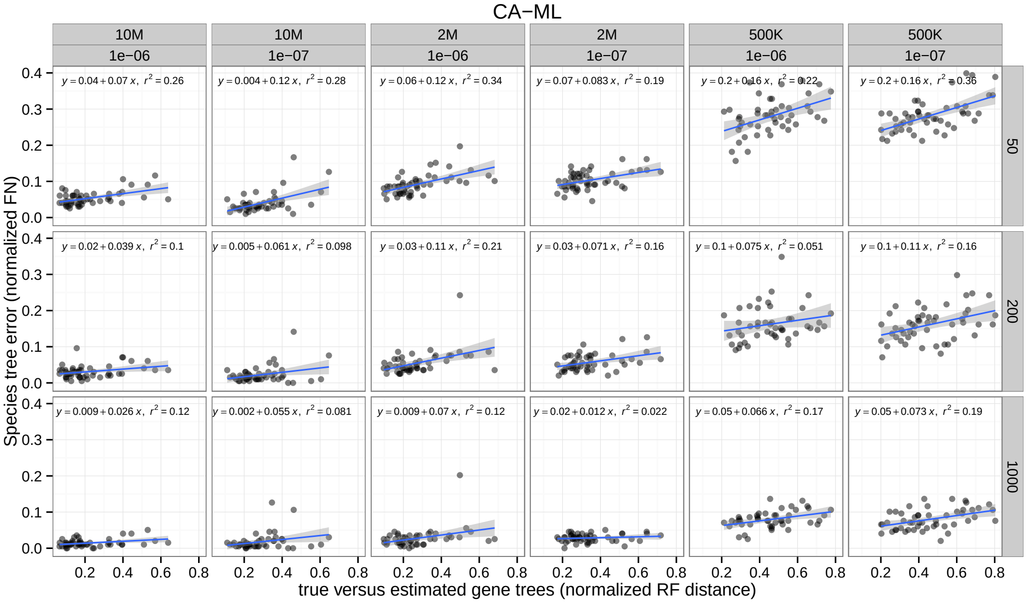


Figure 5.18: **Correlation between gene tree estimation error and species tree error for CA-ML on Dataset-I**. A correlation between gene tree error (controlled by parameters such as alignment length that also affect concatenation) and species tree error is detectable for concatenation, but is smaller compared to NJst and ASTRAL.

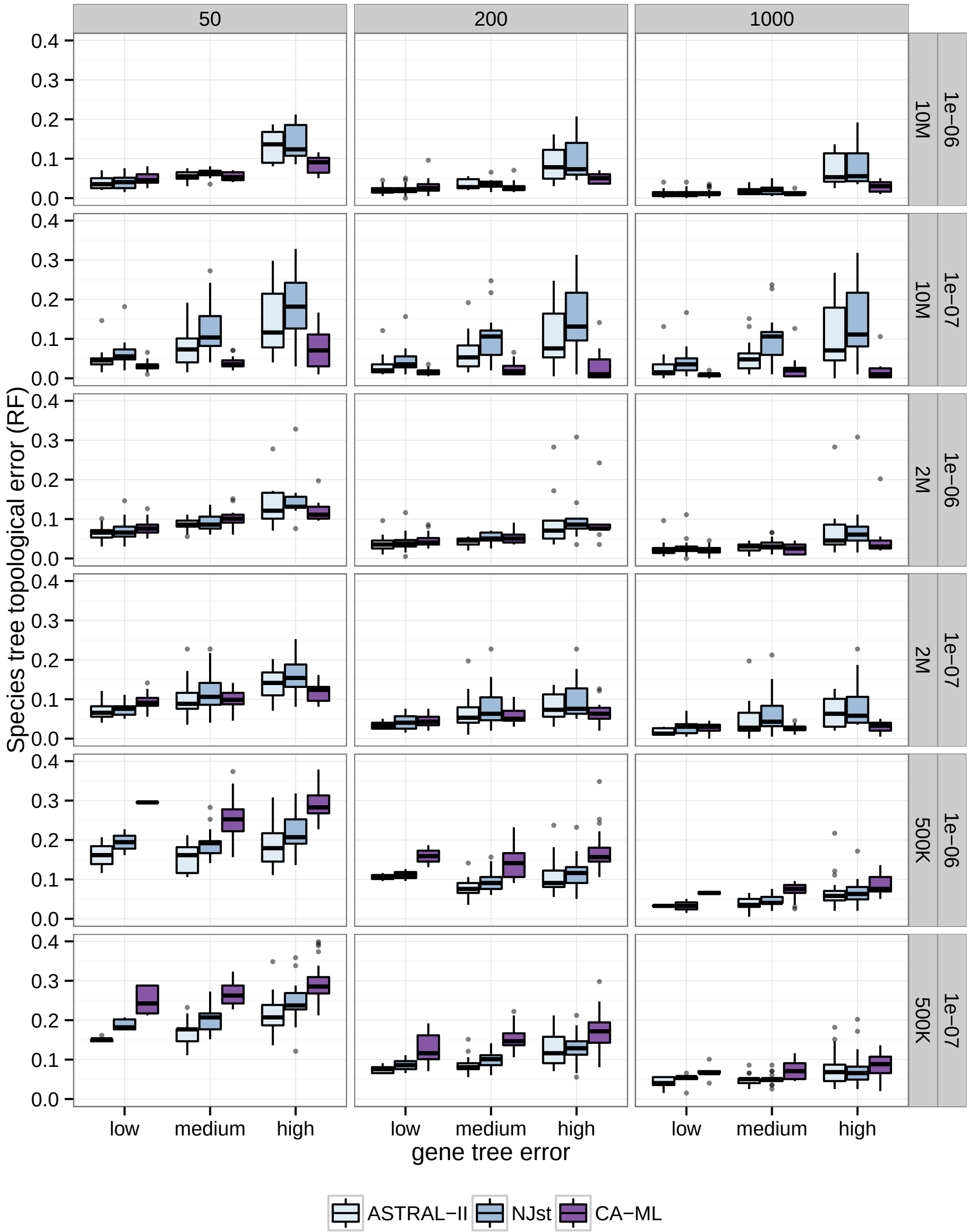


Figure 5.19: **Comparison of species tree error on Dataset-I, divided into three categories of gene tree estimation error**. Results are shown for 200 taxa and varying tree shapes (rows), and varying number of genes (columns), divided into three categories of gene tree estimation error: low, medium, and high.

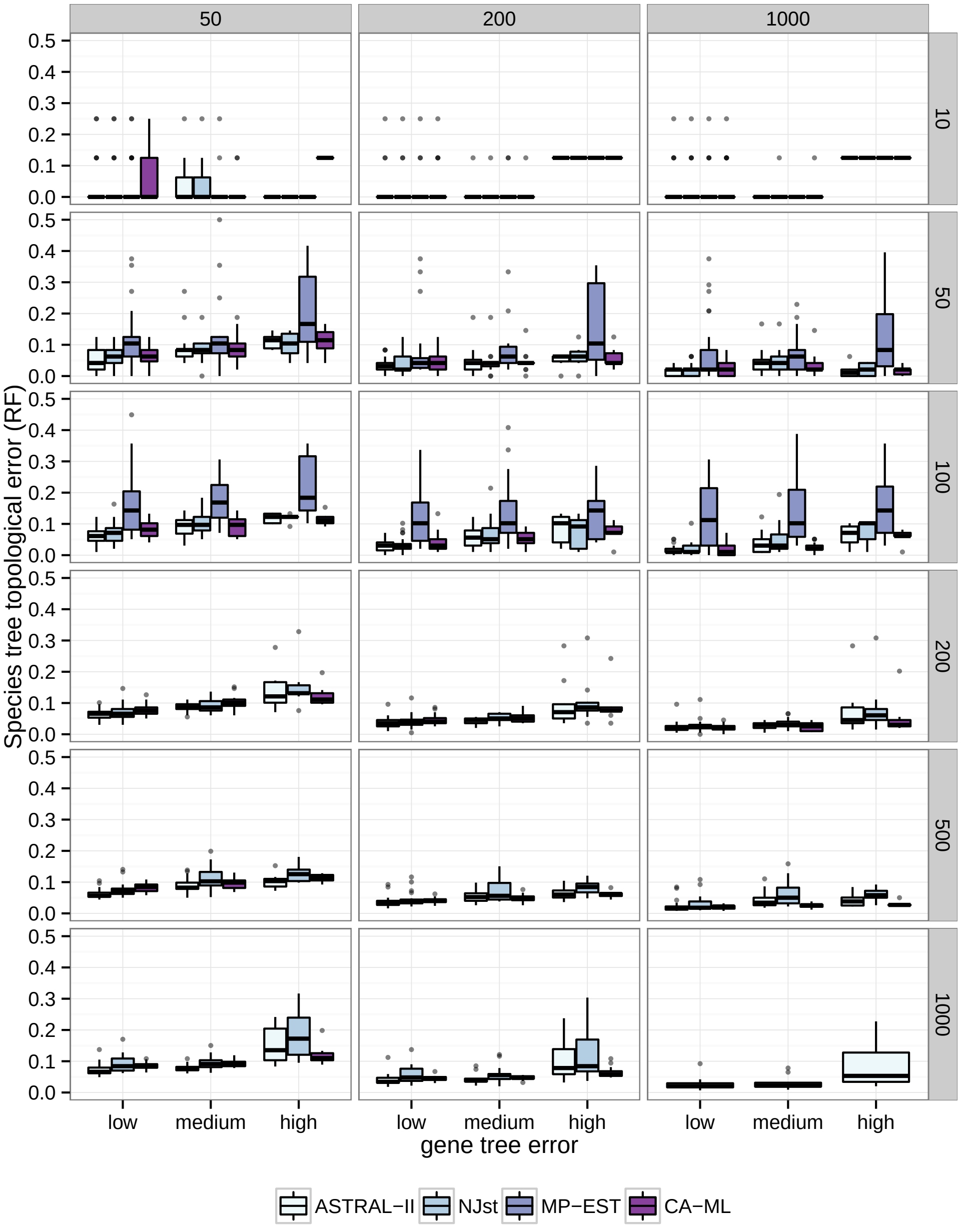


Figure 5.20: **Comparison of species tree error on Dataset-II, divided into three categories of gene tree estimation error**. Results are shown for varying number of taxa (rows), and varying number of genes (columns), divided into three categories of gene tree estimation error: low, medium, and high.

tree error more than CA-ML (Fig. 5.19).

5.4.2.5 RQ5: Collapsing low support branches

ASTRAL-II can handle inputs with polytomies. In this study, because of the prohibitive costs of applying bootstrapping to datasets of this size, we have not done bootstrapping on our genes to get reliable measures of support. However, we do get local SH-like branch support [250] from FastTree-II. Using these SH-like support values, we collapsed low support branches (10%, 33%, and 50%) and ran ASTRAL-II on the resulting unresolved gene trees. We measured the impact of contracting low support branches on the species RF rate. The median delta RF (error before collapsing minus error after collapsing) is typically zero (Fig. 5.21), never above zero, but in a few cases below zero (signifying that accuracy was improved in those few cases). However, these differences are not statistically significant ($p = 0.36$). Since this analysis was performed using SH-like branch support values instead of bootstrap support values (or other ways of estimating support values), it's hard to generalize and make conclusions about the use of other measures of support. Further studies are therefore needed for understanding the effect of collapsing low support branches in other situations.

5.4.3 Summary of results

Our wide-ranging simulation results show that ASTRAL-II, unlike the other methods we studied, can analyze datasets with up to 1000 taxa and

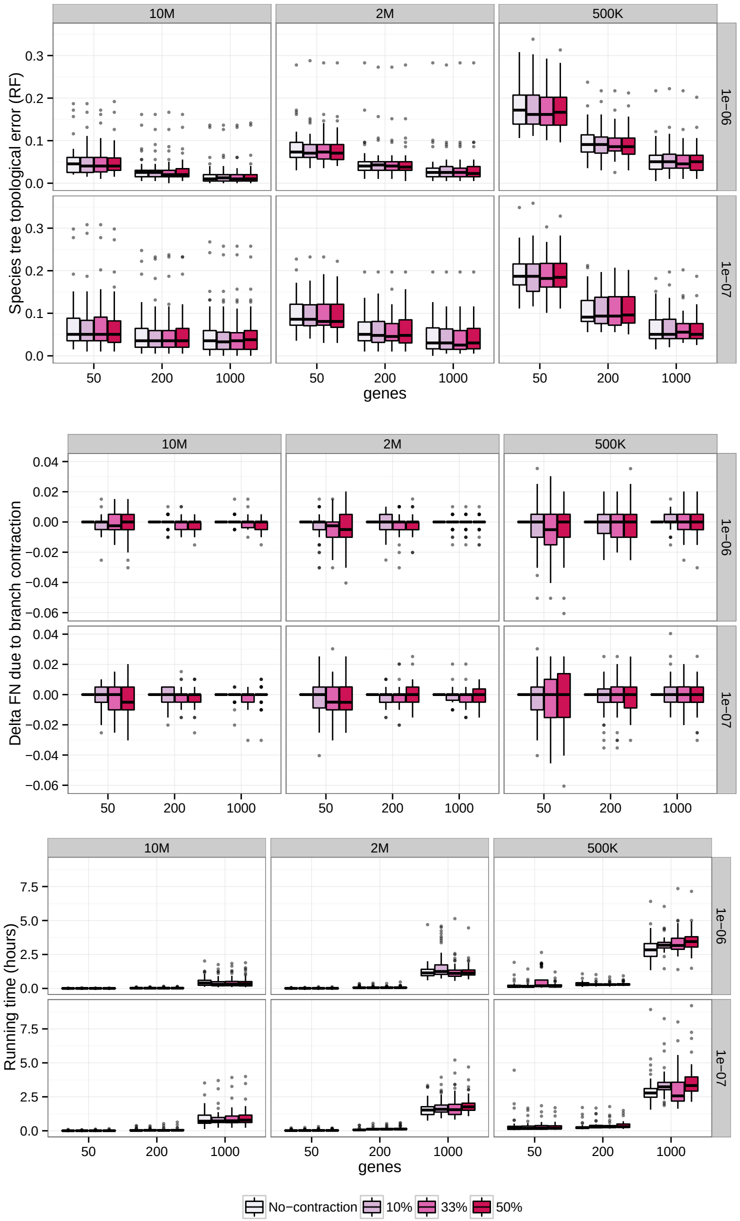


Figure 5.21: **Effect of contracting low support branches on ASTRAL-II**. Gene tree branches with FastTree SH-like local support below 10%, 33%, and 50% were contracted before running ASTRAL-II. Species tree error (top), change in species tree accuracy (middle) and running times (bottom) are shown. Delta FN (middle) shows changes in error compared to using binary trees, and so Delta FN < 0 indicates collapsing low branches improved accuracy.

1000 genes within reasonable running times. The next most computationally feasible method we explored was NJst, but ASTRAL-II was faster and more accurate than NJst. ASTRAL-II was also much more accurate than MP-EST, especially with larger numbers of species, but MP-EST was much slower and could not run on datasets with more than 100 species. Finally, ASTRAL-II improved upon ASTRAL-I in terms of both accuracy and running time. ASTRAL-II was more accurate than CA-ML, except when gene tree estimation error was high and ILS levels sufficiently low.

5.5 Biological Results

5.5.1 Datasets and methods

We analyzed five biological datasets:

- The 1KP dataset from [40], containing 103 plant species and 424 genes.
- The land plant dataset from [211], containing 32 species and 184 genes.
- The angiosperm dataset from [193] containing 42 angiosperm species and 4 outgroups with 310 genes.
- The mammalian dataset from [77], containing 37 species and 447 genes.
- The amniota dataset from [189], containing 16 species and 248 genes.

On these datasets, we compare ASTRAL-II, MP-EST, and concatenation using RAxML (CA-ML). We use gene trees that we estimated for the 1KP project; for the amniota and land plant datasets, gene trees were available from the respective publications. For the mammalian and the angiosperm datasets, we re-estimated gene trees from the gene alignments that were available. We used RAxML under the GTR+Γ model with 200 replicates of bootstrapping and 10 rounds of ML. We used the MLBS procedure [208] to obtain BS values (see Section 4.3.2).

As noted in Chapter 4, in our analysis of the mammalian dataset, we found 21 genes with mis-labelled sequences (easily confused taxon names, subsequently confirmed by the authors of [77]). We removed all those and two outliers genes from the dataset, and re-analyzed the reduced dataset. We used the MLBS procedure with 100 replicates, with both site and gene resampling, in order to be consistent with [77]. We re-estimated the gene trees using RAxML on the gene sequence alignments produced by [77].

