

© 2025 Seyedeh Yasamin Tabatabaeef

NOVEL COMPUTATIONAL METHODS FOR DISCORDANCE-AWARE
PHYLOGENOMIC ANALYSIS

BY

SEYEDEH YASAMIN TABATABAEE

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Computer Science
in the Graduate College of the
University of Illinois Urbana-Champaign, 2025

Urbana, Illinois

Doctoral Committee:

Professor Tandy Warnow, Chair and Director of Research
Associate Professor Mohammed El-Kebir
Professor William Gropp
Assistant Professor Ge Liu
Professor Siavash Mirarab, University of California San Diego

ABSTRACT

Inferring the evolutionary history of a set of species is a key step in many biological and medical research projects, as species trees provide a context in which problems in comparative genomics, biodiversity, phylogeography and epidemiology can be addressed. Recent advances in sequencing technologies have led to an increasing availability of genome-scale data, and today phylogenomics projects construct species trees using hundreds to thousands of loci, potentially whole genomes. However, species tree estimation from multi-locus datasets presents several statistical and computational challenges, as most problems in this area are NP-hard. Also, due to a phenomenon known as “gene tree heterogeneity”, different locations within the genome of a species can evolve differently due to biological processes such as incomplete lineage sorting, gene duplication and loss and horizontal gene transfer, that further complicate species tree estimation. Despite advances in developing methods that can estimate an unrooted and non-parameterized topology of a species tree in the presence of gene tree discordance, less attention has been paid to estimating the root location, quantifying branch lengths in units that are usable for downstream analysis, and estimating divergence times. All of these are necessary for many applications of phylogenomics, such as constructing the tree of life and analyzing the origins of diseases, such as HIV and COVID-19. In this dissertation, we introduce new computational methods developed for these tasks, collectively referred to as “post-species tree analysis”, that address different sources of gene tree discordance. For these methods, we present rigorous theoretical results including proofs of statistical consistency, sample complexity, and running time analyses, as well as extensive empirical results on simulated and biological datasets ranging from the root of the tree of life to recent speciations. Overall, these methods provide high accuracy and scalability for estimating the root, branch lengths and divergence times in the presence of gene tree discordance, and some are accompanied with strong theoretical guarantees.

To my family

ACKNOWLEDGMENTS

This dissertation would not have been possible without the help of many people who supported me throughout my PhD journey and provided opportunities for my professional and personal growth.

First, I sincerely thank my advisor, Dr. Tandy Warnow, for her continuous support and guidance throughout my PhD studies. Her dedication, rigor, and passion for science have been a constant source of inspiration for me. I feel extremely fortunate to have been mentored by her and to have had the opportunity to learn research from her. She generously invested her time in my training, offering guidance, constructive feedback, and numerous opportunities to engage with other researchers, that enriched my development as a scientist. Without her encouragement and support early in my graduate studies, much of what I have accomplished would not have been possible.

I am deeply grateful to Dr. Siavash Mirarab, who has been a close collaborator and mentor since the early stages of my graduate studies, and who supervised many of the papers in this dissertation. Much of my understanding of biology stems from what I have learned from him. I am grateful for the opportunities he provided, enabling my work to be used by the biological research community, and for facilitating collaborations with other researchers.

I thank other members of my qualifying, preliminary and dissertation committees, Dr. Nancy Amato, Dr. Arindam Banerjee, Dr. Bill Gropp, Dr. Mohammed El-Kebir and Dr. Ge Liu, for their guidance and valuable feedback on my PhD work. I would especially like to thank Dr. Mohammed El-Kebir, whose Computational Cancer Genomics course during the first year of my graduate studies deepened my passion for computational biology and introduced me to many interesting ideas in the field. I am also grateful for his continuous support and constructive feedback on my PhD research throughout my graduate studies.

I am grateful to the many faculty members with whom I had the opportunity to collaborate during my graduate and undergraduate studies, including Dr. Sebastien Roch, Dr. Chao Zhang, Dr. Santiago Claramunt, Dr. Steven N. Evans, Dr. George Chacko, Dr. Minh Bui, Dr. Djordje Jevdjic, and Dr. Abolfazl Motahari. I appreciate the support I received from them in my research and the knowledge and perspectives I gained through our interactions. I am beyond grateful to Dr. Darko Marinov for the many hours he spent meeting with me and for the helpful conversations we had when I was having a difficult time in graduate school. I thank Dr. Mariana Silva for the amazing experience of working with her as a teaching assistant in Fall 2025 semester. I thank Dr. Ben Raphael, Dr. Sriram Sankararaman, Dr.

Rasmus Nielsen, Dr. Jian Ma, their lab members, and other faculty at the Department of Computational Biology at Carnegie Mellon University for their hospitality and for inviting me to present my work in their labs and departmental seminars. I also thank all others from whom I have received guidance and feedback throughout my PhD studies, including Dr. Erin Molloy, Dr. Carl Kingsford, and Dr. Matthew Hahn.

I thank current and former members of the Warnow lab, Gillian Chu, Dr. Baqiao Liu, Minhyuk Park, Dr. Chengze Shen, The-Anh Vu-Le, Eleanor Wedell, James Willson, and Dr. Paul Zaharias, many of whom I had the pleasure of working with, for their support and for the productive collaborations we had. I am also grateful to the members of the Mirarab lab at UCSD, where I was hosted during the summer of 2024, for their helpful feedback on my PhD research, particularly Shayesteh Arasti and Dr. Puoya Tabaghi, with whom I collaborated closely. I thank Dr. Minh Bui and Nhan Ly-Trong from the Australian National University for providing me the opportunity to collaborate with them during the spring and summer of 2025, and for introducing me to new topics in phylogenetics.

I am also grateful to the members of the graduate office in the Siebel School, including but not limited to Viveka Kudaligama, Kara MacGregor, Cassandra Phelps, Jennifer Comstock, Dr. Lana Lazebnik, and Dr. Robin Kravets, for their hard work which made my time as a graduate student easier and more productive.

I am grateful for the financial support of the C.L. and Jane Liu Award from the Siebel School of Computing and Data Science, Dissertation Completion Fellowship and Firdawsi Science Award from the Graduate College, and the Mavis Future Faculty Fellowship from the Grainger College of Engineering at UIUC that supported me throughout my graduate studies. The research presented in this dissertation has been supported in part by the Grainger Foundation Breakthroughs Initiative to Tandy Warnow, the National Institute of Health grant #1R35GM142725 to Siavash Mirarab, and the National Science Foundation grants #1845967, #1636933, and #1920920 to Siavash Mirarab.

The computational experiments in this dissertation were mainly performed on the University of Illinois Campus Cluster that is a computing resource supported by funds from UIUC and is operated by the Illinois Campus Cluster Program in conjunction with the National Center for Supercomputing Applications. In addition, this work used Expanse at San Diego Supercomputing Center through allocation ASC150046 from the Advanced Cyberinfrastructure Coordination Ecosystem Services and Support (ACCESS) program, which is supported by the U.S. National Science Foundation grants #2138259, #2138286, #2138307, #2137603, and #2138296.

I cannot thank my parents, Zohreh and Mohammad, enough. Despite being separated by distance over the past four years, I spoke with them every single day. They instilled

in me a passion for science during my childhood, did all they could to provide me with opportunities for a good education, and encouraged me to pursue a career in STEM and follow my aspirations. I am grateful to my brother Ali, who has been a constant source of support and encouragement, and whose interest in math and computer science inspired mine. I cannot thank my aunt Shahla enough for always being there for me, filling the gap left by my family's absence, and for traveling to the U.S. multiple times each year to visit me. I thank Shahrnaz and Sassan for their hospitality and for inviting me to their home in California whenever I needed a respite from graduate work. I am also grateful to many other family members who have supported me throughout the years. I would like to thank several friends, from my undergraduate and graduate studies and earlier, especially Ava, Tina, Rozhin, Shakiba, Phoenix, and Tahereh, with whom I have stayed in close touch, for their continuous support. I especially thank Amir for all his support over the past year, and for helping me learn machine learning research.

TABLE OF CONTENTS

| | |
|------------------------------------------------------------------------------------------------------------|-----|
| CHAPTER 1 INTRODUCTION | 1 |
| CHAPTER 2 BACKGROUND | 5 |
| 2.1 Phylogenetic Trees | 5 |
| 2.2 Gene Trees and Species Trees | 10 |
| 2.3 Phylogenomics Pipeline | 11 |
| 2.4 Models of Evolution | 12 |
| 2.5 Gene Tree Estimation | 20 |
| 2.6 Species Tree Estimation | 23 |
| 2.7 Post-Species Tree Analysis | 29 |
| CHAPTER 3 ROOTING SPECIES TREES UNDER THE MULTI-SPECIES COALESCENT MODEL | 33 |
| 3.1 Introduction | 33 |
| 3.2 Background | 34 |
| 3.3 Quintet Rooting | 37 |
| 3.4 Experimental Study | 41 |
| 3.5 Results and Discussion | 46 |
| 3.6 Conclusions | 56 |
| 3.7 Methods and Software Commands | 57 |
| 3.8 Additional Figures and Tables | 61 |
| CHAPTER 4 POLYNOMIAL-TIME STATISTICALLY CONSISTENT ROOTING OF SPECIES TREES UNDER THE COALESCENT | 73 |
| 4.1 Introduction | 73 |
| 4.2 Background | 75 |
| 4.3 QR-STAR | 80 |
| 4.4 Theoretical Results | 83 |
| 4.5 Experimental Study | 103 |
| 4.6 Conclusions | 112 |
| 4.7 Methods and Software Commands | 114 |
| 4.8 Additional Figures and Tables | 115 |
| CHAPTER 5 PHYLOGENOMIC BRANCH LENGTH ESTIMATION USING QUARTETS | 125 |
| 5.1 Introduction | 125 |
| 5.2 Theoretical Results | 127 |
| 5.3 CASTLES | 150 |

| | | |
|----------------------------------------------------------------------------------------------------------------------------|-----------------------------------------|------------|
| 5.4 | Experimental Study | 153 |
| 5.5 | Results | 156 |
| 5.6 | Discussion | 162 |
| 5.7 | Conclusion | 165 |
| 5.8 | Methods and Software Commands | 166 |
| 5.9 | Additional Figures and Tables | 170 |
| CHAPTER 6 SPECIES TREE BRANCH LENGTH ESTIMATION DESPITE INCOMPLETE LINEAGE SORTING, DUPLICATION, AND LOSS | | 187 |
| 6.1 | Introduction | 187 |
| 6.2 | CASTLES-Pro | 189 |
| 6.3 | Experimental Study | 198 |
| 6.4 | Results | 204 |
| 6.5 | Discussion and Conclusions | 212 |
| 6.6 | Methods and Software Commands | 216 |
| 6.7 | Additional Figures and Tables | 219 |
| CHAPTER 7 COALESCENT-BASED BRANCH LENGTH ESTIMATION IM- PROVES DATING OF SPECIES TREES | | 242 |
| 7.1 | Introduction | 242 |
| 7.2 | Material and Methods | 245 |
| 7.3 | Results | 251 |
| 7.4 | Discussion and Conclusions | 261 |
| 7.5 | Methods and Software Commands | 263 |
| 7.6 | Additional Figures and Tables | 268 |
| CHAPTER 8 CONCLUSIONS | | 291 |
| REFERENCES | | 293 |
| APPENDIX A LIST OF ABBREVIATIONS | | 323 |
| APPENDIX B LIST OF TERMS | | 326 |

CHAPTER 1: INTRODUCTION

Phylogenomics – estimating evolutionary histories using genome-scale data– has enabled inference of species trees across several domains of life and addressed long-standing questions in evolutionary biology [1, 2, 3]. Reconstructing species trees is a critical step in many areas of biology and biomedical research, and provides an essential framework for investigations in comparative genomics [4], biodiversity analysis [5], phylogeography [6], and infectious disease epidemiology [7]. While early phylogenetic studies relied on morphological traits or data from a single gene or a few loci [8, 9] to reconstruct species trees, recent advances in sequencing technologies have enabled the collection of genome-scale datasets. Modern phylogenomic analyses now routinely use hundreds to thousands of loci, and in some cases, entire genomes, to infer species trees with unprecedented resolution [10, 11, 12].

A major challenge in estimating species trees using genome-scale datasets is the pervasive discordance among evolutionary histories of individual loci, a phenomenon known as “gene tree heterogeneity” [13, 14]. Gene tree heterogeneity (or incongruence) can be caused by several biological processes, such as Incomplete lineage sorting (ILS), Gene duplication and loss (GDL), Horizontal gene transfer (HGT), hybridization, introgression, recombination, and gene flow [15, 16, 17, 18]. Some of these processes, such as ILS and GDL, result in evolutionary histories of individual genomic regions (i.e., gene trees) being different from the evolutionary history of the species as a whole (i.e., species tree), but still agree with a tree-like pattern of evolution. Others, such as hybridization and introgression, suggest reticulate evolutionary histories and are modeled and studied in the context of phylogenetic networks rather than trees [19, 20]. Among all sources of gene tree discordance, ILS is the most extensively studied, given its widespread impact across the tree of life [21, 22, 23, 24].

A traditional approach to phylogenomic analysis is to concatenate the sequence alignments of all genes into a super-matrix, and then use a typical phylogeny inference method, such as maximum likelihood [25, 26] or distance-based methods [27] to infer the species tree from this super-alignment. However, this approach, referred to as “concatenation”, inherently assumes that all genomic regions have the same evolutionary history, which is an incorrect assumption when gene tree discordance occurs. In fact, it has been shown that concatenation with maximum likelihood is statistically inconsistent in the presence of ILS, and can even be positively misleading [28, 29], meaning that its output can converge to the wrong tree topology with probability going to 1 as the number of genes increase. In practice, the performance of concatenation depends on the amount of gene tree discordance and it can have poor accuracy in the presence of substantial levels of ILS [30, 31, 32, 33, 34].

To address the theoretical and practical limitations of concatenation, several alternative species tree estimation methods have been developed that explicitly model gene tree discordance. The majority of these methods account for **ILS**, that can be mathematically modeled by the **Multi-Species Coalescent (MSC)** model [35, 36]. One group of these methods co-estimate gene trees and species trees [37, 38, 39], mostly using Bayesian **MCMC**, and can be statistically consistent under the **MSC** and have very good performance in **ILS** simulations [40]. A major drawback of co-estimation methods, however, is their scalability: these methods can be computationally intensive even on datasets with only 50 species [38, 41], largely limiting their application in practice. A second group of methods, referred to as “site-based methods” [42, 43, 44], directly estimate a species tree from sequence data while accounting for **ILS**, without a need to estimate gene trees, and can be statistically consistent under the **MSC** [45] model.

Finally, the most widely-adopted and scalable coalescent-based species tree estimation methods, known as “summary methods”, first estimate a gene tree for each individual gene alignment and then combine these gene trees into the species tree while accounting for **ILS** [46, 47, 48, 49]. In particular, the summary method ASTRAL and its family [50, 51, 52, 53, 54, 55, 56, 57] have shown to be accurate in practice when gene tree incongruence occurs [32, 58], and are now in widespread use by the biological research community and have been utilized in many studies [59, 60, 61, 62]. In addition, ASTRAL and many other coalescent-based summary methods, such as ASTRID/NJst [47, 49], that have also shown good accuracy in practice [32], are proven to be statistically consistent under the **MSC** [46, 48, 49, 63].

Despite their good performance and scalability, summary methods such as ASTRAL and ASTRID can have limitations. These methods output an *unrooted* tree topology *without branch lengths*, or with lengths in units that are not useful for downstream analysis. Such a tree is not readily useful for biologists. Most downstream applications of species trees, such as comparative genomics [64], characterization of selection [65], comparative trait analysis [66, 67] and viral phylodynamics [68] require a *rooted* tree with branch lengths in the unit of expected number of substitutions per sequence site, or in the unit of time (referred to as a *dated* tree). Since the output of summary methods (or in general most species tree estimation methods) do not produce these quantities, biologists often use additional tools, which we refer to as “post-species tree analysis” tools (Fig. 2.1), to get a *rooted species tree with branch lengths and dates*. However, almost all existing methods for post-species tree analysis ignore gene tree heterogeneity across the genome, a shortcoming that can lead to poor performance and biases in practice [69, 70, 71, 72, 73, 74, 75, 76].

Several studies have shown that the most accurate methods for estimating the topology of a species tree in the presence of gene tree incongruence, especially **ILS**, are methods

that explicitly take this discordance into account [32, 33, 50, 58, 77], and concatenation becomes less accurate as the amount of discordance increases. However, existing pipelines for post-species tree analysis of trees produced by summary methods still use some form of concatenation analysis and overlook gene tree heterogeneity. For example, the common approach for estimating branch lengths on tree topologies produced by summary methods is concatenation, and most methods for rooting and dating species trees use these branch lengths, and hence are indirectly based on some form of concatenation analysis [60, 78, 79, 80]. This is unsatisfactory, as concatenation is both less scalable than gene-tree-based methods and ignores heterogeneity across genes; therefore, it may suffer in terms of accuracy, and certainly in terms of statistical guarantees.

The goal of this dissertation is to address these gaps in species tree estimation methods by proposing novel scalable approaches for phylogenomic analysis, specifically post-species tree analysis, that explicitly model gene tree heterogeneity across the genome. In particular, the main focus is on three related problems: *rooting*, *branch length estimation* and *dating* of species trees. Unlike existing methods for these tasks that are ignorant of gene tree incongruence, our proposed approaches model discordance between gene trees and species trees to enable more accurate and robust estimation of these quantities under complex evolutionary histories. The methods proposed in this dissertation all have strong theoretical foundation based on the [MSC](#) or other models of gene evolution such as [GDL](#), and some come with guarantees of statistical consistency. However, what mainly distinguishes these methods from previous methods for these problems is not their statistical properties, accuracy or scalability, but that they are *discordance-aware*, meaning that they explicitly model conflicting evolutionary histories across the genome.

The rest of this chapter provides an overview of the main contributions of the dissertation.

We first study the problem of rooting an unrooted species tree given a set of unrooted gene trees, under the assumption that gene trees evolve within the model species tree under the [MSC](#). We present Quintet Rooting (QR) in Chapter 3, a method for rooting species trees based on a proof of identifiability of the rooted species tree under the [MSC](#) established by Allman, Degnan and Rhodes [81]. Our results show that QR is generally more accurate than other rooting methods on datasets with moderate levels of [ILS](#), except under extreme levels of gene tree estimation error. We next turn to the problem of statistical consistency, and present QR-STAR in Chapter 4, a variant of QR with an additional step and a different cost function, and prove that it is statistically consistent under the [MSC](#). Moreover, we derive sample complexity bounds for QR-STAR and show that a particular variant of it based on “short quintets” has polynomial sample complexity. Finally, our simulation study under a variety of model conditions shows that QR-STAR matches or improves on the accuracy of

QR.

We next address the problem of branch length estimation and introduce CASTLES in Chapter 5, a new technique for estimating branch lengths of the species tree from estimated gene trees that uses expected values of gene tree branch lengths in substitution units under an extension of the MSC model that allows substitutions with varying rates across the species tree branches. Our simulation study shows that CASTLES improves on the most accurate prior methods for branch length estimation with respect to both speed and accuracy. We then present CASTLES-Pro in Chapter 6 that extends CASTLES to handle gene duplication and loss, and improves its accuracy on datasets with single-copy gene trees. Our simulation studies show that CASTLES-Pro is generally more accurate than alternatives, eliminating the systematic bias toward overestimating terminal branch lengths often observed when using concatenation. Moreover, while not theoretically designed for horizontal gene transfer, our results show that CASTLES-Pro maintains relatively high accuracy under high rates of HGT.

Finally, we study the problem of estimating divergence times on a species tree in Chapter 7, and introduce a new scalable pipeline for dating species trees that addresses gene tree discordance for both topology and branch length estimation. The pipeline uses discordance-aware methods that account for incomplete lineage sorting for estimating the topology and branch lengths and maximum likelihood-based methods for the dating step. Our simulation study on datasets with gene tree discordance shows that this pipeline produces more accurate and less biased dates than pipelines that use concatenation or unpartitioned Bayesian methods. Furthermore, it is substantially more scalable and can handle datasets with thousands of species and genes. Our results on two biological datasets show that this new pipeline improves the inference of node ages and branch lengths for some nodes, in particular extant taxa, and improves the downstream diversification analysis. We conclude in Chapter 8 with a summary of contributions and discussion of future research directions.

Overall, the methods developed in this dissertation aim to extend the capabilities of species tree estimation methods, in particular summary methods, beyond unrooted topology reconstruction, enabling accurate inference of the root, branch lengths, and divergence times, while taking sources of gene tree discordance into consideration. We hope that the methods and the theoretical results established here can contribute to a more robust and comprehensive framework for species tree estimation that can be used in large-scale evolutionary analysis.

CHAPTER 2: BACKGROUND

This chapter reviews the background material and terminology used throughout this dissertation. Section 2.1 introduces phylogenetic trees and their comparison from a graph-theoretic perspective. Section 2.2 discusses the distinction between gene trees and species trees. Section 2.3 outlines the main steps in a typical phylogenomic analysis pipeline. Section 2.4 reviews models of sequence and gene evolution, as well as the phenomenon of gene tree discordance. Sections 2.5 and 2.6 survey existing methods for gene tree and species tree estimation. Finally, Section 2.7 presents an overview of methods for rooting species trees, estimating branch lengths, and inferring divergence times. If a key term is defined in the text, it is italicized in addition to being highlighted in blue to indicate a hyperlink to its corresponding entry in the Appendix. The first occurrence of each term in subsequent chapters and all abbreviations are also highlighted and linked to the Appendix.

2.1 PHYLOGENETIC TREES

2.1.1 Notations and Terminology

A *phylogenetic tree* $T = (V, E)$ is a leaf-labeled connected acyclic graph with vertex set V , edge set E , and leafset $\mathcal{L}(T) \subseteq \mathcal{X}$ where \mathcal{X} is a finite set of *taxa*. A phylogenetic tree is a mathematical or graphical representation of a *phylogeny*, i.e., the evolutionary history of a set of taxa, which can represent different biological entities, such as species, genes, or populations. T is called *singly-labeled* if each label appears exactly once in $\mathcal{L}(T)$, and is called *multi-labeled* or a *MUL-tree* if some labels appear more than once in $\mathcal{L}(T)$. The edges that are adjacent to the leaves of T are referred to as *terminal* edges, and all other edges are called *internal* edges. Similarly, all nodes other than the leaves are called *internal* nodes. The *terminal* nodes or *tips* of T are the set of leaf nodes. We assume that all internal nodes have degree greater than two. The branches of T can be furnished with edge lengths in different units, for example, representing evolutionary time or genetic change, in which case T is referred to as a *metric* phylogenetic tree. The branch lengths can be represented by a function $\lambda : E \rightarrow \mathbb{R}_{\geq 0}$ that assigns non-negative lengths to each edge, and (T, λ) represents a *weighted* phylogenetic tree. When tree T is weighted, the *topology* of T refers to its graphical model without the edge lengths.

A *rooted* phylogenetic tree is a *directed* tree with a special node $r \in V/\mathcal{L}(T)$ designated as *root* (denoted by $r(T)$), where every edge is directed away from the root. An *unrooted*

phylogenetic tree is an undirected graph with no designated root. In a rooted phylogenetic tree, the *out-degree* of a node u , denoted by $\deg^+(u)$, is the number of directed edges leaving u and the *in-degree* of u , denoted by $\deg^-(u)$, is the number of edges entering u . Note that $\deg^-(r) = 0$ and $\forall v \neq r : \deg^-(v) = 1$ since every node other than the root has a unique parent. A rooted phylogenetic tree T defines a partial order \preceq_T on $V(T)$ where $u \preceq_T v$ if there is a directed path from root $r(T)$ to node v that passes through node u . When $u \preceq_T v$, we say node u is an *ancestor* of node v and similarly v is a *descendant* of node u . The *most recent common ancestor (MRCA)* of a set of nodes $S \in V$ is the *unique* node $m \in V$ such that $m \preceq_T v$ for all $v \in S$, which means that m is an ancestor of all nodes in S , and for every other node $m' \in V$ such that $m' \preceq_T v$ for all $v \in S$, it holds that $m \preceq_T m'$, and therefore m is the *lowest* node with this property. The *internode distance* between two leaves u, v in T is the number of vertices on the unique path between u and v , not including the endpoints. The *patristic distance* between u and v in a *metric* phylogenetic tree (T, λ) is defined similarly, as the sum of the branch lengths along the unique path between nodes u and v .

An unrooted phylogenetic tree is called *fully resolved* (also *binary* or *bifurcating*) if all internal nodes have degree 3, and *unresolved* (or *multifurcating*) if there is an internal node with degree greater than 3. Similarly, a rooted phylogenetic tree is resolved if the root has degree 2 and all internal nodes have in-degree 1 and out-degree 2. A *polytomy* is a node that represents multifurcation events in a phylogenetic tree, that has degree greater than three in unrooted trees and out-degree greater than two in rooted trees. A *star tree* is a tree with no internal edges that has a single polytomy. A *contraction* is an operation that reduces the resolution in a phylogenetic tree, and *refinement* is the reverse operation that adds resolution to an unresolved tree. The *contraction* of an edge $e = (u, v) \in E(T)$ results in the tree T' where nodes u and v are merged into a single node and edge e is removed, hence creating a polytomy. A *refinement* is the reverse of contraction, where a polytomy is turned into a resolved subtree by adding new internal nodes and edges. The tree T' is a refinement of T if T can be obtained from T' by a sequence of edge contractions. Unless stated otherwise, we assume throughout the proofs in this dissertation that all phylogenetic trees are fully resolved.

An unrooted phylogenetic tree can be rooted by selecting one of its edges, inserting a new node (that will be designated as root) on that edge, connecting the endpoints of the selected edge to the new node, and directing all edges away from the root. A fully resolved rooted phylogenetic tree T has $|V| = 2n - 1$ nodes and $|E| = 2n - 2$ edges, and an unrooted binary phylogenetic tree T has $|V| = 2n - 2$ nodes and $|E| = 2n - 3$ edges where $n = |\mathcal{L}(T)|$ is the number of leaves of T . For a subset of taxa $S \in \mathcal{L}(T)$, the *restriction* of T to S ,

denoted $T|_S$ (also referred to as *induced* subtree), can be obtained by taking the subtree of T that connects all leaves in S and *suppressing* (removing) all internal nodes of degree two. For a rooted tree T and a node $v \in V(T)$, we denote the subtree of T below node v by T_v . Two unrooted phylogenetic trees T and T' on label set S are *compatible* if T' is a contraction of $T|_{S'}$ for a set $S' \subseteq S$. We say T' *agrees* with T if T' is isomorphic to $T|_{S'}$. Otherwise, we say T' *disagrees* with T . Note that by definition, tree agreement implies tree compatibility. We can similarly define compatibility for a set of phylogenetic trees, which motivates the definition of a *supertree*, i.e., a tree that combines information from a collection of phylogenetic trees.

Definition 2.1 (Supertree). Let $\mathcal{T} = \{T_1, T_2, \dots, T_k\}$ be a collection of phylogenetic trees where each T_i is defined on a label set \mathcal{L}_i , and let $\mathcal{L} = \bigcup_{i=1}^k \mathcal{L}_i$. A phylogenetic tree T^* on label set \mathcal{L} is called a *supertree* for the set \mathcal{T} if for every $i \in \{1, \dots, k\}$, the restriction of T^* to \mathcal{L}_i , denoted $T^*|_{\mathcal{L}_i}$, is compatible with T_i .

Supertrees are particularly useful when we have a collection of phylogenetic trees defined on different (and possibly overlapping) sets of taxa, often estimated from distinct genomic loci or data partitions, and wish to combine this information into a single phylogeny. See Sec. 2.2 and Sec. 2.6 for further discussion.

We now define the set of *clades* and *bipartitions* of a phylogenetic tree.

Definition 2.2 (Clade). For a rooted phylogenetic tree T , the clades of T are given by the set $\text{Clades}(T) = \{\mathcal{L}(T_v) | v \in V(T)\}$. The clades that appear in every possible tree on the leafset $\mathcal{L}(T)$, i.e., the clade containing all of $\mathcal{L}(T)$ and the clades containing singleton elements $s \in \mathcal{L}(T)$, are called *trivial* clades, and the rest of the clades are called *non-trivial* clades.

A phylogenetic tree can be uniquely determined from its set of clades by constructing a *Hasse diagram* from the partial order defined by this set (see Sec. 2.2.3 in [82] for details). Therefore, we can use the set of clades to compare two rooted phylogenetic trees. The rooted trees T and T' are identical if and only if $\text{Clades}(T) = \text{Clades}(T')$. We can use a similar technique to encode unrooted phylogenetic trees.

Definition 2.3 (Bipartition). For an unrooted phylogenetic tree T , the *bipartition encoding* (or *split* encoding) of T is given by the set $\text{Bip}(T) = \{\pi(e) | e \in E(T)\}$ where $\pi(e) = A|B$ denotes the bipartition on $\mathcal{L}(T)$ produced by removing edge e (while preserving its endpoints) from T , where A and B are the set of leaves in each half of the bipartition. The bipartitions that appear in every possible tree on the leafset $\mathcal{L}(T)$, i.e., the bipartitions

that have a single leaf in one side with the rest of nodes in the other side, are called *trivial* bipartitions, and the rest of the bipartitions are called *non-trivial*.

According to this definition, there is a bijection between $\text{Bip}(T)$ and the set of internal edges of T . Similar to clades for rooted trees, the set of bipartitions can be used to compare two unrooted phylogenetic trees. The unrooted trees T and T' are identical if and only if they have the same set of bipartitions, so that $\text{Bip}(T) = \text{Bip}(T')$. T' is compatible with T if and only if $\text{Bip}(T') \subseteq \text{Bip}(T|_R)$ for $R \subseteq \mathcal{L}(T)$. Each internal edge in an unrooted tree corresponds to a bipartition. An unrooted binary phylogenetic tree with four leaves has only one bipartition and is referred to as a *quartet*. Note that we sometimes use *quartet* to refer to a set of four taxa sampled from $\mathcal{L}(T)$. The set of all $\binom{n}{4}$ induced four-taxon subtrees of an unrooted tree T is denoted by $Q(T)$. Note that there is no unrooted fully resolved phylogenetic tree with three taxa. A *triplet* is a rooted phylogenetic tree with three leaves, and the set of all induced three-taxon subtrees of rooted tree T is denoted by $\text{Triplet}(T)$.

2.1.2 Comparison of Phylogenetic Trees

While the set of clades and bipartitions of rooted and unrooted phylogenetic trees can be used to determine if they are identical, we are generally interested in measuring the difference between two trees. A popular metric for quantifying the distance between two unrooted phylogenetic trees is the *Robinson-Foulds (RF)* [83] distance (also known as *bipartition* distance).

Definition 2.4 (Robinson-Foulds (RF) distance). For two unrooted phylogenetic trees T and T' on the same leafset $\mathcal{L}(T)$, the **RF** distance between T and T' , denoted by $RF(T, T')$ is defined as

$$RF(T, T') = |\text{Bip}(T) \Delta \text{Bip}(T')| = |\text{Bip}(T) \setminus \text{Bip}(T')| + |\text{Bip}(T') \setminus \text{Bip}(T)| \quad (2.1)$$

When T and T' are fully resolved, each of them has $|\mathcal{L}(T)| - 3$ internal edges. The maximum possible value for $RF(T, T')$ is $2|\mathcal{L}(T)| - 6$, that happens when T and T' have no internal edges in common, and the minimum value is 0 when T and T' are identical. We usually normalize the **RF** distance by its maximum possible value, to get the *normalized RF distance* or the *RF error rate*, defined as

$$\frac{RF(T, T')}{2|\mathcal{L}(T)| - 6} \quad (2.2)$$

In simulation studies, we are usually interested in comparing the *true* (or ground-truth) phylogenetic tree T^* with an *estimated* tree T . In this case, $\frac{|\text{Bip}(T^*) \setminus \text{Bip}(T)|}{|\text{Bip}(T^*)|}$ is referred to as the *false negative (FN)* rate (also called *missing branch rate*), that is the proportion of the total number of non-trivial bipartitions that are in the ground-truth tree but are missing from the estimated tree. Similarly, the *false positive (FP)* rate is defined as $\frac{|\text{Bip}(T) \setminus \text{Bip}(T^*)|}{|\text{Bip}(T)|}$, that is the fraction of edges in the estimated tree that are missing from the true tree.

In addition to comparing the topologies of two phylogenetic trees, we are often interested in comparing the branch lengths in two metric phylogenetic trees, for example, where these lengths represent genetic change or divergence time. In this dissertation, we use the following measures to compare the branch lengths of a metric ground-truth phylogenetic tree (T^*, λ^*) with an estimated tree (T, λ) on the same set of shared branches E .

- Bias: $\frac{1}{|E|} \sum_{e \in E} (\lambda_e^* - \lambda_e)$
- Mean absolute error: $\frac{1}{|E|} \sum_{e \in E} |\lambda_e^* - \lambda_e|$
- Logarithmic error: $\frac{1}{|E|} \sum_{e \in E} |log_{10}(\lambda_e^*) - log_{10}(\lambda_e)|$
- Root mean square error (RMSE): $\sqrt{\frac{1}{|E|} \sum_{e \in E} (\lambda_e^* - \lambda_e)^2}$

where λ_e^*, λ_e are the lengths of branch e in trees T^* and T respectively.

In addition to comparing two phylogenetic trees, it is often of interest to evaluate a single tree in isolation, for example, by measuring its height (commonly referred to as the *time of the most recent common ancestor (tMRCA)* in molecular dating studies), or by examining the distribution of internal versus terminal branch lengths. Metrics such as *treeness* and *root-to-tip distance* provide a way to quantify these distributions.

Definition 2.5 (Treeness). For a metric phylogenetic tree (T, λ) , the *treeness* of T is defined as the sum of its internal branch lengths divided by sum of all branch lengths, i.e.,

$$\text{Treeness}(T) = \frac{\sum_{e \in E_I(T)} \lambda_e}{\sum_{e \in E(T)} \lambda_e} \quad (2.3)$$

where λ_e is the length of branch e and $E_I(T)$ is the set of internal branches of T .

Note that $0 \leq \text{Treeness}(T) \leq 1$. A higher treeness (value closer to 1) indicates that internal branches account for most of the tree's structure, and implies a strong phylogenetic signal. A lower treeness suggests that the tree is dominated by long terminal branches, that can result from rapid diversification or limited resolution of internal branches.

Definition 2.6 (Root-to-tip distance (RTT)). For a rooted metric phylogenetic tree (T, λ) , the average *root-to-tip (RTT)* distance is defined as

$$\text{RTT}(T) = \frac{1}{|\mathcal{L}(T)|} \sum_{u \in \mathcal{L}(T)} d(u, r(T)) \quad (2.4)$$

where $d(u, r(T))$ is the patristic distance between leaf u and $r(T)$, the root of T .

A rooted metric phylogenetic tree is called *ultrametric* if all leaves have the same root-to-tip distance. Examining the distribution of RTT distances can be useful for detecting rate heterogeneity across branches, calculating molecular divergences and testing deviations from the molecular clock (see Section 2.4 for more details).

2.2 GENE TREES AND SPECIES TREES

Gene trees and species trees are both phylogenetic trees, but they have different biological interpretations. A *gene tree* represents the evolutionary history of a segment of DNA (referred to as *gene* or *locus*) sampled across multiple species. This DNA sequence typically corresponds to a contiguous region of the genome and does not necessarily encode a functional protein-coding gene. Such regions are also referred to as *coalescent genes* or *c-genes* [84]. The length of a gene can vary, ranging from a few *nucleotides* (units of genetic information in a DNA, represented with *A*, *C*, *T*, or *G*) to tens of thousands of nucleotides. Over time, mutations in the DNA sequence of a gene can create different genetic variants called *alleles*. A gene tree can be formed by tracing the evolution of alleles backward in time, forming a *lineage* that is a sequence of ancestral-descendant relationships. *Coalescence* happens when two lineages merge at a common ancestor and form a single lineage. Therefore, terminal nodes in a gene tree represent alleles and internal nodes represent *coalescent* events.

Coalescent genes are assumed to evolve without *recombination*, meaning that all *sites* (positions) within their DNA sequence have the same evolutionary history. Recombination is the biological process in which genetic material is exchanged between parental DNA sequences during meiosis, creating a new combination of alleles in the offspring. When recombination occurs in a genomic region, the evolutionary history of that region can not be accurately represented by a single tree, and more complex structures such as *ancestral recombination graphs (ARGs)* [85] are used to represent the ancestral relationships.

A *species tree* represents the evolutionary history of a set of species, or a set of populations of individuals. The internal nodes of a species tree represent *speciation* events and

terminal nodes (leaves) correspond to extant species. Each internal branch of a species tree represents a population of alleles evolving over time between successive speciation events. It is important to note that the evolutionary history of individual genomic regions (gene trees) may differ from the evolutionary history of species as a whole [15]. This phenomenon, referred to as *gene tree heterogeneity*, poses a major challenge in phylogenomic studies and is discussed in detail in Section 2.4.

2.3 PHYLOGENOMICS PIPELINE

A typical phylogenomics workflow involves several steps (Figure 2.1). The process begins with data collection and preprocessing, which includes assembling genome or transcriptome sequences, identifying gene boundaries, and performing quality control procedures such as contamination removal [86]. These procedures ensure that only high-quality and biologically relevant data are used in downstream analyses. The second step in many phylogenomic studies is orthology inference [87], that is the identification of genes in different species that descend from the same gene in their most recent common ancestor. This step is crucial when using species tree estimation methods that rely on single-copy gene trees, but can be skipped when using recent methods that can work directly with multi-copy gene family trees [54, 58, 88, 89, 90]. Following orthology inference, the workflow proceeds to *multiple sequence alignment (MSA)* [91], where the nucleotide or amino acid sequences of each gene are aligned to identify homologous sites across species.

The next step in a phylogenomics pipeline is species tree estimation, which can be performed using different techniques. A traditional approach is the *concatenation* method, where all gene sequence alignments are concatenated into a super-matrix, and a phylogenetic tree is then estimated from this super-alignment, typically using maximum likelihood methods [25, 92]. While concatenation can be effective, it assumes that all gene sequences share the same underlying evolutionary history, an assumption that is often violated due to gene tree discordance. A more scalable technique for species tree estimation is using summary methods [93] that first estimate an individual gene tree for each gene alignment and then combine these gene trees into the species tree, while accounting for sources of gene tree discordance. The species tree estimation step typically produces only an unrooted tree topology, without useful branch lengths. Therefore, additional processing is needed for the tree to become suitable for biological interpretation. The final step, referred to as *post-species tree analysis*, is the main focus of this dissertation, and includes rooting, estimating branch lengths and estimating divergence times (dates) on the inferred species tree topology to make it useful for downstream biological analysis.

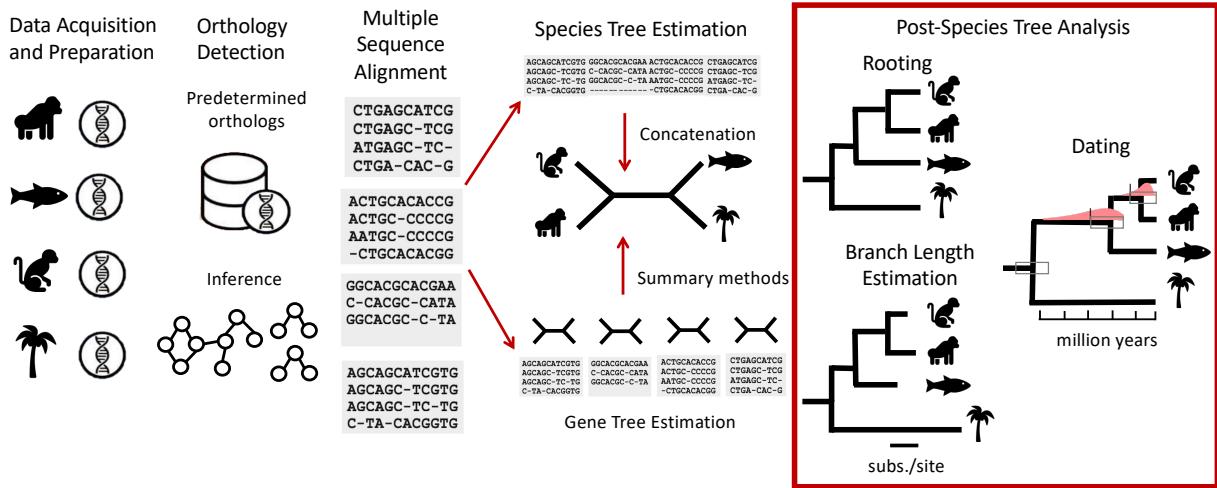


Figure 2.1: **A typical phylogenomics pipeline.** The typical steps in a phylogenomics workflow are 1) data gathering and preparation 2) orthology inference 3) multiple sequence alignment 4) species tree estimation and 5) post-species tree analysis that includes rooting, branch length estimation and dating, and is the main focus of this dissertation. While many methods for species tree estimation account for gene tree discordance, most existing approaches for post-species tree analysis overlook this complexity. This figure includes elements adapted from Figure 1 in [94], which is available under a Creative Commons Attribution license.

2.4 MODELS OF EVOLUTION

Genome-wide evolution can be modeled using a two-step generative hierarchical model [15, 18] where gene trees are sampled from a distribution defined by the parameters of the species tree and according to a *gene evolution* model, such as the **Multi-Species Coalescent (MSC)** [95], and then sequences evolve down each gene tree according to a *sequence evolution* model, such as the **Generalized Time Reversible (GTR)** model [96]. In this section, we briefly describe models of gene and sequence evolution commonly used in phylogenomic studies.

2.4.1 Models of gene evolution

Multi-species coalescent model. The **Multi-Species Coalescent (MSC)** [95, 97, 98] model is a probabilistic model that describes the evolution of gene trees inside a species tree, and is the multi-species extension of the coalescent process originally proposed by Kingman [97]. Under the **MSC**, the model species tree is parameterized by (T, λ) where T is a rooted phylogenetic tree, and λ is a set of numerical parameters for T , including the **effective population size** and the number of generations for each branch. Under this model, gene trees are assumed to evolve **i.i.d.** inside the species tree. The model species tree defines a specific

probability distribution on rooted and unrooted gene tree topologies, and these probabilities can be exactly calculated based on the numerical parameters of the species tree, λ . Under the **MSC**, all gene tree topologies have strictly positive probability when λ has positive values.

MSC models a population-level process that describes the way alleles coalesce in populations of individuals. Under the coalescent process, lineages from different genes may fail to coalesce within the population of their most recent common ancestor. As a result, multiple lineages can persist and enter deeper ancestral populations; in this case, any two lineages are equally likely to be the first to coalesce inside that branch. Therefore, coalescence may occur between more distantly related lineages before more closely related ones. This phenomenon is called *Incomplete lineage sorting (ILS)* or *deep coalescence*, that can result in a gene tree having a different topology than the species tree. The level of **ILS** in a simulated dataset can be calculated as the average normalized **RF** distance between the model species tree and true gene trees, which is denoted by *average distance (AD)* in this dissertation.

ILS is more likely to happen when a population has few generations (i.e., its corresponding branch is short), or when the population size is large, as both of these factors make it more unlikely for two lineages to coalesce within that population. In empirical datasets, **ILS** is frequently observed when the timing between consecutive speciation events is short, that is characterized by a succession of short internal branches in the species tree, a phenomenon known as *rapid radiation* [99]. Many major groups of species are known to have undergone rapid radiations, including birds [24], placental mammals [100], lizards [101] and bees [69, 102].

We say that the model species tree is in the *anomaly zone* when the most probable rooted gene tree under the **MSC** has a different topology than the species tree [103]. Similarly, the unrooted version of the species tree is called *anomalous* if its topology does not match that of the most probable unrooted gene tree [104]. There is no anomalous three-taxon rooted species trees or anomalous four-taxon unrooted species tree [81, 103, 105]. However, for all numbers $n \geq 4$ of leaves for rooted trees, and all numbers $n \geq 5$ for the unrooted case, there exists an n -leaf **MSC** model species tree, with some choice of numeric parameters, that lies in the anomaly zone [103].

The internal branches of the **MSC** model species tree are generally represented in *coalescent units (CU)*, and the length of a branch can be calculated using the formula

$$\tau = \frac{t}{2N_e} \tag{2.5}$$

where τ is the length in coalescent units, t is the number of generations, and N_e is the

effective population size for the branch.

Under the Kingman coalescent model [97], coalescence events follow a Poisson process, and waiting times between coalescent events are exponentially distributed with parameter rate $\frac{\binom{k}{2}}{N_e}$ where k is the number of lineages entering the interval between two coalescent events [106]. The probability of i lineages coalescing into j lineages after T coalescent time units is

$$g_{ij}(T) = \sum_{k=j}^i e^{-\frac{k(k-1)T}{2}} \frac{(2k-1)(-1)^{k-j}}{j!(k-j)!(j+k-1)} \prod_{x=0}^{k-1} \frac{(j+x)(i-x)}{i+x} \quad (2.6)$$

where $1 \leq j \leq i$ [107, 108]. Therefore, the probability of two lineages coalescing in an interval with length T is $g_{21} = 1 - e^{-T}$ and the probability of two lineages not coalescing in that interval is $g_{22} = e^{-T}$. Based on Equation 2.6, we can calculate the probability of each rooted and unrooted gene tree topology inside an **MSC** model species tree.

Probabilistic models of gene duplication and loss. *Gene duplication and loss (GDL)* is a major evolutionary process that can result in multiple copies of a gene being present in the genome of a species. **GDL** can give rise to gene family trees, also known as multi-copy gene trees or **MUL-trees**, in contrast to single-copy gene trees that contain exactly one homologous gene per species. In gene family trees, each leaf may represent a different copy of the gene from the same or different species. Two gene copies are referred to as *orthologs* if their most recent common ancestor corresponds to a speciation event and are called *paralogs* if their **MRCA** is a gene *duplication event*. While single-copy gene trees only have orthologous genes, multi-copy trees have paralogs in addition to orthologs. **GDL** is commonly seen in the genome of land plants [61, 109] and fish [110] among other groups.

Probabilistic models of gene duplication and loss model gene family evolution as a stochastic birth-death process along the branches of a species tree. These models are typically parametrized by a rooted species tree T , a *duplication rate* λ and a *loss rate* μ , and the gene tree is generated within the species tree according to these rates. A major model is the Duplication–Loss model of Arvestad et al. [111], that assumes independent duplication and loss events across lineages, but does not model ILS. An extension of this model is the joint model of **Duplication-Loss-Coalescence (DLCoal)** [112] that incorporates incomplete lineage sorting along with GDL.

Probabilistic models of horizontal gene transfer. *Horizontal gene transfer (HGT)*, also known as *lateral gene transfer (LGT)*, refers to the non-vertical transmission of genetic material between organisms, that is, the transfer of genes from one organism to another that

is not its offspring. Unlike vertical inheritance, where genetic information is passed from parent to offspring through reproduction, **HGT** involves the movement of genetic segments between unrelated organisms, sometimes even across different species or domains of life [113]. **HGT** is especially prevalent in prokaryotic organisms, including bacteria and archaea, and can introduce novel functions, such as antibiotic resistance or new metabolic capabilities [114, 115].

Different mathematical models of gene evolution incorporate horizontal transfer. Most of these models, such as DTLSR and ODT [116, 117] incorporate duplication, loss and transfer events together. Similar to the model proposed by Szöllősi et al., [118], we describe a birth-death-transfer model that is parametrized by a rooted species tree T , a *duplication rate* λ , a *loss rate* μ , and a *transfer rate* τ . The transfer rate specifies the rate at which a gene lineage jumps from its current species branch to a contemporaneous branch of another species. The recipient species is typically chosen uniformly at random from those species existing at the same time. The duplication and loss rates are defined as in the basic duplication-loss model by Arvestad et al. [111]. Each gene lineage evolves inside the species tree according to this stochastic birth-death-transfer process, producing a rooted gene tree that may have a different topology than the species tree.

Probabilistic models of reticulate evolution. Processes such as *hybridization*, *introgression*, and *recombination*, that describe the reproduction or exchange of genomic material between individuals of different species, give rise to evolutionary histories that are best modeled by a *phylogenetic network* rather than a tree. These reticulate evolutionary processes have been documented for diverse groups of species, including plants [119], algae [120], and fungi [121]. Modeling such histories requires probabilistic frameworks that can jointly account for both vertical inheritance and horizontal genetic exchange.

When inheritance is not fully vertical, evolutionary relationships among taxa can be modeled by a network. A *phylogenetic network* $N = (V, E)$ is a leaf-labeled rooted directed acyclic graph with vertex set V , edge set E , and leafset $\mathcal{L}(N)$. Internal vertices of N are of two types. *Tree nodes* represent speciation events and have in-degree 1 and out-degree ≥ 2 and their incident edges are called *tree edges*. *Reticulation nodes* represent reticulate events such as *hybridization*, *introgression*, and *recombination* and have in-degree ≥ 2 and out-degree 1 and their incident edges are called *reticulation edges*. When N is *metric*, every tree edge $(u, v) \in E$ is assigned a branch length λ_{uv} , and every reticulation edge (u, v) is assigned an *inheritance probability* or *hybridization parameter* γ_{uv} , representing the proportion of genetic material inherited along that edge. A phylogenetic tree is a special case of a phylogenetic network in which every internal vertex has in-degree 1. N has a single node

$r \in V$ with in-degree 0, that is referred to as the *root* node.

The *unrooted* version of N is obtained by collapsing the root and undirecting all edges. The *semidirected* version of N , sometimes denoted N^- , is obtained by collapsing the root and undirecting all tree edges while retaining the direction of reticulate edges. While the rooted version of N cannot have directed cycles and is therefore acyclic, the unrooted and semidirected versions can contain cycles. A *blob* of N is a maximal subgraph that cannot be disconnected by removing a single node, and can contain multiple cycles. The *level* of the phylogenetic network N is defined as the maximum number of reticulation nodes in any blob. When all cycles are node-disjoint, each blob corresponds to a single cycle, and N is referred to as a level-1 network. A level-0 network corresponds to a simple tree.

The **Network Multi-Species Coalescent (NMSC)** extends the **MSC** model to phylogenetic networks by accounting for both **ILS** and reticulation events such as hybridization. The model was introduced by Meng and Kubatko [122] and subsequently developed by others [123, 124, 125, 126]. The model parameters of **NMSC** are specified by a metric, binary, rooted phylogenetic network $(N, (\lambda, \gamma))$, where λ denotes the set of branch lengths in **coalescent units** and γ denotes inheritance probabilities on reticulation edges.

The **NMSC** determines the probability of observing any rooted or unrooted gene-tree topology given the species network; these probabilities can be computed exactly from the network parameters, similar to results established for trees by Allman et al. [81]. Of primary interest are the probabilities that a gene tree displays each of the three possible unrooted quartet topologies on four taxa A, B, C, D , referred to as the *quartet concordance factors*:

$$CF_{ABCD} = (CF_{AB|CD}, CF_{AC|BD}, CF_{AD|BC}), \quad (2.7)$$

where each component denotes the probability of the corresponding unrooted quartet (and the three probabilities sum to 1).

Banos [127] and Solís-Lemus and Ané [128] have shown that, under the **NMSC**, the semi-directed topology of a rooted level-1 network can be recovered from the set of all unrooted quartets (equivalently, from quartet concordance factors) provided the network has no cycles of size less than four; moreover, for cycles of size at least five, the locations of hybrid nodes and the directions of reticulation edges are also identifiable (for 4-cycles the topology is identifiable but the hybrid edge orientations are not). Allman et al. [129] further prove that the full semi-directed level-1 network (including topology, branch lengths, and inheritance probabilities) is identifiable from quartet concordance factors assuming cycles are of size at least five.

2.4.2 Models of sequence evolution

Models of sequence evolution describe how genomic sequences (e.g., DNA or proteins) evolve over time along the branches of a phylogenetic tree. Mutations, such as insertions, deletions and substitutions, can change the gene sequence over time. In this section, we review models that describe the evolution of DNA sequences.

DNA evolution can be described as a continuous-time Markov process with state space $S_{\text{DNA}} = \{A, C, T, G\}$, a rooted phylogenetic tree T , a probability distribution π of states at the root of T where $\forall x \in S : \pi_x > 0$, and a $|S| \times |S|$ rate matrix Q where Q_{ij} is the rate of substitution from character state i to character state j . The diagonal entries of rate matrix Q are typically set so that each row sums to zero, i.e.,

$$Q_{ii} = - \sum_{j \neq i} Q_{ij} \quad (2.8)$$

The transition probability matrix P gives the probabilities of changes for each edge of T as a function of branch lengths, where $P_{ij}(t)$ is the probability of transition from character state $i \in S$ to character state $j \in S$ for branch length $t \in \mathbb{R}_{\geq 0}$. Note that P is related to matrix Q by matrix exponentiation, so that

$$P(t) = e^{Qt} = \sum_{n=0}^{\infty} Q^n \frac{t^n}{n!} \quad (2.9)$$

Markov models of DNA substitution are typically *stationary*, meaning that the transition rate matrices do not depend on time, and *non-reversible*, meaning that the probability distribution of character states at the leaves of T do not depend on the location of the root. For a reversible stationary model of DNA substitution, it follows that

$$\forall i, j : \pi_i Q_{ij} = \pi_j Q_{ji} \quad (2.10)$$

A continuous-time Markov model on a tree T is *homogeneous* if

$$\forall e \in E(T) : Q^e = Q \quad (2.11)$$

and non-homogeneous if each branch has its own rate matrix Q^e .

Markov models of sequence evolution describe a generative process where first, an element $X_r \in S$ is chosen at root r of tree T according to the probability distribution π , so that $\mathbb{P}(x = S_i) = \pi_i$ where S_i is the i th element of state space S . Then, the site evolves down the

tree according to the Markov process so that for each node v with parent u , the site at v is generated according to the probability

$$\mathbb{P}(X_v = j | X_u = i) = [\mathbb{P}(t_{uv})]_{ij} = [e^{Qt_{uv}}]_{ij} \quad (2.12)$$

where X_v and X_u are the sites at nodes v and u respectively, and t_{uv} is the length of branch from u to v in T . This process is repeated for all branches of the tree, traversing from root to the leaves, until states are generated for all internal and terminal nodes in T . Once the process ends, the states at the leaves of T form a *site pattern*. The process is repeated independently for each site in the sequence alignment, until sequences with the desired length form at the leaves of the tree.

Similar Markov models can be used to describe the evolution of amino acid or codon sequences, but the state space and transition matrices differ; for example, amino acid models have a state space of size 20, while codon models use a state space of size 61–64 (including or excluding stop codons).

The simplest stationary, reversible and homogeneous Markov model of DNA evolution is the [Jukes-Cantor \(JC69\)](#) model [130], which assumes equal base frequencies and equal substitution rates between all pairs of nucleotides, so that

$$\forall x \in S_{DNA} : \pi_x = \frac{1}{4} \quad , \quad Q_{ij} = \begin{cases} -\alpha & \text{if } i = j \\ \frac{\alpha}{3} & \text{if } i \neq j \end{cases} \quad , \quad P_{ij}(t) = \begin{cases} \frac{1}{4} + \frac{3}{4}e^{-4\alpha t/3} & \text{if } i = j \\ \frac{1}{4} - \frac{1}{4}e^{-4\alpha t/3} & \text{if } i \neq j \end{cases} \quad (2.13)$$

More complex models, such as Kimura 2-parameter (K2P) [131] and HKY85 [132] further generalize this by distinguishing between some forms of substitutions, or allowing for unequal base frequencies. The most general stationary time-reversible Markov model of DNA evolution commonly used in phylogenetics is the [Generalized Time Reversible \(GTR\)](#) model [133], that allows for different rates for all six types of nucleotide substitutions and arbitrary equilibrium base frequencies. Therefore, the constraints of the [GTR](#) Markov model is

$$\sum_x \pi_x = 1, \pi_x > 0 \quad , \quad Q_{ij} = r_{ij}\pi_j \text{ for } i \neq j \quad , \quad Q_{ii} = -\sum_{j \neq i} Q_{ij} \quad , \quad \pi Q_{ij} = \pi_j Q_{ji} \quad (2.14)$$

These *time-reversible*, *stationary*, and *homogeneous* Markov models form a hierarchy of increasing flexibility, and model selection is often performed to choose the best-fitting model for a given dataset [82]. These models are special cases of the [General Markov \(GM\)](#) model [134] that does not assume stationarity and homogeneity and is non-reversible. The unrooted

GM model tree topology is identifiable from the distribution of states at the leaves of the tree, and the same is true for all of its submodels [135].

Rate heterogeneity across sites. In standard Markov models of sequence evolution, all sites are assumed to evolve at the same rate, and therefore have the same transition probability matrix. However, in practice, different locations within a sequence can evolve at different rates; some sites are highly conserved while others mutate more rapidly. Markov models can be extended to account for this *rate heterogeneity across sites*, by scaling the branch lengths of the model tree T for each site. To model this variation, each site i is assigned a *relative rate multiplier* r_i drawn from a probability distribution, that is most commonly a Gamma distribution with shape parameter α_i . Then the length of each branch in tree T is multiplied by r_i and site i is evolved down the re-scaled tree. This process is repeated for each site in the sequence. For example, when the sequence evolution model is **GTR** and a Gamma distribution is used to model rate heterogeneity, we get the popular GTR+GAMMA [136, 137] model. In practice, implementations often discretize the Gamma distribution into different rate categories, assigning each site to one of the categories; this model is referred to as GTR+CAT [136]. Accounting for rate heterogeneity across sites generally improves model fit and accuracy in phylogenetic inference and likelihood calculations.

No Common Mechanism model. While rate-across-sites models can account for rate heterogeneity, they only allow for sites to be scaled versions of each other. However, sometimes patterns of site variation are not consistent with a rate-across-sites model, so that the relative rates of substitution among sites change over time or differ across sites in ways that cannot be explained by a simple scalar rate factor. This is referred to as *heterotachy* [138], that can be modeled with the **No Common Mechanism (NCM)** [139] model, where each site and each edge has its own independent transition probability matrix, and only the tree topology is shared. In the model proposed by Tuffley and Steel [139], the distribution of states at the root and the probability transition matrix for each edge and site are defined as

$$\forall x \in S_{DNA} : \pi_x = \frac{1}{4} \quad , \quad P_{ij}(e, s) = \begin{cases} 1 - p(e, s) & \text{if } i = j \\ p(e, s)/3 & \text{if } i \neq j \end{cases} \quad (2.15)$$

where $0 \leq p(e, s) \leq 3/4$ is the probability that a substitution occurs on edge e for site s .

Molecular clock. Clock models are useful in molecular phylogenetics for relating genetic divergence to time. The *molecular clock* hypothesis states that genetic mutations accumulate

at a roughly constant rate over time, and therefore the rate of molecular sequence change can be used to estimate the timing of evolutionary events [140]. When the rate of nucleotide or amino-acid substitution is constant over time, the evolutionary distance between the root and all leaves of a phylogenetic tree is the same; this assumption is referred to as the *strict molecular clock*. However, in empirical data, evolutionary rates typically vary across lineages. *relaxed molecular clock* models accommodate this by allowing substitution rates to vary across branches of a phylogenetic tree. These include *uncorrelated* models, that assign each branch an independent rate drawn from a specified distribution (such as lognormal or exponential), and *correlated* models, in which rates on adjacent branches are statistically dependent [141].

Long branch attraction. *Long branch attraction (LBA)* is a phenomenon in phylogenetics where long branches (e.g., rapidly evolving lineages) are incorrectly inferred to be closely related, even if they are not. A simple example that demonstrates this phenomenon is the *Felsenstein Zone* tree [142]. Assume a rooted quartet with topology $((A, B), (C, D))$ where the edge separating A, B from C, D has a short length s , the edges adjacent to taxa A and C also have length s , and the edges adjacent to taxa B and D have a long length l such that $s \ll l$. The two long branches adjacent to B and D may independently accumulate the same character state by chance. *LBA* occurs when phylogenetic estimation methods mistakenly group the taxa B and D as siblings in the inferred tree, although they are not adjacent in the true tree. Methods such as *Maximum parsimony (MP)*, that seeks the tree that requires the minimum number of substitutions that explain the observed input character data, are particularly prone to *LBA*. *MP* is positively misleading in the Felsenstein Zone, meaning that it converges to the wrong tree by grouping the two long branches together with probability going to 1 as the branch length l increases. *ML* and distance-based methods are less sensitive to *LBA*, but they can recover the wrong tree when *LBA* occurs when the substitution model is mis-specified or the number of sites is bounded [29]. Increasing *taxon sampling* by adding more taxa to the tree to break long branches can prevent *LBA* [143].

2.5 GENE TREE ESTIMATION

Gene tree estimation is a critical step in phylogenomic studies, especially in multi-locus analysis when a tree is independently inferred for each gene based on its aligned sequences. Gene trees are estimated from different types of genetic markers, including *exons* that are coding regions of the genome, *introns* that are non-coding regions, and *ultraconserved elements (UCEs)* that are highly conserved non-coding regions. Different approaches can be

used to estimate gene trees, including maximum likelihood-based methods such as RAxML [25], IQ-TREE [92] or FastTree-2 [144], Bayesian methods such as MrBayes [145], or distance-based methods such as [Neighbor Joining \(NJ\)](#) [146]. Estimated gene trees typically have some estimation error, and this error can be high when the sequence alignments are short or when phylogenetic signal is low, for example, due to genes evolving slowly and not accumulating enough informative substitutions. In simulations, the level of gene tree estimation error is typically measured as the average normalized RF distance between the true and estimated gene trees, and is denoted by [GTEE](#). We now provide a brief overview of the main approaches for gene tree estimation.

2.5.1 Maximum likelihood methods

[Maximum likelihood \(ML\)](#)-based phylogeny estimation methods are widely used approaches for reconstructing phylogenetic trees from sequence data. These methods attempt to find the tree topology and numeric parameters that maximize the likelihood of observing the input sequence alignment under a specific model of sequence evolution, such as Jukes-Cantor or GTR. Finding the maximum likelihood tree is an NP-hard optimization problem [147], as the space of possible trees grows super-exponentially with the number of taxa. Therefore, ML methods rely on *heuristic* search strategies, such as hill climbing, nearest-neighbor interchanges, or subtree pruning and regrafting to search tree space [148]. While maximum likelihood methods are computationally intensive, they are known for their statistical rigor and often produce highly accurate trees. Example software tools that implement ML phylogeny inference include RAxML [149], IQ-TREE [26, 92], FastTree [144] and PhyML [150] that vary in scalability and features.

2.5.2 Distance-based methods

Distance-based methods are a class of phylogenetic tree reconstruction techniques that construct trees using a matrix of pairwise distances between sequences. These methods first compute an evolutionary distance between pairs of sequences, often using models of sequence evolution that correct for multiple substitutions at a site, and then apply algorithms like [Neighbor Joining \(NJ\)](#) or FastME on the distance matrix to infer a tree. These methods are typically more scalable than maximum likelihood-based methods and many of them are polynomial time, but they can be less accurate when evolutionary rates vary substantially across branches, or when sequence divergence is high.

The input to distance-based methods is typically a set of sequences $S = \{s_1, s_2, \dots, s_n\}$,

each of the same length k . Evolutionary distances between the pair (i, j) can be calculated by computing the fraction of sites that is different between the sequences s_i and s_j (referred to as the *Hamming distance*), and then corrected under a model of sequence evolution, such as [JC69](#), to account for multiple substitutions at a site. This is referred to as *Jukes-Cantor distance correction*. Note that the JC-corrected distances form a *dissimilarity* matrix D , i.e., a matrix that is symmetric and zero on the diagonal, but may not satisfy the *triangle inequality*. The JC-corrected distances converge to the true evolutionary distance for each pair of sequences as $k \rightarrow \infty$. Two important concepts in establishing theoretical guarantees for distance-based methods are *additivity* and *safety radius*, which we now define.

Definition 2.7 (Additive and Nearly Additive). Let (T, λ) be a weighted phylogenetic tree on label set $\mathcal{L}(T) = \{s_1, s_2, \dots, s_n\}$ with non-negative branch lengths. A distance matrix D is *additive* for T if the element $D[i, j]$ is equal to the patristic distance between the leaves labeled s_i and s_j in T , i.e., the sum of edge weights on the unique path between them. D is *nearly additive* for T if there exists an additive matrix d_T for T such that $|D[i, j] - d_T[i, j]| < \rho r$ for all pairs i, j where $\rho > 0$ is a constant and r is the shortest internal edge length in T .

A matrix D is additive if and only if it satisfies the *Four-Point Condition* [151], that is, for every four distinct taxa a, b, c, d , the two largest sums among $D[a, b] + D[c, d]$, $D[a, c] + D[b, d]$, and $D[a, d] + D[b, c]$ are equal. Note that when k is finite, the JC-corrected distance matrix may not be *additive*.

Definition 2.8 (Safety Radius). The *safety radius* of a distance-based phylogenetic tree estimation method Θ is the largest value ρ such that, whenever a dissimilarity matrix D is nearly additive for the edge-weighted tree (T, λ) with $|D[i, j] - d_T[i, j]| < \rho \cdot r$ for all pairs of taxa i, j , where d_T is the additive distance matrix induced by T and r is the length of the shortest internal edge in T , the method Θ is guaranteed to return an additive matrix corresponding to a weighted version of tree topology T .

As established by Atteson [152] and Erdős et al. [153], the optimal (largest possible) safety radius for any distance-based method is $1/2$, and the [Neighbor Joining \(NJ\)](#) algorithm and the Buneman Tree [154] method have the optimal safety radii.

2.5.3 Bayesian methods

Bayesian methods provide a probabilistic framework for inferring phylogenetic trees given molecular sequence data and a model of sequence evolution. Unlike [ML](#)-based methods

that find a single tree that maximizes the likelihood of the observed sequence data, Bayesian methods typically use [Markov Chain Monte Carlo \(MCMC\)](#) sampling to sample the posterior distribution of trees. This set can then be used to test different evolutionary hypotheses or be summarized into a single tree with posterior probability of each clade represented as support values [155]. Bayesian methods can be very accurate and are particularly useful for incorporating complex models of sequence evolution and rate variation across sites [156]. However, they are computationally intensive, and the runtime can become prohibitive (often taking days to weeks) for datasets with more than a few hundred taxa [157].

2.5.4 Branch support calculation

While the output of phylogenetic estimation methods is typically a single tree, downstream biological analyses often need a measure of reliability for each branch of the tree. The *statistical support* of an internal branch (corresponding to a bipartition or split) quantifies the confidence or reliability of that branch based on the input sequence data. One of the most widely used techniques for calculating branch support is [non-parametric bootstrapping](#) [66, 158], where multiple *bootstrap replicate* datasets are created by sampling sites (i.e., alignment columns) with replacement from the input [MSA](#), and a separate phylogenetic tree is estimated for each bootstrap replicate. The bootstrap support of a branch e that induces the bipartition $A|B$ in the original estimated tree T is defined as the proportion of bootstrap trees in which the same bipartition $A|B$ appears. While the interpretation of support values is not always straightforward, values above 95% are generally considered reliable and values below 50% often indicate weak and unreliable relationships [159].

2.6 SPECIES TREE ESTIMATION

Species tree estimation is a fundamental task in phylogenomics, and aims to reconstruct the evolutionary relationships among species while accounting for biological processes that cause gene tree heterogeneity. Unlike gene tree estimation, which is performed independently for each locus, species tree estimation integrates information across multiple loci to infer a single tree that reflects the overall evolutionary history of species. Accurate species tree estimation is often challenging due to gene tree estimation error, high levels of gene tree discordance, or model violations, and it requires careful consideration of data quality, model assumptions, and computational scalability [160].

2.6.1 Theoretical and statistical properties of methods.

In addition to evaluating empirical performance, we study species tree estimation methods based on their statistical and theoretical properties, including parameter identifiability, statistical consistency, and sample complexity. A model parameter θ is called *identifiable*, when it can be uniquely determined from the distribution of observed data. An estimation method M is an *statistically consistent* estimator of parameter θ under a model when as the amount of data generated under the model goes to infinity, the output of M converges in probability to the true value of the parameter θ . When we find a condition for which this doesn't hold, we say that M is *statistically inconsistent*. If the output of M converges to the wrong parameter $\theta' \neq \theta$ with probability going to 1 as the amount of data increases, we say that M is *positively misleading*. Note that M can be statistically inconsistent, but not positively misleading. *sample complexity* refers to the amount of data required for an statistical estimation method M to return parameter θ with probability at least $1 - \epsilon$ for a desired constant $\epsilon > 0$. Note that identifiability is a property of a statistical *model*, and statistical consistency and sample complexity are properties of an estimation *method*.

We now bring examples for these concepts in the context of species tree estimation. The unrooted topology of an **MSC** model species tree (T, λ) is identifiable from the distribution of unrooted gene trees under the **MSC** when no two different model trees on $\mathcal{L}(T)$ produce the same unrooted gene tree distribution. A species tree estimation method is an statistically consistent estimator of the unrooted species tree topology under the **MSC** if as the number of gene trees (or the sequence length) goes to infinity, the probability that the method returns the correct unrooted topology converges to 1. Sample complexity analysis of a species tree estimation method determines the number of genes or the amount of sequence length required for the method to correctly recover the species tree topology with probability at least $1 - \epsilon$ for a desired $\epsilon > 0$.

2.6.2 Species tree estimation in the presence of ILS.

Concatenation using maximum-likelihood. Concatenation analysis using maximum likelihood (**CA-ML**) is a common technique for species tree estimation where multiple sequence alignments from different loci are combined into a single supermatrix and a tree is estimated from this super-alignment using **ML**-based phylogeny inference methods such as RAxML or IQ-TREE under a model of sequence evolution. This approach assumes that all genes evolve along the same underlying tree *topology*, an assumption that can be violated when different loci have different evolutionary histories. However, *partitioned* analyses

allow different genes to have their own substitution models or rate parameters, while *unpartitioned* analyses assumes a single substitution model for all loci. Partitioning generally improves model fit and can lead to more accurate trees, but increases model complexity and therefore comes at a computational cost. Roch and Steel [28] have proved that unpartitioned CA-ML can be statistically inconsistent and even positively misleading under the MSC model. Performance in practice has been mixed, and while CA-ML typically has good accuracy when ILS is low, it can have poor accuracy in the presence of moderate or high levels of ILS [32].

Coalescent-based methods. Several statistical methods have been developed for species tree estimation that account for gene tree discordance due to ILS, and many of these methods have been proven to be statistically consistent under the MSC [37, 38, 42, 43, 46, 47, 49, 63, 93, 93, 161, 162, 163, 164, 165], meaning that their output is guaranteed to converge to the true unrooted species tree topology as the amount of error-free input data increases (see [160] for an entry into this literature). These methods either produce a single point estimate of the species tree or infer a posterior distribution over the space of possible trees, from which a point estimate can be derived subsequently. We now describe different classes of coalescent-based species tree estimation methods.

Summary methods. The input to summary methods is a set of rooted or unrooted gene trees estimated on different loci and their output is an inferred species tree. Some summary methods require gene trees with numeric parameters (e.g., branch lengths) as input and output a species tree with branch lengths, but many of them only require tree topologies as input and output a tree without numeric parameters. Many of these methods, including STEM [164], STAR [93], MP-EST [166], and GLASS [167] use rooted gene trees as input and have been proven to be statistically consistent under the MSC. In practice, however, accurate rooting of gene trees can be difficult, and commonly used approaches such as outgroup rooting can be unreliable in the presence of gene tree discordance [168] (see Sec. 2.7 for further discussion on rooting methods). Therefore, summary methods that combine unrooted gene trees are often preferred in empirical analysis.

A group of summary methods that use unrooted gene trees are based on the key fact that for four species, the most probable unrooted gene tree under the MSC has the same unrooted topology as the model species tree [81]. Therefore, on a quartet of taxa, the unrooted species tree topology and its internal branch lengths (in coalescent units) are identifiable from the distribution of unrooted gene tree topologies. This result does not extend beyond quartets, and for five or more species, there exist conditions where unrooted gene

trees that do not match the model species tree, i.e. *anomalous* gene trees, have a higher probability of appearing than the unrooted species tree [169, 170]. This result has motivated the development of several statistically consistent quartet-based methods for species tree estimation (e.g., [48, 171]), the most well-known of which are the ASTRAL family of methods [48, 50, 51, 57, 172] that are now widely used in phylogenomic studies.

The ASTRAL family of methods solve variants of the [Maximum Quartet Support Supertree \(MQSS\)](#) problem, which we now define.

Definition 2.9 (Maximum Quartet Support Supertree). Let $\mathcal{T} = \{T_1, T_2, \dots, T_k\}$ be a collection of phylogenetic trees. A tree T^* on label set $\mathcal{L} = \bigcup_{i=1}^k \mathcal{L}_i$ is called a *maximum quartet support supertree* for set \mathcal{T} if it is a solution to

$$T^* = \arg \max_T \sum_{t \in \mathcal{T}} |Q(T) \cap Q(t)|, \quad (2.16)$$

where $Q(T)$ denotes the set of resolved quartets induced by T .

The [MQSS](#) problem is NP-hard [173, 174] and is equivalent to the *Quartet Median Tree* problem [173, 175]. However, several heuristics have been developed for it, including *Quartet Puzzling* [176] and *Quartet MaxCut* [177]. A special case known as the *bipartition-constrained* variant of [MQSS](#), in which the search space for T^* is restricted to trees whose set of bipartitions $\text{Bip}(T)$ is a subset of a given input set of bipartitions Φ on \mathcal{L} , can be solved in polynomial time [178]. The ASTRAL algorithm efficiently solves this constrained version using [dynamic programming \(DP\)](#) [179].

There are also distance-based summary methods that create a dissimilarity matrix of average internode distances from the input set of unrooted gene trees, and then use a distance-based tree estimation method to infer a species tree from this distance matrix. These methods mainly differ in the choice of distance-based method they use, and include NJst [180] that uses [Neighbor Joining \(NJ\)](#) for tree estimation and ASTRID [181] that uses FastME (or BioNJ* [182] for inputs with missing data). ASTRID is faster and typically more accurate than NJst [181] and both methods are statistically consistent under the [MSC](#), since [NJ](#) and FastME both have positive safety radii and as the number of gene trees increases, the matrix of average internode distances will converge to an additive matrix for the model species tree. Weighted ASTRID [183] is a new variant of ASTRID that takes gene tree branch uncertainty into account and is faster and typically more accurate than ASTRID.

Site-based methods. Site-based methods estimate species trees directly from sequence data, without first estimating gene trees, by modeling the distribution of site patterns un-

der the [MSC](#). Examples of these methods include SNAPP [184], SVDQuartets [185], and METAL [186]. Among these methods, SVDQuartets is the most well-known, and works by analyzing the distribution of site patterns by evaluating all quartets of taxa. The input to SVDQuartets is a set of sites sampled randomly from across the genome, each from an unlinked loci. For each quartet of taxa, SVDQuartets constructs a matrix of site pattern frequencies and applies [singular value decomposition \(SVD\)](#) to test which of the three possible unrooted quartet topologies best satisfies rank conditions predicted by the [MSC](#). Once topologies for individual quartets are determined, a quartet amalgamation method is used to construct the species tree on the full set of taxa. In the implementation of SVDQuartets in PAUP* [187], this amalgamation method is a modified version of Quartet Fiducia–Mattheyses (QFM) [188]. SVDQuartets is proven to be statistically consistent under the [MSC](#) [45].

One major advantage of site-based methods compared to other coalescent-based species tree estimation methods is that they do not directly or indirectly rely on estimating gene trees. Given that gene tree estimation error negatively impacts summary methods [32, 189, 190], site-based methods can be especially useful on datasets with short loci or when gene tree estimation error is high. Depending on the size of the alignments, they can also be more scalable than two-step summary methods when the number of taxa is small, as they skip gene tree estimation which is usually the most time-consuming step of a phylogenomics pipeline [191]. However, computing all quartets is expensive for large numbers of species, and these methods are best suited for datasets with at most 100 taxa [18].

Co-estimation methods. Bayesian co-estimation methods use a probabilistic framework to jointly infer gene trees and species trees given sequence data under the [MSC](#) model. Examples of these methods include BEST [192], *BEAST [193] and StarBeast2,3 [38, 194]. One of the most well-known co-estimation methods, *BEAST, uses [MCMC](#) sampling techniques to approximate the posterior distribution of gene trees and species trees conditioned on the observed sequence data. Species trees produced by *BEAST have shown to be more accurate than those produced by summary methods [31], and gene trees produced by *BEAST can also be more accurate than trees estimated using maximum likelihood on individual gene alignments. However, co-estimation methods typically scale poorly with respect to the number of loci and number of taxa. For example, *BEAST does not converge in reasonable time on datasets with more than 25 species and 100 loci [41, 195, 196].

2.6.3 Species tree estimation in the presence of GDL

Gene duplication and loss creates paralogous genes that violate the basic assumption of orthology typically required by many species tree estimation methods. Despite this, recent advances have enabled accurate species tree estimation in the presence of [GDL](#). For instance, methods such as DupTree [197], Notung [198], and SpeciesRax [199] infer species trees by reconciling gene trees under probabilistic or parsimony-based [GDL](#) models. DupTree uses a duplication-loss parsimony framework to efficiently search for the species tree minimizing the total number of duplications. Notung incorporates both duplication and loss events within a flexible reconciliation framework that allows for root inference and uncertainty analysis. Finally, SpeciesRax employs a maximum likelihood approach to estimate a rooted species tree and its numeric parameters from a set of gene family trees accounting for both [GDL](#) and [HGT](#).

Similar to [ILS](#), theoretical guarantees have also been established for species tree estimation in the presence of gene duplication and loss. In particular, the identifiability of the unrooted species tree topology from the distribution of MUL-trees has been proven under various models of [GDL](#) [89, 200]. Several quartet-based methods, such as ASTRAL-multi [53], have been shown to be statistically consistent under these models [89, 200, 201]. In addition, other methods with strong practical performance have been developed for species tree estimation under [GDL](#), such as ASTRAL-Pro [54, 56], FastMulRFS [88], and DISCO [90], which offer statistical guarantees under certain restricted conditions.

2.6.4 Species tree estimation in the presence of HGT

There has also been progress in developing methods for species tree estimation in the presence of [Horizontal gene transfer \(HGT\)](#). In particular, Roch and Snir [202] and Daskalakis and Roch [203] showed that, under bounded levels of random [HGT](#), the unrooted species tree topology is identifiable from the distribution of unrooted gene trees, and the most probable unrooted quartet gene tree is identical to the quartet model species tree. This theoretical result enabled the proof of statistical consistency for ASTRAL under bounded [HGT](#) [204], where it also demonstrated strong empirical performance compared to alternative methods.

2.6.5 Species tree estimation in the presence of reticulation

There has also been significant progress in estimating phylogenetic networks (that model reticulate evolutionary scenarios) under the [Network Multi-Species Coalescent \(NMSC\)](#)

model that extends the [MSC](#) to model reticulation events, such as hybridization and introgression, in addition to [ILS](#) [20, 123]. Several methods have been developed for species network inference under the [NMSC](#) framework [128, 205, 206, 207, 208, 209, 210, 211], many of which are implemented inside the PhyloNet [212] and PhyloNetworks [210] software packages. In addition, theoretical results have been established for identifying the topology and some parameters of level-1 phylogenetic networks (i.e., networks whose cycles are node-disjoint) from the distribution of unrooted quartet gene trees [127, 129, 213]. However, methods for estimating phylogenetic networks are typically substantially less scalable than methods for species tree inference, and their application is limited to small datasets or simple networks scenarios with few reticulations.

These findings indicate that accurate species tree inference is achievable even in the presence of multiple sources of gene tree discordance. However, most existing methods for estimating species trees and networks produce unrooted topologies without associated branch lengths or divergence times. Aside from a few methods that use gene duplication events to root species trees [214, 215] and one approach that uses horizontal gene transfer events to infer divergence times [216], there are no post-species tree analysis methods that explicitly account for sources of gene tree discordance. However, prior work has demonstrated that, much like species tree estimation, post-species tree analyses can be affected by incomplete lineage sorting and other forms of gene tree heterogeneity [69, 70, 217]. This highlights the need for new approaches that address gene tree discordance in post-species tree inference, which is the main focus of this dissertation.

2.7 POST-SPECIES TREE ANALYSIS

Rooting, branch length estimation and dating are important steps in a phylogenomics pipeline that add temporal context to the reconstructed species tree topology. *Rooting* specifies the direction of evolution by identifying an ancestral node, that determines the ancestor-descendant relationships across the tree. The root can be inferred using different techniques, including the use of [outgroup](#) taxa or algorithmic methods based on molecular clock analysis [148]. *Branch length estimation* assigns lengths to the branches of the tree, that reflect the amount of evolutionary change and are typically in the unit of the expected number of substitutions (mutations) per site or coalescent units. *Dating* extends this by translating branch lengths into absolute or relative divergence times, using models that relate genetic distance to time [218]. Dating typically requires *calibration* information, such as fossil calibrations or sampling times for fast-evolving organisms (e.g., viruses) [219]. Together these steps transform an unrooted tree topology into a biologically interpretable evolutionary

history.

2.7.1 Rooting

Methods for estimating phylogenetic trees often rely on time-reversible models of DNA substitution that do not preserve the direction of time. Therefore, both gene trees and species trees inferred under these models are typically unrooted. Rooting determines the direction of evolution by finding the most recent common ancestor of all taxa in a tree. The most widely used approach for rooting phylogenetic trees is outgroup rooting, in which one or more species known to be evolutionarily distant from the ingroup are included in the analysis. Next, a tree is estimated on the combined set of species, and the root is placed on the edge that separates the ingroup from the outgroup. Another class of methods, including Midpoint rooting [220] and Minimum Ancestor Deviation (MAD) [221], use different optimization criteria based on molecular clock assumptions to find the position of the root. Finally, some methods use signal from discordance between gene trees and species trees to infer the root of the species tree, such as STRIDE [222] that uses gene duplication events. Rooted trees can be compared using the rooted version of the [RF](#) distance, often referred to as the clade distance. When the tree topology is fixed, this is proportional to the distance between the inferred and true root positions.

2.7.2 Branch length estimation

Branch length estimation provides quantitative measures of evolutionary change along the edges of a phylogenetic tree. Branch lengths of a phylogenetic tree can be represented in different units. For a tree T and a branch $i \in E(T)$, the length of i can be represented in the following units:

- *Generation units* g_i that correspond to the number of generations between two divergence events along branch i .
- If the average generation time is τ , then the branch length in absolute *time units* (e.g., years) is $t_i = g_i\tau$.
- Under the [MSC](#), time is often re-scaled to the effective population size N_e . For diploid organisms, the expected coalescent time is $2N_e$ generations. Therefore, the branch length in [coalescent units \(CU\)](#) in this case is $T_i = \frac{g_i}{2N_e}$.

- In molecular phylogenetics, branch lengths are often expressed in terms of the expected number of substitutions per sequence site. If the per-generation mutation rate is ν_i (substitutions per site per generation), then the expected substitutions along branch i is $s_i = g_i \nu_i$. Combining with the coalescent unit expression, we get the branch length in substitution units $s_i = T_i(2N_e \nu_i) = T_i \mu_i$ where $\mu_i = 2N_e \nu_i$ is the substitution rate per sequence site per CU.

On empirical datasets, branch lengths typically represent the expected number of nucleotide or amino-acid substitutions per site, and are estimated from sequence data alongside the tree topology under models of sequence evolution such as GTR using maximum likelihood or Bayesian frameworks. When estimating branch lengths on species trees, many studies use concatenation with maximum likelihood to jointly estimate tree topology and branch lengths, or to optimize branch lengths on a fixed species tree topology. However, concatenation does not account for gene tree discordance due to processes such as ILS. On the other hand, Bayesian methods, such as BPP [223] and StarBEAST [38], use sequence data from multiple loci to infer a single set of branch lengths that reflects average genetic divergence between species across a set of potentially discordant loci.

2.7.3 Dating and diversification analysis

Molecular dating is the estimation of divergence times at internal nodes of a phylogenetic tree from molecular sequence data. Dating typically involves converting a phylogenetic tree with branch lengths in mutation units to an ultrametric *time-calibrated* tree, where branch lengths are measured in absolute or relative time. Common approaches for dating phylogenetic trees include Bayesian methods and maximum likelihood-based methods. Bayesian methods, such as those implemented in BEAST [224], MCMCTree [225, 226] and MrBayes [145], can infer divergence times jointly with tree topology and substitution rates given sequence data and fossil calibrations as priors. Maximum likelihood-based methods, such as Least-square dating (LSD) [227] and TreePL [228], typically convert branch lengths in substitution units into estimates of divergence time, using clock models that relate evolutionary rates to time and a set of calibration priors. These calibration priors provide information on the time of certain nodes in the phylogeny, often based on evidence from fossils or geological events. The most common type of these priors include *point calibrations*, that specify the exact timing of some nodes, *hard bounds* that limit the minimum and maximum age of a node, and *soft bounds* (also referred to as *calibration densities*), that model uncertainty in the node age using probability distributions such as lognormal or exponential.

Diversification analysis is the study of changes in rates and patterns of speciation and extinction across a dated phylogenetic tree. The *speciation rate* μ_S denotes the rate at which new species arise, and the *extinction rate* μ_E is the rate at which species go extinct. The *net diversification rate* is defined as $r_{net} = \mu_S - \mu_E$ and captures the overall pace of lineage accumulation in a species tree. Diversification analysis can be performed by constructing *lineage-through-time (LTT)* plots that visualize the number of lineages $N(t)$ as a function of time t , and is usually represented in log scale. A straight line in a log-scaled LTT plot under a pure birth model indicates a constant diversification rate, while deviations from linearity can suggest events such as mass extinctions or rate shifts [229].

CHAPTER 3: ROOTING SPECIES TREES UNDER THE MULTI-SPECIES COALESCENT MODEL

This chapter contains material previously published in “Y. Tabatabae, K. Sarkar, and T. Warnow (2022). Quintet Rooting: rooting species trees under the multi-species coalescent model. Bioinformatics, Vol. 38, Supplement 1, pages i109-i117, special issue for Intelligent Systems for Molecular Biology (ISMB) 2022”[230]. The Quintet Rooting software is available in open-source form at <https://github.com/ytabatabae/Quintet-Rooting>. The datasets and scripts used in this study are available in open-source form at <https://github.com/ytabatabae/QR-paper>.

3.1 INTRODUCTION

Phylogenetic trees provide insight into many biological questions and are typically estimated using statistical methods that assume a model of evolution. While **rooted** phylogenies are the final objective, estimated gene trees (i.e., trees on a single **locus**) and estimated species trees (i.e., trees considering multiple loci, potentially full genomes) are usually unrooted: gene trees are generally estimated under time reversible models of sequence evolution, and species trees are then estimated under models for gene evolution within species trees, using techniques such as ASTRAL [48], StarBEAST2 [38], or SVDquartets [42] that also do not produce **rooted** trees. As a result, additional techniques for rooting species trees are used.

When evolutionary rates follow a **strict molecular clock** (so that the expected number of substitutions is proportional to time), then rooting trees is straightforward. However, since evolution does not follow the **strict molecular clock** (see discussion and references in [231]), “relaxed clock” models have been proposed and then used to root phylogenies [232]. Outgroup rooting (i.e., rooting the species tree on the edge that separates the **outgroup** species from the rest of the species) is another common approach. Both relaxed clock and **outgroup** rooting methods are used in practice, but no methods have been found to be entirely successful [233, 234]. Moreover, very few methods for rooting species trees have been developed that specifically address genome-scale processes, such as **Incomplete lineage sorting (ILS)** or **Gene duplication and loss (GDL)**, that result in discordance between gene trees and species trees [15, 105].

Here, we introduce a new statistical method, “Quintet Rooting”, for rooting species trees that can be used with multi-locus datasets and takes into consideration **gene tree** discordance due to **Incomplete lineage sorting**, as modeled by the **Multi-Species Coalescent (MSC)** [98] model. Quintet Rooting (QR) is based on the theoretical work by Allman, Degnan, and

Rhodes (henceforth, “ADR”) [81] that establishes that under the [MSC](#), each 5-leaf rooted species tree is [identifiable](#) from the distribution of the [unrooted](#) gene trees that it defines. [ADR](#) provides phylogenetic invariants and inequalities defined by the distribution on [unrooted](#) quintet trees that they prove hold under the [MSC](#) and that establish identifiability of any [rooted](#) 5-leaf species tree. However, [ADR](#) does not suggest an estimation method that uses these invariants and inequalities. This study shows that QR is able to use these invariants and inequalities to more accurately root species trees under the [MSC](#) than previous methods.

3.2 BACKGROUND

3.2.1 Previous methods for rooting trees

Rooting methods can generally be divided into three groups based on their assumptions about the input data. The technique most commonly used by biologists is [outgroup](#) rooting [235]. In this approach, one or more species (called “outgroups”) that are distantly related to all the [taxa](#) in the original input set (called the “ingroup”) are added to this set, and a species tree is then estimated on the resulting expanded set. When the [outgroup](#) species are separated by an edge from the ingroup species, then the tree can be [rooted](#) on that edge [236]. Although [outgroup](#) rooting is a natural and simple technique, it is not always reliable. If the [outgroup](#) species are too distant from the ingroup, they can function as “rogue taxa” and estimation methods may be unable to find the correct placement for them in the final tree (and with sufficiently distant relationships, they can be placed on any edge in the tree with equal probability); this makes the root specification impossible. Conversely, when they are too closely related to the ingroup, this can produce very short edges in the species tree that also makes it difficult to find the correct root location (as accurate resolution of trees with very short edges is known to be difficult [237]). Finally, sometimes a species that is supposed to be an [outgroup](#) is actually part of the ingroup. Moreover, previous work has shown that adding outgroups can even change the [topology](#) of the induced estimated tree on the ingroup species set, making the approach unreliable in some situations [238]. Therefore, choosing an appropriate [outgroup](#) needs prior biological knowledge as well as careful consideration of these scenarios, which may not be readily possible for some datasets and groups of species [239, 240].

Distance-based methods are another type of rooting method that can be used. These methods take an [unrooted phylogeny](#) with branch lengths as input and estimate the most likely location for the root based on specific assumptions about the model of evolution

[221, 241, 242, 243]. Most methods of this type (e.g., midpoint rooting) are either based on molecular clock analysis or non-reversible models of DNA substitution. When a strict molecular clock holds, midpoint rooting and other distance-based methods can be statistically consistent and estimated root locations can be highly accurate. However, as the evolutionary rate deviates from strict molecular clock, midpoint rooting can perform poorly. Minimum Ancestor Deviation (MAD) [79] and Minimum Variance Rooting (MinVar) [241] address this problem by minimizing a cost based on the deviation from the strict molecular clock. However, even MinVar and MAD can have poor accuracy as the deviation from the molecular clock and ILS level increases [241].

Another method for rooting a species tree was introduced in Tian and Kubatko (2017) [244]. This method assumes that genes evolve within a species tree under the MSC and that sequences evolve within each gene tree under the Jukes-Cantor (JC69) [130] substitution model and with the strict molecular clock. This method takes a set of unrooted gene trees and their sequence alignments as input and uses a series of hypothesis tests on site pattern probabilities to infer a rooted quartet tree for every four species; it then uses these rooted quartet trees to root a given larger species tree. While the method performs well when the strict clock holds, its accuracy degrades as the clock is violated [244].

RootDigger [243] is a new statistical method for rooting species trees. Given an unrooted species tree T and a sequence alignment, RootDigger computes the likelihood of each rooted version of T under a non-reversible model of evolution (specifically UNREST+ \mathcal{T} + I). In its default mode (i.e., "search"), RootDigger uses local search heuristics to find the most likely position for the root; for small enough trees, the "exhaustive" mode can be used, which scores every rooted version to quantify root placement uncertainty by computing the likelihood weight ratio for placing the root on each branch of the tree. Because RootDigger is very recent, less is understood about its performance compared to the other methods we have discussed here.

Finally, STRIDE [214] is relevant to rooting species trees when genes evolve with duplication and losses, so that the gene trees have multiple copies of the species. STRIDE uses properties about gene duplication and loss models to estimate the probability of the root being located on each edge in the species tree. However, by design, STRIDE cannot be used for rooting a species tree given single copy gene trees.

3.2.2 The Allman, Degnan, and Rhodes theory

Allman, Degnan, and Rhodes (ADR) [81] provided one of the fundamental theorems underlying species tree estimation under the MSC: they proved that for any four species,

Table 3.1: Examples of invariants, inequalities, and equivalence classes for *rooted* species trees of different categories according to [ADR](#).

| | $((((a, b), c), d), e)$ Caterpillar | $((a, b), c), (d, e))$ Balanced | $((a, b), (d, e)), c)$ Pseudo-Caterpillar |
|------------------------|-------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------|
| Invariants | $u_{14} - u_{15} = 0$ | $u_{14} - u_{15} = 0$ | $u_{14} - u_{15} = 0$ |
| | $u_{11} - u_{15} = 0$ | $u_{11} - u_{15} = 0$ | $u_{12} - u_{15} = 0$ |
| | $u_{10} - u_{15} = 0$ | $u_{10} - u_{15} = 0$ | $u_{10} - u_{15} = 0$ |
| | $u_8 - u_{15} = 0$ | $u_9 - u_{12} = 0$ | $u_9 - u_{15} = 0$ |
| | $u_7 - u_{15} = 0$ | $u_8 - u_{15} = 0$ | $u_8 - u_{11} = 0$ |
| | $u_6 - u_9 = 0$ | $u_7 - u_{15} = 0$ | $u_7 - u_{15} = 0$ |
| | $u_5 - u_{12} = 0$ | $u_6 - u_{12} = 0$ | $u_6 - u_{15} = 0$ |
| | $u_4 - u_{13} = 0$ | $u_5 - u_{12} = 0$ | $u_5 - u_{15} = 0$ |
| | $u_2 - u_3 + u_9 - u_{12} = 0$ | $u_4 - u_{13} = 0$ | $u_4 - u_{13} = 0$ |
| | | $u_2 - u_3 = 0$ | $u_2 - u_3 = 0$ |
| Inequalities | $u_1 > u_2, u_4 > u_5 > u_7$ $u_3 > u_2, u_6 > u_5 > u_7$ | $u_1 > u_2, u_4 > u_5 > u_7$ | $u_1 > u_2, u_4, u_8 > u_5$ |
| Equivalence Classes | $\{u_1\} > \{u_3\}$ $\{u_4, u_{13}\}, \{u_2\}, \{u_6, u_9\} > \{u_5, u_{12}\}$ $\{u_7, u_8, u_{10}, u_{11}, u_{14}, u_{15}\}$ | $\{u_1\} > \{u_2, u_3\}, \{u_4, u_{13}\} > \{u_5, u_6, u_9, u_{12}\} > \{u_7, u_8, u_{10}, u_{11}, u_{14}, u_{15}\}$ | $\{u_1\} > \{u_2, u_3\}, \{u_4, u_{13}\}, \{u_8, u_{11}\} > \{u_5, u_6, u_7, u_9, u_{10}, u_{12}, u_{14}, u_{15}\}$ |

the [unrooted topology](#) of the species tree is the same as the [topology](#) of the most probable [unrooted](#) gene tree. This theorem has been used to establish statistical consistency for quartet-based methods, such as ASTRAL, wQFM [171], and the population tree in BUCKY [162]. Interestingly, Theorem 9 from [ADR](#) has received much less attention and (to the best of our knowledge) is not used in any species tree estimation method. This theorem states that every 5-leaf *rooted* species tree [topology](#) is [identifiable](#) from the distribution of the [unrooted gene tree](#) topologies.

In deriving Theorem 9, [ADR](#) note that for any five species there are 105 possible [rooted binary](#) trees (i.e., there are 15 different [unrooted binary gene tree](#) topologies on five leaves, and each can be [rooted](#) on any of its seven edges). Using the [MSC](#), for any given [rooted](#) species tree on five leaves, [ADR](#) establish relationships (invariants and inequalities) between the probabilities for each of the 15 [unrooted gene tree](#) topologies, denoted by u_1, u_2, \dots, u_{15} . Every [rooted](#) 5-leaf species tree has a particular topological shape (i.e., caterpillar, balanced, or pseudo-caterpillar [245]), and [ADR](#) establish that the set of inequalities and invariants for a given [rooted binary](#) model species tree only depends on its shape and not on the numeric model parameters (i.e., branch lengths in coalescent units). Thus, the set of [ADR](#) invariants and inequalities fall into three categories, one for each shape. An example of these linear

invariants and inequalities for each tree shape category is provided in Table 3.1. Theorem 9 in ADR establishes that under the MSC, these invariants and inequalities suffice to identify the rooted species tree and its internal branch lengths, for all rooted model species trees with five leaves. In other words, if the probability distribution on unrooted gene tree topologies is known exactly, then there will be exactly one rooted species tree topology that satisfies all the invariants and inequalities.

3.3 QUINTET ROOTING

3.3.1 General algorithmic design for Quintet Rooting

We propose a general class of methods for rooting that can be used when given an unrooted 5-leaf species tree and a set of k unrooted 5-leaf gene trees: First, compute the empirical probability distribution of the unrooted gene tree topologies and then score each rooted version of the given species tree to determine how well its topology fits the distribution as predicted by the ADR theory. Computing the (empirical) distribution on gene tree topologies is straightforward: divide the frequency for each unrooted gene tree topology by k . Scoring a rooted species tree, however, presents several non-trivial challenges. The first challenge is computational: each rooted species tree defines a different set of invariants and inequalities, and these must be calculated separately. The more significant challenge is how to define the fit between the ADR invariants and inequalities and a given rooted species tree so that the rooted species tree with the best fit to the input data is likely to be the true species tree. As we will see in the next section, defining the fit appropriately required that we correct for a topological bias in a naive definition of the fit.

In what follows, we will define cost functions for measuring the fit between the ADR invariants and inequalities of a rooted 5-leaf species tree and a given distribution of 5-leaf unrooted gene trees, g_1, g_2, \dots, g_k . Then, given a cost function, distribution of 5-leaf unrooted gene trees, and an unrooted 5-leaf tree T , we will seek the rooting of T that minimizes the cost. Quintet Rooting (QR) follows this design:

- Estimate the unrooted gene tree probability distribution $\vec{\hat{u}} = (\hat{u}_1, \hat{u}_2, \dots, \hat{u}_{15})$ from g_1, \dots, g_k .
- For a given cost function $\text{Cost}(R, \vec{\hat{u}})$ (for an example, see Equation 3.2), search all rooted versions of T to find \hat{R} such that

$$\hat{R} = \arg \min_R \text{Cost}(R, \vec{\hat{u}}) \quad (3.1)$$

and return \hat{R} as the [rooted](#) species tree.

In Section 3.3.3, we present extensions of this approach to enable us to root trees with more than five leaves.

3.3.2 Cost Function

The [ADR](#) invariants lead to equivalence classes for u_i values, examples of which are shown in Table 3.1. All the u_i values in one equivalence class must be the same, and the [ADR](#) inequalities suggest certain inequality relationships across these classes. We define C_R as the set of equivalence classes for a 5-taxon [rooted](#) species tree R . For example, according to Table 3.1, the set of equivalence classes for the caterpillar tree $R_1 = (((a, b), c), d), e)$ is given by

- $C_{R_1} = \{\{u_1\}, \{u_2\}, \{u_3\}, \{u_4, u_{13}\}, \{u_6, u_9\}, \{u_5, u_{12}\}, \{u_7, u_8, u_{10}, u_{11}, u_{14}, u_{15}\}\}$.

For two classes $c, c' \in C_R$ we write $c > c'$ if the inequalities for that specific tree state that each u_i value in class c must be larger than each u_i value in class c' . For instance, for R_1 , the inequalities in Table 3.1 requires that $\{u_1\} > \{u_4, u_{13}\}$.

Each of the 105 possible 5-taxon [rooted](#) trees has a unique set of equivalence classes, meaning that no two C_R 's are exactly the same when considering the inequalities that hold among classes (see Section 3.8 for a complete list of equivalence classes for all trees).

For a given vector of estimated [gene tree](#) probabilities $\vec{u} = (\hat{u}_1, \hat{u}_2, \dots, \hat{u}_{15})$ and a [rooted](#) tree R , we seek to define a *cost* that measures the degree to which \vec{u} violates the [ADR](#) invariants and inequalities defined for that [rooted](#) tree R . Therefore, we seek to find a [rooted](#) species tree \hat{R} for which this cost is minimized, which we will interpret as indicating that \hat{R} best explains the given [unrooted gene tree](#) distribution according to [ADR](#) theory. If we let $\text{Cost}(R, \vec{u})$ denote this cost function, then we want $\text{Cost}(R, \vec{u}) = 0$ to indicate that the given [gene tree](#) distribution satisfies the [ADR](#) invariants and inequalities for that [rooted](#) species tree perfectly, and we want $\text{Cost}(R, \vec{u}) > 0$ to indicate that the [gene tree](#) distribution violates some of these [ADR](#) invariants and inequalities.

In Experiment 1 (see Section 3.4), we explored different cost functions on our training datasets (based on a mammalian simulation); here we present the one we selected.

$$\text{Cost}(R, \vec{u}) = \underbrace{\sum_{c \in C_R} \frac{1}{|c|} \sum_{u_a, u_b \in c} |\hat{u}_a - \hat{u}_b|}_{\text{Invariants Penalty}} + \underbrace{\sum_{c > c' \in C_R} \frac{1}{|c'|} \sum_{u_a \in c, u_b \in c'} \max(0, \hat{u}_b - \hat{u}_a)}_{\text{Inequalities Penalty}} \quad (3.2)$$

The “Invariants Penalty” component of Equation 3.2 considers a penalty for each pair \hat{u}_a, \hat{u}_b that are in the same equivalence class for R but have different values in $\vec{\hat{u}}$. Since membership in the same equivalence class indicates that $u_a = u_b$, the penalty for any pair \hat{u}_a, \hat{u}_b with different values is the magnitude of their difference. Summing these penalty terms is a natural way of penalizing the variation of u_i ’s inside equivalence classes. The “Inequalities Penalty” component of Equation 3.2 is defined similarly, this time considering the inequalities between classes. For two classes c and c' where $c > c'$, we expect to have $\hat{u}_a > \hat{u}_b$ for all $u_a \in c, u_b \in c'$ for the true tree (provided we have a good enough estimation of the probability distribution on quintet tree topologies). Therefore, we consider a penalty term equal to $\hat{u}_b - \hat{u}_a$ when this relationship is reversed (which means $\hat{u}_b - \hat{u}_a$ is a positive penalty). Note that all the individual penalty terms will be 0 when scoring the true [species tree](#) with respect to the true [gene tree](#) distribution.

Normalization: correcting for bias. The number of invariant penalty terms (i.e., $|\hat{u}_a - \hat{u}_b|$) in the cost function provided in Equation 3.2 for a [rooted](#) tree R is equal to $\sum_{c \in C_R} \binom{|c|}{2}$. Therefore, the number of invariant penalty terms for caterpillar, balanced, and pseudo-caterpillar trees are 18, 23, and 31 respectively, which can be computed from class sizes in Table 3.1. Similarly, the number of inequality penalty terms for a tree R is equal to $\sum_{c > c' \in C_R} |c||c'|$. This leads to 28 inequality penalty terms for caterpillar trees, 44 for balanced trees, and 54 for pseudo-caterpillar trees. Therefore, the overall number of penalty terms varies significantly between different tree shapes (46, 67, and 85 respectively), which could lead the algorithm to generally assign smaller costs to one tree category, and return an output tree from that category with much higher probability. In other words, using a cost function without modification would produce a “category bias”.

To alleviate this problem, we have used a normalization factor $\frac{1}{|c|}$ for the sum of invariant terms in a class c and a factor $\frac{1}{|c'|}$ for the sum of inequality terms between classes c and c' , where c' is the class with smaller values. After including these factors, the *weighted* number of inequalities and invariants becomes 19, 20, and 19 for caterpillar, balanced, and pseudo-caterpillar trees respectively. Hence, this results in roughly the same number of invariant and inequality penalty terms for each tree category.

3.3.3 Extending to larger trees

We have explored two ways of extending Quintet Rooing to trees with more than five leaves. Both operate by examining each of the $2n - 3$ possible ways to root the input [unrooted](#) n -leaf tree T (i.e., on each of the $2n - 3$ edges), and then evaluates the cost of the

resultant **rooted** tree R by adding up the costs for a selected subset Q^* of the **rooted** quintet trees within R . We describe this as a two-step process: (1) Preprocessing: compute the cost for each **rooted** quintet tree using Equation 3.2 and (2) For each of the $2n - 3$ candidate **rooted** trees R corresponding to T , compute the score of R as below:

$$Score(R, T) = \sum_{q \in Q^*} \text{Cost}(q, \hat{u}_q) \quad (3.3)$$

where Q^* is the selected subset of **rooted** quintet trees and \hat{u}_q is the probability distribution of **unrooted** 5-taxon gene trees corresponding to a **rooted** quintet q . Thus, the two methods differ only in how the subset Q^* of **rooted** quintet trees is defined. The first way looks at all possible **rooted** quintet trees, and so has $\Theta(n^5)$ quintets to examine, but the second way looks at a carefully selected subset of $O(n)$ quintets (where the selection is based on the **unrooted species tree** being evaluated). Moreover, by calculating the costs of the **rooted** quintet trees in a preprocessing step, the first approach uses $O(kn^5)$ time and the second approach uses only $O(kn)$ time, for k gene trees and n species. Thus, while the first approach has the advantage that it relies on more information, it is much less computationally tractable than the second approach. We now describe the second approach.

Linear Encoding of Trees by Quintets. Since using all quintets for scoring different rootings can be computationally expensive for large trees, we propose a sparse sampling of quintets that leads to an optimized version of the Quintet Rooting algorithm with overall complexity of $O(nk)$. Our experimental study in Section 3.4 shows that this sparse sampling of quintets has a very similar accuracy to the original algorithm.

Encoding. Let R be a **rooted binary** tree with n leaves and let T denote its **unrooted** topology. For every edge e in $E(T)$, we define a quintet $q(e)$ of leaves in T so that e corresponds to a single edge in $q(e)$. The following cases can happen:

- (a) e is incident with a leaf x . In this case, e shares an endpoint with exactly two other edges, so these three edges together define a tripartition of the leafset of T into A, B, x , as shown in Figure 3.1. Form a quintet by picking x and at least one element from A and B , with the remaining two elements picked arbitrarily. Let $q(e)$ denote the tree **induced** on these five leaves by tree T .
- (b) e is not incident with any leaf, and so e shares an endpoint with exactly four other edges. These five edges together define a partition of the leafset of T into four sets

A_1, A_2, B_1 , and B_2 , as shown in Figure 3.1. Pick one leaf from each of the four sets and the remaining leaf from any of the sets arbitrarily.

For each edge e we have defined a quintet of leaves, and we will denote by $q(e)$ the tree induced on these five leaves by tree T . Note that the set of trees in these quintets $Q^* = \{q(e) : q \in E(T)\}$ uniquely defines the tree T . Note also that if each $q(e)$ is rooted correctly, then the rooted tree R can be inferred.

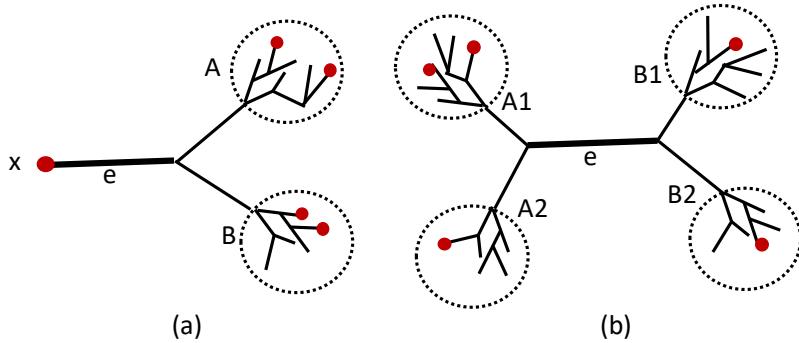


Figure 3.1: Linear mapping of edges in a tree to quintets of taxa. (a) edge e is adjacent to a leaf or (b) edge e shares an endpoint with four other edges.

The number of edges in an unrooted binary tree with n leaves is $2n - 3$, and therefore this encoding uses $O(n)$ quintets. Hence, the overall runtime of the Quintet Rooting algorithm with this sampling becomes $O(nk)$.

3.4 EXPERIMENTAL STUDY

Overview. In Experiment 1, we explored different cost functions for use in Quintet Rooting (QR) on a set of *training* simulated datasets; this produced our selected cost function, presented in Section 4.2.2, which we used in all subsequent experiments for QR. In Experiments 2 and 3, we compared QR to four other rooting methods (three based on distances and RootDigger, which is based on gene sequence alignments as well as distances). In Experiment 2, we evaluated rooting methods on simulated “testing” datasets with up to 30 leaves. In Experiment 3 we explored rooting methods on 5-leaf subsets of an avian biological dataset.

We used the avian and mammalian simulated datasets from Mirarab et al. (2014) [246] and the biological avian dataset from Jarvis et al. (2014) [247]. We created subsets of five and ten species each from these datasets to explore accuracy for rooting methods, using both true and estimated species trees. The model trees for the simulated datasets have model

parameters (topology and coalescent unit branch lengths) based on the trees constructed on associated biological datasets, as described in Mirarab et al. (2014) [246] and below. We used both true species trees and estimated species trees, as computed using ASTRAL [248], and we reported the “clade distance” between the estimated **rooted** trees and the true rooted trees as the error.

All experiments were run on the University of Illinois campus cluster, which limits each run to four hours, and explored the following questions: (1) How do different cost functions impact the accuracy of QR?, (2) How does the accuracy of the **species tree** impact the relative and absolute accuracy of the different rooting methods?, and (3) How do the number of species and **gene tree estimation error** impact the relative and absolute accuracy of the different rooting methods?

Evaluation Criteria. We use the normalized clade distance between the estimated **rooted** tree T and the true **rooted** tree T^* as the main measure of error, where clade distance is the natural extension of the standard **Robinson-Foulds (RF)** [83] error rate that is used to evaluate methods for estimating **unrooted** trees. Thus, the normalized clade distance between two **binary rooted** trees T^* and T , both on the same set of n leaves, is given by:

$$\frac{|Clades(T^*) \setminus Clades(T)| + |Clades(T) \setminus Clades(T^*)|}{2n - 4} \quad (3.4)$$

where $Clades(t)$ denotes the set of **clades** of the **rooted** tree t . Thus, the normalized clade distance is a value between 0 and 1, and indicates the fraction of the non-trivial **clades** in the estimated **rooted** tree that are not in the true **rooted** tree. When evaluating error in rooting the **unrooted** true tree t , another technique that can be used is the **root distance**, which is the distance in t between the edge containing the correct root location and the estimated root location. Letting T denote the result of applying a method to root t and letting T^* denote the true **rooted** tree, the clade distance between T and T^* is twice the root distance between T and T^* , as we prove in Lemma 3.1.

Equivalence of root distance and clade distance. Let R be a **rooted binary** tree with n leaves. For every node v , let R_v denote the subtree of tree R below node v and $\mathcal{L}(R_v)$ denote the leafset of subtree R_v . The set of **clades** of R is defined as $Clades(R) = \{\mathcal{L}(R_v) : v \in V(R)\}$.

For two **binary** trees R and R' with n leaves and the same **unrooted topology** T , we define the **root distance** of R and R' , denoted by $RD(R, R')$, as the number of nodes on the path between the root nodes of these trees. We also define the **clade distance** of R and R' , denoted by $CD(R, R')$, as the symmetric difference between $Clades(R)$ and $Clades(R')$;

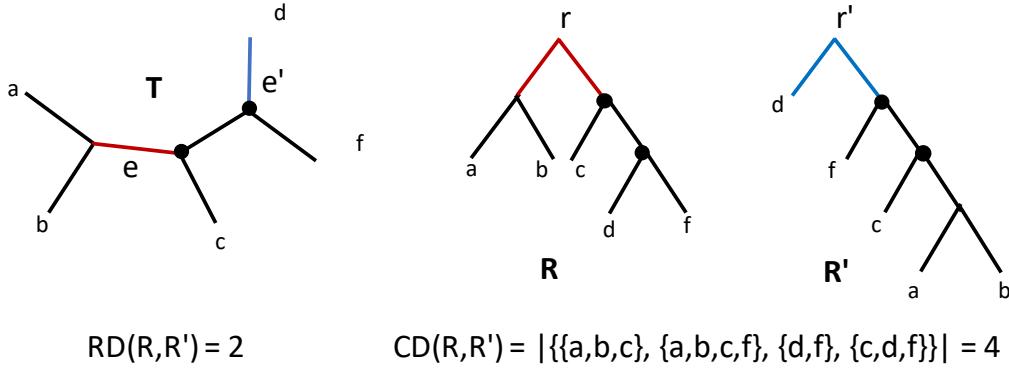


Figure 3.2: Relationship between clade distance and root distance for an example 5-taxon tree T , with different rootings R and R' . Only nodes on path P between r and r' define different clades, and are counted twice in the clade distance between two trees.

thus $CD(R, R') =$

$$|Clades(R) \Delta Clades(R')| = |Clades(R) \setminus Clades(R')| + |Clades(R') \setminus Clades(R)| \quad (3.5)$$

Lemma 3.1. For rooted binary trees R and R' with unrooted topology T , we have $CD(R, R') = 2RD(R, R')$.

Proof. Let r denote the root node of R and r' denote the root of R' . Since R and R' have the same unrooted topology T , let e and e' be edges of the tree T containing the roots r and r' (a rooted tree can be created by picking up the unrooted topology at any edge). Let $P = \{v_1, v_2, \dots, v_k\}$ be the set of nodes on the path between e and e' in T ; note that $|P| = k$ is the root distance between R and R' . For every vertex $v \in V(T)$, we can consider the clades in R and R' that are rooted at v , and the clade distance between R and R' is twice the number of all nodes that define different clades. However, it is easy to see that any node $v \notin P$ defines a clade that is in both R and R' , and also every node $v \in P$ defines different clades in R and R' . Hence, the clade distance between R and R' is $2|P|$, so that $CD(R, R') = 2|P| = 2 \times RD(R, R')$. QED.

In our experiments, however, we are also interested in rooting estimated species trees, and for this case we cannot use the root distance to measure error. For this reason, we use the clade distance (in its normalized form), which allows us to evaluate error in both cases. We also reported the proportion of test cases in which the tree was correctly rooted as a measure of accuracy for each method; results for this criterion show nearly identical relative performance as for the clade distance criterion, and are provided in Section 3.8.

Biological Dataset. We used the biological dataset studied in [247] containing 48 avian species and 4 non-avian outgroups (American Alligator, Green Sea Turtle, Green Anole Lizard, and Human). Jarvis et al. (2014) [247] used 14,446 genes. (8251 exons, 2516 introns, and 3679 ultraconserved elements (UCEs)); the main tree produced (referred to as the total evidence nucleotide tree (TENT)) was constructed using maximum likelihood, and has branch lengths and branch support values. Because of the substantial levels of gene tree heterogeneity (e.g., every estimated gene tree was different from the estimated species tree) and because the estimated species tree had many very short branches suggestive of a rapid radiation which would produce high ILS, the avian dataset is considered to be a good example of a dataset with a high level of ILS [247]. The gene trees in this dataset exhibited exceptionally low branch support (on average about 32%), due to low rates of evolution in the exons and UCEs [247], so that this is a challenging dataset for methods that are based on estimated gene trees.

We produced 12 subsets, each a random selection of 5 avian species. For each set of 5 avian species, we included any gene from the 14,446 genes that had all 5 species; this resulted in varying numbers of genes (ranging from approximately 10K to 13K genes) for each of the five-species subsets. For each selected subset of five species, we gave the unrooted TENT (restricted to those five species) to each of the rooting methods. We provided the published estimated gene trees to QR (after restriction to the selected five species) and we derived branch lengths on the 5-leaf trees using the implied branch lengths in the TENT for the distance-based methods. To evaluate accuracy, we used the 48-species TENT, rooted at the edge leading to the outgroup, as the “true tree”.

Simulated Datasets. We used mammalian simulated datasets for Experiment 1 (designing QR) and avian simulated datasets for Experiments 2 and 3 (evaluating QR in comparison to other methods). These datasets were generated by Mirarab et al. (2014) [246], and the true species trees, estimated and true gene trees, and sequence alignments per gene are available at [249]).

Here we briefly describe how Mirarab et al. (2014) [246] produced these datasets. The mammalian simulated datasets were evolved down a 37-species model tree based on a species tree constructed in [250] and the avian simulated datasets were evolved down a 48-species model tree based on the TENT constructed for Jarvis et al. (2014) [247]. (The Mirarab et al. (2014) [246] paper varied the ILS level by rescaling the branch lengths before simulating gene evolution, but here we only use the initial (default) species tree branch lengths.) 1000 gene trees were evolved down these model trees under the MSC, and the branch lengths were modified to create deviations from the molecular clock. Sequences of varying lengths

were then evolved down each [gene tree](#) under a GTRGAMMA model of sequence evolution. Estimated gene trees were produced for each gene sequence alignment using maximum likelihood. Thus, the public repository contains true and estimated gene trees, true alignments, and the true species trees (with branch lengths) for the mammalian and avian simulated datasets.

In Experiment 1 we only used true gene trees from the mammalian simulation, but in Experiment 2 we used both estimated and true gene trees from the avian simulation. We report the average [gene tree estimation error \(GTEE\)](#) for each model condition, where [GTEE](#) is the [RF](#) error rate between true and estimated gene trees (i.e., the percentage of the non-trivial [bipartitions](#) in the true [gene tree](#) that are not found in the estimated gene tree). For the avian simulation, sequence lengths ranging from as long as 1600 down to 250 were provided, so that [GTEE](#) rates for the [ML](#) trees ranged from 30% to 67%. The [ILS](#) levels for these model conditions is reported using [average distance \(AD\)](#), defined as follows: [AD](#) is the average normalized bipartition distance between the true [species tree](#) and true gene trees. Thus, [AD](#) measures the percentage of [bipartitions](#) defined by [internal](#) branches in the [species tree](#) that are not in the true gene trees. The [ILS](#) level for the mammalian simulation used in our experiments is 29%, which is a moderate level of [ILS](#), and the [AD](#) level for the avian simulation is 47%, which is moderately high.

To generate datasets with k -leaf trees, we randomly sampled k species from the species set and extracted the [induced](#) subtrees on these [taxa](#) from the model [species tree](#) and all gene trees in each model condition. For the 5-taxon datasets, we generated 20 replicates (corresponding to the 20 replicates in the original datasets) of a dataset with 1000 random samples, and for all other datasets, we generated 20 replicates with 200 samples.

Methods. We compared QR to other rooting methods: midpoint rooting (Midpoint) as provided by the FastRoot [241] package, minimum variance (MinVar) rooting [241], minimum ancestor deviation (MAD) [79], and RootDigger [243]. The software versions and commands are provided in Section 3.7. We did not include [outgroup](#) rooting, as it needs additional information about the [taxa](#) in the species set. We did not include STRIDE [222], as it cannot be used on single-copy gene trees, and we did not include the rooting method from Tian and Kubatko (2017) [244], as the software is not publicly available.

These methods require different types of input. The input to QR is a set of [unrooted](#) gene trees in addition to an [unrooted species tree](#) topology. The input to MinVar, MAD, and Midpoint is an [unrooted](#) tree with branch lengths. Following the recommendations in Binet et al. (2016) [251], who found that maximum likelihood was one of the two most accurate techniques for estimating branch lengths in species trees, we used RAxML (under

GTRGAMMA) to estimate branch lengths on the given species trees, using a concatenated alignment of all gene sequences. Finally, the input to RootDigger is an **unrooted** tree with branch lengths as well as a multiple sequence alignment. To produce a multiple sequence alignment for RootDigger, we concatenated gene sequence alignments for all genes in a replicate. RootDigger has two modes of running that are called “search” and “exhaustive” modes. The default search mode provides a prediction of the root location quickly using heuristics, with early stopping on by default, and the exhaustive mode can be used to do a more thorough search and compute the confidence probabilities for the predicted root position. We ran RootDigger in both of these modes in our experiments, although the original study [243] only compared RootDigger in the “search” mode with other methods. Our experiments included both the true **unrooted species tree topology** and an estimated **unrooted** species tree, computed by ASTRAL on the given gene trees.

3.5 RESULTS AND DISCUSSION

3.5.1 Experiment 1: Designing the cost function

Recall that QR can be used with any given cost function; hence, here we compare four different cost functions to understand the impact of the cost function on the final accuracy. Each cost function is defined by its own way of weighting the different penalties for violating invariants or inequalities. $Cost_1$ only considers penalties for the invariants and not the inequalities, $Cost_2$ considers both but does not normalize them, and $Cost_3$ and $Cost_4$ consider both types but use different weighting schemes.

$$Cost_1(R, \vec{\hat{u}}) = \underbrace{\sum_{c \in C_R} \frac{1}{|c|} \sum_{u_a, u_b \in c} |\hat{u}_a - \hat{u}_b|}_{\text{Invariants Penalty}} \quad (3.6)$$

$$Cost_2(R, \vec{\hat{u}}) = \underbrace{\sum_{c \in C_R} \sum_{u_a, u_b \in c} |\hat{u}_a - \hat{u}_b|}_{\text{Invariants Penalty}} + \underbrace{\sum_{c > c' \in C_R} \sum_{u_a \in c, u_b \in c'} \max(0, \hat{u}_b - \hat{u}_a)}_{\text{Inequalities Penalty}} \quad (3.7)$$

$$Cost_3(R, \vec{u}) = \underbrace{\sum_{c \in C_R} \frac{1}{|c|} \sum_{u_a, u_b \in c} |\hat{u}_a - \hat{u}_b|}_{\text{Invariants Penalty}} + \underbrace{\sum_{c > c' \in C_R} \frac{1}{|c|} \sum_{u_a \in c, u_b \in c'} \max(0, \hat{u}_b - \hat{u}_a)}_{\text{Inequalities Penalty}} \quad (3.8)$$

$$Cost_4(R, \vec{u}) = \underbrace{\sum_{c \in C_R} \frac{1}{|c|} \sum_{u_a, u_b \in c} |\hat{u}_a - \hat{u}_b|}_{\text{Invariants Penalty}} + \underbrace{\sum_{c > c' \in C_R} \frac{1}{|c'|} \sum_{u_a \in c, u_b \in c'} \max(0, \hat{u}_b - \hat{u}_a)}_{\text{Inequalities Penalty}} \quad (3.9)$$

The final cost function, $Cost_4$, is identical to the cost function given in Equation 3.2, and the first three cost functions are obtained by modifying $Cost_4$. $Cost_1$ only considers penalties for the invariants and not the inequalities, $Cost_2$ considers both but does not normalize them, and $Cost_3$ is similar to $Cost_4$ in structure (i.e., it considers penalties for both invariants and inequalities) but uses a different normalization scheme.

Figure 3.3 shows that QR using the first three cost functions produces biased results and overall lower accuracy: the rooting error rates (average normalized clade distance) on caterpillar trees are much lower than on balanced and pseudo-caterpillar trees. However, QR with $Cost_4$ has approximately the same rooting error across the four different shape categories and overall has the lowest rooting error.

This bias results from the number of penalty terms (based on invariants and inequalities) for the caterpillar category being smaller than the number of penalty terms for the other categories, so that appropriate normalization is needed to eliminate the bias. Clearly not using the inequalities ($Cost_1$) or not normalizing at all ($Cost_2$) does not produce overall good results. A comparison between $Cost_3$ and $Cost_4$ is also interesting, as each uses both types of penalty terms and differ only in how they weight the inequalities. The approach used in $Cost_4$ comes close to an equal weighted cost per category (see discussion in Section 4.2.2), explaining why using $Cost_4$ results in lower overall error and greatly reduced bias.

3.5.2 Experiment 2: Evaluation on simulated datasets

Here we compare QR to other rooting methods on 5-leaf and 10-leaf subtrees of the avian simulated datasets, with varying gene sequence lengths. We explore all conditions with both the true [species tree](#) and with the estimated [species tree](#) computed using ASTRAL. The distance-based methods and RootDigger were given branch lengths estimated by RAxML on the concatenated sequence alignments of all genes. Results shown for “true gene trees” reflect performance when QR is given 1000 true gene trees and the other methods (which all require branch lengths on the species trees) are given branch lengths based on gene sequences of length 1600; all other conditions reflect performance given shorter sequences.

Results for rooting 5-leaf trees. Figure 3.4 explores rooting error (computed using normalized clade distances) for three distance-based methods, two ways of running RootDigger, and QR, when rooting 5-leaf subtrees of the true [species tree](#) and varying the sequence length.

As expected, gene trees computed on shorter sequences have higher [GTEE](#). Given true gene trees, QR has very low rooting error, but as [GTEE](#) increases its rooting error also increases. Even so, QR has much lower error than the other methods when [GTEE](#) is not high (i.e., [GTEE](#) less than 54%). However, for [GTEE](#) = 54%, QR is better than the distance-

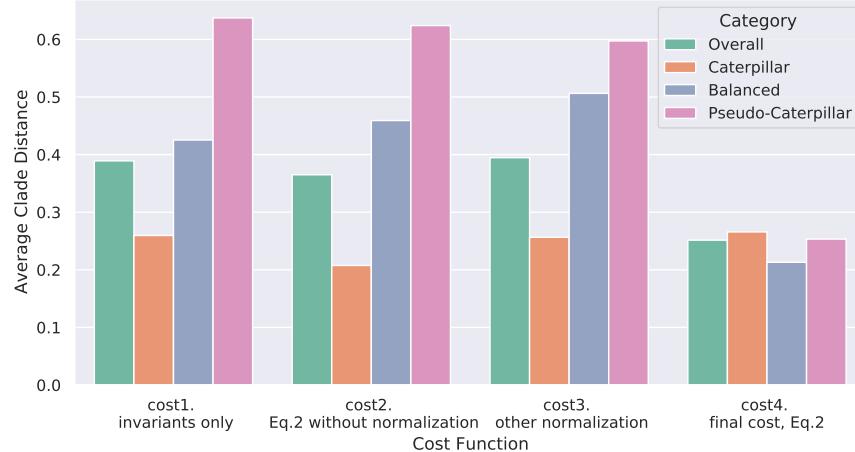


Figure 3.3: Average normalized clade distance for Quintet Rooting on mammalian simulated datasets using four different cost functions, across different shape categories. The results are shown across 1000 sample 5-leaf trees with 800 true gene trees. The ratio of caterpillar, balanced and pseudo-caterpillar trees in this dataset is 53.8%, 21.2% and 25% respectively. The [ILS](#) level is 29% [AD](#). This figure shows the effect of *Category Bias*, and the importance of how the cost function is defined. Based on this experiment, we selected *Cost*₄ as our cost function.

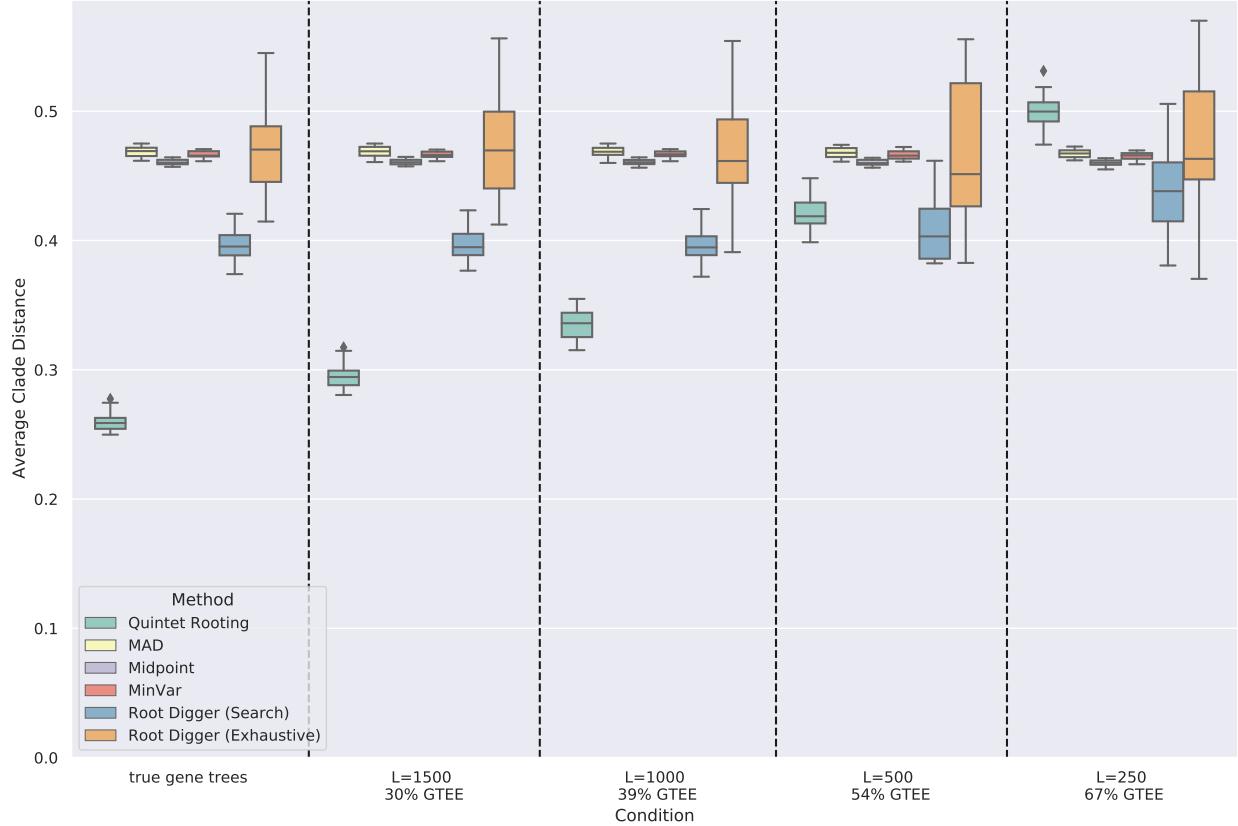


Figure 3.4: Average normalized clade distance of rooting methods on 5-leaf subsets of the avian simulated datasets when given the true species tree topology. The number of genes is 1000 and the error bars are shown across 20 replicates, each on 1000 samples. Branch lengths are estimated using RAxML on the concatenated alignment (CA) of the gene sequence alignments. Results shown for “true gene trees” reflect Quintet Rooting given 1000 true gene trees and the other methods given branch lengths estimated on the true species tree using gene sequences of length 1600.

based methods but slightly worse than RootDigger run in its default search mode, and when [GTEE = 67%](#) then QR is less accurate than the other methods.

Other trends in Figure 3.4 are also worth noting. For example, the three distance-based methods (MAD, Midpoint, and MinVar) do not seem impacted by the gene sequence length, so that at all sequence lengths the rooting error is very high. RootDigger run in search mode is impacted by sequence length, but not when run in exhaustive mode, and RootDigger run in exhaustive mode is much less accurate than RootDigger run in search mode for all sequence lengths (with big differences for longer sequences). Finally, although the differences between the distance-based methods are very small and to some extent depends on the model condition, Midpoint rooting is slightly more accurate than both MinVar and MAD, and when there was a difference between MinVar and MAD it tended to favor MinVar.

Figure 3.5 shows a comparison between methods when rooting 5-leaf estimated species trees computed by ASTRAL. In this experiment we omitted RootDigger in exhaustive mode due to its poor accuracy when given the true species tree. Here we observe the same trends as the experiments on the true species tree, with the relative performance between methods remaining the same. In particular, we still see QR having lower error than the other methods except for high GTEE (54%) and very high GTEE (67%), and Midpoint rooting more accurate than the other remaining methods. We also see that increases in sequence length improve accuracy for QR but do not impact the distance-based methods, and have only a minor impact on RootDigger.

Results for rooting larger trees. We now discuss results on rooting trees with 10 to 30 species. We begin with 10-leaf subtrees, comparing two ways of running QR, the three distance-based methods, and RootDigger in search mode. When rooting the true species tree (Figure 3.6), both ways of running QR (i.e., looking at all five-leaf subtrees or the linear encoding) are more accurate than the other methods (except for the very high GTEE condition), and the linear encoding version is less accurate than the version that uses all five-leaf subtrees. Because both ways of running QR are (relatively) close in accuracy and have the same relative performance to other methods, we will refer to them jointly as “QR” henceforth.

RootDigger is the least accurate, and the three distance-based methods are in between. For the highest GTEE rate (67%), QR is less accurate than the distance-based methods but still more accurate than RootDigger. The relative performance between distance-based methods is also noteworthy: Midpoint is the most accurate, followed by MinVar, and then by MAD. Changes in sequence length impact QR but not the distance-based methods, and impact RootDigger only slightly. Overall these trends are similar to trends observed on 5-leaf trees, except that there are larger differences between the three distance-based methods and QR maintains its superior accuracy until the highest GTEE.

Figure 3.7 presents results when rooting 10-leaf trees computed by ASTRAL, instead of the true species trees. We see the same relative performance as before, but error rates are slightly higher than when rooting true species trees (which is unsurprising). And as when rooting true species trees, both ways of running QR are more accurate than the other methods except for the highest GTEE.

Results on trees with up to 30 leaves are shown in Figure 3.10, where we evaluate QR with the linear encoding in comparison to the other rooting methods when rooting subsets of the true species tree and given 1000 genes of length 1000. Note that QR with the linear encoding differs from default QR only on trees with more than 5 leaves, so that results for

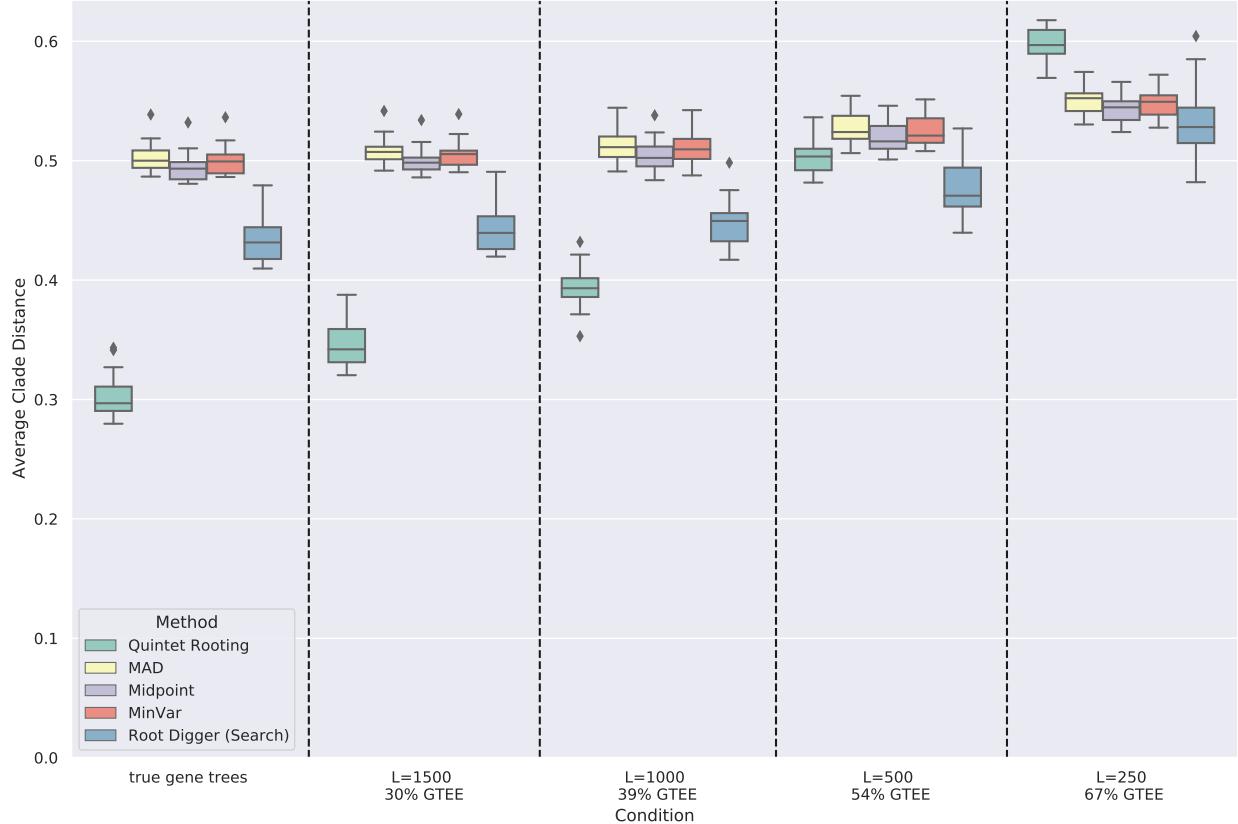


Figure 3.5: Average normalized clade distance on 5-leaf avian simulated datasets by each rooting method given an estimated species tree computed by ASTRAL. The branch lengths on the estimated species tree are estimated using RAxML with the concatenated gene multiple sequence alignments. The results are averaged over 1000 sample 5-taxon trees. The number of genes is 1000 and the error bars are shown across 20 replicates.

rooting 5-leaf trees are identical to results in Figure 3.5. Here we focus on how the trends on the larger trees, and especially on results with more than 10 leaves.

QR using the linear encoding produces more accurate rootings than the other methods. Midpoint is the most accurate of the remaining methods, and RootDigger is the least accurate method (except when rooting 5-leaf trees). All methods have relatively high error for 5-leaf trees, but improve in accuracy as the tree size increases. Although there is a large gap between QR and the next best method starting at 10-leaf trees, the gap between QR and the other methods decreases with the tree size. However, even at 30 leaves, the error rate for QR is very low (6.2%) and about half that of the next best method, Midpoint (12.3%).

Discussion of results on simulated datasets. A comparison of results across these different conditions is informative. We see that QR run using all quintets is slightly more

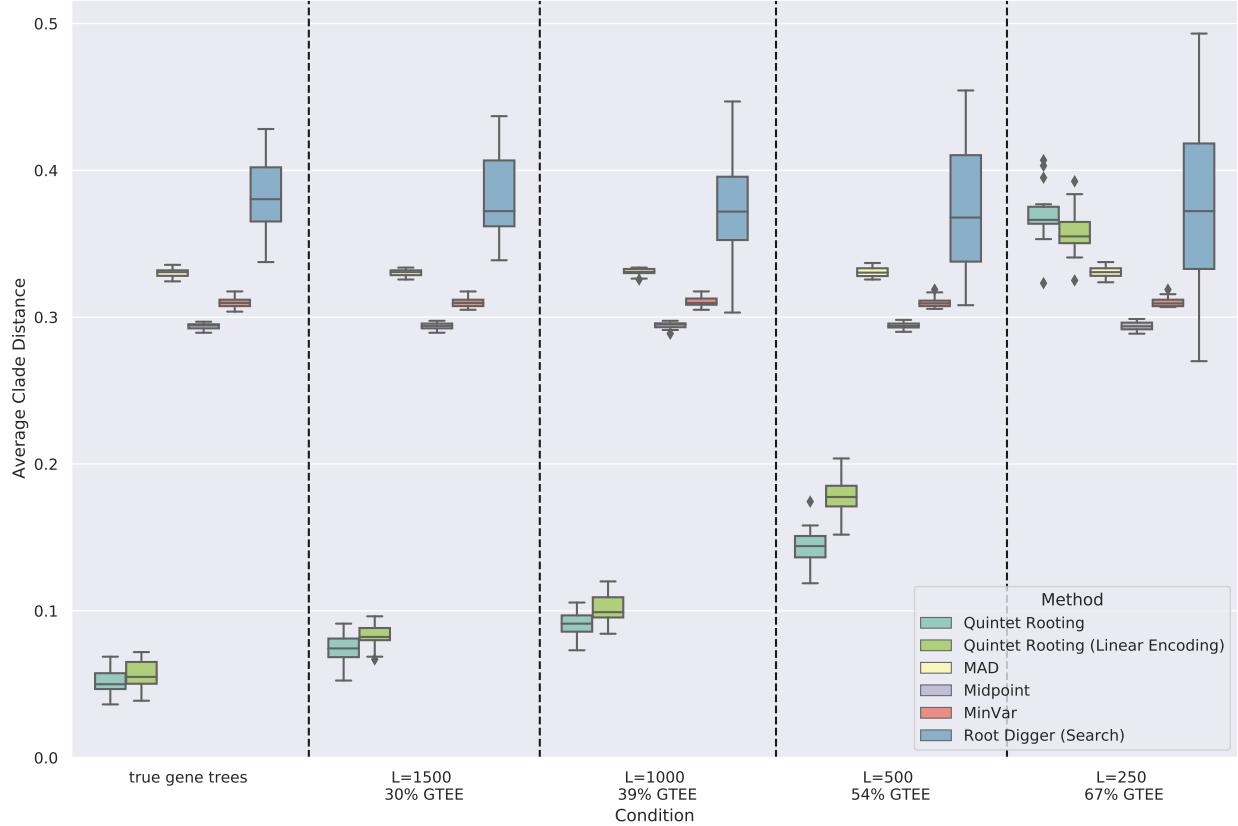


Figure 3.6: Average normalized clade distance on 10-leaf avian simulated datasets given the true (model) species tree by each rooting method. The results are averaged over 200 sample 10-species trees. The number of genes is 1000 and the error bars are shown across 20 replicates. The branch lengths are estimated using RAxML on the concatenated gene sequence alignments.

accurate than QR using the linear encoding and that both ways of running QR are much more accurate than all the other methods whenever GTEE is at most moderate. When GTEE is sufficiently high (i.e., at least 54%), then the relative performance between QR and the other methods depends on the number of species and whether true or estimated species trees are used. Specifically, QR maintains an advantage even at GTEE=54% when rooting estimated species trees or when given 10-leaf trees. The distance-based methods were generally close in accuracy, but comparisons in terms of clade distances showed a consistent pattern of Midpoint somewhat more accurate than MinVar, and MinVar more accurate than MAD. Another trend that is consistently seen across all these conditions is that sequence length per gene always impacts QR, but does not impact distance-based methods, and has a very minor impact on RootDigger.

Some aspects of the relative performance, however, depended on the model condition. As

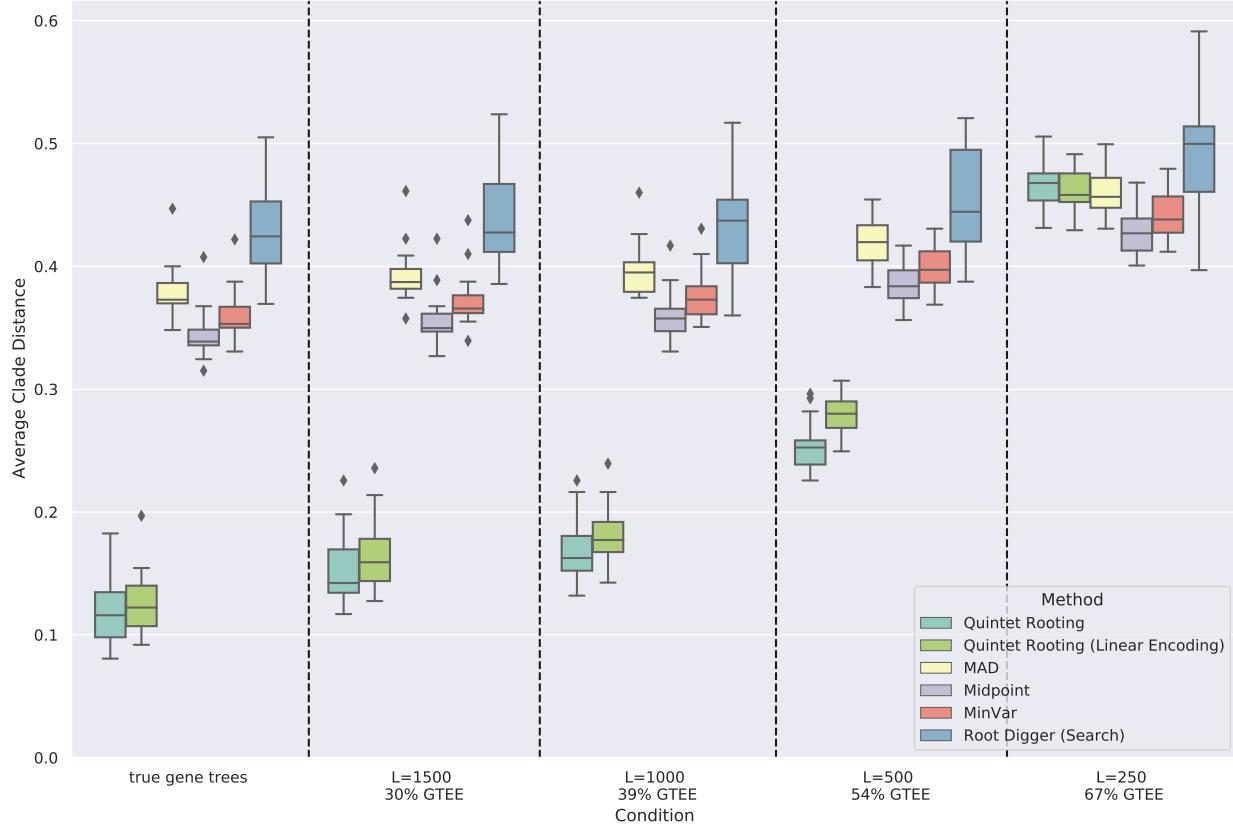


Figure 3.7: Average normalized clade distance on 10-leaf avian simulated datasets by each rooting method given an estimated species tree computed by ASTRAL. The branch lengths on the ASTRAL tree are estimated using RAxML with the concatenated gene multiple sequence alignments. The results are averaged over 200 sample 10-species trees. The number of genes is 1000 and the error bars are shown across 20 replicates.

an example, while RootDigger run in default mode was more accurate than the distance-based methods on 5-leaf trees, it was less accurate when rooting 10-leaf trees. We also saw that QR, using the linear encoding, was able to root trees with up to 30 leaves, and maintained an advantage over the other methods across all tree sizes we tested. However, the gap between QR and the next best method, Midpoint, decreased with the tree size. Finally, we observed that differences between methods were larger on true species trees than on estimated species trees, but relative performance remained the same.

In interpreting these trends, we make the following hypotheses. Clearly QR is impacted by GTEE, which is why sequence length has an impact on it. However, the lack of impact on distance-based methods as well as the very minor impact on RootDigger (which also uses branch lengths) suggest that using 1000 genes is sufficient, even for short gene alignments, to produce a reasonable estimate of the branch lengths of the given species tree. The

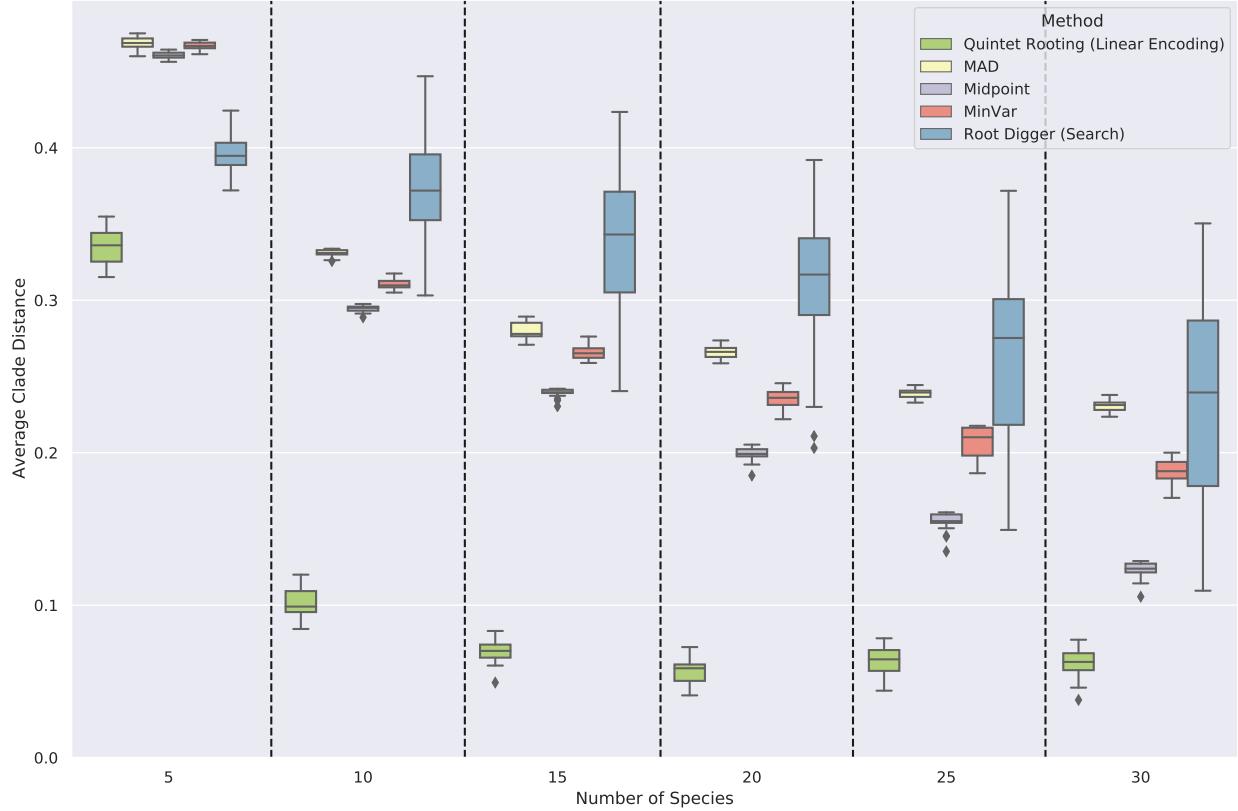


Figure 3.8: Average normalized clade distance on subsets of the avian simulated datasets with 5 to 30 leaves for each rooting method, given the true species tree [topology](#) and estimated gene trees with $L=1000$ (39% [GTEE](#)). The branch lengths on the model tree are estimated using RAxML with the concatenated gene multiple sequence alignments. The results are averaged over 200 samples for each number of species. The number of genes is 1000 and the error bars are shown across 20 replicates.

improvement in accuracy for the distance-based methods and even RootDigger as the tree size increases suggests that these methods benefit from denser taxon sampling, which may break up long branches and possibly improves branch length estimation.

3.5.3 Experiment 3: Evaluation on the biological dataset

Because we observed that QR is impacted by [gene tree](#) estimation error, we selected a biological dataset in which this was likely to be a challenge for it: the Avian Phylogenomics project dataset studied in [247]. Gene tree branch support (measured using bootstrapping) was on average about 32% [246], resulting from low phylogenetic signal in the gene sequences, and suggesting that the gene trees likely had high [gene tree estimation error](#) [246]. Evaluating how well QR performed on this dataset would give us an estimate of its reliability under

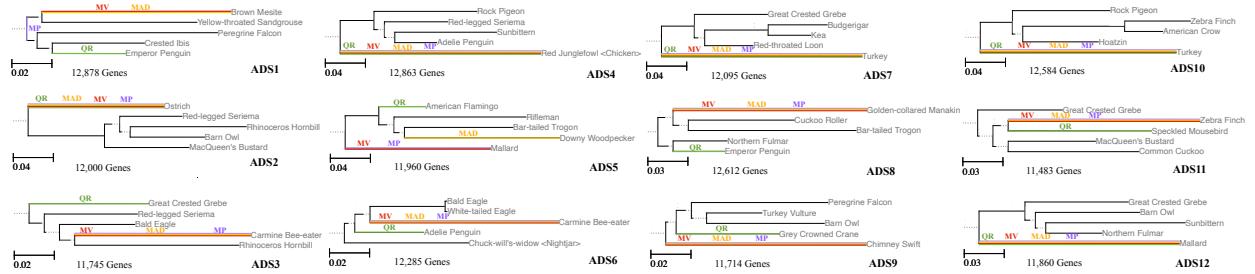


Figure 3.9: Analyses on the 5-leaf subsets of the [total evidence nucleotide tree \(TENT\)](#) computed on the avian biological dataset by [247]. The branch lengths are obtained from the published [TENT](#) species tree using Dendropy [252]. The branch selected as the root by each method is color-coded. The number of genes used in each analysis (defined to be those genes that have all 5 of the selected species) is also shown beside each figure. Results are shown for Quintet Rooting (QR), Midpoint (MP), MinVar (MV), and Minimum Ancestor Deviation (MAD). The trees are visualized using ETEToolkit v3 [253].

very challenging conditions. Here we note that many other phylogenomic datasets are not as challenging; for example, the average branch support in the Thousand Plant Transcriptome project from Wickett et al. (2014) [109] was much higher.

We selected 12 random subsets of 5 avian species each from the maximum likelihood “TENT” tree computed on the genome-scale data, and used the outgroups to root the trees as the “reference [rooted](#) tree” for evaluation purposes. The normalized clade distances for QR and the three distance-based rooting methods on the 12 avian subtrees are provided in Table 3.2, with the trees shown in Figure 3.9. The two methods with the lowest average error are QR and Midpoint, but MinVar is close behind and Minimum Ancestor Deviation (MAD) is in last place. One of the striking observations is that the distance-based methods tend to root the tree on the longest branch in the tree, but this is not true for QR. We also see that there are datasets for which no method is able to find the correct root (see ADS8 and ADS11, for example).

The trends on these biological datasets are consistent with results on the simulated datasets. As noted, the biological dataset gene trees have low bootstrap support indicating that for these datasets the [gene tree estimation error](#) is probably in the 50-80% range. Thus, the corresponding cases among the simulated data are when [GTEE](#) is high or very high. On the biological dataset, QR was in first place but tied with Midpoint Rooting (MP), with MinVar somewhat less accurate than QR and MP, and then MAD in last place. On simulated datasets, we saw QR in first place except when [gene tree estimation error](#) is high or very high. The relative performance between the three distance-based methods is also similar on the biological as well as simulated data, since MP was first in the simulated data,

followed by MinVar and then by MAD, which is what we see here.

3.6 CONCLUSIONS

We have presented Quintet Rooting (QR), a polynomial time method for rooting a given species tree that uses phylogenetic invariants and inequalities established by Allman et al. (2011) [81] under the [Multi-Species Coalescent \(MSC\)](#) model. The trends observed on both biological and simulated data suggest that QR is a promising approach to rooting species trees under a range of model conditions, generally more accurate than all the tested competing methods except when [GTEE](#) is very high, in which case all methods have poor accuracy.

The study suggests several directions for future work. For example, we have not established whether QR is a [statistically consistent](#) method for rooting species trees, not even for the simplest case of 5-leaf trees. We conjecture that this is the case, but a proof is needed. In addition, alternative cost functions need to be explored to determine if even more accurate methods can be developed and established statistically consistent.

Further study under a wider range of conditions is also needed. In particular, although QR (using the linear encoding) was able to maintain an accuracy advantage over the competing methods with up to 30 leaves, the gap between QR and the next most accurate method had reduced, and it is possible that on much larger trees QR might be less accurate than

Table 3.2: Normalized clade distances are shown for Quintet Rooting, Midpoint, MinVar, and Minimum Ancestor Deviation (MAD) for rooting 5-leaf subtrees of the [TENT](#) avian tree. The reference tree is the [TENT](#) tree from Jarvis et al. (2014) [247], rooted on the edge leading to the outgroups.

| Dataset | #Genes | Quintet Rooting | Midpoint | MinVar | MAD |
|---------|---------|-----------------|----------|--------|------|
| ADS1 | 12,878 | 0.67 | 0.00 | 0.33 | 0.33 |
| ADS2 | 12,000 | 0.00 | 0.00 | 0.00 | 0.00 |
| ADS3 | 11,745 | 0.00 | 1.00 | 1.00 | 1.00 |
| ADS4 | 12,863 | 0.00 | 0.00 | 0.00 | 0.00 |
| ADS5 | 11,960 | 0.33 | 0.00 | 0.00 | 1.00 |
| ADS6 | 12,285 | 0.33 | 0.67 | 0.67 | 0.67 |
| ADS7 | 12,095 | 0.00 | 0.00 | 0.00 | 0.00 |
| ADS8 | 12,612 | 0.33 | 0.33 | 0.33 | 0.33 |
| ADS9 | 11,714 | 0.33 | 0.00 | 0.00 | 0.00 |
| ADS10 | 12,584 | 0.00 | 0.00 | 0.00 | 0.00 |
| ADS11 | 11,483 | 0.67 | 0.67 | 0.67 | 0.67 |
| ADS12 | 11,860 | 0.00 | 0.00 | 0.00 | 0.00 |
| Average | ~12,173 | 0.22 | 0.22 | 0.25 | 0.33 |

other methods. Thus, evaluating QR and other rooting methods on larger trees is needed. Evaluating methods when given larger numbers of genes is another important direction to consider, especially since many phylogenomic studies use 1000 or more genes to estimate species trees (e.g., Jarvis et al. (2014) [247] used 14,000 genes). Future work should also evaluate conditions with varying levels of ILS, and the model conditions we explored were based on datasets with moderately high ILS. Deviation from the strict molecular clock has the potential to impact all methods, even if the main impact would be on methods that try to minimize the deviation from the clock or from a relaxed clock model, and its impact should be explored systematically. Another question is why RootDigger performed poorly in this study. Since RootDigger depends on likelihood calculations, one possibility is inadequate search of the parameter space, but another possibility is model misspecification. All the methods we explore depend on stochastic models of evolution, either for computing distances or for estimating gene trees; hence, the impact of model misspecification should be explored more generally.

The theoretical foundations for the method, provided in Allman et al. (2011) [81], are specific to the MSC model, but this does not mean that the approach would not perform well under other conditions, such as cases where there is (for example) HGT or GDL. Future work should evaluate Quintet Rooting under a wider range of model conditions with these and other causes for gene tree discord.

Finally, a closely related problem is the estimation of the rooted species tree from a set of unrooted gene trees, under the MSC. This is a strictly harder problem than estimating a rooted species tree from rooted gene trees under the MSC, which has been addressed by methods such as MP-EST [46]. Hence, this is another direction for future work.

3.7 METHODS AND SOFTWARE COMMANDS

3.7.1 Methods for rooting trees

Minimum Ancestor Deviation. The python script for MAD (v2.2) software is available at <https://www.mikrobio.uni-kiel.de/de/ag-dagan/ressourcen>. The command we used is:

```
python3 mad.py <input-tree.tre> -n
```

Minimum Variance Rooting. MinVar and Midpoint rooting are both available as part of the FastRoot (v1.5) python package at <https://github.com/uym2/MinVar-Rooting>. The

command we used is:

```
python3 FastRoot.py -m MV -i <input-tree.tre> -o <output-tree.tre>
```

Midpoint Rooting. The command we used is:

```
python3 FastRoot.py -m MP -i <input-tree.tre> -o <output-tree.tre>
```

Quintet Rooting. Quintet Rooting (v1.0) is available at https://github.com/ytabatabae/Quintet_Rooting. The option `-sm LE` can be used to sample a sparse set of quintets using a linear encoding and runs considerably faster than the default mode. We used the following command:

```
python3 quintet_rooting.py -t <species-topology.tre>
-g <input-genes.tre> -o <output.tre> [-sm LE]
```

RootDigger. We used RootDigger (v1.7.0), which is available at <https://github.com/computations/root-digger>. This method has two modes, the default is search and the other can be identified with `--exhaustive`. We used the following command:

```
./rd --msa <msa-file.fasta> --tree <input-tree.tre>
--seed 4321 [--exhaustive]
```

3.7.2 Other commands

Species tree estimation. We used ASTRAL (v5.7.8) to estimate species trees. ASTRAL is available at <https://github.com/smirarab/ASTRAL> as a jar file. The command we used is:

```
java -jar astral.5.7.8.jar -i <input-genes.tre> -o <output.tre>
```

Branch length estimation. We used RAxML (v8.2.12) to estimate branch lengths on given species trees, using the concatenated alignment. RAxML is available at <https://github.com/stamatak/standard-RAxML>. We used the following command:

```
raxmlHPC-PTHREADS -f e -t species.tre -m GTRGAMMA -s alignment.fasta
-n RES -p 4321 -T 16
```

Calculating clade distance The following function was used to compute the normalized clade distance between two *rooted* trees, where `t1` and `t2` are Dendropy `rooted Tree` objects with `n` leaves, constrained to the same set of `taxa` (i.e. sharing the same `TaxonNamespace` object).

```
def normalized_clade_distance(t1, t2, n):
    t1.encode_bipartitions()
    t2.encode_bipartitions()
    return dendropy.calculate.treecompare.symmetric_difference(t1,
        t2) / (2n - 4)
```

Random subset selection We selected random subsets of `taxa` in a tree using Python's default `random` module with the following command:

```
sample_taxa = random.sample(taxon_set, n)
```

where `taxon_set` is the list of all `taxa` in the original tree and `n` is the size of the subset tree.

Extracting induced trees After selecting subsets of taxa, we extracted the `induced` subset trees using Dendropy's `extract_tree_with_taxa_labels` on a `Tree` object using the following command:

```
subtree = tree.extract_tree_with_taxa_labels(labels=sample_taxa,
    suppress_unifurcations=True)
```

where `tree` is the original `unrooted` species tree or the corresponding `unrooted` gene trees and `sample_taxa` are the labels selected at previous step. After constraining the tree to the selected taxa, the `suppress_unifurcations` option removes nodes with degree two and connects their two adjacent nodes and adjusts the edge lengths accordingly [252].

3.7.3 Biological dataset analyses

Experiment 3 is based on analyses of the avian phylogenomics project [247] dataset, where we attempted to root 5-leaf subsets of the `total evidence nucleotide tree (TENT)`. The data repository for the avian phylogenomics project (available at [254]) contains: (a) alignments for each gene, (b) maximum likelihood gene trees, and (c) the `TENT topology` as well as branch lengths. We used these for our rooting analyses.

We selected 12 different 5-leaf subsets of the ingroup species from the `TENT`, using the random subset selection method described above and obtained the 5-leaf `induced` subtrees of the species tree and gene trees as explained above. We used the branch lengths values

produced by `extract_tree_with_taxa_labels` function, that adjusts the branch lengths after `suppressing` unifurcations from the `induced` subtrees. [252].

We used a similar constraining and `suppressing` approach to get `induced` subtrees on the 14,446 gene trees (including all exons, introns and `UCEs`). Since some gene trees had missing taxa, for each data subset, only the gene trees that had all `taxa` in that dataset were considered, reducing the number of genes used in each analysis to a value ranging from 11k to 13k.

The trees in Figure 3.9 are visualized using the online `phylogenetic tree` viewer tool from ETEToolkit v3 [253] and are provided below in newick format (in `rooted` form, as `rooted` by the `outgroup` species). The mapping between the 5-letter codes of the leaves in these trees and the actual species names are available on the dataset repository at <http://gigadb.org/dataset/101041> in the `Scripts/namemap/name.csv` directory.