On the amniota dataset, since the number of taxa is small, we ran the exact version of ASTRAL; in other cases, we ran ASTRAL-II.

5.5.2 Results

5.5.2.1 1KP dataset

As we noted earlier, analyzing 1KP dataset was one of our motivations for designing a new summary method. This dataset was very challenging for existing summary methods; it had 103 species, which is larger than what most methods are designed for and tested on. Also, since the 103 taxa span close to a billion years of evolution, rooting gene trees was challenging; finally, no single gene tree was complete, and some gene trees had substantial levels of missing data (note that this also affects the ability to root gene trees) As we noted before, the other summary methods were not able to produce reliable species trees on this dataset.

There are several interesting questions about plant evolution that this dataset can help answering, but three stand out.

Sister to land plants: The sister species to a clade including all the land plants remains unresolved. Two sets of streptophyte algae, Charales, and Coleochaetales, share complex characteristics with land plants (e.g., oogamous sexual reproduction and parental retention of the egg), which traditionally lead to the belief that Charlales, or Charales+Coleochaetales are sister to land plants. However, previous molecular analyses have inferred many different possible sister clades, including the following four major hypotheses: Zygnematales [251–253], Coleochaetales [254], Zygnematales + Coleochaetales [255], and Charales [256].

Bryophytes: Mosses, liverworts, and hornworts (collectively called bryophytes) are plants that separated out from other land plants early in the evolution of land plants. All various possible hypothesis of branching order involving these groups has been proposed in the literature and many have been supported by various data [257–259].

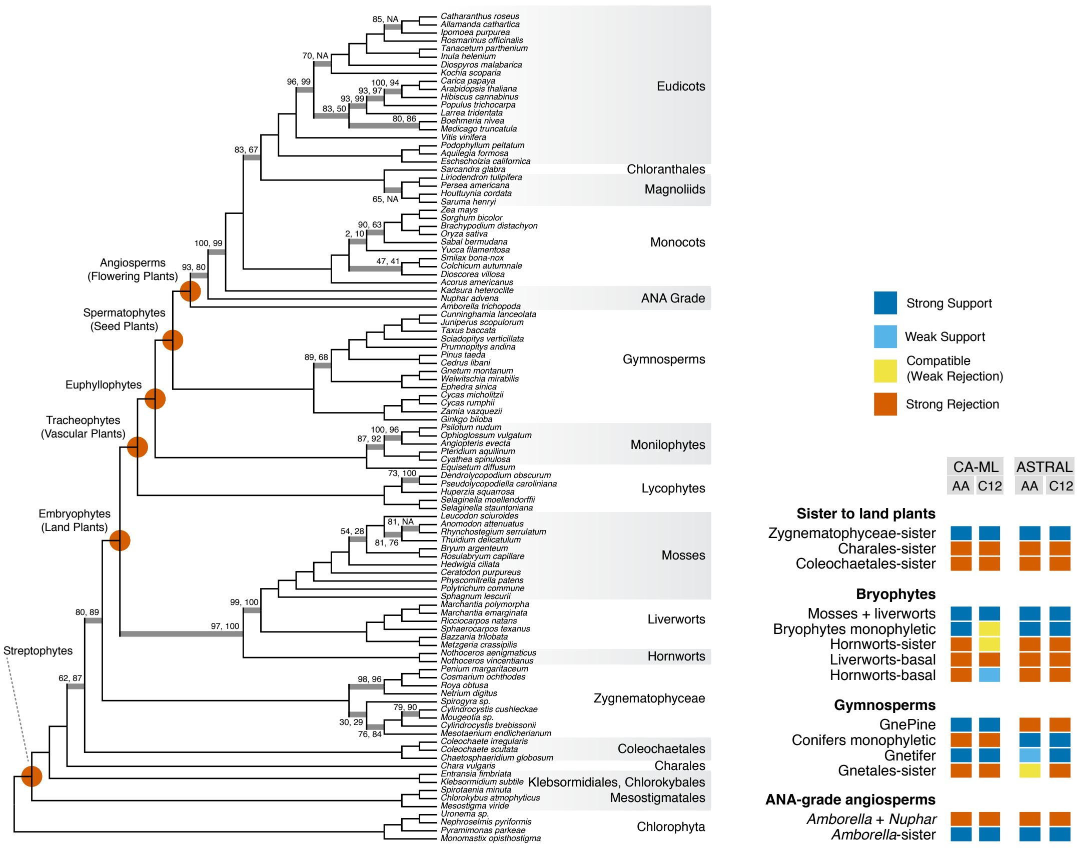
Gnetales: The position of Gnetales within a monophyletic gymnosperm clade is also unresolved, with various hypotheses recovered in the literature [260–262].

Angiosperms: The earliest branch that diverged from the remaining flowing plants (angiosperms) has been the subjective of debate. Amborella and Nymphaeales (water lilies), have been identified as earliest branches of the tree [263, 264]; however, it is not clear whether Amborella [264, 265] or a clade containing Nymphaeales+Amborella [266, 267] should be placed as sister to all other extant angiosperm lineages.

In its initial phase, the 1KP project gathered entire transcriptomes of 103 different plant species, and from those gathered a set of 852 single-copy putatively orthologous genes [40]. As part of the 1KP project, we estimated gene trees on all 852 genes, and then analyzed them in various ways, including various ways of filtering data. An important filtering was to remove fragmentary data from gene alignments. Fragments can reduce alignment accuracy [173], and can also result in poorly estimated gene trees. After removing sequences that were more than 66% gaps, and removing genes that were missing more than 50% of the sequence data, we obtained a dataset that included 424 gene trees (close to half of gene trees had less than half of the species and these were removed). We estimated gene trees based on amino acid sequences and also on DNA sequences with 3rd codon position removed (to avoid effects of GC bias [40, 268]). We report results on these two sets of 424 gene trees, and refer the reader to [40] for other analyses on the complete dataset. As mentioned in Section 5.1, our attempts at running MP-EST on this dataset had limited success.

Figure 5.22 shows the ASTRAL tree on 1KP and summarizes the differences between CA-ML and ASTRAL trees. Both concatenation and ASTRAL recover Zygnematales as sister to land plants, with high support. Similarly, the sister to flowering plants is recovered to be Amborella with high support, regardless of the dataset used or whether ASTRAL or CA-ML was used.

The relationships among Bryophytes and Gymnosperms are less consistent. In all analyses, mosses and liverworts were sister groups. However, in the CA-ML analysis of DNA sequences, hornworts were recovered with low support as the sister to all remaining land plants (a clade containing mosses, liverworts, and all the other land plants) whereas in both ASTRAL analyses and the CA-ML analysis of the AA data, hornworts were sister to mosses + liverworts, and this clade was at the base of land plants. The correct relationship is not known, but the fact that ASTRAL and concatenation recover different relationships is important, especially given short branch lengths at the base of land plants. Similarly, within Gymnosperms, the exact relationships recovered depend on the method used. ASTRAL analyses both recover Conifers as a monophyletic clade and Gnetales as the base of Gymnosperms, a topology previously recovered in other analyses [269]. However, CA-ML analyses put Gnetales as sister to pines, breaking the monophyly of Conifers (this topology was also previously observed [260]).



(a) ASTRAL tree

(b) Results summary

Figure 5.22: **Summary of results of 1KP dataset**. (a) ASTRAL results on the 1KP dataset (ASTRAL-I and ASTRAL-II produced identical results); the DNA tree is shown and the support values are shown for both DNA and AA astral analyses. Branches without designation have 100% support in both analyses. NA means a branch was missing from the AA analysis. (b) Summary of results. Rows show hypotheses of plant evolution for four parts of the tree. Columns show two ASTRAL and two CA-ML analyses (using RAxML). Colors indicate whether a hypothesis was supported, or rejected and whether support or rejection had support that was at least 75%.

5.5.2.2 Land plant dataset

The question of greatest interest on this dataset is the sister group to land plants. As noted before, our recent 1KP analysis recovered Zygnematales as sister to Land plants with high confidence using both ASTRAL and concatenation. Zhong *et al.* used MP-EST to analyze their data, and inferred Zygnematales as the sister with 64% BS [211]. A re-analysis of the same data using STAR was performed by Springer and Gatesy [33], who obtained Zygnematales + Coleochaetales with 44% BS.

We analyzed this dataset using ASTRAL-II and obtained a tree that generally has high BS on most branches (i.e., with the exception of four branches, all branches have support at least 86%, and most have 100% support). However, one edge had very low support (only 18%). After collapsing the single branch with very low support, we obtained a tree (see Fig. 5.23) in which the Charales + Land plants hypothesis is rejected with moderately high support (86%); however, it is not determined whether Zygnematales, Coleochaetales, or Zygnematales + Coleochaetales are the sister group to Land plants (the branch that distinguishes between these three hypotheses is the one with 18% support). Thus, ASTRAL's analysis of this dataset can be seen as suggesting that this dataset is insufficient to completely resolve the sister relationship to Land plants. However, the most interesting question is whether Charales are sister to Land plants, and the ASTRAL tree rejects that hypothesis with 86% support. The ASTRAL results, therefore, are consistent between the Zhong *et al.* dataset and 1KP dataset.

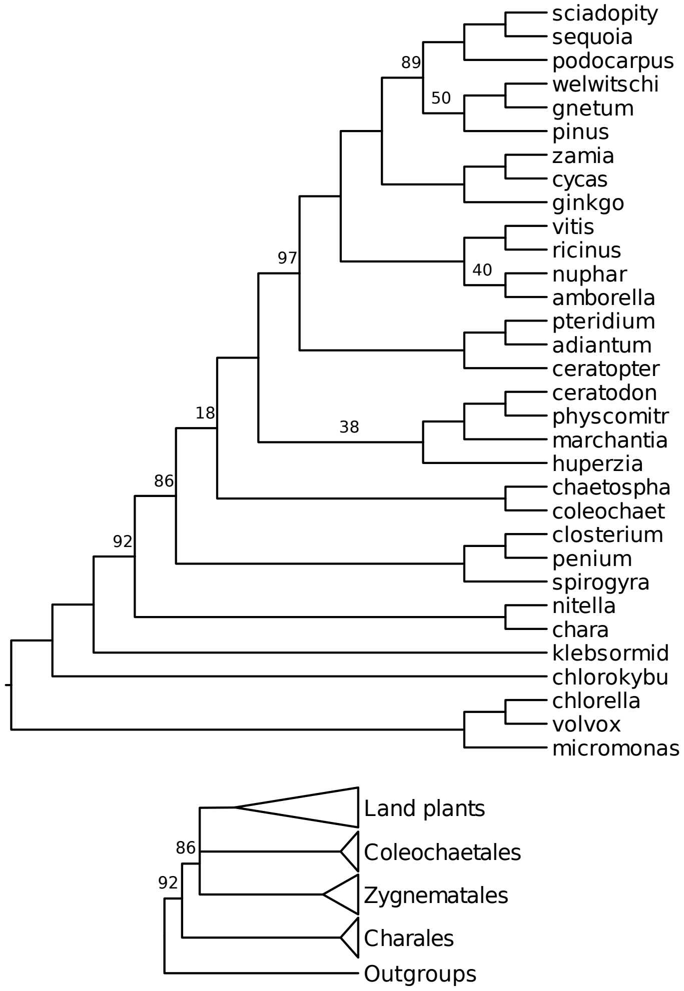


Figure 5.23: **ASTRAL tree on the Zhong *et al.* land plant dataset**. We analyzed a plant dataset with 32 species and 184 genes from [211]. Bootstrap support values were obtained using the multi-locus bootstrapping procedure with 100 replicates; values not shown indicate 100% support. ASTRAL-II tree (with bootstrap support values) is shown on top, and we show a cartoon version of the tree below. The cartoon version only shows the relationship between the 5 groups – Land plants, Coleochaetales, Zygnematales, Charales, and the outgroups, after collapsing the branch with bootstrap support of 18%. Note that there are three possible sister groups to Land plants: Coleochaetales, Zygnematales, or the two together (Zygnematales+Coleochaetales); however, Charlaes is strongly rejected as the sister group to Land plants.

5.5.2.3 Angiosperms

The evolution of angiosperms, and the placement of *Amborella trichopoda Baill.*, is one of the challenging questions in Land plant evolution. One hypothesis recovered in some recent molecular studies and all of our 1KP analyses is that *Amborella trichopoda* is sister to the rest of angiosperms, followed by Nymphaeales (e.g., see [40, 270–272]). A competing hypothesis is that Amborella is sister to Nymphaeales and this whole group is sister to other angiosperms [267, 272]. Xi *et al.* [193] have examined this question using a collection of 310 genes sampled from 42 angiosperms and 4 outgroups. They observed that concatenation using maximum likelihood (CA-ML) produced the first hypothesis and MP-EST produced the second hypothesis, and they argued that these differences are due to the fact that CA-ML does not model ILS, whereas MP-EST does.

We ran MP-EST and ASTRAL on the gene tees that we re-estimated on this dataset, and we obtained two different species trees (Fig. 5.24). Reproducing results by Xi *et al.*, MP-EST recovered the sister relationship of Amborella and Nymphaeales with 100% support. However, ASTRAL, just like CA-ML (using RAxML), recovers Amborella as sister to other angiosperms, with 75% support. While the exact position of Amborella is debated, our analysis shows that the differences between CA-ML and MP-EST results cannot be simply attributed to the fact that CA-ML does not consider ILS.

There are several possible reasons for the differences between the ASTRAL and MP-EST on this dataset, including the possibility that rooting

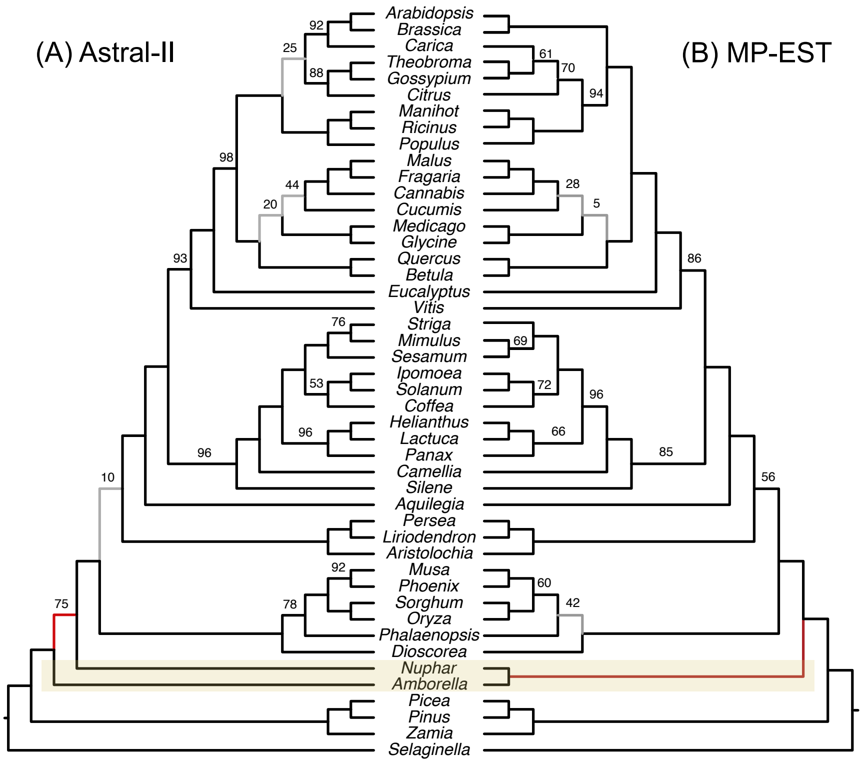


Figure 5.24: **Comparison of species trees computed on the angiosperm dataset**. MP-EST and ASTRAL-II differ in the placement of Amborella; the concatenation tree agrees with ASTRAL-II.

gene trees (required by MP-EST but not by ASTRAL-II) by *Selaginella* can be problematic for some genes, or that the impact of the gene tree estimation error is different for the two methods. We also note that ASTRAL-II is a non-parametric method that does not estimate branch lengths, and it is possible that non-parametric methods are less sensitive to gene tree estimation error than parametric methods (like MP-EST).

Our reanalysis of this dataset and our results on the 1KP dataset taken together point to more support for the hypothesis that Amborella is sister to the remaining flowering plants.