3.8 ADDITIONAL FIGURES AND TABLES

Table 3.3: 12 avian biological 5-leaf tree subsets from the Avian Phylogenomics project dataset studied in [247] in newick format. The mapping between 5-letter codes for the leaves in the trees and actual species names is provided at the original dataset repository.

| Dataset | Species Subtree (in newick format) |
|---------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ADS1 | ((MESUN:0.08694290426167792, PTEGU:0.071415818897068)100:0.00514859723893367, (FALPE:0.08306071570403102, (NIPNI:0.0423883355661155, APTFO:0.03349709647931403)100:0.005206840108496662)93:0.0032472606038615546)100:0.06743839999387245; |
| ADS2 | ((STRCA:0.12274985497254273, ((CARCR:0.05643578593379514, (BUCRH:0.10530648854898142, TYTAL:0.06741501268170091)94:0.0014309181653352882)100:0.006338016412161908, CHLUN:0.07664747894993916)100:0.06853118805605843)100; |
| ADS3 | ((PODCR:0.07220590793996046, (CARCR:0.05643578593379514, (HALLE:0.045566642130790286, (MERNU:0.10658290270301818, BUCRH:0.0994958728004391)100:0.0064298085261464575)100:0.0008117253877311482)100:0.0074308044743478935)100:0.06743839999387245; |
| ADS4 | ((COLLI:0.09853337060176218, (CARCR:0.06061932980428148, (EURHE:0.0976862721822968, PYGAD:0.04011775509160112)49:0.0011991271659342246)93:0.0032472606038615546)100:0.03161522291278337, GALGA:0.19209772733128544)100:0.035823177081089085; |
| ADS5 | ((PHORU:0.045008036433236975, (ACACH:0.12720876507668982, (APAVI:0.10459378362169088, PICPUP:0.1493872121305442)80:0.004687827119484025)100:0.0074308044743478935)100:0.03161522291278337, ANAPL:0.1400222823017829)100:0.035823177081089085; |
| ADS6 | ((((HALLE:0.0009993079648479574, HALAL:0.0011481494716151777)100:0.04456733416594233, MERNU:0.11301271122916463)100:0.004995269258217487, PYGAD:0.041316882257535346)93:0.0021544725416755696, CAPCA:0.07442083096028874)100:0.06853118805605843; |
| ADS7 | ((PODCR:0.07220590793996046, ((MELUN:0.05843745032225126, NESNO:0.04201569459493401)100:0.061839565146662875, GAVST:0.04769342999296146)93:0.0032472606038615546)100:0.03161522291278337, MELGA:0.19796647631793823)100:0.035823177081089085; |
| ADS8 | ((MANVI:0.12207721157673541, (LEPDI:0.06682837969931472, APAVI:0.10585054810092069)100:0.0034310626402542107)100:0.004183543870486339, (FULGL:0.03407937342341265, APTFO:0.03230433568829369)100:0.006399600899517001)93:0.07068566059773401; |
| ADS9 | ((((FALPE:0.07887717183354469, (CATAU:0.03200310265259699, TYTAL:0.06803420545930505)100:0.0008117253877311482)100:0.00530612570207519, BALRE:0.05940380102308789)88:0.0010318907100867183, CHAPE:0.11587887735386451)100:0.06853118805605843; |
| ADS10 | ((COLLI:0.09853337060176218, ((TAEGU:0.0664816204499164, CORBR:0.03848548508393826)100:0.1021859557356423, OPHHO:0.07635447103047725)88:0.0021246787722727033)100:0.03161522291278337, MELGA:0.19796647631793823)100:0.035823177081089085; |
| ADS11 | ((PODCR:0.07220590793996046, ((TAEGU:0.16336145048348355, COLST:0.12215804393381353)100:0.006338016412161908, (CHLUN:0.07443664039789928, CUCCA:0.1112755055198169)100:0.0022108385520398783)100:0.001092788062185985)100:0.06743839999387245; |
| ADS12 | ((PODCR:0.07220590793996046, (TYTAL:0.07302947471752254, (EURHE:0.0976862721822968, FULGL:0.03927984715699542)49:0.0011991271659342246)93:0.0032472606038615546)100:0.03161522291278337, ANAPL:0.1400222823017829)100:0.035823177081089085; |

3.8.1 Caterpillar Trees

Table 3.4: Equivalence classes for caterpillar 5-taxon **rooted** species trees. The u_i s are named according to the **unrooted** trees in Table 5 in [81]. A script for generating these classes is available in the Quintet Rooting software repository in Github.

| Tree | Equivalence Classes |
|-------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------|
| $R_1 \quad (((a, b), c), d), e)$ | $\{u_1\} > \{u_4, u_{13}\}$ $\{u_2\} > \{u_5, u_{12}\} > \{u_7, u_8, u_{10}, u_{11}, u_{14}, u_{15}\}$ $\{u_3\} > \{u_6, u_9\}$ |
| $R_2 \quad (((a, b), c), e), d)$ | $\{u_1\} > \{u_4, u_{13}\}$ $\{u_3\} > \{u_6, u_9\} > \{u_7, u_8, u_{10}, u_{11}, u_{14}, u_{15}\}$ $\{u_2\} > \{u_5, u_{12}\}$ |
| $R_3 \quad (((a, b), d), c), e)$ | $\{u_2\} > \{u_7, u_{14}\}$ $\{u_1\} > \{u_8, u_{11}\} > \{u_4, u_5, u_{10}, u_{12}, u_{13}, u_{15}\}$ $\{u_3\} > \{u_9, u_6\}$ |
| $R_4 \quad (((a, b), d), e), c)$ | $\{u_2\} > \{u_7, u_{14}\}$ $\{u_3\} > \{u_6, u_9\} > \{u_4, u_5, u_{10}, u_{12}, u_{13}, u_{15}\}$ $\{u_1\} > \{u_8, u_{11}\}$ |
| $R_5 \quad (((a, b), e), c), d)$ | $\{u_3\} > \{u_{10}, u_{15}\}$ $\{u_1\} > \{u_8, u_{11}\} > \{u_4, u_6, u_7, u_9, u_{13}, u_{14}\}$ $\{u_2\} > \{u_5, u_{12}\}$ |
| $R_6 \quad (((a, b), e), d), c)$ | $\{u_3\} > \{u_{10}, u_{15}\}$ $\{u_2\} > \{u_5, u_{12}\} > \{u_4, u_6, u_7, u_9, u_{13}, u_{14}\}$ $\{u_1\} > \{u_8, u_{11}\}$ |
| $R_7 \quad (((a, c), b), d), e)$ | $\{u_4\} > \{u_1, u_{13}\}$ $\{u_5\} > \{u_2, u_{12}\} > \{u_7, u_8, u_{10}, u_{11}, u_{14}, u_{15}\}$ $\{u_6\} > \{u_3, u_9\}$ |
| $R_8 \quad (((a, c), b), e), d)$ | $\{u_4\} > \{u_1, u_{13}\}$ $\{u_6\} > \{u_3, u_9\} > \{u_7, u_8, u_{10}, u_{11}, u_{14}, u_{15}\}$ $\{u_5\} > \{u_2, u_{12}\}$ |
| $R_9 \quad (((a, c), d), b), e)$ | $\{u_5\} > \{u_8, u_{15}\}$ $\{u_4\} > \{u_7, u_{10}\} > \{u_1, u_2, u_{11}, u_{12}, u_{13}, u_{14}\}$ $\{u_6\} > \{u_3, u_9\}$ |
| $R_{10} \quad (((a, c), d), e), b)$ | $\{u_5\} > \{u_8, u_{15}\}$ $\{u_6\} > \{u_3, u_9\} > \{u_1, u_2, u_{11}, u_{12}, u_{13}, u_{14}\}$ $\{u_4\} > \{u_7, u_{10}\}$ |
| $R_{11} \quad (((a, c), e), b), d)$ | $\{u_6\} > \{u_{11}, u_{14}\}$ $\{u_4\} > \{u_7, u_{10}\} > \{u_1, u_3, u_8, u_9, u_{13}, u_{15}\}$ $\{u_5\} > \{u_2, u_{12}\}$ |
| $R_{12} \quad (((a, c), e), d), b)$ | $\{u_6\} > \{u_{11}, u_{14}\}$ $\{u_5\} > \{u_2, u_{12}\} > \{u_1, u_3, u_8, u_9, u_{13}, u_{15}\}$ $\{u_4\} > \{u_7, u_{10}\}$ |
| $R_{13} \quad (((a, d), b), c), e)$ | $\{u_7\} > \{u_2, u_{14}\}$ $\{u_8\} > \{u_1, u_{11}\} > \{u_4, u_5, u_{10}, u_{12}, u_{13}, u_{15}\}$ $\{u_9\} > \{u_3, u_6\}$ |
| $R_{14} \quad (((a, d), b), e), c)$ | $\{u_7\} > \{u_2, u_{14}\}$ $\{u_9\} > \{u_3, u_6\} > \{u_4, u_5, u_{10}, u_{12}, u_{13}, u_{15}\}$ $\{u_8\} > \{u_1, u_{11}\}$ |

Table 3.5: Equivalence classes for caterpillar 5-taxon [rooted](#) species trees (continued).

| Tree | Equivalence Classes |
|---------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------|
| $R_{15} \quad (((((a, d), c), b), e)$ | $\{u_8\} > \{u_5, u_{15}\}$ $\{u_7\} > \{u_4, u_{10}\} > \{u_1, u_2, u_{11}, u_{12}, u_{13}, u_{14}\}$ $\{u_9\} > \{u_3, u_6\}$ |
| $R_{16} \quad (((((a, d), c), e), b)$ | $\{u_8\} > \{u_5, u_{15}\}$ $\{u_9\} > \{u_3, u_6\} > \{u_1, u_2, u_{11}, u_{12}, u_{13}, u_{14}\}$ $\{u_7\} > \{u_4, u_{10}\}$ |
| $R_{17} \quad (((((a, d), e), b), c)$ | $\{u_9\} > \{u_{12}, u_{13}\}$ $\{u_7\} > \{u_4, u_{10}\} > \{u_2, u_3, u_5, u_6, u_{14}, u_{15}\}$ $\{u_8\} > \{u_1, u_{11}\}$ |
| $R_{18} \quad (((((a, d), e), c), b)$ | $\{u_9\} > \{u_{12}, u_{13}\}$ $\{u_8\} > \{u_1, u_{11}\} > \{u_2, u_3, u_5, u_6, u_{14}, u_{15}\}$ $\{u_7\} > \{u_4, u_{10}\}$ |
| $R_{19} \quad (((((a, e), b), c), d)$ | $\{u_{10}\} > \{u_3, u_{15}\}$ $\{u_{11}\} > \{u_1, u_8\} > \{u_4, u_6, u_7, u_9, u_{13}, u_{14}\}$ $\{u_{12}\} > \{u_2, u_5\}$ |
| $R_{20} \quad (((((a, e), b), d), c)$ | $\{u_{10}\} > \{u_3, u_{15}\}$ $\{u_{12}\} > \{u_2, u_5\} > \{u_4, u_6, u_7, u_9, u_{13}, u_{14}\}$ $\{u_{11}\} > \{u_1, u_8\}$ |
| $R_{21} \quad (((((a, e), c), b), d)$ | $\{u_{11}\} > \{u_6, u_{14}\}$ $\{u_{10}\} > \{u_4, u_7\} > \{u_1, u_3, u_8, u_9, u_{13}, u_{15}\}$ $\{u_{12}\} > \{u_2, u_5\}$ |
| $R_{22} \quad (((((a, e), c), d), b)$ | $\{u_{11}\} > \{u_6, u_{14}\}$ $\{u_{12}\} > \{u_2, u_5\} > \{u_1, u_3, u_8, u_9, u_{13}, u_{15}\}$ $\{u_{10}\} > \{u_4, u_7\}$ |
| $R_{23} \quad (((((a, e), d), b), c)$ | $\{u_{12}\} > \{u_9, u_{13}\}$ $\{u_{10}\} > \{u_4, u_7\} > \{u_2, u_3, u_5, u_6, u_{14}, u_{15}\}$ $\{u_{11}\} > \{u_1, u_8\}$ |
| $R_{24} \quad (((((a, e), d), c), b)$ | $\{u_{12}\} > \{u_9, u_{13}\}$ $\{u_{11}\} > \{u_1, u_8\} > \{u_2, u_3, u_5, u_6, u_{14}, u_{15}\}$ $\{u_{10}\} > \{u_4, u_7\}$ |
| $R_{25} \quad (((((b, c), a), d), e)$ | $\{u_{13}\} > \{u_1, u_4\}$ $\{u_{12}\} > \{u_2, u_5\} > \{u_7, u_8, u_{10}, u_{11}, u_{14}, u_{15}\}$ $\{u_9\} > \{u_3, u_6\}$ |
| $R_{26} \quad (((((b, c), a), e), d)$ | $\{u_{13}\} > \{u_1, u_4\}$ $\{u_9\} > \{u_3, u_6\} > \{u_7, u_8, u_{10}, u_{11}, u_{14}, u_{15}\}$ $\{u_{12}\} > \{u_2, u_5\}$ |
| $R_{27} \quad (((((b, c), d), a), e)$ | $\{u_{12}\} > \{u_{10}, u_{11}\}$ $\{u_{13}\} > \{u_{14}, u_{15}\} > \{u_1, u_2, u_4, u_5, u_7, u_8\}$ $\{u_9\} > \{u_3, u_6\}$ |
| $R_{28} \quad (((((b, c), d), e), a)$ | $\{u_{12}\} > \{u_{10}, u_{11}\}$ $\{u_9\} > \{u_3, u_6\} > \{u_1, u_2, u_4, u_5, u_7, u_8\}$ $\{u_{13}\} > \{u_{14}, u_{15}\}$ |
| $R_{29} \quad (((((b, c), e), a), d)$ | $\{u_9\} > \{u_7, u_8\}$ $\{u_{13}\} > \{u_{14}, u_{15}\} > \{u_1, u_3, u_4, u_6, u_{10}, u_{11}\}$ $\{u_{12}\} > \{u_2, u_5\}$ |
| $R_{30} \quad (((((b, c), e), d), a)$ | $\{u_9\} > \{u_7, u_8\}$ $\{u_{12}\} > \{u_2, u_5\} > \{u_1, u_3, u_4, u_6, u_{10}, u_{11}\}$ $\{u_{13}\} > \{u_7, u_8\}$ |

Table 3.6: Equivalence classes for caterpillar 5-taxon [rooted](#) species trees (continued).

| Tree | Equivalence Classes |
|-------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------|
| $R_{31} \quad (((b, d), a), c), e)$ | $\{u_{14}\} > \{u_2, u_7\}$ $\{u_{11}\} > \{u_1, u_8\} > \{u_4, u_5, u_{10}, u_{12}, u_{13}, u_{15}\}$ $\{u_6\} > \{u_3, u_9\}$ |
| $R_{32} \quad (((b, d), a), e), c)$ | $\{u_{14}\} > \{u_2, u_7\}$ $\{u_6\} > \{u_3, u_9\} > \{u_4, u_5, u_{10}, u_{12}, u_{13}, u_{15}\}$ $\{u_{11}\} > \{u_1, u_8\}$ |
| $R_{33} \quad (((b, d), c), a), e)$ | $\{u_{11}\} > \{u_{10}, u_{12}\}$ $\{u_{14}\} > \{u_{13}, u_{15}\} > \{u_1, u_2, u_4, u_5, u_7, u_8\}$ $\{u_6\} > \{u_3, u_9\}$ |
| $R_{34} \quad (((b, d), c), e), a)$ | $\{u_{11}\} > \{u_{10}, u_{12}\}$ $\{u_6\} > \{u_3, u_9\} > \{u_1, u_2, u_4, u_5, u_7, u_8\}$ $\{u_{14}\} > \{u_{13}, u_{15}\}$ |
| $R_{35} \quad (((b, d), e), a), c)$ | $\{u_6\} > \{u_4, u_5\}$ $\{u_{14}\} > \{u_{13}, u_{15}\} > \{u_2, u_3, u_7, u_9, u_{10}, u_{12}\}$ $\{u_{11}\} > \{u_1, u_8\}$ |
| $R_{36} \quad (((b, d), e), c), a)$ | $\{u_6\} > \{u_4, u_5\}$ $\{u_{11}\} > \{u_1, u_8\} > \{u_2, u_3, u_7, u_9, u_{10}, u_{12}\}$ $\{u_{14}\} > \{u_{13}, u_{15}\}$ |
| $R_{37} \quad (((b, e), a), c), d)$ | $\{u_{15}\} > \{u_3, u_{10}\}$ $\{u_8\} > \{u_1, u_{11}\} > \{u_4, u_6, u_7, u_9, u_{13}, u_{14}\}$ $\{u_5\} > \{u_2, u_{12}\}$ |
| $R_{38} \quad (((b, e), a), d), c)$ | $\{u_{15}\} > \{u_3, u_{10}\}$ $\{u_5\} > \{u_2, u_{12}\} > \{u_4, u_6, u_7, u_9, u_{13}, u_{14}\}$ $\{u_8\} > \{u_1, u_{11}\}$ |
| $R_{39} \quad (((b, e), c), a), d)$ | $\{u_8\} > \{u_7, u_9\}$ $\{u_{15}\} > \{u_{13}, u_{14}\} > \{u_1, u_3, u_4, u_6, u_{10}, u_{11}\}$ $\{u_5\} > \{u_2, u_{12}\}$ |
| $R_{40} \quad (((b, e), c), d), a)$ | $\{u_8\} > \{u_7, u_9\}$ $\{u_5\} > \{u_2, u_{12}\} > \{u_1, u_3, u_4, u_6, u_{10}, u_{11}\}$ $\{u_{15}\} > \{u_{13}, u_{14}\}$ |
| $R_{41} \quad (((b, e), d), a), c)$ | $\{u_5\} > \{u_4, u_6\}$ $\{u_{15}\} > \{u_{13}, u_{14}\} > \{u_2, u_3, u_7, u_9, u_{10}, u_{12}\}$ $\{u_8\} > \{u_1, u_{11}\}$ |
| $R_{42} \quad (((b, e), d), c), a)$ | $\{u_5\} > \{u_4, u_6\}$ $\{u_8\} > \{u_1, u_{11}\} > \{u_2, u_3, u_7, u_9, u_{10}, u_{12}\}$ $\{u_{15}\} > \{u_{13}, u_{14}\}$ |
| $R_{43} \quad (((c, d), a), b), e)$ | $\{u_{15}\} > \{u_5, u_8\}$ $\{u_{10}\} > \{u_4, u_7\} > \{u_1, u_2, u_{11}, u_{12}, u_{13}, u_{14}\}$ $\{u_3\} > \{u_6, u_9\}$ |
| $R_{44} \quad (((c, d), a), e), b)$ | $\{u_{15}\} > \{u_5, u_8\}$ $\{u_3\} > \{u_6, u_9\} > \{u_1, u_2, u_{11}, u_{12}, u_{13}, u_{14}\}$ $\{u_{10}\} > \{u_4, u_7\}$ |
| $R_{45} \quad (((c, d), b), a), e)$ | $\{u_{10}\} > \{u_{11}, u_{12}\}$ $\{u_{15}\} > \{u_{13}, u_{14}\} > \{u_1, u_2, u_4, u_5, u_7, u_8\}$ $\{u_3\} > \{u_6, u_9\}$ |
| $R_{46} \quad (((c, d), b), e), a)$ | $\{u_{10}\} > \{u_{11}, u_{12}\}$ $\{u_3\} > \{u_6, u_9\} > \{u_1, u_2, u_4, u_5, u_7, u_8\}$ $\{u_{15}\} > \{u_{13}, u_{14}\}$ |

Table 3.7: Equivalence classes for caterpillar 5-taxon [rooted](#) species trees (continued).

| Tree | Equivalence Classes |
|---------------------------------|---------------------------------------------------------------------------------------------------------------------------------------|
| $R_{47} (((((c, d), e), a), b)$ | $\{u_3\} > \{u_1, u_2\}$ $\{u_{15}\} > \{u_{13}, u_{14}\} > \{u_5, u_6, u_8, u_9, u_{11}, u_{12}\}$ $\{u_{10}\} > \{u_4, u_7\}$ |
| $R_{48} (((((c, d), e), b), a)$ | $\{u_3\} > \{u_1, u_2\}$ $\{u_{10}\} > \{u_4, u_7\} > \{u_5, u_6, u_8, u_9, u_{11}, u_{12}\}$ $\{u_{15}\} > \{u_{13}, u_{14}\}$ |
| $R_{49} (((((c, e), a), b), d)$ | $\{u_{14}\} > \{u_6, u_{11}\}$ $\{u_7\} > \{u_4, u_{10}\} > \{u_1, u_3, u_8, u_9, u_{13}, u_{15}\}$ $\{u_2\} > \{u_5, u_{12}\}$ |
| $R_{50} (((((c, e), a), d), b)$ | $\{u_{14}\} > \{u_6, u_{11}\}$ $\{u_2\} > \{u_5, u_{12}\} > \{u_1, u_3, u_8, u_9, u_{13}, u_{15}\}$ $\{u_7\} > \{u_4, u_{10}\}$ |
| $R_{51} (((((c, e), b), a), d)$ | $\{u_7\} > \{u_8, u_9\}$ $\{u_{14}\} > \{u_{13}, u_{15}\} > \{u_1, u_3, u_4, u_6, u_{10}, u_{11}\}$ $\{u_2\} > \{u_5, u_{12}\}$ |
| $R_{52} (((((c, e), b), d), a)$ | $\{u_7\} > \{u_8, u_9\}$ $\{u_2\} > \{u_5, u_{12}\} > \{u_1, u_3, u_4, u_6, u_{10}, u_{11}\}$ $\{u_{14}\} > \{u_{13}, u_{15}\}$ |
| $R_{53} (((((c, e), d), a), b)$ | $\{u_2\} > \{u_1, u_3\}$ $\{u_{14}\} > \{u_{13}, u_{15}\} > \{u_5, u_6, u_8, u_9, u_{11}, u_{12}\}$ $\{u_7\} > \{u_4, u_{10}\}$ |
| $R_{54} (((((c, e), d), b), a)$ | $\{u_2\} > \{u_1, u_3\}$ $\{u_7\} > \{u_4, u_{10}\} > \{u_5, u_6, u_8, u_9, u_{11}, u_{12}\}$ $\{u_{14}\} > \{u_{13}, u_{15}\}$ |
| $R_{55} (((((d, e), a), b), c)$ | $\{u_{13}\} > \{u_9, u_{12}\}$ $\{u_4\} > \{u_7, u_{10}\} > \{u_2, u_3, u_5, u_6, u_{14}, u_{15}\}$ $\{u_1\} > \{u_8, u_{11}\}$ |
| $R_{56} (((((d, e), a), c), b)$ | $\{u_{13}\} > \{u_9, u_{12}\}$ $\{u_1\} > \{u_8, u_{11}\} > \{u_2, u_3, u_5, u_6, u_{14}, u_{15}\}$ $\{u_4\} > \{u_7, u_{10}\}$ |
| $R_{57} (((((d, e), b), a), c)$ | $\{u_4\} > \{u_5, u_6\}$ $\{u_{13}\} > \{u_{14}, u_{15}\} > \{u_2, u_3, u_7, u_9, u_{10}, u_{12}\}$ $\{u_1\} > \{u_8, u_{11}\}$ |
| $R_{58} (((((d, e), b), c), a)$ | $\{u_4\} > \{u_5, u_6\}$ $\{u_1\} > \{u_8, u_{11}\} > \{u_2, u_3, u_7, u_9, u_{10}, u_{12}\}$ $\{u_{13}\} > \{u_{14}, u_{15}\}$ |
| $R_{59} (((((d, e), c), a), b)$ | $\{u_1\} > \{u_2, u_3\}$ $\{u_{13}\} > \{u_{14}, u_{15}\} > \{u_5, u_6, u_8, u_9, u_{11}, u_{12}\}$ $\{u_4\} > \{u_7, u_{10}\}$ |
| $R_{60} (((((d, e), c), b), a)$ | $\{u_1\} > \{u_2, u_3\}$ $\{u_4\} > \{u_7, u_{10}\} > \{u_5, u_6, u_8, u_9, u_{11}, u_{12}\}$ $\{u_{13}\} > \{u_{14}, u_{15}\}$ |

3.8.2 Pseudo-caterpillar Trees

Table 3.8: Equivalence classes for pseudo-caterpillar 5-taxon **rooted** species trees. The u_i s are named according to the **unrooted** trees in Table 5 in [81]. A script for generating these classes is available in the Quintet Rooting software repository in Github.

| Tree | Equivalence Classes |
|--------------------------------|---------------------------------------------------------------------------------------------------------------------|
| $R_{61} (((a, b), (c, d)), e)$ | $\{u_3\} > \{u_1, u_2\}, \{u_{10}, u_{15}\}, \{u_6, u_9\} > \{u_4, u_5, u_7, u_8, u_{11}, u_{12}, u_{13}, u_{14}\}$ |
| $R_{62} (((a, c), (b, d)), e)$ | $\{u_6\} > \{u_4, u_5\}, \{u_{11}, u_{14}\}, \{u_3, u_9\} > \{u_1, u_2, u_7, u_8, u_{10}, u_{12}, u_{13}, u_{15}\}$ |
| $R_{63} (((a, d), (b, c)), e)$ | $\{u_9\} > \{u_7, u_8\}, \{u_{12}, u_{13}\}, \{u_3, u_6\} > \{u_1, u_2, u_4, u_5, u_{10}, u_{11}, u_{14}, u_{15}\}$ |
| $R_{64} (((a, b), (c, e)), d)$ | $\{u_2\} > \{u_1, u_3\}, \{u_7, u_{14}\}, \{u_5, u_{12}\} > \{u_4, u_6, u_8, u_9, u_{10}, u_{11}, u_{13}, u_{15}\}$ |
| $R_{65} (((a, c), (b, e)), d)$ | $\{u_5\} > \{u_4, u_6\}, \{u_8, u_{15}\}, \{u_2, u_{12}\} > \{u_1, u_3, u_7, u_9, u_{10}, u_{11}, u_{13}, u_{14}\}$ |
| $R_{66} (((a, e), (b, c)), d)$ | $\{u_{12}\} > \{u_{10}, u_{11}\}, \{u_9, u_{13}\}, \{u_2, u_5\} > \{u_1, u_3, u_4, u_6, u_7, u_8, u_{14}, u_{15}\}$ |
| $R_{67} (((a, b), (d, e)), c)$ | $\{u_1\} > \{u_2, u_3\}, \{u_4, u_{13}\}, \{u_8, u_{11}\} > \{u_5, u_6, u_7, u_9, u_{10}, u_{12}, u_{14}, u_{15}\}$ |
| $R_{68} (((a, d), (b, e)), c)$ | $\{u_8\} > \{u_7, u_9\}, \{u_5, u_{15}\}, \{u_1, u_{11}\} > \{u_2, u_3, u_4, u_6, u_{10}, u_{12}, u_{13}, u_{14}\}$ |
| $R_{69} (((a, e), (b, d)), c)$ | $\{u_{11}\} > \{u_{10}, u_{12}\}, \{u_6, u_{14}\}, \{u_1, u_8\} > \{u_2, u_3, u_4, u_5, u_7, u_9, u_{13}, u_{15}\}$ |
| $R_{70} (((a, c), (d, e)), b)$ | $\{u_4\} > \{u_5, u_6\}, \{u_1, u_{13}\}, \{u_7, u_{10}\} > \{u_2, u_3, u_8, u_9, u_{11}, u_{12}, u_{14}, u_{15}\}$ |
| $R_{71} (((a, d), (c, e)), b)$ | $\{u_7\} > \{u_8, u_9\}, \{u_2, u_{14}\}, \{u_4, u_{10}\} > \{u_1, u_3, u_5, u_6, u_{11}, u_{12}, u_{13}, u_{15}\}$ |
| $R_{72} (((a, e), (c, d)), b)$ | $\{u_{10}\} > \{u_{11}, u_{12}\}, \{u_3, u_{15}\}, \{u_4, u_7\} > \{u_1, u_2, u_5, u_6, u_8, u_9, u_{13}, u_{14}\}$ |
| $R_{73} (((b, c), (d, e)), a)$ | $\{u_{13}\} > \{u_9, u_{12}\}, \{u_1, u_4\}, \{u_{14}, u_{15}\} > \{u_2, u_3, u_5, u_6, u_7, u_8, u_{10}, u_{11}\}$ |
| $R_{74} (((b, d), (c, e)), a)$ | $\{u_{14}\} > \{u_6, u_{11}\}, \{u_2, u_7\}, \{u_{13}, u_{15}\} > \{u_1, u_3, u_4, u_5, u_8, u_9, u_{10}, u_{12}\}$ |
| $R_{75} (((b, e), (c, d)), a)$ | $\{u_{15}\} > \{u_5, u_8\}, \{u_3, u_{10}\}, \{u_{13}, u_{14}\} > \{u_1, u_2, u_4, u_6, u_7, u_9, u_{11}, u_{12}\}$ |

3.8.3 Balanced Trees

Table 3.9: Equivalence classes for balanced 5-taxon **rooted** species trees. The u_i s are named according to the **unrooted** trees in Table 5 in [81]. A script for generating these classes is available in the Quintet Rooting software repository in Github.

| Tree | Equivalence Classes |
|---------------------------------|----------------------------------------------------------------------------------------------------------------------|
| R_{76} (((a, b), c), (d, e)) | $\{u_1\} > \{u_2, u_3\}, \{u_4, u_{13}\} > \{u_5, u_6, u_9, u_{12}\} > \{u_7, u_8, u_{10}, u_{11}, u_{14}, u_{15}\}$ |
| R_{77} (((a, c), b), (d, e)) | $\{u_4\} > \{u_1, u_{13}\}, \{u_5, u_6\} > \{u_2, u_3, u_9, u_{12}\} > \{u_7, u_8, u_{10}, u_{11}, u_{14}, u_{15}\}$ |
| R_{78} (((b, c), a), (d, e)) | $\{u_{13}\} > \{u_1, u_4\}, \{u_9, u_{12}\} > \{u_2, u_3, u_5, u_6\} > \{u_7, u_8, u_{10}, u_{11}, u_{14}, u_{15}\}$ |
| R_{79} (((a, b), d), (c, e)) | $\{u_2\} > \{u_1, u_3\}, \{u_7, u_{14}\} > \{u_6, u_8, u_9, u_{11}\} > \{u_4, u_5, u_{10}, u_{12}, u_{13}, u_{15}\}$ |
| R_{80} (((a, d), b), (c, e)) | $\{u_7\} > \{u_2, u_{14}\}, \{u_8, u_9\} > \{u_1, u_3, u_6, u_{11}\} > \{u_4, u_5, u_{10}, u_{12}, u_{13}, u_{15}\}$ |
| R_{81} (((b, d), a), (c, e)) | $\{u_{14}\} > \{u_2, u_7\}, \{u_6, u_{11}\} > \{u_1, u_3, u_8, u_9\} > \{u_4, u_5, u_{10}, u_{12}, u_{13}, u_{15}\}$ |
| R_{82} (((a, c), d), (b, e)) | $\{u_5\} > \{u_4, u_6\}, \{u_8, u_{15}\} > \{u_3, u_7, u_9, u_{10}\} > \{u_1, u_2, u_{11}, u_{12}, u_{13}, u_{14}\}$ |
| R_{83} (((a, d), c), (b, e)) | $\{u_8\} > \{u_5, u_{15}\}, \{u_7, u_9\} > \{u_3, u_4, u_6, u_{10}\} > \{u_1, u_2, u_{11}, u_{12}, u_{13}, u_{14}\}$ |
| R_{84} (((c, d), a), (b, e)) | $\{u_{15}\} > \{u_3, u_{10}\}, \{u_5, u_8\} > \{u_4, u_6, u_7, u_9\} > \{u_1, u_2, u_{11}, u_{12}, u_{13}, u_{14}\}$ |
| R_{85} (((b, c), d), (a, e)) | $\{u_{12}\} > \{u_9, u_{13}\}, \{u_{10}, u_{11}\} > \{u_3, u_6, u_{14}, u_{15}\} > \{u_1, u_2, u_4, u_5, u_7, u_8\}$ |
| R_{86} (((b, d), c), (a, e)) | $\{u_{11}\} > \{u_6, u_{14}\}, \{u_{10}, u_{12}\} > \{u_3, u_9, u_{13}, u_{15}\} > \{u_1, u_2, u_4, u_5, u_7, u_8\}$ |
| R_{87} (((c, d), b), (a, e)) | $\{u_{10}\} > \{u_3, u_{15}\}, \{u_{11}, u_{12}\} > \{u_6, u_9, u_{13}, u_{14}\} > \{u_1, u_2, u_4, u_5, u_7, u_8\}$ |
| R_{88} (((a, b), e), (c, d)) | $\{u_3\} > \{u_1, u_2\}, \{u_{10}, u_{15}\} > \{u_5, u_8, u_{11}, u_{12}\} > \{u_4, u_6, u_7, u_9, u_{13}, u_{14}\}$ |
| R_{89} (((a, e), b), (c, d)) | $\{u_{10}\} > \{u_3, u_{15}\}, \{u_{11}, u_{12}\} > \{u_1, u_2, u_5, u_8\} > \{u_4, u_6, u_7, u_9, u_{13}, u_{14}\}$ |
| R_{90} (((b, e), a), (c, d)) | $\{u_{15}\} > \{u_3, u_{10}\}, \{u_5, u_8\} > \{u_1, u_2, u_{11}, u_{12}\} > \{u_4, u_6, u_7, u_9, u_{13}, u_{14}\}$ |
| R_{91} (((a, c), e), (b, d)) | $\{u_6\} > \{u_4, u_5\}, \{u_{11}, u_{14}\} > \{u_2, u_7, u_{10}, u_{12}\} > \{u_1, u_3, u_8, u_9, u_{13}, u_{15}\}$ |
| R_{92} (((a, e), c), (b, d)) | $\{u_{11}\} > \{u_6, u_{14}\}, \{u_{10}, u_{12}\} > \{u_2, u_4, u_5, u_7\} > \{u_1, u_3, u_8, u_9, u_{13}, u_{15}\}$ |
| R_{93} (((c, e), a), (b, d)) | $\{u_{14}\} > \{u_2, u_7\}, \{u_6, u_{11}\} > \{u_4, u_5, u_{10}, u_{12}\} > \{u_1, u_3, u_8, u_9, u_{13}, u_{15}\}$ |
| R_{94} (((b, c), e), (a, d)) | $\{u_9\} > \{u_7, u_8\}, \{u_{12}, u_{13}\} > \{u_2, u_5, u_{14}, u_{15}\} > \{u_1, u_3, u_4, u_6, u_{10}, u_{11}\}$ |
| R_{95} (((b, e), c), (a, d)) | $\{u_8\} > \{u_5, u_{15}\}, \{u_7, u_9\} > \{u_2, u_{12}, u_3, u_{14}\} > \{u_1, u_3, u_4, u_6, u_{10}, u_{11}\}$ |
| R_{96} (((c, e), b), (a, d)) | $\{u_7\} > \{u_2, u_{14}\}, \{u_8, u_9\} > \{u_5, u_{12}, u_3, u_{15}\} > \{u_1, u_3, u_4, u_6, u_{10}, u_{11}\}$ |
| R_{97} (((a, d), e), (b, c)) | $\{u_9\} > \{u_7, u_8\}, \{u_{12}, u_{13}\} > \{u_1, u_4, u_{10}, u_{11}\} > \{u_2, u_3, u_5, u_6, u_{14}, u_{15}\}$ |
| R_{98} (((a, e), d), (b, c)) | $\{u_{12}\} > \{u_9, u_{13}\}, \{u_{10}, u_{11}\} > \{u_1, u_4, u_7, u_8\} > \{u_2, u_3, u_5, u_6, u_{14}, u_{15}\}$ |
| R_{99} (((d, e), a), (b, c)) | $\{u_{13}\} > \{u_1, u_4\}, \{u_9, u_{12}\} > \{u_7, u_8, u_{10}, u_{11}\} > \{u_2, u_3, u_5, u_6, u_{14}, u_{15}\}$ |
| R_{100} (((b, d), e), (a, c)) | $\{u_6\} > \{u_4, u_5\}, \{u_{11}, u_{14}\} > \{u_1, u_8, u_{13}, u_{15}\} > \{u_2, u_3, u_7, u_9, u_{10}, u_{12}\}$ |
| R_{101} (((b, e), d), (a, c)) | $\{u_5\} > \{u_4, u_6\}, \{u_8, u_{15}\} > \{u_1, u_{11}, u_3, u_{14}\} > \{u_2, u_3, u_7, u_9, u_{10}, u_{12}\}$ |
| R_{102} (((d, e), b), (a, c)) | $\{u_4\} > \{u_1, u_{13}\}, \{u_5, u_6\} > \{u_8, u_{11}, u_{14}, u_{15}\} > \{u_2, u_3, u_7, u_9, u_{10}, u_{12}\}$ |
| R_{103} (((c, d), e), (a, b)) | $\{u_3\} > \{u_1, u_2\}, \{u_{10}, u_{15}\} > \{u_4, u_7, u_{13}, u_{14}\} > \{u_5, u_6, u_8, u_9, u_{11}, u_{12}\}$ |
| R_{104} (((c, e), d), (a, b)) | $\{u_2\} > \{u_1, u_3\}, \{u_7, u_{14}\} > \{u_4, u_{10}, u_{13}, u_{15}\} > \{u_5, u_6, u_8, u_9, u_{11}, u_{12}\}$ |
| R_{105} (((d, e), c), (a, b)) | $\{u_1\} > \{u_2, u_3\}, \{u_4, u_{13}\} > \{u_7, u_{10}, u_{14}, u_{15}\} > \{u_5, u_6, u_8, u_9, u_{11}, u_{12}\}$ |

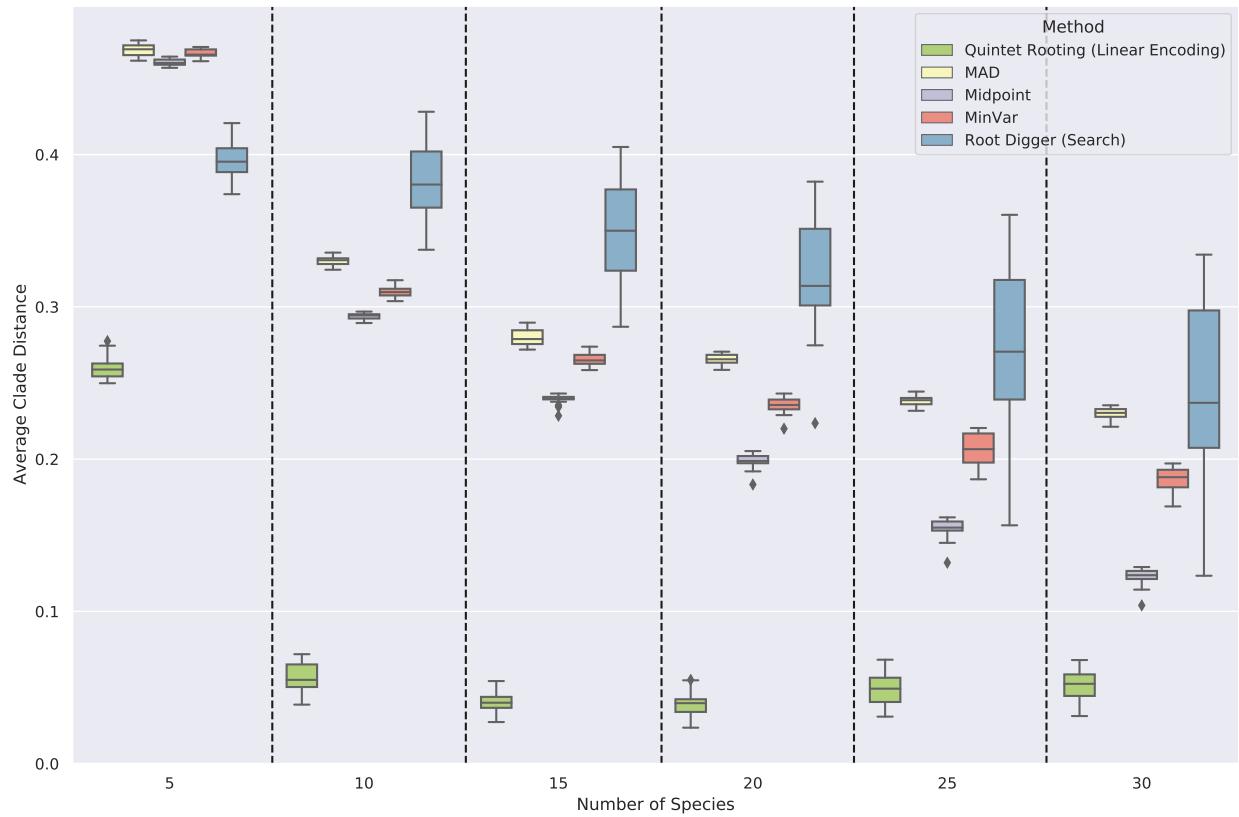


Figure 3.10: Average clade distance on subsets of the avian simulated datasets with 5 to 30 leaves for each rooting method, given the true species tree [topology](#) and true gene trees. The branch lengths on the model tree are estimated using RAxML with the concatenated gene multiple sequence alignments. The results are averaged over 200 samples for each value of k . The number of genes is 1000 and the error bars are shown across 20 replicates.

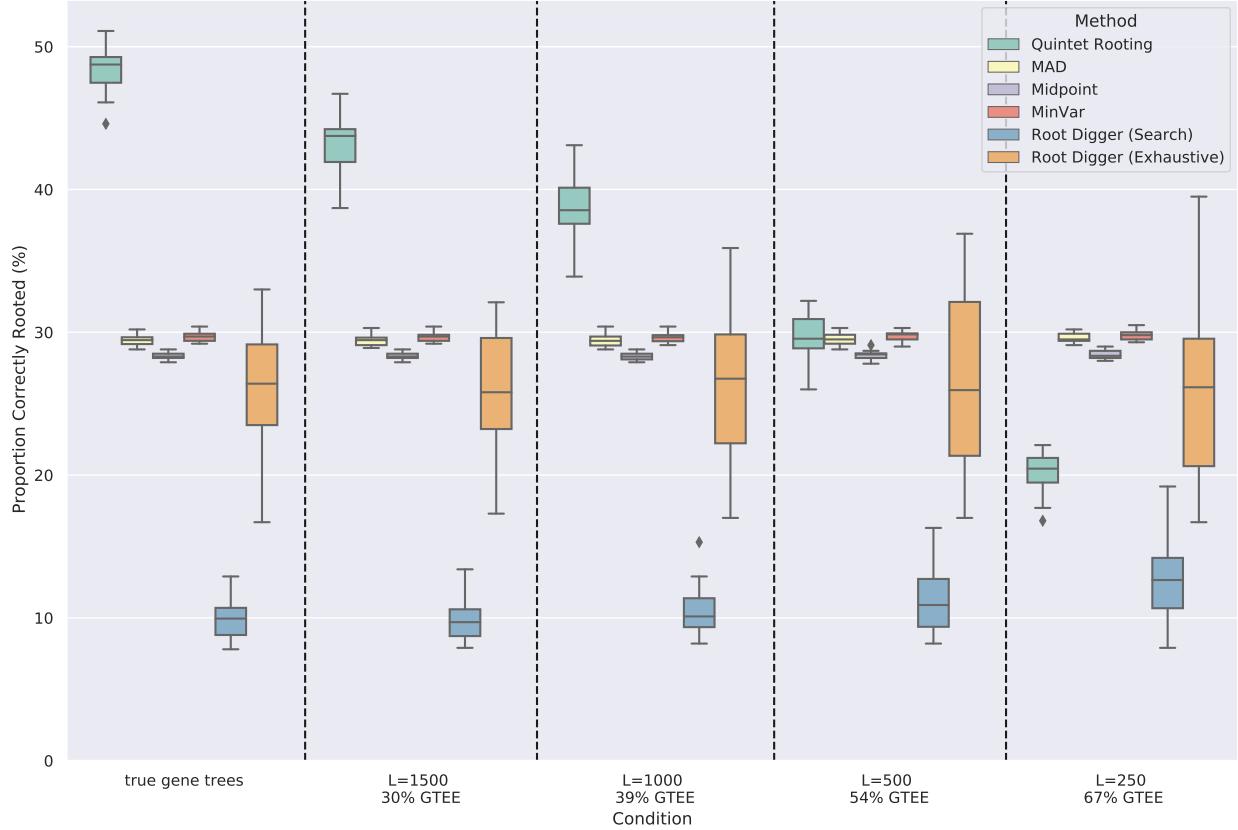


Figure 3.11: Proportion of the 5-leaf subtrees of the avian model tree correctly [rooted](#) by each rooting method. The results are averaged over 1000 sample 5-taxon trees. The number of genes is 1000 and the error bars are shown across 20 replicates. All methods root the true species tree [topology](#) and branch lengths are estimated using RAxML on the concatenated gene sequence alignments.

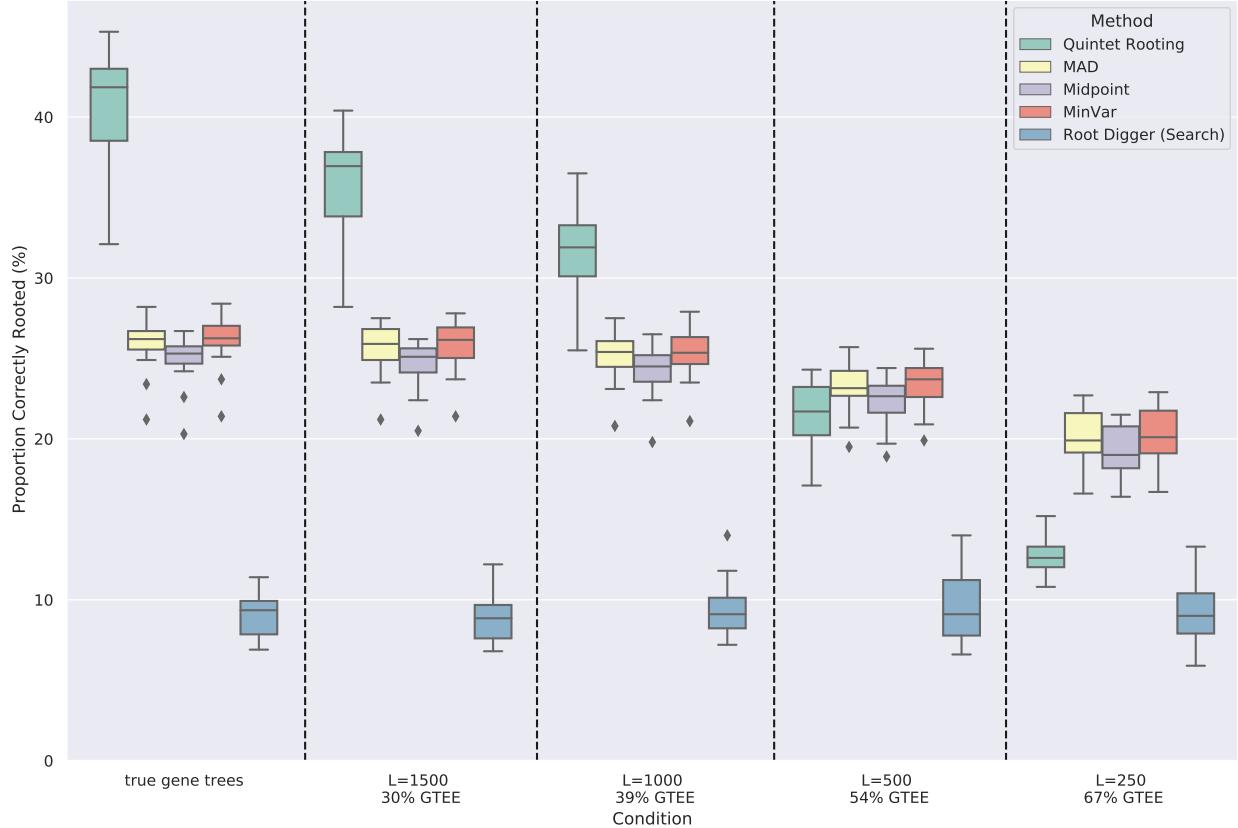


Figure 3.12: Proportion of the trees correctly rooted on 5-leaf avian simulated datasets by each rooting method given an estimated species tree computed by ASTRAL. The branch lengths on the estimated species tree are estimated using RAxML with the concatenated gene multiple sequence alignments. The results are averaged over 1000 sample 5-taxon trees. The number of genes is 1000 and the error bars are shown across 20 replicates.

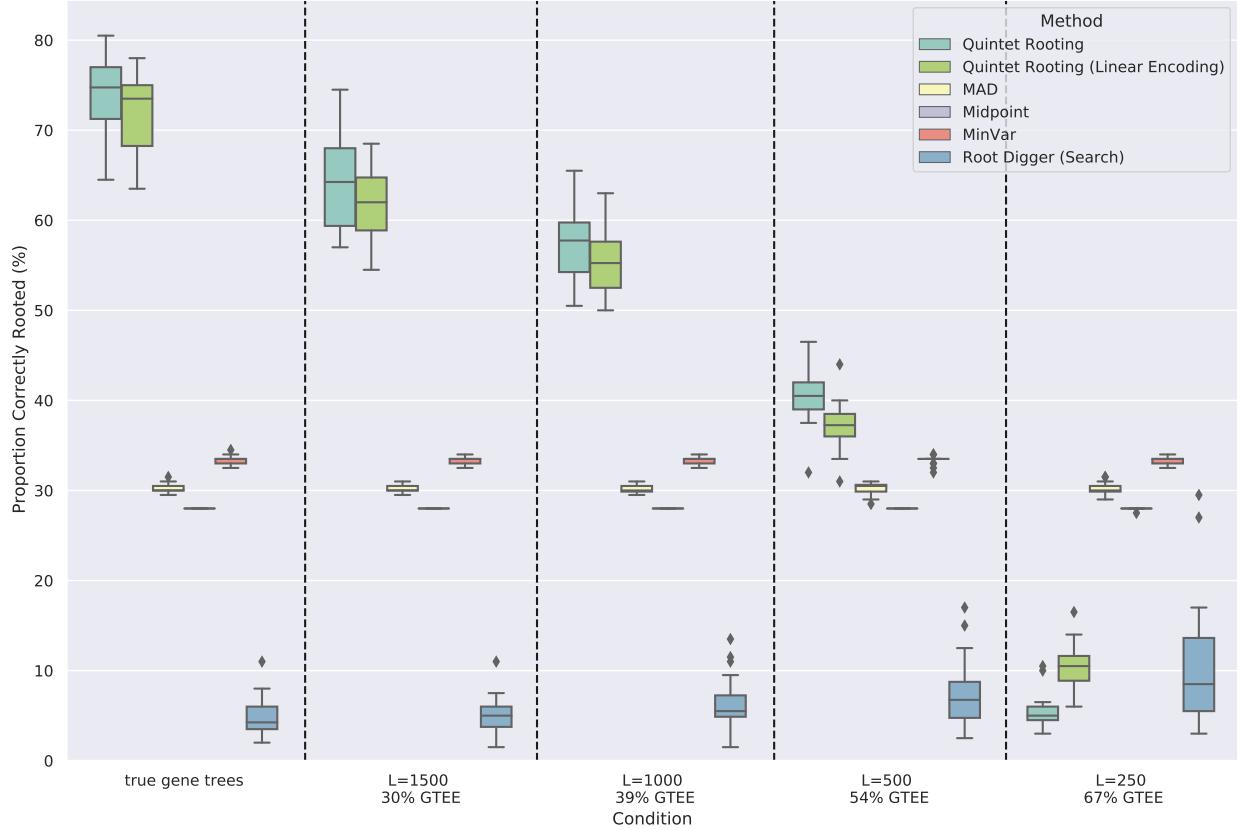


Figure 3.13: Proportion of the trees correctly rooted on 10-leaf avian simulated datasets given the true (model) species tree by each rooting method. The results are averaged over 200 sample 10-species trees. The number of genes is 1000 and the error bars are shown across 20 replicates. The branch lengths are estimated using RAxML on the concatenated gene sequence alignments.

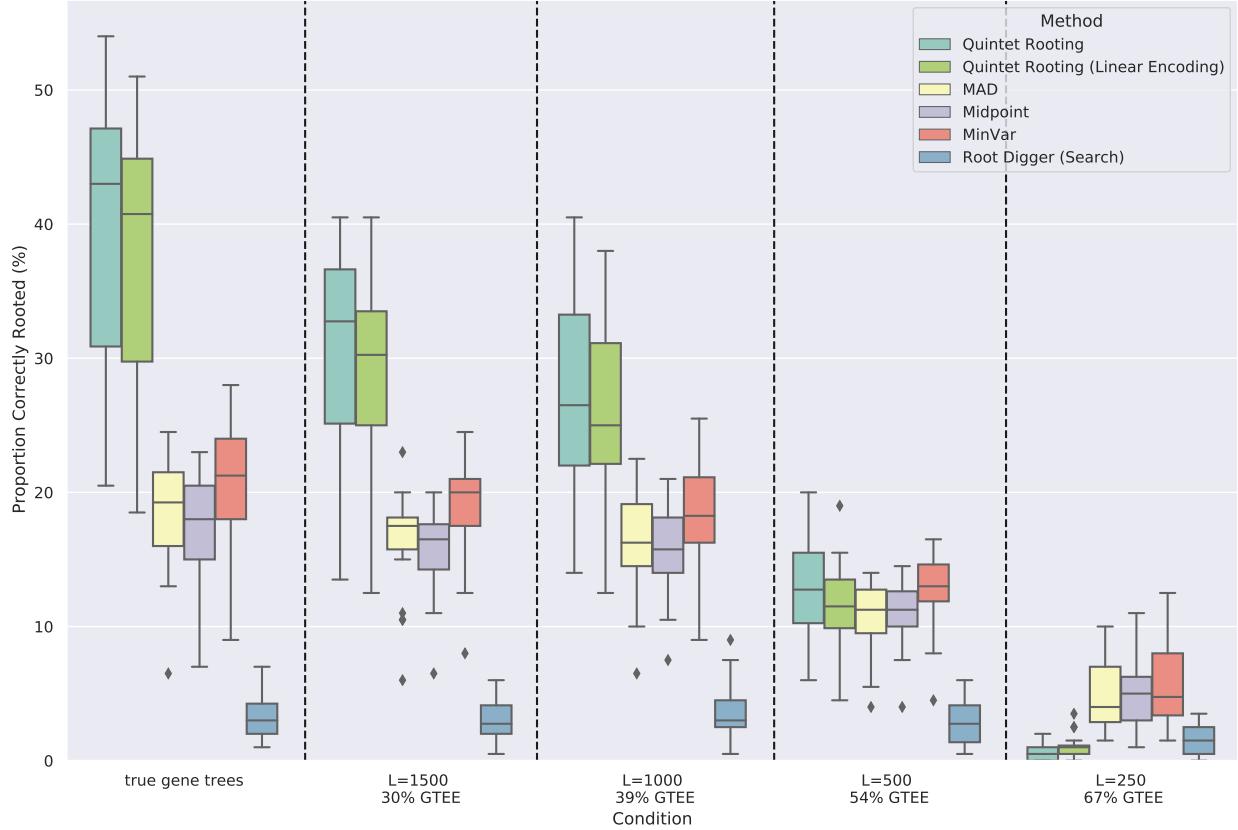


Figure 3.14: Proportion of the trees correctly rooted on 10-leaf avian simulated datasets by each rooting method given an estimated species tree computed by ASTRAL. The branch lengths on the ASTRAL tree are estimated using RAxML with the concatenated gene multiple sequence alignments. The results are averaged over 200 sample 10-species trees. The number of genes is 1000 and the error bars are shown across 20 replicates.

CHAPTER 4: POLYNOMIAL-TIME STATISTICALLY CONSISTENT ROOTING OF SPECIES TREES UNDER THE COALESCENT

This chapter contains material previously published in “Y. Tabatabaee, S. Roch and T. Warnow. (2023). QR-STAR: A polynomial-time [statistically consistent](#) method for rooting species trees under the coalescent. Journal of Computational Biology, Volume 30, Number 11”[255] and “Y. Tabatabaee, S. Roch and T. Warnow. (2023). [statistically consistent](#) rooting of species trees under the multispecies coalescent model. International Conference on Research in Computational Molecular Biology, pages 41-57, Cham: Springer Nature Switzerland”[256]. The QR-STAR software is available in open-source form at <https://github.com/ytabatabaee/Quintet-Rooting>. The datasets and scripts used in this study are available at <https://github.com/ytabatabaee/QR-STAR-paper>.

4.1 INTRODUCTION

Rooted species trees are needed for many biological research problems, including comparative genomics [257, 258] and dating [259]. The availability of genome-wide sequencing data for many species has made it possible to estimate species trees using different loci from across the genome, thus enabling “multi-locus” species tree estimation. Typically, rooted species trees are estimated in two steps: first the [unrooted topology](#) of the species tree is inferred using a multi-locus [species tree](#) estimation method, and then that [unrooted species tree](#) is rooted. Alternatively, [rooted](#) gene trees can be inferred and then combined into a [rooted](#) species tree, using methods such as MP-EST [46], STAR [93] and GLASS [165]. However, the estimation of [rooted](#) gene trees is itself challenging, making this approach less reliable than methods that operate in the two step procedure where the [unrooted](#) species tree is estimated first and then [rooted](#) [168, 260].

The problem of estimating an [unrooted species tree](#) has been actively investigated over the last several decades. The classical approach is “concatenation”, where the alignments for the different loci are concatenated into one large “super-alignment”, which is then given to a tree estimation method, such as RAxML [25]. Yet, evolutionary processes, such as [Incomplete lineage sorting \(ILS\)](#) or [Gene duplication and loss \(GDL\)](#), can result in different genomic regions (referred to as “loci”) having different evolutionary histories, so that gene trees and species trees can have different topologies [15]. Moreover, when [ILS](#) is high, then standard concatenation analyses can have poor accuracy [30, 32], and may even be [statistically inconsistent](#) [28, 29].

Therefore, in order to estimate highly accurate species trees in the presence of [ILS](#) or

GDL, new methods have been developed that take the source of heterogeneity into consideration [38, 42, 48]. **species tree** estimation in the presence of **ILS**, as modeled by the **Multi-Species Coalescent (MSC)** model [98], is the most well-studied, and many methods have been developed for this problem; see [11] for a survey.

Given an estimate of the **unrooted** species tree, various methods can be used to infer the root location. Perhaps the most commonly used approach is the use of one or more **outgroup** species (e.g., the addition of a lizard within a collection of bird species), which allows the **unrooted** tree on this enlarged set of species to be **rooted** on the edge leading to the **outgroup** [235]. While this approach is natural, there are many challenges in selecting an appropriate **outgroup** species: if the **outgroup** is too distantly related to the other species, then it may be attached fairly randomly to the tree containing the remaining species, and if it is too closely related, it may even be an ingroup taxon rather than an **outgroup** [142, 238, 261, 262]. It is also possible to use estimated branch lengths on the **species tree** to find the root based on specific optimization criteria, often using **molecular clock** analysis [263]; however, these approaches may only be highly accurate when evolutionary rates are close to following the **strict molecular clock** (which assumes that all **sites** along the genome evolve under a constant rate) [78, 79, 220]. There are also recent developments that seek to find the root based on non-reversible models of DNA substitution [243, 264]. However, none of the methods mentioned so far consider biological processes that cause discord between species trees and gene trees, and are mainly used and evaluated for rooting gene trees [265].

Recently, a few methods have been developed that are specifically designed for rooting species trees under the **MSC**; these include a rooting method by [244] that uses **site pattern** probabilities, and a method that uses approximate Bayesian computation by [266]. The first method assumes a **strict molecular clock** and degrades in accuracy when there is deviation from the clock [244], and the second approach relies on a large number of calculations and may not be scalable [266]. Furthermore, the software for these methods is not publicly available, and their performance compared to other methods is not explored in the literature.

We recently introduced Quintet Rooting (QR) [230], a polynomial-time method for rooting an **unrooted species tree** with at least five leaves given a set of **unrooted** gene trees, which is designed for use when the gene trees differ from the **species tree** due to **ILS**. QR is based on the mathematical theory by Allman, Degnan, and Rhodes [81] that established that the **rooted topology** of every 5-leaf **species tree** is **identifiable** from the distribution of the **unrooted** 5-leaf gene tree topologies; a trivial extension to any number $n \geq 5$ species then follows.

The experimental study in [230] showed that QR had good accuracy on simulated **ILS** datasets in comparison to alternative methods. However, we did not establish whether it

was statistically consistent under the MSC. That is, we did not establish whether QR would return the correct root location with probability converging to 1 as the number of true gene trees in the input increases, when given the true unrooted species tree as input. While there has been much focus on proving statistical consistency for species tree estimation methods and several methods such as ASTRAL [48], SVDQuartets [45] and BUCKY [162] have been proven statistically consistent estimators of the *unrooted species tree* under the MSC, to the best of our knowledge, no prior study has addressed the statistical consistency properties of methods for rooting species trees.

In this chapter, we argue that QR is not guaranteed to be statistically consistent under the MSC. We introduce a variant of QR called QR-STAR that is also polynomial-time and uses much of the same algorithmic structure of QR, but with some important changes that enable us to prove statistical consistency under the MSC. We also analyze the sample complexity for QR-STAR, and provide a variant that achieves polynomial sample complexity. Finally, our simulation study evaluating QR and QR-STAR under a range of model conditions shows that QR-STAR matches or improves on the accuracy of QR, and its error is close to the error of the optimal rooting under many conditions.

The rest of this chapter is organized as follows. We provide background information on QR in Section 4.2, as well as the theory established by Allman, Degnan, and Rhodes [81]. We introduce QR-STAR in Section 4.3. The theoretical results are provided in Section 4.4. In Section 4.5, we report on the results of a simulation study, including the design of QR-STAR and the evaluation of QR-STAR in comparison to QR. We conclude in Section 4.6 with a discussion of future research.

4.2 BACKGROUND

We present the theory from [81] first, which establishes identifiability of the rooted species tree from unrooted quintet trees, and then we describe Quintet Rooting (QR), our earlier method for rooting species trees. Together these form the basis for deriving our new method, QR-STAR, which we present in the next section.

4.2.1 Allman, Degnan, and Rhodes (ADR) Theory

Allman, Degnan, and Rhodes (ADR) [81] established that the unrooted topology of the species tree is identifiable from four-leaf unrooted gene trees under the MSC, a result that is well known and used in several “quartet-based” methods for estimating species trees under the MSC [48, 162, 171]. ADR also proved that the rooted species tree topology is identifiable

from unrooted five-leaf gene tree topologies; this result is much less well known, but was recently used in the development of QR for rooting species trees.

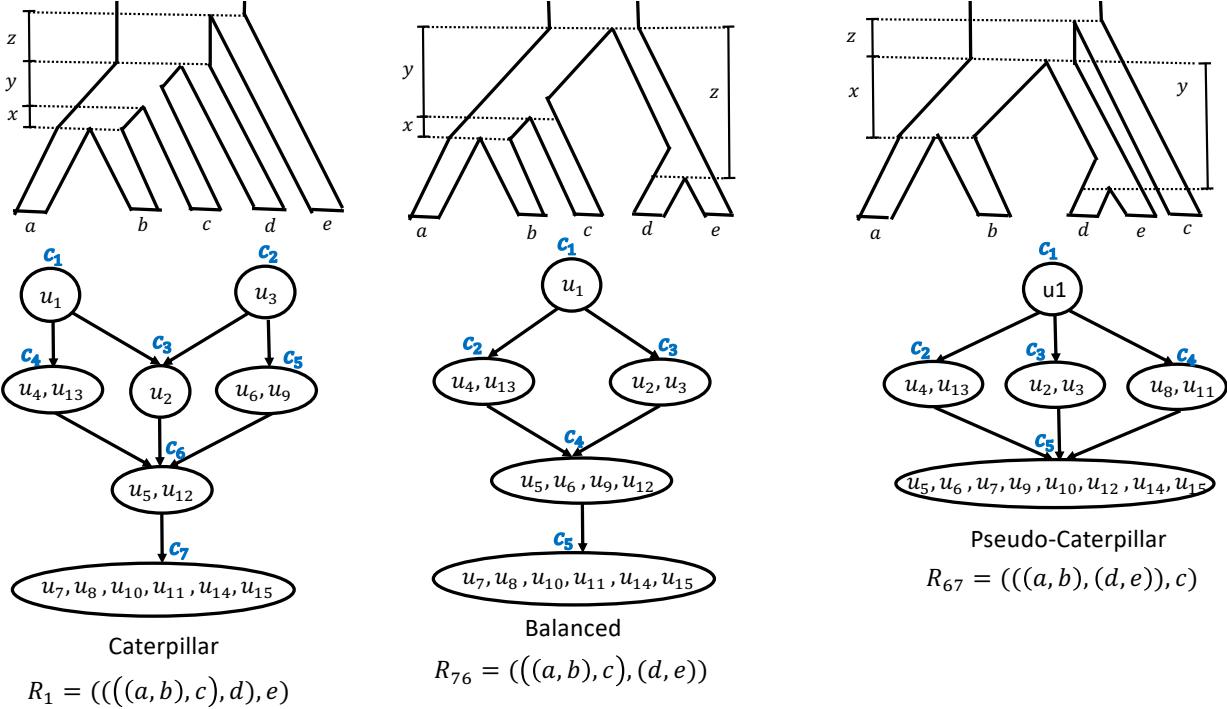


Figure 4.1: **ADR** invariants and inequalities for different **rooted** topological shapes. The invariants (i.e., equalities) and inequalities found by **ADR** define a partial order on the probabilities of **unrooted** 5-taxon gene tree topologies for **rooted** 5-taxon model species trees with different **rooted** shapes (caterpillar, balanced and pseudo-caterpillar). There are 15 **unrooted binary** trees on a given set of 5 leaves. Each of the 105 5-taxon **rooted** species trees define a specific distribution on the probabilities of these **unrooted** trees. The **topology** of the **rooted binary species tree** can be determined from this distribution (i.e., it is identifiable, as established by **ADR**). While the branch lengths of the **rooted species tree** depend on the actual probabilities, the linear invariants and inequalities that hold for these distributions is enough to determine the **rooted topology** of the model species tree.

ADR have described the probability distribution of **unrooted** gene tree topologies under each 5-taxon **MSC** model species tree. On a given set of five taxa, there exist 105 different **rooted binary** trees, labeled with R_1, \dots, R_{105} ¹, that can be categorized into three groups based on their (unlabeled) **rooted** shapes: caterpillar, balanced and pseudo-caterpillar [245]. An example of a tree from each category is shown in Figure 4.1. Each 5-taxon model **species tree** defines a specific probability distribution over the 15 different **unrooted** gene tree topologies on the same leafset, shown with T_1, \dots, T_{15} (Fig. 4.2). Theorem 9 in [81]

¹The labeling of *rooted* and *unrooted* trees in this chapter is consistent with the notations and leaf-labeling used in Tables 4-5 in [81] as well as in [230].

states that this distribution uniquely determines the [rooted](#) tree [topology](#) and its [internal](#) branch lengths for trees with at least five taxa.

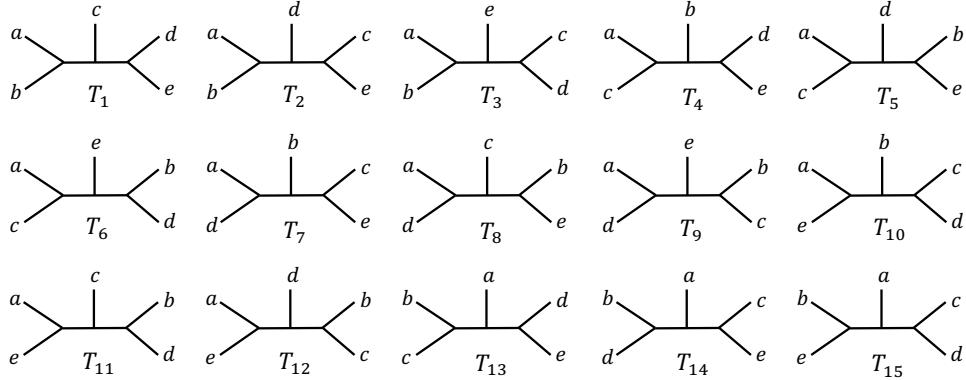


Figure 4.2: Topologies of the 15 [unrooted](#) 5-taxon gene trees labeled according to [81].

To prove this identifiability result, the [ADR](#) theory specifies a set of linear invariants (i.e., equalities) and inequalities that must hold between the probabilities of [unrooted](#) 5-taxon gene trees, for any choice of the parameters of the model species tree. These linear invariants and inequalities define a partial order on the probabilities of 5-taxon [unrooted](#) gene tree topologies. In other words, two gene tree probabilities $u_i = \mathbb{P}(T_i)$ and $u_j = \mathbb{P}(T_j)$ can have one of four possible relationships: $u_i > u_j$, $u_j > u_i$, $u_i = u_j$, or u_i and u_j are not comparable.

Figure 4.1 shows examples of these partial orders, described using [Hasse diagrams](#), for a particular leaf labeling of trees from each [rooted](#) shape. Note that some probabilities are members of the same set (e.g., for R_1 , set c_4 contains both u_4 and u_{13} , indicating that $u_4 = u_{13}$), and so we refer to the sets c_i as equivalence classes on these probabilities. Furthermore, we will denote the set of equivalence classes associated with a 5-taxon [rooted](#) tree R with C_R . As can be seen in Figure 4.1, the number of equivalence classes for caterpillar, balanced and pseudo-caterpillar trees is 7, 5, and 5 respectively. Each directed edge between two equivalence classes in these [Hasse diagrams](#) defines an inequality, so that all gene tree probabilities in class c_a at the source of an edge are greater than all gene tree probabilities in class c_b at the target, and we show this by $c_a > c_b$. The exact values of the [unrooted](#) gene tree probabilities depend on the [internal](#) branch lengths of the model tree, and [ADR](#) provide a set of formulas that relate the model tree parameters to the probability distribution of the [unrooted](#) gene trees in Appendix B of [81], which will be used in our proofs.

4.2.2 Quintet Rooting

The input to QR is an [unrooted species tree](#) T with n leaves and a set \mathcal{G} of k single-copy [unrooted](#) gene trees where the gene trees draw their leaves from the leafset of T , denoted by $\mathcal{L}(T)$. Given this input, QR searches over all possible rootings of T and returns a tree most consistent with the distribution of quintets (i.e., 5-taxon trees) in the input gene trees.

QR approaches this problem by selecting a set Q of quintets of [taxa](#) from $\mathcal{L}(T)$ (called the “quintet sampling” step), and scoring all [rooted](#) versions of T based on their induced trees on these quintets. The subtree $T|_q$, i.e., T restricted to [taxa](#) in quintet set q , can be [rooted](#) on any of its seven edges. In a preprocessing step, QR computes a score for each of these seven different rootings for all trees induced on the quintets in set Q , based on a cost function (described below). This results in $7 \times |Q|$ computations, and therefore the preprocessing step takes $O(k(|Q| + n))$. Next, for every [rooted](#) version of T , QR sums up the costs of all its induced [rooted](#) trees on quintets in Q using the scores computed in the preprocessing step, and returns the rooting with the minimum overall cost. Since T can be [rooted](#) on any of its $2n - 3$ edges, the scoring step takes $O(n + |Q|)$ time.

Thus, QR provides an exact solution to the optimization problem with the following input and output:

- **Input:** An [unrooted](#) tree [topology](#) T , a set of k [unrooted](#) gene tree topologies $\mathcal{G} = \{g_1, g_2, \dots, g_k\}$, a set Q containing quintets of [taxa](#) from leafset $\mathcal{L}(T)$, and a cost function $\text{Cost}(r, \vec{u})$.
- **Output:** Rooted tree R with [topology](#) T such that $\sum_{q \in Q} \text{Cost}(R|_q, \vec{\hat{u}}_q)$ is minimized, where $\vec{\hat{u}}_q$ is the distribution of [unrooted](#) quintet trees in $\mathcal{G}|_q = \{g_1|_q, g_2|_q, \dots, g_k|_q\}$.

Cost Function. The cost function $\text{Cost}(R|_q, \vec{\hat{u}}_q)$ measures the fitness of the [rooted](#) quintet tree $R|_q$ with the distribution of the [unrooted](#) gene trees restricted to q (i.e., $\vec{\hat{u}}_q$), according to the linear invariants and inequalities derived from the [ADR](#) theory. In particular, this cost function is designed to penalize a [rooted](#) tree $R|_q$ if the estimated quintet distribution $\vec{\hat{u}}_q$ violates some of the inequalities or invariants in its partial order. To this end, a penalty term was considered for each invariant and inequality in the partial order of a 5-taxon [rooted](#) tree that is violated in a quintet distribution. The cost function was defined based on a linear combination of these penalty terms, and had the following form, where r is a 5-taxon [rooted](#)

tree and $\vec{\hat{u}}$ is an estimated quintet distribution:

$$\text{Cost}(r, \vec{\hat{u}}) = \underbrace{\sum_{c \in C_r} \frac{1}{|c|} \sum_{u_a, u_b \in c} |\hat{u}_a - \hat{u}_b|}_{\text{Invariants Penalty}} + \underbrace{\sum_{c > c' \in C_r} \frac{1}{|c'|} \sum_{u_a \in c, u_b \in c'} \max(0, \hat{u}_b - \hat{u}_a)}_{\text{Inequalities Penalty}}. \quad (4.1)$$

The normalization factors $\frac{1}{|c|}$ and $\frac{1}{|c'|}$ were used to reduce a topological bias that arose from differences in the sizes of the equivalence classes for each tree shape.

4.2.3 Quintet Sampling

The set Q of quintets in the QR algorithm can be selected in different ways, and here we consider sampling strategies that lead to statistical consistency. These sampling strategies differ in the number of quintets they sample and therefore their runtime. A straightforward sampling strategy is to use all $\Theta(n^5)$ quintets, which was the main approach used in [230]. Alternatively, an $O(n)$ sampling method called “Linear Encoding” was proposed, as we now describe. The set Q_{LE} contains one quintet for each edge in the tree T , with the quintets computed as follows. If the edge e is incident to a leaf x , we note that deleting e partitions the set of **taxa** into three sets $\{x\}, A, B$; for this edge, the quintet q_e is formed by selecting $\{x\}$ and at least one leaf from each of the sets A and B , with the other two leaves of q_e being randomly selected. For any edge e that is **internal** in the tree, the removal of e partitions the set of **taxa** into four subsets A_1, A_2, B_1 and B_2 , and a quintet q_e is formed by picking one taxon from each of these four sets, and then picking a fifth taxon arbitrarily. Since a tree with n leaves has $2n-3$ edges, $|Q_{LE}| = 2n-3$ and therefore the runtime of the preprocessing step is $O(nk)$ when using the linear encoding. Note also that $Q_{LE}(T)$ may not be unique for trees with $n > 5$ taxa.

4.2.4 Lack of consistency for QR

QR uses the cost function in Eq. 4.1 to select between different rootings of a 5-taxon **unrooted** species tree, given the estimated quintet distribution $\vec{\hat{u}}$. Lemma 4.3 shows that for each balanced tree, there are two caterpillar trees for which the set of violated inequalities becomes empty. Consider the caterpillar tree R_1 and balanced tree R_{76} shown in Figure 4.1. Note that class c_3 in R_{76} is the result of merging the classes c_2 and c_3 in R_1 . Moreover, class c_4 in R_{76} is the result of merging classes c_5 and c_6 in R_1 . Assume that the model tree is R_{76} , and we have estimated $\vec{\hat{u}}$ given a set of k **unrooted** quintet gene trees. We now argue

informally that even as k increases, there is no guarantee that eventually $\text{Cost}(R_{76}, \vec{\hat{u}}) < \text{Cost}(R_1, \vec{\hat{u}})$ for all large enough k .

According to the proof of Lemma 4.6 in Sec. 4.4, as k increases, for the model tree R_{76} , all inequality penalty terms in the form of $\max(0, \hat{u}_b - \hat{u}_a)$ will converge to zero in probability. Therefore, roughly speaking, the cost of R_{76} eventually consists primarily of the invariant penalty terms. For the caterpillar tree R_1 , most of its inequality penalty terms are also penalty terms in the cost of R_{76} , but it also has additional penalty terms between classes c_2 and c_3 as well as classes c_5 and c_6 that are merged in R_{76} . By simplifying the penalty terms that are eventually zero with high probability or are included in the cost of both trees, we get

$$\begin{aligned} \text{Cost}(R_1, \vec{\hat{u}}) - \text{Cost}(R_{76}, \vec{\hat{u}}) \approx \\ \max(0, \hat{u}_2 - \hat{u}_3) + \frac{1}{2} \max(0, \hat{u}_5 - \hat{u}_6) + \frac{1}{2} \max(0, \hat{u}_5 - \hat{u}_9) + \frac{1}{2} \max(0, \hat{u}_{12} - \hat{u}_6) \\ + \frac{1}{2} \max(0, \hat{u}_{12} - \hat{u}_9) + \frac{1}{2} |\hat{u}_6 - \hat{u}_9| + \frac{1}{2} |\hat{u}_5 - \hat{u}_{12}| \\ - \left(\frac{1}{2} |\hat{u}_2 - \hat{u}_3| + \frac{1}{4} |\hat{u}_5 - \hat{u}_6| + \frac{1}{4} |\hat{u}_5 - \hat{u}_9| + \frac{1}{4} |\hat{u}_5 - \hat{u}_{12}| + \frac{1}{4} |\hat{u}_6 - \hat{u}_9| + \frac{1}{4} |\hat{u}_6 - \hat{u}_{12}| + \frac{1}{4} |\hat{u}_9 - \hat{u}_{12}| \right) \end{aligned} \quad (4.2)$$

In the limit as $k \rightarrow +\infty$, all remaining terms in that difference also go to 0, since each term corresponds to a difference between two probabilities that are in the same equivalence class under the model tree. Hence, intuitively, there is no guarantee that R_{76} will be selected by QR. Based on this informal argument, we conjecture that QR is not statistically consistent.

4.3 QR-STAR

QR-STAR is an extension to QR that has an additional step for determining the **rooted** shape (i.e., the **rooted topology** without the leaf labels) of each quintet tree, as well as an associated penalty term in its cost function. This penalty term compares the **rooted** shape of the 5-taxon tree, denoted by $S(r)$, with the **rooted** shape inferred by QR-STAR from the given quintet distribution, denoted by $\hat{S}(\hat{u})$. The motivation for this additional preprocessing step is that, as we argued in the previous section, the cost function of QR does not guarantee statistical consistency. The cost function of QR-STAR takes the following general form:

$$\text{Cost}^*(r, \vec{\hat{u}}) = \underbrace{\sum_{c \in C_r} \sum_{u_a, u_b \in c} \alpha_{a,b} |\hat{u}_a - \hat{u}_b|}_{\text{Invariants Penalty}} + \underbrace{\sum_{c > c' \in C_r} \sum_{u_a \in c, u_b \in c'} \beta_{a,b} \max(0, \hat{u}_b - \hat{u}_a)}_{\text{Inequalities Penalty}} + \underbrace{C \mathbb{1}|S(r) \neq \hat{S}(\hat{u})|}_{\text{Shape Penalty}} \quad (4.3)$$

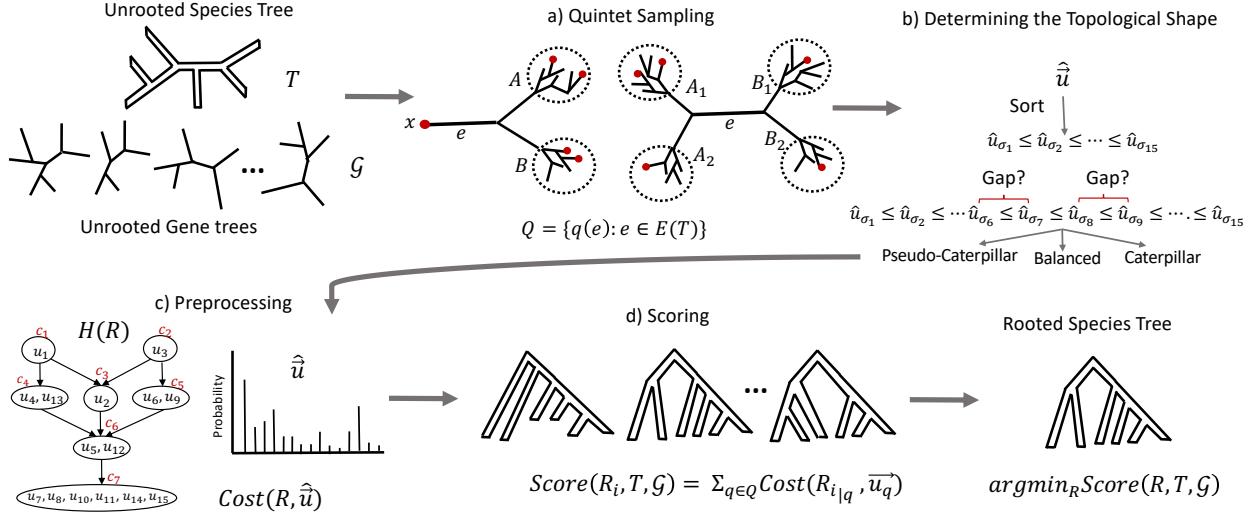


Figure 4.3: **QR-STAR Pipeline.** The input is an **unrooted species tree** T and a set of **unrooted** gene trees \mathcal{G} on the same leafset. a) The sampling step selects a set Q of quintets from the leafset of T (shown is the linear encoding sampling). b) The step that determines the **rooted** shape for each selected quintet. c) The preprocessing step computes a cost for each of the seven possible rootings of each selected quintet. d) The scoring step computes a score for each **rooted** tree in the search space based on the costs computed in the preprocessing step, and returns a rooting of T with minimum score. The QR pipeline skips the step that determines the **rooted** shape for each selected quintet, and has a simpler cost function.

where for all a, b , we require $\alpha_{a,b} \geq 0$ and $\beta_{a,b}, C > 0$ ². Let $\alpha_{\max} = \max_{a,b}(\alpha_{a,b})$ and $\beta_{\min} = \min_{a,b}(\beta_{a,b})$ where a, b ranges over all pairs of indices a, b used in the penalty terms in Eq. 4.3.

Each of the 105 **rooted binary** trees on a given set of 5 leaves have a unique set of inequalities and invariants that can be derived from the **ADR** theory. The cost function in Eq. 4.3 considers a penalty term for these inequalities and invariants as well as the shape of the tree, so that $\text{Cost}^*(r, \vec{u})$ is minimized for a **rooted** 5-taxon tree r that best describes the given estimated quintet distribution \vec{u} .

4.3.1 Determining the Rooted Shape

The different **rooted** shapes (i.e., caterpillar, balanced, pseudo-caterpillar) of model 5-taxon species trees define equivalence classes with different class sizes on the **unrooted** gene tree probability distribution. These class sizes can be used to determine the unlabeled shape of a **rooted** tree, when given the *true* gene tree probability distribution. For example, the size of the equivalence class with the smallest gene tree probabilities is 8 for the pseudo-

²See Remark 4.1 for why $\alpha_{a,b}$ does not need to be strictly positive.

caterpillar trees and 6 for balanced or caterpillar trees. Therefore, the size of the equivalence class corresponding to the minimal element in the partial order can differentiate a pseudo-caterpillar tree from other tree shapes. Moreover, both caterpillar and balanced trees have a unique class with the second smallest probability, which is of size 2 for caterpillar trees and size 4 for balanced trees, and this can be used to differentiate a caterpillar tree from a balanced tree. This approach is used in Theorem 9 in [81] for establishing the identifiability of **rooted** 5-taxon trees from **unrooted** gene trees.

However, given an *estimated* gene tree distribution, it is likely that none of the invariants derived from the **ADR** theory exactly hold, and so the class sizes cannot be directly determined and the approach above cannot be used as is to infer the shape of a **rooted** quintet. Here we propose a simple modification for determining the **rooted** shape of a tree from the estimated distribution of **unrooted** gene trees, by looking for significant gaps between quintet gene tree probabilities.

Let T be the **unrooted species tree** with $n \geq 5$ leaves given to QR-STAR and q be a quintet of **taxa** from $\mathcal{L}(T)$. Let $\vec{\hat{u}}$ be the quintet distribution estimated from input gene trees induced on **taxa** in set q . QR-STAR first sorts $\vec{\hat{u}}$ in ascending order to get $\hat{u}_{\sigma_1} \leq \hat{u}_{\sigma_2} \leq \dots \leq \hat{u}_{\sigma_{15}}$.

We propose a general design for QR-STAR based on a given error probability $\delta > 0$, so that the algorithm returns the true **rooted** tree with probability at least $1 - \delta$. Given δ , we define $A_{Q,\delta}(k) = \sqrt{\frac{2}{k} \ln(\frac{30|Q|}{\delta})}$ (refer to Lemma 4.4 for the derivation), where k is the number of input gene trees and Q is the set of sampled quintets, which depends on the number n of **taxa** and is assumed fixed. The first step of QR-STAR computes an estimate of the **rooted** shape of a quintet q , denoted by $\hat{S}(\hat{u})$ in Eq. 4.3, as follows:

- estimate the rooted shape $\hat{S}(\hat{u})$ as pseudo-caterpillar if $\hat{u}_{\sigma_7} - \hat{u}_{\sigma_6} < A_{Q,\delta}(k)$;
- estimate the rooted shape $\hat{S}(\hat{u})$ as balanced if $\hat{u}_{\sigma_7} - \hat{u}_{\sigma_6} \geq A_{Q,\delta}(k)$ and $\hat{u}_{\sigma_9} - \hat{u}_{\sigma_8} < A_{Q,\delta}(k)$;
- estimate the rooted shape $\hat{S}(\hat{u})$ as caterpillar if $\hat{u}_{\sigma_7} - \hat{u}_{\sigma_6} \geq A_{Q,\delta}(k)$ and $\hat{u}_{\sigma_9} - \hat{u}_{\sigma_8} \geq A_{Q,\delta}(k)$.

The runtime of QR-STAR is the same as QR, as determining the topological shape for each quintet is done in constant time, and the overall runtime remains $O(nk)$, when a linear sampling of quintets is used. Figure 4.3 shows the pipeline of QR-STAR and its individual steps.

4.4 THEORETICAL RESULTS

In this section, we provide the main theoretical results, starting with a series of lemmas and theorems that will be used in the proof of statistical consistency of QR-STAR in Theorem 4.2. Throughout this chapter, we assume that discordance between species trees and gene trees is solely due to ILS. In establishing statistical consistency, we assume that input gene trees are true gene trees and, thus, have no gene tree estimation error. All trees are assumed to be **fully resolved** (i.e., **binary**).

4.4.1 Preliminaries

We begin with some definitions and key observations.

Definition 4.1 (Path length parameter). Let R be an **MSC** model species tree. Let $f(R)$ be the length of the shortest **internal** branch of R and $g(R)$ be the length of the longest **internal** path (i.e., a path formed from only the **internal** branches) of R . We define the path length parameter of R as

$$h(R) = \frac{1}{18} e^{-3g(R)} (1 - e^{-f(R)})^2 \quad (4.4)$$

Note that $h(R) \in (0, \frac{1}{18})$ since $\exp(-x) \in (0, 1)$ for all $x > 0$ and the branch lengths have positive values. The formula for Eq. 4.4 is derived from the proof of Lemma 4.2.