5.5.2.4 Mammalian

On the mammalian dataset, two of the questions of greatest interest were the placement of bats (Chiroptera) and tree shrew (Scandentia), where their MP-EST analysis differed from the concatenated analyses they performed. We recomputed the MP-EST tree, obtaining a tree topologically identical to the MP-EST tree reported in [77], but with lower bootstrap for the placement of Scandentia (62% in our analysis). CA-ML analyses of the full and reduced datasets using RAxML were topologically identical and had similar branch support. Thus, the CA-ML and MP-EST trees on the reduced dataset still differed in the placement of both Scandentia and Chiroptera.

We compare ASTRAL to MP-EST in Figure 5.25. Both ASTRAL and MP-EST trees placed Chiroptera as the sister to all other Laurasiatheria except Eulipotyphyla, while CA-ML placed Chiroptera as the sister to Cetar-

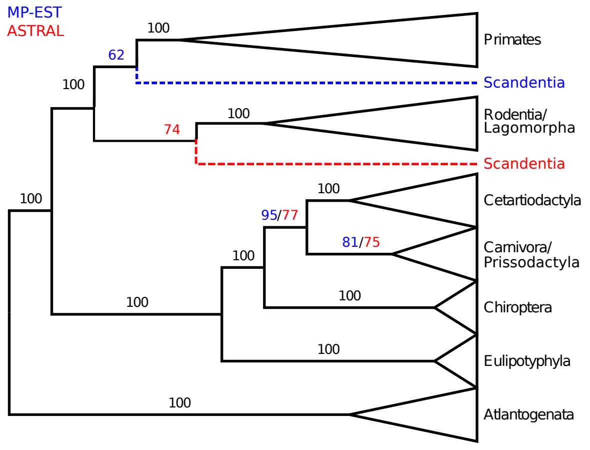


Figure 5.25: **Analysis of the Song et al. mammals dataset using ASTRAL and MP-EST**. We show the result of applying ASTRAL and MP-EST to 424 gene trees on 37-taxon mammalian species. MP-EST is based on rooted gene trees; ASTRAL is based on unrooted gene trees, and then rooted at the branch leading to the outgroup. Branch support values in black are for both methods, those in red are for ASTRAL, and values in blue are for MP-EST. See Chapter 4 for the resolution of collapsed clades.

tiodactyla. The ASTRAL tree placed Scandentia as sister to Glires with 74% support, and thus agrees with the CA-ML tree but differs from the MP-EST tree. Thus, the differences between CA-ML and MP-EST cannot simply be attributed to use of a coalescent-based method, as Song *et al.* conjectured, since ASTRAL, which is also coalescent-based, recovers the same relationship as MP-EST.

5.5.2.5 Amniota dataset

Chiari *et al.* [189] assembled a dataset of Amniota to resolve the position of turtles relative to birds and crocodiles. Most recent studies favor an

Archosaurus hypotheses that unites birds and crocodiles as sister groups [273]. The MP-EST analyses by [189] resolved this relationship differently when AA and DNA gene trees were used; thus, AA had 99% support for the Archosaurus clade, but DNA rejected Archosaurus with 90% support. We analyzed the same dataset using the exact version of ASTRAL and found that both AA and DNA recover Archosaurus; however, while ASTRAL on AA gene trees recovered Archosaurus with 100% support, ASTRAL on DNA gene trees had only 55% support for Archosaurus.

5.6 Discussions and future work

This study introduced ASTRAL, a method for estimating species trees from unrooted gene trees. We introduced two versions of ASTRAL, and proved that both versions are statistically consistent under the MSC model, but our second version, ASTRAL-II, has lowered running time and better empirical performance. Our simulation and biological results show that upcoming multi-gene datasets with large numbers of species can be accurately analyzed using ASTRAL-II. For example, we are currently analyzing the next of 1KP dataset that includes 400 genes, but more than 1,100 species.

Our biological analyses suggest that interestingly, some of the observed discrepancies between existing coalescent-based analyses and concatenation in previous studies [33] might be the result of the choice of coalescent-based method. Therefore, improved coalescent-based analyses might not only help to identify alternate relationships, but might also confirm prior hypotheses produced using concatenation.

An interesting observation was that in our simulations, concatenation was under certain conditions more accurate than ASTRAL and other summary methods. These results suggest that CA-ML should not be rejected, even though it is not statistically consistent. Conversely, proofs of consistency of standard summary methods assume gene trees estimated without error [147], and this assumption limits the relevance of consistency results in practice.

Our analyses also highlighted a problem that we addressed in Chapter 4: gene tree estimation error can affect the species tree, and that the accuracy of summary methods is depended on the accuracy of gene trees. This results in an interesting question: can the statistical binning approach also improve the accuracy of ASTRAL? Our preliminary results suggest that the answer is yes. We analyzed the avian simulated dataset presented in the previous chapter and observed that 1) ASTRAL-II has better accuracy than MP-EST on this dataset, and 2) binning used with ASTRAL-II further improved its accuracy for many model conditions (see results in Fig. 5.26 and see [191] for more). We also noted some interesting cases (e.g., the 1000bp model condition in Fig. 5.26) where ASTRAL, unlike MP-EST, did not improve using binning, but with or without binning ASTRAL had better accuracy than MP-EST. Nevertheless, our results make it clear that the use of all summary methods, including ASTRAL should be with the understanding that gene tree error can impact their results, and that practitioners need to make an effort to obtain the best gene trees possible using their data. The requirement to use

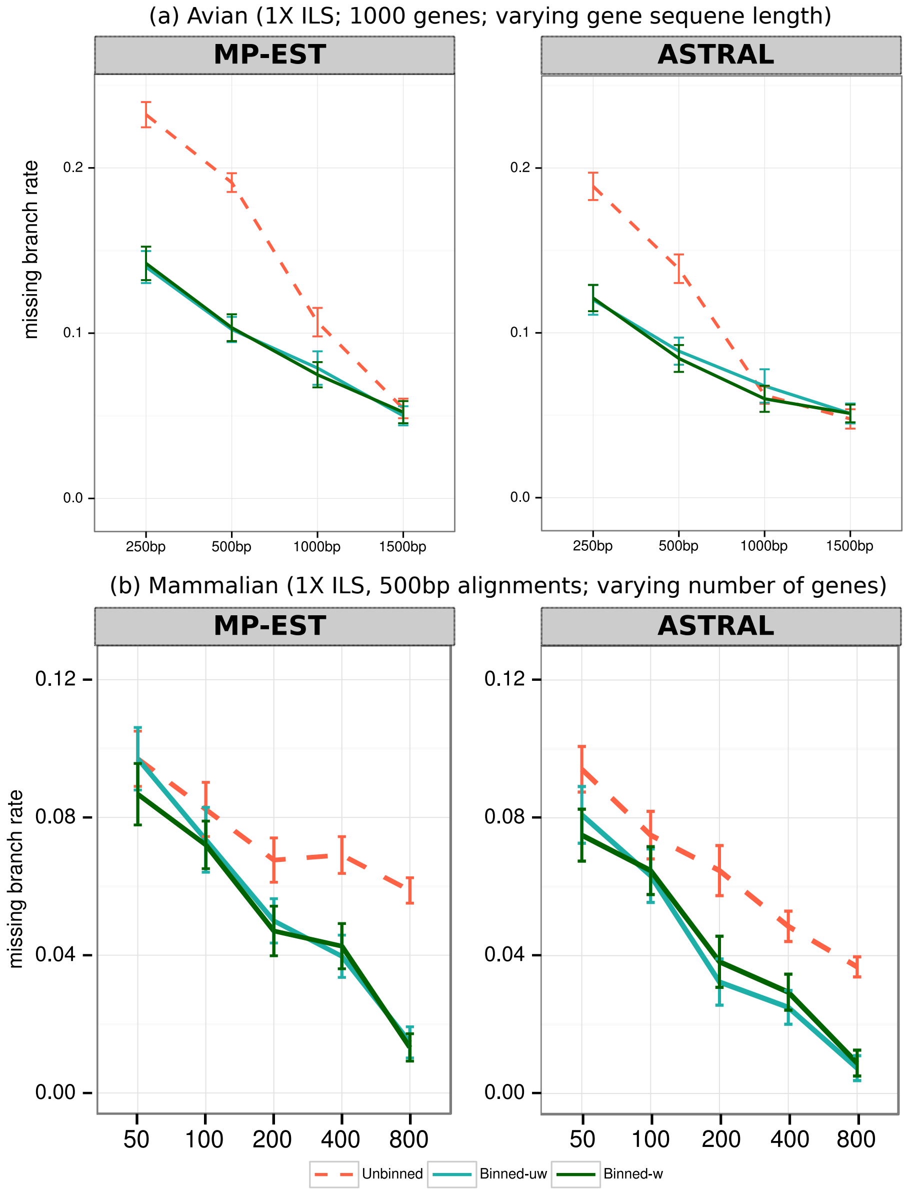


Figure 5.26: **Impact of binning on ASTRAL**. We compare weighted and unweighted statistical binning when run using MP-EST or ASTRAL-II as the summary method on simulated (a) avian and (b) mammalian datasets ($\mathcal{S}$ = 50% for avian and $\mathcal{S}$ = 75% for mammalian). ASTRAL, just like MP-EST, is improved in terms of accuracy when used with binned supergene trees. Also note that ASTRAL has lower error than MP-EST with or without binning, except with the longest sequences.

recombination-free regions complicates this pursuit as recombination-free "c-genes" can be very short, especially as the number of taxa increases [34]. Future work is needed to study the impact of using shorter gene sequence alignments, and conversely the presence of recombination events within genes.

Several limitations in ASTRAL need to be addressed in future work.

Comparison to other types of methods: While we compared ASTRAL to simple summary methods, future studies need to compare ASTRAL-II to boosting approaches (e.g., [153, 236]) that enable slower coalescent-based methods to scale to large datasets. Also, the running time of NJst and other simple distance-based methods that we didn't analyze here (e.g., STAR [142] and GLASS [144]) might be improved if better implementation of them is produced. Finally, a comparison to co-estimation methods under conditions where those methods can run (e.g., small numbers of species and genes) would also be interesting.

Missing data: We presented algorithms for handling incomplete gene trees. However, we have not rigorously studied the effect of incomplete gene trees on the accuracy of ASTRAL. A more comprehensive study needs to test the accuracy of ASTRAL in the presence of incomplete gene trees. These studies would be most interesting if they also include cases where missing data are not randomly distributed throughout the tree (e.g., basal taxa could be missing more often). While the optimization problem of ASTRAL is likely sufficient even when there are missing taxa, whether our current construction of set $\mathcal{X}$ from a set of incomplete genes is sufficient remains to be tested.

Multiple individuals: In studies where closely related species are analyzed, it is believed that sampling more than one individual per species can help in resolving the relationships [37, 151]. The optimization problem in ASTRAL can be easily extended to cases where multiple individuals are sampled from each species. Once again, computing the set $\mathcal{X}$ requires more care when multiple individuals are present, and future algorithmic developments are needed to obtain good accuracy on such datasets.

Further running time improvements: Further improvements to the running time of ASTRAL can be potentially obtained. For example, currently, in our traversal of gene trees, we do not exploit similarities between gene trees. If two gene trees are identical, we can traverse only one of them and simply count the resulting score twice. Taking this idea one step further would allow us to find commonalities between gene trees, and to exploit those commonalities to reduce the computational burden.

Chapter 6

Conclusions and Future Work

Evolutionary studies are increasingly relaying on large-scale data now that sequencing has become relatively cheap. In this dissertation, we addressed three challenges arising in analyses of large-scale datasets for evolutionary studies: multiple sequence alignments (MSA) of ultra-large datasets, gene tree estimation error and how it impacts reconstructing species trees using summary methods, and finally, the scalability and accuracy of summary methods. The MSA challenge has implications in many areas of biological studies and relates to increases in the number of sequences for a particular gene. The next two challenges are related to the problem of reconstructing species phylogenies in the presence of gene tree discordance due to ILS; hence, both arise with increases in the number of genes sampled across the genome (potentially from a large number of species). All three challenges are faced in the pipeline that starts from raw sequences and outputs a species phylogeny. We first give a quick summary of each of the three contributions, then present some directions for future research, and finish by some concluding remarks.

6.1 Summary

PASTA: We showed that few existing MSA methods can run on ultra-large datasets, i.e., those with many tens of thousands to even a million sequences. Those methods that could run typically had degraded accuracy, especially when datasets also had high rates of evolution. We introduced PASTA, a new algorithm for co-estimation of alignments and trees. PASTA is built on SATé [30, 31], and just like SATé, it divides sequences into subsets using a guide tree, obtains alignments for each subset, merges alignments, and then estimates a tree from the alignment; it repeats this process until some stopping criterion is met. The main improvement of PASTA over SATé is in the merge step. Unlike SATé, which aligned alignments using external alignment merging tools, PASTA combines alignments using a combination of building a spanning tree, pairwise mergers of alignments using external tools, and application of transitivity. We showed that the running time of our merging strategy is $O(n \log n)$ for n sequences, and demonstrated the scalability of the method empirically as well. Furthermore, we showed in simulation and biological studies that PASTA had better accuracy than competing methods on most nucleotide and amino acid datasets. We were able to align a dataset with a million sequences in two weeks of running time, and achieved high accuracy. Thus, accurate alignment of ultra-large datasets is possible.

Statistical binning: We showed that when large numbers of genes are sampled from across the genome for reconstructing the species phylogeny, many of these genes are typically short and uninformative regarding the true topology of their respective gene trees. We thoroughly demonstrated this problem in simulations, and on the avian biological dataset, among others. We also showed that high levels of estimation error in gene trees translate to high levels of error in the estimated species tree. We proposed the statistical binning approach for re-estimating the gene trees by grouping them together. Binning divides the set of genes into bins such that no two genes in the same bin have any detected strong conflict. Sequence data from genes binned together are concatenated, and these are used to estimate a set of supergene trees, which are then used as input to a summary method. In our simulation studies, gene tree estimation error, species tree topological error, and species tree branch length error were all reduced using binning, and branch support values were improved. We introduced two versions of binning, one with and one without weighting bins by their size. We proved that weighted statistical binning is statistically consistent under the multi-species coalescent (MSC) model if we allow the number of genes and the number of sites per gene to both increase, but unweighted binning is not consistent under those assumptions.

ASTRAL: Summary methods used to estimate a species tree from a collection of gene trees are relatively new. We showed that existing summary methods either simply do not scale to datasets that are large in terms of both the number of genes and the number of species, or have reduced accuracy for large datasets; moreover, even on moderate size datasets, the accuracy of summary methods had room for improvement. We introduced ASTRAL, a new summary method, which finds the species tree that agrees with the largest number of induced quartet trees form the gene trees. We showed that the solution to this problem is statistically consistent under the MSC model. ASTRAL solves this problem using dynamic programming, and also solves a constrained version of the problem where the species tree bipartitions are restricted to those in the gene trees (and in ASTRAL-II, some additional bipartitions that we heuristically compute). We showed that with increased number of genes, the constrained version of ASTRAL also converges in probability to the true species tree, and is therefore statistically consistent. We demonstrated scalability and accuracy of ASTRAL on a large set of simulated datasets. On biological datasets, we demonstrated that the comparison between ILS-aware summary methods and concatenation depends on which summary methods is used, and some of the results obtained using concatenation and rejected by previous summary methods are recovered using ASTRAL.

6.2 Future directions

Our three main contributions address *some* challenges of analyzing large datasets in evolutionary studies, but many such challenges remain. We pointed some directions of future work for each of these three approaches in their respective chapters. Here we point out some additional directions for future work.

Systematic bias: We explored gene tree estimation error arising from insufficient phylogenetic signal in the gene sequences; however, gene tree estimation error can also come from poorly estimated alignments (see Chapter 3) or systematic errors introduced during the tree inference [111, 274]. These sources of error usually arise from imperfect modeling of sequence evolution processes, and can lead to estimated gene trees that are positively misleading. Since our studies focused on insufficient phylogenetic signal, we have no evidence that statistical binning or ASTRAL could reduce phylogenetic error due to alignment error or misspecification for the sequence evolution model. Consequently, appropriate care should be devoted to obtaining good alignments and choosing an adequate model of sequence evolution to reconstruct both gene and supergene trees. Future studies should evaluate performance of ASTRAL and statistical binning when at least some of the genes have properties that cause bias (e.g., unbalanced GC content that violates stationarity assumptions of GTR [39]). A central question is whether the summary method pipeline or concatenation would work better in the presence of systematic biases.

Multiple sources of discordance: Throughout Chapters 4 and 5, we only considered ILS as a source of discord between true gene trees and the species tree. As we discussed in Section 2.2, biological discordance can also be due to other factors (e.g., duplication and loss, incorrect orthology assessments, recombination, introgression, horizontal gene transfer, and hybridization). We don't have evidence that either binning or ASTRAL helps when discordance is due to some of these other processes. A consequence of simulating only ILS in our studies is that our simulations should favor ILS-aware summary methods (such as ASTRAL and MP-EST) that are based on the same model used for simulations over concatenation (which assumes no ILS is present). Given this, the fact that unbinned MP-EST and even ASTRAL are less accurate than concatenation under some conditions is interesting. Future studies based on model conditions in which other sources of gene tree discord are included would enable a better understanding of the relative accuracy of concatenation and coalescent-based species tree estimation, and the impact of using binning and ASTRAL under those conditions. For example, it would be very interesting to see if ASTRAL performs well when horizontal gene transfer and ILS act simultaneously to create gene tree discordance.

Variations of the species tree estimation pipeline: Throughout the thesis, we used only maximum likelihood for estimating gene trees, and only a handful of summary methods for estimating the species tree. Other variations of the pipeline might lead to different patterns of performance. For example, gene trees could be estimated using Bayesian methods instead of maximum likelihood, and some recent studies suggest these result in improved accuracy for the species tree [239]. If Bayesian methods are used, the input to ASTRAL can be a distribution on each gene tree, and not a single tree. These specific variants might improve species tree estimations but would also result in substantially increased running time. Finally, we did not compare our methods with pipelines other than summary methods and concatenation. There are alternative pipelines such as gene tree species tree co-estimation [150, 151] and species tree estimation directly from data without computing gene trees [156, 157]. Co-estimation methods have prohibitive running time; however, attempts to improve the scalability of co-estimation methods are underway, some by us [153], and such attempts may enable running co-estimation methods on larger datasets. Methods for direct estimation of species tree without gene trees are new and some are limited to specific types of data. Future work needs to evaluate these methods thoroughly.

Parallelization: We utilized parallelization throughout this dissertation in simple forms. In PASTA, we run different alignment and merge jobs on different threads, producing plenty of parallelism. However, the tree estimation step is not well-parallelized inside PASTA, and future work can look into creative ways of improving the parallelization in the tree estimation step (e.g., through approaches similar to DACTAL [275]). In binning and ASTRAL, we exploit parallelization only in the sense that independent parts of the pipeline are ran independently. But much more can be done. Even though ASTRAL is fast on fully binary gene trees, its running time can be prohibitive when a very large number of multifurcating gene trees is available. Since the theoretical running time of ASTRAL might not be improvable for unresolved gene trees, the use of parallelism can enable analyses that would otherwise be intractable. The use of GPUs in particular seems promising for ASTRAL.

6.3 Conclusions

All tools and datasets presented in this dissertation are publicly available in open source. We wish that our contributions would advance biological evolutionary studies with large-scale datasets. In the short time since we published these methods, new studies have started using them, and some biologists have even published results using these methods (e.g., see [276–284]). These are in addition to the biological studies that we have published using these methods (e.g., [39, 40]), and others that we are currently analyzing (e.g., next phases of both avian and 1KP projects, and other datasets on mammals, raptors, hummingbirds, and others).

The work presented in this dissertation demonstrated that analyzing large-scale sequence datasets is possible, but it requires developing new methods that can scale while maintaining accuracy as the size of the datasets grows. All three methods presented here increase the set of datasets that can be analyzed accurately. PASTA and ASTRAL enable accurate analyses of large numbers of species, and binning enables using low signal genes that previously were discarded routinely from summary method analyses. Thus, with our new methods, more of the data can be analyzed and the need to data filtering is reduced. We believe our future research needs to address issues that we have not addressed here, but with a similar goal: developing scalable and accurate methods that enable analyses of large-scale data without extensive filtering.

Appendices

Appendix A

Commands

Here, we give the exact commands used in various analyses presented throughout this dissertation.

A.1 PASTA

A.1.1 Method commands

- **Muscle version 3.8.31**:

  ```
  muscle -in [input_sequences] -out [output_alignment] <-maxiters 2>*
  ```

  (*Only for datasets with more than 3,000 sequences.)

- **Clustal-Omega version 1.2.0**:

  ```
  clustalo --threads=12 -i [input_sequences] -o [output_alignment]
  ```

- **HMMBUILD version 3.0**:

  ```
  hmmbuild --symfrac 0.0 --dna [output_profile] [backbone_alignment]
  ```

- **HMMALIGN version 3.0**:

  ```
  hmmalign [--dna | --rna | --amino] [output_profile] [query_file] > [output_alignment]
  ```

- **Mafft default version 7.143b**:

  ```
  mafft --ep 0.123 --auto --anysymbol --thread 12 [input_sequences] > [output_alignment]
  ```

- **Mafft-LNSI version 7.143b**:

  ```
  mafft --ep 0.123 --localpair --maxiterate 1000 --quiet --anysymbol --thread 12 [input_sequences] > [output_alignment]
  ```

- **Mafft-PartTree version 7.143b**:

  ```
  mafft --ep 0.123 --partsize 1000 --retree 2 --parttree --quiet --anysymbol --thread 12 [input_sequences] > [output_alignment]
  ```

- **FastTree version 2.1.5 SSE3**:

  ```
  fasttree [-nt -gtr]* [input_fasta] > [output_tree]
  ```

  (*Only for nucleotide datasets.)

- **RAxML version 7.5.7**:

  ```
  raxmlHPC-PTHREADS -T 12 -m PROTGAMMA[model] -j -n [output_name] -s [input_fasta] -p 1
  ```

- **SATé version 2.2.7**:

  ```
  python run_sate.py config.sate2.txt
  ```

  (The **config.sate2.txt** file is defined as follows:)

  ```
  [commandline]
  ```

```
two_phase = False
datatype = <dna, rna, or protein>
untrusted = False
multilocus = False
input = <input_fasta>
treefile = <starting_tree>
aligned = False
raxml_search_after = False
auto = False

[fasttree]
model = -gtr
args =
options = -nosupport -fastest

[sate]
time_limit = -1.0
iter_without_imp_limit = -1
time_without_imp_limit = -1.0
break_strategy = centroid
start_tree_search_from_current = True
blind_after_iter_without_imp = -1
max_mem_mb = 4024
```

```
blind_mode_is_final = True

blind_after_time_without_imp = -1.0

max_subproblem_size = 200

merger = muscle

num_cpus = 12

after_blind_time_without_imp_limit = -1.0

max_subproblem_frac = 0.0

blind_after_total_time = -1.0

after_blind_time_term_limit = -1.0

aligner = mafft

iter_limit = 3

blind_after_total_iter = -1

tree_estimator = fasttree

after_blind_iter_term_limit = -1

return_final_tree_and_alignment = False

move_to_blind_on_worse_score = True

after_blind_iter_without_imp_limit = -1
```

- **PASTA version 1.1.0**:

  run_pasta.py -i input.fasta -t starting.tree

  (used this version for all nucleotide results, except Indelible; for stating.tree, used the approach described in the paper, with the backbone alignment estimated using SATé.)

- **PASTA version 1.5.1**:
  run_pasta.py -i input.fasta
  (used for all AA results and Indelible)

Notes:

- PASTA versions 1.1.0 and 1.5.1 were algorithmically identical, but used different versions of internal tools. PASTA 1.1.0 internally used version 6.903 of Mafft for aligning subsets and version 1.0.2 of OPAL for pairwise merges. In PASTA version 1.5.1, OPAL has moved to version 2.1.2 and Mafft was moved to version v7.149b.
- Version 1.1.0 of PASTA did not estimate the starting tree internally, and so we gave PASTA the starting tree that we computed separately. PASTA version 1.5.1 internally estimates the starting tree. The starting tree provided to PASTA in version 1.1.0 used the approach we describe in the paper and uses SATé for estimating the backbone alignment on 100 randomly selected sequences. PASTA version 1.5.1 uses a similar technique, but uses Mafft for estimating the backbone alignment on 100 randomly selected techniques.

A.1.2 Indelible control files

Here is the `control.txt` file used for Indelible simulations of the 10K Indelible dataset with 10000M2 condition

```
//////////////////////////////
[TYPE] NUCLEOTIDE 2
/*  ---------------------------------------------
         FROM ||     T    |    C     |    A     |    G
        ------++----------+----------+----------+----------
          T   ||     -    |  a Pi_C  |  b Pi_A  |  c Pi_G
          C   ||  a Pi_T  |     -    |  d Pi_A  |  e Pi_G
          A   ||  b Pi_T  |  d Pi_C  |     -    |  f Pi_G
          G   ||  c Pi_T  |  e Pi_C  |  f Pi_A  |     -
A-C 1.24284
A-G 3.47484
A-T 0.48667
C-G 1.07118
C-T 4.38510
G-T 1.0   */

[MODEL] GTRexample
  [submodel]  GTR 1.2619573850882344 0.14005536945585983 0.2877830346145434
                                0.35766826674033914 0.3082674310184066
                                //  GTR: a=0.2, b=0.4, c=0.6, d=0.8, e=1.2, f=1
  [statefreq]  .311475 .191363 .300414 .196748 //  T=0.1, C=0.2, A=0.3, G=0.4
  [rates]     0 1 0           //   continuous gamma with alpha=1
  [indelmodel] USER m_indel_model.txt //custom model; see below
  [indelrate]    0.0001          // insertion rate = deletion rate = 0.1
                                  // relative to average substitution rate of 1.

[TREE] tree1
  [unrooted] 10000 6.7 2.5 1 0.24  // ntaxa birth death sample mut
  [treedepth] 5
  [seed] 1

[PARTITIONS] pGTR  [tree1 GTRexample 1000]

[SETTINGS]
[output] FASTA

[EVOLVE]
  pGTR 10 GTRout
////////////////////////////
```

The control files for 10000M3 and 10000M4 are similar, with differences only in the following parts:

10000M3:

```
[TREE] tree1
  [unrooted] 10000 6.7 2.5 1 0.24  // ntaxa birth death sample mut
  [treedepth] 2.5
  [seed] 1
```

10000M4:

```
[TREE] tree1
  [unrooted] 10000 6.7 2.5 1 0.06  // ntaxa birth death sample mut
  [treedepth] 1
  [seed] 1
```

The `m_indel_model.txt` file is the same for all three model conditions, and is based on values used in [30].

`m_indel_model.txt`:

```
0.2012
0.1600
0.1280
0.1024
0.0819
0.0655
0.0524
0.0419
0.0336
0.0268
0.0215
0.0172
0.0137
0.0110
0.0088
0.0070
0.0056
```

```
0.0045
0.0036
0.0029
0.0023
0.0018
0.0015
0.0012
0.0009
0.0008
0.0006
0.0005
0.0004
0.0003
0.0002
```

A.2 Binning

A.2.1 Simulations

The parameters of bppseqgen for our simulations were:

- The substitution model parameters (GTR parameters):

  $a = 1.062409952497, b = 0.133307705766, c = 0.195517800882,$

  $d = 0.223514845018, e = 0.294405416545,$

  $\theta = 0.469075709819, \theta 1 = 0.558949940165, \theta 2 = 0.488093447144$

- The rate distribution parameters (Gamma parameters):

  $n = 4, \alpha = 0.370209777709$

The McCoal control file for simulations of the 10-taxon species tree is:

```
SimulatedData.txt
```

```
1234567
10 A B C D E F G H I J
  1 1 1 1 1 1 1 1 1 1
((((((((((A #.05,B #.05):0.005 #.05,C #.05):0.01 #.05,
D #.05):0.015 #.05, E #.05):0.02 #.05,F #.05):0.025 #.05,
G #.05):0.03 #.05,H #.05):0.035 #.05,I #.05):0.04 #.05,
J #.05):0.54 #.05;
```

A.2.2 Methods

A.2.2.1 Estimating ML gene trees

We used RAxML version 7.3.5 [120] to estimate gene trees.

Maximum likelihood trees:

```
raxmlHPC-SSE3 -m GTRGAMMA
-s [input_MRP_file] -n [a_name] -N 20
-p [random_seed_number]
```

Bootstrapping:

```
raxmlHPC-SSE3 -m GTRGAMMA
-s [input_MRP_file] -n [a_name] -N 200
-p [random_seed_number] -b [random_seed_number]
```

A.2.2.2 MP-EST

MP-EST version 1.0.3 was used in all runs. We used a custom shell script to run MP-EST 10 times with different random seed numbers and take the tree with the highest likelihood. For estimating branch length on a fixed topology (used in the simulation procedure) we used version 1.0.4 of MP-EST.

A.2.2.3 MRP

MRP data matrices are built using a custom Java program available at https://github.com/smirarab/mrpmatrix. The following command was used to create the MRP matrix.

```
 java -jar mrp.jar [input_file] [output_file] NEXUS
```

The default heuristic in PAUP* (v. 4. 0b10) [107] was used for solving the parsimony problem. This heuristic operates by first generating an initial tree through random sequence addition and then using Tree Bisection and Reconnection (TBR) moves to reach a local optimum. 1000 iterations are used, and the most parsimonious tree is returned. When multiple trees have the same maximum parsimony score, the greedy consensus of those trees is returned. The following shows the PAUP* commands used.

```
 begin paup;
set criterion=parsimony maxtrees=1000
increase=no;
hsearch start=stepwise addseq=random
nreps=100 swap=tbr;
filter best=yes;
savetrees file = <treeFile> replace=yes
format=altnex;
```

```
contree all/ strict=yes
treefile = <strictConsensusTreeFile>
replace=yes;
tcontree all/ majrule=yes strict=no
treefile = <majorityConsensusTreeFile>
replace=yes;
contree all/ majrule=yes strict=no
le50=yes
treefile = <greedyConsensusTreeFile>
replace=yes;
log stop;
quit; end;
```

A.2.2.4 Greedy

We use Dendropy version 3.12.0 [204] to compute the greedy consensus tree.

A.3 ASTRAL

A.3.1 ASTRAL-I analyses

A.3.1.1 Gene tree estimation

RAxML version 7.3.5 [120] was used to estimate gene trees. The following command was used for estimating the best ML trees.

```
 raxmlHPC-SSE3 -m GTRGAMMA -s [input_file] -n [a_name]
-N 20 -p [random_seed_number]
```

The following command was used for bootstrapping.

```
 raxmlHPC-SSE3 -m GTRGAMMA -s [input_file] -n [a_name] -N 200
-p [random_seed_number] -b [random_seed_number]
```

A.3.1.2 ASTRAL

We ran version 3.1.1 of ASTRAL (corresponding to the github commit `fb21c0ce6140e9e238575356bc174c88c6cfc597` from March 6th on `https://github.com/smirarab/ASTRAL` with the following command:

```
java -jar astra_3.1.1.jar -wq -in [input_tree]
```

Where the exact version of ASRAL was used, we ran it with the following command:

```
java -jar astra_3.1.1.jar -wq -in [input_tree] -xt
```

To add new bipartitions to $\mathcal{X}$, we used it with the following command:

```
java -jar astra_3.1.1.jar -wq -in [input_tree] -ex [extra_trees]
```

A.3.1.3 BUCKy-population

We ran BUCKy with the default settings, except for the number of generations that we changed from 100K to one million. The following command was used to run BUCKy.

```
bucky -n <numberOfGenerations> -o <outputFileRoot> <inputFiles>
```

A.3.1.4 MRP and MRL

MRP trees are built using a custom Java program available at `https://github.com/smirarab/mrpmatrix`. The following command was used to create the MRP matrix.

```
 java -jar mrp.jar [input_file] [output_file] NEXUS
```

We used the default heuristic in PAUP* (v. 4. 0b10) [107] for maximum parsimony. This heuristic first generates an initial tree through random sequence addition and then uses Tree Bisection and Reconnection (TBR) moves to reach a local optimum. This process is repeated 1000 times, and the most parsimonious tree is returned. When multiple trees have the same maximum parsimony score, the greedy consensus of those trees is returned. The following shows the PAUP* commands used.

```
 begin paup;
set criterion=parsimony maxtrees=1000
increase=no;
hsearch start=stepwise addseq=random
nreps=100 swap=tbr;
filter best=yes;
savetrees file = <treeFile> replace=yes
format=altnex;
contree all/ strict=yes
treefile = <strictConsensusTreeFile>
replace=yes;
tcontree all/ majrule=yes strict=no
treefile = <majorityConsensusTreeFile>
replace=yes;
contree all/ majrule=yes strict=no
le50=yes
treefile = <greedyConsensusTreeFile>
replace=yes;
log stop;
quit; end;
```

MRL stands for "Matrix Representation with Likelihood", and is the supertree method obtained by running two-state symmetric maximum likelihood on the MRP matrix [238]. We computed maximum likelihood trees on the same MRP matrix using RAxML under the two-state maximum likelihood model, to obtain MRL (matrix representation with likelihood) trees.