Lemma 4.1. Let R be an **MSC** model **species tree** with $n \geq 5$ leaves and q be an arbitrary set of 5 leaves from $\mathcal{L}(R)$. Then $h(R|_q) \geq h(R)$ where $R|_q$ is the rooted tree R restricted to **taxa** in set q .

Proof. Every **internal** path of $R|_q$ is also an **internal** path in R , and therefore $g(R|_q) \leq g(R)$. Also, every branch in $R|_q$ is formed from one or more branches in R , so the shortest branch in $R|_q$ is at least as long as the shortest branch in R , and therefore $f(R|_q) \geq f(R)$. Hence,

$$h(R|_q) = \frac{1}{18} e^{-3g(R|_q)} (1 - e^{-f(R|_q)})^2 \geq \frac{1}{18} e^{-3g(R)} (1 - e^{-f(R)})^2 = h(R) \quad (4.5)$$

QED.

Lemma 4.2. Let R be an **MSC** model **species tree** with 5 leaves and **internal** branch lengths x, y , and z . Let \vec{u} be the probability distribution that R defines on the **unrooted** 5-taxon gene tree topologies. If $\vec{\hat{u}}$ is an estimate of \vec{u} such that given $\epsilon > 0$, we have $|\hat{u}_i - u_i| < \epsilon$ for

all $1 \leq i \leq 15$, then the following inequality holds:

$$\forall_{c>c' \in C_R} \forall_{u_a \in c, u_b \in c'} : \hat{u}_a - \hat{u}_b > h(R) - 2\epsilon. \quad (4.6)$$

Proof. Let $X = e^{-x}$, $Y = e^{-y}$ and $Z = e^{-z}$. According to the explicit formulas for the probability distribution of unrooted gene trees \vec{u} under a 5-taxon model species tree provided in Appendix B of [81], the exact value of each u_i can be expressed as a polynomial with variables X , Y and Z . We show that the lemma holds for all pairs u_a, u_b from different equivalence classes, for each tree category. For each category, we only show that the lemma holds for one example tree with that rooted shape (trees in Table 1 in [230]), as the rest of the trees have \vec{u} distributions that are only permutations of the distributions of these three example trees (see Supplementary Materials Sec. S2 in [230]) and the explicit formulas remain the same. The following equations can be derived using elementary algebraic arguments, and the fact that $X, Y, Z \in (0, 1)$.

Caterpillar Trees. For a caterpillar tree $R = (((((a, b) : x, c) : y, d) : z, e))$ with $C_R = \{c_1 : \{u_1\}, c_2 : \{u_3\}, c_3 : \{u_2\}, c_4 : \{u_4, u_{13}\}, c_5 : \{u_6, u_9\}, c_6 : \{u_5, u_{12}\}, c_7 : \{u_7, u_8, u_{10}, u_{11}, u_{14}, u_{15}\}\}$, we have $c_1 > c_3, c_4 > c_6 > c_7$ and $c_2 > c_3, c_5 > c_6 > c_7$. Therefore,

- $u_a \in c_1, u_b \in c_3$

$$\begin{aligned} u_a - u_b &= \left(1 - \frac{2}{3}X - \frac{2}{3}Y + \frac{1}{3}XY + \frac{1}{18}XY^3 + \frac{1}{90}XY^3Z^6\right) - \\ &\quad \left(\frac{1}{3}Y - \frac{1}{6}XY - \frac{1}{9}XY^3 + \frac{1}{90}XY^3Z^6\right) = \\ 1 - \frac{2}{3}X - Y + \frac{1}{2}XY + \frac{1}{6}XY^3 &= (1 - Y) - \frac{1}{6}X(4 - 3Y - Y^3) = \\ (1 - Y) - \frac{1}{6}X(1 - Y)(4 + Y + Y^2) &\geq (1 - Y) - \frac{1}{6}X(1 - Y)6 = (1 - Y)(1 - X) \\ \Rightarrow u_a - u_b &\geq (1 - Y)(1 - X) \geq (1 - e^{-f(R)})^2 > h(R) \end{aligned} \quad (4.7)$$

where $(1 - Y)(1 - X) \geq (1 - e^{-f(R)})^2$ follows from the fact that $(1 - X) = 1 - e^{-x} \geq 1 - e^{-f(R)}$ as $x \geq f(R)$, and same is true for y .

- $u_a \in c_1, u_b \in c_4$

$$\begin{aligned}
u_a - u_b &= (1 - \frac{2}{3}X - \frac{2}{3}Y + \frac{1}{3}XY + \frac{1}{18}XY^3 + \frac{1}{90}XY^3Z^6) - \\
&\quad (\frac{1}{3}X - \frac{1}{3}XY + \frac{1}{18}XY^3 + \frac{1}{90}XY^3Z^6) = \\
1 - X - \frac{2}{3}Y + \frac{2}{3}XY &= (1 - X)(1 - \frac{2}{3}Y) > \frac{1}{3}(1 - X) \\
\Rightarrow u_a - u_b &> \frac{1}{3}(1 - X) \geq \frac{1}{3}(1 - e^{-f(R)}) > h(R)
\end{aligned} \tag{4.8}$$

- $u_a \in c_2, u_b \in c_3$

$$\begin{aligned}
u_a - u_b &= (\frac{1}{3}Y - \frac{1}{6}XY - \frac{1}{18}XY^3 - \frac{2}{45}XY^3Z^6) - \\
&\quad (\frac{1}{3}Y - \frac{1}{6}XY - \frac{1}{9}XY^3 + \frac{1}{90}XY^3Z^6) = \\
\frac{1}{18}XY^3 - \frac{1}{18}XY^3Z^6 &= \frac{1}{18}XY^3(1 - Z^6) = \frac{1}{18}XY^3(1 - Z)(1 + Z + Z^2)(1 + Z^3) \\
&> \frac{1}{18}XY^3(1 - Z) \\
\Rightarrow u_a - u_b &> \frac{1}{18}XY^3(1 - Z) \geq \frac{1}{18}e^{-3g(R)}(1 - e^{-f(R)}) > h(R)
\end{aligned} \tag{4.9}$$

where $XY^3 \geq e^{-3g(R)}$ follows from the fact that $XY = e^{-x}e^{-y} = e^{-(x+y)} \geq e^{-g(R)}$ as $x + y$ correspond to a (sub)-length of an **internal** path in R and hence $x + y \leq g(R)$ and $Y^2 \geq e^{-2g(R)}$ as $y \leq g(R)$.

- $u_a \in c_2, u_b \in c_5$

$$\begin{aligned}
u_a - u_b &= (\frac{1}{3}Y - \frac{1}{6}XY - \frac{1}{18}XY^3 - \frac{2}{45}XY^3Z^6) - (\frac{1}{6}XY - \frac{1}{18}XY^3 - \frac{2}{45}XY^3Z^6) = \\
&\quad \frac{1}{3}Y - \frac{1}{3}XY = \frac{1}{3}Y(1 - X) \\
\Rightarrow u_a - u_b &= \frac{1}{3}Y(1 - X) \geq \frac{1}{3}e^{-g(R)}(1 - e^{-f(R)}) > h(R)
\end{aligned} \tag{4.10}$$

- $u_a \in c_3, u_b \in c_6$

$$\begin{aligned}
u_a - u_b &= \left(\frac{1}{3}Y - \frac{1}{6}XY - \frac{1}{9}XY^3 + \frac{1}{90}XY^3Z^6 \right) - \left(\frac{1}{6}XY - \frac{1}{9}XY^3 + \frac{1}{90}XY^3Z^6 \right) = \\
&\quad \frac{1}{3}Y - \frac{1}{3}XY = \frac{1}{3}Y(1 - X) \\
\Rightarrow u_a - u_b &= \frac{1}{3}Y(1 - X) \geq \frac{1}{3}e^{-g(R)}(1 - e^{-f(R)}) > h(R)
\end{aligned} \tag{4.11}$$

- $u_a \in c_4, u_b \in c_6$

$$\begin{aligned}
u_a - u_b &= \left(\frac{1}{3}X - \frac{1}{3}XY + \frac{1}{18}XY^3 + \frac{1}{90}XY^3Z^6 \right) - \left(\frac{1}{6}XY - \frac{1}{9}XY^3 + \frac{1}{90}XY^3Z^6 \right) = \\
&\quad \frac{1}{3}X - \frac{1}{2}XY + \frac{1}{6}XY^3 = \frac{1}{6}X(2 - 3Y + Y^3) = \frac{1}{6}X(1 - Y)^2(Y + 2) > \frac{1}{3}X(1 - Y)^2 \\
\Rightarrow u_a - u_b &> \frac{1}{3}X(1 - Y)^2 \geq \frac{1}{3}e^{-g(R)}(1 - e^{-f(R)})^2 > h(R)
\end{aligned} \tag{4.12}$$

- $u_a \in c_5, u_b \in c_6$

$$\begin{aligned}
u_a - u_b &= \left(\frac{1}{6}XY - \frac{1}{18}XY^3 - \frac{2}{45}XY^3Z^6 \right) - \left(\frac{1}{6}XY - \frac{1}{9}XY^3 + \frac{1}{90}XY^3Z^6 \right) = \\
&\quad \frac{1}{18}XY^3 - \frac{1}{18}XY^3Z^6 = \frac{1}{18}XY^3(1 - Z^6) = \frac{1}{18}XY^3(1 - Z)(1 + Z + Z^2)(1 + Z^3) \\
&\quad > \frac{1}{18}XY^3(1 - Z) \\
\Rightarrow u_a - u_b &> \frac{1}{18}XY^3(1 - Z) \geq \frac{1}{18}e^{-3g(R)}(1 - e^{-f(R)}) > h(R)
\end{aligned} \tag{4.13}$$

- $u_a \in c_6, u_b \in c_7$

$$\begin{aligned}
u_a - u_b &= \left(\frac{1}{6}XY - \frac{1}{9}XY^3 + \frac{1}{90}XY^3Z^6 \right) - \left(\frac{1}{18}XY^3 + \frac{1}{90}XY^3Z^6 \right) = \\
&\quad \frac{1}{6}XY - \frac{1}{6}XY^3 = \frac{1}{6}XY(1 - Y^2) = \frac{1}{6}XY(1 - Y)(1 + Y) > \frac{1}{6}XY(1 - Y) \\
\Rightarrow u_a - u_b &> \frac{1}{6}XY(1 - Y) \geq \frac{1}{6}e^{-g(R)}(1 - e^{-f(R)}) > h(R)
\end{aligned} \tag{4.14}$$

Balanced Trees. For a balanced model species tree $R = (((a, b) : x, c) : y, (d, e) : z)$, with $C_R = \{c_1 : \{u_1\}, c_2 : \{u_4, u_{13}\}, c_3 : \{u_2, u_3\}, c_4 : \{u_5, u_6, u_9, u_{12}\}, c_5 : \{u_7, u_8, u_{10}, u_{11}, u_{14}, u_{15}\}\}$, we have $c_1 > c_2, c_3 > c_4 > c_5$. Therefore,

- $u_a \in c_1, u_b \in c_3$

$$\begin{aligned}
u_a - u_b &= (1 - \frac{2}{3}X - \frac{2}{3}YZ + \frac{1}{3}XYZ + \frac{1}{15}XY^3Z) - (\frac{1}{3}YZ - \frac{1}{6}XYZ - \frac{1}{10}XY^3Z) = \\
&1 - \frac{2}{3}X - YZ + \frac{1}{2}XYZ + \frac{1}{6}XY^3Z = 1 - YZ - \frac{1}{6}X(4 - 3YZ - Y^3Z) > \\
&1 - YZ - \frac{1}{6}(4 - 3YZ - Y^3Z) = \frac{1}{3} - \frac{1}{6}YZ(3 - Y^2) > \\
&\frac{1}{3} - \frac{1}{6}Y(3 - Y^2) = \frac{1}{6}(2 - 3Y + Y^3) = \frac{1}{6}(1 - Y)^2(2 + Y) > \frac{1}{3}(1 - Y)^2 \\
&\Rightarrow u_a - u_b > \frac{1}{3}(1 - Y)^2 \geq \frac{1}{3}(1 - e^{-f(R)})^2 > h(R)
\end{aligned} \tag{4.15}$$

- $u_a \in c_1, u_b \in c_2$

$$\begin{aligned}
u_a - u_b &= (1 - \frac{2}{3}X - \frac{2}{3}YZ + \frac{1}{3}XYZ + \frac{1}{15}XY^3Z) - (\frac{1}{3}X - \frac{1}{3}XYZ + \frac{1}{15}XY^3Z) = \\
&1 - X - \frac{2}{3}YZ + \frac{2}{3}XYZ = (1 - X)(1 - \frac{2}{3}YZ) > \frac{1}{3}(1 - X) \\
&\Rightarrow u_a - u_b > \frac{1}{3}(1 - X) \geq \frac{1}{3}(1 - e^{-f(R)}) > h(R)
\end{aligned} \tag{4.16}$$

- $u_a \in c_3, u_b \in c_4$

$$\begin{aligned}
u_a - u_b &= (\frac{1}{3}YZ - \frac{1}{6}XYZ - \frac{1}{10}XY^3Z) - (\frac{1}{6}XYZ - \frac{1}{10}XY^3Z) = \\
&\frac{1}{3}YZ - \frac{1}{3}XYZ = \frac{1}{3}YZ(1 - X) \\
&\Rightarrow u_a - u_b = \frac{1}{3}YZ(1 - X) \geq \frac{1}{3}e^{-g(R)}(1 - e^{-f(R)}) > h(R)
\end{aligned} \tag{4.17}$$

- $u_a \in c_2, u_b \in c_4$

$$\begin{aligned}
u_a - u_b &= (\frac{1}{3}X - \frac{1}{3}XYZ + \frac{1}{15}XY^3Z) - (\frac{1}{6}XYZ - \frac{1}{10}XY^3Z) = \\
&\frac{1}{3}X - \frac{1}{2}XYZ + \frac{1}{6}XY^3Z = \frac{1}{6}X(2 - 3YZ + Y^3Z) > \frac{1}{6}X(2Z - 3YZ + Y^3Z) = \\
&\frac{1}{6}XZ(2 - 3Y + Y^3) = \frac{1}{6}XZ(2 + Y)(1 - Y)^2 > \frac{1}{3}XZ(1 - Y)^2 \\
&\Rightarrow u_a - u_b > \frac{1}{3}XZ(1 - Y)^2 \geq \frac{1}{3}e^{-g(R)}(1 - e^{-f(R)})^2 > h(R)
\end{aligned} \tag{4.18}$$

- $u_a \in c_4, u_b \in c_5$

$$\begin{aligned}
u_a - u_b &= \left(\frac{1}{6}XYZ - \frac{1}{10}XY^3Z \right) - \left(\frac{1}{15}XY^3Z \right) = \\
\frac{1}{6}XYZ - \frac{1}{6}XY^3Z &= \frac{1}{6}(XYZ)(1 - Y^2) = \frac{1}{6}(XYZ)(1 - Y)(1 + Y) > \frac{1}{6}(XYZ)(1 - Y) \\
\Rightarrow u_a - u_b &> \frac{1}{6}(XYZ)(1 - Y) \geq \frac{1}{6}e^{-g(R)}(1 - e^{-f(R)}) > h(R)
\end{aligned} \tag{4.19}$$

where $XYZ \geq e^{-g(R)}$ follows from $x + y + z \leq g(R)$.

Pseudo-caterpillar Trees. For a pseudo-caterpillar model [species tree](#) $R = (((a, b) : x, (d, e) : y) : z, c)$ with $C_R = \{c_1 : \{u_1\}, c_2 : \{u_4, u_{13}\}, c_3 : \{u_2, u_3\}, c_4 : \{u_8, u_{11}\}, c_5 : \{u_5, u_6, u_7, u_9, u_{10}, u_{12}, u_{14}\}\}$ we have $c_1 > c_2, c_3, c_4 > c_5$. Therefore,

- $u_a \in c_1, u_b \in c_2$

$$\begin{aligned}
u_a - u_b &= \left(1 - \frac{2}{3}X - \frac{2}{3}Y + \frac{4}{9}XY - \frac{2}{45}XYZ^6 \right) - \left(\frac{1}{3}X - \frac{5}{18}XY + \frac{1}{90}XYZ^6 \right) = \\
1 - X - \frac{2}{3}Y + \frac{13}{18}XY - \frac{1}{18}XYZ^6 &= 1 - X - \frac{2}{3}Y + \frac{2}{3}XY + \frac{1}{18}XY - \frac{1}{18}XYZ^6 \\
&= (1 - X)\left(1 - \frac{2}{3}Y\right) + \frac{1}{18}XY(1 - Z^6) > \frac{1}{18}XY(1 - Z)(1 + Z + Z^2)(1 + Z^3) \\
&> \frac{1}{18}XY(1 - Z) \Rightarrow u_a - u_b > \frac{1}{18}XY(1 - Z) \geq \frac{1}{18}e^{-g(R)}(1 - e^{-f(R)}) > h(R)
\end{aligned} \tag{4.20}$$

- $u_a \in c_1, u_b \in c_3$

$$\begin{aligned}
u_a - u_b &= \left(1 - \frac{2}{3}X - \frac{2}{3}Y + \frac{4}{9}XY - \frac{2}{45}XYZ^6 \right) - \left(\frac{1}{3}Y - \frac{5}{18}XY + \frac{1}{90}XYZ^6 \right) = \\
1 - \frac{2}{3}X - Y + \frac{13}{18}XY - \frac{1}{18}XYZ^6 &= 1 - \frac{2}{3}X - Y + \frac{2}{3}XY + \frac{1}{18}XY - \frac{1}{18}XYZ^6 \\
&= (1 - Y)\left(1 - \frac{2}{3}X\right) + \frac{1}{18}XY(1 - Z^6) > \frac{1}{18}XY(1 - Z)(1 + Z + Z^2)(1 + Z^3) > \\
\frac{1}{18}XY(1 - Z) &\Rightarrow u_a - u_b > \frac{1}{18}XY(1 - Z) \geq \frac{1}{18}e^{-g(R)}(1 - e^{-f(R)}) > h(R)
\end{aligned} \tag{4.21}$$

- $u_a \in c_1, u_b \in c_4$

$$\begin{aligned}
u_a - u_b &= \left(1 - \frac{2}{3}X - \frac{2}{3}Y + \frac{4}{9}XY - \frac{2}{45}XYZ^6\right) - \left(\frac{1}{9}XY - \frac{2}{45}XYZ^6\right) = \\
1 - \frac{2}{3}X - \frac{2}{3}Y + \frac{1}{3}XY &= (1-X)(1-Y) - \frac{2}{3}XY + \frac{1}{3}X + \frac{1}{3}Y = \\
(1-X)(1-Y) + \frac{1}{3}(X+Y-2XY) &> (1-X)(1-Y) + \frac{1}{3}(X^2+Y^2-2XY) = \\
(1-X)(1-Y) + \frac{1}{3}(X-Y)^2 &> (1-X)(1-Y) \\
\Rightarrow u_a - u_b &> (1-X)(1-Y) \geq (1-e^{-f(R)})^2 > h(R)
\end{aligned} \tag{4.22}$$

- $u_a \in c_2, u_b \in c_5$

$$\begin{aligned}
u_a - u_b &= \left(\frac{1}{3}X - \frac{5}{18}XY + \frac{1}{90}XYZ^6\right) - \left(\frac{1}{18}XY + \frac{1}{90}XYZ^6\right) = \\
\frac{1}{3}X - \frac{1}{3}XY &= \frac{1}{3}X(1-Y) \\
\Rightarrow u_a - u_b &= \frac{1}{3}X(1-Y) \geq \frac{1}{3}e^{-g(R)}(1-e^{-f(R)}) > h(R)
\end{aligned} \tag{4.23}$$

- $u_a \in c_3, u_b \in c_5$

$$\begin{aligned}
u_a - u_b &= \left(\frac{1}{3}Y - \frac{5}{18}XY + \frac{1}{90}XYZ^6\right) - \left(\frac{1}{18}XY + \frac{1}{90}XYZ^6\right) = \\
\frac{1}{3}Y - \frac{1}{3}XY &= \frac{1}{3}Y(1-X) \\
\Rightarrow u_a - u_b &= \frac{1}{3}Y(1-X) \geq \frac{1}{3}e^{-g(R)}(1-e^{-f(R)}) > h(R)
\end{aligned} \tag{4.24}$$

- $u_a \in c_4, u_b \in c_5$

$$\begin{aligned}
u_a - u_b &= \left(\frac{1}{9}XY - \frac{2}{45}XYZ^6\right) - \left(\frac{1}{18}XY + \frac{1}{90}XYZ^6\right) = \\
\frac{1}{18}XY - \frac{1}{18}XYZ^6 &= \frac{1}{18}XY(1-Z^6) = \frac{1}{18}XY(1-Z)(1+Z+Z^2)(1+Z^3) > \\
\frac{1}{18}XY(1-Z) &\Rightarrow u_a - u_b > \frac{1}{18}XY(1-Z) \geq \frac{1}{18}e^{-g(R)}(1-e^{-f(R)}) > h(R)
\end{aligned} \tag{4.25}$$

QED.

Therefore, we have

$$\forall_{c>c' \in C_R} \forall_{u_a \in c, u_b \in c'} : u_a - u_b > \frac{1}{18}e^{-3g(R)}(1-e^{-f(R)})^2 = h(R) \tag{4.26}$$

Since $|u_i - \hat{u}_i| < \epsilon$, we have $-\epsilon < \hat{u}_i - u_i < \epsilon$ and $-\epsilon < u_i - \hat{u}_i < \epsilon$. According to Eq. 4.26,

$$\begin{aligned}\hat{u}_a - \hat{u}_b &= (u_a - u_b) + (\hat{u}_a - u_a) + (u_b - \hat{u}_b) \Rightarrow \hat{u}_a - \hat{u}_b > (u_a - u_b) - \epsilon - \epsilon \\ &\Rightarrow \hat{u}_a - \hat{u}_b > h(R) - 2\epsilon\end{aligned}\quad (4.27)$$

Definition 4.2. For a 5-taxon rooted tree R , we define I_R as the set of ordered pairs (i, j) , $1 \leq i \neq j \leq 15$, corresponding to inequalities in the form $u_i > u_j$ defined according to the partial order of R . The inequalities that are a result of transitivity (i.e. $u_i > u_j$ and $u_j > u_k$ implies $u_i > u_k$) are not included in I_R .

Definition 4.3. Let $V(R, R')$ be the set of violated inequalities of two rooted 5-taxon trees R and R' , i.e., all pairs $\{i, j\}$ such that $(i, j) \in I_R$ and $(j, i) \in I_{R'}$.

Figure 4.4(a) shows an example of $V(R, R')$ computed for caterpillar trees and Figure 4.4(b) is a heatmap showing the function $|V(R, R')|$ computed for the seven possible rootings of an unrooted quintet tree. The set $V(R, R')$ can be easily computed from I_R and $I_{R'}$ for all pairs of rooted 5-taxon trees, and I_R is derived from the ADR theory for all 105 5-taxon rooted trees in the Supplementary Materials, Sec. S2 in [230].

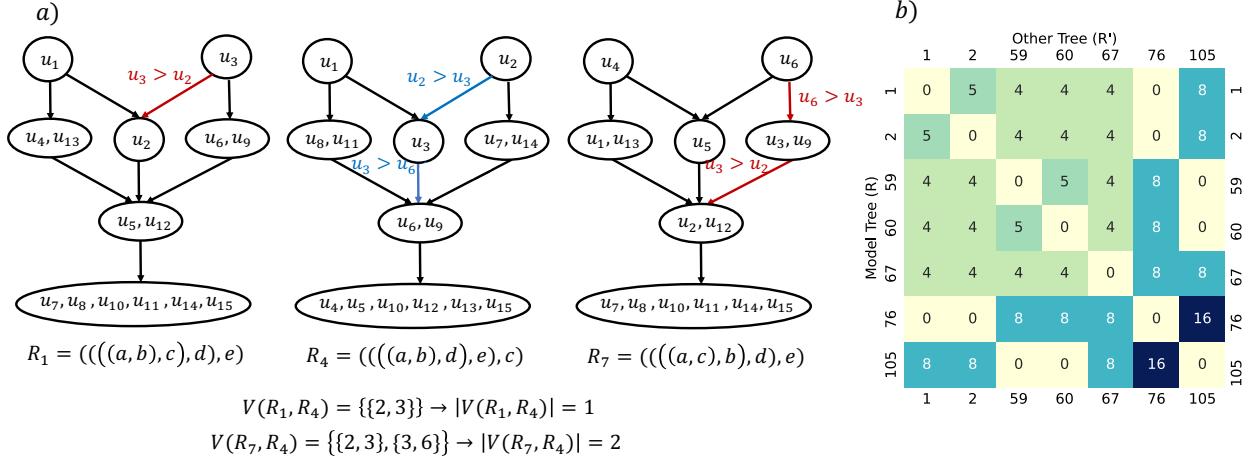


Figure 4.4: Conflicting inequality penalty terms between rooted 5-taxon species trees. a) Set of violated inequality penalty terms in the partial orders of R_1 and R_7 with respect to R_4 , which are all caterpillar trees. The red edges show violations of inequalities in tree R_4 , highlighted in blue. b) Heatmap showing the number of pairwise violated penalty terms (function $|V(R, R')|$) of seven possible rooted trees having unrooted topology with bipartitions $ab|cde$ and $abc|de$. The dark colors indicate more violations, and the lightest color corresponds to no violations ($|V(R, R')| = 0$).

Lemma 4.3. (a) For 5-taxon [binary](#) rooted trees R and R' with the same rooted shape, the set $V(R, R')$ is always non-empty. (b) For each balanced tree B , there exist two caterpillar trees C_1 and C_2 such that $V(B, C_i) = \emptyset$ for $i = 1, 2$.

Proof. (a) Figures 4.11, 4.12 and 4.10 show the function $|V(R, R')|$ (number of violated inequalities) for all rooted quintet tree pairs R and R' with the same unlabeled topological shape (i.e., caterpillar, balanced and pseudo-caterpillar), computed using the invariants and inequalities derived from the [ADR](#) theory (for details on how these are computed, refer to [230], Supplementary Material Sec. 2). It is clear that, except for the numbers on the main diagonal, all other values are non-zero. Therefore, $V(R, R')$ is always non-empty when R and R' have the same rooted topological shape.

(b) W.L.O.G. assume we have a particular [unrooted](#) quintet tree T_1 (see Table 5 in [81]) so that its seven possible rootings are caterpillar trees R_1, R_2, R_{59}, R_{60} , pseudo-caterpillar tree R_{67} and balanced trees R_{76} and R_{105} . Figure 4.4(b) shows the function $|V(R, R')|$ for all these trees, and it is evident that for the balanced trees R_{76} and R_{105} , there are two caterpillar trees (R_1 and R_2 for R_{76} , and R_{59} and R_{60} for R_{105}) for which $|V(R, R')|$ becomes zero. The same can be observed for trees with other [unrooted](#) topologies in Figure 4.13. QED.

Lemma 4.4. Let R be an [MSC](#) model [species tree](#) with $n \geq 5$ leaves and Q be a set of quintets of [taxa](#) from $\mathcal{L}(R)$. Given $\delta > 0$ and $k > 0$ [unrooted](#) gene tree topologies, the following inequality holds, where $A_{Q,\delta}(k) = \sqrt{\frac{2}{k} \ln(\frac{30|Q|}{\delta})}$

$$\mathbb{P}\left(\forall_{q \in Q} \forall_{1 \leq i \leq 15} |(\hat{u}_q)_i - (u_q)_i| < \frac{A_{Q,\delta}(k)}{2}\right) \geq 1 - \delta. \quad (4.28)$$

Proof. For an arbitrary $\epsilon > 0$, we have

$$\begin{aligned} \mathbb{P}(\forall_{q \in Q} \forall_{1 \leq i \leq 15} |(\hat{u}_q)_i - (u_q)_i| < \epsilon) &= 1 - \mathbb{P}(\exists_{q \in Q, 1 \leq i \leq 15} |(\hat{u}_q)_i - (u_q)_i| \geq \epsilon) \\ &\geq 1 - \sum_{q \in Q} \sum_{i=1}^{15} \mathbb{P}(|(\hat{u}_q)_i - (u_q)_i| \geq \epsilon) \end{aligned} \quad (4.29)$$

according to the union bound. Using the Hoeffding inequality [267] for each of the 15 [unrooted](#) 5-taxon tree topologies, we get

$$\mathbb{P}\left(\left|\frac{1}{k} \sum_{j=1}^k X_{q,j,i} - \mu\right| \geq \epsilon\right) \leq 2e^{\frac{-2k\epsilon^2}{(b-a)^2}} \Rightarrow \mathbb{P}(|(\hat{u}_q)_i - (u_q)_i| \geq \epsilon) \leq 2e^{-2k\epsilon^2} \quad (4.30)$$

where $X_{q,j,i}$ is a [binary](#) random variable that is 1 when the quintet gene tree $g|_{q_j}$ has the

unrooted topology T_i and is zero otherwise, and so $0 \leq X_{q,j,i} \leq 1$ almost surely. Substituting Eq. 4.30 in Eq. 4.29, we obtain

$$\mathbb{P}(\forall_{q \in Q} \forall_{1 \leq i \leq 15} |(\hat{u}_q)_i - (u_q)_i| < \epsilon) \geq 1 - \sum_{q \in Q} \sum_{i=1}^{15} \mathbb{P}(|(\hat{u}_q)_i - (u_q)_i| \geq \epsilon) \geq 1 - 30|Q|e^{-2k\epsilon^2} \quad (4.31)$$

Setting $\epsilon = \sqrt{\frac{1}{2k} \ln(\frac{30|Q|}{\delta})} = \frac{A_{Q,\delta}(k)}{2}$ in the equation above proves the lemma

$$\mathbb{P}(\forall_{q \in Q} \forall_{1 \leq i \leq 15} |(\hat{u}_q)_i - (u_q)_i| < \epsilon) \geq 1 - 30|Q|e^{-2k\epsilon^2} = 1 - \delta. \quad (4.32)$$

QED.

4.4.2 Statistical Consistency for 5-leaf Trees

We now establish statistical consistency for QR-STAR under the MSC and provide a sufficient condition for a set of sampled quintets that leads to consistency. That is, we prove that as the number of input true gene trees increases, the probability that QR-STAR and its variants correctly root the given unrooted species tree converges to 1. We first prove statistical consistency for QR-STAR when the model tree has only five taxa in Theorem 4.1 and then extend the proofs to trees with arbitrary numbers of taxa in Theorem 4.2. The main idea of the proof of consistency for 5-taxon trees is that we show as the number of input gene trees increases, the cost of the true rooted tree becomes arbitrarily close to zero, but the cost of any other rooted tree is bounded away from zero, where the bound depends on the path length parameter of the model tree $h(R)$ (see Definition 4.1). To establish statistical consistency in Theorems 4.1 and 4.2, we assume that δ (now seen as a sequence depending on k) is such that $\lim_{k \rightarrow \infty} \delta = 0$ and $\lim_{k \rightarrow \infty} A_{Q,\delta}(k) = 0$. For instance, the choice $\delta = 1/k$ satisfies these assumptions. Throughout this section, we will write $A(k)$ instead of $A_{Q,\delta}(k)$ since the species tree has only five leaves and the δ -sequence is fixed.

Lemma 4.5 (Correct determination of rooted shape). Let R be a 5-taxon model species tree and \vec{u} be the probability distribution that it defines on the unrooted 5-taxon gene tree topologies. There is an integer $k > 0$ such that if we are given at least k unrooted gene trees drawn i.i.d. from the distribution \vec{u} , the first step of QR-STAR will correctly determine the rooted shape of R with probability at least $1 - \delta$.

Proof. Let \mathcal{E} be the event that $|\hat{u}_i - u_i| < \frac{A(k)}{2}$ for all $1 \leq i \leq 15$. Assume k is large enough so that $A(k) < \frac{1}{2}h(R)$. This is indeed possible under our assumption that $\lim_{k \rightarrow \infty} A(k) = 0$.

According to Lemma 4.4, the probability that \mathcal{E} occurs is at least $1 - \delta$. We assume that \mathcal{E} holds in the rest of this proof. In this case, according to Lemma 4.2, we have

$$\forall_{c>c' \in C_R} \forall_{u_a \in c, u_b \in c'} : \hat{u}_a - \hat{u}_b > h(R) - A(k) > A(k) \quad (4.33)$$

where the last inequality is a result of the assumption $A(k) < \frac{1}{2}h(R)$. Therefore, the minimum distance between elements of any two equivalence classes that are related by an inequality in the partial order, i.e., $c > c' \in C_R$, is greater than $A(k)$. Moreover, we now show that the maximum distance between elements inside an equivalence class is less than $A(k)$. Since $u_a = u_b$ and according to the triangle inequality, we obtain:

$$u_a, u_b \in c : |\hat{u}_a - \hat{u}_b| < |\hat{u}_a - u_a| + |u_a - u_b| + |u_b - \hat{u}_b| < \frac{A(k)}{2} + 0 + \frac{A(k)}{2} = A(k) \quad (4.34)$$

The partial orders on **unrooted** gene trees defined for each of the three topological shapes has a unique equivalence class whose members have the minimum probability, and for the caterpillar and balanced shapes, there is a unique class whose members have the second smallest probability. Since the distance between elements in different equivalence classes related by an inequality is greater than $A(k)$, and the distance between elements inside an equivalence class is less than $A(k)$, after sorting $\vec{\hat{u}}$ in ascending order, the elements of the equivalence class with the smallest probability appear at the beginning, followed by the elements of the second smallest class (for caterpillar and balanced shapes). Let $\hat{u}_{\sigma_1} \leq \hat{u}_{\sigma_2} \leq \dots \leq \hat{u}_{\sigma_{15}}$ be the result of sorting $\vec{\hat{u}}$ in ascending order. For a pseudo-caterpillar tree, the class with the minimum probability has 8 elements, and for caterpillar or balanced trees it has 6 elements.

- The first step of QR-STAR determines the tree shape as pseudo-caterpillar if $\hat{u}_{\sigma_7} - \hat{u}_{\sigma_6} < A(k)$ and else it will determine the shape as either caterpillar or balanced. When \hat{u}_{σ_7} and \hat{u}_{σ_6} belong to different classes, their distance must be greater than $A(k)$. Therefore, when $\hat{u}_{\sigma_7} - \hat{u}_{\sigma_6} < A(k)$ holds, \hat{u}_{σ_7} and \hat{u}_{σ_6} must belong to the same equivalence class, and this only happens when the model tree is a pseudo-caterpillar tree. On the other hand, if R is a pseudo-caterpillar tree, then $\hat{u}_{\sigma_7} - \hat{u}_{\sigma_6} < A(k)$, as the first eight elements in the sorted list must belong to the same class. Therefore, condition $\hat{u}_{\sigma_7} - \hat{u}_{\sigma_6} < A(k)$ holds if and only if R has a pseudo-caterpillar shape and QR-STAR determines the correct unlabeled shape in this case.
- If $\hat{u}_{\sigma_7} - \hat{u}_{\sigma_6} > A(k)$, then R is either a balanced or a caterpillar tree. The equivalence class with the second smallest probability values, for both caterpillar and balanced

trees, is unique and has size 2 for caterpillar trees and size 4 for balanced trees. The first step of QR-STAR determines the tree shape as balanced if condition $\hat{u}_{\sigma_9} - \hat{u}_{\sigma_8} < A(k)$ holds and else it would determine the tree shape as caterpillar. Similar to the explanation above for the case of pseudo-caterpillar trees, when $\hat{u}_{\sigma_9} - \hat{u}_{\sigma_8} < A(k)$, \hat{u}_{σ_8} and \hat{u}_{σ_9} must belong to the same equivalence class and this only happens for balanced trees. Moreover, when R is balanced, $\hat{u}_{\sigma_9} - \hat{u}_{\sigma_8} < A(k)$. Therefore, conditions $\hat{u}_{\sigma_7} - \hat{u}_{\sigma_6} > A(k)$ and $\hat{u}_{\sigma_9} - \hat{u}_{\sigma_8} < A(k)$ hold if and only if R is a balanced tree, so that QR-STAR correctly determines the tree shape in this case as well.

- Finally, when R is a caterpillar tree, \hat{u}_{σ_8} and \hat{u}_{σ_9} belong to different equivalence classes; therefore, $\hat{u}_{\sigma_9} - \hat{u}_{\sigma_8} > A(k)$, and the other side can be shown similarly. Therefore, by comparing $\hat{u}_{\sigma_7} - \hat{u}_{\sigma_6}$ and $\hat{u}_{\sigma_9} - \hat{u}_{\sigma_8}$ against $A(k)$ when k is large enough so that $A(k) < \frac{1}{2}h(R)$, QR-STAR will correctly determine the rooted shape of the model tree with probability at least $1 - \delta$.

The argument is summarized below:

$$\begin{aligned}
 \text{Pseudo-caterpillar: } & \hat{u}_{\sigma_1} \leq \hat{u}_{\sigma_2} \leq \cdots \leq \underbrace{\hat{u}_{\sigma_6} \leq \hat{u}_{\sigma_7}}_{\hat{u}_{\sigma_7} - \hat{u}_{\sigma_6} < A(k)} \leq \underbrace{\hat{u}_{\sigma_8} < \hat{u}_{\sigma_9}}_{\hat{u}_{\sigma_9} - \hat{u}_{\sigma_8} > A(k)} \leq \hat{u}_{\sigma_{10}} \leq \cdots \leq \hat{u}_{\sigma_{15}} \\
 \text{Balanced: } & \hat{u}_{\sigma_1} \leq \hat{u}_{\sigma_2} \leq \cdots \leq \underbrace{\hat{u}_{\sigma_6} < \hat{u}_{\sigma_7}}_{\hat{u}_{\sigma_7} - \hat{u}_{\sigma_6} > A(k)} \leq \underbrace{\hat{u}_{\sigma_8} \leq \hat{u}_{\sigma_9}}_{\hat{u}_{\sigma_9} - \hat{u}_{\sigma_8} < A(k)} \leq \hat{u}_{\sigma_{10}} \leq \cdots \leq \hat{u}_{\sigma_{15}} \\
 \text{Caterpillar: } & \hat{u}_{\sigma_1} \leq \hat{u}_{\sigma_2} \leq \cdots \leq \underbrace{\hat{u}_{\sigma_6} < \hat{u}_{\sigma_7}}_{\hat{u}_{\sigma_7} - \hat{u}_{\sigma_6} > A(k)} \leq \underbrace{\hat{u}_{\sigma_8} < \hat{u}_{\sigma_9}}_{\hat{u}_{\sigma_9} - \hat{u}_{\sigma_8} > A(k)} \leq \hat{u}_{\sigma_{10}} \leq \cdots \leq \hat{u}_{\sigma_{15}}
 \end{aligned} \tag{4.35}$$

QED.

Lemma 4.6 (Upper bound on the cost of the model tree). Let R be a 5-taxon model species tree and \vec{u} be the probability distribution that it defines on the unrooted 5-taxon gene tree topologies. There is an integer $k > 0$ such that if we are given at least k unrooted gene trees drawn i.i.d. from distribution \vec{u} , then $\text{Cost}^*(R, \vec{u})$ is less than $31\alpha_{\max}A(k)$ with probability at least $1 - \delta$.

Proof. Let \mathcal{E} be the event that $|\hat{u}_i - u_i| < \frac{A(k)}{2}$ for all $1 \leq i \leq 15$. According to Lemma 4.4, the probability that \mathcal{E} holds is at least $1 - \delta$. When \mathcal{E} holds, according to Lemma 4.2, the following inequality is true for \vec{u} and the path length parameter $h(R)$ of the model tree R :

$$\forall_{c>c' \in C_R} \forall_{u_a \in c, u_b \in c'} : \hat{u}_a - \hat{u}_b > h(R) - A(k). \tag{4.36}$$

When k is sufficiently large that $A(k) < \frac{1}{2}h(R)$ (which is possible under our assumption that $\lim_{k \rightarrow \infty} A(k) = 0$), then $\hat{u}_a - \hat{u}_b$ will be positive. Therefore, all inequality penalty terms which are defined as $\max(0, \hat{u}_b - \hat{u}_a)$ in $\text{Cost}^*(R, \vec{\hat{u}})$ become zero, since $\hat{u}_b - \hat{u}_a$ is a negative term. Therefore, the total sum of the inequality penalty terms in $\text{Cost}^*(R, \vec{\hat{u}})$ will be zero for large enough k . Moreover, Lemma 4.5 states that the topological shape of R can be correctly determined when $A(k) < \frac{1}{2}h(R)$ and \mathcal{E} holds, and hence the shape penalty term $\mathbb{1}|S(R) \neq \hat{S}(\hat{u})|$ also becomes zero. Therefore, all elements of $\text{Cost}^*(R, \vec{\hat{u}})$ except the invariants penalty terms become zero.

According to Eq. 4.34, for each invariant penalty term, we have $|\hat{u}_a - \hat{u}_b| < A(k)$. Hence,

$$\begin{aligned} \text{Cost}^*(R, \vec{\hat{u}}) &= \underbrace{\sum_{c \in C_R} \sum_{u_a, u_b \in c} \alpha_{a,b} |\hat{u}_a - \hat{u}_b|}_{\text{Invariants Penalty}} < \alpha_{\max} \sum_{c \in C_R} \sum_{u_a, u_b \in c} A(k) \\ &= \alpha_{\max} A(k) \sum_{c \in C_R} \binom{|c|}{2} \leq 31\alpha_{\max} A(k). \end{aligned} \quad (4.37)$$

The last inequality holds since (i) caterpillar trees have 7 equivalence classes with class sizes 1,1,1,2,2,2,6 and therefore $\sum_{c \in C_R} \binom{|c|}{2} = 18$, (ii) balanced trees have 5 equivalence classes with sizes 1,2,2,4,6 and $\sum_{c \in C_R} \binom{|c|}{2} = 23$, and (iii) pseudo-caterpillar trees have 5 equivalence classes with sizes 1,2,2,2,8 and $\sum_{c \in C_R} \binom{|c|}{2} = 31$. Hence, in all cases we have $\sum_{c \in C_R} \binom{|c|}{2} \leq 31$ and the inequality follows. QED.

Theorem 4.1 (Statistical Consistency of QR-STAR for 5-taxon trees). Let R be a rooted 5-taxon model [species tree](#) and \vec{u} be the distribution that it defines on the [unrooted](#) 5-taxon gene tree topologies. Given a set \mathcal{G} of [unrooted](#) true quintet gene trees drawn i.i.d. from \vec{u} , QR-STAR is a [statistically consistent](#) estimator of R under the [MSC](#).

Proof. We will show that we can find k large enough so that QR-STAR will correctly return the rooted version of R with probability at least $1 - \delta$ when given at least k true gene trees. Hence, QR-STAR is [statistically consistent](#) for rooting R , since $\lim_{k \rightarrow \infty} \delta = 0$ by assumption.

According to Lemma 4.6, when k is large enough so that $A(k) < \frac{1}{2}h(R)$, if $\vec{\hat{u}}$ is the distribution estimated from \mathcal{G} , then $\text{Cost}^*(R, \vec{\hat{u}})$ is at most $31\alpha_{\max}A(k)$ with probability at least $1 - \delta$. We now prove that for every other rooted 5-taxon tree R' , $\text{Cost}^*(R', \vec{\hat{u}})$ is bounded away from zero. Note that according to Lemma 4.3, for every rooted 5-taxon tree $R' \neq R$ with the same rooted shape as R , we have $V(R, R') \neq \emptyset$ and therefore, there exists $1 \leq x \neq y \leq 15$ such that $(x, y) \in I_R, (y, x) \in I_{R'}$.

Let \mathcal{E} be the event that $|\hat{u}_i - u_i| < \frac{A(k)}{2}$ for all $1 \leq i \leq 15$. According to Lemma 4.2, when \mathcal{E} holds, then $\hat{u}_x - \hat{u}_y > h(R) - A(k)$ as $(x, y) \in I_R$. However, since $(y, x) \in I_{R'}$, an

inequality penalty term in the form of $\max(0, \hat{u}_x - \hat{u}_y)$ is added to $\text{Cost}^*(R', \vec{\hat{u}})$ when R' has the same shape as R . Moreover, according to Lemma 4.5, when \mathcal{E} holds and $A(k) < \frac{1}{2}h(R)$, the first step of QR-STAR correctly determines the rooted shape of R . Therefore, if R' has a different rooted shape than R , then the penalty $\mathbb{1}|S(R') \neq \hat{S}(\hat{u})|$ becomes 1 and a positive cost C is added to the cost of R' . Therefore, both cases (i.e. whether R' has a different **topology** from R or not) lead to a positive penalty in the cost function for R' that is bounded away from zero. Hence,

$$\begin{aligned} \text{Cost}^*(R', \vec{\hat{u}}) &= \underbrace{\sum_{c \in C_{R'}} \sum_{u_a, u_b \in c} \alpha_{a,b} |\hat{u}_a - \hat{u}_b|}_{\text{Invariants Penalty}} + \underbrace{\sum_{c > c' \in C_{R'}} \sum_{u_a \in c, u_b \in c'} \beta_{a,b} \max(0, \hat{u}_b - \hat{u}_a)}_{\text{Inequalities Penalty}} + \underbrace{C \mathbb{1}|S(R') \neq \hat{S}(\hat{u})|}_{\text{Shape Penalty}} \\ &\Rightarrow \forall_{R' \neq R} \text{Cost}^*(R', \vec{\hat{u}}) \geq \min(\beta_{\min}(\hat{u}_x - \hat{u}_y), C) > \min(\beta_{\min}(h(R) - A(k)), C) \end{aligned} \quad (4.38)$$

Eq. 4.38 defines a lower bound for the cost of any tree other than the true rooted tree and Lemma 4.6 gives an upper bound for the cost of the true tree, both with respect to the estimated quintet distribution $\vec{\hat{u}}$. Therefore, when k is large enough so that

$$A(k) < \min\left(\frac{C}{31\alpha_{\max}}, \frac{h(R)}{31\frac{\alpha_{\max}}{\beta_{\min}} + 1}, \frac{1}{2}h(R)\right) \quad (4.39)$$

(which, once again, is possible under our assumption that $\lim_{k \rightarrow \infty} A(k) = 0$), we will have

$$\text{Cost}^*(R, \vec{\hat{u}}) < 31\alpha_{\max}A(k) < \min(h(R) - A(k), C) < \text{Cost}^*(R', \vec{\hat{u}}) \quad (4.40)$$

which means that the cost of the true rooted tree will be less than the cost of any other rooted tree on the same leafset with probability at least $1 - \delta$. Precisely, $\forall_{R' \neq R} \text{Cost}^*(R, \vec{\hat{u}}) < \text{Cost}^*(R', \vec{\hat{u}})$ when Eq. 4.39 holds, where $\beta_{\min}, C, h(R) > 0$ and $\alpha_{\max} \geq 0$ are constants. As a result, QR-STAR will return the true rooted **species tree topology** with probability converging to 1 as the number of gene trees grows large, proving the statistical consistency for 5-leaf trees. QED.

Remark 4.1. Note that when $\alpha_{\max} = 0$, meaning that the invariant penalty terms are removed from the cost function, the cost of the true tree is exactly zero according to the proof of Lemma 4.6, and the cost of any other tree is positive when k is large enough. Hence in this case, the condition in Eq. 4.39 reduces to $A(k) < \frac{1}{2}h(R)$.

Remark 4.2. Note that Lemma 4.3(a) holds for *all* pairs of 5-taxon rooted trees with the same rooted shape and with different permutations of the leaf-labeling, regardless of

whether they have the same (leaf-labeled) [unrooted topology](#) or not. Due to this property, it is possible to differentiate all pairs of 5-taxon rooted trees in a [statistically consistent](#) manner with the cost function of QR-STAR without prior knowledge about the [unrooted](#) tree topology, and hence Theorem 4.1 does not need to assume that the [unrooted topology](#) is given as input.

4.4.3 Extending to Larger Trees

The next lemma and theorem extend the proof of statistical consistency to trees with $n > 5$ taxa. Recall that linear encodings of T were defined in Section 4.2.3.

Lemma 4.7 (Identifiability of the root from the linear encoding). Let R and R' be rooted trees with [unrooted topology](#) T and distinct roots. Let $Q_{LE}(T)$ be the set of quintets of leaves in a linear encoding of T . There is at least one quintet of [taxa](#) $q \in Q_{LE}(T)$ so that $R|_q$ and $R'|_q$ have different rooted topologies.

Proof. Let e be the edge in T corresponding to the root of R . Let $q(e)$ be the quintet of leaves corresponding to edge e in $Q_{LE}(T)$. It is clear that $R|_{q(e)}$ is also rooted at edge e . The following cases can happen:

- When edge e is not incident to a leaf, it partitions the set of leaves of T into four subsets. Let A_1, A_2, B_1 and B_2 be the subsets resulting from deleting edge e from T , where A_1 and A_2 are incident to one endpoint of e and B_1 and B_2 are incident to the other. According to the definition of quintets in the linear encoding, $q(e)$ must have at least one leaf in each subset, which we call a_1, a_2, b_1 , and b_2 respectively. For every other rooted tree R' with [topology](#) T , the root edge e' of R' will fall into one of the four subsets A_1, A_2, B_1, B_2 , including the edges sharing an endpoint with e . W.L.O.G. assume that e' falls into A_1 . Then when T is rooted at edge e' (resulting in R'), the leaves a_2, b_1 and b_2 in $q(e)$ fall into one side of the root and a_1 falls into another. Therefore, the leaves a_1, a_2 and b_1 form the rooted [triplet](#) $((a_2, b_1), a_1)$. However, when T is rooted at e (producing R), these leaves form the rooted [triplet](#) $((a_1, a_2), b_1)$, as edge e separates a_1 and a_2 from b_1 . Hence, in this case $R|_{q(e)}$ and $R'|_{q(e)}$ are topologically different because they induce different rooted [triplets](#).
- When edge e is adjacent to a leaf x , it partitions the set of [taxa](#) into three subsets, where one of them contains the single node $\{x\}$. Let A, B be the two other subsets resulting from deleting the edge e , where e separates x from the sets A and B . According to

the definition of linear encoding, $q(e)$ must contain x and at least one leaf in A and B which we call a and b respectively. For every other rooted tree R' with topology T , the root edge e' of R' will fall into A or B (including the edges directly adjacent to e). W.L.O.G. assume that it falls in A . Then when T is rooted at e (producing R), the nodes a, b, x form the rooted triplet $((a, b), x)$, but when T is rooted at e' (producing R'), they form the rooted triplet $((b, x), a)$. Therefore, this case also leads to different rooted topologies for $R|_{q(e)}$ and $R'|_{q(e)}$.

Therefore, in both cases, R and R' restricted to the leaves in quintet $q(e)$ produce topologically different rooted quintet trees, completing the proof. QED.

Lemma 4.7 states that no two distinct rooted trees with topology T induce the same set of rooted quintet trees on quintets of **taxa** in a linear encoding $Q_{LE}(T)$. Clearly, the same is true for any superset Q such that $Q_{LE}(T) \subseteq Q$, including the set Q_5 of all quintets of **taxa** on the leafset of T . There are also other quintet sets that are not a superset of $Q_{LE}(T)$, but have the property that no two rooted versions of T define the same set of rooted quintets on their elements. We generalize the proof of consistency to all sets of sampled quintets with this property.

Definition 4.4. Let T be an **unrooted** tree and Q be a set of quintets of **taxa** from $\mathcal{L}(T)$. We say Q is “root-identifying” if every rooted tree R with topology T is identifiable from T and the set of rooted quintet trees in $\{R|_q : q \in Q\}$, i.e., no two rooted trees with topology T induce the same set of rooted quintet trees on Q .

Theorem 4.2 (Statistical Consistency of QR-STAR). Let R be an **MSC** model **species** tree with $n \geq 5$ leaves and let T denote its **unrooted** topology. Given T and a set \mathcal{G} of **unrooted** true gene trees on the leafset $\mathcal{L}(T)$, QR-STAR is a **statistically consistent** estimator of the rooted version of T under the **MSC**, if the set of sampled quintets Q is root-identifying.

Proof. QR-STAR computes the score of each rooted tree R with topology T as the sum $\sum_{q \in Q} \text{Cost}^*(R|_q, \hat{\vec{u}}_q)$. Let \mathcal{E}_q be the event that $|(\hat{u}_q)_i - (u_q)_i| < \frac{A_{Q,\delta}(k)}{2}$ for all $1 \leq i \leq 15$ for a quintet $q \in Q$. Arguing as in the proof of Theorem 4.1, when \mathcal{E}_q holds and k is large enough so that $A_{Q,\delta}(k) < \min\left(\frac{C}{31\alpha_{\max}}, \frac{h(R)}{31\frac{\alpha_{\max}}{\beta_{\min}} + 1}, \frac{1}{2}h(R)\right)$ (that guarantees $A_{Q,\delta}(k) < \min\left(\frac{C}{31\alpha_{\max}}, \frac{h(R|_q)}{31\frac{\alpha_{\max}}{\beta_{\min}} + 1}, \frac{1}{2}h(R|_q)\right)$ according to Lemma 4.1), $\text{Cost}(R|_q, \hat{\vec{u}}_q)$ will be less than the cost of each of the six alternative rooted trees on the five **taxa** in q with high probability, as $R|_q$ is the *true* rooting of the **unrooted** quintet tree $T|_q$.

According to Lemma 4.4, \mathcal{E}_q simultaneously holds for all $q \in Q$ with probability at least $1 - \delta$. In this event, for every other rooted tree $R' \neq R$ and each $q \in Q$, we have $\text{Cost}^*(R'|_q, \hat{\vec{u}}_q) \geq$

$\text{Cost}^*(R|_q, \hat{\vec{u}}_q)$, according to Theorem 4.1. This means that for every rooted tree R' with topology T , $\sum_{q \in Q} \text{Cost}^*(R'|_q, \hat{\vec{u}}_q) \geq \sum_{q \in Q} \text{Cost}^*(R|_q, \hat{\vec{u}}_q)$. Also, according to Definition 4.4, Q has the property that no other rooted tree R' induces the exact same set of rooted quintet trees as R on Q . Hence, there exists $q^* \in Q$ such that $\text{Cost}^*(R'|_{q^*}, \hat{\vec{u}}_{q^*}) > \text{Cost}^*(R|_{q^*}, \hat{\vec{u}}_{q^*})$, as the cost of $R|_{q^*}$ is strictly less than the cost of each of the six alternative rooted trees on q^* when the conditions in Theorem 4.1 hold. Therefore, the function $\sum_{q \in Q} \text{Cost}^*(r|_q, \hat{\vec{u}}_q)$ obtains its *unique* minimum for the rooted tree R . As a result, QR-STAR returns the true rooted topology of T with probability converging to 1 as the number of input gene trees increases, establishing the statistical consistency given trees with an arbitrary number of taxa.

QED.

4.4.4 Sample Complexity of QR-STAR

Having established statistical consistency, we now discuss **sample complexity** – i.e., the number of genes that suffice for QR-STAR to correctly root the model **species tree** with probability at least $1 - \delta$, for an arbitrary $\delta > 0$.

Theorem 4.3 (Sample Complexity of QR-STAR). Let R be an **MSC** model species tree with $n \geq 5$ leaves and let T denote its **unrooted** topology. Given T, Q (a root-identifying set of sampled quintet trees), $\delta > 0$, and k true **unrooted** gene trees on the leafset of T , QR-STAR returns the true tree R with probability at least $1 - \delta$, when the number of gene trees satisfies

$$k > 2 \times 36^2 \ln \left(\frac{30|Q|}{\delta} \right) \frac{e^{6g}}{(1 - e^{-f})^4} \quad (4.41)$$

where f and g are the lengths of the shortest **internal** branch and the longest **internal** path in R respectively. When the linear encoding is used so that $|Q| = 2n - 3$ and in the limit of small f , QR-STAR returns the true **rooted** tree with probability at least $1 - \delta$ when the number k of gene trees satisfies

$$k = \Omega(f^{-4}e^{6g}(\ln(n) - \ln(\delta))). \quad (4.42)$$

Proof. According to the proof of Theorem 4.2, when k is large enough so that $A_{Q,\delta}(k) < \min \left(\frac{C}{31\alpha_{\max}}, \frac{h(R)}{31\frac{\alpha_{\max}}{\beta_{\min}} + 1}, \frac{1}{2}h(R) \right)$, the function $\sum_{q \in Q} \text{Cost}^*(r|_q, \hat{\vec{u}}_q)$ will be minimized for **rooted** tree R and QR-STAR will return the true tree with probability at least $1 - \delta$. For simplicity, we consider the case where the weights α_i, β_i, C in the cost function of QR-STAR are set such

that $\min\left(\frac{C}{31\alpha_{\max}}, \frac{h(R)}{31\frac{\alpha_{\max}}{\beta_{\min}}+1}, \frac{1}{2}h(R)\right)$ reduces to $\frac{1}{2}h(R)$ (also see Remark 4.1). Substituting the definitions of $A_{Q,\delta}(k)$ and $h(R)$, we get

$$\begin{aligned} A_{Q,\delta}(k) < \frac{1}{2}h(R) &\iff \\ \sqrt{\frac{2}{k}\ln\left(\frac{30|Q|}{\delta}\right)} < \frac{1}{36}e^{-3g}(1-e^{-f})^2 &\iff \\ \frac{2}{k}\ln\left(\frac{30|Q|}{\delta}\right) < \left(\frac{1}{36}\right)^2 e^{-6g}(1-e^{-f})^4 &\iff \\ k > 2 \times (36)^2 \ln\left(\frac{30|Q|}{\delta}\right) \frac{e^{6g}}{(1-e^{-f})^4} \end{aligned} \tag{4.43}$$

When the linear encoding is used so that $|Q| = 2n - 3$ and in the limit of small f , we have $\lim_{f \rightarrow 0} \frac{1-e^{-f}}{f} = 1$, and hence QR-STAR returns the true **rooted** tree with probability at least $1 - \delta$ when the number of gene trees satisfy

$$k = \Omega(f^{-4}e^{6g}(\ln(n) - \ln(\delta))). \tag{4.44}$$

QED.

Theorem 4.3 yields a **sample complexity** that is exponential in g , the length of the longest **internal** path in R . However, an improved **sample complexity** can be obtained through a more nuanced analysis as well as using a modified version of the linear encoding, as we now show.

Definition 4.5 (Q -Restricted path length parameter). Let R be an **MSC** model species tree. Let Q be a set of quintets of **taxa** from $\mathcal{L}(R)$, and let $R|_Q = \{R|_q : q \in Q\}$ be the corresponding set of **rooted** quintet trees. Let f_Q be the length of the shortest **internal** branch in any (rooted) quintet tree in $R|_Q$ and g_Q be the length of the longest **internal** path of any quintet tree in $R|_Q$. We define the path length parameter of R restricted to Q as

$$h_Q(R) = \frac{1}{18}e^{-3g_Q}(1-e^{-f_Q})^2 \tag{4.45}$$

Note that $f_Q \geq f(R)$ and $g_Q \leq g(R)$, where $f(R)$ and $g(R)$ are defined in Definition 4.1. However, if Q is the set of all quintets of **taxa** on the leafset of R , then $f_Q = f(R)$ and $g_Q = g(R)$, so that the Q -restricted path length parameter of R is identical to the path

length parameter of R , as given in Definition 4.1. Finally, note that we can replace $h(R)$ by $h_Q(R)$, without any change in the theoretical results.

We now define a variant of the linear encoding, which we refer to as a “short quintet encoding” of the tree. This variant is motivated by the concept of “short quartets” (which are quartets of leaves sampled around each **internal** edge in a tree so that they are the closest leaves to that edge) and the strong theoretical properties of the “short quartet methods”, which estimate **phylogenetic trees** from aligned sequences by first estimating quartet trees that seem likely to be short quartets and then combining the quartet trees into a tree on the full dataset [268, 269, 270, 271]. As proven in [268, 269], even simple versions of these methods have provably polynomial **sample complexity** under standard models of sequence evolution down trees (e.g., the **Generalized Time Reversible (GTR)** model [96]), after bounding (arbitrarily) the length of the shortest and longest **internal** edges in the model tree.

Definition 4.6 (Short Quintet Encoding). Let T be an **unrooted** tree and e an **internal** edge in T , so that deleting e and its endpoints produces four subtrees. A short quintet around edge e contains a nearest leaf (in topological distance) in each of the four subtrees around e , and one other leaf that is chosen so that it is either tied for nearest in its subtree or second nearest within its subtree. For the case where e is incident with a leaf x , removing e and its endpoints splits the tree into two subtrees, A and B . A short quintet around e will include x and then four other leaves. If each of A and B has at least two leaves, then we pick the two nearest leaves in each of them. Otherwise we pick the single leaf from one subtree and the three nearest leaves from the other subtree. We modify the linear encoding algorithm to ensure that each of the sampled quintets is a short quintet, and we refer to this as a Short Quintet Encoding of T .

Note that there can be more than one short quintet encoding of a tree, and that every short quintet encoding is root-identifying (since each is a linear encoding).

Theorem 4.4. Let R be an **MSC** model species tree with $n \geq 5$ leaves with **unrooted topology** T , let $f(R)$ be the length of the shortest **internal** edge in R , and let z_T be the length of the longest **internal** edge in the **unrooted topology** T for R . Given $\delta > 0$, k true **unrooted** gene trees on the leafset of T , short quintet encoding Q of T , and tree T , in the limit of small f with probability at least $1 - \delta$, QR-STAR returns the correct rooting of T (i.e., true tree R), when the number k of gene trees satisfies

$$k = \Omega\left(\frac{n^{O(z_T)}(\ln(n) - \ln(\delta))}{f(R)^4}\right). \quad (4.46)$$

Proof. Recall that for any set Q of quintets, $f_Q \geq f(R)$. Note also that any short quintet encoding is root-identifying since this is just a special case of a linear encoding.

By arguments similar to the proof provided for Theorem 4.3 and by substituting $h(R)$ with $h_Q(R)$, in the limit of small $f(R)$, QR-STAR returns R with probability at least $1 - \delta$ if

$$k = \Omega(f(R)^{-4} e^{6g_Q} (\ln(n) - \ln(\delta))). \quad (4.47)$$

Note also that g_Q (the length of the longest internal path of the quintet trees using the short quintets in Q) is $O(z_T \log n)$, since the topological diameter within T of any short quintet is $O(\log n)$ (based on the same arguments that short quartets have topological diameters that are $O(\log n)$ [268]). Hence, $e^{6g_Q} = e^{O(z_T \log n)} = O(n^{O(z_T)})$. The result follows. QED.

This means that QR-STAR has a polynomial sample complexity when we fix $f(R)$ and z_T , the length of the shortest internal edge in R and longest internal edge in T , respectively.

4.4.5 Computing the optimal rooting score

To conclude this section, we present theoretical results for computing the optimal rooting score of an estimated species tree that will be used in the experimental study. In Experiments 2 and 3, we report the lowest possible normalized clade distance (nCD) rate achievable across all rootings of the estimated species tree; this is performed through an exhaustive search (rooting on all possible edges and computing the nCD rate). Here we show that this best possible nCD has a close relationship to the missing branch (FN) rate of the unrooted estimated species tree with respect to the model species tree.

Let R be the true rooted species tree and let T denote its unrooted topology. Let \hat{T} be an estimate of T . We are interested in rooting \hat{T} to minimize the number of its missing clades with respect to R , and we will call this the missing clade number (not a rate). We define $FN(\hat{T}, T)$ to be the number of bipartitions in T that are not found in \hat{T} . We will show that the missing clade number for an optimal rooting of \hat{T} with respect to R is either $FN(\hat{T}, T)$ or $FN(\hat{T}, T) + 1$, depending on whether the bipartition at the root of R is present in \hat{T} or not.

Lemma 4.8. Let R be the true rooted species tree with T the unrooted version. We draw R with root r having two children v_A and v_B . Let A be the clade below v_A and B be the clade below v_B . Let \hat{T} be an estimate of T . If bipartition $A|B$ is present in \hat{T} , induced by edge e , then the optimal rooting of \hat{T} that minimizes the number of missing clades in R is achieved by rooting on edge e , and results in $FN(\hat{T}, T)$ missing clades. If $A|B$ is not present

in \hat{T} , then the optimal rooting of \hat{T} that minimizes the number of **clades** in R missing from the **rooted** version of \hat{T} results in $FN(\hat{T}, T) + 1$ missing clades.

Proof. Assume $A|B$ is not present in \hat{T} . Take R and mark all edges of T that define **bipartitions** that do not appear in \hat{T} . Since the bipartition that is defined by the edge that the root bisects is not present in \hat{T} , those two edges incident to r (i.e., the edges (r, v_A) and (r, v_B)) are both marked. Make r a leaf, by attaching a new leaf that is adjacent to the original root location. Collapse all marked edges, to obtain an **unresolved** tree. Now refine this **unresolved** tree so that it induces \hat{T} (and so produces \hat{T} when the root leaf is removed). Since this tree contains the root, this can be seen as a **rooted** version of \hat{T} . Note that the missing clade number for this **rooted** tree is identical to $FN(\hat{T}, T) + 1$. That is, the *two* edges that define the bipartition $A|B$ in T each contribute 1 to the missing clade number, and every other missing edge also contributes 1 to the missing clade number. The proof for when $A|B$ is present in \hat{T} is similar and is omitted. QED.

4.5 EXPERIMENTAL STUDY

4.5.1 Overview

We performed four experiments in this study. Experiment 0 was used for the design of QR-STAR, where we used a training dataset with 101-taxon species trees to set the numeric parameters in its cost function. Experiments 1–3 are on test datasets, which are separate from the training data. Experiments 1 and 2 examine rooting of the true or estimated species trees, respectively, on a dataset with 201-taxon trees generated using SimPhy [272] under different model conditions. Experiment 3 examines rooting of estimated species trees on two simulated datasets with model trees resembling real biological datasets (a 48-taxon avian species tree from [247] and a 37-taxon mammalian tree from [250]). Overall, the model conditions in the test datasets vary in terms of the number of taxa, number of genes, **gene tree** estimation error, level of **ILS**, and topological shape of the species tree.

For each model condition (both in training and in test datasets), we report the level of **ILS** using the average normalized **RF** (i.e., Robinson-Foulds [83]) distance between the model species tree and true gene trees, and denote this value by **AD**, or average distance. We also report the average **gene tree estimation error (GTEE)** using normalized **RF** distance between true and estimated gene trees. We evaluated rooting error using **normalized clade distance (nCD)** [230], which is a **rooted** version of the normalized **RF** distance. For the training experiments, we also report the proportion of the trees that are correctly rooted.

For the training dataset (101-taxon dataset from [51]), the **ILS** level (measured using the average distance between the true gene trees and the model species tree, or **AD**) for most replicates ranged from 0.3 to 0.6 with an average of 0.46. The mean **GTEE** values for the four sequence lengths was 0.23, 0.31, 0.42 and 0.55 for the 1600bp, 800bp, 400bp and 200bp sequences respectively. The speciation rate for this dataset was 1e-07.

For the training experiment, we only **rooted** the true species tree **topology** to directly observe the rooting error. In the test experiments, we **rooted** both the model species tree and estimated species tree, as produced by ASTRAL, using both true and estimated gene trees. Throughout these experiments, we set $\delta = \frac{1}{k}$ in QR-STAR, where k is the number of gene trees in the input. All datasets, along with the estimated gene trees, are from prior studies [50, 51, 189] and are available online.

4.5.2 Designing QR-STAR

We used the 101-taxon simulated datasets from [51] as our training data, which had model conditions characterized by four **GTEE** levels, ranging from 0.23 to 0.55 for 1000 genes. The normalized **RF** distance between the model species tree and true gene trees (denoted average distance, or **AD**) in this dataset was 0.46, which indicates moderate **ILS**.

We explored a range of values for the shape coefficient (parameter C) and the relative weight of inequalities and invariants (the ratio $\frac{\alpha_{max}}{\beta_{min}}$) in the cost function of QR-STAR on the training dataset. When $\alpha_{max} > 0$, these two values can impact the **sample complexity** of QR-STAR as Eq. 4.39 suggests. We report the proportion of the trees (from the 50 replicates in each condition) that are correctly rooted, as well as the rooting error (nCD values) for rooting the true species tree topology.

Figure 4.5 shows the impact of shape coefficient on the accuracy of QR-STAR, where the weights of invariant and inequality penalty terms are fixed to the weights in the original cost function of QR. For small C values (i.e. less than 1E-02), the accuracy of QR-STAR does not seem to be affected by the shape coefficient, but as C gets larger, the accuracy degrades until it reaches a stationary point again. This suggest that the shape coefficient should be kept relatively small compared to the invariant and inequality penalty weights, as they may better capture the difference between two **rooted** quintet trees. Since Eq. 4.39 suggests that larger C values are theoretically preferred, on the experiments on the test dataset, we set the value of C as 1E-02 (the largest value before accuracy degrades).

Figure 4.6 shows the impact of the ratio $\frac{\alpha_{max}}{\beta_{min}}$ on QR-STAR. Here all α and β values are set as equal. The results suggest that when the inequalities are weighed more than the invariants (and so $\frac{\alpha_{max}}{\beta_{min}}$ is less than 1), QR-STAR has its optimal accuracy, and the

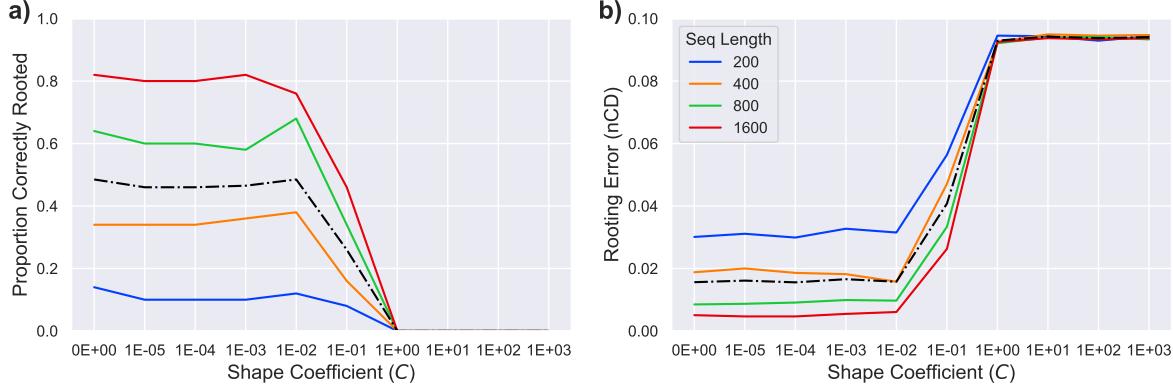


Figure 4.5: Impact of shape coefficient (C) on QR-STAR. a) Proportion of the trees correctly [rooted](#) and b) rooting error ([nCD](#)) are shown for the 101-taxon dataset from [51] averaged over 50 replicates. The number of genes is 1000 and the average [AD](#) level is 0.46. The sequence length used to produce estimated gene trees varies between 200bp to 1600bp. The black dashed line corresponds to the average among sequence lengths. The weights of invariant and inequality penalty terms are set as in the cost function of QR. The value of C varies between 0 and 10^3 , with $C = 0$ corresponding to the cost function of QR that does not guarantee consistency.

accuracy degrades when the invariants are weighed more. For both figures, the trends for different sequence lengths are similar, and the degradation in accuracy starts almost at the same point, but the accuracy is higher for longer sequence lengths, which is expected as shorter sequence lengths correspond to higher levels of [GTEE](#). In general, these experiments show that optimal accuracy could be achieved for a wide range of parameters in QR-STAR. For experiments on the test dataset, we set C as $1E-02$ and $\frac{\alpha_{max}}{\beta_{min}}$ as 0 (essentially removing invariants from the cost function), although we note that the optimal values could be dataset-dependant, and better training procedures might be needed to find robust parameter values that work well across different datasets.

4.5.3 Evaluating QR-STAR

Using the numeric parameters selected in Experiment 0, we compared QR-STAR to QR in two basic experiments on the test datasets. Experiment 1 compares QR and QR-STAR when rooting the true (model) species tree, given true or estimated gene trees, where the final error solely shows the rooting error. Experiment 2 compares these methods when rooting an estimated species tree produced by ASTRAL, given true or estimated gene trees, where the final error is a combination of species tree estimation and rooting error. For this second experiment, as the clade distance from the [rooted](#) version of the ASTRAL tree is a

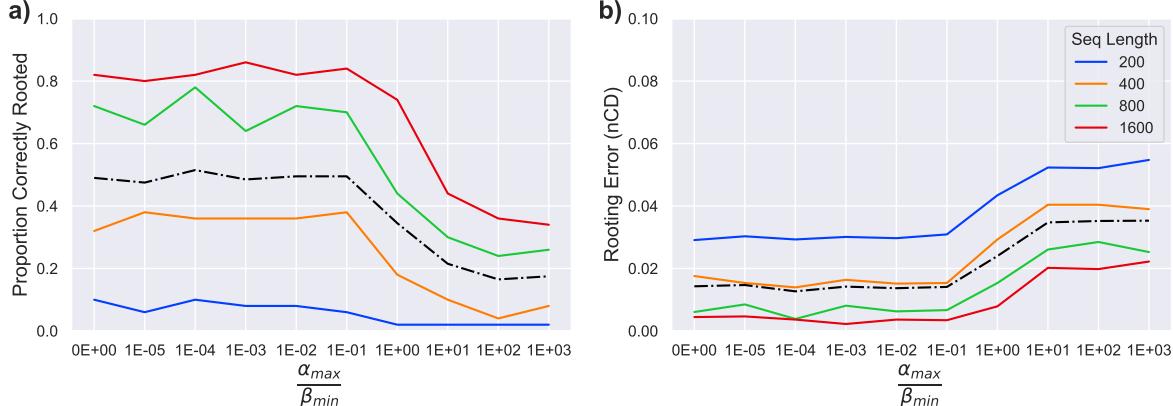


Figure 4.6: **Impact of $\frac{\alpha_{max}}{\beta_{min}}$ on QR-STAR.** a) Proportion of the trees correctly rooted and b) rooting error (nCD) are shown for the 101-taxon dataset from [51] averaged over 50 replicates. The number of genes is 1000 and the average AD level is 0.46. The sequence length used to produce estimated gene trees varies between 200bp to 1600bp. The black dashed line corresponds to the average among sequence lengths. The value of $\frac{\alpha_{max}}{\beta_{min}}$ varies between 10^{-5} to 10^3 in addition to 0. All α and β values are set as equal for all invariant or inequality penalty terms.

combination of RF distance between the estimated species tree and the true species tree as well as the error produced by the rooting method, we also report the optimal rooted species tree error, which is the lowest nCD error rate achieved across all possible rootings of the ASTRAL tree (see Appendix 4.4.5 for additional comments).

Datasets. We used a set of 201-taxon simulated datasets from [50] as our test data; these are characterized by two different speciation rates and three tree heights (thus six tree shapes), and three number of genes for each tree shape. The AD levels for this dataset for 1000 genes ranged from 0.09 (for the 10M, 1e-07 condition) to 0.69 (for the 500K, 1e-06 condition). The estimated gene trees were inferred using FastTree 2 [273]. The GTEE levels on the test data varied from 0.22 (for the 10M, 1e-06 condition) to 0.49 (for the 500K, 1e-06 condition). Table 4.1 summarizes these statistics. The number of replicates for each model condition in this dataset was 50.

We also performed experiments on the 48-taxon avian-like and 37-taxon mammalian-like simulated datasets from [189], which had model species trees based on biological datasets from [247] and [250], respectively. The default model condition in these datasets (shown with 1X ILS) had an ILS level that resembled the gene tree discordance in the corresponding biological data, but additional model conditions were created by multiplying or dividing branch lengths by two, thus decreasing or increasing the level of ILS, respectively (i.e., the

highest **ILS** level we test for each biological dataset is indicated by 0.5X). True gene trees were simulated within the model species trees under the **MSC**, and then sequences with varying lengths were evolved under each gene tree. Finally, RAxML [25] was used to estimate gene trees from these sequences alignments, creating conditions with varying **GTEE** levels. These datasets had 20 replicates in each model condition, but the model tree in all replicates was the same tree from the corresponding biological study. Tables 4.2 and 4.3 in the Appendix summarize the statistics for these two datasets.

Results for Experiment 1: Rooting the true species tree. Figure 4.7 (left) shows the result of rooting the model species tree with true gene trees on the test datasets. These results show that rooting error for both QR and QR-STAR decreases with the number of genes, as expected. We also see that rooting error is lowest for the highest **ILS** level (leftmost column), and increases as the **ILS** level decreases. The impact of speciation rate on rooting error is seen by comparing the top and bottom rows; this impact is small except for the lowest **ILS** case, where deep speciation (1e-07) generally leads to lower error than recent speciation (1e-06). A comparison of **nCD** error rates for QR and QR-STAR shows that the two methods are close in accuracy for some conditions (notably for high or moderately high **ILS** with a sufficient number of genes) but when there are differences, QR-STAR has lower rooting error. The advantage for QR-STAR over QR is largest for conditions with moderate to low **ILS**, and few genes. However, under the lowest **ILS** condition and with speciation rate 1E-06 (bottom right subfigure), there is a consistent advantage to QR-STAR across all numbers of genes.

Figure 4.7 (right) shows the same comparison with estimated gene trees. As with true gene trees, increasing the **ILS** level (by reducing tree height) decreases the rooting error, increasing the number of genes also generally reduces rooting error (although much less under the lowest **ILS** level where tree height is 10M), and changing the speciation rate has a small impact (even on the low **ILS** condition). A comparison between QR and QR-STAR shows that the relative accuracy depends on the **ILS** level. For the highest **ILS** condition (leftmost column), QR and QR-STAR are very close but with possibly a small advantage to QR. However, for moderate to low **ILS** conditions, QR-STAR matched or improved on QR.

Results for Experiment 2: Rooting an estimated species tree. Figures 4.8 show results on the test dataset, when rooting species trees estimated using ASTRAL with true or estimated gene trees. For all three methods (QR, QR-STAR, and optimal rooting), and using both true and estimated gene trees, increasing the number of genes improves accuracy, but increasing the **ILS** level reduces accuracy (in contrast to Experiment 1). We also see

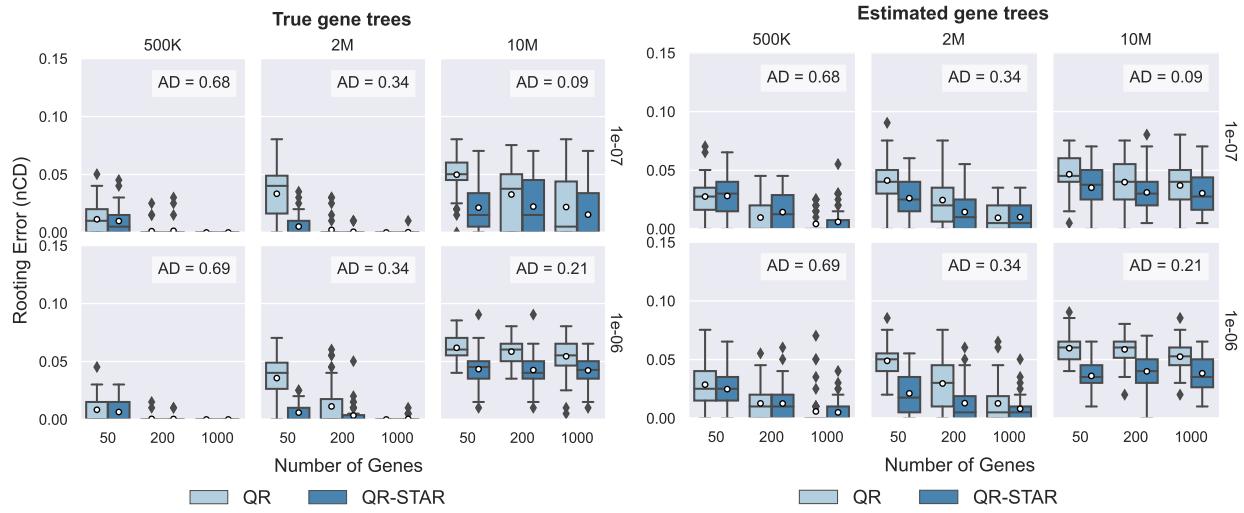


Figure 4.7: Rooting the model species tree on 201-taxon simulated datasets. Comparison between QR and QR-STAR in terms of rooting error (nCD) for rooting the true unrooted species tree topology using true or estimated gene trees on the 201-taxon datasets, with 50 replicates in each model condition. The columns show tree height (500K for high ILS, 2M for moderate ILS, and 10M for low ILS) and the rows show speciation rate (1e-06 for recent speciation, 1e-07 for deep speciation).

that under high ILS, the error in the rooted species tree is high (on average 14.6% when using only 50 true gene trees, and 21.4% when using 50 estimated gene trees), but decreases rapidly as the number of genes increases. Error is higher for speciation close to the leaves (1e-06) than for speciation closer to the root (1e-07), a pattern that was also observed when rooting the model species tree.

The trends relating QR-STAR and QR are interesting to discuss. When using true gene trees, the relative accuracy of QR and QR-STAR depends on the ILS level, with essentially identical error for the high ILS condition, but then an advantage to QR-STAR for the moderate or low ILS conditions (except when there is a sufficient number of genes). When used with estimated gene trees, the relative accuracy between QR and QR-STAR depends on the ILS level, but the gap between QR and QR-STAR is smaller. There is essentially no difference for the high ILS condition, a very small difference for the moderate ILS condition (but only if the number of genes is small), and a small difference for the low ILS condition that holds across both low and moderate numbers of genes. Thus, the trends for rooting ASTRAL species trees are somewhat different in terms of absolute rooting error (which increases, compared to rooting the true species tree), but the relative performance of QR-STAR and QR shows similar results as for rooting true species trees. The main difference is that the difference between the methods seems to have decreased.