A.3.1.5 Concatenation

We used RAxML version 7.3.5 to create the parsimony starting trees:

```
 raxmlHPC-SSE3 -y -s supermatrix.phylip -m GTRGAMMA
-n [a_name] -p [random_seed_number] -s [alignment]
```

We then used RAxML-light version 1.0.6 with the following command to search for the ML tree.

```
 raxmlLight-PTHREADS -T 4 -s supermatrix.phylip -m GTRGAMMA -n name
-t [parsimony_tree] -s [alignment]
```

A.4 ASTRAL-II

A.4.1 SimPhy parameters

We used the following parameters in our simulation using SimPhy. The scripts for the simulation are given at `http://www.cs.utexas.edu/users/phylo/software/astral/`.

Table A.1: **Parameters used in SimPhy simulations**.

Arg.	Description	Value	Notes
RS	number of replicates	50	
RL	number of loci	1000	
RG	number of genes	1	no duplications
ST	maximum tree length	500K, 2M, or 10M	
SI	number of individuals per species	1	
SL	number of leaves	10,50,100,200,500, or 1000	
SB	birth rates	0.000001, 0.0000001	
P	global population sizes	200000	
HS	Species-specific branch rate heterogeneity modifiers	Log normal (1.5,1)	
HL	Locus-specific rate heterogeneity modifiers	Log normal (1.2,1)	
HG	Gene-tree-branch-specific rate heterogeneity modifiers	Log normal (1.4,1)	
U	Global substitution rate	Exponential (10000000)	
SO	Outgroup branch length relative to half the tree length	1	
CS	Random number generator seed	293745	

A.4.2 Indelible parameters

We used a perl script available also at `http://www.cs.utexas.edu/users/phylo/software/astral/` to draw parameters for the Indelible simulations. For each replicate, some hyperparameters are first drawn and these hyperparameters affect how the actual parameters are drawn for each gene in that replicate.

Gene Length: The alignments lengths are drawn from log normal distributions for genes of each replicate. For each replicate, a hyperparameter controls the two model parameters of the log normal distribution. 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 frequencies: We used a Dirichlet(36,26,28,32) to draw the base frequencies for A, C, G, and T. These values were calculated using maximum likelihood estimation form a collection of three large scale multi-locus datasets: 1KP dataset, Song et al Mammalian dataset, and Avian phylogenomics dataset. The base values used for this maximum likelihood estimation and the corresponding scripts are available at `http://www.cs.utexas.edu/~phylo/software/astral/`.

Substitution matrices: As with base frequencies, GTR matrices were drawn from a Dirichlet(16,3,5,5,6,15) and these parameters were also estimated using maximum likelihood from our empirical data.

Rates-across-sites shape parameter: α was drawn from an exponential distribution with rate 1.2, with values below 0.1 discarded. Like rates and base frequencies, these values were estimated from real data.

Bibliography

[1] Charles Darwin. *The origin of species by means of natural selection.* J. Murray, 1872.

[2] Ernst Mayr. *What evolution is.* Basic books, 2001.

[3] Douglas L Theobald. A formal test of the theory of universal common ancestry. *Nature*, 465(7295):219–222, 2010.

[4] Michael Steel and David Penny. Origins of life: Common ancestry put to the test. *Nature*, 465(7295):168–169, 2010.

[5] David M Hillis, Craig Moritz, and Barbara K Mable, editors. *Molecular Systematics*, volume 2nd. Sinauer Associates, 1996.

[6] Jerry A Coyne and H Allen Orr. *Speciation.* Sinauer Associates, 2004.

[7] Joseph Felsenstein. *Inferring phylogenies.* Sunderland, 2003.

[8] C. Randal Linder and Tandy Warnow. An overview of phylogeny reconstruction. In S. Aluru, editor, *Handbook of Computational Biology.* Chapman & Hall, 2005.

[9] Ziheng Yang and Bruce Rannala. Molecular phylogenetics: principles and practice. *Nature Reviews Genetics*, 13(5):303–314, 2012.

[10] David R. Maddison, Katja Sabine Schulz, and Wayne P. Maddison. The tree of life web project. *Zootaxa*, 1668:19–40, 2007.

[11] Laura Bonetta. Whole-Genome sequencing breaks the cost barrier. *Cell*, 141(6):917–919, 2010.

[12] Antonis Rokas, Barry L Williams, Nicole King, and Sean B Carroll. Genome-scale approaches to resolving incongruence in molecular phylogenies. *Nature*, 425(6960):798–804, 2003.

[13] Shannon M Hedtke, Ted M Townsend, and David M Hillis. Resolution of phylogenetic conflict in large data sets by increased taxon sampling. *Systematic Biology*, 55(3):522–9, 2006.

[14] Michael J. Sanderson and Amy C. Driskell. The challenge of constructing large phylogenetic trees. *Trends in Plant Science*, 8(8):374–379, 2003.

[15] Kevin Liu, C. Randal Linder, and Tandy Warnow. Multiple Sequence Alignment a major challenge to large-scale phylogenetics. *PLoS Currents: Tree of Life*, 2:RRN1198, 2010.

[16] John E McCormack, Michael G. Harvey, Brant C. Faircloth, Nicholas G Crawford, Travis C. Glenn, and Robb T. Brumfield. A Phylogeny of Birds Based on Over 1,500 Loci Collected by Target Enrichment and High-Throughput Sequencing. *PLoS ONE*, 8(1):e54848, 2013.

[17] Md. Shamsuzzoha Bayzid and Tandy Warnow. Naive binning improves phylogenomic analyses. *Bioinformatics*, 29(18):2277–84, 2013.

[18] Wayne P. Maddison. Gene trees in species trees. *Systematic Biology*, 46(3):523–536, 1997.

[19] Roderic D. M. Page and M A Charleston. From gene to organismal phylogeny: reconciled trees and the gene tree/species tree problem. *Molecular Phylogenetics and Evolution*, 7(2):231–240, 1997.

[20] James H. Degnan and Noah A. Rosenberg. Gene tree discordance, phylogenetic inference and the multispecies coalescent. *Trends in Ecology and Evolution*, 24(6):332–340, 2009.

[21] Roy J Britten. Divergence between samples of chimpanzee and human DNA sequences is 5%, counting indels. *Proceedings of the National Academy of Sciences of the United States of America*, 99(21):13633–13635, 2002.

[22] Tarjei S. Mikkelsen, LaDeana W. Hillier, Evan E. Eichler, Michael C. Zody, David B. Jaffe, Shiaw-Pyng Yang, Wolfgang Enard, Ines Hellmann, Kerstin Lindblad-Toh, Tasha K. Altheide, Nicoletta Archidiacono, Peer Bork, Jonathan Butler, Jean L. Chang, Ze Cheng, Asif T. Chinwalla, Pieter DeJong, Kimberley D. Delehaunty, Catrina C. Fronick, Lucinda L. Fulton, Yoav Gilad, Gustavo Glusman, Sante Gnerre, Tina A. Graves, Toshiyuki Hayakawa, Karen E. Hayden, Xiaoqiu Huang, Hongkai Ji, W. James Kent, Mary-Claire King, Edward J. KulbokasIII, Ming K. Lee, Ge Liu, Carlos Lopez-Otin, Kateryna D. Makova, Orna Man, Elaine R. Mardis, Evan Mauceli, Tracie L. Miner, William E. Nash, Joanne O. Nelson, Svante Pääbo, Nick J. Patterson, Craig S. Pohl, Katherine S. Pollard, Kay Prüfer, Xose S. Puente, David Reich, Mariano

Rocchi, Kate Rosenbloom, Maryellen Ruvolo, Daniel J. Richter, Stephen F. Schaffner, Arian F. A. Smit, Scott M. Smith, Mikita Suyama, James Taylor, David Torrents, Eray Tuzun, Ajit Varki, Gloria Velasco, Mario Ventura, John W. Wallis, Michael C. Wendl, Richard K. Wilson, Eric S. Lander, and Robert H. Waterston. Initial sequence of the chimpanzee genome and comparison with the human genome. *Nature*, 437(7055):69–87, 2005.

[23] Aylwyn Scally, JY Dutheil, and LaDeana W. Hillier. Insights into hominid evolution from the gorilla genome sequence. *Nature*, 483(7388):169–175, 2012.

[24] Ingo Ebersberger, Petra Galgoczy, Stefan Taudien, Simone Taenzer, Matthias Platzer, and Arndt Von Haeseler. Mapping human genetic ancestry. *Molecular Biology and Evolution*, 24(10):2266–2276, 2007.

[25] Asger Hobolth, Ole F. Christensen, Thomas Mailund, and Mikkel H. Schierup. Genomic relationships and speciation times of human, chimpanzee, and gorilla inferred from a coalescent hidden Markov model. *PLoS Genetics*, 3(2):0294–0304, 2007.

[26] David Sankoff. Minimal Mutation Trees of Sequences. *SIAM Journal on Applied Mathematics*, 28(1):35–42, 1975.

[27] Lusheng Wang and Tao Jiang. On the complexity of multiple sequence alignment. *Journal of Computational Biology*, 1(4):337–348, 1994.

[28] Ari Löytynoja. Alignment methods: Strategies, challenges, benchmarking, and comparative overview. *Methods in Molecular Biology*, 855:203–235, 2012.

[29] David J Russell, editor. *Multiple sequence alignment methods.* Springer, 2014.

[30] Kevin Liu, Sindhu Raghavan, Serita M Nelesen, C. Randal Linder, and Tandy Warnow. Rapid and Accurate Large-Scale Coestimation of Sequence Alignments and Phylogenetic Trees. *Science*, 324(5934):1561–1564, 2009.

[31] Kevin Liu, Tandy Warnow, Mark T Holder, Serita M Nelesen, Jiaye Yu, Alexandros Stamatakis, and C. Randal Linder. SATe-II: Very Fast and Accurate Simultaneous Estimation of Multiple Sequence Alignments and Phylogenetic Trees. *Systematic Biology*, 61(1):90–106, 2011.

[32] Scott V Edwards, Liang Liu, and Dennis K Pearl. High-resolution species trees without concatenation. *Proceedings of the National Academy of Sciences of the United States of America*, 104(14):5936–5941, 2007.

[33] Mark S Springer and John Gatesy. Land plant origins and coalescence confusion. *Trends in Plant Science*, 19(5):267–9, 2014.

[34] John P. Gatesy and Mark S. Springer. Phylogenetic analysis at deep timescales: Unreliable gene trees, bypassed hidden support, and the coalescence/concatalescence conundrum. *Molecular Phylogenetics and Evolution*, 80:231–266, 2014.

[35] Bojian Zhong, Liang Liu, and David Penny. The multispecies coalescent model and land plant origins: a reply to Springer and Gatesy. *Trends in Plant Science*, 19(5):270–272, 2014.

[36] L Lacey Knowles, Hayley C. Lanier, Pavel B. Klimov, and Qixin He. Full modeling versus summarizing gene-tree uncertainty: Method choice and species-tree accuracy. *Molecular Phylogenetics and Evolution*, 65(2):501–509, 2012.

[37] Swati Patel, Rebecca T Kimball, and Edward L Braun. Error in phylogenetic estimation for bushes in the tree of life. *Journal of Phylogenetics and Evolutionary Biology*, 1(2):110, 2013.

[38] Siavash Mirarab, Md. Shamsuzzoha Bayzid, and Tandy Warnow. Evaluating summary methods for multi-locus species tree estimation in the presence of incomplete lineage sorting. *Systematic Biology*, page syu063, 2014.

[39] Erich D Jarvis, Siavash Mirarab, Andre J Aberer, Bo Li, Peter Houde, Cai Li, Simon Y W Ho, Brant C. Faircloth, Benoit Nabholz, Jason T Howard, Alexander Suh, Claudia C Weber, Rute R da Fonseca, Jianwen Li, Fang Zhang, Hui Li, Long Zhou, Nitish Narula, Liang Liu, Ganeshkumar Ganapathy, Bastien Boussau, Md. Shamsuzzoha Bayzid, Volodymyr Zavidovych, Sankar Subramanian, Toni Gabaldón, Salvador Capella-Gutiérrez, Jaime Huerta-Cepas, Bhanu Rekepalli, Kasper Munch, Mikkel H. Schierup, Bent Lindow, Wesley C Warren, David Ray, Richard E Green, Michael W Bruford, Xiangjiang Zhan, Andrew Dixon, Shengbin Li, Ning Li, Yinhua Huang, Elizabeth P Derryberry, Mads Frost Bertelsen, Frederick H Sheldon, Robb T. Brumfield, Claudio V Mello, Peter V Lovell, Morgan Wirthlin, Maria Paula Cruz Schneider, Francisco Prosdocimi, José Alfredo Samaniego, Amhed Missael Vargas Velazquez, Alonzo Alfaro-Núñez, Paula F Campos, Bent Petersen, Thomas

Sicheritz-Ponten, An Pas, Tom Bailey, Paul Scofield, Michael Bunce, David M Lambert, Qi Zhou, Polina Perelman, Amy C. Driskell, Beth Shapiro, Zijun Xiong, Yongli Zeng, Shiping Liu, Zhenyu Li, Binghang Liu, Kui Wu, Jin Xiao, Xiong Yinqi, Qiuemei Zheng, Yong Zhang, Huanming Yang, Jian Wang, Linnea Smeds, Frank E Rheindt, Michael J Braun, Jon Fjeldsa, Ludovic Orlando, F Keith Barker, Knud Andreas Jø nsson, Warren Johnson, Klaus-Peter Koepfli, Stephen OBrien, David Haussler, Oliver A Ryder, Carsten Rahbek, Eske Willerslev, Gary R Graves, Travis C. Glenn, John E McCormack, Dave W Burt, Hans Ellegren, Per Alström, Scott V Edwards, Alexandros Stamatakis, David P Mindell, Joel Cracraft, Edward L Braun, Tandy Warnow, Wang Jun, M Thomas P Gilbert, and Guojie Zhang. Whole-genome analyses resolve early branches in the tree of life of modern birds. *Science*, 346(6215):1320–1331, 2014.

[40] Norman J. Wickett, Siavash Mirarab, Nam Nguyen, Tandy Warnow, Eric Carpenter, Naim Matasci, Saravanaraj Ayyampalayam, Michael S. Barker, J. Gordon Burleigh, Matthew A. Gitzendanner, Brad R. Ruhfel, Eric Wafula, Joshua P. Der, Sean W. Graham, Sarah Mathews, Michael Melkonian, Douglas E. Soltis, Pamela S. Soltis, Nicholas W. Miles, Carl J. Rothfels, Lisa Pokorny, A. Jonathan Shaw, Lisa DeGironimo, Dennis W. Stevenson, Barbara Surek, Juan Carlos Villarreal, Béatrice Roure, Hervé Philippe, Claude W. dePamphilis, Tao Chen, Michael K. Deyholos, Regina S. Baucom, Toni M. Kutchan, Megan M. Augustin, Jun Wang, Yong Zhang, Zhijian Tian, Zhixiang Yan, Xiaolei Wu, Xiao Sun, Gane Ka-Shu Wong, and James Leebens-

Mack. Phylotranscriptomic analysis of the origin and early diversification of land plants. *Proceedings of the National Academy of Sciences of the United States of America*, 111(45):E4859–E4868, 2014.

[41] J Shoshani, C P Groves, E L Simons, and G F Gunnell. Primate phylogeny: morphological vs. molecular results. *Molecular Phylogenetics and Evolution*, 5(1):102–154, 1996.

[42] Peter Andrews. Aspects of hominoid phylogeny. In C. Patterson, editor, *Molecules and morphology in evolution: conflict or compromise?*, pages 23–53. Cambridge University Press, 1987.

[43] Morris Goodman, John Czelusniak, G. William Moore, A. E. Romero-Herrera, and Genji Matsuda. Fitting the Gene Lineage into its Species Lineage, a Parsimony Strategy Illustrated by Cladograms Constructed from Globin Sequences. *Systematic Biology*, 28(2):132–163, 1979.

[44] Michael L Metzker. Sequencing technologies - the next generation. *Nature Reviews Genetics*, 11(1):31–46, 2010.

[45] Chandra Shekhar Pareek, Rafal Smoczynski, and Andrzej Tretyn. Sequencing technologies and genome sequencing. *Journal of Applied Genetics*, 52(4):413–435, 2011.

[46] Emily Moriarty Lemmon and Alan R. Lemmon. High-Throughput Genomic Data in Systematics and Phylogenetics. *Annual Review of Ecology, Evolution, and Systematics*, 44(1):99–121, 2013.

[47] Simon Tavaré. Some Probabilistic and Statistical Problems in the Analysis of DNA Sequences. *Lectures on Mathematics in the Life Sciences*, 17:57–86, 1986.

[48] William S Klug and Michael P Cummings. *Essentials of genetics.* Prentice-Hall Inc., 1999.

[49] Cédric Notredame. Recent progress in multiple sequence alignment: a survey. *Pharmacogenomics*, 3(1):131–144, 2002.

[50] Robert C. Edgar and Serafim Batzoglou. Multiple sequence alignment. *Current Opinion in Structural Biology*, 16(3):368–373, 2006.

[51] David Posada and Keith A Crandall. Intraspecific gene genealogies: Trees grafting into networks. *Trends in Ecology and Evolution*, 16(1):37–45, 2001.

[52] J Peter Gogarten and Jeffrey P Townsend. Horizontal gene transfer, genome innovation and evolution. *Nature Reviews Microbiology*, 3(9):679–687, 2005.

[53] Luis Boto. Horizontal gene transfer in evolution: facts and challenges. *Proceedings of the Royal Society of London B: Biological Sciences*, 277(1683):819–827, 2010.

[54] Luay Nakhleh. Evolutionary Phylogenetic Networks: Models and Issues. *Networks*, pages 125–158, 2011.

[55] Daniel H. Huson, Regula Rupp, and Celine Scornavacca. *Phylogenetic networks: concepts, algorithms and applications.* Cambridge University Press, 2010.

[56] Roderic D. M. Page. Maps Between Trees and Cladistic Analysis of Historical Associations among Genes,Organisms, and Areas. *Systematic Biology*, 43(1):58–77, 1994.

[57] Matthew D Rasmussen and Manolis Kellis. Unified modeling of gene duplication, loss, and coalescence using a locus tree. *Genome Research*, 22(4):755–765, 2012.

[58] Walter M. Fitch. Homology. *Trends in Genetics*, 16(5):227–231, 2000.

[59] Maido Remm, Christian E Storm, and Erik L Sonnhammer. Automatic clustering of orthologs and in-paralogs from pairwise species comparisons. *Journal of Molecular Biology*, 314(5):1041–1052, 2001.

[60] Feng Chen, Aaron J Mackey, Jeroen K Vermunt, and David S Roos. Assessing performance of orthology detection strategies applied to eukaryotic genomes. *PLoS ONE*, 2(4):e383, 2007.

[61] Bengt Sennblad and Jens Lagergren. Probabilistic Orthology Analysis. *Systematic Biology*, 58(4):411–424, 2009.

[62] Nicholas H. Barton. Why sex and recombination? In *Cold Spring Harbor Symposia on Quantitative Biology*, volume 74, pages 187–195, 2009.

[63] David Posada, Keith A Crandall, and Edward C Holmes. Recombination in evolutionary genomics. *Annual Review of Genetics*, 36:75–97, 2002.

[64] Joshua Lederberg and Edward L Tatum. Gene recombination in Escherichia coli. *Nature*, 158(4016):558, 1946.

[65] Jeff J Doyle. Trees within trees: genes and species, molecules and morphology. *Systematic Biology*, 46(3):537–553, 1997.

[66] Richard R Hudson. Gene genealogies and the coalescent process. *Oxford Surveys in Evolutionary Biology*, 7(1):1–44, 1990.

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

[68] Simon Tavaré. Line-of-descent and genealogical processes, and their applications in population genetics models. *Theoretical Population Biology*, 26(2):119–164, 1984.

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

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

[71] Elizabeth S. Allman, James H. Degnan, and John A Rhodes. Identifying the rooted species tree from the distribution of unrooted gene trees under the coalescent. *Journal of Mathematical Biology*, 62:833–862, 2011.

[72] James H. Degnan and Noah A. Rosenberg. Discordance of species trees with their most likely gene trees. *PLoS Genetics*, 2(5):e68, 2006.

[73] Stephen Jay Gould and Niles Eldredge. Punctuated equilibria: the tempo and mode of evolution reconsidered. *Paleobiology*, pages 115–151, 1977.

[74] Noah A. Rosenberg. Discordance of species trees with their most likely gene trees: a unifying principle. *Molecular Biology and Evolution*, 30(12):2709–2713, 2013.

[75] Alan Feduccia. 'Big bang' for tertiary birds? *Trends in Ecology and Evolution*, 18:172–176, 2003.

[76] Shannon J Hackett, Rebecca T Kimball, Sushma Reddy, Rauri C K Bowie, Edward L Braun, Michael J Braun, Jena L Chojnowski, W Andrew Cox, Kin-Lan Han, John Harshman, Christopher J Huddleston, Ben D Marks, Kathleen J Miglia, William S Moore, Frederick H Sheldon, David W Steadman, Christopher C Witt, and Tamaki Yuri. A phylogenomic study of birds reveals their evolutionary history. *Science*, 320(5884):1763–1768, 2008.

[77] Sen Song, Liang Liu, Scott V Edwards, and Shaoyuan Wu. Resolving conflict in eutherian mammal phylogeny using phylogenomics and the multispecies coalescent model. *Proceedings of the National Academy of Sciences of the United States of America*, 109(37):14942–7, 2012.

[78] John Wiens. The role of morphological data in phylogeny reconstruction. *Systematic Biology*, 53(4):653–661, 2004.

[79] Steven Dodsworth, Mark W Chase, Laura J Kelly, Ilia J Leitch, Jiří Macas, Petr Novák, Mathieu Piednoël, Hanna Weiss-Schneeweiss, and Andrew R Leitch. Genomic Repeat Abundances Contain Phylogenetic Signal. *Systematic Biology*, 64(1):112–126, 2014.

[80] Michael C. Schatz, Arthur L. Delcher, and Steven L. Salzberg. Assembly of large genomes using second-generation sequencing. *Genome Research*, 20(9):1165–1173, 2010.

[81] Brandi L. Cantarel, Hilary G. Morrison, and William Pearson. Exploring the relationship between sequence similarity and accurate phylogenetic trees. *Molecular Biology and Evolution*, 23(11):2090–2100, 2006.

[82] T Heath Ogdenw and Michael S Rosenberg. Multiple sequence alignment accuracy and phylogenetic inference. *Systematic Biology*, 55(2):314–328, 2006.

[83] Li-San Wang, Jim Leebens-Mack, P Kerr Wall, Kevin Beckmann, Claude W DePamphilis, and Tandy Warnow. The impact of multiple protein sequence alignment on phylogenetic estimation. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 8(4):1108–19, 2011.

[84] Ari Löytynoja and Nick Goldman. Phylogeny-aware gap placement prevents errors in sequence alignment and evolutionary analysis. *Science*, 320(5883):1632–1635, 2008.

[85] Ben D Redelings and Marc A Suchard. Joint Bayesian Estimation of Alignment and Phylogeny. *Systematic Biology*, 54(3):401–418, 2005.

[86] Orjan Akerborg, Bengt Sennblad, Lars Arvestad, and Jens Lagergren. Simultaneous Bayesian gene tree reconstruction and reconciliation analysis. *Proceedings of the National Academy of Sciences of the United States of America*, 106(14):5714–9, 2009.

[87] Julie D Thompson, Desmond G Higgins, and Toby J Gibson. CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic Acids Research*, 22(22):4673–4680, 1994.

[88] Robert C. Edgar. MUSCLE: multiple sequence alignment with high accuracy and high throughput. *Nucleic Acids Research*, 32(5):1792–1797, 2004.

[89] Kazutaka Katoh, Kei-ichi Kuma, Hiroyuki Toh, and Takashi Miyata. MAFFT version 5: improvement in accuracy of multiple sequence alignment. *Nucleic Acids Research*, 33(2):511–518, 2005.

[90] Kazutaka Katoh and Hiroyuki Toh. Recent developments in the MAFFT multiple sequence alignment program. *Briefings in Bioinformatics*, 9(4):286–298, 2008.

[91] Saul B Needleman and Christian D Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology*, 48(3):443–453, 1970.

[92] Temple F. Smith and Michael S. Waterman. Identification of common molecular subsequences. *Journal of Molecular Biology*, 147(1):195–197, 1981.

[93] Travis Wheeler and John Kececioglu. Multiple alignment by aligning alignments. In *Proceedings of the 15th ISCB Conference on Intelligent Systems for Molecular Biology*, pages 559–568, 2007.

[94] Cédric Notredame, Desmond G Higgins, and J Heringa. T-Coffee: A novel method for fast and accurate multiple sequence alignment. *Journal of Molecular Biology*, 302(1):205–217, 2000.

[95] Robert K. Bradley, Adam Roberts, Michael Smoot, Sudeep Juvekar, Jaeyoung Do, Colin N Dewey, Ian Holmes, and Lior Pachter. Fast statistical alignment. *PLoS Computational Biology*, 5(5), 2009.

[96] Chuong B. Do, M. S P Mahabhashyam, Michael W Bruford, and Serafim Batzoglou. ProbCons: Probabilistic consistency-based multiple sequence alignment. *Genome Research*, 15(2):330–340, 2005.

[97] Ari Löytynoja and Nick Goldman. An algorithm for progressive multiple alignment of sequences with insertions. *Proceedings of the National Academy of Sciences of the United States of America*, 102(30):10557–10562, 2005.

[98] Ari Löytynoja, Albert J. Vilella, and Nick Goldman. Accurate extension of multiple sequence alignments using a phylogeny-aware graph algorithm. *Bioinformatics*, 28(13):1685–1691, 2012.

[99] Jeffrey L Thorne, Hirohisa Kishino, and Joseph Felsenstein. Inching toward reality: an improved likelihood model of sequence evolution. *Journal of Molecular Evolution*, 34(1):3–16, 1992.

[100] Jeffrey L Thorne, Hirohisa Kishino, and Joseph Felsenstein. An evolutionary model for maximum likelihood alignment of DNA sequences. *Journal of Molecular Evolution*, 33(2):114–24, 1991.

[101] Alexandre Bouchard-Côté and Michael I Jordan. Evolutionary inference via the Poisson Indel Process. *Proceedings of the National Academy of Sciences of the United States of America*, 110(4):1160–6, 2013.

[102] Roland Fleissner, Dirk Metzler, and Arndt von Haeseler. Simultaneous statistical multiple alignment and phylogeny reconstruction. *Systematic Biology*, 54(4):548–561, 2005.

[103] Walter R Gilks. *Markov Chain Monte Carlo*. Wiley Online Library, 2005.

[104] Heejung Shim and Bret R Larget. BayesCAT: Bayesian Co-estimation of Alignment and Tree. *arXiv*, 1411.6150, 2014.

[105] Marc A Suchard and Ben D Redelings. BAli-Phy: simultaneous Bayesian inference of alignment and phylogeny. *Bioinformatics*, 22:2047–2048, 2006.

[106] William H.E. Day, David S. Johnson, and David Sankoff. The computational complexity of inferring rooted phylogenies by parsimony. *Mathematical Biosciences*, 81(1):33–42, 1986.

[107] David L Swofford. *PAUP*. Phylogenetic Analysis Using Parsimony (*and other methods). Version 4.* Sinauer Associates, 2003.

[108] Pablo A Goloboff, James S Farris, and Kevin Nixon. TNT: Tree Analysis Using New Technology. *Systematic Biology*, 54(1):176–178, 2005.

[109] Ganeshkumar Ganapathy, Vijaya Ramachandran, and Tandy Warnow. Better Hill-Climbing Searches for Parsimony. In *Algorithms in Bioinformatics*, pages 245–258. Springer, 2003.

[110] Maria Bonet, Michael Steel, Tandy Warnow, and Shibu Yooseph. Better methods for solving parsimony and compatibility. *Journal of Computational Biology*, 5(3):391–407, 1998.

[111] Joseph Felsenstein. Cases in which Parsimony or Compatibility Methods Will be Positively Misleading. *Systematic Zoology*, 27(4):401–410, 1978.

[112] Thomas H Jukes and Charles R Cantor. Evolution of protein molecules. *Mammalian Protein Metabolism*, 3:21–132, 1969.

[113] Jeffrey L Thorne. Models of protein sequence evolution and their applications. *Current Opinion in Genetics & Development*, 10(6):602–605, 2000.

[114] Richard Durrett. *Probability models for DNA sequence evolution.* Springer Science & Business Media, 2008.

[115] Sebastien Roch. A short proof that phylogenetic tree reconstruction by maximum likelihood is hard. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 3(1):92–94, 2006.

[116] Daniel Money and Simon Whelan. Characterizing the phylogenetic tree-search problem. *Systematic Biology*, 61(2):228–239, 2012.

[117] Derrick Joel Zwickl. *Genetic algorithm approaches for the phylogenetic analysis of large biological sequence datasets under the maximum likelihood criterion.* PhD thesis, University of Texas at Austin, 2006.

[118] Alexandros Stamatakis, T Ludwig, and H Meier. RAxML-II: a program for sequential, parallel and distributed inference of large phylogenetic. *Concurrency and Computation: Practice Experience*, 17(14):1705–1723, 2005.

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

[120] Alexandros Stamatakis. RAxML-VI-HPC: Maximum likelihood-based phylogenetic analyses with thousands of taxa and mixed models. *Bioinformatics*, 22(21):2688–2690, 2006.

[121] Stéphane Guindon, Jean-François Dufayard, Vincent Lefort, Maria Anisimova, Wim Hordijk, and Olivier Gascuel. New Algorithms and Methods to Estimate Maximum-Likelihood Phylogenies: Assessing the Performance of PhyML 3.0. *Systematic Biology*, 59(3):307–321, 2010.

[122] Siavash Mirarab, Nam Nguyen, Sheng Guo, Li-San Wang, Junhyong Kim, and Tandy Warnow. PASTA: Ultra-Large Multiple Sequence Alignment for Nucleotide and Amino-Acid Sequences. *Journal of Computational Biology*, 22(05):377–386, 2015.

[123] Kevin Liu, C. Randal Linder, and Tandy Warnow. RAxML and FastTree: Comparing Two Methods for Large-Scale Maximum Likelihood Phylogeny Estimation. *PLoS ONE*, 6(11), 2011.

[124] John P Huelsenbeck and Fredrik Ronquist. MRBAYES: Bayesian inference of phylogenetic trees. *Bioinformatics*, 17(8):754–755, 2001.

[125] Sebastian Höhna, Tracy A Heath, and Bastien Boussau. Probabilistic graphical model representation in phylogenetics. *Systematic Biology*, 63(5):753–771, 2014.

[126] Alexei J Drummond and Andrew Rambaut. BEAST: Bayesian evolutionary analysis by sampling trees. *BMC Evolutionary Biology*, 7:214, 2007.

[127] Naruya Saitou and Masatoshi Nei. The neighbour-joining method: a new method for reconstructing phylogenetic trees. *Molecular Biology and Evolution*, 4(4):406–425, 1987.

[128] Michael Steel. Recovering a tree from the leaf colourations it generates under a Markov model. *Applied Mathematics Letters*, 7(2):19–23, 1994.

[129] David M Hillis, James J Bull, Mary E White, Marty R Badgett, and Ian J Molineux. Experimental phylogenetics: generation of a known phylogeny. *Science*, 255(5044):589–592, 1992.

[130] David M Hillis and James J Bull. An Empirical Test of Bootstrapping as a Method for Assessing Confidence in Phylogenetic Analysis. *Systematic Biology*, 42(2):182–192, 1993.

[131] Joseph Felsenstein. Confidence Limits on Phylogenies: An Approach Using the Bootstrap. *Evolution*, 39(4):783–791, 1985.

[132] Jonathan A Eisen. Phylogenomics: improving functional predictions for uncharacterized genes by evolutionary analysis. *Genome Research*, 8(3):163–167, 1998.