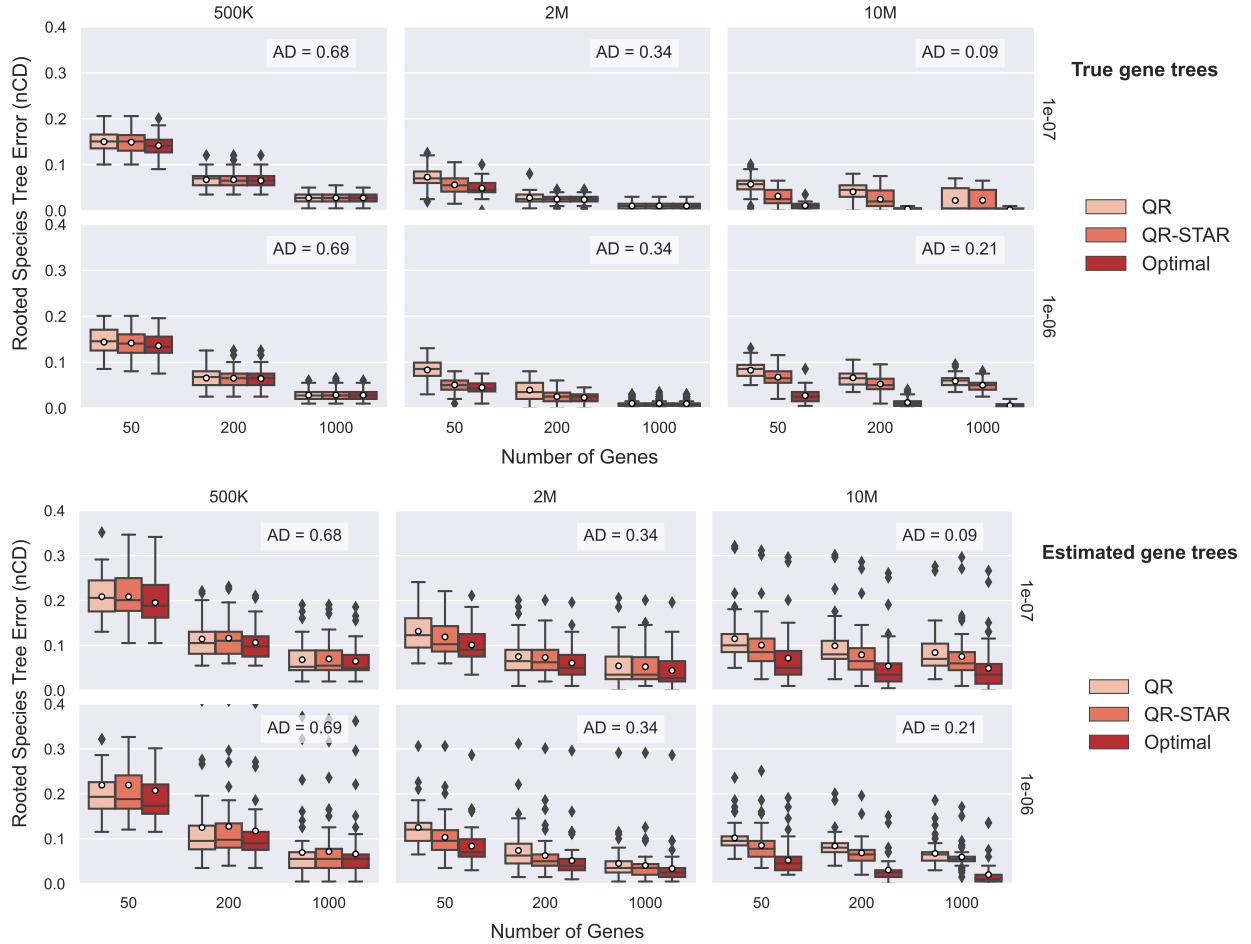


Figure 4.8: Rooting the ASTRAL species tree on 201-taxon datasets. Comparison between QR, QR-STAR and optimal rooting in terms of rooted species tree error (nCD) for rooting the species trees estimated by ASTRAL, using true or estimated gene trees on the 201-taxon datasets across 50 replicates. The columns show tree height (500K for high ILS, 2M for moderate ILS, and 10M for low ILS) and the rows show speciation rate (1e-06 for recent speciation, 1e-07 for deep speciation). The y-axes are cut at 0.4 to improve clarity, removing five outliers in the bottom figure from all methods in the 500K, 1e-06 model condition (see Fig. 4.14 for the full scale figure).

Finally, a comparison between QR-STAR and the optimal rooting provides some noteworthy trends. Specifically, for both true and estimated gene trees and under both high and moderate ILS, QR-STAR and optimal rooting are extremely close in terms of rooting error (with no detectable differences under high ILS and only a small difference under moderate ILS with only 50 genes). Thus, under these conditions, there is little room for improvement over QR-STAR. Interestingly, there is a bigger gap between QR-STAR and optimal rooting for low ILS than under the higher ILS conditions, especially when the speciation rate is 1e-06

(i.e., speciation towards the leaves). We also see that there is a slightly bigger gap between QR-STAR and the optimal rooting when QR-STAR is using estimated gene trees than when using true gene trees; as expected.

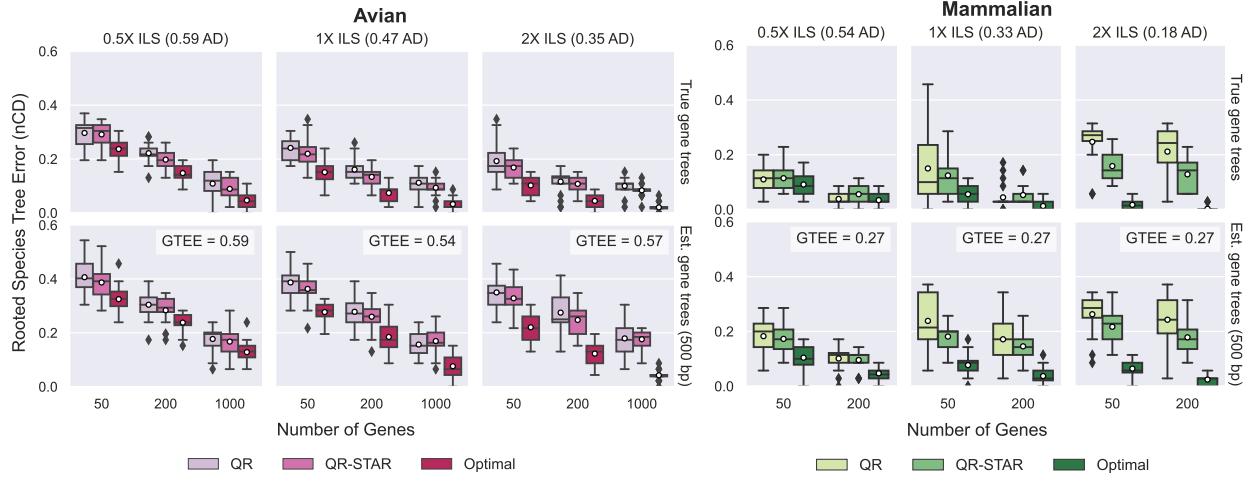


Figure 4.9: Rooting the ASTRAL species tree on biological simulations. Comparison between QR, QR-STAR and optimal rooting on (left) 48-taxon avian simulated datasets, (right) 37-taxon mammalian simulated datasets, both from [189]. The columns show the **ILS** level, and the rows show whether true or estimated gene trees (based on 500bp sequences) were used. For both datasets, the number of replicates in each model condition is 20, but the model species tree is fixed across all replicates.

Results for Experiment 3: Rooting an estimated species tree on biological model trees. Figure 4.9 shows results when using QR or QR-STAR to root the ASTRAL species trees on the avian or mammalian simulated datasets. Most trends are similar to the trends seen on 201-taxon datasets: accuracy with true gene trees is better than with estimated gene trees, and using more genes improve the results, as expected. The accuracy advantage of QR-STAR over QR can be seen in these two datasets as well, especially with true gene trees or low **ILS**.

However, unlike the 201-taxon datasets where the accuracy of QR-STAR was close to optimal in most cases, here we see a bigger gap between the optimal rooting and QR-STAR in general. On the mammalian simulated datasets, this gap is small under the highest **ILS** condition (0.54 **AD**) and the medium **ILS** (0.33 **AD**) with true gene trees, but there is a larger gap in the other three conditions. On the avian simulations, this gap is visible in all model conditions, but it becomes smaller as **ILS** level increases. For the avian simulations, the overall **rooted** species tree error is higher under high **ILS** conditions, suggesting that the error is dominated by species tree estimation error (Table 4.6), and is consistent with the

trend seen on the 201-taxon datasets. However, this trend is reversed for the mammalian simulations, and the final **rooted** tree error is higher under lowest **ILS** (0.18 **AD**) condition when rooting with QR-STAR (also see Table 4.5 for ASTRAL **RF** rates).

4.5.4 Discussion of Experimental Results

Some trends seen here are as expected: for example, accuracy generally improves for both QR and QR-STAR with the number of genes, and using true gene trees produces better accuracy than using estimated gene trees. These trends can be explained by noting that more data and better quality data improve accuracy. On the other hand, we also see that the combination of low **ILS** and deep speciation (towards the root) makes for easier conditions for both QR and QR-STAR, while low **ILS** and recent speciation (towards the leaves) makes for more challenging conditions for QR and QR-STAR; it is not clear why this is true.

An interesting trend seen in our experimental study is that rooting with QR and QR-STAR is more accurate under higher levels of discordance due to **ILS**, and becomes less accurate as the **ILS** level decreases. An explanation for this is that for a fixed number of gene trees, with less discordance due to **ILS**, it is likely that many gene trees that have low probability of appearing will not appear in the input, or will appear with very low frequencies, thus leading to higher error in the estimated probability distribution on quintet trees. This will increase error in the rooting performed by QR and QR-STAR. Furthermore, when enough gene trees fail to appear in the distribution, some estimates of quintet probabilities would become zero, and it may not be possible to differentiate some of the **rooted** quintets using the inequalities and invariants derived from the **ADR** theory. In the extreme case where there is no discordance due to **ILS** (and so all true gene trees are identical to the species tree), there will be only one quintet **gene tree** with non-zero probability: when this happens, the identifiability theorem in [81] would not hold and it becomes impossible to find the root. In contrast, the accuracy of ASTRAL and other species tree estimation methods decreases under higher levels of **ILS** [32, 48, 50].

Inference of the **rooted** species tree depends on both accurate estimation of the **unrooted** species tree **topology** as well as correct rooting of that tree. However, the level of **ILS** has a very different impact on these two steps. In most cases in our study, the overall error was dominated by species tree estimation error, and hence increased as the **ILS** level increased. However, we saw a different trend on the mammalian dataset, in which the species tree estimation error was very low in the lowest **ILS** condition, but the rooting error was high such that the overall error was dominated by rooting error. In general, for the purposes of estimating a **rooted** species tree using this approach, moderate levels of **ILS** may make for a

better overall outcome than very low or very high levels of ILS.

Another important trend is that QR-STAR is nearly always at least as accurate as QR, and is more accurate under most conditions. In general, there is a clear advantage to QR-STAR over QR for the low ILS condition that holds across the different conditions (varying number of genes, using true or estimated gene trees, and rooting true or estimated species trees), and this advantage is also seen for the moderate ILS condition when the number of genes is small. In contrast, for the high ILS condition, there is typically no or very little difference between the two methods, and in some cases QR can be somewhat more accurate. We also note that QR-STAR’s advantage over QR is largest when using true gene trees, even under high ILS (Fig. 4.7), which suggests that QR may be somewhat more robust to GTEE than QR-STAR. Thus, QR-STAR has a theoretical advantage over QR but not always an accuracy advantage.

For many conditions, we observed a small gap between optimal rooting and QR-STAR. For example, on the 201-taxon dataset in moderate or high ILS conditions (Experiment 2), there was a very small difference in rooting error between QR-STAR and the optimal rooting, even using estimated gene trees, suggesting that QR-STAR is doing very well in these conditions. Under the low ILS conditions of the 201-taxon data, however, there is a larger gap between QR-STAR and optimal rooting, especially when using only a small to moderate number of estimated gene trees. We also saw a larger gap between QR-STAR and the optimal rooting in Experiment 3 where the model trees were based on the avian and mammalian datasets, although the gap was less under the high ILS conditions than for the low ILS conditions. These differences indicate that there are conditions where improvements to QR-STAR for its empirical performance should be sought, especially when the ILS level in the data is low. There are at least two ways to improve empirical performance, without sacrificing statistical consistency—modifying the cost function and changing the quintet sampling strategy—and both of these should be explored in future work.

4.6 CONCLUSIONS

We have presented QR-STAR, a polynomial-time statistically consistent method for rooting species trees under the multispecies coalescent model. QR-STAR is an extension to QR, a method for rooting species trees introduced in [230]. QR-STAR differs from QR in that it has an additional step for determining the topological shape of each unrooted quintet selected in the QR algorithm, and incorporates the knowledge of this shape in its cost function, alongside the invariants and inequalities previously used in QR. We also showed that the statistical consistency for QR-STAR holds for a larger family of optimization problems

based on cost functions and sampling methods, and that modifying the linear encoding to be based on short quintets enables QR-STAR to have polynomial sample complexity.

To the best of our knowledge, this is the first work that established the statistical consistency of any method for rooting species trees under a model that incorporates gene tree heterogeneity. It remains to be investigated whether other rooting methods can also be proven statistically consistent under models of gene evolution inside species trees, such as the MSC or models of GDL. For example, STRIDE [214] and DISCO+QR [274] are methods that have been developed for rooting species trees from gene family trees, where genes evolve under GDL; however, it is not known whether these methods are statistically consistent under any GDL model.

This study suggests several directions for future research. For example, we proved statistical consistency for one class of cost functions, which was a linear combination of the invariant, inequality and shape penalty terms; however, cost functions in other forms could also be explored and proven statistically consistent. Theorem 4.3 shows that the sample complexity of QR-STAR depends on both the length of the shortest branch and the longest path in the model tree. This suggests that having very short or very long branches can both confound rooting under ILS, which is also suggested in previous studies [81, 266]. This is unlike what is known for species tree estimation methods such as ASTRAL, where the sample complexity is only affected by the shortest branch of the model tree [275, 276], and trees with long branches are easier to estimate.

Another theoretical direction is the construction of the rooted species tree directly from the unrooted gene trees. As explained in Remark 4.2, the proof of consistency of QR-STAR for 5-taxon trees does not depend upon the knowledge of the unrooted tree topology; this suggests that it is possible to estimate the rooted topology of the species tree in a statistically consistency manner *directly* from unrooted gene tree topologies. Future work could focus on developing statistically consistent methods for this problem, which is significantly harder than the problem of rooting a given tree.

There are also directions for improving empirical results. An important consideration in designing a good cost function is its empirical performance, as many cost functions can lead to statistical consistency but may not provide accurate estimations of the rooted tree in practice (see Figures 4.5 and 4.6). One potential direction is to incorporate estimated branch lengths, whether of the gene trees or the unrooted species tree, into the rooting procedure. These improvements can especially be useful for datasets with low levels of ILS which create the most difficult conditions for QR-STAR and where there is a gap between the accuracy of QR-STAR and the optimal rooting.

Finally, the experiments in this study were limited to comparisons between QR, QR-

STAR, and the optimal rooting of the ASTRAL species trees. In our prior study presenting QR [230], we showed that QR had good accuracy compared to many prior rooting methods. That study, however, was restricted to a small number of model conditions. Hence, future work should also include a comparison of QR-STAR to a larger number of rooting methods, including `outgroup` rooting, and under a wider range of model conditions.

4.7 METHODS AND SOFTWARE COMMANDS

- **ASTRAL:** We used ASTRAL (v5.7.8) to estimate `unrooted` species trees, with the specified number of true or estimated gene trees for each model condition. ASTRAL is available at <https://github.com/smirarab/ASTRAL>. We used the following command:

```
java -jar astral.5.7.8.jar -i <input-genes.tre> -o <output.tre>
```

- **QR:** We used QR (v1.2.4) to root `unrooted` species trees. QR is available at https://github.com/ytabatabae/Quintet_Rooting. We used the following command:

```
python3 quintet_rooting.py -t <input-tree.tre>
-g <input-genes.tre> -o <output.tre> -sm le
```

The `-LE` option specifies the quintet sampling method as “linear encoding”.

- **QR-STAR:** QR-STAR is available as part of the QR software package, and we ran it using the following command in the comparisons to QR, that sets $C = 1E-02$ and $\frac{\alpha_{max}}{\beta_{min}} = 0$:

```
python3 quintet_rooting.py -t <input-tree.tre>
-g <input-genes.tre> -o <output.tre> -sm le -c STAR
-abratio 0 -coef 0.01
```

- **Optimal rooting:** We used the script available at https://github.com/ytabatabae/QR-STAR-paper/scripts/optimal_rooting.py to find a rooting of an estimated tree that has minimum `nCD` error with respect to a reference `rooted` tree.

```
python3 optimal_rooting.py -r <reference-rooted.tre> -t
<unrooted.tre> -o <output.tre>
```

- **GTEE, AD and Rooting (nCD) Error:** Gene tree estimation error and average distance between model species trees and true gene was computed using a script

for computing normalized RF distance available at https://github.com/ekmolloy/njmerge/python/compare_trees.py.

Rooting error was measured in terms of average normalized clade distance (**nCD**) using the script available at https://github.com/ytabatabae/Quintet-Rooting/scripts/clade_distance.py

4.8 ADDITIONAL FIGURES AND TABLES

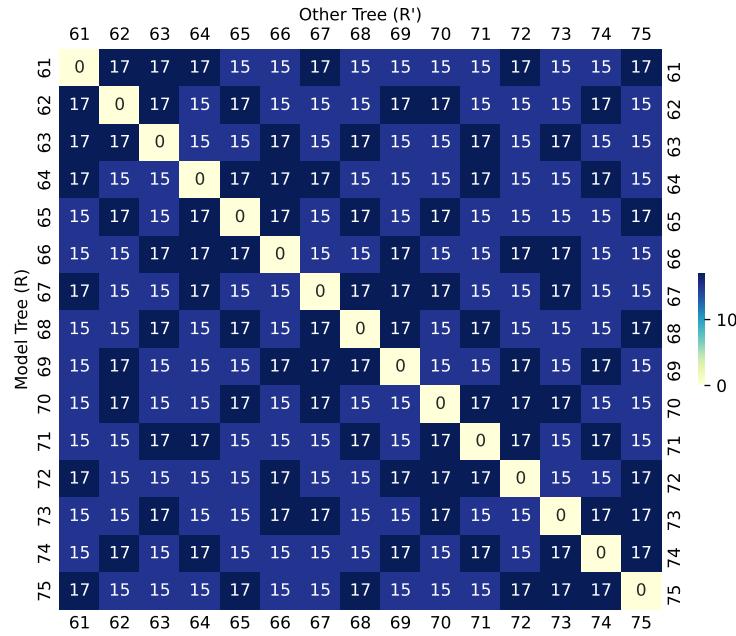


Figure 4.10: **Conflicts between 5-taxon pseudo-caterpillar trees.** Heatmap showing the number of conflicting inequality penalty terms (the function $|V(R, R')|$) for pairs of pseudo-caterpillar 5-taxon rooted trees.

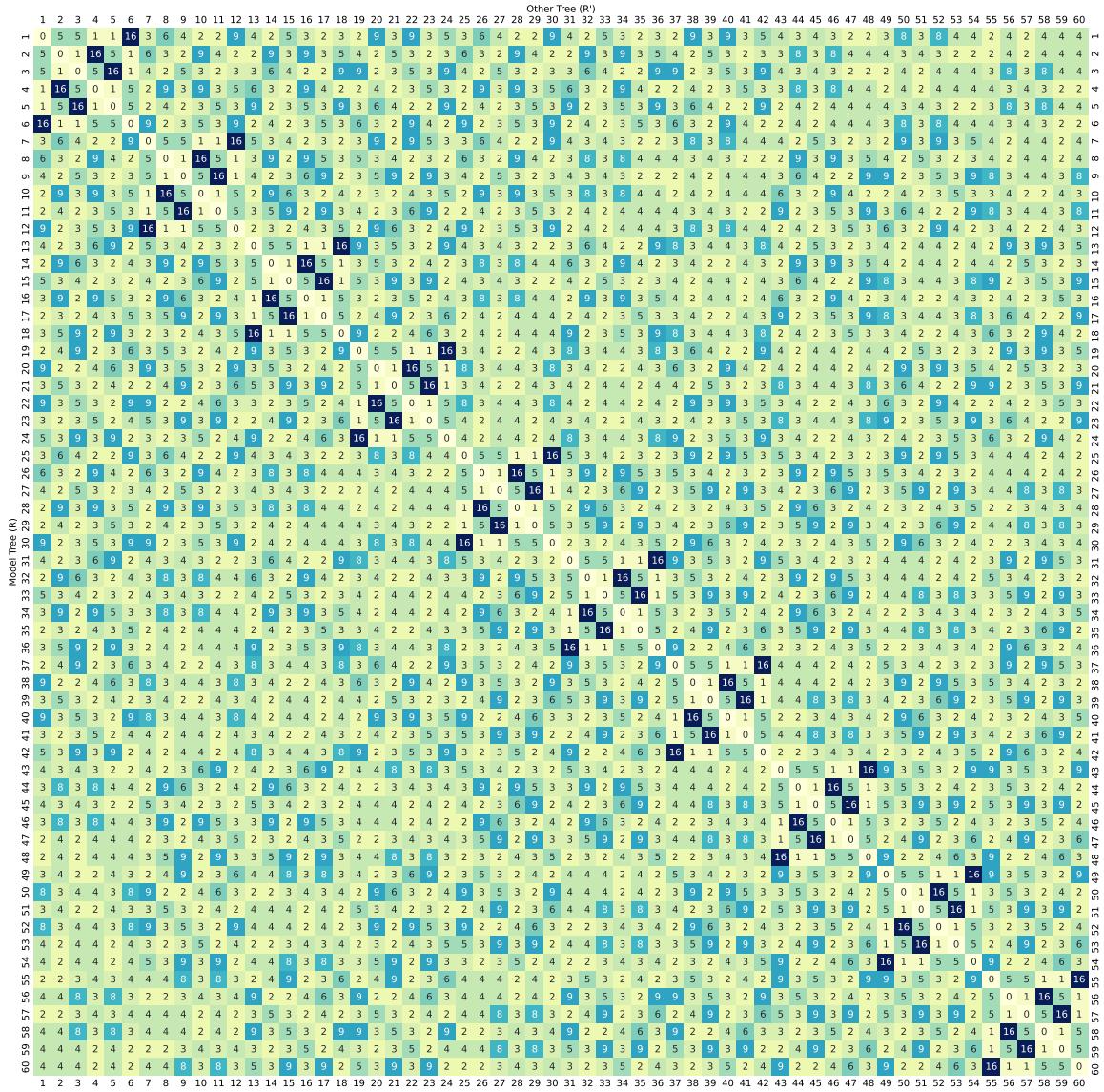


Figure 4.11: Conflicts between 5-taxon caterpillar trees. Heatmap showing the number of conflicting inequality penalty terms (the function $|V(R, R')|$) for pairs of caterpillar 5-taxon rooted trees.

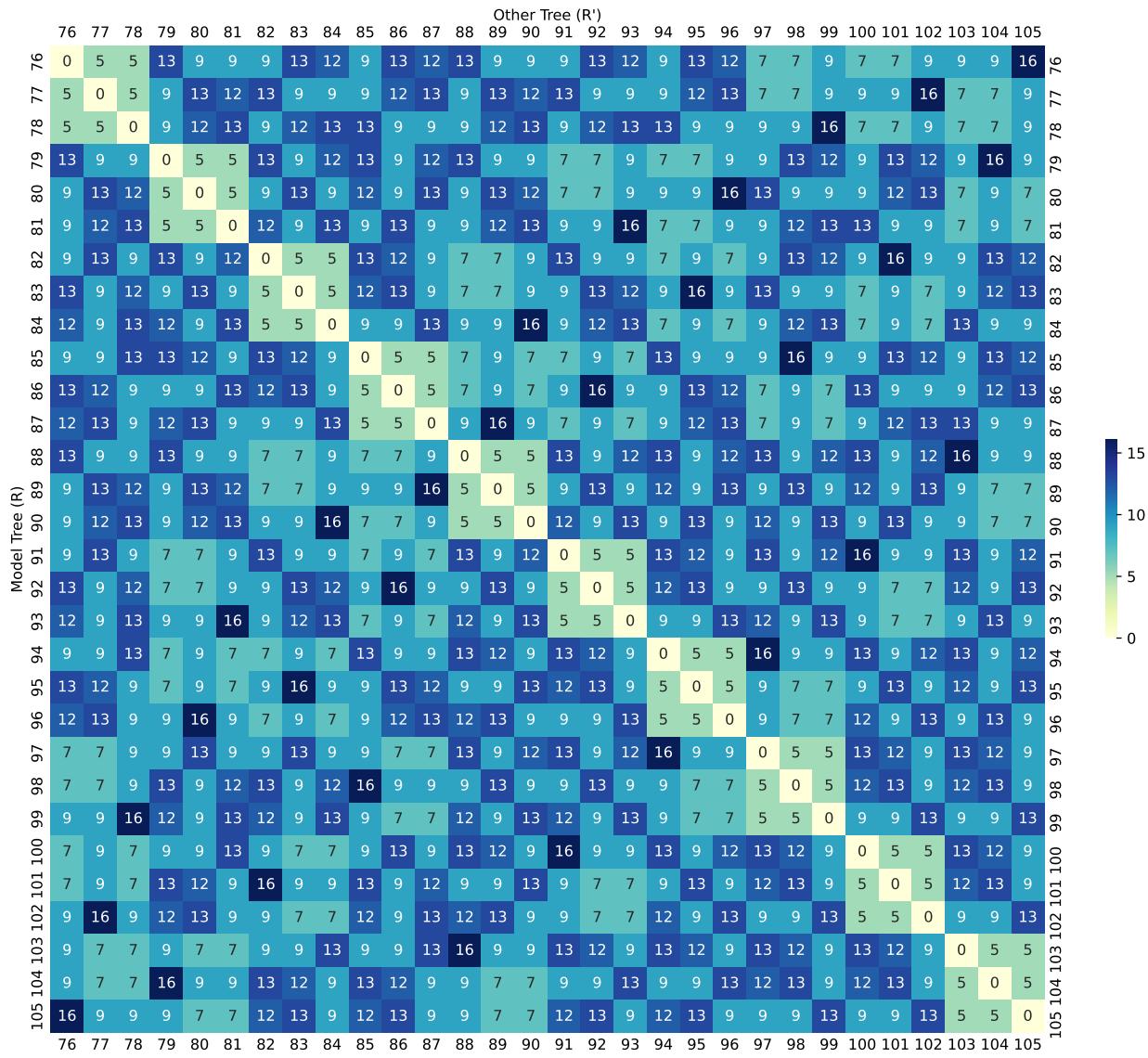


Figure 4.12: **Conflicts between 5-taxon balanced trees.** Heatmap showing the number of conflicting inequality penalty terms (the function $|V(R, R')|$) for pairs of balanced 5-taxon rooted trees.

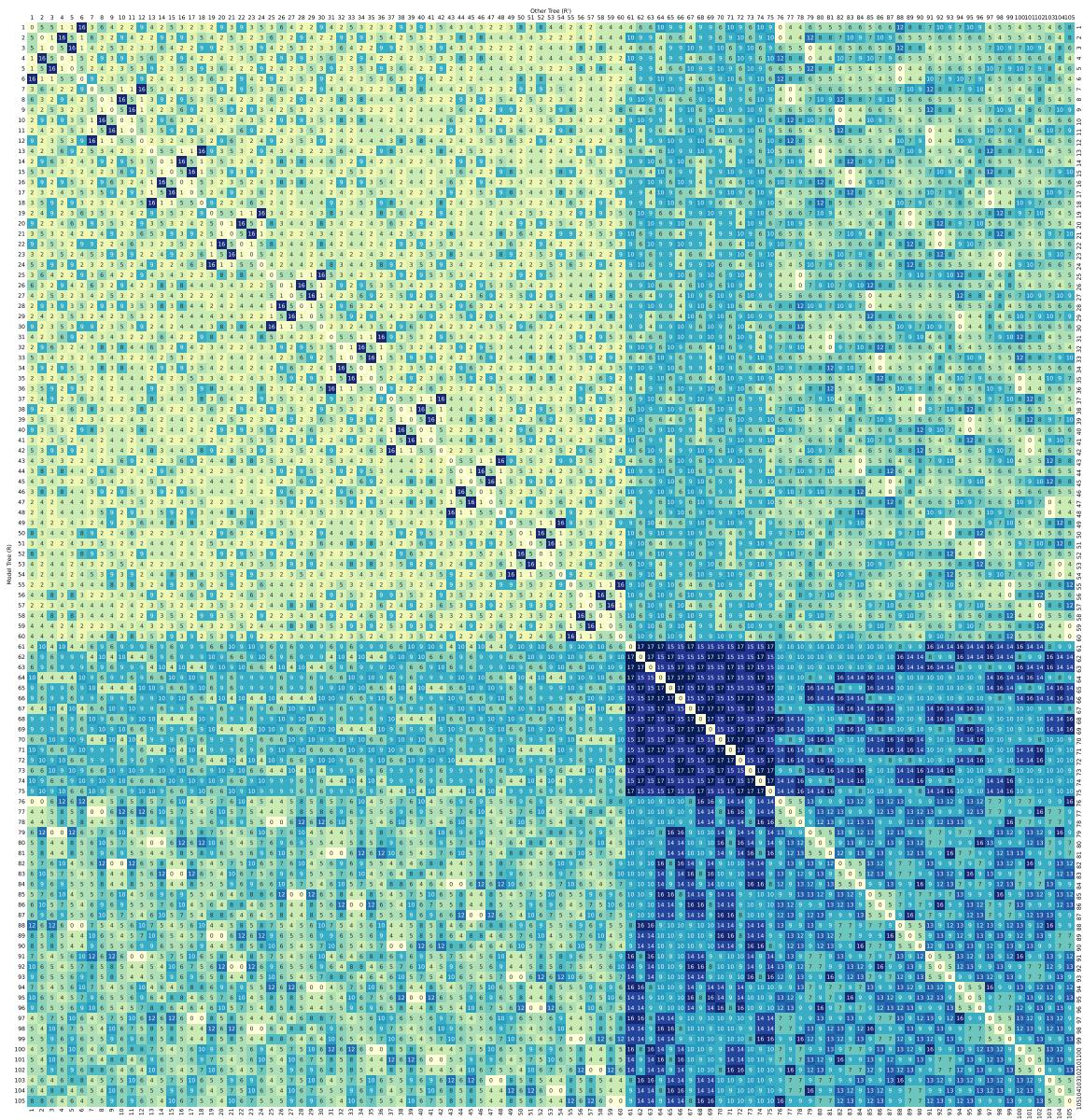


Figure 4.13: Conflicts between all 5-taxon rooted trees. Heatmap showing the number of conflicting inequality penalty terms (the function $|V(R, R')|$) for all pairs of binary 5-taxon rooted trees.

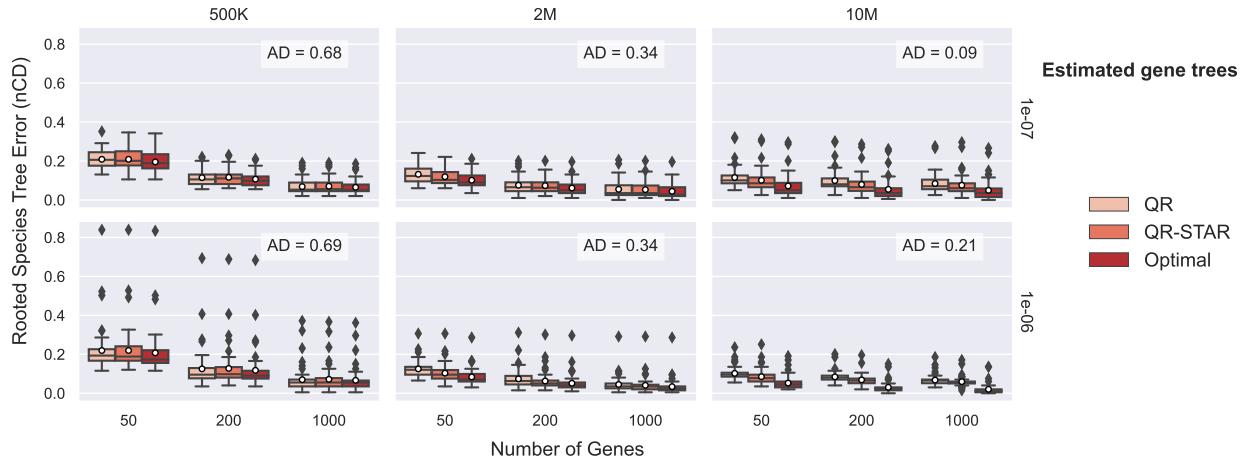


Figure 4.14: **Rooting the ASTRAL species tree on 201-taxon datasets with estimated gene trees.** Full version of Figure 4.8 for estimated gene trees (including outliers). The results are shown across 50 replicates. The columns show tree height (500K for high ILS, 2M for moderate ILS, and 10M for low ILS) and the rows show speciation rate (1e-06 for recent speciation, 1e-07 for deep speciation).

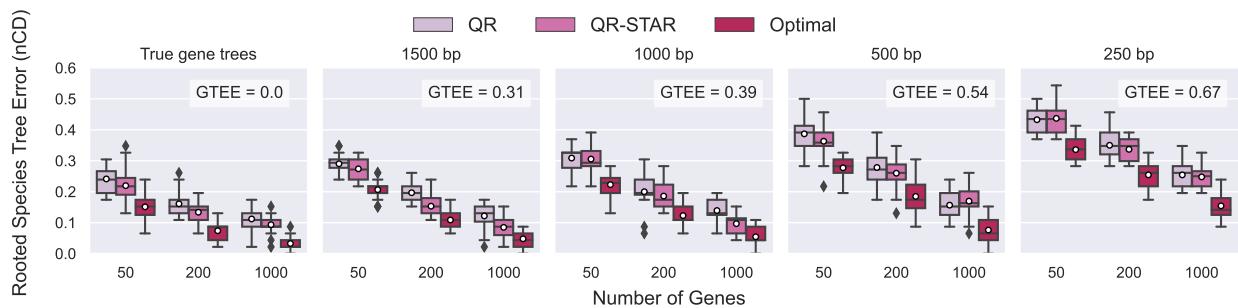


Figure 4.15: **Rooting the ASTRAL species tree on avian simulated dataset.** Comparison between QR, QR-STAR and optimal rooting on 48-taxon avian simulated dataset for 1X ILS model condition. The columns show whether true or estimated gene trees were used, and the sequence length used to produce the gene trees. The number of replicates in each model condition is 20.

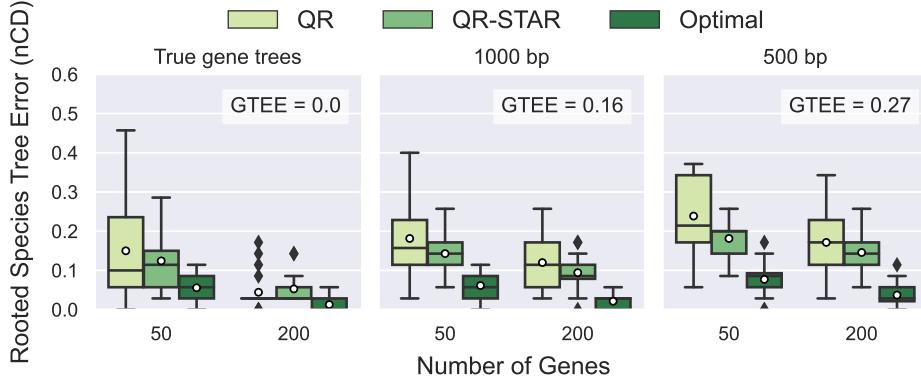


Figure 4.16: Rooting the ASTRAL species tree on mammalian simulated dataset. Comparison between QR, QR-STAR and optimal rooting on 37-taxon mammalian simulated dataset for 1X ILS model condition. The columns show whether true or estimated gene trees were used, and the sequence length used to produce the gene trees. The number of replicates in each model condition is 20.

Table 4.1: Statistics for the 201-taxon simulated datasets. ILS level (measured using the average distance between the true gene trees and the model species tree, or AD) and average gene tree estimation error (GTEE) of the estimated gene trees for the six model conditions of the 201-taxon datasets with 1000 gene trees. The two speciation rates used in this dataset indicate whether speciation happened close to the leaves (i.e. recent speciation for 1e-06) or close to the root (i.e. deep speciation for 1e-07). The shorter tree height (500K), indicates shorter branches and therefore higher levels of ILS.

| speciation rate | tree height | ILS level (AD) | GTEE |
|-----------------|-------------|----------------|------|
| 1E-06 | 500K | 0.69 | 0.49 |
| | 2M | 0.34 | 0.28 |
| | 10M | 0.21 | 0.22 |
| 1E-07 | 500K | 0.68 | 0.46 |
| | 2M | 0.34 | 0.34 |
| | 10M | 0.09 | 0.29 |

Table 4.2: **Statistics for the 48-taxon avian simulated dataset.** ILS level (measured using the average distance between the true gene trees and the model species tree, or AD) and average gene tree estimation error (GTEE) of the estimated gene trees for different model conditions of the avian datasets with 1000 gene trees.

| model condition | sequence length (bp) | ILS level (AD) | GTEE |
|-----------------|----------------------|----------------|------|
| 0.5X | 500 | 0.59 | 0.59 |
| | 250 | 0.47 | 0.67 |
| | 500 | 0.47 | 0.54 |
| | 1000 | 0.47 | 0.39 |
| | 1500 | 0.47 | 0.31 |
| 2X | 500 | 0.35 | 0.57 |

Table 4.3: **Statistics for the 37-taxon mammalian simulated dataset.** ILS level (measured using the average distance between the true gene trees and the model species tree, or AD) and average gene tree estimation error (GTEE) of the estimated gene trees for different model conditions of the mammalian datasets with 200 gene trees.

| model condition | sequence length (bp) | ILS level (AD) | GTEE |
|-----------------|----------------------|----------------|------|
| 0.5X | 500 | 0.54 | 0.27 |
| | 500 | 0.33 | 0.27 |
| | 1000 | 0.33 | 0.16 |
| 2X | 500 | 0.18 | 0.27 |

Table 4.4: **ASTRAL RF error rates on the 201-taxon datasets.** Species tree estimation error in terms of normalized RF distance for ASTRAL trees given true and estimated gene trees for the 201-taxon datasets. The values show mean and standard deviation of RF errors across 50 replicates for each model condition.

| speciation rate | tree height | number of genes | RF (true gene trees) | RF (est gene trees) |
|-----------------|-------------|-----------------|----------------------|---------------------|
| 1E-06 | 500K | 1000 | 0.028 ± 0.012 | 0.067 ± 0.065 |
| | | 200 | 0.065 ± 0.021 | 0.118 ± 0.103 |
| | | 50 | 0.137 ± 0.030 | 0.208 ± 0.118 |
| 1E-06 | 2M | 1000 | 0.010 ± 0.008 | 0.034 ± 0.041 |
| | | 200 | 0.023 ± 0.011 | 0.052 ± 0.045 |
| | | 50 | 0.045 ± 0.015 | 0.084 ± 0.042 |
| 1E-06 | 10M | 1000 | 0.006 ± 0.006 | 0.020 ± 0.028 |
| | | 200 | 0.012 ± 0.010 | 0.031 ± 0.029 |
| | | 50 | 0.027 ± 0.014 | 0.053 ± 0.037 |
| 1E-07 | 500K | 1000 | 0.028 ± 0.011 | 0.065 ± 0.036 |
| | | 200 | 0.066 ± 0.018 | 0.107 ± 0.038 |
| | | 50 | 0.143 ± 0.024 | 0.197 ± 0.046 |
| 1E-07 | 2M | 1000 | 0.010 ± 0.007 | 0.045 ± 0.040 |
| | | 200 | 0.024 ± 0.011 | 0.062 ± 0.038 |
| | | 50 | 0.049 ± 0.017 | 0.102 ± 0.041 |
| 1E-07 | 10M | 1000 | 0.002 ± 0.003 | 0.050 ± 0.054 |
| | | 200 | 0.004 ± 0.004 | 0.055 ± 0.055 |
| | | 50 | 0.011 ± 0.007 | 0.072 ± 0.061 |

Table 4.5: **ASTRAL RF error rates on the mammalian simulated dataset.** Species tree estimation error in terms of normalized RF distance for ASTRAL trees on the 37-taxon mammalian simulated dataset. The values show mean and standard deviation of RF errors across 20 replicates for each model condition.

| model condition | sequence length (bp) | number of genes | RF rate |
|-----------------|----------------------|-----------------|---------------|
| 0.5X | NA (true gene trees) | 200 | 0.035 ± 0.024 |
| | | 50 | 0.094 ± 0.044 |
| | 500 | 200 | 0.049 ± 0.025 |
| | | 50 | 0.107 ± 0.049 |
| 1X | NA (true gene trees) | 200 | 0.013 ± 0.017 |
| | | 50 | 0.057 ± 0.039 |
| | 1000 | 200 | 0.022 ± 0.021 |
| | | 50 | 0.063 ± 0.031 |
| | 500 | 200 | 0.038 ± 0.032 |
| | | 50 | 0.079 ± 0.044 |
| 2X | NA (true gene trees) | 200 | 0.003 ± 0.009 |
| | | 50 | 0.018 ± 0.020 |
| | 500 | 200 | 0.025 ± 0.021 |
| | | 50 | 0.066 ± 0.031 |

Table 4.6: **ASTRAL RF error rates on the avian simulated dataset.** Species tree estimation error in terms of normalized RF distance for ASTRAL trees on the 48-taxon avian simulated dataset. The values show mean and standard deviation of RF errors across 20 replicates for each model condition.

| model condition | sequence length (bp) | number of genes | RF rate |
|-----------------|----------------------|-----------------|---------------|
| 0.5X | NA (true gene trees) | 1000 | 0.048 ± 0.030 |
| | | 200 | 0.151 ± 0.029 |
| | | 50 | 0.242 ± 0.040 |
| | 500 | 1000 | 0.131 ± 0.040 |
| | | 200 | 0.243 ± 0.042 |
| | | 50 | 0.332 ± 0.053 |
| 1X | NA (true gene trees) | 1000 | 0.033 ± 0.023 |
| | | 200 | 0.076 ± 0.030 |
| | | 50 | 0.154 ± 0.041 |
| | 1500 | 1000 | 0.049 ± 0.028 |
| | | 200 | 0.111 ± 0.031 |
| | | 50 | 0.211 ± 0.032 |
| | 1000 | 1000 | 0.056 ± 0.029 |
| | | 200 | 0.126 ± 0.032 |
| | | 50 | 0.228 ± 0.042 |
| | 500 | 1000 | 0.078 ± 0.038 |
| | | 200 | 0.189 ± 0.053 |
| | | 50 | 0.283 ± 0.035 |
| | 250 | 1000 | 0.158 ± 0.045 |
| | | 200 | 0.260 ± 0.043 |
| | | 50 | 0.343 ± 0.040 |
| 2X | NA (true gene trees) | 1000 | 0.021 ± 0.016 |
| | | 200 | 0.046 ± 0.025 |
| | | 50 | 0.104 ± 0.039 |
| | 500 | 1000 | 0.043 ± 0.019 |
| | | 200 | 0.126 ± 0.043 |
| | | 50 | 0.226 ± 0.047 |

CHAPTER 5: PHYLOGENOMIC BRANCH LENGTH ESTIMATION USING QUARTETS

This chapter contains material previously published in “Y. Tabatabae, C. Zhang, T. Warnow and S. Mirarab. (2023). Phylogenomic branch length estimation using quartets. Bioinformatics, Vol. 39, Issue Supplement 1, pages i185-i193, special issue for Intelligent Systems for Molecular Biology and European Conference on Computational Biology (ISMB/ECCB) 2023”[74]. The CASTLES software is available in open-source form at <https://github.com/ytabatabae/CASTLES>. The datasets and scripts used in this study are available at <https://github.com/ytabatabae/CASTLES-paper>.

5.1 INTRODUCTION

Species trees, both their topologies and their branch lengths, are necessary for downstream biological research. For example, branch lengths are required for comparative genomics [277] and comparative trait analysis [278, 279], phylodynamics of disease transmission [280], species delimitation [281], measuring phylogenetic diversity [282, 283], and detecting and characterizing selection [284]. Many of these analyses amount to studying changes in the rate of evolution across the tree [285]. Most statistical methods designed for these applications rely on branch lengths measured in the unit of the expected number of substitutions per site (SU), readily available from tree inference based on sequence data, or unit of time, or both.

The traditional approach to the estimation of species trees and branch lengths has been concatenating gene alignments followed by a tree-building method, such as maximum likelihood [286]. It is now understood [28] that this concatenation approach can be **positively misleading** (i.e., converge to the wrong tree as the number of genes increases) in the face of sufficient gene tree **heterogeneity** across the genome due to **Incomplete lineage sorting (ILS)**, as modeled by the **Multi-Species Coalescent (MSC)** model [287].

Alternative approaches for estimating species trees have been developed that are **statistically consistent** under the **MSC** [see 160]. In particular, methods that combine a set of gene trees to infer a **species tree** (referred to as “summary methods”) are widely used because of their scalability and accuracy (and notably better accuracy than concatenation when **ILS** is high). Well-known examples of such methods are ASTRAL [179] and MP-EST [288], which are used often to analyze phylogenomic datasets. However, the branch lengths produced by summary methods are in **coalescent units (CU)**, and these do not directly lead to branch lengths in substitution units. Moreover, branch lengths in coalescent units are inferable only for the **internal** branches, which further limits their utility.

At the current time, therefore, most coalescent-based analyses estimate species trees and their branch lengths in **substitution units (SU)** following a two-stage approach, where the first stage computes the tree **topology** (e.g., using a summary method, such as ASTRAL or MP-EST) and then estimates branch lengths on the tree using a constrained concatenation analysis, such as using a **Maximum likelihood (ML)** method to infer branch lengths on a fixed tree **topology** [e.g., 289]. However, one major problem with this approach is that the branch length calculation step ignores gene tree **heterogeneity** across the genome, leading to criticisms of this approach in the scientific literature. For example, Moody et al. (2022) [290] criticized the findings by Zhu et al. (2019) [291] who postulated a shorter length than previously reported separating archaea and bacteria, arguing that the use of concatenation for branch length estimation can lead to substantial under-estimation in the face of high levels of horizontal gene transfer, where gene trees have widely discordant topologies.

Another approach for **SU** branch length estimation on species trees is ERaBLE [292], which uses the **SU** branch lengths estimated in a set of gene trees and then solves a weighted least-squared optimization problem to assign **SU** branch lengths to the species tree. However, as with the standard concatenation approach, ERaBLE does not take **heterogeneity** in gene tree *topologies* due to **ILS** into account.

When a **strict molecular clock** holds, then branch length estimation in the **species tree** becomes feasible. However, it is well understood that strict clock-based methods have poor accuracy for many datasets where mutation rates change across the tree [140, 293]. Hence, clock-based approaches do not offer a viable solution.

To summarize, existing methods to compute **SU** branch lengths that take discordance between gene trees due to **ILS** into account, without a **strict molecular clock**, have not yet been developed. And in particular, we currently lack a theoretical basis for inferring **SU** lengths for species trees that addresses **heterogeneity** in gene tree topologies due to **ILS**, as modeled by the **MSC**. This is a glaring gap that needs to be filled.

The unsatisfactory state-of-the-art leads us to ask: How can we estimate branch lengths on species trees that are accurate, even in the face of high levels of **ILS** and that does not depend on a strong **molecular clock**? We specifically seek a method that has a strong theoretical foundation based on the **MSC**. We also seek to develop a method that is sufficiently fast that it is scalable to large genome-wide datasets with hundreds to thousands of genes and species. Because we seek to develop scalable methods, Bayesian co-estimation of gene trees and species trees (topologies and branch lengths) is infeasible as even the best of these methods are computationally intensive on smaller datasets with about 50 species and 200 genes [41, 294].

Here, we propose the Coalescent-Aware Species Tree Length Estimation in Substitution-

unit (CASTLES) method. The input to CASTLES is a [rooted species tree topology](#) and a set of inferred gene trees with [SU](#) branch lengths, which can have missing data, polytomies, and multiple individuals per species. The output is the [species tree](#) furnished with [SU](#) lengths on all branches; at the root, only the sum of the lengths of the two root-incident branches is inferred. CASTLES addresses gene tree [heterogeneity](#) under the [MSC](#) and naturally occurring variation in mutation rates; thus, it does not assume [strict molecular clock](#). Similar to methods like ASTRAL for [species tree topology](#) inference, we use a quartet-based approach. We first derive the expected branch length of gene trees that do or do not match a [quartet species tree](#) under our model as a function of their [CU](#) length and mutation rates. These derivations suggest an algorithm for estimating [SU](#) branch lengths, but the approach is cumbersome to implement. Through approximations and simplifications, we derive a much simpler estimator that still retains non-ultrametricity. Going beyond trees with four species in a naive way, iterating over all $\binom{n}{4}$ quartets, would lead to the loss of scalability. Instead, we design a sophisticated [dynamic programming \(DP\)](#) algorithm to compute quantities needed by our algorithm in quadratic time. We compare CASTLES to leading alternatives using a simulation study and demonstrate its superior accuracy and speed. Finally, we apply it to a biological dataset.

5.2 THEORETICAL RESULTS

We first describe a model that generates gene trees with [SU](#) branch lengths. We then derive the expected gene tree branch lengths under this model for a single quartet. The resulting set of non-linear equations can be (approximately) solved using numerical methods, and will yield values for the model parameters given quantities that can be measured from the gene trees. However, solving these equations is computationally intensive, involves numerical instabilities, may not produce optimal solutions, and is cumbersome; therefore, here we present simplifications that give analytical formulas for every branch of a quartet tree. Using equations for the simplified model, we then develop an algorithm that can handle a tree with arbitrary size n , including a scalable (quadratic) dynamic programming algorithm to compute averages of quartet branch lengths across gene trees for $\Theta(n^4)$ quartets.

5.2.1 Preliminaries.

Under the [MSC](#), waiting times before coalescent events are exponential random variables with rate $\lambda = \binom{k}{2}/N_e$, where k is the number of [lineages](#) entering an interval between two coalescent events and N_e is the effective population size [35]. Assuming $N_e = 1$, the

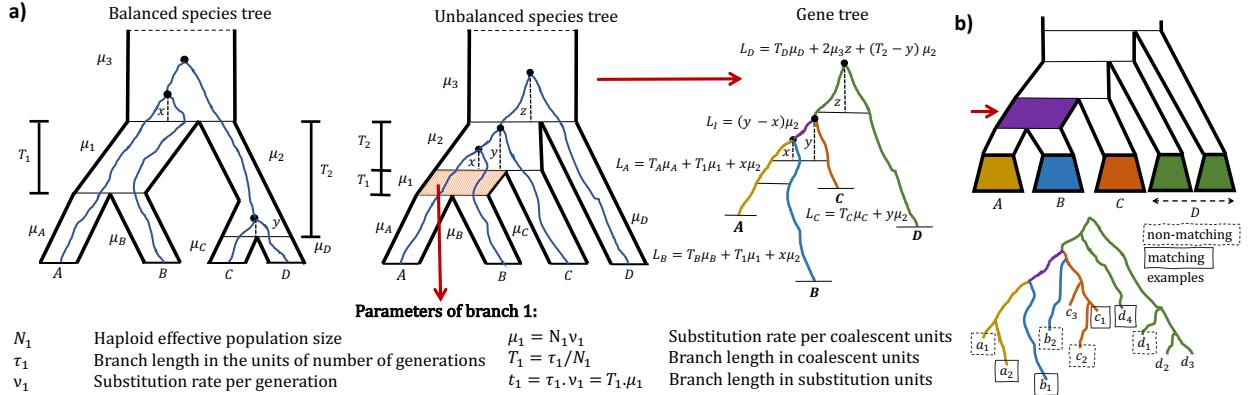


Figure 5.1: a) MSC+Substitution Model. Each branch of the species tree is furnished with parameters described in the legend. As a gene tree evolves inside the species tree, its branches inherit the substitution rates of all the *species tree* branches that they pass through. When mutation rates change across *species tree* branches, the resulting gene tree is non-ultrametric. We match the theoretical expected values of the five branches of a gene tree that matches or does not match the *species tree* (namely, L_A, L_B, L_C, L_D , and L_I for a matching gene tree shown here) to their empirical means, computed from gene trees. b) Handling a tree with more than 4 taxa. Each focal *internal* branch (arrow) divides the tree into four groups, here denoted as *A*, *B*, *C*, and *D*. To use quartet-based equations, we average branch lengths over all quartets with one leaf selected from each of *A*, *B*, *C*, and *D* (e.g., a_1, b_1, c_1, d_1). Note that in one gene tree, some quartets around a *species tree* branch may contribute to matching while others contribute to non-matching average lengths (examples shown). We compute these averages efficiently without listing all $O(n^4)$ quartets using *dynamic programming (DP)*.

probability density function for the coalescence event between k *lineages* in an interval with length x is $f_X(x) = \lambda e^{-\lambda x} = \binom{k}{2} e^{-(\frac{k}{2})x}$; i.e., e^{-x} for two *lineages*, $3e^{-3x}$ for three *lineages*, and $6e^{-6x}$ for four *lineages*. The mean of this random variable is $\frac{1}{\lambda} = \frac{1}{(\frac{k}{2})}$ which is 1 and $\frac{1}{3}$ for two and three *lineages* respectively. Tavaré [108] derived an equation for the function $g_{ij}(T)$ as the probability that i *lineages* coalesce into j *lineages* in time T (see Section 2.4 for the complete formula). Specific cases of this function (shown in Figures 5.2 to 5.4) are tabulated by [107] and can be used to verify our derivations.

5.2.2 MSC+Substitution Model

Our model parameters include a *species tree* \mathcal{T} and several per-branch attributes (Figure 5.1a). Each branch i is furnished with a branch length τ_i in the unit of the number of generations, a haploid effective population size N_i , and a substitutions-per-generation rate ν_i . The *CU* length of the branch is simply $T_i = \tau_i/N_i$. Let $\mu_i = \nu_i \times N_i$ denote the *CU*

substitution rate. The **SU** length of the branch is $t_i = \tau_i \times \nu_i = T_i \times \mu_i$; thus, setting unequal ν values across the tree branches leads to a non-ultrametric species tree. Gene trees are first drawn under the **MSC** model (ignoring ν_i), thus producing trees with lengths in the unit of generations. Gene trees with **SU** length are generated by multiplying the length of every infinitesimally small part of each of their branches passing through a **species tree** branch i by the **species tree** rate μ_i . For example, the length of the **terminal** branch A in Figure 5.1 is $T_A\mu_A + T_1\mu_1 + x\mu_2$. Note that under this model, **species tree CU** lengths connect indirectly to **SU** and time units; inferring **SU** from **CU** requires $\mu_i = \nu_i \times N_i$, inferring the number of generations needs the population size, and inferring time additionally needs the generation time.

5.2.3 Outline of the Approach

As shown in Figures 5.2 and 5.3, we assume an unbalanced model **species tree** $((A, B) : T_1, C) : T_2, D)$ and a **gene tree** on the same leafset with an **unrooted topology** that either matches or does not match the **topology** of the species tree. In Figures 5.5 and 5.4, we work with the balanced **species tree** $((A, B) : T_1, (C, D) : T_2)$. The parameters of the model **species tree** (i.e. μ_i s and T_i s) are defined according to Figure 5.1. We denote the expected length of the **internal** branch in a **gene tree** with **unrooted topology** ψ (where ψ is either $ab|cd, ac|bd$ or $ad|bc$) by $\mathbb{E}(l_I(\psi))$, and the expected length of a **terminal** branch leading to **taxa** X by $\mathbb{E}(l_X(\psi))$. Figures 5.3 and 5.4 only show scenarios for non-matching gene trees that have the **topology** $ad|bc$, as the derivations for the other non-matching **topology** ($ac|bd$) is similar, and in all cases except for cherry branches, the expected lengths of a branch in these two topologies are the same.

We first compute the average expected length of the **internal** and **terminal** branches in a quartet **gene tree** for gene trees matching the **topology** of the **species tree** (denoted by L_I for the **internal** branch, L_X for the **terminal** branch leading to **taxa** X) as well as gene trees not matching the **species tree** (denoted by L'_I for **internal** branch, L'_X for **terminal** branch leading to **taxa** X) for both unbalanced and balanced model species trees.

To calculate these expected lengths, we consider all scenarios that lead to different branch lengths in matching or non-matching gene trees (summarized in Figures 5.2-5.4 for the **internal** branch) and their corresponding probabilities, and compute the following conditional expectation for the **internal** branch (similar expectations can be written for **terminal** branches, modifying l_I to l_X)

$$\mathbb{E}(l_I(\psi)) = \mathbb{E}(l_I|\Psi = \psi) = \int x f_{l_I|\Psi}(x|\psi) dx = \frac{1}{\mathbb{P}(\psi)} \int x f_{l_I,\Psi}(x, \psi) dx \quad (5.1)$$

where Ψ is a random variable denoting the unrooted gene tree topology, and $\mathbb{P}(\psi)$ is the probability of the specific topology ψ under the MSC, which can be computed as follows [81] for the unbalanced and balanced model species trees of Figure 5.1 respectively.

$$\begin{aligned} \text{unbalanced: } \mathbb{P}(ab|cd) &= 1 - \frac{2}{3}e^{-T_1} , \quad \mathbb{P}(ac|bd) = \mathbb{P}(ad|bc) = \frac{1}{3}e^{-T_1} \\ \text{balanced: } \mathbb{P}(ab|cd) &= 1 - \frac{2}{3}e^{-(T_1+T_2)} , \quad \mathbb{P}(ac|bd) = \mathbb{P}(ad|bc) = \frac{1}{3}e^{-(T_1+T_2)} \end{aligned} \quad (5.2)$$

In some figures, one shape corresponds to more than one scenario (specified in the caption); we use this only when both scenarios give the same expected internal branch length and are derived by swapping two lineages. All the calculations (in particular, integrals) in the proofs are verified using Mathematica, and the notebook is available at https://github.com/ytabatabae/CASTLES/blob/main/su_branch_calcs.nb.

5.2.4 Expected quartet branch lengths under the MSC

Focusing on a quartet, we now derive the expected length of all branches as a function of the model parameters. Consider unbalanced and balanced species trees shown in Figure 5.1. We present Lemmas 5.1 – 5.6, which derive the expected length of each terminal and internal branch in the gene trees that do or do not match the unbalanced or balanced species trees. Note that these expectations can be estimated in a statistically consistent manner given true gene trees with SU branch lengths. Combined, we derive 10 equations across the five branches, relating the measurable expected values to the unknown parameters. Since μ_i and T_i only appear as $t_i = \mu_i T_i$ for all terminal branches ($i \in \{A, B, C, D\}$), we have four unknown parameters for terminal branches. With three unknown rates (μ_1, μ_2, μ_3) and two unknown CU internal lengths (T_1 and T_2), we have 9 unknowns in total.

This non-linear system of 10 equations and 9 unknowns can be (approximately) solved using numerical methods to jointly estimate all the parameters. Such an optimization approach, however, is subject to numerical instability, may be slow, and may not give optimal solutions for this (possibly) non-convex optimization problem. Instead of exploring that path, we observe that by making some simplifying assumptions, we can compute all the branch lengths analytically.

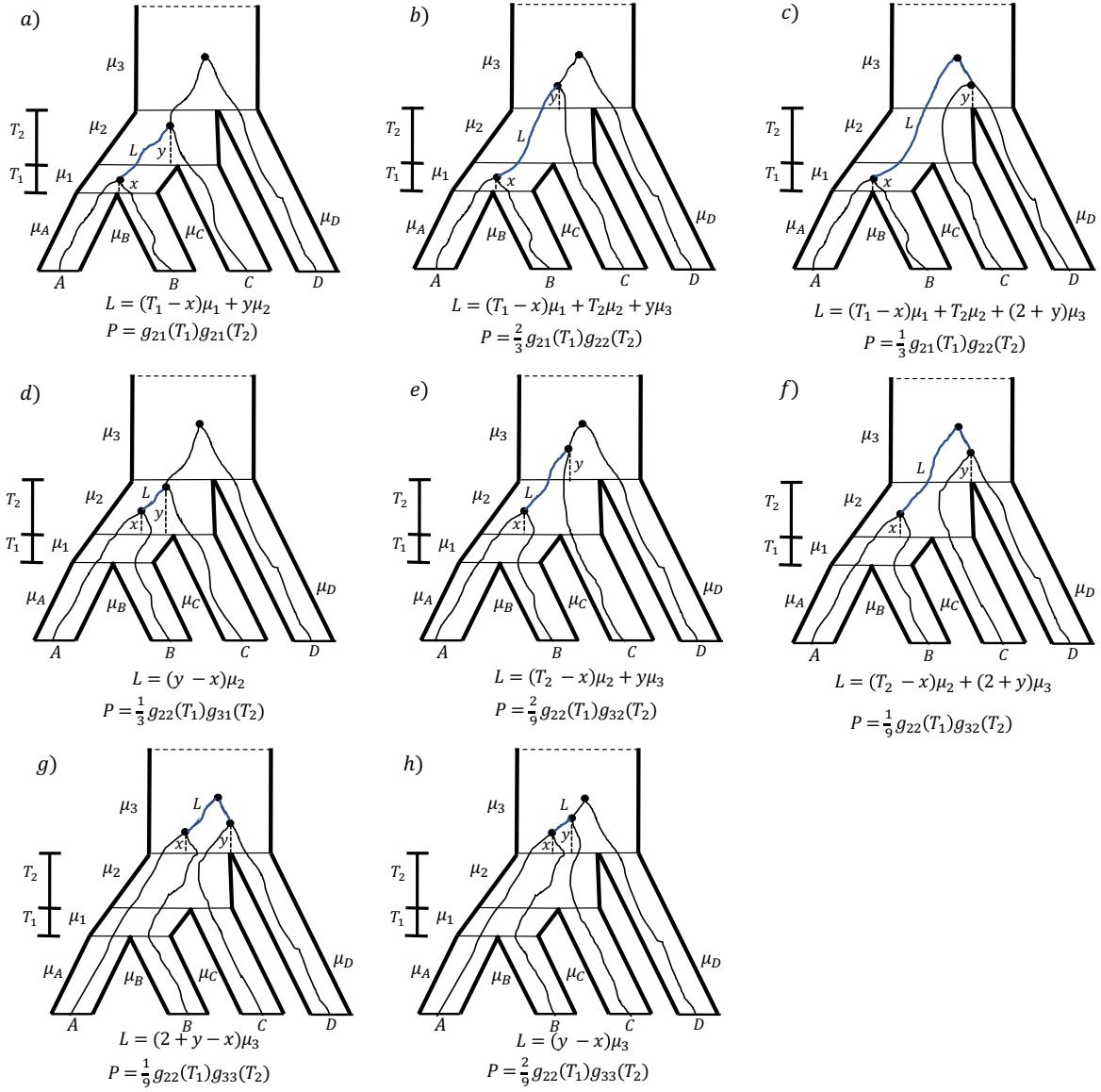


Figure 5.2: Scenarios for gene tree matching the unbalanced species tree. internal branch lengths for unrooted quartet gene tree matching the unbalanced model species tree $((A, B) : T_1, C) : T_2, D$. L denotes the internal branch length in the gene tree and P denotes the probability of each case. Case (b) and (e) correspond to two scenarios and case (h) corresponds to four different scenarios, and the P values reported in these cases show the overall probability of all possible scenarios for that case.

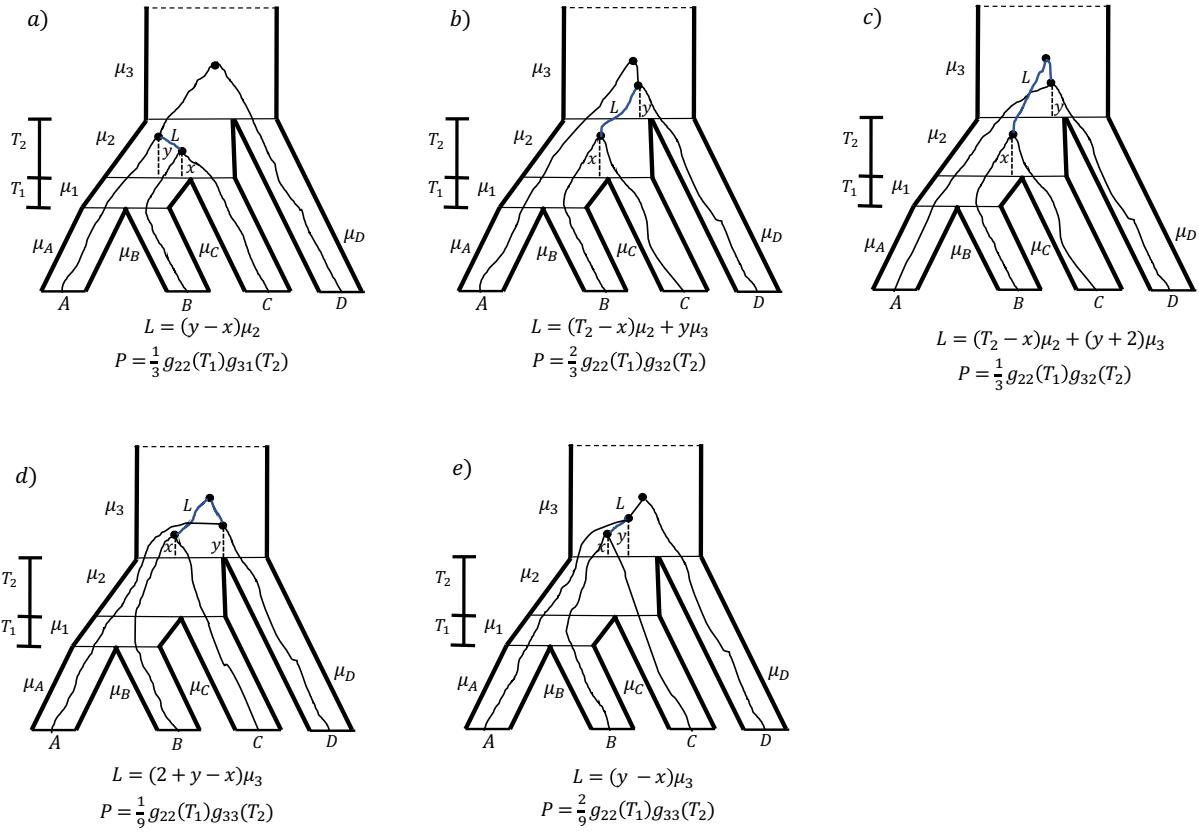


Figure 5.3: Scenarios for gene tree not matching the unbalanced species tree. internal branch lengths for unrooted quartet gene tree not matching the unbalanced model species tree $((A, B) : T_1, C) : T_2, D$. L denotes the internal branch length in the gene tree and P denotes the probability of each case. Case (b) and (d) correspond to two scenarios and case (e) corresponds to four different scenarios, and the P values reported in these cases show the overall probability of all possible scenarios for that case.

Unbalanced species trees. Before proving our main results for unbalanced trees, we introduce and prove four lemmas: All of our results and their proofs are in reference to Figures 5.2 and 5.3.

Lemma 5.1 (Internal unbalanced). For the unbalanced model species tree of Figure 5.1, the expected length of the **internal** branch of a **gene tree** with an **unrooted topology** matching the **species tree** in substitution units is

$$L_I = \mathbb{E}(l_I(ab|cd)) = \frac{(e^{-3T_2} + 3e^{-T_2} - 6e^{T_1-T_2})(\mu_2 - \mu_3) + 6(1 - e^{T_1} + T_1e^{T_1})\mu_1}{2(3e^{T_1} - 2)} + \mu_2 \quad (5.3)$$

and the expected length for gene trees not matching the **species tree topology** is

$$L'_I = \mathbb{E}(l_I(ac|bd)) = \mathbb{E}(l_I(ad|bc)) = \mu_2 + \frac{1}{2}(\mu_2 - \mu_3)(e^{-3T_2} - 3e^{-T_2}) . \quad (5.4)$$

Proof. Referring to Figure 5.2, we can compute

$$\begin{aligned} \mathbb{E}(L_I(ab|cd)) &= \left(\int_0^{T_1} \int_0^{T_2} e^{-x} e^{-y} ((T_1 - x)\mu_1 + y\mu_2) dy dx \right) \text{ (scenario (a))} \\ &\quad + 2e^{-T_2} \int_0^{T_1} \int_0^{\infty} e^{-x} 3e^{-3y} \frac{1}{3} ((T_1 - x)\mu_1 + T_2\mu_2 + y\mu_3) dy dx \text{ (scenario (b))} \\ &\quad + e^{-T_2} \int_0^{T_1} \int_0^{\infty} e^{-x} 3e^{-3y} \frac{1}{3} ((T_1 - x)\mu_1 + T_2\mu_2 + (2 + y)\mu_3) dy dx \text{ (scenario (c))} \\ &\quad + e^{-T_1} \int_0^{T_2} \int_x^{T_2} 3e^{-3x} e^{-(y-x)} \frac{1}{3} (y - x)\mu_2 dy dx \text{ (scenario (d))} \\ &\quad + 2e^{-T_1} \int_0^{T_2} \int_0^{\infty} 3e^{-3x} e^{-(T_2-x)} 3e^{-3y} \frac{1}{3 \times 3} ((T_2 - x)\mu_2 + y\mu_3) dy dx \text{ (scenario (e))} \\ &\quad + e^{-T_1} \int_0^{T_2} \int_0^{\infty} 3e^{-3x} e^{-(T_2-x)} 3e^{-3y} \frac{1}{3 \times 3} ((T_2 - x)\mu_2 + (2 + y)\mu_3) dy dx \text{ (scenario (f))} \\ &\quad + 2e^{-T_1} e^{-3T_2} \int_0^{\infty} \int_x^{\infty} 6e^{-6x} 3e^{-3(y-x)} \frac{1}{6 \times 3} (2 + y - x)\mu_3 dy dx \text{ (scenario (g))} \\ &\quad + 4e^{-T_1} e^{-3T_2} \int_0^{\infty} \int_x^{\infty} 6e^{-6x} 3e^{-3(y-x)} \frac{1}{6 \times 3} (y - x)\mu_3 dy dx \Big) / \left(1 - \frac{2}{3} e^{-T_1} \right) \text{ (scenario (h))} \\ &= \frac{(e^{-3T_2} + 3e^{-T_2} - 6e^{T_1-T_2})(\mu_2 - \mu_3) + 6(1 - e^{T_1} + T_1e^{T_1})\mu_1}{2(3e^{T_1} - 2)} + \mu_2 \quad (5.5) \end{aligned}$$

Similarly, referring to Figure 5.3, we can compute

$$\begin{aligned}
& \mathbb{E}(L_I(ac|bd)) = \left(e^{-T_1} \int_0^{T_2} \int_x^{T_2} 3e^{-3x} e^{-(y-x)} \frac{1}{3} (y-x) \mu_2 dy dx \right. && \text{(scenario (a))} \\
& + 2e^{-T_1} \int_0^{T_2} \int_0^\infty 3e^{-3x} e^{-(T_2-x)} 3e^{-3y} \frac{1}{3 \times 3} ((T_2-x)\mu_2 + y\mu_3) dy dx \\
& \quad \left. \right. && \text{(scenario (b))} \\
& + e^{-T_1} \int_0^{T_2} \int_0^\infty 3e^{-3x} e^{-(T_2-x)} 3e^{-3y} \frac{1}{3 \times 3} ((T_2-x)\mu_2 + (2+y)\mu_3) dy dx \\
& \quad \left. \right. && \text{(scenario (c))} \\
& + 2e^{-T_1} e^{-3T_2} \int_0^\infty \int_x^\infty 6e^{-6x} 3e^{-3(y-x)} \frac{1}{6 \times 3} (2+y-x) \mu_3 dy dx && \text{(scenario (d))} \\
& + 4e^{-T_1} e^{-3T_2} \int_0^\infty \int_x^\infty 6e^{-6x} 3e^{-3(y-x)} \frac{1}{6 \times 3} (y-x) \mu_3 dy dx \Big) / \left(\frac{1}{3} e^{-T_1} \right) \\
& \quad \left. \right. && \text{(scenario (e))} \\
& = \mu_2 + \frac{1}{2} (\mu_2 - \mu_3) (e^{-3T_2} - 3e^{-T_2}) && \text{(5.6)}
\end{aligned}$$

QED.

Lemma 5.2 (Terminal A or B (cherries), unbalanced). For the unbalanced model species tree of Figure 5.1, the expected length of the terminal edge A (equivalently, B) of a gene tree with an unrooted topology matching the species tree in substitution units is

$$L_A = \mathbb{E}(l_A(ab|cd)) = \frac{6T_1\mu_1 + 3\mu_1 - \mu_2 + e^{-3T_2}(\mu_2 - 2\mu_3)}{6 - 9e^{T_1}} + \mu_1 + \mu_A T_A \quad (5.7)$$

and the expected lengths for gene trees not matching the species tree topology are

$$\begin{aligned}\mathbb{E}(l_A(ad|bc)) &= \left(\frac{1}{6}e^{-3T_2} - \frac{3}{2}e^{-T_2}\right)(\mu_2 - \mu_3) - \frac{2}{3}e^{-3T_2}\mu_3 + \frac{4}{3}\mu_2 + T_1\mu_1 + T_A\mu_A \\ \mathbb{E}(l_A(ac|bd)) &= -\frac{1}{3}e^{-3T_2}(\mu_2 - 2\mu_3) + \frac{1}{3}\mu_2 + T_1\mu_1 + T_A\mu_A\end{aligned}\quad (5.8)$$

and therefore

$$\begin{aligned} L'_A &= \frac{1}{2}(\mathbb{E}(l_A(ad|bc)) + \mathbb{E}(l_A(ac|bd))) \\ &= \frac{1}{12}(10\mu_2 - 9e^{-T_2}(\mu_2 - \mu_3) - 3^{-3T_2}(\mu_2 + \mu_3)) + T_1\mu_1 + T_A\mu_A \end{aligned} \quad (5.9)$$

Proof. Referring to Figure 5.2, we can compute

$$\begin{aligned}
\mathbb{E}(L_A(ab|cd)) &= \left(\int_0^{T_1} e^{-x} (x\mu_1 + T_A\mu_A) dx \right) \quad (\text{scenario (a) to (c)}) \\
&\quad + e^{-T_1} \int_0^{T_2} \int_x^{T_2} 3e^{-3x} e^{-(y-x)} \frac{1}{3} (x\mu_2 + T_1\mu_1 + T_A\mu_A) dy dx \quad (\text{scenario (d)}) \\
&\quad + 3e^{-T_1} \int_0^{T_2} \int_0^{\infty} 3e^{-3x} e^{-(T_2-x)} 3e^{-3y} \frac{1}{3 \times 3} (x\mu_2 + T_1\mu_1 + T_A\mu_A) dy dx \\
&\qquad \qquad \qquad (\text{scenario (e), (f)}) \\
&\quad + 2e^{-T_1} e^{-3T_2} \int_0^{\infty} \int_x^{\infty} 6e^{-6x} 3e^{-3(y-x)} \frac{1}{6 \times 3} (y\mu_3 + T_2\mu_2 + T_1\mu_1 + T_A\mu_A) dy dx \\
&\qquad \qquad \qquad (\text{scenario (g), (h)}) \\
&\quad + e^{-T_1} e^{-3T_2} \int_0^{\infty} \int_x^{\infty} 6e^{-6x} 3e^{-3(y-x)} \frac{1}{6 \times 3} ((y+2)\mu_3 + T_2\mu_2 + T_1\mu_1 + T_A\mu_A) dy dx \\
&\qquad \qquad \qquad (\text{scenario (h)}) \\
&\quad + 3e^{-T_1} e^{-3T_2} \int_0^{\infty} \int_x^{\infty} 6e^{-6x} 3e^{-3(y-x)} \frac{1}{6 \times 3} (x\mu_3 + T_2\mu_2 + T_1\mu_1 + T_A\mu_A) dy dx \Big) \\
&/ \left(1 - \frac{2}{3} e^{-T_1} \right) \quad (\text{scenario (g), (h)}) \\
&= \frac{6T_1\mu_1 + 3\mu_1 - \mu_2 + e^{-3T_2}(\mu_2 - 2\mu_3)}{6 - 9e^{T_1}} + \mu_1 + \mu_A T_A \tag{5.10}
\end{aligned}$$

Referring to Figure 5.3, we can compute

$$\begin{aligned}
\mathbb{E}(L_A(ac|bd)) &= \left(e^{-T_1} \int_0^{T_2} \int_x^{T_2} 3e^{-3x} e^{-(y-x)} \frac{1}{3} (x\mu_2 + T_1\mu_1 + T_A\mu_A) dy dx \right) \quad (\text{scenario (a)}) \\
&\quad + 3e^{-T_1} \int_0^{T_2} \int_0^{\infty} 3e^{-3x} e^{-(T_2-x)} 3e^{-3y} \frac{1}{3 \times 3} (x\mu_2 + T_1\mu_1 + T_A\mu_A) dy dx \\
&\qquad \qquad \qquad (\text{scenario (b), (c)}) \\
&\quad + 2e^{-T_1} e^{-3T_2} \int_0^{\infty} \int_x^{\infty} 6e^{-6x} 3e^{-3(y-x)} \frac{1}{6 \times 3} (y\mu_3 + T_2\mu_2 + T_1\mu_1 + T_A\mu_A) dy dx \\
&\qquad \qquad \qquad (\text{scenario (d), (e)}) \\
&\quad + e^{-T_1} e^{-3T_2} \int_0^{\infty} \int_x^{\infty} 6e^{-6x} 3e^{-3(y-x)} \frac{1}{6 \times 3} ((y+2)\mu_3 + T_2\mu_2 + T_1\mu_1 + T_A\mu_A) dy dx \\
&\qquad \qquad \qquad (\text{scenario (e)}) \\
&\quad + 3e^{-T_1} e^{-3T_2} \int_0^{\infty} \int_x^{\infty} 6e^{-6x} 3e^{-3(y-x)} \frac{1}{6 \times 3} (x\mu_3 + T_2\mu_2 + T_1\mu_1 + T_A\mu_A) dy dx \Big) \\
&/ \left(\frac{1}{3} e^{-T_1} \right) \quad (\text{scenario (d), (e)}) \\
&= -\frac{1}{3} e^{-3T_2} (\mu_2 - 2\mu_3) + \frac{1}{3} \mu_2 + T_1\mu_1 + T_A\mu_A \tag{5.11}
\end{aligned}$$

and

QED.

Lemma 5.3 (Terminal unbalanced C). For the unbalanced species tree, the expected length of the terminal edge C of a gene tree with an unrooted topology matching the species tree in substitution units is

$$L_C = \mathbb{E}(l_C(ab|cd)) = -e^{-T_2}(\mu_2 - \mu_3) + \mu_2 + \mu_C T_C + \frac{2\mu_2 - (3e^{-T_2} - e^{-3T_2})(\mu_2 - \mu_3) - 4\mu_3 e^{-3T_2}}{6(3e^{T_1} - 2)} \quad (5.13)$$

and the expected length for gene trees not matching the species tree **topology** is

$$L'_C = \mathbb{E}(l_C(ac|bd)) = \frac{1}{3}\mu_2(1 + e^{-3T_2}) + \mu_C T_C \quad (5.14)$$

Proof. Referring to Figure 5.2, we can compute

$$\begin{aligned}
\mathbb{E}(L_C(ab|cd)) &= \left(\int_0^{T_1} \int_0^{T_2} e^{-x} e^{-y} (y\mu_2 + T_C\mu_C) dy dx \right. && \text{(scenario (a))} \\
&\quad + e^{-T_2} \int_0^{T_1} \int_0^{\infty} e^{-x} 3e^{-3y} \frac{1}{3} ((y+2)\mu_3 + T_2\mu_2 + T_C\mu_C) dy dx && \text{(scenario (b))} \\
&\quad + 2e^{-T_2} \int_0^{T_1} \int_0^{\infty} e^{-x} 3e^{-3y} \frac{1}{3} (y\mu_3 + T_2\mu_2 + T_C\mu_C) dy dx && \text{(scenario (b), (c))} \\
&\quad + e^{-T_1} \int_0^{T_2} \int_x^{T_2} 3e^{-3x} e^{-(y-x)} \frac{1}{3} (y\mu_2 + T_C\mu_C) dy dx && \text{(scenario (d))} \\
&\quad + e^{-T_1} \int_0^{T_2} \int_0^{\infty} 3e^{-3x} e^{-(T_2-x)} 3e^{-3y} \frac{1}{3 \times 3} ((y+2)\mu_3 + T_2\mu_2 + T_C\mu_C) dy dx && \text{(scenario (e))} \\
&\quad + 2e^{-T_1} \int_0^{T_2} \int_0^{\infty} 3e^{-3x} e^{-(T_2-x)} 3e^{-3y} \frac{1}{3 \times 3} (y\mu_3 + T_2\mu_2 + T_C\mu_C) dy dx && \text{(scenario (e), (f))} \\
&\quad + 2e^{-T_1} e^{-3T_2} \int_0^{\infty} \int_x^{\infty} 6e^{-6x} 3e^{-3(y-x)} \frac{1}{6 \times 3} (y\mu_3 + T_2\mu_2 + T_C\mu_C) dy dx && \text{(scenario (g), (h))} \\
&\quad + e^{-T_1} e^{-3T_2} \int_0^{\infty} \int_x^{\infty} 6e^{-6x} 3e^{-3(y-x)} \frac{1}{6 \times 3} ((y+2)\mu_3 + T_2\mu_2 + T_C\mu_C) dy dx && \text{(scenario (h))} \\
&\quad \left. + 3e^{-T_1} e^{-3T_2} \int_0^{\infty} \int_x^{\infty} 6e^{-6x} 3e^{-3(y-x)} \frac{1}{6 \times 3} (x\mu_3 + T_2\mu_2 + T_C\mu_C) dy dx \right) && \\
&/ \left(1 - \frac{2}{3} e^{-T_1} \right) && \text{(scenario (g), (h))} \\
&= -e^{-T_2} (\mu_2 - \mu_3) + \frac{2\mu_2 - (3e^{-T_2} - e^{-3T_2}) (\mu_2 - \mu_3) - 4\mu_3 e^{-3T_2}}{6(3e^{T_1} - 2)} + \mu_2 + \mu_C T_C && \text{(5.15)}
\end{aligned}$$

and referring to Figure 5.3, we can compute

$$\begin{aligned}
\mathbb{E}(L_C(ac|bd)) &= \mathbb{E}(L_C(ad|bc)) = \\
&= \left(e^{-T_1} \int_0^{T_2} \int_x^{T_2} 3e^{-3x} e^{-(y-x)} \frac{1}{3} (x\mu_2 + T_C\mu_C) dy dx \right. && \text{(scenario (a))} \\
&\quad + 3e^{-T_1} \int_0^{T_2} \int_0^\infty 3e^{-3x} e^{-(T_2-x)} 3e^{-3y} \frac{1}{3} \times \frac{1}{3} (x\mu_2 + T_C\mu_C) dy dx \\
&\quad \left. \right. && \text{(scenario (b), (c))} \\
&\quad + 2e^{-T_1} e^{-3T_2} \int_0^\infty \int_x^\infty 6e^{-6x} 3e^{-3(y-x)} \frac{1}{6 \times 3} (y\mu_3 + T_2\mu_2 + T_C\mu_C) dy dx \\
&\quad \left. \right. && \text{(scenario (d), (e))} \\
&\quad + e^{-T_1} e^{-3T_2} \int_0^\infty \int_x^\infty 6e^{-6x} 3e^{-3(y-x)} \frac{1}{6 \times 3} ((y+2)\mu_3 + T_2\mu_2 + T_C\mu_C) dy dx \\
&\quad \left. \right. && \text{(scenario (e))} \\
&\quad + 3e^{-T_1} e^{-3T_2} \int_0^\infty \int_x^\infty 6e^{-6x} 3e^{-3(y-x)} \frac{1}{6 \times 3} (x\mu_3 + T_2\mu_2 + T_C\mu_C) dy dx \Big) \\
&/ \left(\frac{1}{3} e^{-T_1} \right) && \text{(scenario (d), (e))} \\
&= \frac{1}{3} \mu_2 (1 + e^{-3T_2}) + \mu_C T_C && \text{(5.16)}
\end{aligned}$$

QED.

Lemma 5.4 (Terminal unbalanced D). For the unbalanced species tree, the expected length of the [terminal](#) edge D of a [gene tree](#) with an [unrooted topology](#) matching the species tree in substitution units is

$$L_D = \mathbb{E}(l_D(ab|cd)) = e^{-T_2}(\mu_2 - \mu_3) - \mu_2 + 2\mu_3 + T_2\mu_2 + \mu_D T_D + \frac{-2\mu_2 + (3e^{-T_2} - e^{-3T_2})(\mu_2 - \mu_3)}{6(3e^{T_1} - 2)} \quad (5.17)$$

and the expected length for gene trees not matching the species tree [topology](#) is

$$L'_D = \mathbb{E}(l_D(ad|cb)) = \left(\frac{3}{2} e^{-T_2} - \frac{1}{6} e^{-3T_2} \right) (\mu_2 - \mu_3) - \frac{4}{3} \mu_2 + 2\mu_3 + T_2\mu_2 + \mu_D T_D \quad (5.18)$$

Proof. Referring to Figure 5.2, we can compute

$$\begin{aligned}
\mathbb{E}(L_D(ab|cd)) &= \left(\int_0^{T_1} \int_0^{T_2} e^{-x} e^{-y} ((T_2 - y)\mu_2 + 2\mu_3 + T_D\mu_D) dy dx \right) \quad (\text{scenario (a)}) \\
&\quad + e^{-T_2} \int_0^{T_1} \int_0^{\infty} e^{-x} 3e^{-3y} \frac{1}{3} ((y+2)\mu_3 + T_D\mu_D) dy dx \quad (\text{scenario (b)}) \\
&\quad + 2e^{-T_2} \int_0^{T_1} \int_0^{\infty} e^{-x} 3e^{-3y} \frac{1}{3} (y\mu_3 + T_D\mu_D) dy dx \quad (\text{scenario (b), (c)}) \\
&\quad + e^{-T_1} \int_0^{T_2} \int_x^{T_2} 3e^{-3x} e^{-(y-x)} \frac{1}{3} ((T_2 - y)\mu_2 + 2\mu_3 + T_D\mu_D) dy dx \\
&\qquad\qquad\qquad (\text{scenario (d)}) \\
&\quad + e^{-T_1} \int_0^{T_2} \int_0^{\infty} 3e^{-3x} e^{-(T_2-x)} 3e^{-3y} \frac{1}{3 \times 3} ((y+2)\mu_3 + T_D\mu_D) dy dx \\
&\qquad\qquad\qquad (\text{scenario (e)}) \\
&\quad + 2e^{-T_1} \int_0^{T_2} \int_0^{\infty} 3e^{-3x} e^{-(T_2-x)} 3e^{-3y} \frac{1}{3 \times 3} (y\mu_3 + T_D\mu_D) dy dx \\
&\qquad\qquad\qquad (\text{scenario (e), (f)}) \\
&\quad + 2e^{-T_1} e^{-3T_2} \int_0^{\infty} \int_x^{\infty} 6e^{-6x} 3e^{-3(y-x)} \frac{1}{6 \times 3} (y\mu_3 + T_D\mu_D) dy dx \\
&\qquad\qquad\qquad (\text{scenario (g), (h)}) \\
&\quad + e^{-T_1} e^{-3T_2} \int_0^{\infty} \int_x^{\infty} 6e^{-6x} 3e^{-3(y-x)} \frac{1}{6 \times 3} ((y+2)\mu_3 + T_D\mu_D) dy dx \\
&\qquad\qquad\qquad (\text{scenario (h)}) \\
&\quad + 3e^{-T_1} e^{-3T_2} \int_0^{\infty} \int_x^{\infty} 6e^{-6x} 3e^{-3(y-x)} \frac{1}{6 \times 3} (x\mu_3 + T_D\mu_D) dy dx \Big) / \left(1 - \frac{2}{3} e^{-T_1} \right) \\
&\qquad\qquad\qquad (\text{scenario (g), (h)}) \\
&= e^{-T_2} (\mu_2 - \mu_3) + \frac{-2\mu_2 + (3e^{-T_2} - e^{-3T_2}) (\mu_2 - \mu_3)}{6(3e^{T_1} - 2)} - \mu_2 + 2\mu_3 + T_2\mu_2 + \mu_D T_D
\end{aligned} \tag{5.19}$$

Referring to Figure 5.3, we can compute

$$\begin{aligned}
\mathbb{E}(L_D(ac|bd)) &= \mathbb{E}(L_D(ad|bc)) = \\
&= \left(e^{-T_1} \int_0^{T_2} \int_x^{T_2} 3e^{-3x} e^{-(y-x)} \frac{1}{3} ((T_2 - y)\mu_2 + 2\mu_3 + T_D\mu_D) dy dx \right. \\
&\quad \left. \text{(scenario (a))} \right) \\
&+ e^{-T_1} \int_0^{T_2} \int_0^\infty 3e^{-3x} e^{-(T_2-x)} 3e^{-3y} \frac{1}{3 \times 3} ((y+2)\mu_3 + T_D\mu_D) dy dx \\
&\quad \text{(scenario (b))} \\
&+ 2e^{-T_1} \int_0^{T_2} \int_0^\infty 3e^{-3x} e^{-(T_2-x)} 3e^{-3y} \frac{1}{3 \times 3} (y\mu_3 + T_D\mu_D) dy dx \\
&\quad \text{(scenario (b), (c))} \\
&+ 2e^{-T_1} e^{-3T_2} \int_0^\infty \int_x^\infty 6e^{-6x} 3e^{-3(y-x)} \frac{1}{6 \times 3} (y\mu_3 + T_D\mu_D) dy dx \\
&\quad \text{(scenario (d), (e))} \\
&+ e^{-T_1} e^{-3T_2} \int_0^\infty \int_x^\infty 6e^{-6x} 3e^{-3(y-x)} \frac{1}{6 \times 3} ((y+2)\mu_3 + T_D\mu_D) dy dx \\
&\quad \text{(scenario (e))} \\
&+ 3e^{-T_1} e^{-3T_2} \int_0^\infty \int_x^\infty 6e^{-6x} 3e^{-3(y-x)} \frac{1}{6 \times 3} (x\mu_3 + T_D\mu_D) dy dx \Big) / \left(\frac{1}{3} e^{-T_1} \right) \\
&\quad \text{(scenario (d), (e))} \\
&= \left(\frac{3}{2} e^{-T_2} - \frac{1}{6} e^{-3T_2} \right) (\mu_2 - \mu_3) - \frac{4}{3} \mu_2 + 2\mu_3 + T_2\mu_2 + \mu_D T_D
\end{aligned} \tag{5.20}$$

QED.

Theorem 5.1 follows from Lemmas 5.1-5.4.

Theorem 5.1 (Unbalanced). For the unbalanced species tree of Figure 5.1a, let Δ_I be the difference in the expected internal branch length in substitution units of gene trees with an unrooted topology matching the species tree and those not matching the species tree. Then,

$$\Delta_I = \frac{3(e^{-T_2} - e^{-3T_2})(1 - e^{-T_1})(\mu_2 - \mu_3) + 6\mu_1(e^{-T_1} - 1 + T_1)}{2(3 - 2e^{-T_1})} \tag{5.21}$$

Similarly, let Δ_A , Δ_C , and Δ_D be the difference in the expected length of matching and non-matching gene trees for the terminal branch leading to a cherry, the middle terminal

branch, and the root-adjacent **terminal** branch, respectively.

$$\Delta_A = \frac{4\mu_2 - 6\mu_1 - (3e^{-T_2} + e^{-3T_2} + \frac{9}{2}e^{T_1-T_2})(\mu_2 - \mu_3)}{2(-2 + 3e^{T_1})} + \frac{e^{T_1} (\frac{1}{2}(\mu_2 + \mu_3)e^{-3T_2} + 6\mu_1(1 - T_1) - 5\mu_2)}{2(-2 + 3e^{T_1})} \quad (5.22)$$

$$\Delta_C = \frac{(2 - e^{-T_1}) ((e^{-3T_2} + 2)\mu_2 - 3e^{-T_2}(\mu_2 - \mu_3))}{2(3 - 2e^{-T_1})} + \frac{\mu_3 e^{-3T_2}(e^{-T_1} - 4)}{2(3 - 2e^{-T_1})} \quad (5.23)$$

$$\Delta_D = \frac{(1 - e^{-T_1})(2\mu_2 - (3e^{-T_2} - e^{-3T_2})(\mu_2 - \mu_3))}{2(3 - 2e^{-T_1})} \quad (5.24)$$

Proof. Subtracting Eq. 5.6 from Eq. 5.5, we get

$$\Delta_I = \mathbb{E}(L_I(ab|cd)) - \mathbb{E}(L_I(ac|bd)) = \frac{3(e^{-T_2} - e^{-3T_2})(1 - e^{-T_1})(\mu_2 - \mu_3) + 6\mu_1(e^{-T_1} - 1 + T_1)}{2(3 - 2e^{-T_1})} \quad (5.25)$$

The average length of the **terminal** branch of A in a **gene tree** not matching the **topology** of the species tree is therefore the average of Eq. 5.12 and Eq. 5.11

$$\begin{aligned} & \frac{1}{2}(\mathbb{E}(L_A(ad|bc)) + \mathbb{E}(L_A(ac|bd))) \\ &= \frac{1}{12}(10\mu_2 - 9e^{-T_2}(\mu_2 - \mu_3) - 3^{-3T_2}(\mu_2 + \mu_3)) + T_1\mu_1 + T_A\mu_A \end{aligned} \quad (5.26)$$

Subtracting Eq. 5.26 from Eq. 5.10, we get

$$\Delta_A = \mathbb{E}(L_A(ab|cd)) - \frac{1}{2}(\mathbb{E}(L_A(ad|bc)) + \mathbb{E}(L_A(ac|bd))) \quad (5.27)$$

$$\begin{aligned} &= \frac{4\mu_2 - 6\mu_1 - (3e^{-T_2} + e^{-3T_2})(\mu_2 - \mu_3)}{2(-2 + 3e^{T_1})} \\ &+ \frac{e^{T_1} (\frac{1}{2}(\mu_2 + \mu_3)e^{-3T_2} + \frac{9}{2}e^{-T_2}(\mu_2 - \mu_3) - 6T_1\mu_1 + 6\mu_1 - 5\mu_2)}{2(-2 + 3e^{T_1})} \end{aligned} \quad (5.28)$$

Subtracting Eq. 5.16 from Eq. 5.15, we get

$$\begin{aligned} \Delta_C &= \mathbb{E}(L_C(ab|cd)) - \mathbb{E}(L_C(ac|bd)) \\ &= \frac{(2 - e^{-T_1}) ((e^{-3T_2} + 2)\mu_2 - 3e^{-T_2}(\mu_2 - \mu_3)) + \mu_3 e^{-3T_2}(e^{-T_1} - 4)}{2(3 - 2e^{-T_1})} \end{aligned} \quad (5.29)$$

Subtracting Eq. 5.20 from Eq. 5.19, we get

$$\Delta_D = \mathbb{E}(L_D(ab|cd)) - \mathbb{E}(L_D(ac|bd)) = \frac{(1 - e^{-T_1})(2\mu_2 - (3e^{-T_2} - e^{-3T_2})(\mu_2 - \mu_3))}{2(3 - 2e^{-T_1})} \quad (5.30)$$

QED.

Simplifications for unbalanced equations. We simplify the equations of Theorem 5.1 by computing their limit as $T_2 \rightarrow \infty$ or $\mu_2 \rightarrow \mu_3$. Note that neither assumption completely breaks non-ultrametricity assumptions because we ignore the rate for only one branch (μ_2) and not the others.

Internal branch calculation. To compute $t_1 = \mu_1 T_1$, we simplify Equation (5.21) so that it only depends on T_1 and μ_1 , by computing its limits:

$$\lim_{T_2 \rightarrow \infty} \Delta_I = \lim_{T_2 \rightarrow 0} \Delta_I = \lim_{\mu_2 \rightarrow \mu_3} \Delta_I = \frac{3\mu_1(e^{-T_1} - 1 + T_1)}{3 - 2e^{-T_1}} \quad (5.31)$$

Replacing Δ_I with the observed difference between mean **internal** branches among matching and non-matching gene trees ($\bar{\Delta}_I$), we get an equation with two unknowns, μ_1 and T_1 . One way to move forward is to estimate T_1 using quartet discordance, as shown by [295]. Then, we can estimate μ_1 and thus $t_1 = \mu_1 \times T_1$. However, the accuracy of the **CU** estimate of T_1 is known to degrade for inaccurate gene trees [295]. Instead, we use a local clock approximation to estimate μ_1 and then solve for T_1 . If mutation rates of the two branches above the focal branch are assumed the same (e.g., $\mu_2 = \mu_3$), then, the expected length of gene trees not matching the species tree is simply μ_2 by Theorem 5.1. Further assuming $\mu_1 = \mu_2$ allows us to estimate μ_1 as the mean length of the **internal** quartet branch among gene trees not matching the species tree ($\mu_1 = \bar{L}'_I$), obtaining:

$$\frac{\bar{\Delta}_I}{\bar{L}'_I} = \frac{3(T_1 + e^{-T_1} - 1)}{3 - 2e^{-T_1}} \quad (5.32)$$

The solution to this equation is:

$$\bar{\delta} + W\left(-\frac{1}{3}e^{-\bar{\delta}-1}(2\bar{\delta} + 3)\right) + 1 \quad (5.33)$$

where $W(\cdot)$ is the Lambert W function and $\bar{\delta} = \bar{\Delta}_I/\bar{L}'_I$. Since Lambert's function does not have a closed-form solution (and can be imaginary), we resort to the Taylor expansion

$e^{T_1} \approx 1 + T_1$, which is a good approximation for small T_1 . Using this approximation, the solution to Equation (6.1) becomes:

$$\hat{T}_1 = \frac{1}{2}\bar{\delta} + \frac{1}{6}\sqrt{3\bar{\delta}(3\bar{\delta} + 4)} \quad (5.34)$$

In our current implementation of CASTLES, we use this approximation to avoid numerical issues. When $\delta < 0$, we set the branch length to a small value (10^{-6} by default).

Terminal branch calculation. To simplify Equation (5.22), we compute its limit as $T_2 \rightarrow \infty$

$$\lim_{T_2 \rightarrow \infty} \Delta_A = \frac{-6\mu_1(e^{-T_1} - 1 + T_1) - (5 - 4e^{-T_1})\mu_2}{6 - 4e^{-T_1}}. \quad (5.35)$$

The expected length of the terminal branch of A in non-matching gene trees in the limit is

$$\lim_{T_2 \rightarrow \infty} L'_A = T_1\mu_1 + T_A\mu_A + \frac{5}{6}\mu_2 \quad (5.36)$$

based on Equation (5.26). To compute $t_A = \mu_A T_A$ we replace the expected value $\lim_{T_2 \rightarrow \infty} \Delta_A$ in Equation (5.35) with the observed mean difference $\bar{\Delta}_A$ and replace the expected value $\lim_{T_2 \rightarrow \infty} L'_A$ in Equation (5.36) with the observed mean terminal branch of A among non-matching gene trees (\bar{L}'_A). Solving for $T_A\mu_A$ gives us an estimator of t_A :

$$\hat{t}_A = \bar{L}'_A + \frac{\mu_1(e^{-T_1} - 1 + T_1) + \bar{\Delta}_A(1 - 2/3e^{-T_1})}{1 - 4/5e^{-T_1}} - T_1\mu_1. \quad (5.37)$$

Similarly, for branch C ,

$$\lim_{T_2 \rightarrow \infty} \Delta_C = \frac{\mu_2(2 - e^{-T_1})}{(3 - 2e^{-T_1})} \text{ and } \lim_{T_2 \rightarrow \infty} L'_C = \frac{1}{3}\mu_2 + T_C\mu_C \quad (5.38)$$

where the expected length of C in non-matching gene trees (L'_C) is given in Equation (5.14). Replacing $\lim_{T_2 \rightarrow \infty} L'_C$ with the observed length of C in non-matching gene trees \bar{L}'_C and replacing $\lim_{T_2 \rightarrow \infty} \Delta_C$ with the observed $\bar{\Delta}_C$ in Equation (5.38) gives us the estimate for $t_C = T_C\mu_C$:

$$\hat{t}_C = \bar{L}'_C - \frac{1}{3}(2 - \frac{1}{2 - e^{-T_1}})\bar{\Delta}_C. \quad (5.39)$$

For D , we use a different limit:

$$\begin{aligned}\lim_{\mu_2 \rightarrow \mu_3} \Delta_D &= \frac{\mu_2 (1 - e^{-T_1})}{3 - 2e^{-T_1}} \\ \lim_{\mu_2 \rightarrow \mu_3} L'_D &= \mu_2 T_2 + \mu_D T_D + \frac{2}{3} \mu_2\end{aligned}\quad (5.40)$$

where the expected length of D in non-matching gene trees (L'_D) is given in Eq. (5.20). The pendant branch of D in **SU** in the *unrooted* species tree is $\mu_2 T_2 + \mu_D T_D$, representing both branches below the root. Substituting expected values Δ_D and L'_D with observed values $\bar{\Delta}_D$ and \bar{L}'_D , we get our estimate:

$$\hat{t}_2 + \hat{t}_D = \bar{L}'_D - \frac{2}{3} \left(2 + \frac{1}{1 - e^{-T_1}} \right) \bar{\Delta}_D . \quad (5.41)$$

To summarize, we use equations (5.37), (5.39), and (5.41) to compute **terminal** branch lengths, setting the length to a small value (10^{-6} by default) when results are negative.

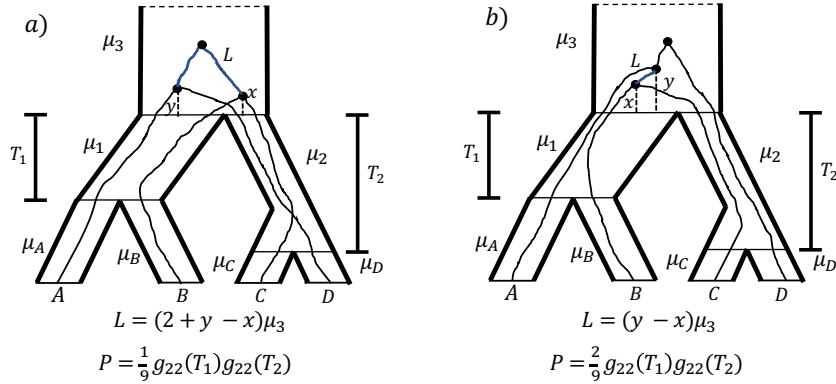


Figure 5.4: **Scenarios for gene tree not matching the balanced species tree.** internal branch lengths for unrooted quartet gene tree not matching the balanced model species tree $((A, B) : T_1, (C, D) : T_2)$. L denotes the internal branch length in the gene tree and P denotes the probability of each case. Case (b) corresponds to four different scenarios.

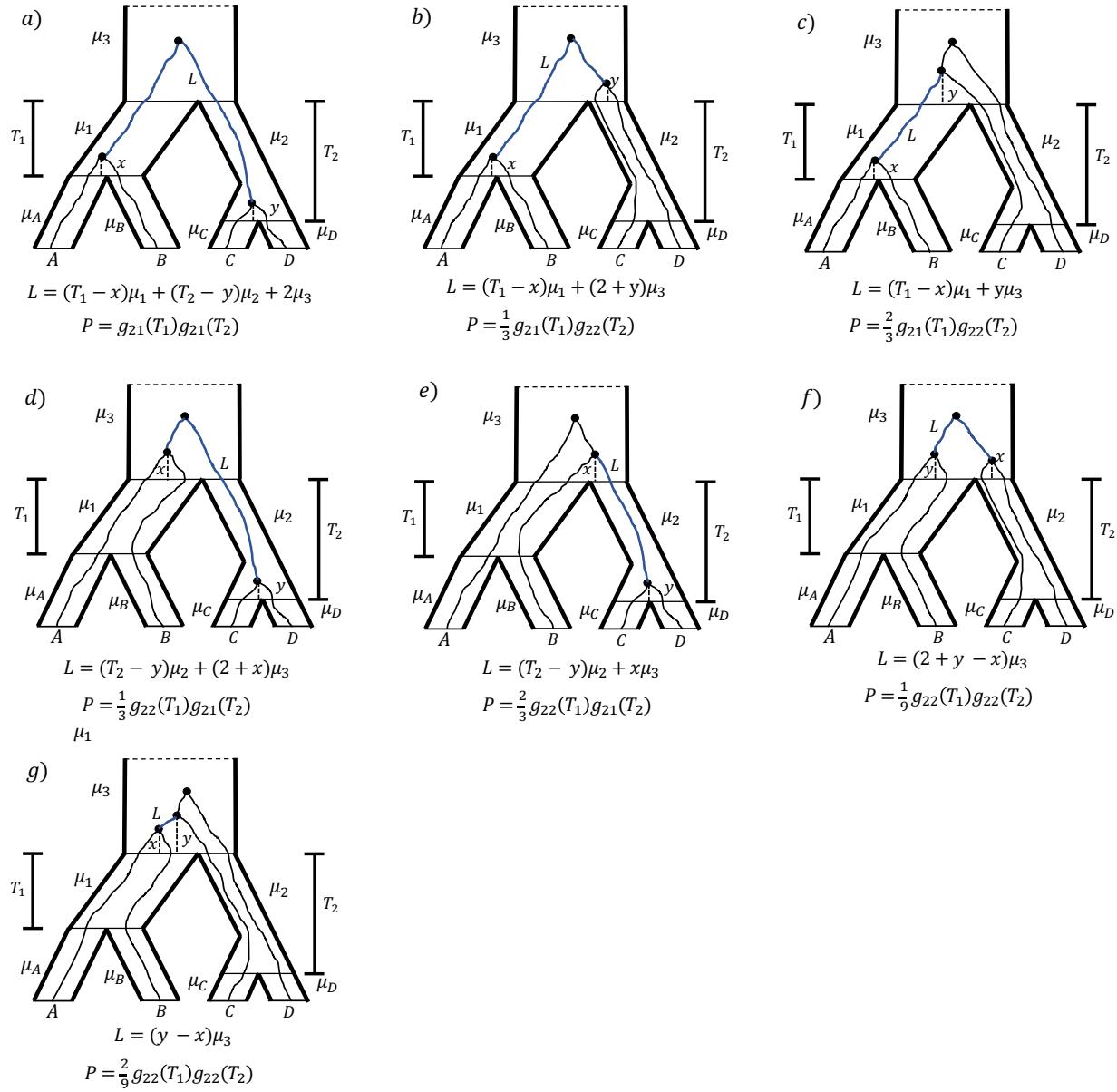


Figure 5.5: Scenarios for gene tree matching the balanced species tree. internal branch lengths for unrooted quartet gene tree matching the balanced model species tree $((A, B) : T_1, (C, D) : T_2)$. Here, L denotes the internal branch length in the gene tree and P denotes the probability of each case. Case (c) and (e) correspond to two scenarios and case (g) corresponds to four different scenarios, and the P values reported in these cases show the overall probability of all possible scenarios for that case.

Balanced species tree. Similar to Theorem 5.1 for unbalanced trees, we first introduce and prove two lemmas, and then follow up with Theorem 5.2 for balanced trees. All of our results and their proofs are in reference to Figures 5.5 and 5.4.

Lemma 5.5 (Internal balanced). For the unbalanced model species tree of Figure 5.1, the expected length of the **internal** branch of a **gene tree** with an **unrooted topology** matching the species tree in substitution units is

$$L_I = \mathbb{E}(l_I(ab|cd)) = \frac{3e^{-T_1}(\mu_1 - \mu_3) + \mu_3 e^{-(T_1+T_2)} + 3e^{-T_2}(\mu_2 - \mu_3) + 3(T_1\mu_1 + T_2\mu_2 - \mu_1 - \mu_2 + 2\mu_3)}{3 - 2e^{-(T_1+T_2)}} \quad (5.42)$$

and the expected length for gene trees not matching the species tree **topology** is

$$L'_I = \mathbb{E}(l_I(ac|bd)) = \mathbb{E}(l_I(ad|bc)) = \mu_3 \quad (5.43)$$

Proof. Referring to Figure 5.5, we can compute

$$\begin{aligned} \mathbb{E}(L_I(ab|cd)) &= \left(\int_0^{T_1} \int_0^{T_2} e^{-x} e^{-y} ((T_1 - x)\mu_1 + (T_2 - y)\mu_2 + 2\mu_3) dy dx \right. && \text{(scenario (a))} \\ &\quad + e^{-T_2} \int_0^{T_1} \int_0^{\infty} e^{-x} 3e^{-3y} \frac{1}{3} ((T_1 - x)\mu_1 + (2 + y)\mu_3) dy dx && \text{(scenario (b))} \\ &\quad + e^{-T_2} \int_0^{T_1} \int_0^{\infty} e^{-x} 3e^{-3y} \frac{2}{3} ((T_1 - x)\mu_1 + y\mu_3) dy dx && \text{(scenario (c))} \\ &\quad + e^{-T_1} \int_0^{\infty} \int_0^{T_2} 3e^{-3x} e^{-y} \frac{1}{3} ((T_2 - y)\mu_2 + (2 + x)\mu_3) dy dx && \text{(scenario (d))} \\ &\quad + e^{-T_1} \int_0^{\infty} \int_0^{T_2} 3e^{-3x} e^{-y} \frac{2}{3} ((T_2 - y)\mu_2 + x\mu_3) dy dx && \text{(scenario (e))} \\ &\quad + 2e^{-T_1} e^{-T_2} \int_0^{\infty} \int_x^{\infty} 6e^{-6x} 3e^{-3(y-x)} \frac{1}{6 \times 3} (2 + y - x)\mu_3 dy dx && \text{(scenario (f))} \\ &\quad \left. + 4e^{-T_1} e^{-T_2} \int_0^{\infty} \int_x^{\infty} 6e^{-6x} 3e^{-3(y-x)} \frac{1}{6 \times 3} (y - x)\mu_3 dy dx \right) / \left(1 - \frac{2}{3} e^{-(T_1+T_2)} \right) && \text{(scenario (g))} \\ &= \frac{3e^{-T_1}(\mu_1 - \mu_3) + \mu_3 e^{-(T_1+T_2)} + 3e^{-T_2}(\mu_2 - \mu_3) + 3(T_1\mu_1 + T_2\mu_2 - \mu_1 - \mu_2 + 2\mu_3)}{3 - 2e^{-(T_1+T_2)}} \end{aligned} \quad (5.44)$$

For a balanced model tree, there are only two scenarios for a **gene tree** not matching the species tree, shown in Figure 5.4. The expected length of the **internal** branch in this case is

$$\begin{aligned}
\mathbb{E}(L_I(ad|bc)) &= \mathbb{E}(L_I(ac|bd)) = \\
&\left(2e^{-T_1}e^{-T_2} \int_0^\infty \int_x^\infty 6e^{-6x} 3e^{-3(y-x)} \frac{1}{6 \times 3} (2+y-x)\mu_3 dy dx \right) \text{ (scenario (a))} \\
&+ 4e^{-T_1}e^{-T_2} \int_0^\infty \int_x^\infty 6e^{-6x} 3e^{-3(y-x)} \frac{1}{6 \times 3} (y-x)\mu_3 dy dx \Big) / \left(\frac{1}{3} e^{-(T_1+T_2)} \right) \\
&\quad \text{ (scenario (b))} \\
&= \mu_3
\end{aligned} \tag{5.45}$$

QED.

Lemma 5.6 (Terminal (cherries), balanced). For the unbalanced model species tree of Figure 5.1, the expected length of the [terminal](#) edge A (equivalently, B) of a [gene tree](#) with an [unrooted topology](#) matching the species tree in substitution units is

$$L_A = \mathbb{E}(l_A(ab|cd)) = \frac{e^{-(T_1+T_2)}(-6T_1\mu_1 - 7\mu_3) + 9((1 - e^{-T_1})\mu_1 + \mu_3 e^{-T_1})}{9 - 6e^{-(T_1+T_2)}} + \mu_A T_A \tag{5.46}$$

and the expected lengths for gene trees not matching the species tree [topology](#) are

$$L'_A = \mathbb{E}(l_A(ad|bc)) = \mathbb{E}(l_A(ac|bd)) = T_1\mu_1 + \frac{2}{3}\mu_3 + \mu_A T_A \tag{5.47}$$

Proof. Referring to Figure 5.4, we can compute

$$\begin{aligned}
\mathbb{E}(L_A(ad|bc)) &= \mathbb{E}(L_A(ac|bd)) = \\
&\left(2e^{-T_1}e^{-T_2} \int_0^\infty \int_x^\infty 6e^{-6x} 3e^{-3(y-x)} \frac{1}{6 \times 3} (y\mu_3 + T_1\mu_1 + T_A\mu_A) dy dx \right. \\
&\quad \text{ (scenario (a), (b))} \\
&+ e^{-T_1}e^{-T_2} \int_0^\infty \int_x^\infty 6e^{-6x} 3e^{-3(y-x)} \frac{1}{6 \times 3} ((y+2)\mu_3 + T_1\mu_1 + T_A\mu_A) dy dx \\
&\quad \text{ (scenario (b))} \\
&+ 3e^{-T_1}e^{-T_2} \int_0^\infty \int_x^\infty 6e^{-6x} 3e^{-3(y-x)} \frac{1}{6 \times 3} (x\mu_3 + T_1\mu_1 + T_A\mu_A) dy dx \Big) \\
&/ \left(\frac{1}{3} e^{-(T_1+T_2)} \right) \text{ (scenario (a), (b))} \\
&= T_1\mu_1 + \frac{2}{3}\mu_3 + \mu_A T_A
\end{aligned} \tag{5.48}$$

Similarly, referring to Figure 5.5, we can compute

$$\begin{aligned}
\mathbb{E}(L_A(ab|cd)) &= \left(\int_0^{T_1} \int_0^{T_2} e^{-x} e^{-y} (x\mu_1 + T_A\mu_A) dy dx \right. && \text{(scenario (a))} \\
&\quad + e^{-T_2} \int_0^{T_1} \int_0^{\infty} e^{-x} 3e^{-3y} \frac{1}{3} (x\mu_1 + T_A\mu_A) dy dx && \text{(scenario (b))} \\
&\quad + e^{-T_2} \int_0^{T_1} \int_0^{\infty} e^{-x} 3e^{-3y} \frac{2}{3} (x\mu_1 + T_A\mu_A) dy dx && \text{(scenario (c))} \\
&\quad + 2e^{-T_1} \int_0^{\infty} \int_0^{T_2} 3e^{-3x} e^{-y} \frac{1}{3} (x\mu_3 + T_1\mu_1 + T_A\mu_A) dy dx && \text{(scenario (d), (e))} \\
&\quad + e^{-T_1} \int_0^{\infty} \int_0^{T_2} 3e^{-3x} e^{-y} \frac{1}{3} ((x+2)\mu_3 + T_1\mu_1 + T_A\mu_A) dy dx && \text{(scenario (e))} \\
&\quad + 2e^{-T_1} e^{-T_2} \int_0^{\infty} \int_x^{\infty} 6e^{-6x} 3e^{-3(y-x)} \frac{1}{6 \times 3} (y\mu_3 + T_1\mu_1 + T_A\mu_A) dy dx && \text{(scenario (f), (g))} \\
&\quad + e^{-T_1} e^{-T_2} \int_0^{\infty} \int_x^{\infty} 6e^{-6x} 3e^{-3(y-x)} \frac{1}{6 \times 3} ((y+2)\mu_3 + T_1\mu_1 + T_A\mu_A) dy dx && \text{(scenario (g))} \\
&\quad \left. + 3e^{-T_1} e^{-T_2} \int_0^{\infty} \int_x^{\infty} 6e^{-6x} 3e^{-3(y-x)} \frac{1}{6 \times 3} (x\mu_3 + T_1\mu_1 + T_A\mu_A) dy dx \right) \\
&/ \left(1 - \frac{2}{3} e^{-(T_1+T_2)} \right) && \text{(scenario (f), (g))} \\
&= \frac{e^{-(T_1+T_2)}(-6T_1\mu_1 - 7\mu_3) + 9((1 - e^{-T_1})\mu_1 + \mu_3 e^{-T_1})}{9 - 6e^{-(T_1+T_2)}} + \mu_A T_A && (5.49)
\end{aligned}$$

QED.

Using these lemmas, we now prove Theorem 5.2.

Theorem 5.2 (Balanced). For the balanced model species tree of Figure 5.1, let Δ_I be the difference in the expected internal branch length in substitution units of gene trees with an unrooted topology matching the species tree and those not matching the species tree. Then,

$$\Delta_I = \frac{3(e^{-T_1}(\mu_1 - \mu_3) + \mu_3 e^{-(T_1+T_2)} + e^{-T_2}(\mu_2 - \mu_3) + (T_1\mu_1 + T_2\mu_2 - \mu_1 - \mu_2 + \mu_3))}{3 - 2e^{-(T_1+T_2)}} \quad (5.50)$$

Similarly, let Δ_A be the difference in the expected length of matching and non-matching

gene trees for the **terminal** branch leading to a cherry A .

$$\Delta_A = \frac{-\mu_3 e^{-(T_1+T_2)} + 3\mu_1(1 - e^{-T_1} - T_1) + \mu_3(-2 + 3e^{-T_1})}{3 - 2e^{-(T_1+T_2)}} \quad (5.51)$$

Note that since all **taxa** in a balanced quartet tree are part of a cherry, B, C and D would follow similar equations as (5.51) by substituting the appropriate μ and T values, following the symmetry of the tree.

Proof. Subtracting Eq. 5.45 from Eq. 5.44, we get

$$\Delta_I = \mathbb{E}(L_I(ab|cd)) - \mathbb{E}(L_I(ac|bd)) = \mathbb{E}(L_I(ab|cd)) - \mathbb{E}(L_I(ad|bc)) \quad (5.52)$$

$$= \frac{3(e^{-T_1}(\mu_1 - \mu_3) + \mu_3 e^{-(T_1+T_2)} + e^{-T_2}(\mu_2 - \mu_3) + (T_1\mu_1 + T_2\mu_2 - \mu_1 - \mu_2 + \mu_3))}{3 - 2e^{-(T_1+T_2)}} \quad (5.53)$$

Subtracting Eq. 5.48 from Eq. 5.49, we get

$$\Delta_A = \mathbb{E}(L_A(ab|cd)) - \mathbb{E}(L_A(ad|bc)) = \mathbb{E}(L_A(ab|cd)) - \mathbb{E}(L_A(ac|bd)) \quad (5.54)$$

$$= \frac{-\mu_3 e^{-(T_1+T_2)} + 3\mu_1(1 - e^{-T_1} - T_1) + \mu_3(-2 + 3e^{-T_1})}{-2e^{-(T_1+T_2)} + 3} \quad (5.55)$$

QED.

Simplifications for balanced equations. Similar to the simplifications for the unbalanced tree equations, we simplify equations of Theorem 5.2 by computing their limit as $T_2 \rightarrow 0$ or $\mu_3 \rightarrow \mu_1$.

To compute the length of the **internal** branch, i.e. $t_1 + t_2$, we simplify Equation 5.50 by computing its limit as $T_2 \rightarrow 0$ (note that by symmetry, $T_1 \rightarrow 0$ can also be used and will lead to the same final formula).

$$\lim_{T_2 \rightarrow 0} \Delta_I = \frac{3\mu_1(e^{-T_1} - 1 + T_1)}{3 - 2e^{-T_1}} \quad (5.56)$$

which is exactly the same as Equation 5.31 for unbalanced trees. Note that by Equation 5.43, we have $\bar{L}'_I = \mu_3$, therefore, further assuming $\mu_1 = \mu_3$ allows us to estimate μ_1 as the average length of the **internal** branch in non-matching gene trees. We replace Δ_I with the observed difference between average **internal** branch lengths in matching and non-matching gene trees in Equation 5.56, leading to the same equation as Figure 6.1 for unbalanced trees. The solution to this equation is based on the Lambert W function, which we approximate using a Taylor approximation as in Equation 5.34, leading to the following formula:

$$\hat{T}_1\hat{\mu}_1 = \bar{L}'_I \left(\frac{1}{2}\bar{\delta} + \frac{1}{6}\sqrt{3\bar{\delta}(3\bar{\delta}+4)} \right) \quad (5.57)$$

where $\bar{\delta} = \frac{\bar{\Delta}_I}{\bar{L}'_I}$. Note that since we initially assumed $T_2 \rightarrow 0$, the length of the **internal** branch in the limit only depends on $\hat{T}_1\hat{\mu}_1$, and we use Equation 5.57 as an estimate of $t_1 + t_2$ (the same estimator can be derived using $T_1 \rightarrow 0$).

For the **terminal** branch of A , assuming $\mu_3 \rightarrow \mu_1$, we can simplify Equation ??, as follows:

$$\lim_{\mu_3 \rightarrow \mu_1} \Delta_A = \frac{\mu_1 (-e^{-(T_1+T_2)} + 1 - 3T_1)}{-2e^{-(T_1+T_2)} + 3} \quad (5.58)$$

and

$$\lim_{\mu_3 \rightarrow \mu_1} L'_A = T_1\mu_1 + \frac{2}{3}\mu_1 + \mu_A T_A \quad (5.59)$$

We replace the expected value $\lim_{\mu_3 \rightarrow \mu_1} L'_A$ in Equation 5.59 with the observed mean difference $\bar{\Delta}_A$ and $\lim_{\mu_3 \rightarrow \mu_1} L'_A$ with the observed mean \bar{L}'_A . Solving for $\mu_A T_A$ gives the following estimate for t_A :

$$\hat{t}_A = \bar{L}'_A - \frac{2}{3}\mu_1 - \frac{1}{3}(\mu_1(1 - e^{-(T_1+T_2)}) - \bar{\Delta}_A(3 - 2e^{-(T_1+T_2)})) \quad (5.60)$$

Note that due to symmetry, all nodes in a balanced quartet are part of a cherry, and therefore the same equations and simplifications can be used for all **terminal** branches, only replacing the appropriate T s and μ s. In particular, for C and D , we use a different assumption $\mu_3 \rightarrow \mu_2$, so that the final equation depends on T_2 and μ_2 , i.e. parameters of the branch above the cherry. Tables 5.1, 5.2 and 5.3 summarize the expected lengths, simplified formulas and the final branch length estimators for both unbalanced and balanced trees.

5.3 CASTLES

To extend the algorithm to more than four species, we apply the same calculations to each branch of the species tree, one at a time. Each internal branch of the species tree creates a quadripartition of species (e.g., $A, B|C, D$ in Figure 5.1b). Any quartet of species (e.g., $ab|cd$) with a selection of one taxon from each part of the quadripartition ($a \in A$, $b \in B$, $c \in C$, and $d \in D$) gives us a quartet species tree where all of our previous theoretical results

Algorithm 5.1 CASTLES algorithm. The input is a rooted species tree s with $n > 4$ taxa and a set of gene trees \mathcal{G} with SU branch lengths, and the output is s annotated with SU branch lengths. t_e denotes the length of branch e in SU.

```

1: procedure CASTLES( $s, \mathcal{G}$ )
2:    $\bar{L}_a, \bar{L}_b, \bar{L}_v, \bar{L}_p, \bar{L}'_a, \bar{L}'_b, \bar{L}'_v, \bar{L}'_p$  for each branch  $\leftarrow$  Alg. 5.2
3:   for  $u \in$  post order traverse of internal nodes of  $s$  do
4:     if  $u$  is root then
5:       break
6:     end if
7:      $p \leftarrow parent(u); v \leftarrow sibling(u); a, b \leftarrow children(u)$ 
8:     if  $p$  is root then
9:       if  $v$  is leaf then
10:         $t_{p \rightarrow u} + t_{p \rightarrow v} \leftarrow$  Eq. (5.41) (terminal D)
11:      else
12:         $t_{p \rightarrow u} + t_{p \rightarrow v} \leftarrow$  Eq. (5.57) (internal bal.)
13:      end if
14:    else
15:       $t_{p \rightarrow u} \leftarrow$  Eq. (5.34) (internal unbal.)
16:      if  $v$  is leaf then
17:         $t_{p \rightarrow v} \leftarrow$  Eq. (5.39) (terminal C)
18:      end if
19:      for  $w \in children(u)$  do
20:        if  $w$  is leaf and  $t_{u \rightarrow w}$  is null then
21:           $t_{u \rightarrow w} \leftarrow$  Eq. (5.37) (terminal A)
22:        end if
23:      end for
24:    end if
25:  end for
26: end procedure

```

hold, and they all lead to identical expected values for their corresponding gene tree quartets. Thus, it is valid to compute the length of this species tree branch using the quartet-based approach by simply taking the average lengths across all quartets.

Assuming the averages are already calculated, we can use Figure 5.1 to assign a length to each branch. The algorithm visits the internal nodes of the tree in a post-order traversal. For each internal node, it assigns the length of the edge above, in addition to (some of the) adjacent terminal branches. If a node u is the parent of a cherry, it assigns the length to both children; otherwise, it ignores the children. If u is sister to a leaf, it also assigns the length to the sister, using Equation (5.39). When the tree has more than four taxa, almost all branch lengths are assigned using unbalanced quartet equations. The only exception is the root branch, which may need to be set based on the balanced quartet equations (Figure

5.6).

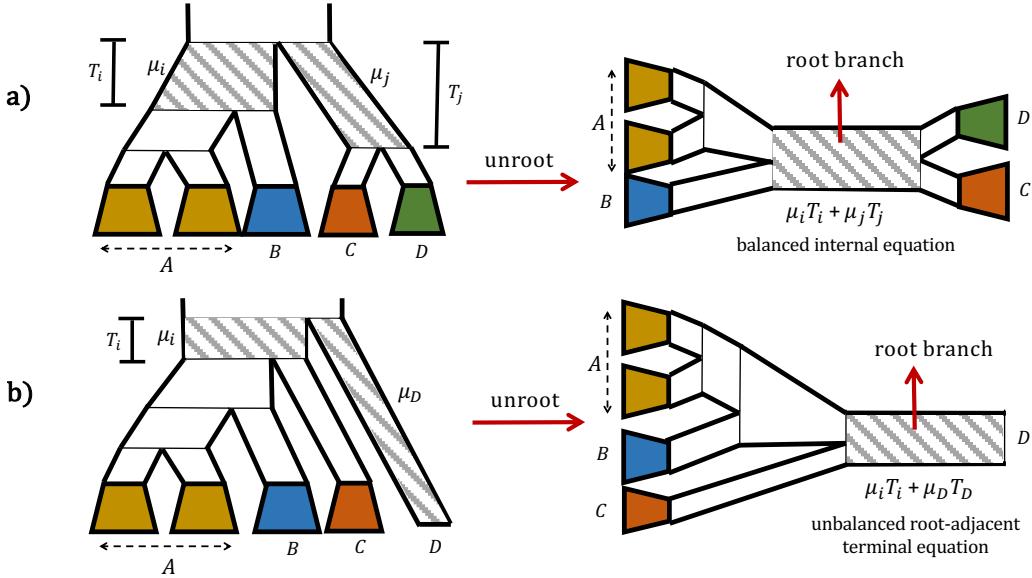


Figure 5.6: Root branch calculation based on balanced or unbalanced quartets. The root branch in the unrooted species tree corresponds to two branches in the rooted tree (the highlighted branches). Our equations calculate the *sum* of the two root-adjacent branches, that correspond to a single branch in the unrooted species tree. Hence, the length of the individual parts of this branch, and therefore the exact position of the root, is not inferred in our approach. a) When none of the children of the root is a leaf, then the length of the root branch can be computed from balanced quartets, using the *internal* branch equation. b) When one child of the root is a leaf, the root branch length can be calculated using the equations for the root-adjacent *terminal* branch (referred to as *terminal D* in the proofs) from unbalanced quartets.

The most challenging part of this algorithm, then, is computing mean length across $O(n^4)$ quartets in a scalable fashion. These quantities can be computed using a sophisticated dynamic programming algorithm, which borrows many ideas from weighted ASTRAL [296]. The running time of the algorithm is $O(n^2k)$ for n leaves and k genes. We present the full details for this algorithm in the next section.

5.3.1 Dynamic Programming Algorithm

In this section, we will provide an $O(n^2)$ large tree algorithm for computing branch lengths for all *internal* and *terminal* branches. For conciseness, we use A, B, C, D to denote sets of **taxa** and use a, b, c, d to denote individual taxa. To compute all branch lengths, it is sufficient to compute the following counters for a set of ordered leafset quadripartitions, in which each leafset quadripartition (A, B, C, D) – up to permutations – corresponds to an *internal* branch:

- $n(A, B; C, D)$: the number of quartet and gene tree combinations $(a, b, c, d, G) \in A \times B \times C \times D \times \mathcal{G}$ such that $G \restriction \{a, b, c, d\}$ has topology $ab|cd$.
- $x(A, B; C, D)$: the total internal branch lengths of quartet trees in the form of $G \restriction \{a, b, c, d\}$ with topology $ab|cd$, where $(a, b, c, d, G) \in A \times B \times C \times D \times \mathcal{G}$.
- $a(A; B; C, D)$: the total length of the terminal branches leading to A in quartet trees in the form of $G \restriction \{a, b, c, d\}$ with topology $ab|cd$, where $(a, b, c, d, G) \in A \times B \times C \times D \times \mathcal{G}$.

All three counters for each quadripartition (A, B, C, D) can be computed in a single post-order traversal of the gene tree nodes in $O(n)$ using Algorithm 5.2. Therefore, computing counters for all $O(n)$ quadripartitions has time complexity $O(n^2)$. Notice that Algorithm 5.2 assumes that all input gene trees are fully resolved. Otherwise, input gene trees should be arbitrarily resolved by adding ghost branches of zero lengths (not counted towards $n(A, B; C, D)$).

5.4 EXPERIMENTAL STUDY

Overview We performed a simulation study comparing CASTLES to four other methods by estimating branch lengths on the fixed *true* species tree topology. We report error measured as the absolute error averaged over all branches of each tree. Since absolute error hides the contribution of bias versus variance, we also report the mean error (without absolute), which is a valid measure of the bias of a method. Since the mean absolute error emphasizes long branches more than short branches, we also report two metrics that emphasize shorter branches successively more: root-mean square error (RMSE) and mean absolute log error. For all the methods, negative and zero branch lengths are replaced with 10^{-6} (the pseudo-count used by RAxML for identical sequences). Since negative lengths are not usable in downstream analyses, this step emulates practice.

We performed two experiments: The first, using a new simulated quartet dataset, is not meant to be realistic but to examine accuracy under idealized conditions and to show the impact of successively more challenging models of rate variation and the level of ILS. The second experiment uses two previously published simulated datasets with larger (30-taxon and 101-taxon) trees and more realistic settings, and examines the effect of gene tree estimation error (GTEE), the level of ILS, rate heterogeneity and deviation from the molecular clock, and the inclusion of an outgroup. Additional information about the simulation are provided in Section 5.8.

Algorithm 5.2 Large tree algorithm. The input is a set of gene trees \mathcal{G} and an ordered quadripartition of its leafset (A, B, C, D) , and the outputs are $n(A, B; C, D)$, $x(A, B; C, D)$, and $a(A; B; C, D)$. For each node u we keep a list of counters $C^{\cdot}(u)$ described in Table 5.4.

```

1: procedure UPDATELEAFCOUNTERS( $u, A, B, C, D$ )
2:   Set all counters  $C^{\cdot}(u)$  to 0
3:   if  $u$  corresponds to a taxon in  $A$  then
4:      $C_A^1(u) \leftarrow 1$ 
5:      $C_A^a(u) \leftarrow$  the parental branch length of  $u$ 
6:   else if  $u$  corresponds to a taxon in  $B$  then
7:      $C_B^1(u) \leftarrow 1$ 
8:      $C_B^b(u) \leftarrow$  the parental branch length of  $u$ 
9:   else if  $u$  corresponds to a taxon in  $C$  then
10:     $C_C^1(u) \leftarrow 1$ 
11:     $C_C^c(u) \leftarrow$  the parental branch length of  $u$ 
12:   else if  $u$  corresponds to a taxon in  $D$  then
13:      $C_D^1(u) \leftarrow 1$ 
14:      $C_D^d(u) \leftarrow$  the parental branch length of  $u$ 
15:   end if
16: end procedure
17: procedure LARGETREEALGORITHM( $\mathcal{G}, A, B, C, D$ )
18:   Set  $n(A, B; C, D)$ ,  $x(A, B; C, D)$ ,  $a(A; B; C, D)$  to 0
19:   for each gene  $G \in \mathcal{G}$  do
20:     for  $u \in$  post order traverse of internal nodes of  $G$  do
21:       if  $u$  is a leaf node then
22:         UPDATELEAFCOUNTERS( $u, A, B, C, D$ )
23:       else
24:         Update all counters  $C^{\cdot}(u)$  using the recursive formula in Table 5.4
25:          $n(A, B; C, D) \leftarrow n(A, B; C, D) + C_{AB|CD}^1(w)$ 
26:          $x(A, B; C, D) \leftarrow x(A, B; C, D) + C_{AB|CD}^x(w)$ 
27:          $a(A, B; C, D) \leftarrow a(A, B; C, D) + C_{AB|CD}^a(w)$ 
28:       end if
29:     end for
30:   end for
31: end procedure

```

Datasets To measure the accuracy, we need a species tree with **SU** branch lengths. The leading simulation method SimPhy [272] produces species trees in the unit of the number of generations (τ_i). However, SimPhy does select a global substitution rate ν and assigns a mutation rate multiplier (r_i) to each species tree branch (**-hs** option); setting $\nu_i = r_i \times \nu$ matches with the assumed model. Thus, the **SU** lengths on the species tree can be easily defined as $t_i = \tau_i \times \nu_i$. Unfortunately, SimPhy does not output the r_i rates; we modified its code to output these and the species tree with **SU** lengths. We used this modified version of SimPhy to regenerate species trees used in our datasets mentioned below and confirmed that the same trees are generated. This procedure gives us the ground truth **SU** lengths. We use SimPhy to evolve gene trees within each model species tree under the **MSC**, which allows us to explore the impact of **ILS** on branch length estimation. We quantify the level of **ILS** using the **average distance (AD)** between true species trees and true gene trees, in terms of the normalized **Robinson-Foulds (RF)** [83] distance, producing values that can range from 0% (no discordance) to 100% (no shared branches).

Quartet dataset. We generated a new quartet dataset using the modified version of SimPhy. We created six different model conditions by changing the level of **ILS** (by varying population size) and varying rate **heterogeneity** multipliers. Our model conditions start from a **strict molecular clock** with no rate variation (i.e., *Homogeneous*) and becomes successively more complex. Next, we add rate variations across species tree branches only (**-hs** option), creating a model (*Sp*) akin to MSC+Substitution mentioned earlier. We then create models that have rate variation only across genes but not species (*Loc* using **-hl**) and both across species and across genes (*Sp, Loc* using **-hs -hl**). Finally, we add rate variations specific to each branch of each **gene tree** (*Sp, Loc, Sp/Loc: -hs -hl -hg*), which creates **heterotachy**; this most complex model is how Simphy is usually used (e.g., in the next datasets) and goes beyond our theoretical model. The first five conditions have an **AD**=0.29, indicating a moderate level of **ILS**. The final condition increases the **ILS** level to 0.51 **AD**. Each model condition has 200 replicates, each with 10,000 true gene trees. We intentionally used a large number of true gene trees to verify our formulas and compare methods in an ideal situation. Further details and parameters are provided in Tables 5.6 and 5.7.

S100 dataset. We used a 101-taxon simulated dataset from [248] (100 ingroup and one outgroup), that had model conditions characterized by different levels of gene tree estimation error, ranging from 0 (for true gene trees) to 0.55, measured in terms of the **RF** distance between true and estimated gene trees. The **ILS** level changes dramatically across replicates (average: 0.46 **AD**). The estimated gene trees were created using FastTree2 [144]. These datasets had 50 replicates, each with 1000 gene trees.

MVRoot dataset. We used a 30-taxon dataset from [241] that had model conditions that varied in terms of deviation from the molecular clock and inclusion of an outgroup. Deviation from the clock was specified with the parameter α of the gamma distribution, choosing 0.15 (*High* variation), 1.5 (*Med*), or 5 (*Low*). This dataset had 100 replicates with 500 gene trees (estimated using FastTree2) in each replicate. The replicates were highly heterogeneous in terms of ILS and GTEE level (average 0.46 AD and 0.38 GTEE across all model conditions).

Methods compared We compare CASTLES to four other methods: concatenation using maximum likelihood, FastME [297] on two different distance matrices, and ERaBLE [292]:

- Concatenation with maximum likelihood using RAxML [298] is perhaps the dominant method used in the literature, and estimates branch lengths on the given species trees assuming all the sites in the concatenated alignment evolve down a single model tree.
- FastME [297] can estimate branch lengths using the balanced minimum evolution criterion given a distance matrix. We use it with two distance matrices. First, we compute the patristic (path-length) distance between pairs of taxa for each gene tree using Dendropy [299]. Genes with no signal (all branch lengths zero) are excluded. We then take either the average or the minimum for each pair across genes. In the absence of rate heterogeneity, the minimum is appropriate and has been used in GLASS and its variants [167].
- ERaBLE [292] is specifically designed for branch-length estimation from a set of gene trees and is similar to FastME but uses weighted means.

5.5 RESULTS

5.5.1 Quartet simulations

When considering all conditions, CASTLES has the best accuracy overall (Figure 5.7a). Patristic(MIN)+FastME has the lowest error in conditions with no rate heterogeneity across loci. As soon as rate heterogeneity across loci is added (i.e., Loc), it goes from being the best method to being the worst. As expected, the error for all methods tends to increase as the models become more challenging (i.e., more rate variation or higher ILS). In the penultimate condition with default ILS and all sources of rate variation, CASTLES has substantially lower error than alternatives. When ILS is increased, we observe a huge increase in error for ERaBLE and Patristic(AVG)+FastME, but not for Patristic(MIN)+FastME. Since the

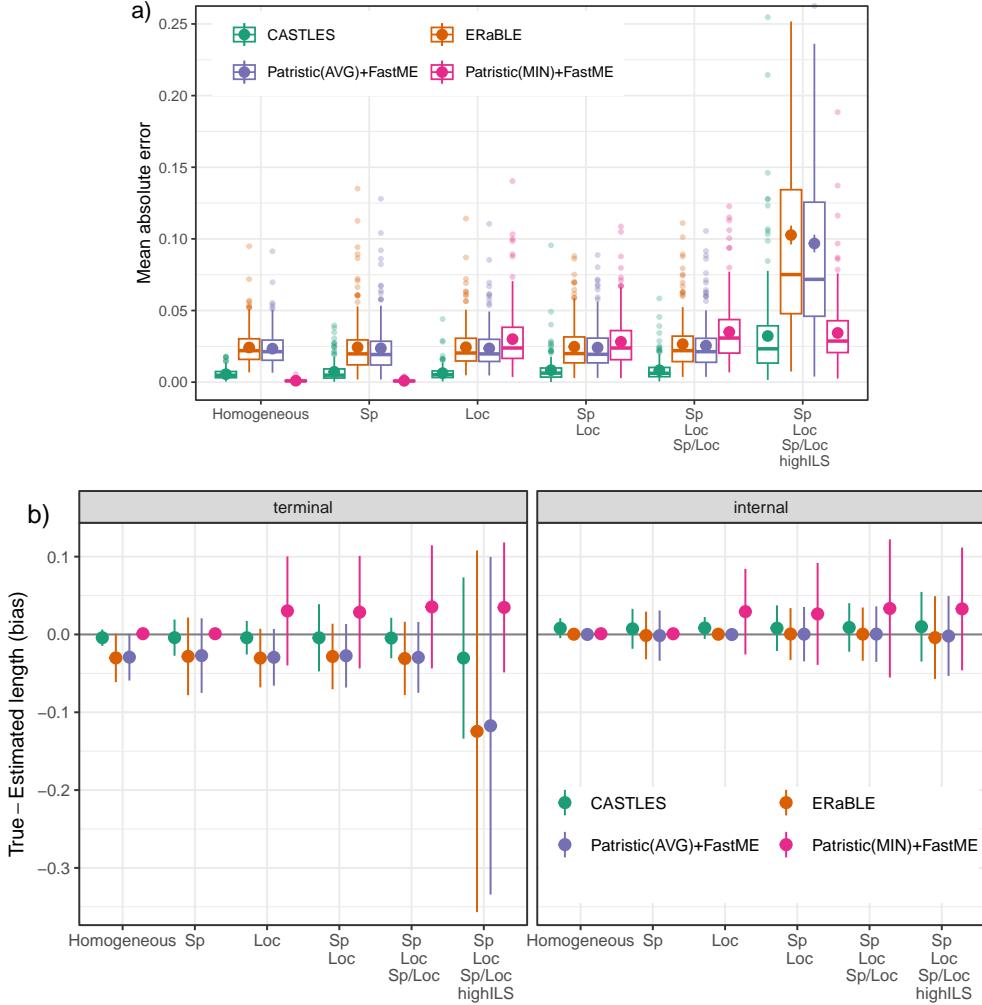


Figure 5.7: Quartet datasets: Mean absolute error (a) and bias (b) of branch lengths estimated using different methods. From left to right, the conditions include more rate variation or higher **ILS**, creating more challenges for branch length estimation. a) Mean and standard error across replicates in addition to boxplots. The y-axis is cut at 0.25, eliminating 16 outlier cases with unusually high errors (none from CASTLES). b) Mean and standard deviation.

mean absolute error emphasizes long branches more than short branches, we also examine **RMSE** and log error (Figure 5.17). The trends with these metrics are very similar to mean absolute error, except that with the log error (emphasizing short branches and long branches alike), Patristic(MIN)+FastME is far worse than the other methods in conditions with rate variation across loci.

Switching from accuracy to bias, we observe little or no bias for CASTLES for **terminal** branches in all conditions except at the highest **ILS** level (Figure 5.7b and 5.18). In contrast, ERaBLE and Patristic(AVG)+FastME, have a clear over-estimation bias for **terminal** branches, and Patristic(MIN)+FastME has a clear underestimation bias (for all branches),

except in the absence of rate variation across genes (signified by *Loc*). Terminal branches seem particularly biased in the condition with the highest rate variation and the highest level of **ILS**. In this condition, while CASTLES does seem to have some bias for **terminal** branches, it is far less biased than alternative methods. Comparing the last two conditions, we observe that higher **ILS** has a larger impact on bias than rate variation. In contrast to **terminal** branches, for **internal** branches, ERaBLE and Patristic(AVG)+FastME also have low bias (slightly lower than CASTLES).

5.5.2 101-taxon **ILS** simulations

On this dataset, CASTLES has the best accuracy across all model conditions, followed by Patristic(AVG)+FastME and ERaBLE, which are very similar to each other (Figure 5.8b). Concat+RAxML has substantially higher errors than these three methods that use gene trees as input. However, Patristic(MIN)+FastME has the highest error in all conditions. These patterns remain largely similar, according to the **RMSE** and log error (Figure 5.20).

CASTLES shows no substantial bias for **terminal** branches regardless of the level of **gene tree** error and a small bias for **internal** branches (Figures 5.19,5.8). This bias is towards under-estimation for true gene trees and gradually moves towards over-estimation as **gene tree** error increases. In contrast to CASTLES, ERaBLE, Patristic(AVG)+FastME, and Concat+RAxML have a large over-estimation bias for **terminal** branches. ERaBLE and Patristic(AVG)+FastME have a negligible bias for **internal** branches. Concat+RAxML has the highest over-estimation bias and is the only method with a substantial over-estimation bias for **internal** branches. Patristic(MIN)+FastME has a large under-estimation bias. Similar to quartet simulations, all methods are less biased for **internal** branches than **terminal** ones. Comparing conditions, we observe that the level of **gene tree** error has a relatively small impact on under/overestimation for all the methods tested.

On this relatively large dataset, we also examine running times and observe that CASTLES is substantially faster than alternatives (Figure 5.8c). Note that **gene tree** estimation running time is not included for methods based on gene trees because those are often inferred *regardless* of branch length estimation. Concat+RaxML becomes successively slower and uses more memory (Figure 5.21) as the genes become longer. In the most extreme case, CASTLES can be more than an order of magnitude faster than Concat+RAxML.

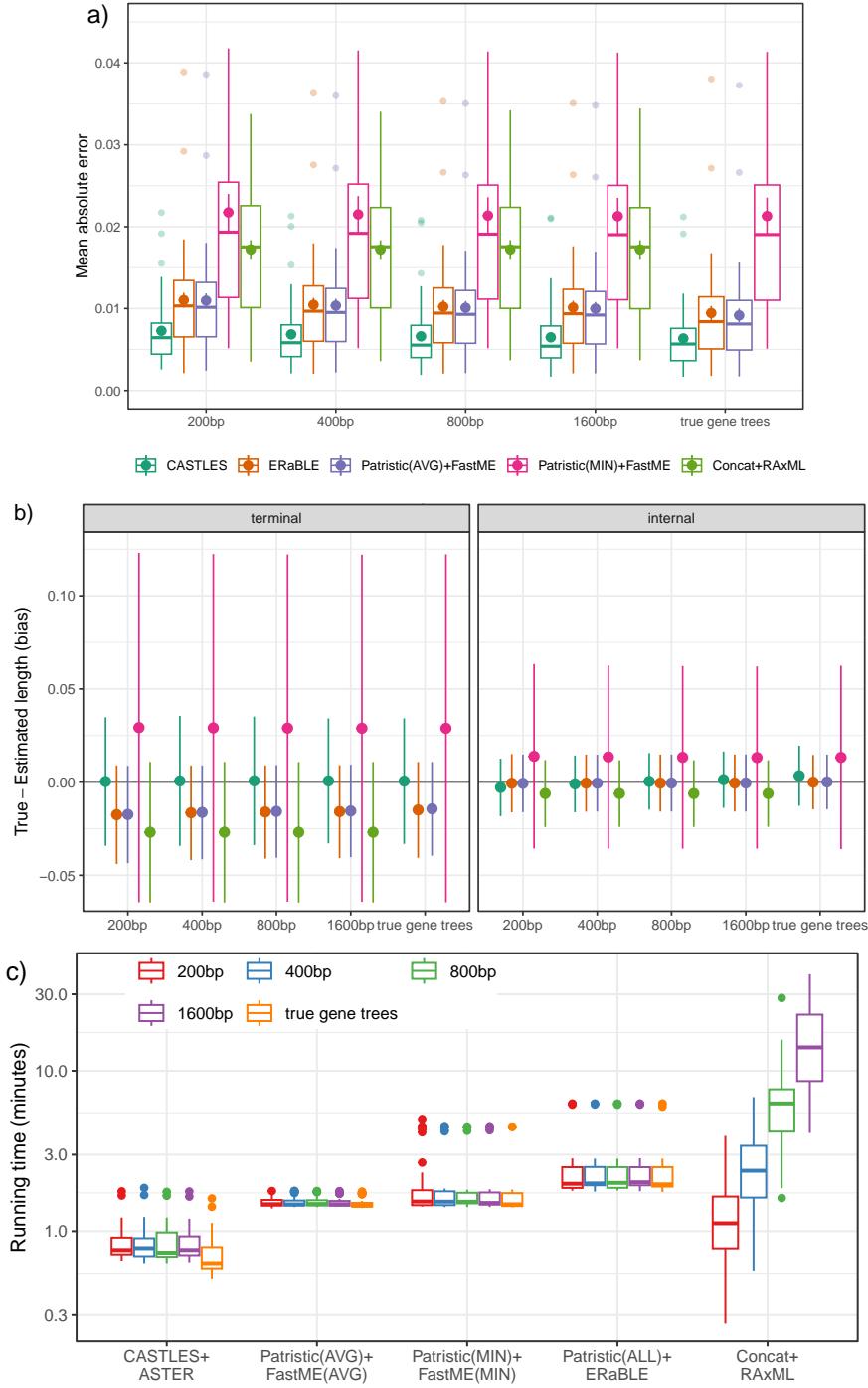


Figure 5.8: 101-taxon datasets: Mean and standard error of mean absolute error (a) and mean and standard deviation of bias (b) of branch lengths estimated using different methods. The average GTEE level varies between 0% (for true gene trees) to 23% (for 1600bp) and then to 55% (for the 200bp sequences). The number of genes is 1000 and the results are shown across 50 replicates. The y-axis is cut at 0.045, eliminating ten outlier cases (none from CASTLES). (c) Running time (log scale), including distance matrix calculation and species tree annotation (by mean branch lengths) but not gene tree estimation; concatenation includes branch length estimation for fixed topology.

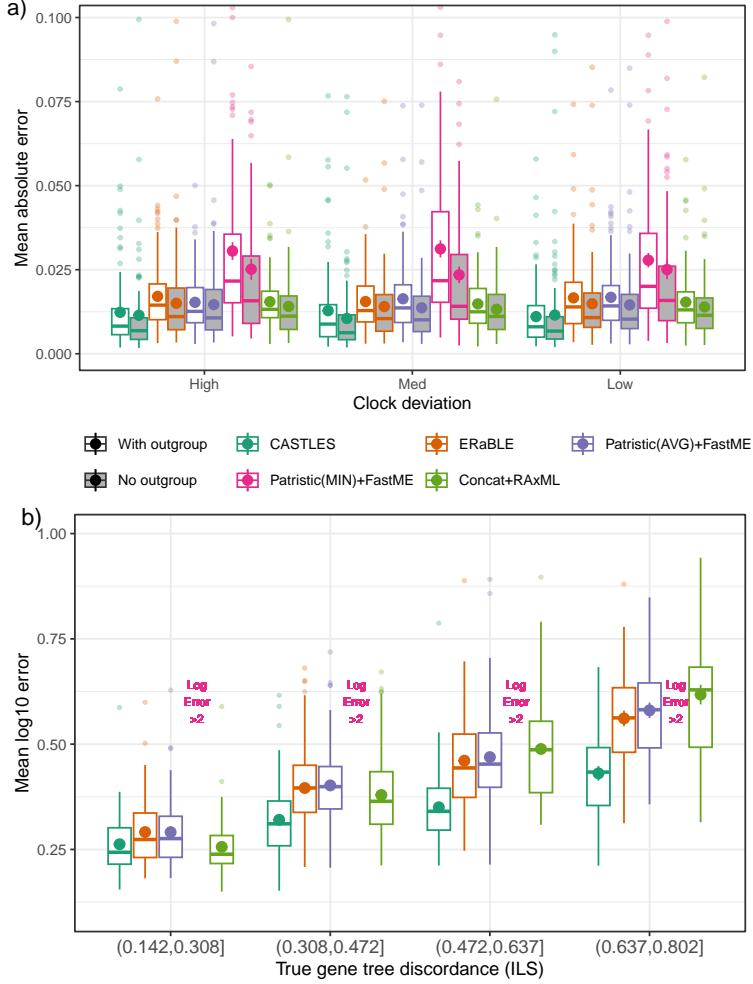


Figure 5.9: 30-taxon MVRoot dataset. a) Mean absolute error of estimated branch lengths on the 30-taxon MVRoot dataset, with or without an [outgroup](#) and with different levels of deviation from a strict clock. The number of genes is 500 and the results are shown across 100 replicates; the y-axis is cut at 0.11, leaving 16 outliers out of the graph (one from CASTLES). b) Focusing on cases without outgroups, we divide replicates based on their level of true [gene tree](#) discordance due to [ILS](#) into four groups. We show mean log error to control for the correlation between [ILS](#) and branch length. Patristic(MIN)+FastME has mean log error above 2 (see Figure 5.24) and is excluded.

5.5.3 30-taxon MVRoot simulations

On the 30-taxon MVRoot datasets, we further evaluate the impact of outgroups, deviation from the clock, and [ILS](#) level (Figure 5.9). Whether an [outgroup](#) is included and independent of deviation from the clock, CASTLES has the lowest error (Figure 5.9a). On these datasets, CASTLES has no discernible bias, ERaBLE, Patristic(AVG)+FastME and Concat+RAxML have a bias towards over-estimation (see an example replicate in Figure 5.11a),

and Patristic(MIN)+FastME has a more severe bias towards under-estimation; `outgroup` inclusion and deviation from the clock impact the bias of methods only marginally (Figure 5.22). For all methods, including an `outgroup` leads to an increase in the mean absolute error (Figure 5.9a). Increasing deviation from the clock does not substantially impact the accuracy of CASTLES or other methods (Figures 5.9, 5.22).

To compare across levels of `ILS`, we resort to the logarithmic error because true branch lengths correlate with `ILS` (i.e., are shorter for higher `ILS`), and hence, the absolute error confuses the interpretation of the impact of `ILS`. Across all `ILS` levels (Figure 5.9b), Patristic(MIN)+FastME has very poor performance in terms of log error and is not further discussed below. With the lowest `ILS`, CASTLES and Concat+RAxML have very similar performance. As `ILS` increases, all methods become less accurate, but CASTLES degrades in accuracy *slower* than the rest of the methods and hence dominates the other methods for accuracy, with ERaBLE in second place. Concat+RAxML matches CASTLES for the lowest `ILS` but gradually moves to be the second least accurate of all methods (only better than Patristic(MIN)+FastME) at the highest `ILS` level. In fact, Concat+RAxML is substantially more sensitive to `ILS` ($R^2 = 0.57$ Pearson correlation with AD; Figure 5.23) than CASTLES ($R^2 = 0.23$). Comparing the relative accuracy of methods as `ILS` changes using the mean absolute error shows similar trends (Figure 5.24) with one notable difference: Concat+RAxML is *better* than CASTLES at the lowest `ILS` level but is worse in other conditions (just as with the log error).

5.5.4 Mammalian biological dataset

We apply CASTLES and Concat+RAxML on the 37-taxon mammalian dataset of [289] (perhaps the first paper that used concatenation to estimate branch lengths on a species tree estimated using a summary method) after removing 23 mislabelled gene trees, retaining 424 genes (Figure 5.11b). We observe patterns similar to the simulated dataset. Branch lengths tend to be longer using Concat+RAxML than using CASTLES (Figure 5.10). For example, primates are roughly twice as distant from the root of placental mammals in the Concat+RAxML tree as they are in the CASTLES tree.

While the two trees are similar in their longest branches and have similar diameters (i.e., from rat to platypus), many of the other `internal` branches are substantially shorter in CASTLES. While the truth is not known on real data, we note that a similar pattern is observed in simulations, and in simulations, Concat+RAxML is biased towards over-estimation; in contrast, CASTLES is far less biased (e.g., Figure 5.11a).

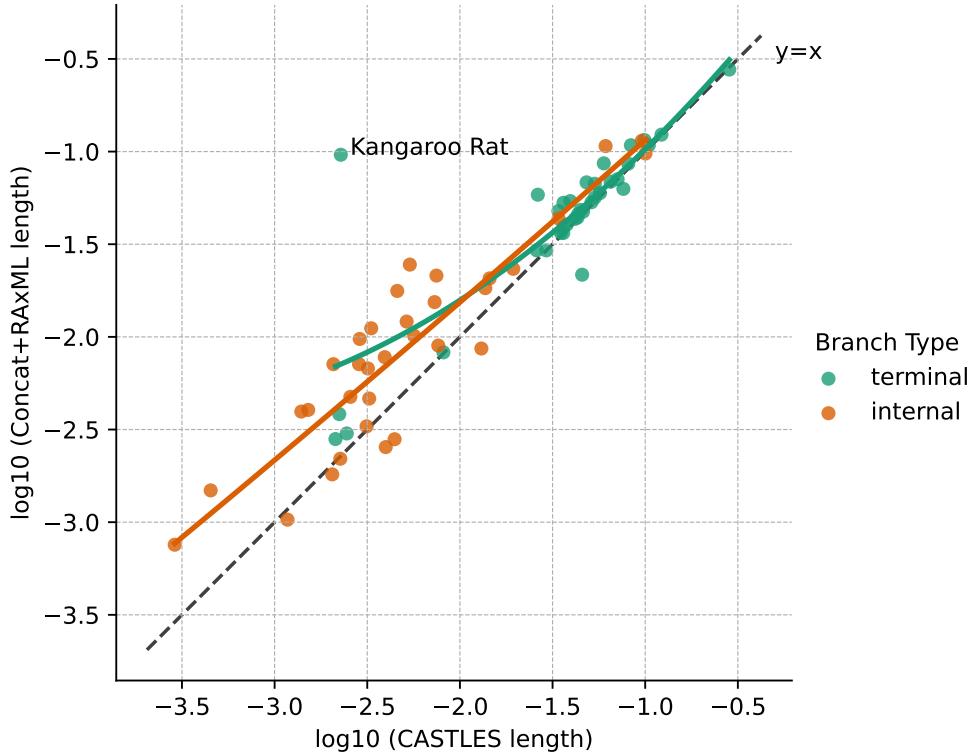


Figure 5.10: Correlations between branch lengths produced by CASTLES and Concat+RAxML on the 37-taxon mammalian dataset [289]. The branch lengths are drawn on a species tree topology constructed by ASTRAL. The [outgroup](#) (Chicken) is removed before drawing branch lengths on the tree. The number of genes used in the analysis is 424. Each point corresponds to a single branch in the species tree. The colored lines show a fitted degree-two polynomial.

5.6 DISCUSSION

Although CASTLES was almost universally more accurate than the competing methods, comparisons across the experiments revealed some interesting trends. Concatenation performed well on low [ILS](#) cases and much worse with high [ILS](#), as expected. Increasing [ILS](#) did increase error for all methods but also note that more [ILS](#) in our simulation is often (though not always) accompanied by shorter branch lengths, which are in general harder to estimate (Figures 5.19, 5.18). However, the fact that concatenation degraded in accuracy faster than other methods as [ILS](#) increased confirmed that it is less able to deal with [gene tree](#) discordance. Thus, the current standard method (concatenation) does suffer from a predictable shortcoming. CASTLES is meant to address that shortcoming.

We observed that estimating [terminal](#) branches was harder than [internal](#) branches across all datasets for all methods other than CASTLES. As we expected, methods that ignore

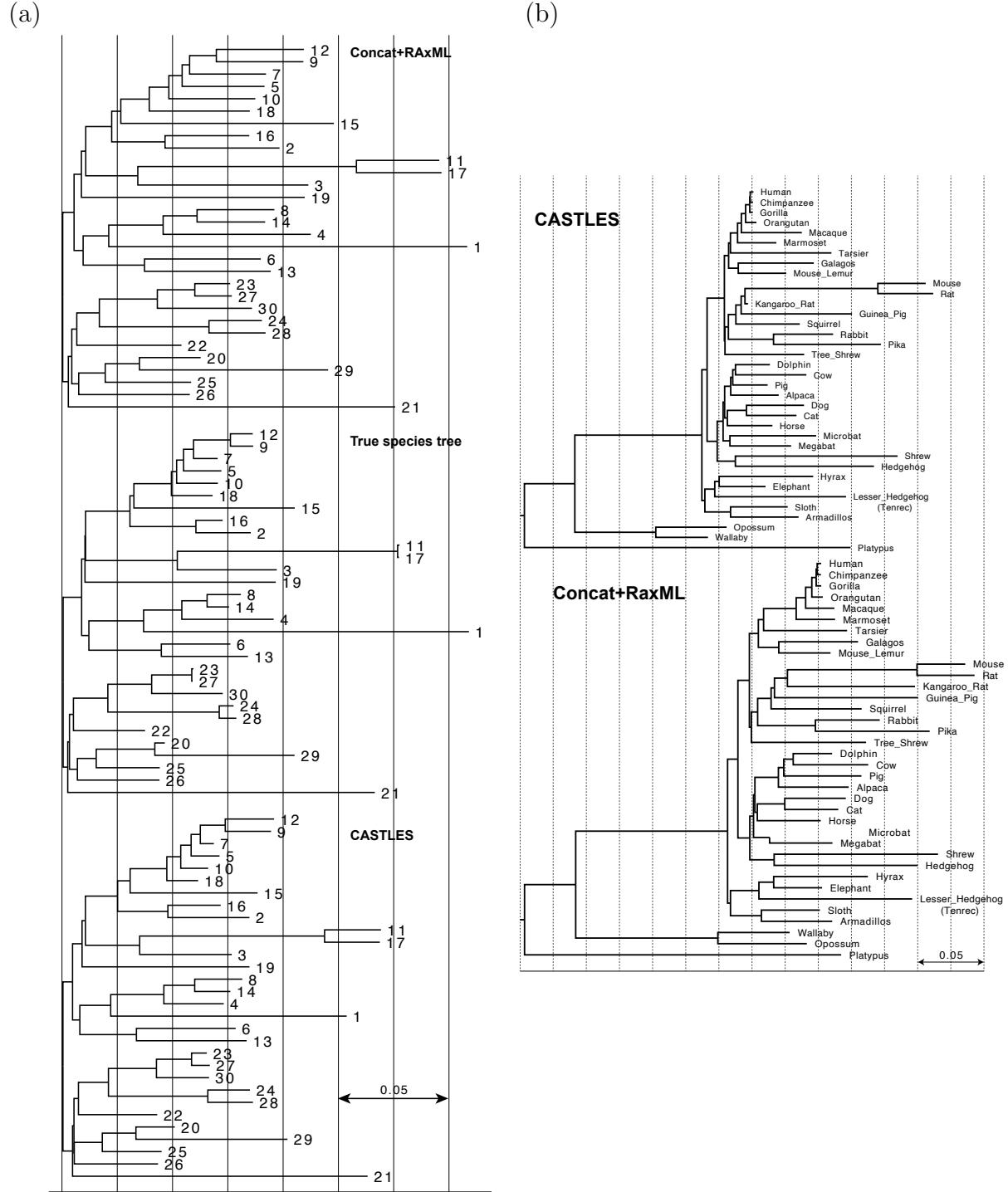


Figure 5.11: (a) Visualization of the true species tree (middle), and trees with branch lengths produced by CASTLES (bottom) and Concat+RAxML (top) for replicate 1 of MVRoot dataset in the default model condition (no outgroup, medium deviation from the strict clock). The ILS level and GTEE level for this replicate are 0.53 and 0.34 respectively. (b) Same methods applied on the real mammalian dataset [289]. Here, the outgroup (Chicken) is removed *before* drawing branch lengths on the tree. The trees are visualized using FigTree [300].

coalescent (e.g., concatenation and FastME based on *average patristic distance*) had a consistent overestimation bias. What was surprising was that this overestimation showed its effects more on **terminal** branches than **internal** branches. The reasons for this clear trend are not clear to us.

Another consistent pattern was that including an **outgroup** reduced accuracy for all methods and especially for CASTLES. Outgroups are often connected via long branches and have been found problematic for phylogenetic inference and downstream analyses [301]. Our results suggest that they can also confound **SU** branch length estimation. While not surprising, this pattern suggests that unless the **outgroup** is needed for a downstream analysis, the outgroups should be removed after rooting the species tree and before estimating branch length.

We surprisingly saw little impact caused by **gene tree** estimation error and deviation from a clock. The robustness to deviation from the strict clock can perhaps be explained by the fact that none of the methods used here other than Patristic(MIN)+FastME assume a clock. Note that in CASTLES, even with our simplifying assumptions, each branch is at the end assigned a different mutation rate (the calculation of which assumes surrounding branches have the same rate). The lack of sensitivity to per-gene signal (controlled here by sequence length) is more surprising, especially for the **coalescent**-based CASTLES. One possibility is that while short sequences can affect the estimated **gene tree** topologies, they have a more subdued effect on distances within gene trees [302]; thus, even given short sequences, estimated branch lengths (which change in a continuous space) are broadly consistent with true values, especially when averaged over genes. In contrast, **CU** branch lengths are sensitive to **gene tree** error, but these are not used in CASTLES.

Our theory did not explicitly discuss rate **heterogeneity** across genes. However, across-gene rate changes do not impact our calculations under reasonable models of rate variation. Assume that each **gene tree** is scaled up or down by a constant factor drawn i.i.d. from some rate multiplier distribution with expected value one and independently from the **MSC** process. Under any such model, all the derived expected values remain intact and hence the method remains valid. It is easy to see the same is not true for GLASS-like distances that involve taking the minimum across genes: they only work if all the genes have equal rates. When rates of evolution of genes are allowed to vary, as the number of genes goes to infinity, all estimated branch lengths go to zero; a pattern that would imply under-estimation bias, as we observed in our data. More broadly, if rate changes are not i.i.d and correlate with other factors such as missing data, accounting for them becomes far more difficult.

Finally, we note that the estimation of **terminal** branches in CASTLES is sensitive to rooting of the species tree; hence, care must be applied in rooting the species tree before

running CASTLES. When an [outgroup](#) is not available, the species tree root is [identifiable](#) under the [MSC](#) and can be inferred using QR-STAR [?] in a [statistically consistent](#) manner. Alternative methods such as tripVote [303] or methods that assume a [strict molecular clock](#) (e.g., midpoint rooting) are also available, though they do not enjoy the same theoretical guarantees.

5.7 CONCLUSION

We proposed CASTLES, a method for estimating branch lengths of a given species tree using [gene tree](#) branch lengths. CASTLES uses derivations made under the multi-species coalescent (MSC) model to design a set of [coalescent](#)-based equations that correct for the fact that under the [MSC](#), gene trees can be substantially longer than the species tree. Our study provided evidence that CASTLES produces highly accurate branch lengths in substitution units (SUs), improving on prior methods under a wide range of model conditions.

There are several directions for future work. For example, the derivation of CASTLES assumed that the input gene trees differed from the species tree due to [ILS](#) alone. This work could be extended to the case where genes evolve within the species tree due to [Gene duplication and loss \(GDL\)](#) as well as [ILS](#). Developing methods for branch length estimation in that context could be potentially enabled through the DISCO [90] technique, which replaces every gene family tree by a set of single-copy gene trees, which could then be passed to CASTLES for branch length estimation on the given species tree. A related question left for future work is whether CASTLES is robust to the presence of horizontal transfer or gene flow. Finally, the behavior of the method should be tested when inputs have low taxon occupancy across genes.

Another question of interest is whether CASTLES is a [statistically consistent](#) estimator of [SU](#) branch lengths. Given that CASTLES is [coalescent](#)-based and that we use expected values under the model, we consider this likely, but two technical challenges need to be addressed. First, for the method to be consistent, we need the model to be [identifiable](#), and we did not establish identifiability in this chapter. Thus, we ask: Is it possible to design different sets of mutation rates and [CU](#) lengths that lead to the same patterns of [gene tree](#) distribution? If not, are the expected values enough to uniquely identify branch lengths or are higher moments necessary? Moreover, while we had a system of equations that could be optimized directly, we opted for a more stable approach that had several simplifications. It is possible (and perhaps likely) that those simplifications could result in inconsistent branch length estimation. These questions will need to be addressed in future work, and may require that we explore a more complex estimation scheme that does not rely on simplifications.

5.8 METHODS AND SOFTWARE COMMANDS

5.8.1 Quartet Simulations

We used a modified version of SimPhy [272], available at https://github.com/ytabatabae/CASTLES/simulation_files, that outputs species trees with substitution units branch lengths. We simulated quartet species trees and true gene trees under six different model conditions using the following example command, with parameters given in Tables 5.6 and 5.7.

```
q=no_variation # Homogeneous
./simphy -rs 200 -rl f:10000 -rg 1 -sb f:0.0001 -sd f:0 -st f:400
-sl f:4 -so f:0 -si f:1 -sp f:200 -su ln:-9,0.5 -hs f:10000 -hl
f:10000 -hh f:10000 -hg f:10000 -cs 12964 -v 4 -o $q -ot 0 -op 1
-od 1 > log-$q.txt
```

In this section, we bring the details of the experimental pipeline and software commands. The code for CASTLES as well as the scripts for error and distance matrix calculation use functions from DendroPy [252]. All experiments were run on the University of Illinois campus cluster.

Branch Length Estimation We used the following commands to estimate branch lengths in substitution units on a given species tree topology:

- **CASTLES:** Running CASTLES is a two-step approach: 1) Annotate branches of the species tree with mean quartet branch lengths around it using the tool ASTER, and 2) assign final branch lengths to each branch using the `castles.py` (v1.0.0) code available at <https://github.com/ytabatabae/CASTLES/castles.py>.