[133] Bret Larget, Satish K Kotha, Colin N Dewey, and Cécile Ané. BUCKy: Gene tree/species tree reconciliation with the Bayesian concordance analysis. *Bioinformatics*, 26(22):2910–2911, 2010.

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

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

[136] Sebastien Roch and Michael Steel. Likelihood-based tree reconstruction on a concatenation of aligned sequence data sets can be statistically inconsistent. *Theoretical Population Biology*, 100:56–62, 2014.

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

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

[139] Cuong Than and Luay Nakhleh. Species tree inference by minimizing deep coalescences. *PLoS Computational Biology*, 5(9):e1000501, 2009.

[140] Yun Yu, Tandy Warnow, and Luay Nakhleh. Algorithms for MDC-based multi-locus phylogeny inference: beyond rooted binary gene trees on single alleles. *Journal of Computational Biology*, 18(11):1543–1559, 2011.

[141] Cuong Than and Noah A. Rosenberg. Consistency properties of species tree inference by minimizing deep coalescences. *Journal of Computational Biology*, 18(1):1–15, 2011.

[142] Liang Liu, Lili Yu, Dennis K Pearl, and Scott V Edwards. Estimating species phylogenies using coalescence times among sequences. *Systematic Biology*, 58(5):468–477, 2009.

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

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

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

[146] Liang Liu and Lili Yu. Estimating species trees from unrooted gene trees. *Systematic Biology*, 60:661–667, 2011.

[147] Sebastien Roch and Tandy Warnow. On the robustness to gene tree estimation error (or lack thereof) of coalescent-based species tree methods. *Systematic Biology*, page syv016, 2015.

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

[149] Md. Shamsuzzoha Bayzid and Tandy Warnow. Estimating optimal species trees from incomplete gene trees under deep coalescence. *Journal of Computational Biology*, 19(6):591–605, 2012.

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

[151] Josef Heled and Alexei J Drummond. Bayesian inference of species trees from multilocus data. *Molecular Biology and Evolution*, 27:570 – 580, 2010.

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

[153] Théo Zimmermann, Siavash Mirarab, and Tandy Warnow. BBCA: Improving the scalability of *BEAST using random binning. *BMC Genomics*, 15(Suppl 6):S11, 2014.

[154] Cécile Ané, Bret R Larget, David A Baum, Stacey D Smith, and Antonis Rokas. Bayesian estimation of concordance among gene trees. *Molecular Biology and Evolution*, 24(2):412–426, 2007.

[155] Yujin Chung and Cécile 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, 2011.

[156] David Bryant, Remco Bouckaert, Joseph Felsenstein, Noah A. Rosenberg, and Arindam 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.

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

[158] David F Robinson and Leslie R Foulds. Comparison of phylogenetic trees. *Mathematical Biosciences*, 53(1-2):131–147, 1981.

[159] Tandy Warnow. Tree compatibility and inferring evolutionary history. *Journal of Algorithms*, 16(3):388–407, 1994.

[160] Siavash Mirarab and Tandy Warnow. FastSP: linear time calculation of alignment accuracy. *Bioinformatics*, 27:3250–8, 2011.

[161] Siavash Mirarab, Nam Nguyen, and Tandy Warnow. PASTA: ultra-large multiple sequence alignment. In *Proceedings of the International Conference on Research in Computational Molecular Biology*, pages 177–191. Springer International Publishing, 2014.

[162] F. Sievers, D. Dineen, A. Wilm, and D. G. Higgins. Making automated multiple alignments of very large numbers of protein sequences. *Bioinformatics*,

29(8):989–995, 2013.

[163] Robert C. Edgar. MUSCLE: a multiple sequence alignment method with reduced time and space complexity. *BMC Bioinformatics*, 5(113):113, 2004.

[164] Kazutaka Katoh, Kazutaka, Hiroyuki Toh, and Hiroyuki. PartTree: an algorithm to build an approximate tree from a large number of unaligned sequences. *Bioinformatics*, 23(3):372–374, 2007.

[165] Kazutaka Katoh and M C Frith. Adding unaligned sequences into an existing alignment using MAFFT and LAST. *Bioinformatics*, 28(23):3144–3146, 2012.

[166] Faruck Morcos, Andrea Pagnani, Bryan Lunt, Arianna Bertolino, Debora S Marks, Chris Sander, Riccardo Zecchina, José N Onuchic, Terence Hwa, and Martin Weigt. Direct-coupling analysis of residue coevolution captures native contacts across many protein families. *Proceedings of the National Academy of Sciences of the United States of America*, 108(49):E1293–E1301, 2011.

[167] Derrick Joel Zwickl and David M Hillis. Increased taxon sampling greatly reduces phylogenetic error. *Systematic Biology*, 51(4):588–98, 2002.

[168] David M Hillis, David D Pollock, Jimmy A McGuire, and Derrick Joel Zwickl. Is sparse taxon sampling a problem for phylogenetic inference? *Systematic Biology*, 52(1):124–126, 2003.

[169] Tracy A Heath, Shannon M Hedtke, and David M Hillis. Taxon sampling and the accuracy of phylogenetic analyses. *Journal of Systematics and Evolution*, 46:239–257, 2008.

[170] iPlant Collaborative. iPTOL, Assembling the Tree of Life for the Plant Sciences. `https://pods.iplantcollaborative.org/wiki/display/iptol/Home`, 2013.

[171] Sean R. Eddy. A new generation of homology search tools based on probabilistic inference. *Genome Informatics*, 23:205–211, 2009.

[172] Robert D. Finn, Jody Clements, and Sean R. Eddy. HMMER web server: interactive sequence similarity searching. *Nucleic Acids Research*, 39:W29–W37, 2011.

[173] Nam Nguyen, Siavash Mirarab, Keerthana Kumar, and Tandy Warnow. Ultra-large alignments using phylogeny-aware profiles. *Genome Biology*, 16(1):124, 2015.

[174] G Udny Yule. A mathematical theory of evolution, based on the conclusions of Dr. JC Willis, FRS. *Philosophical Transactions of the Royal Society of London*, pages 21–87, 1925.

[175] Jen Stoye, Dirk Evers, and Folker Meyer. Rose: generating sequence families. *Bioinformatics*, 14(2):157–163, 1998.

[176] William Fletcher and Ziheng Yang. Indelible: A flexible simulator of biological sequence evolution. *Molecular Biology and Evolution*, 26(8):1879–1888, 2009.

[177] Sheng Guo, Li-San Wang, and Junhyong Kim. Large-scale simulation of RNA macroevolution by an energy-dependent fitness model. *arXiv*, 0912.2326,

2009.

[178] Jamie J Cannone, Sankar Subramanian, Murray N Schnare, James R Collett, Lisa M D'Souza, Yushi Du, Brian Feng, Nan Lin, Lakshmi V Madabusi, Kirsten M Müller, Nupur Pande, Zhidi Shang, Nan Yu, and Robin R. Gutell. The Comparative RNA Web (CRW) Site: An Online Database of Comparative Sequence and Structure Information for Ribosomal, Intron and Other RNAs. *BMC Bioinformatics*, 3(15), 2002.

[179] Julie D Thompson, Benjamin Linard, Odile Lecompte, and Olivier Poch. A comprehensive benchmark study of multiple sequence alignment methods: current challenges and future perspectives. *PLoS ONE*, 6(3):e18093, 2011.

[180] Gregory B Gloor, Louise C Martin, Lindi M Wahl, and Stanley D Dunn. Mutual information in protein multiple sequence alignments reveals two classes of coevolving positions. *Biochemistry*, 44(19):7156–65, 2005.

[181] Fabian Sievers, Andreas Wilm, David Dineen, Toby J Gibson, Kevin Karplus, Weizhong Li, Rodrigo Lopez, Hamish McWilliam, Michael Remmert, Johannes Söding, Julie D Thompson, and Desmond G Higgins. Fast, scalable generation of high-quality protein multiple sequence alignments using Clustal Omega. *Molecular Systems Biology*, 7(539), 2011.

[182] Kenji Mizuguchi, Charlotte M. Deane, Tom L. Blundell, and John P. Overington. HOMSTRAD: a database of protein structure alignments for homologous families. *Protein Science*, 7:2469–2471, 1998.

[183] Jay Boisseau and Dan Stanzione. TACC: Texas Advanced Computing Center (webpage). `http://www.tacc.utexas.edu`, 2013.

[184] Siavash Mirarab, Nam Nguyen, and Tandy Warnow. SEPP: SATé-Enabled Phylogenetic Placement. *Pacific Symposium On Biocomputing*, pages 247–58, 2012.

[185] David Sankoff and Joseph H Nadeau. *Comparative genomics.* Springer, 2000.

[186] Matthew R Helmus, Thomas J Bland, Christopher K Williams, and Anthony R Ives. Phylogenetic Measures of Biodiversity. *The American Naturalist*, 169(3):E68–E83, 2007.

[187] Hervé Philippe, Romain Derelle, Philippe Lopez, Kerstin Pick, Carole Borchiellini, Nicole Boury-Esnault, Jean Vacelet, Emmanuelle Renard, Evelyn Houliston, Eric Quéinnec, and Others. Phylogenomics revives traditional views on deep animal relationships. *Current Biology*, 19(8):706–712, 2009.

[188] Leonidas Salichos and Antonis Rokas. Inferring ancient divergences requires genes with strong phylogenetic signals. *Nature*, 497(7449):327–31, 2013.

[189] Ylenia Chiari, Vincent Cahais, Nicolas Galtier, and Frédéric Delsuc. Phylogenomic analyses support the position of turtles as the sister group of birds and crocodiles (archosauria). *BMC Biology*, 10(1):65, 2012.

[190] Siavash Mirarab, Md. Shamsuzzoha Bayzid, Bastien Boussau, and Tandy Warnow. Statistical binning enables an accurate coalescent-based estimation of the avian tree. *Science*, 346(6215), 2014.

[191] Md. Shamsuzzoha Bayzid, Siavash Mirarab, Bastien Boussau, and Tandy Warnow. Weighted Statistical Binning: enabling statistically consistent genome-scale phylogenetic analyses. *PLoS ONE*, 10(6):e0129183, 2015.

[192] Liang Liu and Dennis K Pearl. Species trees from gene trees: reconstructing Bayesian posterior distributions of a species phylogeny using estimated gene tree distributions. *Systematic Biology*, 56(3):504–14, 2007.

[193] Zhenxiang Xi, Liang Liu, Joshua S Rest, and Charles C Davis. Coalescent versus Concatenation Methods and the Placement of Amborella as Sister to Water Lilies. *Systematic Biology*, 63(6):919–932, 2014.

[194] Vikas Kumar, Björn M Hallström, and Axel Janke. Coalescent-based genome analyses resolve the early branches of the euarchontoglires. *PLoS ONE*, 8(4):e60019, 2013.

[195] Rebecca T Kimball, Ning Wang, Victoria Heimer-McGinn, Carly Ferguson, and Edward L Braun. Identifying localized biases in large datasets: A case study using the avian tree of life. *Molecular Phylogenetics and Evolution*, 69:1021–1032, 2013.

[196] Michael DeGiorgio and James H. Degnan. Fast and consistent estimation of species trees using supermatrix rooted triples. *Molecular Biology and Evolution*, 27(3):552–69, 2010.

[197] Luay Nakhleh, Katherine St. John, Usman Roshan, Jerry Sun, and Tandy Warnow. Designing fast converging phylogenetic methods. *Bioinformatics*,

17:S190–S198, 2001.

[198] Indrajit Nanda, Zhihong Shan, Manfred Schartl, Dave W Burt, Michael Koehler, Hans-Gerd Nothwang, Frank Grützner, Ian R Paton, Dawn Windsor, and Ian Dunn. 300 million years of conserved synteny between chicken Z and human chromosome 9. *Nature Genetics*, 21(3):258–259, 1999.

[199] Daniel Brélaz. New methods to color the vertices of a graph. *Communications of the ACM*, 22(4):251–256, 1979.

[200] Douglas Brent West. *Introduction to graph theory*, volume 2. Prentice-Hall Inc., 2001.

[201] Richard M Karp. Reducibility among combinatorial problems. In *Complexity of Computer Computations*, The IBM Research Symposia Series, pages 85–103. Springer, 1972.

[202] Tandy Warnow. Concatenation Analyses in the Presence of Incomplete Lineage Sorting. *PLoS Currents: Tree of Life*, 1, 2015.

[203] Michael Steel. A consistency lemma in statistical phylogenetics. *arXiv*, 1501.06623, 2015.

[204] Jeet Sukumaran and Mark T Holder. Dendropy: a Python library for phylogenetic computing. *Bioinformatics*, 26(12):1569–71, 2010.

[205] Julien Dutheil and Bastien Boussau. Non-homogeneous models of sequence evolution in the Bio++ suite of libraries and programs. *BMC Evolutionary Biology*, 8:255, 2008.

[206] Ziheng Yang, 2015. MCCoal: http://abacus.gene.ucl.ac.uk/software/MCMCcoal.html.

[207] Mark A Ragan. Phylogenetic inference based on matrix representation of trees. *Molecular Phylogenetics and Evolution*, 1(1):53–58, 1992.

[208] Tae Kun Seo. Calculating bootstrap probabilities of phylogeny using multilocus sequence data. *Molecular Biology and Evolution*, 25(5):960–971, 2008.

[209] Adam D Leaché, Rebecca B Harris, Bruce Rannala, and Ziheng Yang. The influence of gene flow on species tree estimation: a simulation study. *Systematic Biology*, 63(1):17–30, 2014.

[210] Lei Zhao, Ning Zhang, Peng-Fei Ma, Qi Liu, De-Zhu Li, and Zhen-Hua Guo. Phylogenomic analyses of nuclear genes reveal the evolutionary relationships within the BEP clade and the evidence of positive selection in poaceae. *PLoS ONE*, 8(5):e64642, 2013.

[211] Bojian Zhong, Liang Liu, Zhen Yan, and David Penny. Origin of land plants using the multispecies coalescent model. *Trends in Plant Science*, 18(9):492–495, 2013.

[212] James H. Degnan, Michael DeGiorgio, David Bryant, and Noah A. Rosenberg. Properties of consensus methods for inferring species trees from gene trees. *Systematic Biology*, 58(1):35–54, 2009.

[213] Thomas P Wilcox, Derrick J Zwickl, Tracy A Heath, and David M Hillis. Phylogenetic relationships of the dwarf boas and a comparison of Bayesian

and bootstrap measures of phylogenetic support. *Molecular Phylogenetics and Evolution*, 25(2):361–371, 2002.

[214] Solomon Kullback and Richard A. Leibler. On Information and Sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86, 1951.

[215] Yoav Benjamini and Yosef Hochberg. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society*, 57:289–300, 1995.

[216] Bernard W Silverman. *Density estimation for statistics and data analysis*, volume 26. CRC press, 1986.

[217] R Development Core Team. *R: A Language and Environment for Statistical Computing.* R Foundation for Statistical Computing, 2011.

[218] Alexander Suh, Martin Paus, Martin Kiefmann, Gennady Churakov, Franziska Anni Franke, Jürgen Brosius, Jan Ole Kriegs, and Jürgen Schmitz. Mesozoic retroposons reveal parrots as the closest living relatives of passerine birds. *Nature Communications*, 2:443, 2011.

[219] Ning Wang, Edward L Braun, and Rebecca T Kimball. Testing hypotheses about the sister group of the Passeriformes using an independent 30-locus data set. *Molecular Biology and Evolution*, 29(2):737–750, 2012.

[220] Jan E Janecka, Webb Miller, Thomas H Pringle, Frank Wiens, Annette Zitzmann, Kristofer M Helgen, Mark S Springer, and William J Murphy. Molec-

ular and genomic data identify the closest living relative of primates. *Science*, 318(5851):792–794, 2007.

[221] Bastien Boussau, GJ Szöllősi, Laurent Duret, M. Gouy, E. Tannier, and V. Daubin. Genome-scale coestimation of species and gene trees. *Genome Research*, 23(2):323–330, 2013.

[222] Claus Nielsen. *Animal evolution: interrelationships of the living phyla.* Oxford University Press, 2012.

[223] Frédéric Delsuc, Henner Brinkmann, Daniel Chourrout, and Hervé Philippe. Tunicates and not cephalochordates are the closest living relatives of vertebrates. *Nature*, 439(7079):965–968, 2006.