Step 1) The algorithm to annotate branches with mean lengths of quartets around it is implemented in ASTER (v1.13.2.4) available at <https://github.com/chaoszhang/ASTER>. To annotate a fixed species tree `topology` with quartet statistics for each branch, we ran it with the option `-C` to score a fixed species tree specified after `-c`. Note that ASTER needs to be complied with a particular option for this annotation to work:

```
g++ -std=gnu++11 -D"ASTRALIV" -march=native -Ofast -pthread
src/astral.cpp -o bin/astral
```

Assuming this version is used, we used the following command, where the annotated tree is printed to the log file:

```
astral -C -i <gene_tree> -c <species_tree> -o <output_path>
> annotated.tre
```

Particularly, when we have multiple individuals per species and the individual names do not match the species names, we run the following command:

```
astral -C -i <gene_tree> -m <name_map>
       -c <species_tree> -o <output_path> > annotated.tre
```

where the “name map” file contains maps from individual names to species names in the following format:

```
individual_name1      species_name1
individual_name2      species_name2
individual_name3      species_name3
...
...
```

Step 2) The annotated tree is given to CASTLES to compute branch lengths in substitution units from these quartet statistics. To run CASTLES, we used the following command (note that the log file from ASTER is directly passed to CASTLES as input):

```
python3 castles.py -t annotated.tre -g <gene_tree_path>
                   -o <output_path>
```

- **FastME+Mean or FastME+Min:** Running FastME is also a two-step approach: 1) estimating the distance matrix, 2) inferring branch lengths.

Step 1) We used our custom script to compute patristic (path-length) distance matrices from gene trees in Phylip format. The core of this script is the `PhylogeneticDistanceMatrix` class from Dendropy. The script is available at https://github.com/ytabata/baee/CASTLES/scripts/patristic_dist_matrix.py. The option `-m` specifies the type of distance matrix: `'avg'` and `'min'` compute a single squared matrix corresponding to the average and minimum path-length distances between pairs of nodes in a set of gene trees, respectively.

Step 2) We used FastME (v2.1.6.2) [304] available at <http://www.atgc-montpellier.fr/fastme/> to assign branch lengths with balanced minimum evolution (BME) criteria (specified with `-w` `BaLLS`) to a given species tree topology, specified with `-u`, given a single distance matrix corresponding to average or minimum `patristic distances` computed across a set of gene trees, using the following command:

```
fastme-2.1.6.2-linux64 -i <dist_mat.phylip> -w BaLLS
                       -u <species_tree_path> -o <output_path>
```

- **ERaBLE**: Similar to FastME, ERaBLE requires pre-computation of distances. We used the same script as FastME but with option ‘all’ to compute one **patristic distance** matrix *per* gene, for a set of gene trees. We used the following command:

```
python3 patristic_dist_matrix.py -t <species_tree_path>
-g <gene_tree_path> -o <output_path> -m all
```

We then used ERaBLE (v1.0) [292] available at <http://www.atgc-montpellier.fr/erable/>. The input to ERaBLE is an **unrooted tree topology** in newick format, specified with the option **-t**, and a set of k distance matrices in Phylip format, each corresponding to a single gene tree, generated above. We used the following command:

```
erable -i <dist_mat.phylip> -t <species_tree_path>
-o <output_path>
```

- **RAxML**: We used RAxML (v8.2.12) [25] to estimate branch lengths on a given species tree topology, using a concatenated sequence alignment, with the option **-f e**. RAxML is available at <https://github.com/stamatak/standard-RAxML>. We used the following command:

```
raxmlHPC-PTHREADS -f e -t <species_tree_path> -m GTRGAMMA
-s <alignment_path> -n RES -p 4321 -T 16
```

Error Calculation We used the script available at https://github.com/ytabatabae/CASTLES/scripts/compare_trees.bl.py to compare branch lengths on two trees, and compute the **root-mean square error (RMSE)** and the average logarithmic error between the trees for all branches. We used the following command:

```
python3 compare_trees.bl.py -t1 <true_species_tree_path>
-t2 <est_species_tree_path>
```

which tabulates the set of true and estimated branch lengths. These data are then analyzed using an R script available at <https://github.com/ytabatabae/CASTLES/results/draw.R>. Note that for all methods, before computing the error, negative and zero branch lengths are replaced with 1e-6.

The formulas for the error metrics used in this study for a species tree \mathcal{T} with b branches are as follows, where t_i and \hat{t}_i are the true and estimated lengths of branch i in **SU** respectively:

- Bias: $\frac{1}{b} \sum_{i=1}^b (t_i - \hat{t}_i)$
- Mean absolute error: $\frac{1}{b} \sum_{i=1}^b |t_i - \hat{t}_i|$

- Logarithmic error: $\frac{1}{b} \sum_{i=1}^b |log_{10}(t_i) - log_{10}(\hat{t}_i)|$
- Root mean square error (RMSE): $\sqrt{\frac{1}{b} \sum_{i=1}^b (t_i - \hat{t}_i)^2}$

Runtime and Memory We measure runtime as the total running time of all steps of each method, assuming that the estimated gene trees, alignments, and species tree [topology](#) are already available. This includes the time needed to calculate the distance matrices for ERaBLE and FastME, as well as the time needed for annotating trees with ASTER that is given as input to CASTLES. To measure the runtime and peak memory usage of a command `<cmd>`, we use the following command:

```
/usr/bin/time -v -o out.stat -f "\t%e\t%M" <cmd>
```

We record runtime from the elapsed wall clock time, and peak memory usage from maximum resident set size from the generated `out.stat` file.

5.8.2 Biological Data Analysis

We analyzed the mammalian biological dataset of [289], which had 37 species (36 ingroups and one outgroup). The original dataset had 447 genes, but we used a processed version of the dataset with 424 that had 23 mislabeled or outlier genes removed. We estimated an [unrooted](#) species tree using ASTRAL (v5.7.8) [248] available at <https://github.com/smirarab/ASTRAL> using the following command.

```
java -jar astral.5.7.8.jar -i <gene-trees.tre> -o <species-tree.tre>
```

We then removed the [outgroup](#) (Chicken, specified with the name “GAL” in the dataset) from the ASTRAL tree, as our simulations show that branch length estimation methods benefit from the removal of outgroup, and then estimated branch lengths on the tree. All files associated with this analysis are available at <https://github.com/ytabatabae/Castles/results/mammalian-biological-analysis>.

5.9 ADDITIONAL FIGURES AND TABLES

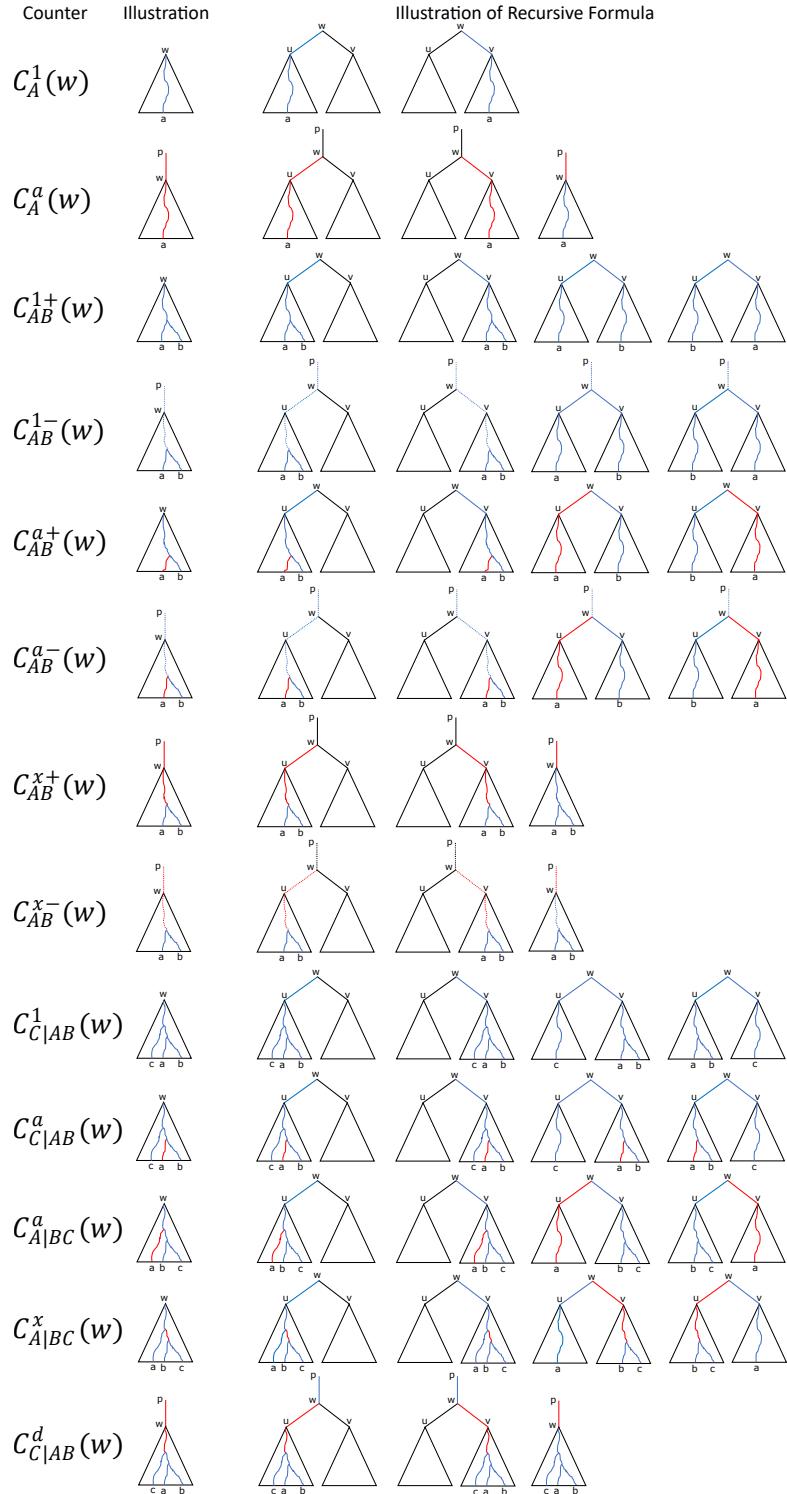


Figure 5.12: Illustration of counters and their recursive formulas. Branches colored red are counted by lengths, and dotted branches must be ghost branches.

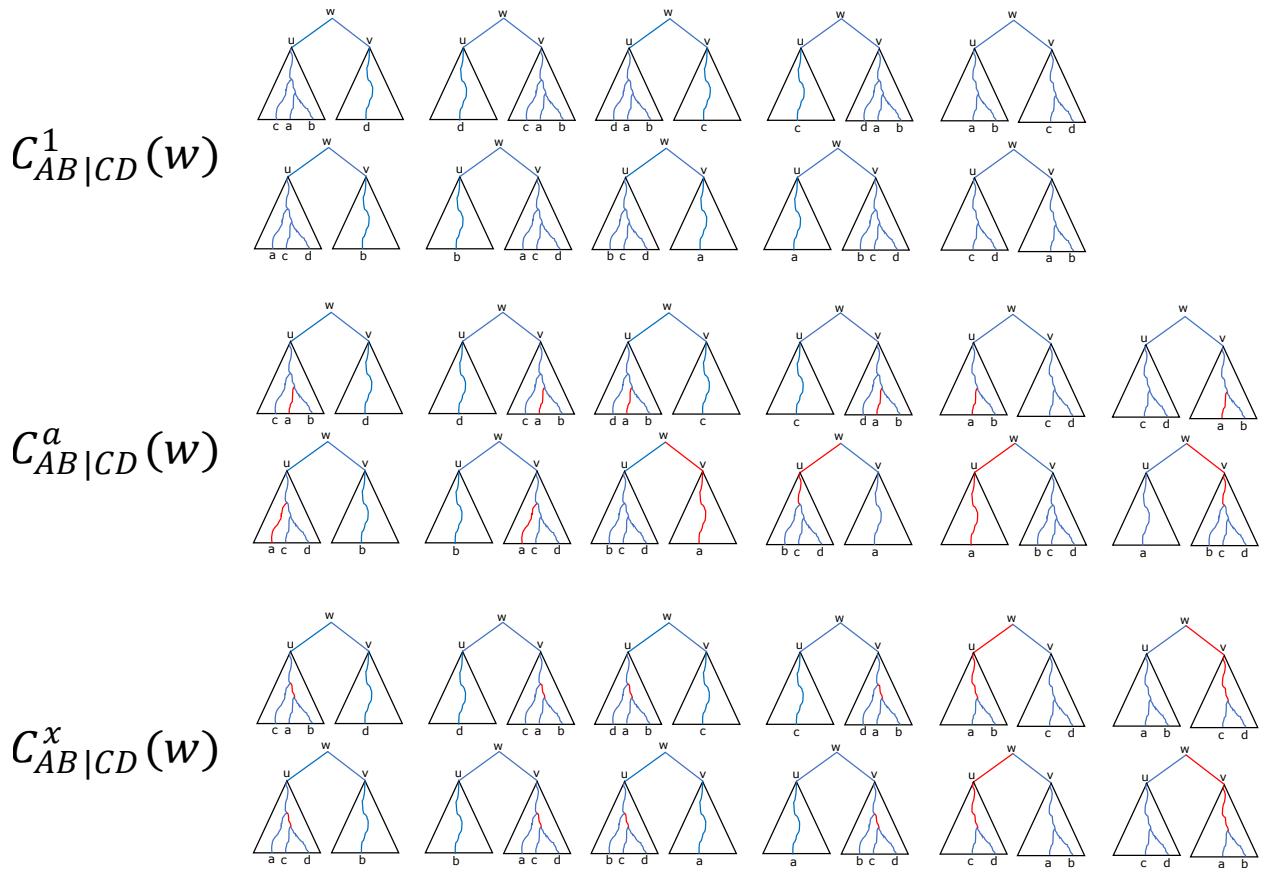


Figure 5.13: Illustration of recursive formulas (continued).

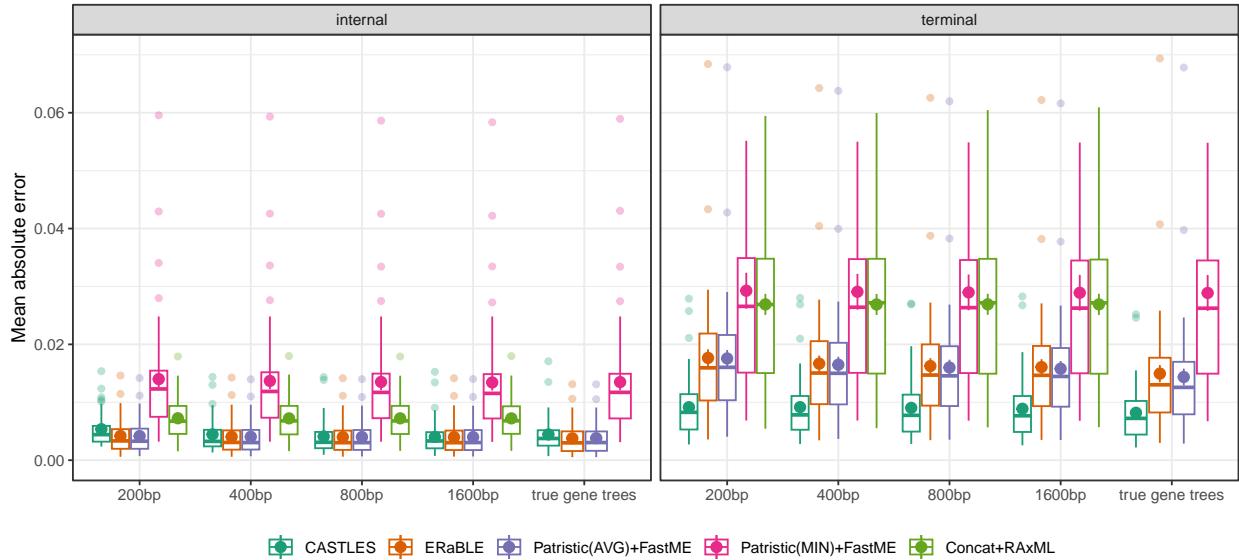


Figure 5.14: Mean absolute error on [terminal](#) and [internal](#) branches for simulated 101-taxon datasets. The plot shows mean and standard error across 50 replicates, in addition to boxplots. The y-axes is cut at 0.07, eliminating a few outlier cases with unusually high errors (none from CASTLES). Figure 5.8.a in the main text shows the same error aggregated for [terminal](#) and [internal](#) branches.

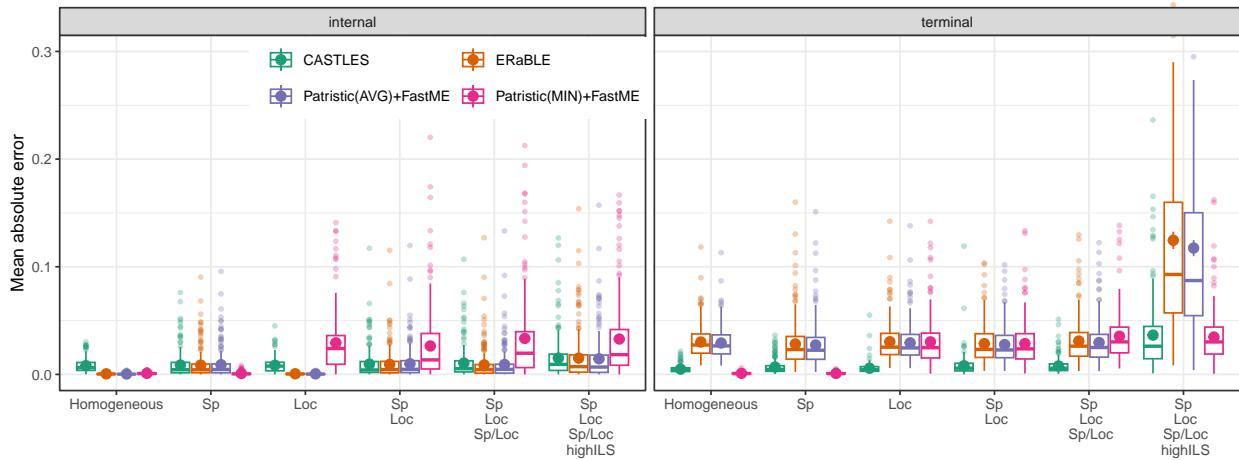


Figure 5.15: Mean absolute error on [terminal](#) and [internal](#) branches for simulated quartet datasets. The plot shows mean and standard error across 200 replicates, in addition to boxplots. The y-axes is cut at 0.3, eliminating a couple of outlier cases with unusually high errors (none from CASTLES). Figure 5.7.a in the main text shows the same error aggregated for [terminal](#) and [internal](#) branches.

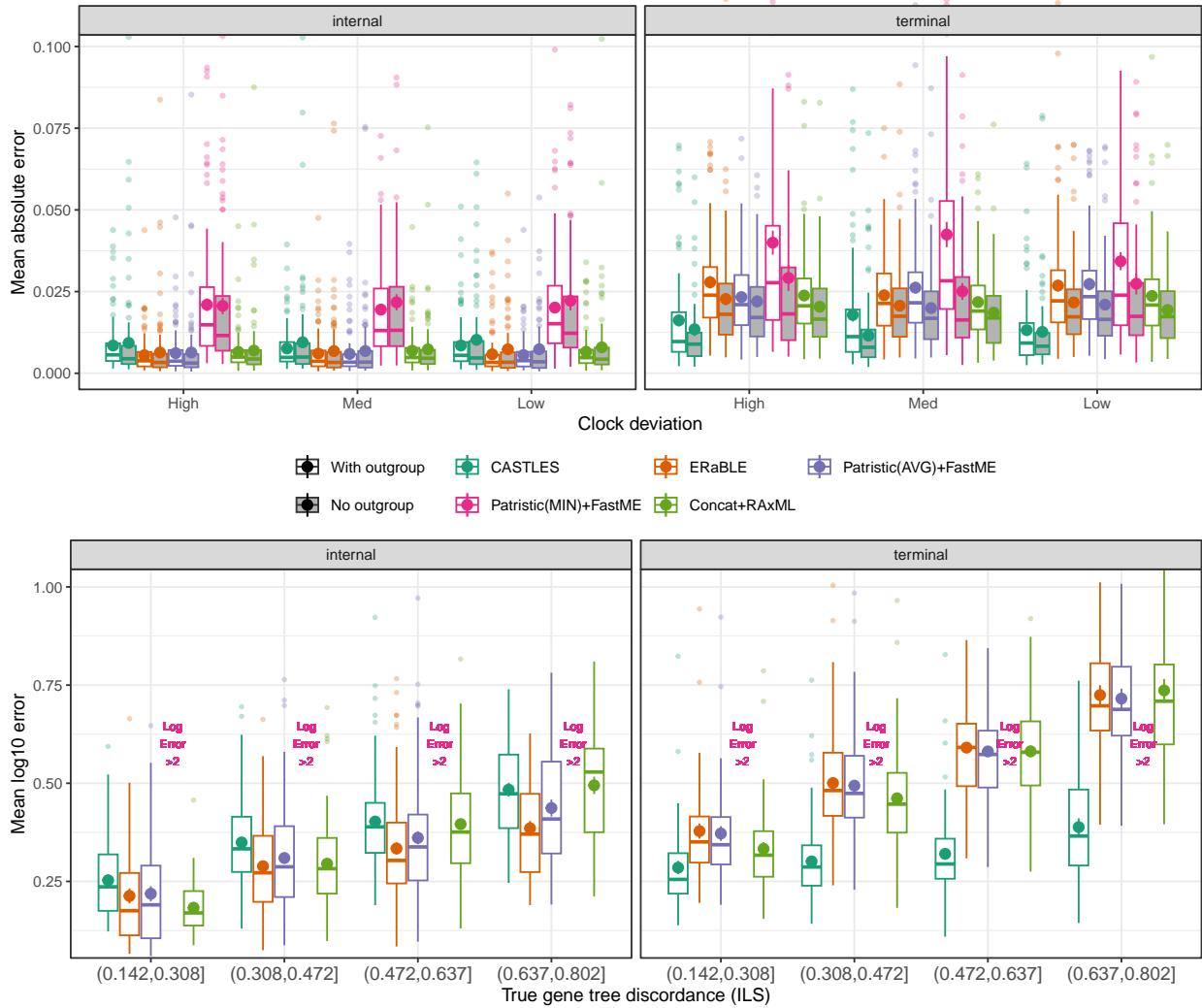


Figure 5.16: Mean absolute error (top) and mean log error (bottom) on [terminal](#) and [internal](#) branches for simulated MVRoot datasets. The plot shows mean and standard error across 100 replicates, in addition to boxplots. The y-axis is cut at 0.11, leaving a few outliers out of the graph (one from CASTLES). Patristic(MIN)+FastME has mean log error above 2 (see Fig. 5.24) and is excluded. Figure 5.9 in the main text shows the same errors aggregated for [terminal](#) and [internal](#) branches.

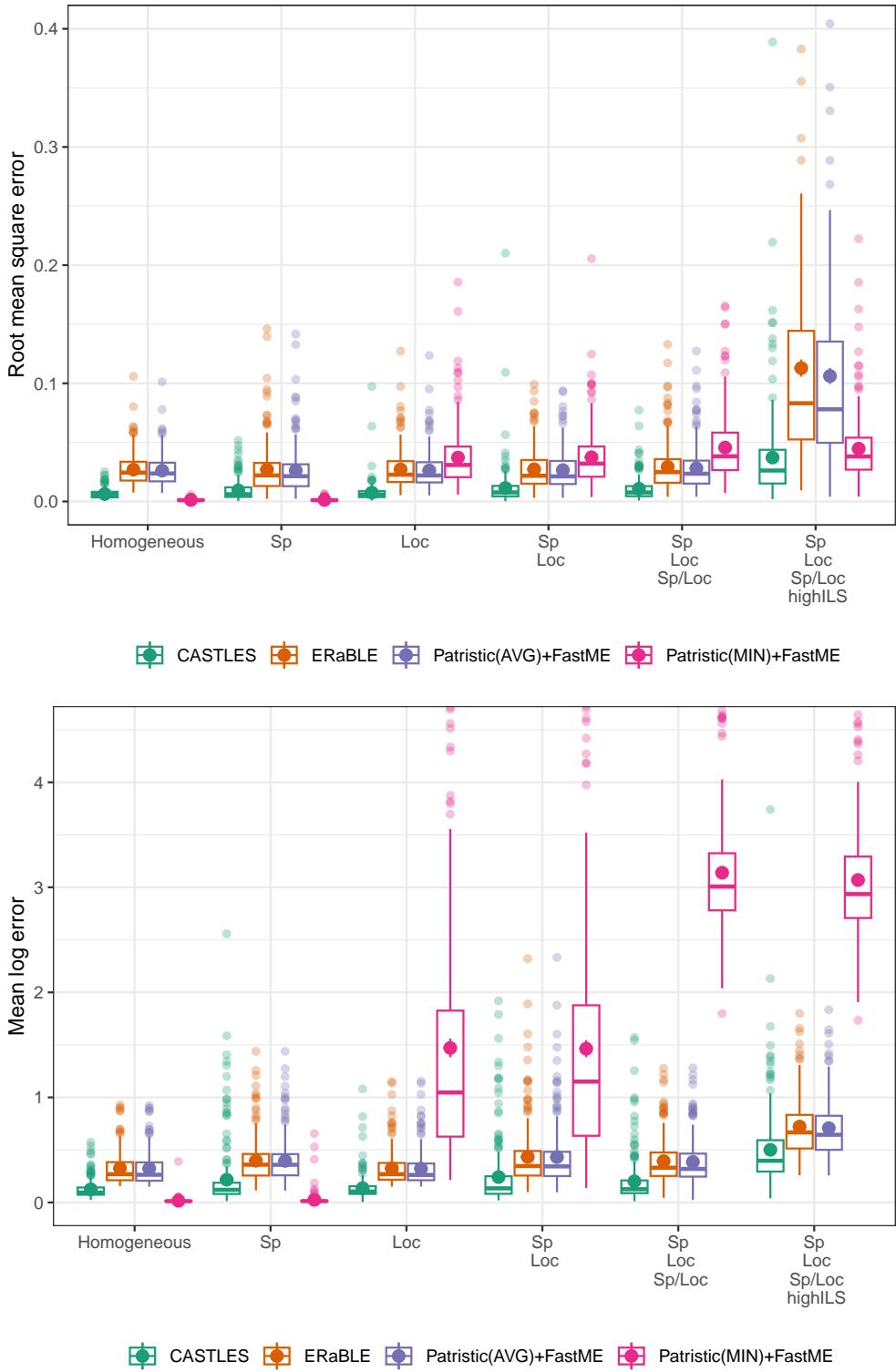


Figure 5.17: RMSE (top) and mean log error (bottom) of branch lengths estimated using different methods on simulated quartet datasets. Both panels show mean and standard deviation across 200 replicates, in addition to boxplots. The y-axes are cut at 0.4 and 4.5 for the top and bottom panels respectively, eliminating a few outlier cases with unusually high errors (none from CASTLES).

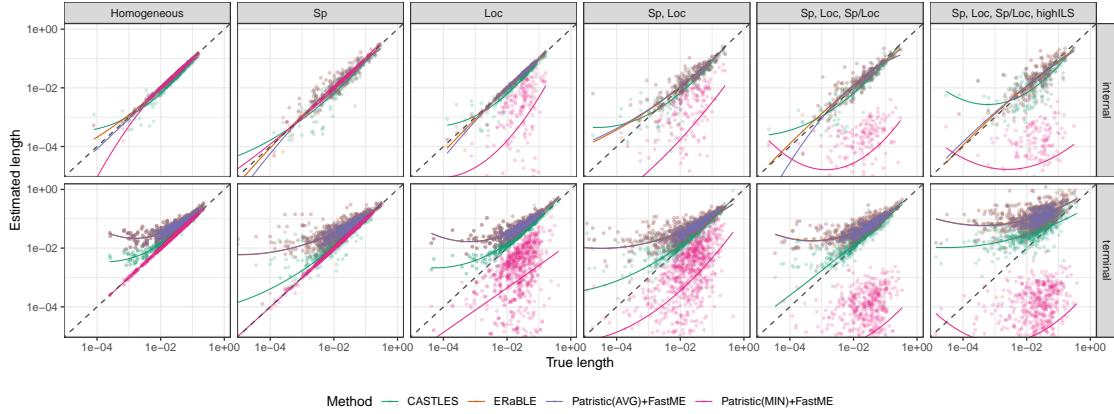


Figure 5.18: Correlations between true and estimated branch lengths on the quartet datasets. Each dot corresponds to a single branch in an [unrooted](#) quartet species tree, and the results are shown across 200 replicates (therefore 5×200 points for each condition and method). The lines show a fitted degree-two polynomial with smoothing.

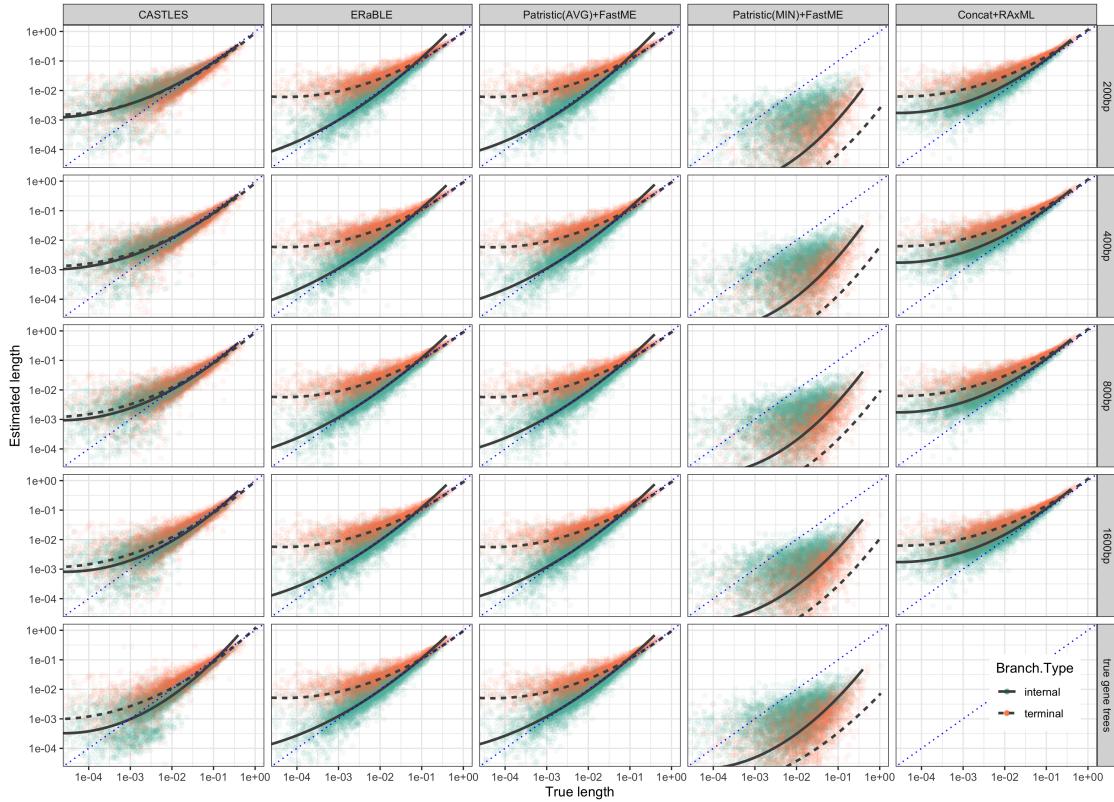


Figure 5.19: Correlations between true and estimated branch lengths on the S100 datasets. Each dot corresponds to a single branch in an [unrooted](#) 101-taxon species tree, and the results are shown across 50 replicates (therefore 50×199 points in each subfigure). The lines show a fitted degree-two polynomial with smoothing.

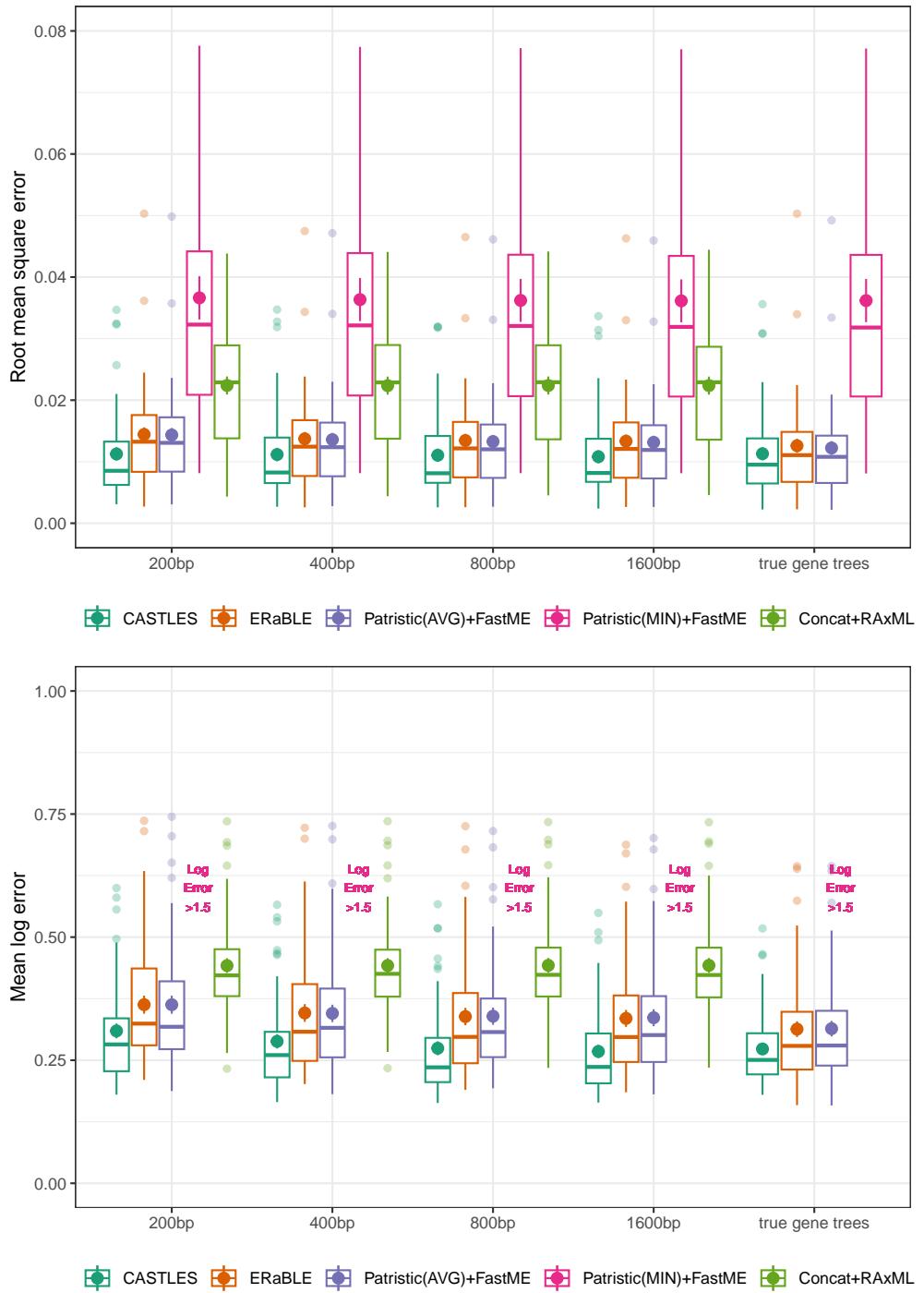


Figure 5.20: RMSE (top) and mean log error (bottom) of branch lengths estimated using different methods on simulated S100 datasets. Both panels show mean and standard deviation across 50 replicates, in addition to boxplots. The average GTEE level varies between zero (for true gene trees) to 23% (for 1600bp) and then to 55% (for the 200bp sequences). The average ILS level is 0.46 AD, and the number of genes is 1000.

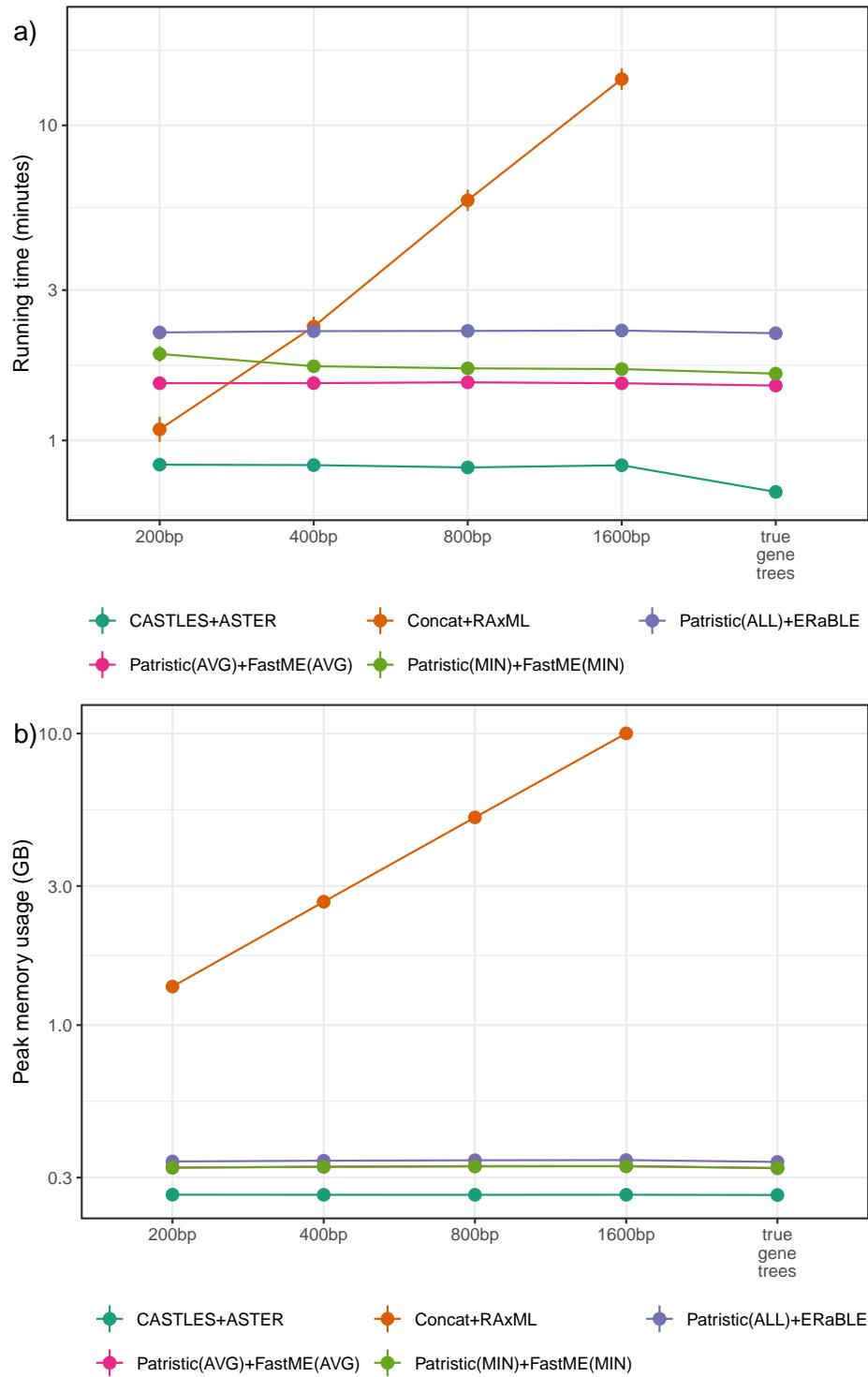


Figure 5.21: (a) Average total running time in minutes and (b) peak memory usage in gigabytes (both in log scale) of different branch length estimation pipelines on the 101-taxon datasets with 1000 genes. The runtimes are reported as the total running time of all steps for each method (see Sec. 5.8.1). The results are averaged over 50 replicates in each model condition.

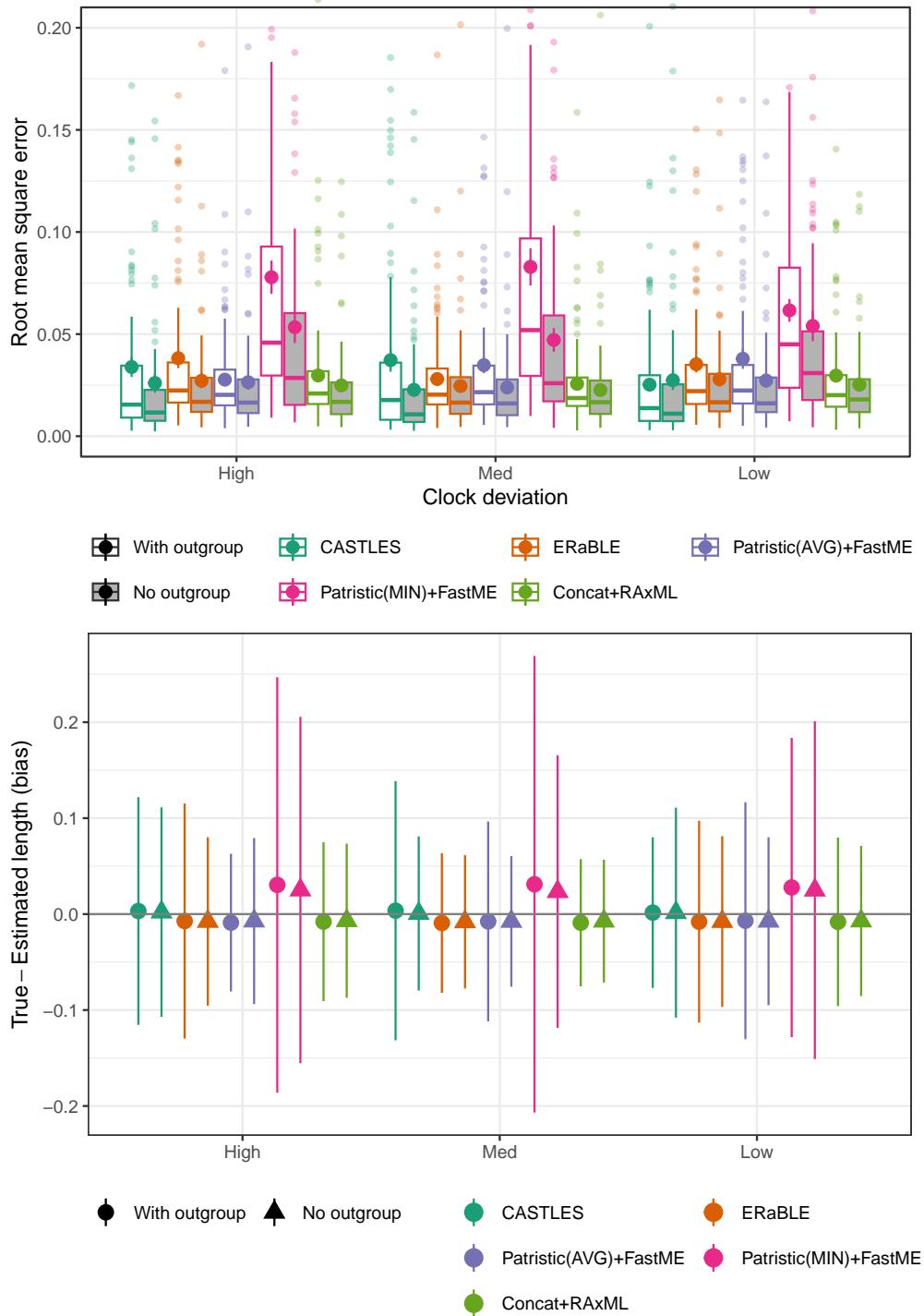


Figure 5.22: RMSE (top) and bias (bottom) of branch lengths estimated using different methods on MVRoot datasets. The top panel shows mean and standard deviation across 100 replicates, in addition to boxplots, and the bottom panel shows mean and standard deviation. The number of genes is 500 and the results are shown across 100 replicates.

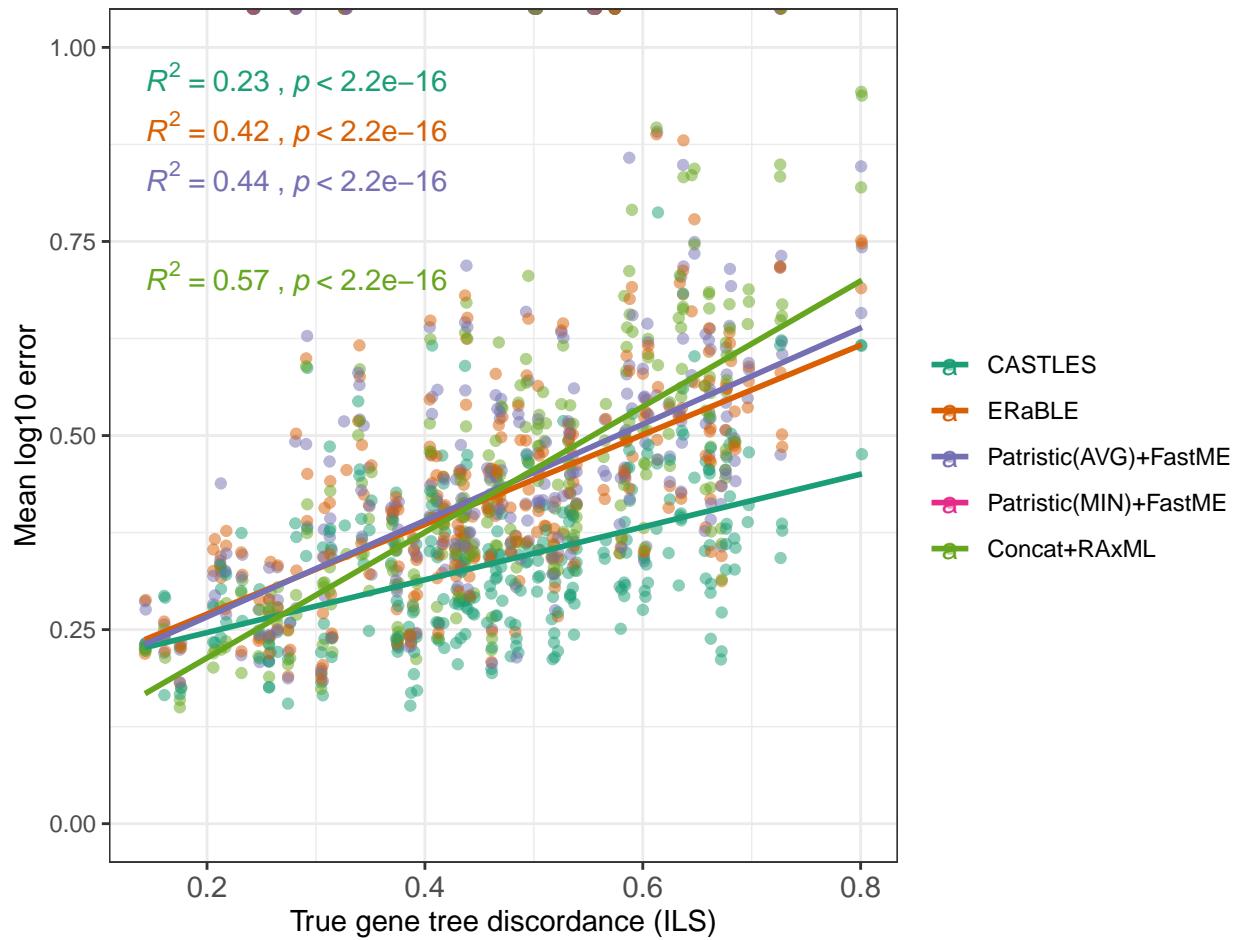


Figure 5.23: Mean log error (in base 10) of branch lengths estimated using different methods (excluding Patristic(MIN)+FastME) versus ILS measured using AD (mean RF distance of true gene trees to the model species tree) on the MVRoot dataset in the no-outgroup model conditions. The number of genes is 500 and the results are shown across 100 replicates.

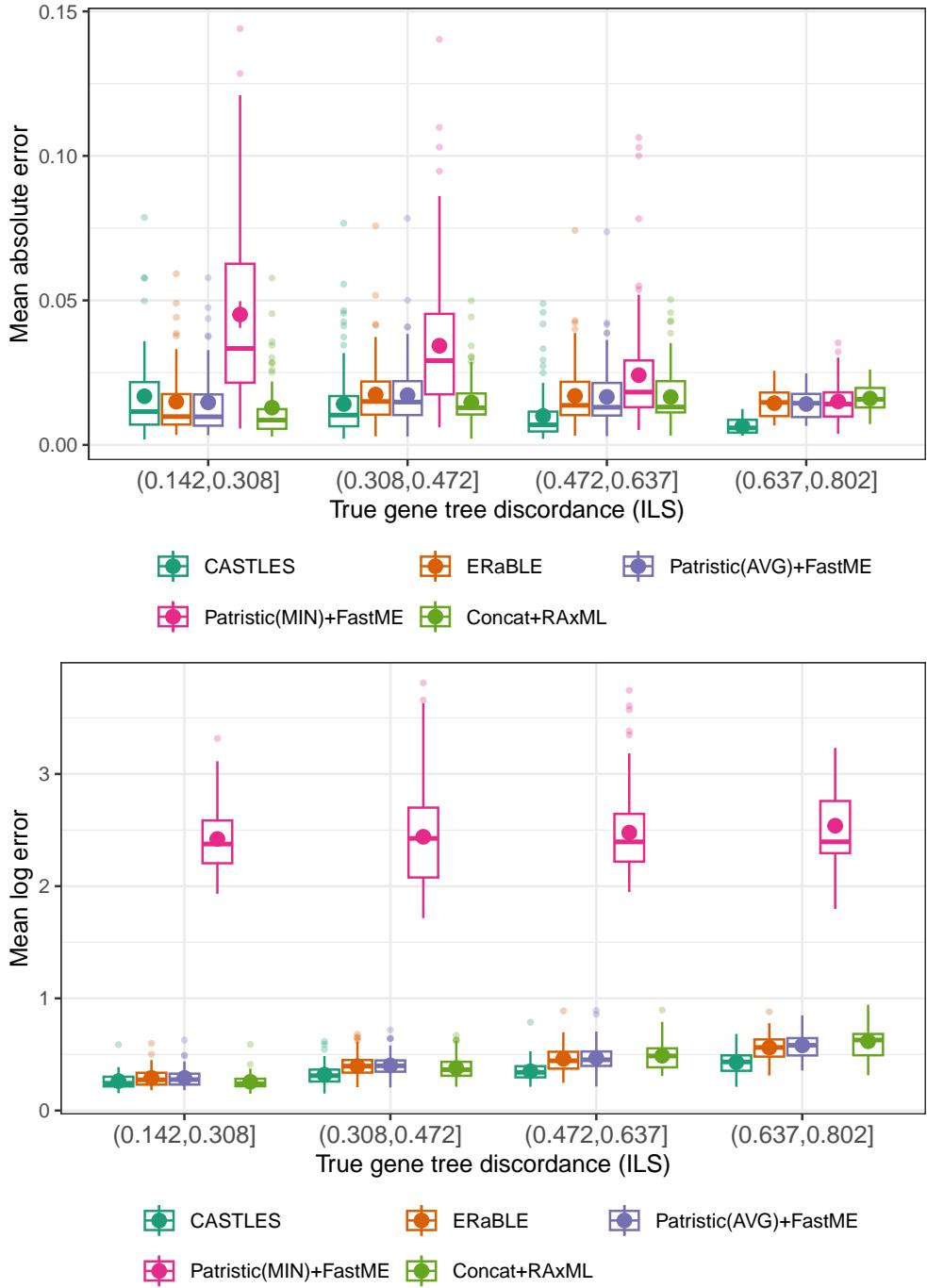


Figure 5.24: Mean absolute error (top) and mean log error (bottom) of branch lengths estimated using different methods on MVRoot datasets. Focusing on cases without outgroups, we divide replicates based on their level of true gene tree discordance due to ILS into four groups. The number of genes is 500 and the results are shown across 100 replicates. Figure 5.9 in the main text excluded Patristic(MIN)+FastME due to very high error.

Table 5.1: Summary of formulas for expected branch lengths in matching and non-matching gene trees.

Unbalanced

| Parameter | Formula | Derivation |
|------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------|
| L_I | $\frac{(e^{-3T_2}+3e^{-T_2}-6e^{T_1-T_2})(\mu_2-\mu_3)+6(1-e^{T_1}+T_1e^{T_1})\mu_1}{2(3e^{T_1}-2)} + \mu_2$ | Lemma 5.1 (Eq. 5.5) |
| L'_I | $\mu_2 + \frac{1}{2}(\mu_2 - \mu_3) (e^{-3T_2} - 3e^{-T_2})$ | Lemma 5.1 (Eq. 5.6) |
| Δ_I | $\frac{3(e^{-T_2}-e^{-3T_2})(1-e^{-T_1})(\mu_2-\mu_3)+6\mu_1(e^{-T_1}-1+T_1)}{2(3-2e^{-T_1})}$ | Theorem 5.1 (Eq. 5.25) |
| L_A | $\frac{6T_1\mu_1+3\mu_1-\mu_2+e^{-3T_2}(\mu_2-2\mu_3)}{6-9e^{T_1}} + \mu_1 + \mu_A T_A$ | Lemma 5.2 (Eq. 5.10) |
| L'_A | $\frac{1}{12}(10\mu_2 - 9e^{-T_2}(\mu_2 - \mu_3) - 3^{-3T_2}(\mu_2 + \mu_3)) + T_1\mu_1 + T_A\mu_A$ | Lemma 5.2 (Eq. 5.26) |
| Δ_A | $\frac{(4\mu_2-6\mu_1-(3e^{-T_2}+e^{-3T_2})(\mu_2-\mu_3)+e^{T_1}(\frac{1}{2}(\mu_2+\mu_3)e^{-3T_2}+\frac{3}{2}e^{-T_2}(\mu_2-\mu_3)-6T_1\mu_1+6\mu_1-5\mu_2))}{2(-2+3e^{T_1})}$ | Theorem 5.1 (Eq. 5.28) |
| L_C | $-e^{-T_2}(\mu_2 - \mu_3) + \mu_2 + \mu_C T_C + \frac{2\mu_2-(3e^{-T_2}-e^{-3T_2})(\mu_2-\mu_3)-4\mu_3e^{-3T_2}}{6(3e^{T_1}-2)}$ | Lemma 5.3 (Eq. 5.15) |
| L'_C | $\frac{1}{3}\mu_2(1+e^{-3T_2}) + \mu_C T_C$ | Lemma 5.3 (Eq. 5.16) |
| Δ_C | $\frac{(2-e^{-T_1})(e^{-3T_2}+2)\mu_2-3e^{-T_2}(\mu_2-\mu_3))+\mu_3e^{-3T_2}(e^{-T_1}-4)}{2(3-2e^{-T_1})}$ | Theorem 5.1 (Eq. 5.29) |
| L_D | $e^{-T_2}(\mu_2 - \mu_3) - \mu_2 + 2\mu_3 + T_2\mu_2 + \mu_D T_D + \frac{-2\mu_2+(3e^{-T_2}-e^{-3T_2})(\mu_2-\mu_3)}{6(3e^{T_1}-2)}$ | Lemma 5.4 (Eq. 5.19) |
| L'_D | $\frac{(\frac{3}{2}e^{-T_2}-\frac{1}{6}e^{-3T_2})(\mu_2-\mu_3)-\frac{4}{3}\mu_2+2\mu_3+T_2\mu_2+\mu_D T_D}{(1-e^{-T_1})(2\mu_2-(3e^{-T_2}-e^{-3T_2})(\mu_2-\mu_3))}$ | Lemma 5.4 (Eq. 5.20) |
| Δ_D | $\frac{2(3-2e^{-T_1})}{2(3-2e^{-T_1})}$ | Theorem 5.1 (Eq. 5.30) |

Balanced

| | | |
|------------|---------------------------------------------------------------------------------------------------------------------------------------|------------------------|
| L_I | $\frac{3e^{-T_1}(\mu_1-\mu_3)+\mu_3e^{-(T_1+T_2)}+3e^{-T_2}(\mu_2-\mu_3)+3(T_1\mu_1+T_2\mu_2-\mu_1-\mu_2+2\mu_3)}{3-2e^{-(T_1+T_2)}}$ | Lemma 5.5 (Eq. 5.44) |
| L'_I | μ_3 | Lemma 5.5 (Eq. 5.45) |
| Δ_I | $\frac{3(e^{-T_1}(\mu_1-\mu_3)+\mu_3e^{-(T_1+T_2)}+e^{-T_2}(\mu_2-\mu_3)+(T_1\mu_1+T_2\mu_2-\mu_1-\mu_2+\mu_3))}{3-2e^{-(T_1+T_2)}}$ | Theorem 5.2 (Eq. 5.53) |
| L_A | $\frac{e^{-(T_1+T_2)}(-6T_1\mu_1-7\mu_3)+9((1-e^{-T_1})\mu_1+\mu_3e^{-T_1})}{9-6e^{-(T_1+T_2)}} + \mu_A T_A$ | Lemma 5.6 (Eq. 5.49) |
| L'_A | $T_1\mu_1 + \frac{2}{3}\mu_3 + \mu_A T_A$ | Lemma 5.6 (Eq. 5.48) |
| Δ_A | $\frac{-\mu_3e^{-(T_1+T_2)}+3\mu_1(1-e^{-(T_1)-T_1})+\mu_3(-2+3e^{-(T_1)})}{-2e^{-(T_1+T_2)}+3}$ | Theorem 5.2 (Eq. 5.55) |

Table 5.2: Summary of simplifying assumptions and the corresponding simplified formulas for expected branch lengths in matching and non-matching quartet gene trees.

Unbalanced

| Parameter | Simplified formula | Simplifying assumption |
|------------|-----------------------------------------------------------------------------------------------------------|---------------------------|
| L_I | $\lim_{\mu_3 \rightarrow \mu_2} L_I = \frac{3\mu_1(e^{-T_1}-1+T_1)}{3-2e^{-T_1}} + \mu_2$ | $\mu_3 \rightarrow \mu_2$ |
| L'_I | $\lim_{\mu_3 \rightarrow \mu_2} L'_I = \mu_2$ | $\mu_3 \rightarrow \mu_2$ |
| Δ_I | $\lim_{\mu_3 \rightarrow \mu_2} \Delta_I = \frac{3\mu_1(e^{-T_1}-1+T_1)}{3-2e^{-T_1}}$ | $\mu_3 \rightarrow \mu_2$ |
| L_A | $\lim_{T_2 \rightarrow \infty} L_A = \frac{6T_1\mu_1+3\mu_1-\mu_2}{6-9e^{T_1}} + \mu_1 + \mu_A T_A$ | $T_2 \rightarrow \infty$ |
| L'_A | $\lim_{T_2 \rightarrow \infty} L'_A = \frac{5}{6}\mu_2 + T_1\mu_1 + T_A\mu_A$ | $T_2 \rightarrow \infty$ |
| Δ_A | $\lim_{T_2 \rightarrow \infty} \Delta_A = \frac{-6\mu_1(e^{-T_1}-1+T_1)-(5-4e^{-T_1})\mu_2}{6-4e^{-T_1}}$ | $T_2 \rightarrow \infty$ |
| L_C | $\lim_{T_2 \rightarrow \infty} L_C = \mu_2 + \mu_C T_C + \frac{\mu_2}{3(3e^{T_1}-2)}$ | $T_2 \rightarrow \infty$ |
| L'_C | $\lim_{T_2 \rightarrow \infty} L'_C = \frac{1}{3}\mu_2 + \mu_C T_C$ | $T_2 \rightarrow \infty$ |
| Δ_C | $\lim_{T_2 \rightarrow \infty} \Delta_C = \frac{(2-e^{-T_1})\mu_2}{(3-2e^{-T_1})}$ | $T_2 \rightarrow \infty$ |
| L_D | $\lim_{\mu_3 \rightarrow \mu_2} L_D = \mu_2 + T_2\mu_2 + \mu_D T_D - \frac{\mu_2}{3(3e^{T_1}-2)}$ | $\mu_3 \rightarrow \mu_2$ |
| L'_D | $\lim_{\mu_3 \rightarrow \mu_2} L'_D = \frac{2}{3}\mu_2 + T_2\mu_2 + \mu_D T_D$ | $\mu_3 \rightarrow \mu_2$ |
| Δ_D | $\lim_{\mu_3 \rightarrow \mu_2} \Delta_D = \frac{(1-e^{-T_1})\mu_2}{3-2e^{-T_1}}$ | $\mu_3 \rightarrow \mu_2$ |

Balanced

| | | |
|------------|------------------------------------------------------------------------------------------------------|---------------------------|
| L_I | $\lim_{T_2 \rightarrow 0} L_I = \frac{3\mu_1(e^{-T_1}-1+T_1)}{3-2e^{-T_1}} + \mu_3$ | $T_2 \rightarrow 0$ |
| L'_I | μ_3 | — |
| Δ_I | $\lim_{T_2 \rightarrow 0} \Delta_I = \frac{3\mu_1(e^{-T_1}-1+T_1)}{3-2e^{-T_1}}$ | $T_2 \rightarrow 0$ |
| L_A | $\lim_{\mu_3 \rightarrow \mu_1} L_A = \mu_1 + \frac{\mu_1(1+6T_1)}{6-9e^{(T_1+T_2)}} + \mu_A T_A$ | $\mu_3 \rightarrow \mu_1$ |
| L'_A | $\lim_{\mu_3 \rightarrow \mu_1} L'_A = T_1\mu_1 + \frac{2}{3}\mu_1 + \mu_A T_A$ | $\mu_3 \rightarrow \mu_1$ |
| Δ_A | $\lim_{\mu_3 \rightarrow \mu_1} \Delta_A = \frac{\mu_1(-e^{-(T_1+T_2)}+1-3T_1)}{-2e^{-(T_1+T_2)}+3}$ | $\mu_3 \rightarrow \mu_1$ |

Table 5.3: Summary of formulas for estimating unbalanced or balanced quartet species tree branch lengths in SU.

Unbalanced

| Parameter | Estimation formula | Simplifying assumption(s) |
|-------------|----------------------------------------------------------------------------------------------------------------------|----------------------------------------------------|
| t_1 | $\hat{t}_1 = \bar{L}'_I \left(\frac{1}{2} \bar{\delta} + \frac{1}{6} \sqrt{3\bar{\delta}(3\bar{\delta}+4)} \right)$ | $\mu_3 \rightarrow \mu_2; \mu_1 \rightarrow \mu_2$ |
| t_A | $\hat{t}_A = \bar{L}'_A + \frac{\mu_1(e^{-T_1}-1+T_1)+\bar{\Delta}_A(1-2/3e^{-T_1})}{1-4/5e^{-T_1}} - T_1\mu_1$ | $T_2 \rightarrow \infty$ |
| t_B | $\hat{t}_B = \bar{L}'_B + \frac{\mu_1(e^{-T_1}-1+T_1)+\bar{\Delta}_B(1-2/3e^{-T_1})}{1-4/5e^{-T_1}} - T_1\mu_1$ | $T_2 \rightarrow \infty$ |
| t_C | $\hat{t}_C = \bar{L}'_C - \frac{1}{3}(2 - \frac{1}{2-e^{-T_1}})\bar{\Delta}_C$ | $T_2 \rightarrow \infty$ |
| $t_2 + t_D$ | $\hat{t}_2 + \hat{t}_D = \bar{L}'_D - \frac{2}{3}(2 + \frac{1}{1-e^{-T_1}})\bar{\Delta}_D$ | $\mu_3 \rightarrow \mu_2$ |

Balanced

| | | |
|-------------|----------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------|
| $t_1 + t_2$ | $\hat{t}_1 + \hat{t}_2 = \bar{L}'_I \left(\frac{1}{2} \bar{\delta} + \frac{1}{6} \sqrt{3\bar{\delta}(3\bar{\delta}+4)} \right)$ | $T_2 \rightarrow 0; \mu_1 \rightarrow \mu_3$ |
| t_A | $\hat{t}_A = \bar{L}'_A - \frac{2}{3}\mu_1 - \frac{1}{3}(\mu_1(1 - e^{-(T_1+T_2)}) - \bar{\Delta}_A(3 - 2e^{-(T_1+T_2)}))$ | $\mu_3 \rightarrow \mu_1$ |
| t_B | $\hat{t}_B = \bar{L}'_B - \frac{2}{3}\mu_1 - \frac{1}{3}(\mu_1(1 - e^{-(T_1+T_2)}) - \bar{\Delta}_B(3 - 2e^{-(T_1+T_2)}))$ | $\mu_3 \rightarrow \mu_1$ |
| t_C | $\hat{t}_C = \bar{L}'_C - \frac{2}{3}\mu_2 - \frac{1}{3}(\mu_2(1 - e^{-(T_1+T_2)}) - \bar{\Delta}_C(3 - 2e^{-(T_1+T_2)}))$ | $\mu_3 \rightarrow \mu_2$ |
| t_D | $\hat{t}_D = \bar{L}'_D - \frac{2}{3}\mu_2 - \frac{1}{3}(\mu_2(1 - e^{-(T_1+T_2)}) - \bar{\Delta}_D(3 - 2e^{-(T_1+T_2)}))$ | $\mu_3 \rightarrow \mu_2$ |

Table 5.4: We define several counters for every node w , each of which computes $\sum_{e \in S(w)} f(e)$ for some S and f . We define several notations: let p denote the parent node of w and u, v denote the child nodes of w ; let $\mathcal{L}(w)$ denote the set of leaves under w ; let $\mathcal{D}(\cdot, \cdot)$ denote the distance of two nodes; let $\mathcal{M}(\cdot, \cdot)$ denote the most recent common ancestor of two nodes; let $\mathcal{H}(\cdot, \cdot)$ be the 0/1 indicator of whether all branches on the path between the two nodes are ghost branches. Superscripts $-$ and $+$ signify ghost branches and all branches, respectively. Note that A and B are interchangeable in the names; e.g., $C_{BA}^{1+}(w)$ is defined similarly to $C_{AB}^{1+}(w)$. Table continues to the next page. See Figures 5.12 and 5.13 for illustrations of recursive formulas.

| Counter | Set $S(w)$ | Function $f(e)$ | Recursive formula | Similarly defined |
|------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------|--------------------------------------------------------------------------------------------|-----------------------------------------|
| $C_A^1(w)$ | $a : a \in A \cap \mathcal{L}(w)$ | 1 | $C_A^1(u) + C_A^1(v)$ | $C_B^1(w)$... |
| $C_A^a(w)$ | $a : a \in A \cap \mathcal{L}(w)$ | $\mathcal{D}(a, p)$ | $C_A^a(u) + C_A^a(v) + \mathcal{D}(w, p)C_A^1(w)$ | $C_B^b(w)$... |
| $C_{AB}^{1+}(w)$ | $(a, b) : a \in A \cap \mathcal{L}(w), b \in B \cap \mathcal{L}(w)$ | 1 | $C_{AB}^{1+}(u) + C_{AB}^{1+}(v) + C_A^1(u)C_B^1(v) + C_B^1(u)C_A^1(v)$ | $C_{AC}^{1+}(w)$... |
| $C_{AB}^{1-}(w)$ | $(a, b) : a \in A \cap \mathcal{L}(w), b \in B \cap \mathcal{L}(w), \mathcal{H}(\mathcal{M}(a, b), p) = 1$ | 1 | $(C_{AB}^{1-}(u) + C_{AB}^{1-}(v) + C_A^1(u)C_B^1(v) + C_B^1(u)C_A^1(v))\mathcal{H}(w, p)$ | $C_{AC}^{1-}(w)$... |
| $C_{AB}^1(w)$ | $(a, b) : a \in A \cap \mathcal{L}(w), b \in B \cap \mathcal{L}(w), \mathcal{H}(\mathcal{M}(a, b), p) \neq 1$ | 1 | $C_{AB}^{1+}(w) - C_{AB}^{1-}(w)$ | $C_{AC}^1(w)$... |
| $C_{AB}^{a+}(w)$ | $(a, b) : a \in A \cap \mathcal{L}(w), b \in B \cap \mathcal{L}(w)$ | $\mathcal{D}(a, \mathcal{M}(a, b))$ | $C_{AB}^{a+}(u) + C_{AB}^{a+}(v) + C_A^a(u)C_B^1(v) + C_B^1(u)C_A^a(v)$ | $C_{AB}^{b+}(w), C_{AC}^{a+}(w)$... |
| $C_{AB}^{a-}(w)$ | $(a, b) : a \in A \cap \mathcal{L}(w), b \in B \cap \mathcal{L}(w), \mathcal{H}(\mathcal{M}(a, b), p) = 1$ | $\mathcal{D}(a, \mathcal{M}(a, b))$ | $(C_{AB}^{a-}(u) + C_{AB}^{a-}(v) + C_A^a(u)C_B^1(v) + C_B^1(u)C_A^a(v))\mathcal{H}(w, p)$ | $C_{AB}^{b-}(w), C_{AC}^{a-}(w)$... |
| $C_{AB}^a(w)$ | $(a, b) : a \in A \cap \mathcal{L}(w), b \in B \cap \mathcal{L}(w), \mathcal{H}(\mathcal{M}(a, b), p) \neq 1$ | $\mathcal{D}(a, \mathcal{M}(a, b))$ | $C_{AB}^a(w) - C_{AB}^{a-}(w)$ | $C_{AB}^b(w), C_{AC}^a(w)$... |
| $C_{AB}^{x+}(w)$ | $(a, b) : a \in A \cap \mathcal{L}(w), b \in B \cap \mathcal{L}(w)$ | $\mathcal{D}(\mathcal{M}(a, b), p)$ | $C_{AB}^{x+}(u) + C_{AB}^{x+}(v) + \mathcal{D}(w, p)C_{AB}^{1+}(w)$ | $C_{AC}^{x+}(w)$... |
| $C_{AB}^{x-}(w)$ | $(a, b) : a \in A \cap \mathcal{L}(w), b \in B \cap \mathcal{L}(w), \mathcal{H}(\mathcal{M}(a, b), p) = 1$ | $\mathcal{D}(\mathcal{M}(a, b), p)$ | $(C_{AB}^{x-}(u) + C_{AB}^{x-}(v) + \mathcal{D}(w, p)C_{AB}^{1-}(w))\mathcal{H}(w, p)$ | $C_{AC}^{x-}(w)$... |
| $C_{AB}^x(w)$ | $(a, b) : a \in A \cap \mathcal{L}(w), b \in B \cap \mathcal{L}(w), \mathcal{H}(\mathcal{M}(a, b), p) \neq 1$ | $\mathcal{D}(\mathcal{M}(a, b), p)$ | $C_{AB}^{x+}(w) - C_{AB}^{x-}(w)$ | $C_{AC}^x(w)$... |
| $C_{C AB}^1(w)$ | $(a, b, c) : a \in A \cap \mathcal{L}(w), b \in B \cap \mathcal{L}(w), c \in C \cap \mathcal{L}(w), \mathcal{L}(\mathcal{M}(a, b)) \subsetneq \mathcal{L}(\mathcal{M}(a, c)), \mathcal{H}(\mathcal{M}(a, b), \mathcal{M}(a, c)) \neq 1$ | 1 | $C_{C AB}^1(u) + C_{C AB}^1(v) + C_{AB}^1(u)C_C^1(v) + C_C^1(u)C_{AB}^1(v)$ | $C_{A BC}^1(w)$... |

Table 5.5: Counters for recursive formulas (continued). Here, we omit the column $S(w)$. For all $C_{AB|CD}(w)$ shown below, $S(w)$ is the set of quartets a, b, c, d , whose MRCA is w , form a tree with topology $ab|cd$, and are elements of the sets A, B, C, D , respectively.

| Counter | Function $f(e)$ | Recursive formula | Similarly defined |
|------------------|-----------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------|
| $C_{C AB}^a(w)$ | $\mathcal{D}(a, \mathcal{M}(a, b))$ | $C_{C AB}^a(u) + C_{C AB}^a(v) + C_{AB}^a(u)C_C^1(v) + C_C^1(u)C_{AB}^a(v)$ | $C_{C AB}^b(w),$ $C_{A BC}^b(w)$... |
| $C_{C AB}^c(w)$ | $\mathcal{D}(c, \mathcal{M}(a, c))$ | $C_{C AB}^c(u) + C_{C AB}^c(v) + C_{AB}^1(u)C_C^c(v) + C_C^c(u)C_{AB}^1(v)$ | $C_{A BC}^a(w)$... |
| $C_{C AB}^d(w)$ | $\mathcal{D}(\mathcal{M}(a, c), p)$ | $C_{C AB}^d(u) + C_{C AB}^d(v) + \mathcal{D}(w, p)C_{C AB}^1(w)$ | $C_{A BC}^d(w)$... |
| $C_{C AB}^x(w)$ | $\mathcal{D}(\mathcal{M}(a, b), \mathcal{M}(a, c))$ | $C_{C AB}^x(u) + C_{C AB}^x(v) + C_{AB}^x(u)C_C^1(v) + C_C^1(u)C_{AB}^x(v)$ | $C_{A BC}^x(w)$... |
| $C_{AB CD}^1(w)$ | 1 | $C_{C AB}^1(u)C_D^1(v) + C_D^1(u)C_{C AB}^1(v) + C_{D AB}^1(u)C_C^1(v) + C_C^1(u)C_{D AB}^1(v) + C_{B CD}^1(u)C_A^1(v) + C_A^1(u)C_{B CD}^1(v) + C_{A CD}^1(u)C_B^1(v) + C_B^1(u)C_{A CD}^1(v) + C_{AB}^{1+}(u)C_{CD}^{1+}(v) - C_{AB}^{1-}(u)C_{CD}^{1-}(v) + C_{CD}^{1+}(u)C_{AB}^{1+}(v) - C_{CD}^{1-}(u)C_{AB}^{1-}(v)$ | $C_{AB CD}^1(w),$ $C_{AC BD}^1(w)$ |
| $C_{AB CD}^a(w)$ | $\mathcal{D}(a, \mathcal{M}(a, b))$ | $C_{C AB}^a(u)C_D^1(v) + C_D^1(u)C_{C AB}^a(v) + C_{D AB}^a(u)C_C^1(v) + C_C^1(u)C_{D AB}^a(v) + C_{B CD}^a(u)C_A^1(v) + C_A^1(u)C_{B CD}^a(v) + C_{A CD}^a(u)C_B^1(v) + C_B^1(u)C_{A CD}^a(v) + C_{AB}^{a+}(u)C_{CD}^{a+}(v) - C_{AB}^{a-}(u)C_{CD}^{a-}(v) + C_{CD}^{a+}(u)C_{AB}^{a+}(v) - C_{CD}^{a-}(u)C_{AB}^{a-}(v)$ | $C_{AB CD}^b(w),$ $C_{AC BD}^a(w)$... |
| $C_{AB CD}^x(w)$ | $\mathcal{D}(\mathcal{M}(a, b), \mathcal{M}(c, d))$ | $C_{C AB}^x(u)C_D^1(v) + C_D^1(u)C_{C AB}^x(v) + C_{D AB}^x(u)C_C^1(v) + C_C^1(u)C_{D AB}^x(v) + C_{B CD}^x(u)C_A^1(v) + C_A^1(u)C_{B CD}^x(v) + C_{A CD}^x(u)C_B^1(v) + C_B^1(u)C_{A CD}^x(v) + C_{AB}^{x+}(u)C_{CD}^{x+}(v) - C_{AB}^{x-}(u)C_{CD}^{x-}(v) + C_{CD}^{x+}(u)C_{AB}^{x+}(v) - C_{CD}^{x-}(u)C_{AB}^{x-}(v) + C_{CD}^{1+}(u)C_{AB}^{1+}(v) - C_{CD}^{1-}(u)C_{AB}^{1-}(v)$ | $C_{AC BD}^x(w),$ $C_{AD BC}^x(w)$ |

Table 5.6: Parameters used in SimPhy quartet simulations for all model conditions.

| Arg. | Description | Value |
|------|---------------------------------------------------------|------------------------|
| RS | Number of replicates | 200 |
| RL | Number of loci | 10000 |
| RG | Number of genes | 1 |
| ST | Maximum tree length | 400 |
| SL | Number of taxa | 4 |
| SB | Speciation rate | 0.0001 |
| SD | Extinction rate | 0 |
| SP | Global population size | 200 or 800 |
| SU | Global substitution rate | LogNormal (-9,0.5) |
| HS | Species-specific branch rate heterogeneity modifiers | 1 or LogNormal (1.5,1) |
| HL | Gene-family-specific rate heterogeneity modifiers | 1 or LogNormal (1.5,1) |
| HG | Gene by lineage specific rate heterogeneity modifiers | 1 or LogNormal (1.5,1) |
| HH | Gene-by-lineage-specific locus tree parameter | 1 |
| SO | Outgroup branch length relative to half the tree length | 0 (no outgroup) |
| CS | Random number generator seed | 12964 |

Table 5.7: Characteristics of the quartet simulation model conditions. **AD** stands for average discordance ([RF](#) distance) between the model species tree and true gene trees.

| Condition | SP | HS | HL | HG | AD |
|-----------------------|-----|-------------------|-------------------|-------------------|------|
| Homogeneous | 200 | 1 | 1 | 1 | 0.28 |
| Sp | 200 | LogNormal (1.5,1) | 1 | 1 | 0.30 |
| Loc | 200 | 1 | LogNormal (1.5,1) | 1 | 0.27 |
| Sp,Loc | 200 | LogNormal (1.5,1) | LogNormal (1.5,1) | 1 | 0.29 |
| Sp,Loc,Sp/Loc | 200 | LogNormal (1.5,1) | LogNormal (1.5,1) | LogNormal (1.5,1) | 0.29 |
| Sp,Loc,Sp/Loc,highILS | 800 | LogNormal (1.5,1) | LogNormal (1.5,1) | LogNormal (1.5,1) | 0.51 |

CHAPTER 6: SPECIES TREE BRANCH LENGTH ESTIMATION DESPITE INCOMPLETE LINEAGE SORTING, DUPLICATION, AND LOSS

This chapter contains material that has appeared in the preprint “Y. Tabatabae, C. Zhang, S. Arasti and S. Mirarab. (2025). Species tree branch length estimation despite incomplete lineage sorting, duplication, and loss., bioRxiv , <https://doi.org/10.1101/2025.02.20.639320>”[305]. The CASTLES-Pro software is implemented inside the species tree estimation software package ASTER that is available in open-source form at <https://github.com/chaoszhang/ASTER>. The datasets and scripts used in this study are available at <https://github.com/ytabatabae/CASTLES-Pro-paper>.

6.1 INTRODUCTION

Summarizing a collection of potentially conflicting trees inferred from different parts of the genome (i.e., gene trees) to obtain a species tree has now become a routine analysis. This approach promises to account for biological processes such as Incomplete lineage sorting (ILS), Gene duplication and loss (GDL), and Horizontal gene transfer (HGT) that create discordance between gene trees and the species tree [15]. Prior studies have confirmed that the most accurate methods for estimating the topology of species trees are those that take biological sources of heterogeneity into account [30, 32, 306]. Several methods use likelihood under a model of genome evolution coupled with Bayesian MCMC inference to jointly estimate both topology and branch lengths of gene trees and species trees [38]. These methods tend to be accurate, but they are computationally intensive. One alternative is the two-step approach of inferring gene trees independently and then using a summary method to build a species tree. This alternative has been more scalable and generally accurate [307], spurring the development of many such methods [e.g., 162, 180, 181, 308, 309, 310]. Some of these summary methods [e.g., 90, 197, 311, 312, 313, 314] account for GDL and can take as input multi-copy gene trees, vastly expanding the set of loci that can be used [315]. For example, the ASTRAL family of methods (now available in the ASTER software package; see [316]) use variants of the median tree problem based on the quartet distance [179, 248, 296] and have been extended to multi-copy input trees [312]. The ASTRAL family is widely used, including the ASTRAL-Pro extension to multi-copy input [e.g., 317, 318, 319, 320].

Species trees are most useful if they are furnished with branch lengths, as many downstream applications, including dating, comparative genomics, and the study of diversification and adaptation, depend on branch lengths. However, widely used summary methods such as ASTRAL do not produce the branch lengths needed for downstream analysis. Species trees

can be furnished with [coalescent units \(CU\)](#) lengths only for internal branches [321] (unless multiple individuals are available), and GDL-based methods often produce no branch length. Meanwhile, downstream applications often require branch lengths in the unit of either substitution per site (SU) or time. The standard *ad-hoc* solution is to estimate the species tree [topology](#) using a summary method and then infer the branch lengths using concatenation. Often, branch lengths are optimized on a fixed [topology](#) using maximum likelihood applied to a concatenation of all genes [24, 60, 250]. An alternative is using distance-based approaches, such as ERaBLE [322] and TCMM [75], to summarize [patristic distances](#) from gene trees onto the species tree. Neither concatenation nor distance-based methods directly model the biological processes that create gene tree [heterogeneity](#), reducing their theoretical justification. Nevertheless, these methods have the potential advantage of being agnostic to the source of discordance. Ultimately, which approach should be preferred is an empirical question with important downstream implications [290].

We recently introduced the CASTLES [74] method for estimating [SU](#) branch length for a fixed species tree topology, specifically designed to handle [ILS](#), as modeled by the [Multi-Species Coalescent \(MSC\)](#) model. CASTLES estimates species divergence times as opposed to *genic* divergences, which are expected to be older [323]. CASTLES had higher accuracy than alternatives in our simulations. Nevertheless, it has several limitations. Most importantly, CASTLES is limited to single-copy gene trees and, therefore, cannot be used with multi-copy input trees, severely limiting its applicability. To our knowledge, the only method that can estimate [SU](#) branch lengths from multi-copy gene trees is SpeciesRax [199], which does model [GDL](#) but does not model [ILS](#) and, thus, [deep coalescence](#). In addition, the study by [90] shows that SpeciesRax can be less accurate than ASTRAL-Pro and other methods in conditions with [ILS](#) and is also less scalable. Even the use of concatenation in the presence of [GDL](#) is complicated and requires additional techniques, such as DISCO [90], to decompose multi-copy genes into single-copy ones. Beyond the lack of support for [GDL](#), CASTLES was not tested under conditions with [HGT](#), and even for [ILS](#), it required several approximations that could reduce its accuracy.

In this article, we dramatically advance the CASTLES methodology to address its major limitations and broaden the scope of conditions under which it is tested. We present a dynamic programming algorithm for estimating branch lengths of a species tree from multi-copy gene family trees that have evolved with [GDL](#) in addition to [ILS](#), leading to a new method called CASTLES-Pro. In addition, we improve upon CASTLES by relaxing some approximations and modifying other assumptions. Beyond [ILS](#) and [GDL](#), for which it is designed, we use simulations to test how CASTLES-Pro performs under conditions that include substantial levels of [ILS](#) and [HGT](#). In simulations, we show that the method is accurate,

robust to various sources of [heterogeneity](#), and scalable to thousands of species and genes. On diverse biological data ranging from the root of the tree of life to recent speciations, we show that using CASLTES-Pro instead of concatenation dramatically alters branch lengths. We have incorporated CASTLES-Pro inside the ASTER package of tools [316], providing a new C++ implementation compared to CASTLES. Thus, any user of ASTRAL-Pro or ASTRAL-IV would automatically obtain [SU](#) branch length with no additional step needed.

6.2 CASTLES-PRO

We first review the CASTLES algorithm, on which CASTLES-Pro is based. We then describe how it is extended to handle multi-copy gene family trees and end by explaining the methods used to enhance CASTLES for both single-copy and multi-copy gene trees.