[224] Sarah J Bourlat, Thorhildur Juliusdottir, Christopher J Lowe, Robert Freeman, Jochanan Aronowicz, Mark Kirschner, Eric S Lander, Michael Thorndyke, Hiroaki Nakano, Andrea B Kohn, Andreas Heyland, Leonid L Moroz, Richard R Copley, and Maximilian J Telford. Deuterostome phylogeny reveals monophyletic chordates and the new phylum Xenoturbellida. *Nature*, 444(7115):85–88, 2006.

[225] Tiratha Raj Singh, Georgia Tsagkogeorga, Frédéric Delsuc, Samuel Blanquart, Noa Shenkar, Yossi Loya, Emmanuel Jp Douzery, and Dorothée Huchon. Tunicate mitogenomics and phylogenetics: peculiarities of the Herdmania momus mitochondrial genome and support for the new chordate phylogeny. *BMC Genomics*, 10:534, 2009.

[226] Bernd Schierwater, Michael Eitel, Wolfgang Jakob, Hans-Jürgen Osigus, Heike Hadrys, Stephen L Dellaporta, Sergios Orestis Kolokotronis, and Rob DeSalle. Concatenated analysis sheds light on early metazoan evolution and fuels a modern urmetazoon hypothesis. *PLoS Biology*, 7(1):e1000020, 2009.

[227] Joseph F Ryan, Kevin Pang, Christine E Schnitzler, Anh-Dao Nguyen, R Travis Moreland, David K Simmons, Bernard J Koch, Warren R Francis, Paul Havlak, Stephen A Smith, Nicholas H Putnam, Steven H D Haddock, Casey W Dunn, Tyra G Wolfsberg, James C Mullikin, Mark Q Martindale, and Andreas D Baxevanis. The Genome of the Ctenophore Mnemiopsis leidyi and Its Implications for Cell Type Evolution. *Science*, 342(6164):1242592+, 2013.

[228] Gregory D Edgecombe, Gonzalo Giribet, Casey W Dunn, Andreas Hejnol, Reinhardt M Kristensen, Ricardo C Neves, Greg W Rouse, Katrine Worsaae, and Martin V Sø rensen. Higher-level metazoan relationships: recent progress and remaining questions. *Organisms Diversity & Evolution*, 11(2):151–172, 2011.

[229] Keisuke Ishiwata, Go Sasaki, Jiro Ogawa, Takashi Miyata, and Zhi-Hui Su. Phylogenetic relationships among insect orders based on three nuclear protein-coding gene sequences. *Molecular Phylogenetics and Evolution*, 58(2):169–180, 2011.

[230] Jingyang HU, Yaping ZHANG, and Li YU. Summary of Laurasiatheria (Mammalia) Phylogeny. *Zoological Research*, 33(E5-6):E65–74, 2012.

[231] Bernard Dujon. Yeast evolutionary genomics. *Nature Reviews Genetics*, 11(7):512–524, 2010.

[232] Hayley C. Lanier and Lacy L. Knowles. Is recombination a problem for species-tree analyses? *Systematic Biology*, 61:691–701, 2012.

[233] Jessica W Leigh, Edward Susko, Manuela Baumgartner, and Andrew J Roger. Testing congruence in phylogenomic analysis. *Systematic Biology*, 57:104–115, 2008.

[234] Siavash Mirarab, Rezwana Reaz, Md. Shamsuzzoha Bayzid, Théo Zimmermann, M Shel Swenson, and Tandy Warnow. ASTRAL: Genome-Scale Coalescent-Based Species Tree. *Bioinformatics*, 30(17):i541–i548, 2014.

[235] 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, 2015.

[236] Md Shamsuzzoha Bayzid, Tyler Hunt, and Tandy Warnow. Disk covering methods improve phylogenomic analyses. *BMC Genomics*, 15(Suppl 6):S7, 2014.

[237] Yuancheng Wang and James H. Degnan. Performance of Matrix Representation with Parsimony for Inferring species from gene trees. *Statistical Applications in Genetics and Molecular Biology*, 10(1):1–39, 2011.

[238] Nam Nguyen, Siavash Mirarab, and Tandy Warnow. MRL and SuperFine+MRL: new supertree methods. *Algorithms for Molecular Biology*, 7(1):3, 2012.

[239] Michael DeGiorgio and James H. Degnan. Robustness to divergence time underestimation when inferring species trees from estimated gene trees. *Systematic Biology*, 63(1):66–82, 2014.

[240] James H. Degnan. Anomalous unrooted gene trees. *Systematic Biology*, 62:574–590, 2013.

[241] Tao Jiang, Paul Kearney, and Ming Li. A Polynomial Time Approximation Scheme for Inferring Evolutionary Trees from Quartet Topologies and Its Application. *SIAM Journal on Computing*, 30(6):1942–1961, 2001.

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

[243] Peter Buneman. The recovery of trees from measures of dissimilarity. *Mathematics in the archaeological and historical sciences*, pages 387–395, 1971.

[244] Michael T. Hallett and Jens Lagergren. New algorithms for the duplication-loss model. In *Proceedings of the International Conference on Research in Computational Molecular Biology*, pages 138–146. ACM, 2000.

[245] David Bryant and Michael Steel. Constructing Optimal Trees from Quartets. *Journal of Algorithms*, 38:237–259, 2001.

[246] Peter Buneman. A note on the metric properties of trees. *Journal of Combinatorial Theory, Series B*, 17(1):48–50, 1974.

[247] Peter Erdos, Michael Steel, László Székely, and Tandy Warnow. A few logs suffice to build (almost) all trees (I). *Random Structures and Algorithms*, 14(2):153–184, 1999.

[248] Diego Mallo, L de Oliveira Martins, and D Posada. Simphy: Comprehensive simulation of gene, locus and species trees at the genome-wide level., 2015. (In Prep, available at `https://code.google.com/p/simphy-project/`).

[249] Alexandros Stamatakis, Andre J Aberer, C Goll, Stephen A Smith, Simon A Berger, and Fernando Izquierdo-Carrasco. RAxML-Light: a tool for computing terabyte phylogenies. *Bioinformatics*, 28(15):2064–2066, 2012.

[250] Hidetoshi Shimodaira and Masami Hasegawa. Multiple Comparisons of Log-Likelihoods with Applications to Phylogenetic Inference. *Molecular Biology and Evolution*, 16(8), 1999.

[251] Sabina Wodniok, Henner Brinkmann, Gernot Glöckner, Andrew J Heidel, Hervé Philippe, Michael Melkonian, and Burkhard Becker. Origin of land plants: do conjugating green algae hold the key? *BMC Evolutionary Biology*, 11:104, 2011.

[252] Ruth E. Timme, Tsvetan R. Bachvaroff, and Charles F. Delwiche. Broad phylogenomic sampling and the sister lineage of land plants. *PLoS ONE*, 7(1), 2012.

[253] Brad R Ruhfel, Matthew A. Gitzendanner, Pamela S Soltis, Douglas E Soltis, and J Gordon Burleigh. From algae to angiosperms - inferring the phylogeny

of green plants (Viridiplantae) from 360 plastid genomes. *BMC Evolutionary Biology*, 14:23, 2014.

[254] Cédric Finet, Ruth E. Timme, Charles F. Delwiche, and Ferdinand Marlétaz. Multigene phylogeny of the green lineage reveals the origin and diversification of land plants. *Current Biology*, 20(24):2217–2222, 2010.

[255] Monique Turmel, Christian Otis, and Claude Lemieux. The chloroplast genome sequence of Chara vulgaris sheds new light into the closest green algal relatives of land plants. *Molecular Biology and Evolution*, 23(6):1324–1338, 2006.

[256] Kenneth G Karol, Richard M McCourt, Matthew T Cimino, and Charles F Delwiche. The closest living relatives of land plants. *Science*, 294(5550):2351–2353, 2001.

[257] Roberto Ligrone, Jeffrey G. Duckett, and Karen S. Renzaglia. Major transitions in the evolution of early land plants: A bryological perspective. *Annals of Botany*, 109(5):851–871, 2012.

[258] Yin-Long Qiu, Libo Li, Bin Wang, Zhi-Duan Chen, Olena Dombrovska, Jungho Lee, Livija Kent, Rui-Qi Li, Richard W. Jobson, Tory A. Hendry, David W. Taylor, Christopher M. Testa, and Mathew Ambros. A Nonflowering Land Plant Phylogeny Inferred from Nucleotide Sequences of Seven Chloroplast, Mitochondrial, and Nuclear Genes. *International Journal of Plant Sciences*, 168(5):691–708, 2007.

[259] Tomoaki Nishiyama, Paul G. Wolf, Masanori Kugita, Robert B. Sinclair, Mamoru Sugita, Chika Sugiura, Tatsuya Wakasugi, Kyoji Yamada, Koichi Yoshinaga, Kazuo Yamaguchi, Kunihiko Ueda, and Mitsuyasu Hasebe. Chloroplast phylogeny indicates that bryophytes are monophyletic. *Molecular Biology and Evolution*, 21(10):1813–1819, 2004.

[260] L Michelle Bowe and Gwénaële Coat. Phylogeny of seed plants based on all three genomic compartments: Extant gymnosperms are monophyletic and Gnetales' closest relatives are conifers. *Proceedings of the National Academy of Sciences of the United States of America*, 97(8):4092–4097, 2000.

[261] Jose Eduardo De La Torre-bárcena, Sergios Orestis Kolokotronis, Ernest K. Lee, Dennis Wm Stevenson, Eric D. Brenner, Manpreet S. Katari, Gloria M. Coruzzi, and Rob DeSalle. The impact of outgroup choice and missing data on major seed plant phylogenetics using genome-wide EST data. *PLoS ONE*, 4(6):e5764, 2009.

[262] Shu-Miaw Chaw, Andrey Zharkikh, Huang-Mo Sung, Tak-Cheung Lau, and Wen-Hsiung Li. Molecular phylogeny of extant gymnosperms and seed plant evolution: analysis of nuclear 18S rRNA sequences. *Molecular Biology and Evolution*, 14(1):56–68, 1997.

[263] Yin-Long Qiu, Jungho Lee, Fabiana Bernasconi-Quadroni, Douglas E Soltis, Pamela S Soltis, Michael Zanis, Elizabeth A Zimmer, Zhi-Duan Chen, Vincent Savolainen, and Mark W Chase. The earliest angiosperms: evidence from mitochondrial, plastid and nuclear genomes. *Nature*, 402:404–407, 1999.

[264] Pamela S Soltis, Douglas E Soltis, and Mark W. Chase. Angiosperm phylogeny inferred from multiple genes as a tool for comparative biology. *Nature*, 402(6760):402–404, 1999.

[265] Michael J Moore, Charles D Bell, Pamela S Soltis, and Douglas E Soltis. Using plastid genome-scale data to resolve enigmatic relationships among basal angiosperms. *Proceedings of the National Academy of Sciences of the United States of America*, 104(49):19363–19368, 2007.

[266] Yin-Long Qiu, Libo Li, Bin Wang, Jia-Yu Xue, Tory a. Hendry, Rui-Qi Li, Joseph W. Brown, Yang Liu, Geordan T. Hudson, and Zhi-Duan Chen. Angiosperm phylogeny inferred from sequences of four mitochondrial genes. *Journal of Systematics and Evolution*, 48(6):391–425, 2010.

[267] Vadim V Goremykin, Svetlana V Nikiforova, Patrick J Biggs, Bojian Zhong, Peter Delange, William Martin, Stefan Woetzel, Robin A Atherton, Patricia A Mclenachan, and Peter J Lockhart. The Evolutionary Root of Flowering Plants. *Systematic Biology*, 62(1):50–61, 2013.

[268] Oliver Jeffroy, Henner Brinkmann, Frédéric Delsuc, and Hervé Philippe. Phylogenomics: the beginning of incongruence? *Trends in Genetics*, 22(4):225–231, 2006.

[269] Ernest K. Lee, Angelica Cibrian-Jaramillo, Sergios Orestis Kolokotronis, Manpreet S. Katari, Alexandros Stamatakis, Michael Ott, Joanna C. Chiu, Damon P. Little, Dennis Wm Stevenson, W. Richard McCombie, Robert A.

Martienssen, Gloria M. Coruzzi, and Rob DeSalle. A functional phylogenomic view of the seed plants. *PLoS Genetics*, 7(12), 2011.

[270] Yin-Long Qiu, Jungho Lee, Fabiana Bernasconi-Quadroni, Douglas E. Soltis, Pamela S. Soltis, Michael Zanis, Elizabeth A. Zimmer, Zhiduan Chen, Vincent Savolainen, and Mark W. Chase. Phylogeny of Basal Angiosperms: Analyses of Five Genes from Three Genomes. *International Journal of Plant Sciences*, 161:S3–S27, 2000.

[271] Ning Zhang, Liping Zeng, Hongyan Shan, and Hong Ma. Highly conserved low-copy nuclear genes as effective markers for phylogenetic analyses in angiosperms. *New Phytologist*, 195:923–937, 2012.

[272] Bryan T Drew, Brad R Ruhfel, Stephen A Smith, Michael J Moore, Barbara G Briggs, Matthew A Gitzendanner, Pamela S Soltis, and Douglas E Soltis. Another Look at the Root of the Angiosperms Reveals a Familiar Tale. *Systematic Biology*, 63(3):368–382, 2014.

[273] Andrew F Hugall, Ralph Foster, and Michael SY Lee. Calibration choice, rate smoothing, and the pattern of tetrapod diversification according to the long nuclear gene rag-1. *Systematic Biology*, 56(4):543–563, 2007.

[274] William G Weisburg, Stephen J Giovannoni, and Carl R Woese. The Deinococcus-Thermus phylum and the effect of rRNA composition on phylogenetic tree construction. *Systematic and Applied Microbiology*, 11:128–134, 1989.

[275] Serita M Nelesen, Kevin Liu, Li-San Wang, C. Randal Linder, and Tandy

Warnow. DACTAL: divide-and-conquer trees (almost) without alignments. *Bioinformatics*, 28(12):i274—-i282, 2012.

[276] Bryn Dentinger, Ester Gaya, Heath O'Brien, Laura M Suz, Robert Lachlan, Jorge R DíazValderrama, Rachel A Koch, and M Catherine Aime. Tales from the crypt: genome mining from fungarium specimens improves resolution of the mushroom tree of life. *Biological Journal of the Linnean Society*, 2015.

[277] Mark P Simmons and John Gatesy. Coalescence vs. concatenation: sophisticated analyses vs. first principles applied to rooting the angiosperms. *Molecular Phylogenetics and Evolution*, 2015.

[278] Thomas C Giarla and Jacob A Esselstyn. The Challenges of Resolving a Rapid, Recent Radiation: Empirical and Simulated Phylogenomics of Philippine Shrews. *Systematic Biology*, page syv029, 2015.

[279] Ya Yang, Michael J Moore, Samuel F Brockington, Douglas E Soltis, Gane Ka-Shu Wong, Eric J Carpenter, Yong Zhang, Li Chen, Zhixiang Yan, Yinlong Xie, Rowan F Sage, Sarah Covshoff, Julian M Hibberd, Matthew N Nelson, and Stephen A Smith. Dissecting molecular evolution in the highly diverse plant clade Caryophyllales using transcriptome sequencing. *Molecular Biology and Evolution*, page msv081, 2015.

[280] Christopher E Laumer, Andreas Hejnol, and Gonzalo Giribet. Nuclear genomic signals of the 'microturbellarian' roots of platyhelminth evolutionary innovation. *eLife*, 4, 2015.

[281] Peter A Hosner, Edward L Braun, and Rebecca T Kimball. Land connectivity changes and global cooling shaped the colonization history and diversification of New World quail (Aves: Galliformes: Odontophoridae). *Journal of Biogeography*, 2015.

[282] Corrinne E Grover, Joseph P Gallagher, Josef J Jareczek, Justin T Page, Joshua A Udall, Michael A Gore, and Jonathan F Wendel. Re-evaluating the phylogeny of allopolyploid Gossypium L. *Molecular Phylogenetics and Evolution*, 2015.

[283] Margaret H Frank, Molly B Edwards, Eric R Schultz, Michael R McKain, Zhangjun Fei, Iben Sø rensen, Jocelyn K C Rose, and Michael J Scanlon. Dissecting the molecular signatures of apical celltype shoot meristems from two ancient land plant lineages. *New Phytologist*, 2015.

[284] Renata Schama, Nicolás Pedrini, M Patricia Juarez, David R Nelson, André Q Torres, Denise Valle, and Rafael D Mesquita. Rhodnius prolixus Supergene Families of Enzymes Potentially Associated with Insecticide Resistance. *Insect Biochemistry and Molecular Biology*, 2015.