6.2.1 CASTLES

The input to CASTLES is a species tree [topology](#) and a set of k single-copy gene trees with branch lengths in units of the expected number of substitutions per site ([SU](#)). Its output is the species tree with [SU](#) lengths on all branches. The model it assumes is parametrized by a species tree topology, and for each branch i , we are given the [CU](#) length (T_i) and per-branch substitution rate (μ_i) in the unit of substitutions per site per [CU](#). Thus, μ_i increases not only with per generation substitution rate *but also* with the effective population size, N_e . Species branch lengths are multiplied by corresponding substitution rates to give the species tree [SU](#) length $t_i = T_i \mu_i$. We use $t = \langle t_a, t_b, \dots \rangle$ and $\mu = \langle \mu_a, \mu_b, \dots \rangle$ as shorthand. The [CU](#) gene trees are generated from the species tree using the [MSC](#) model. Then, each gene tree branch length is scaled by mutation rates corresponding to all species tree branches that it traverses.

CASTLES is based on expected values of gene tree branch lengths under the [MSC](#). Treating species tree branch lengths t and mutation rates μ as unknown parameters, we can analytically calculate the expected length of the gene tree branches. For a simple cherry tree $(a, b) : T$ with mutation rate μ_a , μ_b for [terminal](#) branches and μ_r for the parent branch, this expected length is

$$\mu_a T + \mu_r = t_a + \mu_r, \quad \text{and} \quad \mu_b T + \mu_r = t_b + \mu_r$$

for [terminal](#) branches of a and b , resp. Note that each [terminal](#) length has an extra μ_r term. This gap between speciation and genic divergence (1 [CU](#) in expectation; i.e., N_e generations)

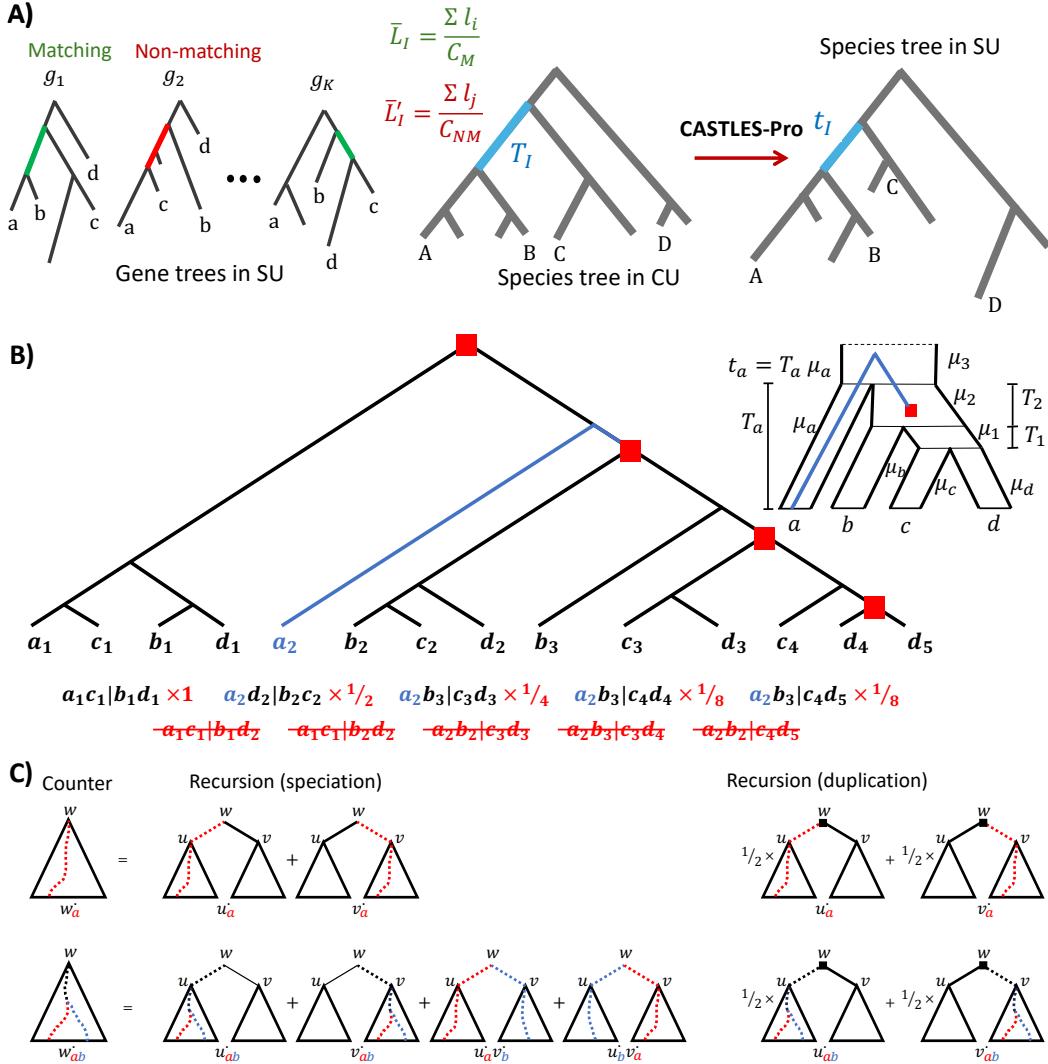


Figure 6.1: A) Illustration of matching and non-matching gene trees and the CASTLES-Pro algorithm. The **SU** length of each species tree branch is calculated using formulas derived from the average **SU** lengths of that branch in quartets **induced** from gene trees that either match or do not match the **topology** of the species tree. B) Illustration of an unbalanced species tree and a gene family tree undergoing consecutive **duplication events** noted in red boxes. Each species tree branch has a substitution rate μ_i ; branches of the gene tree inherit rates of species tree branches they pass through (e.g., the a_2 branch will inherit μ_a and μ_2). We show all quartet trees involving only orthologous genes alongside their respective weights, followed by examples of quartet trees that include paralogous genes (not counted). The weight of a quartet is 2^{-d} , where d is the number of **duplication events** the quartet passes through. The sum of all quartets, including a branch (e.g., a_2), is at most 1 but can be lower (e.g., $1/2$ for b_3). C) Examples of counters for computing the weighted count of gene tree quartets using dynamic programming. For the species tree quadripartition $AB|CD$, we compute the mean length of the branches matching (or not matching) it in a bottom-up traversal. At each node w , several counters are updated; the two simplest counters are shown: $w_a^.$ (weighted number of leaves corresponding to $a \in A$ below w) and $w_{ab}^.$ (weighted number of pairs $(a, b) \in A \times B$ with a speciation node as **LCA** at or below w). Note the weights for duplication.

is not modeled by concatenation or methods that do not account for coalescence and can impact downstream analyses such as dating, a topic that [324] explore.

We can extend this idea to quartet trees using more advanced calculations and distinguishing gene trees that match or do not match the species tree. These equations would be of the form $E(L) = f(t, \mu)$ where L is a random variable representing the length of a terminal (L_T) or [internal](#) (L_I) branch of a quartet gene tree matching the [topology](#) of the species tree or conflicting with it (L'_I and L'_T). We derived those equations, reproduced in Tables 5.1, 5.2 and 5.3. To estimate species tree [SU](#) lengths given a set of gene trees, we first compute the average branch lengths for gene trees that match or conflict with the [topology](#) of the species tree (\bar{L}_I , \bar{L}_T , \bar{L}'_I , \bar{L}'_T); see Figure 6.1A. Equating the theoretical expected values (with unknown parameters) with observed values, we get a set of equations that can be analytically solved. CASTLES employs several simplifying assumptions to reduce the number of parameters further and simplify the equations (Table 5.2). For example, to estimate an [internal](#) branch with length $t_1 = T_1\mu_1$ (Figure 6.1B), we obtain

$$\bar{\delta} := \frac{\bar{L}_I}{\bar{L}'_I} - 1 = \frac{3(T_1 + e^{-T_1} - 1)}{3 - 2e^{-T_1}} \quad (6.1)$$

which we solve for T_1 to obtain

$$\hat{T}_1 = g(\bar{\delta}) := \bar{\delta} + W\left(-\frac{1}{3}e^{-\bar{\delta}-1}(2\bar{\delta} + 3)\right) + 1 \quad (6.2)$$

where $W(\cdot)$ is the Lambert W function. We then estimate substitution rates μ_1 using a second equation with further simplifying assumptions to obtain $\hat{\mu}_1 = \bar{L}'_I$ and thus:

$$\hat{t}_1 = g(\bar{\delta})\bar{L}'_I \quad (6.3)$$

CASTLES extends these calculations to $n > 4$ species by computing averages over all quartets around each species tree branch, a task that can be done in $O(n^2)$ using a dynamic programming algorithm.

Handling Duplication and Loss. Gene duplications create quartet trees with paralogous gene copies. CASTLES-Pro addresses this issue by striving to exclusively use quartets devoid of paralogous genes. Following ASTRAL-Pro2, we first tag each [internal](#) node of each input gene tree as either a duplication or a speciation event, and we assume these [tags](#) are accurate (in practice, they are obtained using parsimony and may have errors). CASTLES-Pro operates similarly to CASTLES, with two major changes. The main change

is that a quartet contributes to empirical mean branch length (\bar{L}_I , \bar{L}_T , \bar{L}'_I , \bar{L}'_T) only if the least common ancestor (LCA)s of all pairs of its leaves are speciation nodes (these are called orthologous quartets). If all the tags are correct, orthologous quartets will follow the MSC expectations [312] and, thus, CASTLES assumptions.

The second change relates to the potentially uneven rates of duplication across gene families, which can lead to certain branches being overrepresented in the final means. Figure 6.1B demonstrates an example; four out of five orthologous quartets share the a_2 branch indicated in blue; however, the [duplication events](#) are on a separate branch and result in using a_2 four times in computing mean branch lengths of a (e.g., \bar{L}_I) for no apparent reason. Counting all orthologous quartets without weights can increase the impact of individual branches and, thus, the variance in estimated means. We use a weighting scheme to mitigate the impact of this overrepresentation. The weights are simply set to 2^{-d} where d is the number of duplication nodes falling on the subtree spanned by the quartet tree. With this scheme, it is easy to show that the total weight of quartets that include a branch will not exceed 1 in any gene family tree.

Beyond eliminating obvious [paralogs](#) and weighting ortholog quartets, the main challenge is computing mean branch lengths efficiently instead of the trivial $O(n^4)$ algorithm that lists all quartets. We designed a dynamic programming algorithm that achieves this goal in $O(n^2)$ time (Algorithm 6.1). Compared to CASTLES, we needed to refine the set of our counters updated in the dynamic programming (demonstrated in Figures 6.2, 6.9.) Crucially, at duplication nodes, the recursive formulas change to ignore non-orthologous quartets and implement the weights (see Figure 6.1B). After calculating the means, CASTLES-Pro assigns lengths to each branch of the species tree in an $O(n)$ pre-order traversal of the tree, and therefore the total runtime of the CASTLES-Pro algorithm is $O(n^2)$.

Recursive Algorithm. We will first provide an $O(n^2)$ algorithm for computing branch lengths for all [internal](#) and [terminal](#) branches. Notice, the complexity of this algorithm can be improved to $O(nH \log n)$, where H denotes the average height of gene family trees, using the dynamic programming algorithm in CASTLES. For conciseness, we use A, B, C, D to denote sets of [taxa](#) and use a, b, c, d to denote individual taxa. Let \mathcal{G} be the set of gene trees.

To compute all branch lengths, it is sufficient to compute the following counters for a set of ordered leafset quadripartitions, in which each leafset quadripartition (A, B, C, D) – up to permutations – corresponds to an [internal](#) branch:

- $n(A, B; C, D)$: the number of quartet and gene tree combinations $(a, b, c, d, G) \in A \times B \times C \times D \times \mathcal{G}$ such that $G \restriction \{a, b, c, d\}$ has [topology](#) $ab|cd$.

- $x(A, B; C, D)$: the total **internal** branch lengths of quartet trees in the form of $G \upharpoonright \{a, b, c, d\}$ with **topology** $ab|cd$, where $(a, b, c, d, G) \in A \times B \times C \times D \times \mathcal{G}$.
- $a(A; B; C, D)$: the total length of the **terminal** branches leading to A in quartet trees in the form of $G \upharpoonright \{a, b, c, d\}$ with **topology** $ab|cd$, where $(a, b, c, d, G) \in A \times B \times C \times D \times \mathcal{G}$.

All three counters for each quadripartition (A, B, C, D) can be computed in a single post-order traversal of the gene tree nodes in $O(n)$ using Algorithm 6.1. Therefore, computing counters for all $O(n)$ quadripartitions has time complexity $O(n^2)$. Notice that Algorithm 6.1 assumes that all input gene trees are **fully resolved**, consistent with the presumption of ASTRAL-Pro.

Better approximations, assumptions, and handling of short branches. Even for single-copy gene trees, CASTLES-Pro improves on CASTLES in three ways and, as we will see, dominates it in terms of accuracy (thus, CASTLES-Pro replaces CASTLES.) One change is related to a simplifying assumption: CASTLES-Pro uses a slightly different approach for handling dependencies on the parent branch when calculating the length of a **terminal** branch of a cherry.

CASTLES employed a weak approximation that we eliminate in CASTLES-Pro. Instead of directly using the Lambert function W in Eq. (6.2), CASTLES used a Taylor approximation to get $g(\bar{\delta}) \approx \frac{1}{2}\bar{\delta} + \frac{1}{6}\sqrt{3\bar{\delta}(3\bar{\delta} + 4)}$. However, this approximation underestimates the true value as we move away from its focal point (0), causing a systematic underestimation bias (Figure 6.3). Oddly, our simulation studies show that the Taylor approximation works better in practice on estimated gene trees despite this bias. Attributing this odd observation to difficulties with numerical precision, in CASTLES, we opted to use the Taylor expansion. However, we have now discovered a different explanation for this pattern.

Our simulations show that when using true gene trees, the Lambert W function is superior, whereas Taylor gradually becomes better as **gene tree estimation error (GTEE)** increases (Figure 6.10). Taylor's better performance is due to the interplay between two opposing sources of bias. As **GTEE** increases, we tend to overestimate $\bar{\delta}$, a pattern that can be explained. With a low phylogenetic signal and hence high **GTEE**, many of the short internal branches, which are more prevalent among gene trees not matching the species tree, become zero-event (i.e., record no substitutions). The inferred length of these supershort branches is often driven by a pseudocount used in the maximum likelihood tools (e.g., 1e-6), which is often an underestimation of the true value. Since $\bar{L}' < \bar{L}$ to begin with, these underestimations happen more for \bar{L}' than \bar{L} , and thus, $\bar{\delta} = \frac{\bar{L}_I - \bar{L}'_I}{\bar{L}'_I}$ tends to get overestimated, which in turn offsets the underestimation bias of the Taylor approximation. This lucky

Algorithm 6.1 Recursive algorithm. The input is a set of gene trees \mathcal{G} and an ordered quadripartition of the leafset (A, B, C, D) , and the output are $n(A, B; C, D)$, $x(A, B; C, D)$, and $a(A; B; C, D)$. For each node u we keep a list of counters u^* described in Figs. 6.2 and 6.9.

```

1: procedure UPDATELEAFCOUNTERS( $u, A, B, C, D$ )
2:   Set all counters  $u^*$  to 0
3:   if  $u$  corresponds to a taxon in  $A$  then
4:      $u_a^* \leftarrow 1$ 
5:      $u_a^a \leftarrow$  the parental branch length of  $u$ 
6:   else if  $u$  corresponds to a taxon in  $B$  then
7:      $u_b^* \leftarrow 1$ 
8:      $u_b^b \leftarrow$  the parental branch length of  $u$ 
9:   else if  $u$  corresponds to a taxon in  $C$  then
10:     $u_c^* \leftarrow 1$ 
11:     $u_c^c \leftarrow$  the parental branch length of  $u$ 
12:   else if  $u$  corresponds to a taxon in  $D$  then
13:      $u_d^* \leftarrow 1$ 
14:      $u_d^d \leftarrow$  the parental branch length of  $u$ 
15:   end if
16: end procedure
17: procedure RECURSIVEALGORITHM( $\mathcal{G}, A, B, C, D$ )
18:   Set  $n(A, B; C, D)$ ,  $x(A, B; C, D)$ ,  $a(A; B; C, D)$ ,  $b(A; B; C, D)$ ,  $c(A; B; C, D)$ ,  $d(A; B; C, D)$  to 0
19:   for each gene  $G \in \mathcal{G}$  do
20:     for  $u \in$  post order traverse of nodes of  $G$  do
21:       if  $u$  is a leaf node then
22:         UPDATELEAFCOUNTERS( $u, A, B, C, D$ )
23:       else
24:         Update counter  $u_a^*$  using the recursive formula in Fig. 6.2            $\triangleright$  ditto for  $u_b^*, u_c^*, u_d^*$ 
25:         Update  $u_{ab}^*$  using Fig. 6.2                                          $\triangleright$  ditto for  $u_{cd}^*$ 
26:         Update  $u_{ab|c}^*$  using Fig. 6.2                                      $\triangleright$  ditto for  $u_{ab|d}^*, u_{cd|a}^*, u_{cd|b}^*$ 
27:         Update  $u_{ab|cd}^*$  using Fig. 6.2
28:         Update counter  $u_a^a$  using the recursive formula in Fig. 6.9            $\triangleright$  ditto for  $u_b^b, u_c^c, u_d^d$ 
29:         Update counter  $u_{ab}^a$  using Fig. 6.9                                $\triangleright$  ditto for  $u_{ab}^b, u_{cd}^c, u_{cd}^d$ 
30:         Update counter  $u_{ab}^x$  using Fig. 6.9                                      $\triangleright$  ditto for  $u_{cd}^x$ 
31:         Update counter  $u_{ab|c}^a$  using Fig. 6.9            $\triangleright$  ditto for  $u_{ab|d}^b, u_{cd|a}^c, u_{cd|b}^d$ 
32:         Update counter  $u_{ab|c}^x$  using Fig. 6.9            $\triangleright$  ditto for  $u_{ab|d}^x, u_{cd|a}^x, u_{cd|b}^x$ 
33:         Update counter  $u_{ab|c}^d$  using Fig. 6.9            $\triangleright$  ditto for  $u_{ab|d}^c, u_{cd|a}^b, u_{cd|b}^a$ 
34:         Update counter  $u_{ab|cd}^a$  using Fig. 6.9            $\triangleright$  ditto for  $u_{ab|cd}^b, u_{ab|cd}^c, u_{ab|cd}^d$ 
35:         Update counter  $u_{ab|cd}^x$  using Fig. 6.9
36:       end if
37:     end for
38:      $u \leftarrow$  the root of  $G$ 
39:      $n(A, B; C, D) \leftarrow n(A, B; C, D) + u_{ab|cd}^*$ 
40:      $x(A, B; C, D) \leftarrow x(A, B; C, D) + u_{ab|cd}^x$ 
41:      $a(A; B; C, D) \leftarrow a(A; B; C, D) + u_{ab|cd}^a$             $\triangleright$  ditto for  $b(A; B; C, D), c(A; B; C, D), d(A; B; C, D)$ 
42:   end for
43: end procedure

```

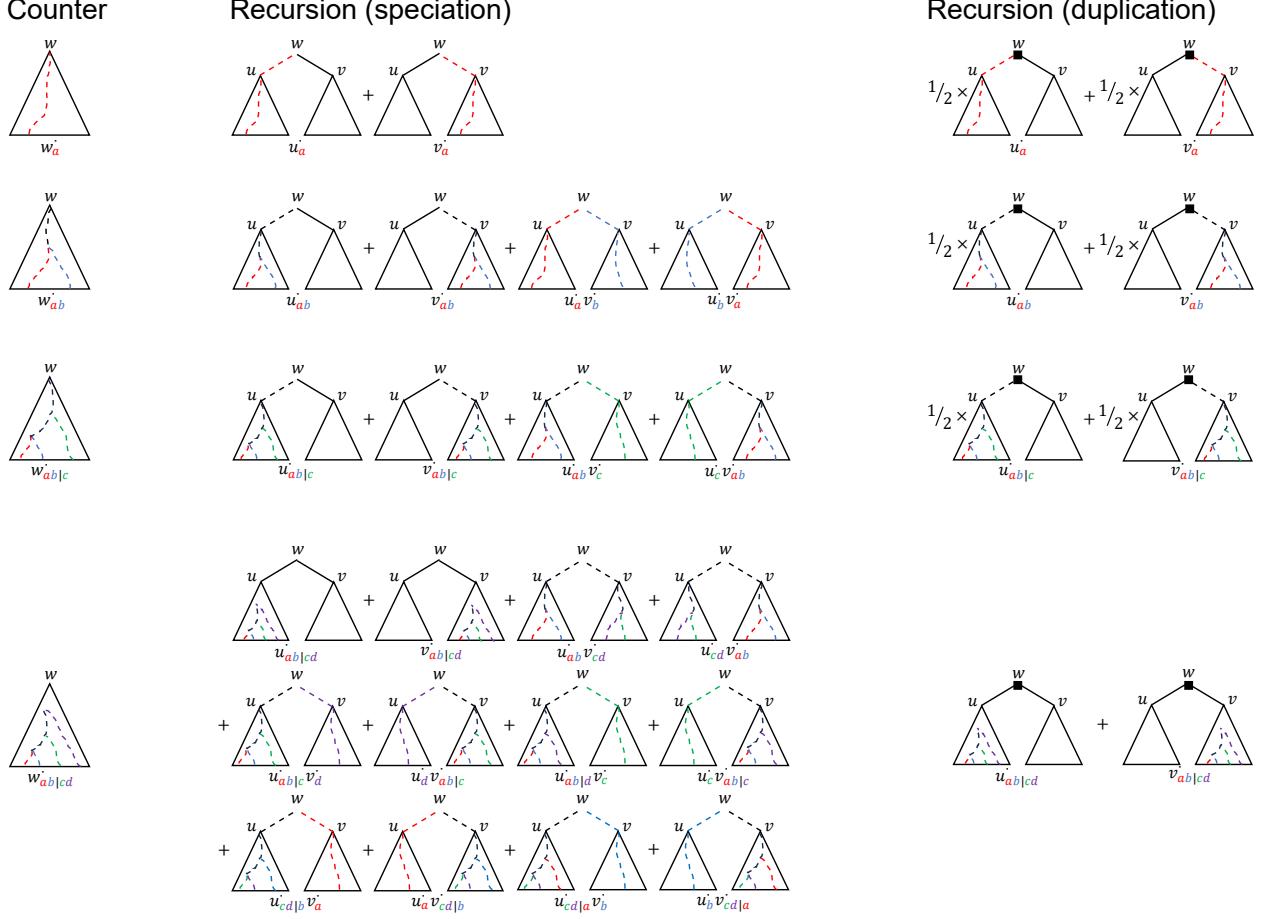


Figure 6.2: Additional counters for computing the weighted count of gene tree quartets aligning with the species quadripartition $ab|cd$. To the left presents a list of counters for each [internal](#) node w ; in the middle illustrates how to recursively compute these counters from counters of the two children of w when w corresponds to a speciation event; to the right illustrates how to compute these counters when w corresponds to a [duplication event](#). For example, $w_a = u_a + v_a$ if w corresponds to a speciation event, and $w_a = u_a/2 + v_a/2$ if w corresponds to a speciation event (u and v denote the children of w). The total weighted count of gene tree quartets aligning with the species quadripartition $ab|cd$ is calculated as $\sum_{w \in R} w_{ab|cd}$, with R representing the set of root nodes across all gene trees.

cancelling is not enjoyed if we use the exact Lambert W function. As the signal increases, or with true gene trees, there is no overestimation to offset Taylor's bias, leading to the exact Lambert equation working better.

In CASTLES-Pro, we directly address the overestimation of $\bar{\delta}$ under high [GTEE](#) conditions and switch to using the exact Lambert W function in Equation 6.2. If the length of an alignment is s , a branch of length $1/s$ would expect to see one substitution. Thus, branches substantially below $1/s$ are often zero-event and underestimated by a pseudocount. To address

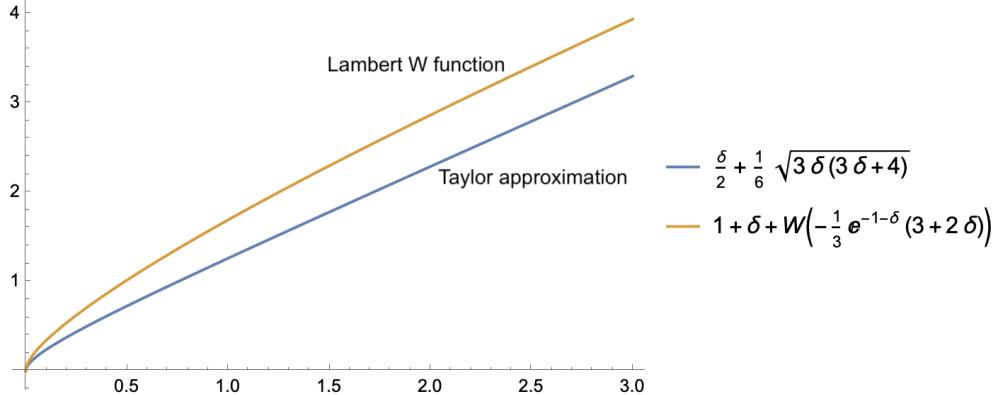


Figure 6.3: Lambert W function vs its Taylor approximation for caluclating the length of the internal branch $g(\bar{\delta})$ in CASTLES-Pro for $\delta \in (0, 3)$.

this, we simply add a pseudocount of $1/s$ to both \bar{L}_I and \bar{L}'_I , obtaining an adjusted value for δ given by $\delta_C = \frac{\bar{L}_I - \bar{L}'_I}{\bar{L}'_I + 1/s}$. This is, in principle, similar to adding a pseudocount to binomial parameter estimation, which is equivalent to a posterior estimate under a Dirichlet prior. As sequence length increases and [GTEE](#) decreases, so does the pseudocount (in the limit, $s = \infty$ for true gene trees, giving a zero pseudocount). The value of s can be adjusted by the user (default: 1000).

Another difficulty arises when $\bar{\delta}$ is negative, indicating that the average length in non-matching gene trees is larger than that of matching gene trees. This is unexpected under [MSC](#) and will not happen with infinitely many error-free gene trees; in practice, however, it can happen for many reasons. CASTLES simply resorted to replacing $\bar{\delta} < 0$ with a fixed pseudocount of $\delta_p := 10^{-3}$. For some causes of negative $\bar{\delta}$, this is not a good approach. When a branch is very long with a low level of [ILS](#), there are very few non-matching gene trees. Furthermore, these non-matching gene trees can differ from the [species tree](#) due to reasons other than [ILS](#), such as paralogy, horizontal transfer, incorrect homology, etc. Thus, the (few) non-matching gene trees can have average lengths that are larger than the average length of matching gene trees, leading to a negative $\bar{\delta}$. In such cases, simply using the mean of matching gene trees (\bar{L}_I) is a better approximation. In contrast, the CASTLES approach of using a small pseudocount δ_p in the original equation $g(\delta_p)\bar{L}'_I$ makes sense for very short branches. CASTLES-Pro handles negative $\bar{\delta}$ using a formula that takes the level of [ILS](#) into account and transitions between these two appraoches. For $\bar{\delta} < 0$ we use:

$$\hat{t}_1 = \frac{\omega_d}{\omega_d + \frac{1}{\omega_d}} \bar{L}_I + \frac{\frac{1}{\omega_d}}{\omega_d + \frac{1}{\omega_d}} g(\delta_p) \bar{L}'_I \quad (6.4)$$

where $\omega_d = \log_{10}(k)d$ is the weight of the two formulas; d is the quartet-based **CU** length of the branch [see 321] and k is the number of gene trees. As gene tree discordance decreases, d and ω_d increase and in the limit, $\lim_{d \rightarrow \infty} \hat{t}_1 = \bar{L}_I$ in Eq. (6.4); this is justified because for long branches, the **deep coalescence** has a *relatively* small impact compared to the full length. For short branches, d decreases, and in the limit $\lim_{d \rightarrow 0} \hat{t}_1 = g(\delta_p) \bar{L}'_I$, we resort to the original formula (6.3) used with the pseudocount. Thus, we transition from relying on average matching gene tree length (\bar{L}_I) for branches with little discordance to our original estimate for high discordance; the rate of transitioning between the two approaches is governed by the number of genes, with more genes leading to faster adoption of \bar{L}_I ; this is because the discordance-based estimates of **CU** length (d) are more accurate with more genes.

Tables 5.1-5.3 summarize the exact and simplified formulas for expected branch lengths in matching and non-matching gene trees under the MSC, and final formulas for estimating branch lengths of an unbalanced or balanced quartet **species tree** that were used in CASTLES [74]; in both figures, parameters are named according to Fig. 6.4. CASTLES-Pro uses much of the same formulas, but computes them in a different way that leads to improvements in accuracy. The changes for the internal branches (avoiding the Taylor approximation and ILS-aware weighting) are described in the main text, and here we describe the changes in calculating the **terminal** branches.

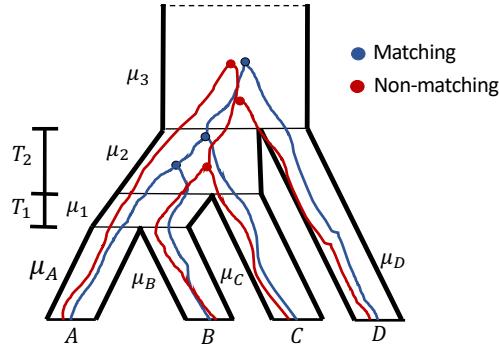


Figure 6.4: An unbalanced model species tree with four **taxa** with a matching and non-matching gene tree shown inside it. T_1 and T_2 denote the **CU** lengths, μ_i s are the mutation rates, and the internal branch has an **SU** length of $t_1 = T_1\mu_1$ (the **SU** length for other branches are defined similarly). **taxa** A and B are referred to as **cherry branches**.

CASTLES uses Eq. (6.5) to calculate the length of the cherry branch leading to **taxa** A (and similarly for B) in an unbalanced quartet species tree:

$$\hat{t}_A = \bar{L}'_A + \frac{\mu_1(e^{-T_1} - 1 + T_1) + \bar{\Delta}_A(1 - 2/3e^{-T_1})}{1 - 4/5e^{-T_1}} - T_1\mu_1 . \quad (6.5)$$

This equation depends on the mutation rate of the internal branch μ_1 and the **CU** branch length T_1 . CASTLES uses the simplified formula for Δ_I in Table 5.2 to calculate μ_1 and substitutes it in Eq. (6.5):

$$\hat{\mu}_1 = \frac{\hat{\Delta}_I(3 - 2e^{-T_1})}{3(e^{-T_1} - 1 + T_1)}. \quad (6.6)$$

Here, \hat{T}_1 , the estimate of the **CU** length of the internal branch, is calculated using the approach from [321]. However, as $T_1 \rightarrow 0$, the denominator of Eq. (6.6) becomes very small, and therefore the formula becomes unstable as $\lim_{T_1 \rightarrow 0} e^{-T_1} = 1 - T_1$. In addition, the value of T_1 calculated using [321]’s algorithm degrades in accuracy as gene tree estimation error increases. While the formulas for the internal branches in CASTLES were calculated so as to not have a dependency on these **CU** lengths, the **terminal** branches still have this dependency.

To improve the estimation of μ_1 and reduce the dependency on the **CU** branch lengths estimated using [321]’s approach in Eq. (6.5), CASTLES-Pro instead first calculates T_1 as a function of δ (see the main text) and μ_1 as \bar{L}'_I . It then directly uses these values to compute the **terminal** branch equations.

6.3 EXPERIMENTAL STUDY

We compare CASTLES-Pro to other branch length estimation methods using three sets of simulated datasets and nine published biological datasets with gene tree discordance due to **ILS**, **GDL**, and **HGT** (Table 6.1).

6.3.1 Simulations.

We studied three sets of simulated datasets with gene tree discordance due to **ILS**, **ILS+GDL**, and **ILS+HGT** (Table 6.1). All simulated datasets are generated using SimPhy [272]; however, we modified SimPhy to output model species trees with **SU** branch lengths using mutation rates already present in SimPhy simulations. Gene sequences were simulated under GTR+ Γ model, and gene trees were estimated from these alignments using FastTree-2 [144]. The **ILS**-only dataset was reused from [74] and has gene alignments of length 200bp – 1600bp to control gene tree estimation error (GTEE). The level of **ILS** is heterogeneous across replicates, with mean equal to 46% according to average [83] (RF) distance between model species trees and true gene trees (AD for short). The **GDL+ILS** dataset was reused from [90, 274] and has two levels of **ILS** (low and high), six **duplication rates** (10^{-13} – 10^{-9}), three sequence lengths, various numbers of species, and genes (Table

Table 6.1: Statistics of the simulated and biological datasets used in this study. n denotes the number of species and k denotes the number of single-copy or multi-copy genes.

| Simulated | n | k | $\ S - G\ ^{\S}$ | $\ G - \hat{G}\ $ | Gene len. (bp) | See |
|-------------|---------|--------------------------------------|------------------|------------------------------|------------------------------|-----------|
| ILS-only | 101 | 1000 | 30%–58% | 23, 31, 42, 55% | 1600, 800, 400, 200 | |
| ILS+GDL | 21–1001 | 50–10000 | 15%–78%* | 18% – 56% | 50, 100, 500 | Table 6.5 |
| ILS+HGT | 51 | 1000 | 30%–68% | 28% | 1000 | Table 6.4 |
| Biological | n | k | Type | $\ \hat{S} - \hat{G}\ ^{\P}$ | Total len. (bp) [#] | See |
| Birds | 363 | 63,430 | ILS | 31.1% | 63,430,000 | Table 6.6 |
| Bees | 32 | 853 | ILS | 39.7% | 576,041 | Table 6.6 |
| Mammals | 37 | 424 | ILS | 26.6% | 1,385,220 | Table 6.6 |
| Fungi | 16 | 706 [†] , 7180 [‡] | ILS+GDL | NA | 286,114 | Table 6.6 |
| Plants(1kp) | 80 | 424 [†] , 9610 [‡] | ILS+GDL | 48.0% | 290,718 | Table 6.6 |
| Eudicots | 40 | 345 [†] , 2573 [‡] | ILS+GDL | 50.5% | 262,343 | Table 6.6 |
| AB core | 72 | 49 | HGT | 69.3% | 10,522 | Table 6.6 |
| AB non-rib. | 108 | 38 | HGT | 61.8% | 6,534 | Table 6.6 |
| AB WoL | 10,575 | 381 | HGT | 53.9% | 1,162,421,084 | Table 6.6 |

\S Average discordance between two types of trees is denoted by $\|a - b\|$ and is measured using the average normalized RF distance. S and \hat{S} denote model and estimated species trees; G and \hat{G} denote true and estimated gene trees.

* RF between true gene trees and the true locus tree, which is only due to ILS.

\dagger single-copy genes, used in concatenation.

\ddagger multi-copy genes used for ASTRAL-Pro and CASTLES-Pro.

\P RF distance between single-copy gene trees and species trees estimated by ASTRAL or ASTRAL-Pro (NA when not available).

The total length of the sequence alignments for the single-copy loci.

6.1). The loss rate relative to the duplication rate is set to 1, 0.5, or 0. For ILS+HGT, we recreated a dataset by [325] with 30% AD due to ILS and six levels of HGT rates, leading to up to 68% AD (Table 6.4). The average number of HGT events per gene for the six model conditions starts from 0 to 0.08, 0.2, 0.8, 8 and 20, corresponding to HGT rates $10^{-9} \times (0, 2, 5, 20, 200, \text{and } 500)$.

In all simulations, we estimate branch lengths on the fixed true species tree topology. We measure branch length estimation error using three metrics: mean absolute error ($|\bar{t} - t|$), mean logarithmic error ($|\log \bar{t}/t|$), and the bias of the estimated length $\bar{t} - t$ (t and \bar{t} are true and estimated branch lengths, resp.) averaged across all species tree branches. The log error emphasizes short branches, while the absolute error and bias emphasize long branches. We remove the outgroup before measuring the branch length estimation error. For all methods, we replace negative and zero branch lengths with a small pseudo-count (10^{-6}) before calculating error metrics.

ILS-only dataset. We reused a published dataset from [74] simulated using Simphy. This dataset has 50 replicates, each with 100 ingroups and one **outgroup** species and 1000 gene trees. In addition to true gene trees, gene alignments of length 1600bp, 800bp, 400bp, and 200bp were simulated under GTR+ Γ , and four sets of gene trees were estimated from these alignments using FastTree-2 [144], producing gene trees with 23%, 31%, 42%, and 55% **GTEE**. Measuring **ILS** using the average [83] (RF) distance (AD for short) between the model **species tree** and true gene trees, **ILS** is heterogeneous across replicates and ranges from 30% to 58%, with a mean of 46% AD.

GDL+ILS. We updated the Simphy-generated GDL+ILS datasets of [90, 326] to have species trees with **SU** branch lengths. Here, **locus** trees evolve inside the **species tree** with **GDL** events only; the final true gene trees evolve on the **locus** tree under **MSC**. Therefore, the topological differences between the final true gene trees and the **locus** tree are only due to ILS; we use the normalized **RF** distance between the **locus** tree and the true gene trees to measure ILS. This dataset has model conditions (10 replicates each) characterized by two levels of **ILS** (low and high **ILS** with 20% and 65% average **ILS** level, respectively), six **duplication rates**, ranging from 10^{-13} to 10^{-9} , three sequence lengths (50bp, 100bp, 500bp), four different numbers of species (21, 51, 101, or 1001), and five numbers of genes (50, 100, 500, 1000, 10,000). The **loss rate** varies based on the **duplication rate**, with three different ratios: 1 (equal loss), 0.5 or 0 (no loss). In the default model condition, the **loss rate** is equal to the **duplication rate**. Gene trees were estimated using FastTree-2. The number of replicates in all model conditions is 10. Table 6.5 summarizes further statistics about the model conditions of this dataset.

HGT+ILS. We recreated a 50-replicate 51-taxon (50 ingroup and one outgroup) dataset with both **HGT** and **ILS** based on the parameters used by [325]. The **ILS** level is fixed at 30% **AD** across the model conditions. The six model conditions differ in **HGT** rates, leading to total discordance that varies between 30% to 68%. The average number of **HGT** events per gene for the six model conditions starts from 0 and increases to 0.08, 0.2, 0.8, 8 and 20 that correspond to **HGT** rates of 0, 2×10^{-9} , 5×10^{-9} , 2×10^{-8} , 2×10^{-7} and 5×10^{-7} . In addition to true gene trees, we simulated 1000bp gene sequence alignments using INDELible [327] under the **GTR** + Γ model and then used FastTree-2 to estimate gene trees under the **GTR** model. **GTEE** is on average 28% and about the same in all model conditions. The number of genes is 1000 and the number of replicates is 50. Table 6.4 summarizes the empirical statistics of this dataset.

We compare CASTLES-Pro to CASTLES, ERaBLE, FastME [297] used on matrices of

average [patristic distances](#) (referred to as FASTME(AVG)), and concatenation using maximum likelihood with RAxML [25]. All these methods are only designed to work with single-copy genes; hence, for datasets with [GDL](#), we create a two-step pipeline where we first use the method DISCO [90] to decompose gene family trees into single-copy gene trees, which we then pass to branch length estimation methods. These two-step methods are referred to as CASTLES-DISCO, ERaBLE-DISCO, and FastME(AVG)-DISCO. To perform concatenation with multi-copy input, we use the CA-DISCO technique of [90]. Sequences for each gene family are broken up into single-copy loci, and these loci are concatenated into a super-alignment. We use RAxML on this alignment to optimize branch lengths on the fixed true [species tree](#) topology. Note that DISCO can produce trees with high levels of missing data, and ERaBLE and FastME(AVG) can fail on inputs with missing data. To enable these methods to run on DISCO output, we imputed the missing values in the distance matrix of each gene tree by the average [patristic distances](#) among gene trees that do include the pair of [taxa](#) associated with the missing value. Finally, while the [species tree](#) estimation method SpeciesRax [199] can produce branch lengths in substitution units, we did not include it in this study as it cannot estimate branch lengths on a fixed input topology.

6.3.2 Biological datasets.

We reexamined nine biological datasets with different sources of gene tree discordance (Table 6.1, Table 6.6). As examples of datasets with [ILS](#), we analyzed the birds dataset by [328], bees by [102], and mammals by [250]. For [GDL](#), we analyzed two plant datasets [109, 319] and a fungal dataset [329] that included multi-copy gene family trees. Finally, following [76], we analyzed three bacterial datasets [60, 330, 331] as examples of cases with high rates of [HGT](#), with a focus on the length of the branch that separates archaea and bacteria at the root of the tree of life ([AB](#) branch).

We compare the branch lengths produced by CASTLES-Pro on ASTRAL or ASTRAL-Pro topologies to concatenation branch lengths drawn on either ASTRAL or concatenation topologies. For single-copy datasets, we used the ASTRAL [topology](#) for both concatenation and CASTLES-Pro branch lengths. For the three datasets with [GDL](#), since directly using concatenation on multi-copy gene sequences was not possible, we compared the concatenation [topology](#) on single-copy genes from original studies to ASTRAL-Pro [topology](#) furnished with CASTLES-Pro branch lengths run on multi-copy genes (the two trees differed by 8–9% RF). In these cases, we focus on branches that are shared between the two trees. For birds, bees, and mammals, ASTRAL trees were already available, while for fungi, we inferred a tree using ASTRAL-Pro2 [332] using all multi-copy gene trees, and for small bacterial datasets,

we inferred a tree using ASTRAL-III [248]. For 1KP, single-copy concatenation and multi-copy ASTRAL-Pro trees have 80 [taxa](#) in common, which we use. For some datasets, original studies ran concatenation on a subset of [sites](#) from the sequence alignments: For the fungi dataset, 30,000 [sites](#) were sampled from 706 [orthologs](#), and for the [WoL](#) bacterial dataset, 100 [sites](#) were randomly selected from [sites](#) with less than 50% gaps for each of the 381 marker genes. The details of the analysis for each biological dataset are provided below.

Birds. We studied the birds dataset of [328] including 363 species and 63,430 genes that was used to resolve family-level relationships among neoavian species and is expected to have high levels of [ILS](#) due to a [rapid radiation](#). The original study had inferred the tree [topology](#) using ASTRAL and then estimated branch lengths on that [topology](#) using the concatenation of all 63K genes. We infer branch lengths on the same ASTRAL [topology](#) using CASTLES-Pro.

Bees. We reanalyzed the bees dataset of [102] containing 32 species (30 ingroups and two outgroups) from the bee subfamily Nomiinae and 853 gene trees (estimated using RAxML). We used the ASTRAL [topology](#) from the original study, and used [partitioned](#) concatenation (with RAxML) and CASTLES-Pro to draw branch lengths on this topology.

Mammals. We studied the mammalian biological dataset from [250], including 37 species (36 ingroup and 1 outgroup) and 447 gene trees, which was reduced to 424 trees after removing gene trees with mis-matching names [179]. We estimated an ASTRAL [topology](#) and estimated branch lengths using concatenation and CASTLES-Pro on that topology.

Fungi. We examined the fungal dataset of [329], including 16 yeast species and 7,180 multi-copy gene family trees. The original study had used MrBayes [333] on a concatenated alignment created by sampling 30,000 [sites](#) from 706 individual gene family orthologous peptide sequences. We used ASTRAL-Pro2 [332] to estimate a [species tree](#) using all 7,180 gene family trees, and used CASTLES-Pro to estimate branch lengths on that tree. The two trees are different in one branch, with an [RF](#) distance of 7.6%.

Plants (1KP). We analyzed the plants dataset of [109], that included 103 species and 424 single-copy gene trees, as well as 9,610 multi-copy gene family trees for 83 of the species, that was left unused in the original study due to lack of proper method for estimating the [species tree](#) from multi-copy input. The gene trees were inferred using RAxML for the first two codon positions (C12) in the transcriptome. We compare the branch lengths of the

concatenation tree inferred from the 424 single-copy gene alignments with a tree inferred using ASTRAL-Pro2 from all 9,610 multi-copy gene trees furnished with CASTLES-Pro branch lengths. The two trees have 80 [taxa](#) in common and are different in 7 branches on the shared set of taxa, resulting in an [RF](#) distance of 9.1%.

Eudicots. We studied the 40-taxon angiosperm dataset of [319] focused on the Eudicots [lineage](#). This study had used three sets of genes to perform phylogenomic analysis using concatenation and coalescent-based summary methods: 345 filereted single-copy Angiosperms353 loci [334], 1248 single-copy [BUSCO](#) [335] genes, and 2,573 multi-copy orthogroups. The authors had performed concatenation analysis on the two sets of single-copy genes (Angiosperms353 loci and BUSCOs) using RAxML and coalescent analysis on all three sets of genes using ASTRAL and ASTRAL-Pro for single and multi-copy input respectively. We compared the two concatenation trees from the original study with a tree we inferred using ASTRAL-Pro2 from the 2,573 orthogroups that was furnished with CASTLES-Pro branch lengths. The two concatenation trees had the same [topology](#) that was different from the ASTRAL-Pro2 tree in three branches, with an [RF](#) distance of 8.1%.

Microbial datasets. We analyzed three microbial datasets including thousands of species of bacteria and archaea and different sets of genes to study a debate about the length of the branch separating domains archaea and bacteria ([AB](#) branch). While the long-standing hypothesis was that these two domains are separated by a long branch [336, 337, 338], a recent study [60] had estimated a far shorter length for the [AB](#) branch than what was previously expected using a concatenation analysis. [76] had further studied this and other bacterial datasets, and suggested that concatenation can severaly underestimate branch lengths on datasets with high levels of [HGT](#), resulting in short estimates of [60].

Here we examine two bacterial datasets analyzed by [76] and the [Web of Life \(WoL\)](#) dataset from [60] with CASTLES-Pro to further study this debate. The two bacterial datasets include a 72-taxon dataset with 49 core genes originally from [330] that includes ribosomal proteins and other conserved elements and a 108-taxon dataset with 38 genes from [331] that only includes non-ribosomal proteins. The [WoL](#) dataset included 10,575 species (9,906 bacteria and 669 archaea) and 381 marker genes including ribosomal and non-ribosomal proteins. For the two smaller bacterial datasets, we estimated a [species tree](#) using ASTRAL on the two gene sets and used concatenation (with RAxML) and CASTLES-Pro to draw branch lengths on these topologies. On the [WoL](#) dataset, we used the ASTRAL tree from the original study that was furnished with branch lengths estimated using RAxML from a concatenation including 100 [sites](#) randomly selected from [sites](#) with less than 50% gaps for

each marker gene. We estimated branch lengths on the same topology using CASTLES-Pro.

6.4 RESULTS

6.4.1 ILS-only simulations

CASTLES-Pro has the best accuracy across all GTEE levels of this dataset, followed by CASTLES (Figures 6.5,6.11). Distance-based methods come next, with TCMM outperforming ERaBLE and FastME(AVG), and concatenation is the least accurate overall. Relative accuracy of methods is mostly consistent across different alignment lengths (Figure 6.5A) and levels of ILS (Figure 6.5B). As alignment length increases (and GTEE decreases), concatenation remains stable while CASTLES-Pro and distance-based methods become successively better. Improvements of CASTLES-Pro are mostly due to better terminal branches, which are substantially more accurate than the other methods in all conditions (Figure 6.5). On internal branches, the relative accuracy depends on the condition; CASTLES-Pro is better for true gene trees and slightly worse than the distance-based methods and CASTLES in the highest GTEE level. Finally, note that pairing TCMM with CASTLES-Pro, as described by [75], substantially reduces the error of TCMM, but in these ILS-only conditions, the combination is not as good as CASTLES-Pro alone (Figure 6.11).

In addition to better accuracy, CASTLES-Pro has the lowest overall bias (Figure ??). In particular, concatenation has a substantial overestimation bias for terminal branches, but it also overestimates internal branches to a lesser degree. CASTLES shifts from a small underestimation bias for true gene trees to a minor overestimation bias for gene trees with high GTEE ; this is due to the effects of the imprecise Lambert approximation used in CASTLES, which is fixed in CASTLES-Pro. Distance-based methods also exhibit an overestimation bias, though it is smaller than that of concatenation. Evaluating terminal and internal branches separately (Figure 6.5) shows that CASTLES-Pro is unbiased for both terminal and internal branches when given true gene trees; as the GTEE increases, it suffers a small overestimation bias for internal branches and a similar underestimation bias for terminal ones; thus, effects of gene tree error are reduced but not fully eliminated in CASTLES-Pro. Finally, distance-based methods have a small overestimation bias for internal branches that increases as GTEE increases, and a much larger bias for terminal branches.

The trends for log error, which emphasizes short branches more than absolute error, are similar, and CASTLES-Pro is the most accurate method overall, followed by CASTLES, distance-based methods, and finally concatenation. The only difference, according to log

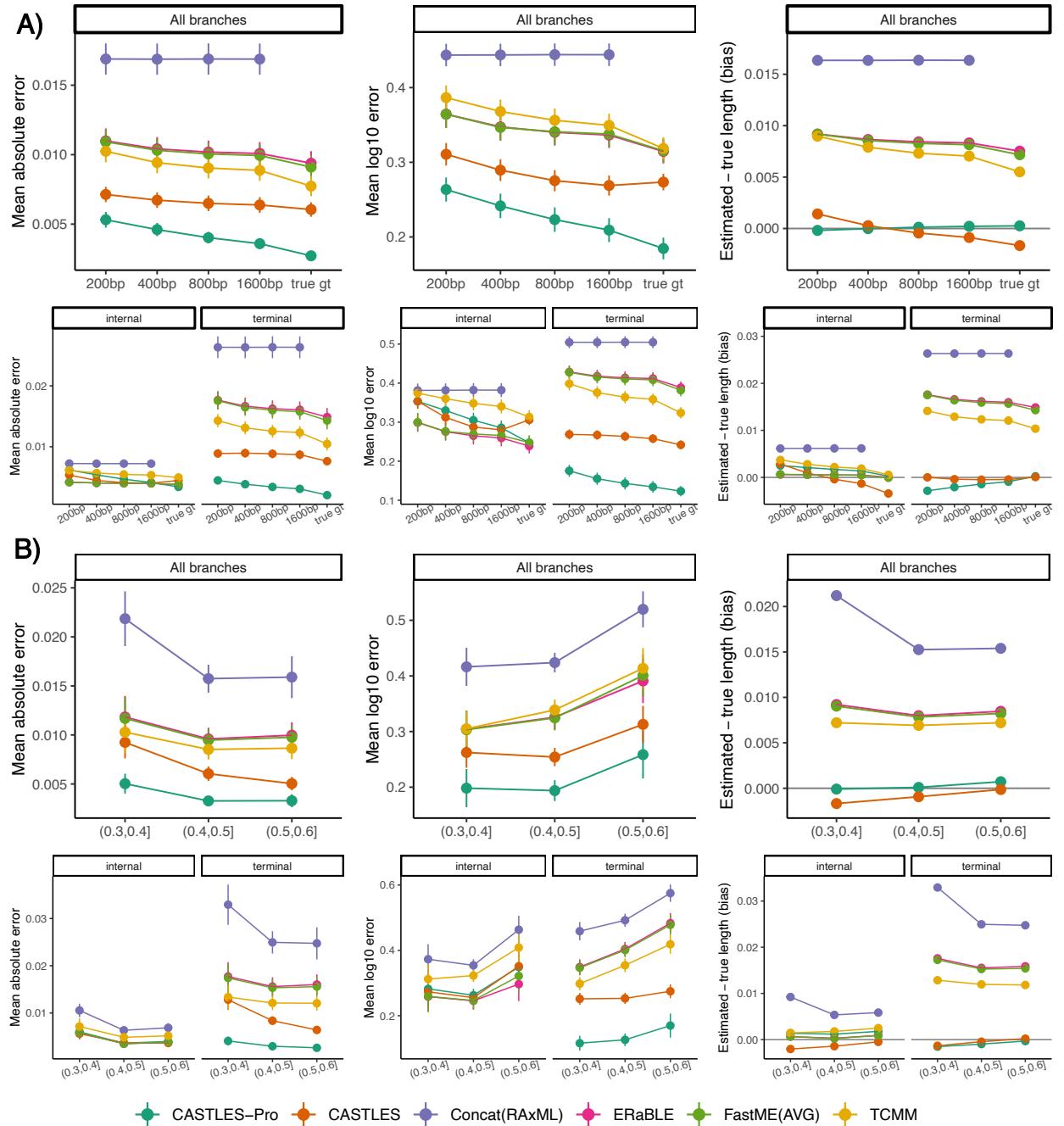


Figure 6.5: Mean absolute error, mean log error and bias for different branch length estimation methods on 100-taxon simulated ILS datasets. The average ILS level on this dataset is 47% AD. The number of genes is 1000, and the number of replicates is 50. A) Varying gene tree error (GTEE) (x-axis) where the GTEE level changes between 0% for true gene trees to 55% for gene trees estimated from 200bp alignments. B) Varying the level of ILS (x-axis) for conditions with 1600bp sequence length. The number of replicates in the three ILS bins are 9, 29, and 12, respectively. See also Figure 6.11 for comparison between CASTLES-Pro and CASTLES-Pro+TCMM.

error, is that TCMM is overall slightly less accurate than the other two distance-based methods (ERaBLE and FastME(AVG)) but still more accurate than them for [terminal](#) branches.

6.4.2 GDL+ILS simulations

In the presence of [GDL](#) and [ILS](#), CASTLES-Pro has the lowest error and bias in many but not all conditions, with concatenation used with DISCO (CA-DISCO) performing better with low [ILS](#) according to some metrics (Figure 6.6). CASTLES-Pro is always the most accurate method in the high [ILS](#) conditions, followed by CA-DISCO, but the gap is particularly large for lower [duplication rates](#) (Figure 6.12) or large numbers of gene trees (Figure ??). For the low [ILS](#) condition, CASTLES-Pro is still better than CA-DISCO according to log error in most conditions but is outperformed in some conditions according to the mean absolute error metric; in particular, CA-DISCO clearly outperforms CASTLES-Pro with 20 species according to the mean absolute error metric regardless of the [duplication rates](#) (Figure 6.12), number of genes (Figure 6.13), or sequence length (Figure 6.15). However, with more species, CASTLES-Pro either outperforms or matches CA-DISCO even with low [ILS](#) (Figures 6.6,6.16). Since log error emphasizes short branches more than mean absolute error, these trends suggest that CASTLES-Pro is doing a consistently better job at estimating short branches, whereas concatenation is sometimes better at estimating long branches. Other methods are less competitive. CASTLES run on DISCO decomposed gene trees is less accurate than CASTLES-Pro in most conditions across both 20-taxon and 100-taxon datasets (Figures 6.12,6.13,6.15,6.16,6.14,6.17). Distance-based methods are the least accurate in almost all conditions.

The number of genes, [sites](#) per gene, and species all impact accuracy in various ways. As the number of genes increases, the error of CASTLES-Pro drops faster than CA-DISCO for both 20 (Figure 6.13) and 100 species (Figure 6.14). Similarly, increasing the sequence length and thus decreasing [GTEE](#) does not impact CA-DISCO, but makes CASTLES-Pro and other methods more accurate (Figure 6.15), with CASTLES-Pro improving the fastest, especially with 100 species (Figure 6.6C). The number of species has a mixed impact, which depends on the rate of duplication, the measure of error, and the choice of method (note that tree heights are fixed when the number of species changes, creating shorter branches with more species). The impact of the [duplication rate](#) also depends on the level of [ILS](#) and method. Overall, CASTLES-Pro is relatively robust, retaining similar error and bias levels across different [duplication rates](#) (Figures 6.6,6.12). Methods that rely on DISCO to decompose the trees tend to become better with higher [duplication rates](#), especially with

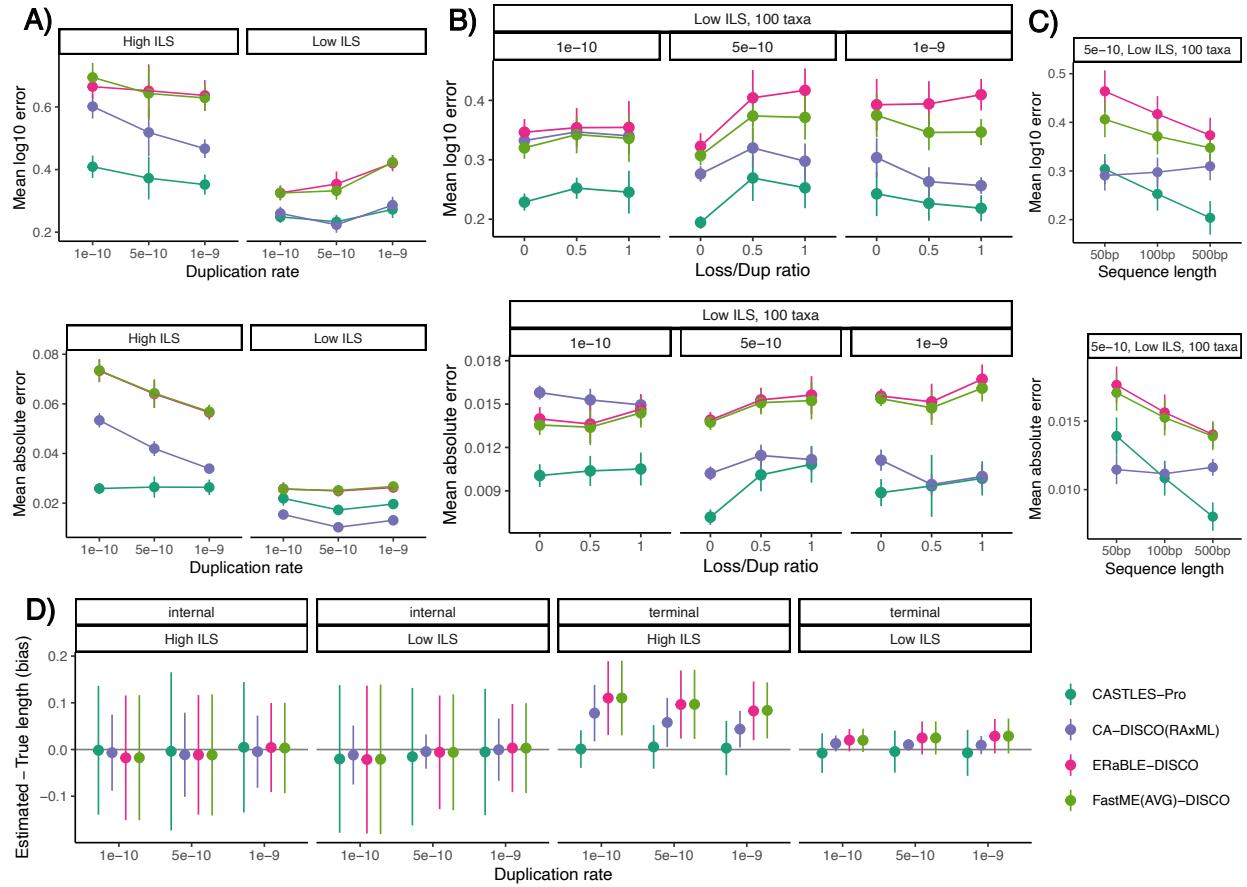


Figure 6.6: Mean log error, mean absolute error, and bias of branch lengths for simulated GDL+ILS datasets, varying number of species, **duplication rate**, **ILS** level, and sequence length. When not specified, each parameter is set to default: 1000 genes, 100bp sequence length, equal loss and **duplication rates**, 20 species. A) 20-taxon datasets varying **duplication rate** and **ILS** rate. B) 100-taxon datasets, low **ILS** condition, varying **duplication rate** and loss/dup ratio. C) 100-taxon datasets, 5e-10 **duplication rate**, low **ILS** condition, varying sequence length. D) Bias divided by **terminal** and **internal** length for the conditions in panel A. The number of replicates is 10. See Figure 6.12-6.17 for full results.

high ILS. Increasing [loss rates](#), however, can increase the error of CASTLES-Pro in some cases but does not introduce any discernible bias (Figures [6.6](#),[6.17](#)).

Overall, CASTLES-Pro has a lower bias than other methods, especially for [terminal](#) branches (Figure [6.6D](#), Figure [6.12](#) to Figure [6.17](#)). CA-DISCO clearly overestimates [terminal](#) branches, especially for higher [ILS](#) levels; in contrast, CASTLES-Pro does not have a clear bias for [terminal](#) branches. For [internal](#) branches, all methods are less biased, with CA-DISCO and CASTLES-Pro performing slightly better for low and high [ILS](#) conditions, respectively. The distance-based methods also have a clear overestimation bias for [terminal](#) branches. Overall, the most glaring form of bias is for [terminal](#) branches for high [ILS](#) conditions in all experiments, a problem that CASTLES-Pro eliminates.

Table 6.2: Runtime and peak memory usage of different methods for 100-taxon GDL+ILS dataset for 10,000 genes with sequence length of 100bp. The [duplication rate](#) is 5×10^{-10} with equal [loss rate](#). The results are averaged across 10 replicates. The runtime does not include gene tree estimation or [species tree topology](#) estimation time, as all methods draw branch lengths on a fixed tree topology. See also Figure [6.18](#).

| | time (minutes) | peak memory (GB) |
|-------------------|----------------|------------------|
| CASTLES-Pro | 0.96 | 4.09 |
| CASTLES-DISCO | 6.45 | 3.64 |
| CA-DISCO(RAxML) | 123.72 | 29.26 |
| ERaBLE-DISCO | 43.35 | 12.78 |
| FastME(AVG)-DISCO | 19.61 | 8.81 |

6.4.3 Scalability

CASTLES-Pro also has a runtime advantage on the 100-taxon [GDL](#) datasets, and its advantage becomes more clear as the number of genes increases (Figure [6.18](#)). In particular, for 10,000 genes, CASTLES-Pro takes on average less than 1 minute to estimate branch lengths on a fixed tree topology, while CA-DISCO takes about 124 minutes (Table [6.2](#)). The gap between CASTLES-Pro and CA-DISCO widens as gene family trees become larger (Table [6.5](#)); with the highest [duplication rate](#) (10^{-9}) and no loss, CASTLES-Pro finishes in 4 minutes on average, while CA-DISCO takes more than 25 hours on average (Figure [6.18](#)). In terms of memory usage, CASTLES-DISCO and CASTLES-Pro are almost identical and use much less memory than other methods (Tables [6.2](#),[6.18](#)). Finally, in the model condition with 1000-taxon trees and 1000 genes, CA-DISCO and distance-based methods fail due to memory limit given 128GB of RAM, while CASTLES-Pro and CASTLES-DISCO finish in 2 and 11 minutes on average resp. and use less than 4 GB of memory.

6.4.4 HGT+ILS simulations

The first four model conditions of this dataset have low **HGT** rates and little discordance beyond **ILS** (increasing from 30% with ILS-only to 34%). Therefore, the log error for all methods has almost no change across these four conditions, and the mean absolute error fluctuates within the bounds of standard error (Figure 6.19A). However, the last two conditions have substantially higher **HGT** rates, with 53.4% and 68.4% total discordance (Table 6.4). Comparing the last three conditions, both log and mean absolute error for all methods generally increase for higher **HGT** rates. Distance-based methods are generally the least accurate across different conditions, except for the highest **HGT** rate, where concatenation has a higher mean absolute error. CASTLES-Pro is the most accurate for both metrics across all conditions, except for the highest **HGT** rate, where it has a tie with CASTLES in terms of log error. While TCMM is inaccurate, following CASTLES-Pro by regularized TCMM, as detailed by [75] further improves its accuracy and obtains the best results overall. The gap between CASTLES-Pro (with or without TCMM) and concatenation widens as **HGT** increases in terms of absolute error (i.e., focusing on long branches) but closes for mean log error (focusing on short branches).

In terms of bias, **terminal** and **internal** branches again show different patterns (Figure 6.19B). CASTLES-Pro and CASTLES have an underestimation bias in all model conditions, especially for **internal** branches. This underestimation mostly disappears if CASTLES-Pro is followed by TCMM. Concatenation and, to a smaller degree, distance-based methods have a large overestimation bias for **terminal** branches that increases with **HGT** rates. The overestimation of **terminal** branches by concatenation is, on average, 2.77 times larger than the underestimation for **terminal** or **internal** branches by CASTLES-Pro and 5.45 times larger than the underestimation bias of CASTLES-Pro+TCMM over all branches.

6.4.5 Biological datasets

ILS. On the three datasets where biological discordance is likely dominated by **ILS** (birds, bees, and mammals), we observe that CASTLES-Pro produces shorter lengths than concatenation, especially for **terminal** branches (Figure 6.7). This pattern is more extreme for the birds and bees datasets (13.8% and 10.6% increase in average **root-to-tip** distance respectively), which have a particularly high level of observed gene tree discordance. In addition, changes are more pronounced for **terminal** branches (Figure 6.7) and especially short **terminal** branches (Figures 6.20,6.21), as expected by theory and results of the simulations.

For **terminal** branches of the bird dataset, concatenation has a slightly shorter length for

only one species and substantially longer for others. The median (1st and 3rd quantiles) reduction from concatenation to CASTLES-Pro lengths is 0.00333 SU (0.00247, 0.00454); if we assume a mean substitution rate of 0.0026 per million years as estimated by [328] and 6.6 years per generation (median across species reported by [339]), the gap between concatenation and CASTLES-Pro corresponds to the expected 1 CU if $N_e = 0.00333/0.0026 \times 10^6/6.6 = 194k$ (144k, 265k), which matches previous estimates using PSMC [340].

Similarly, for bees, there are only three shorter terminal branches in concatenation compared to CASTLES-Pro, and one of them (*S. schubotzi*) becomes substantially longer. This increase is due to one gene tree with the clearly incorrect terminal length of 2.63 for *S. schubotzi*. Removing this single gene tree or using TreeShrink [341] to filter abnormally long branches both reduce the length of this branch in the CASTLES-Pro output from 0.0454 to 0.0145 or 0.0157, which are below concatenation (Figure 6.20). Using original gene trees (including outliers), we see the expected decrease in terminal lengths, with a median of 0.00255 (0.00159, 0.0047), which, divided by the spontaneous mutation rate of 3.5×10^{-9} per generation given by [342], corresponds to $N_e = 729k$ (452k, 1,350k) for 1 CU, which is in line with estimates based on PSMC [343].

Finally, for mammals, four terminal branches become longer in CASTLES-Pro (including one with mislabeled taxa) while others become shorter. The elongated four are due to outliers as using TreeShrink shortens all these four branches, with two becoming shorter than concatenation and the other two remaining only 0.5% and 1.4% longer than concatenation. Even without TreeShrink, terminal branches shrink in CASTLES-Pro by a median of 0.0036 (0.0014, 0.0052), which assuming a per generation mutation rate of 2.5×10^{-8} [344] would correspond to 1 CU for $N_e = 145k$ (57k, 206k).

GDL. Patterns on GDL datasets differ from ILS-dominated datasets (Figure 6.7) perhaps because here concatenation is run on single-copy genes while CASTLES-Pro is run on the full set of multi-copy gene trees. On the 1KP plant dataset, CASTLES-Pro run on 9610 multi-copy gene trees has longer terminal and internal branch lengths than concatenation run on 424 single-copy genes (Figures 6.24 and fig:biological-rtt-ratio), leading to 24.2% higher mean root-to-tip distance. Similarly, on the fungal dataset, CASTLES-Pro based on 7,180 multi-copy gene family trees produces longer branches and 10.1% higher average root-to-tip distance compared to concatenation on 706 single-copy genes (Figures 6.7, 6.25). On the small and less diverse 40-taxon eudicots dataset, CASTLES-Pro based on 2,573 multi-copy gene trees results in shorter terminal branches but longer internal branch lengths than concatenation based on 345 single-copy genes (Figures 6.7 and 6.23), with 7.7% decrease in average root-to-tip distance. Thus, patterns were different between these two plant datasets

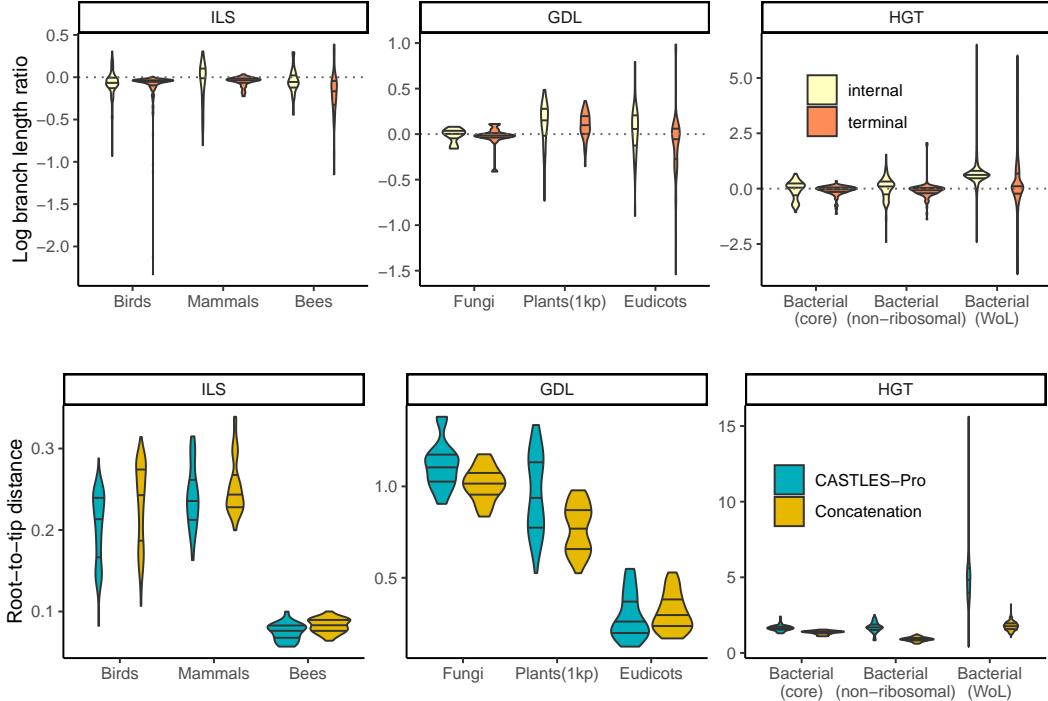


Figure 6.7: (top) The branch lengths produced by CASTLES-Pro divided by branch lengths of concatenation on nine biological datasets with different sources of gene tree heterogeneity in log scale. (bottom) Distribution of the root-to-tip distance for the CASTLES-Pro and concatenation trees on the nine biological datasets. See also Figure 6.29.

and patterns were unlike ILS datasets; we return to this point in the discussions.

Microbial data and AB branch. A long-standing hypothesis has been that bacteria and archaea domains are separated by a long branch [336, 337, 338]. In contrast, [60] (which included some of us) estimated a far shorter length for the AB branch than what was previously reported using a concatenation of 381 marker genes for the branch length estimation step. [76] further studied this and other microbial datasets and suggested (among other criticisms) that concatenation can severely underestimate branch lengths on datasets with high levels of HGT, resulting in underestimation of the AB branch length. They report estimates of AB branch as long as 3.3 based on the core gene set of [330] and 2.52 using 27 most vertically evolving genes selected from a set of manually curated marker genes including both ribosomal and non-ribosomal proteins. We reexamine some of these datasets using CASTLES-Pro.

On all three microbial datasets, we observe a general increase in the length of the internal branches, particularly longer branches, and a small decrease in the length of terminal branches (particularly short ones) for CASTLES-Pro compared to concatenation (Figures 6.7, 6.7, 6.28). Since internal branches increase more than terminal branches decrease, we

observe a substantial increase in the average [root-to-tip](#) distance on all three datasets (Figures 6.7, 6.8, 6.26, 6.27, 6.28). For the [WoL](#) dataset, branches change dramatically between the methods; however, note that [60] limited itself to 100 [sites](#) per gene due to scalability limitations of concatenation, while CASTLES-Pro uses gene trees estimated from full-length sequence alignments. The concatenation tree has a long tail of branches with 1e-6 or 2e-6 length, corresponding to no-event branches among 100×381 [sites](#) chosen, but no such tail exists for CASTLES-Pro (Figure 6.28) since it uses all [sites](#).

On all three datasets, CASTLES-Pro substantially increased the [AB](#) length compared to concatenation (Table 6.3) and made them closer to those estimated by [76]. The increases are dramatic ($17\times$) on the [WoL](#) dataset which contains highly discordant genes, substantial ($2.5\times$) on the less discordant non-ribosomal genes, and relatively small ($1.3\times$) on the presumably HGT-free core genes. On the two small less discordant datasets, there are orders of magnitude more matching quartets than non-matching quartets, signifying lack of discordance, and in both cases, length of matching quartets is much longer than non-matching ones (Table 6.3). In contrast, on [WoL](#), the number of matching and non-matching quartets are both very large, and matching quartets are only 1.7x more than non-matching ones. Nevertheless, the length of the [AB](#) branch in all three cases remains close to the average [AB](#) length in the matching quartets. Overall, CASTLES-Pro produces longer [internal](#) branches and longer [AB](#) lengths for all three bacterial datasets compared to concatenation, a trend that agrees with the observations of [76], who suggest that concatenation can underestimate branch length in the face of high [HGT](#)

Table 6.3: [AB](#) branch length on the three bacterial datasets estimated by CASTLES-Pro, TCMM and concatenation. \bar{L}_I, \bar{L}'_I refer to the average [AB](#) branch length in matching and non-matching gene trees, respectively, and C_M, C_{NM} refer to the number of matching and non-matching quartets around the [AB](#) branch. d refers to the length of the [AB](#) branch in coalescent units.

| | L_I | L'_I | C_M | C_{NM} | d | CASTLES-Pro | TCMM | CA-ML |
|---------------------|-------|--------|-----------------|-----------------|------|-------------|------|-------|
| Core | 1.92 | 0.01 | 922,226 | 32 | 9.83 | 1.79 | 1.94 | 1.43 |
| Non-ribosomal | 1.06 | 0.08 | 1,270,289 | 11,624 | 4.30 | 1.05 | 0.87 | 0.43 |
| WoL | 2.74 | 0.74 | 750,795,570,342 | 444,568,398,602 | 0.58 | 2.68 | 1.89 | 0.16 |

6.5 DISCUSSION AND CONCLUSIONS

We introduced CASTLES-Pro, a summary method that can furnish a given [species tree](#) with substitution-unit branch lengths accounting for [GDL](#) and [ILS](#) based on a given set of potentially multi-copy gene trees. Our simulations with [ILS](#) alone or [ILS+GDL](#) showed that

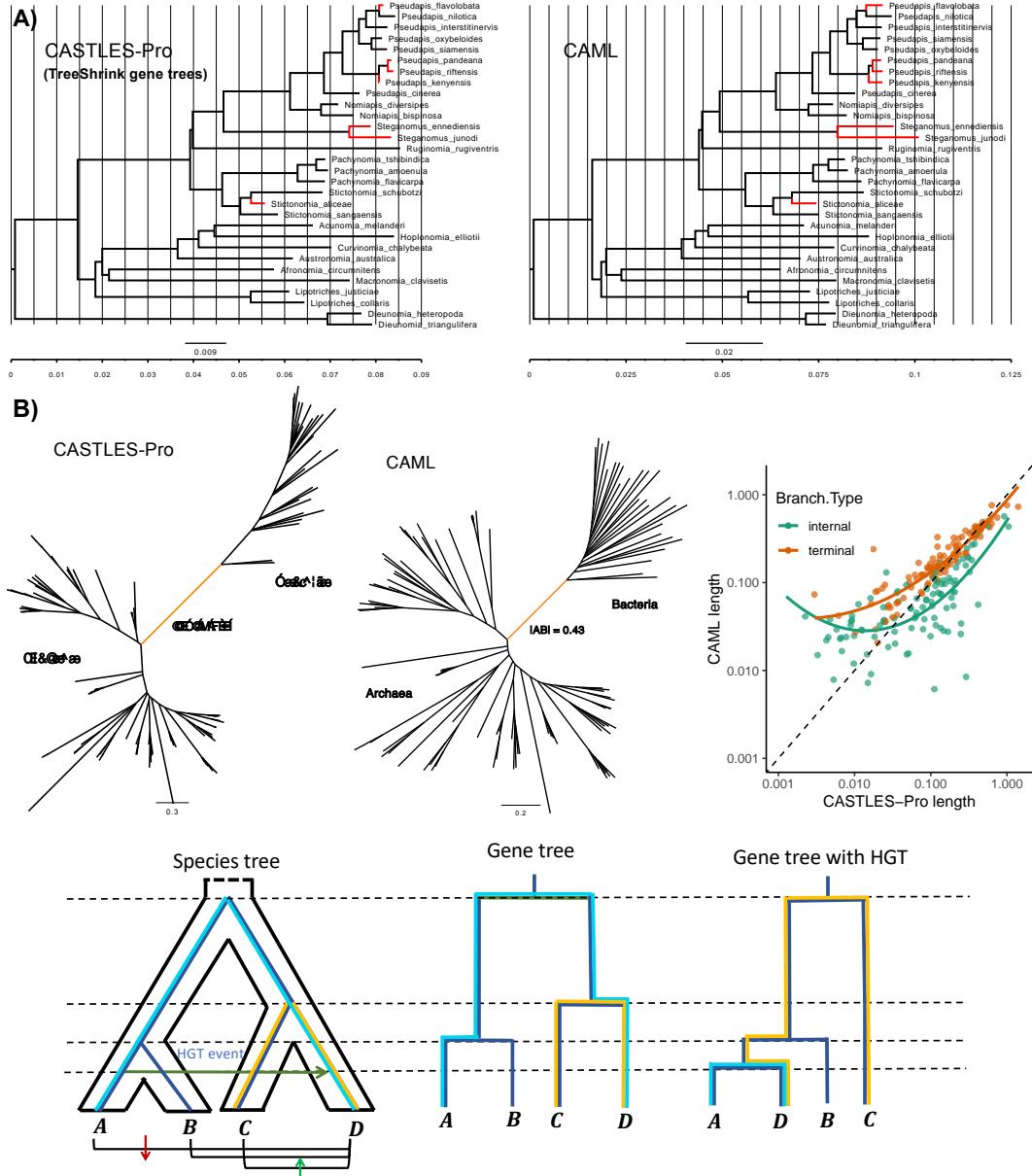


Figure 6.8: A) Comparison between the branch lengths of CASTLES-Pro (with TreeShrink gene trees) and CA-ML on the 32-taxon 853-gene bees dataset of [102] after removing the **out-group taxa** *Lasioglossum albipes* and *Dufourea novaeanglia*. We used the ASTRAL topology from the original study, and used concatenation and CASTLES-Pro to draw branch lengths on this topology. The branch lengths that are at least 2x shorter in CASTLES-Pro tree compared to the concatenation tree are highlighted in red. B) Comparison between the branch lengths produced by CASTLES-Pro and concatenation on the 108-taxon bacterial dataset with 38 non-ribosomal genes. The branch highlighted in orange separates domain Archaea from Bacteria (AB branch). C) **HGT**, if ignored (as in concatenation) can make branches longer or shorter; the **HGT** event shown by the dark green arrow creates the gene tree shown on the right, which has a shorter distance from A and B to D, but higher distances from D to C. This reduces branch length for some branches in the **species tree** (blue) and increases it for others (yellow). See also Figs. 6.20-6.28.

CASTLES-Pro is more accurate than other methods in most conditions and is also more scalable, easily running on datasets with tens of thousands of species or genes (Table 6.7) in less than an hour. In the face of HGT, which CASTLES-Pro does not directly model, it was still more accurate than concatenation but had room for improvement. In particular, using CASTLES-Pro with automated outlier removal methods reduced its bias for high HGT cases, a topic explored by [75]. Paired with summary methods that infer the topology, this advance makes the two-step approach to species tree inference far more useful than before. The ASTER package [316] outputs trees with SU branch lengths when used with ASTRAL-IV (for single-copy genes) or ASTRAL-Pro-2 (for multi-copy genes) using the CASTLES-Pro algorithm. These trees can readily be used as input to downstream analyses such as dating, a topic explored by [324].

The negative impact of concatenation on branch lengths depended on the cause of discordance, as biological analyses clearly show (Figure 6.7). For ILS, as expected, terminal branch lengths, especially shorter ones, are over-estimated using concatenation, but internal branches have far less bias overall. This imbalance between the error for terminal and internal branches can lead to unexplainable patterns in downstream analyses, such as diversification rates, a topic explored by [324]. Note that for internal branches, depending on the coalescent length of surrounding branches, we may still have overestimation or underestimation using concatenation, but looking across all internal branches, those effects diminish.

In contrast to ILS, on GDL datasets where CASTLES-Pro was given all loci and sites per locus and concatenation was based on fewer genes and sites, CASTLES-Pro generally produced longer branches. One explanation for the increase instead of decrease in branch lengths in GDL datasets is ascertainment bias. The small portion of loci that happened to be single-copy tend to be the most conserved ones, giving a biased picture of the substitution rates. By allowing the use of all multi-copy gene trees, CASTLES-Pro reveals higher genome-wide substitution rates compared to single-copy genes. An alternative explanation is that single-copy gene trees are easier to align due to being smaller (and more conserved), and the higher branch lengths from CASTLES-Pro could be a result of over-alignment in multi-copy genes or perhaps saturation. The real GDL data likely suffer from a mixture of both, especially given that angiosperm data (which span a far shorter evolutionary time) did not experience branch length increase.

For HGT, our simulations showed an over-estimation bias for concatenation, whereas, on real microbial data, we seemed to observe the opposite. For the AB branch (but also others), concatenation under-estimates lengths compared to CASTLES-Pro and compared to using HGT-free genes. The difference is likely due to the type of HGT events. Any single HGT event both increases and decreases divergence for some pairs of taxa compared to the

species divergence, leading to both under-estimation and over-estimation bias for different branches (see Figure 6.8C). Our simulations using Simphy augment ILS with *random* HGT events, each of which can create bias in either direction for individual branches. When many such events accumulate, they can cancel each other out, leading to a less clear HGT signature and leaving us with impacts of ILS. Real biological data often experience highways of HGT when large numbers of genes are transferred between two points of the tree. Such highways are expected to have happened around the AB branch [345, 346]. One expects HGT highways between two branches to create a strong under-estimation bias for branches connecting the donor and recipient; a large portion of the concatenated alignment will have reduced divergence between pairs of species, one from the donor and one from the recipient (see Figure 6.8C). Our results are consistent with the claim by [76] that the AB length seems to be underestimated by concatenation for this reason.

The underestimation around the AB branch seemed to be fixed in CASTLES-Pro, but the reason needs explanation since this phenomenon is separate from the coalescent dynamics CASTLES-Pro models. When a branch has a long estimated CU length, the relatively few gene tree quartets that disagree with the species tree can have abnormally long lengths, even exceeding the matching ones on average. This is exactly the situation for the AB branch (Table 6.3). In such cases, CASTLES-Pro resorts to using the mean length of matching gene tree quartets for the species tree and thus effectively ignores coalescent effects, which is defensible for a branch with a large CU length. In doing so, it eliminates gene tree quartets that disagree with the species tree, which are presumably due to HGT for the AB branch. This feature (and not expectations under coalescence) leads CASTLES-Pro to output a large distance for the AB branch in contrast to concatenation.

Our method (together with [75, 324]) paves the way for a new four-stage phylogenomics pipeline that uses scalable methods in each step: Estimate gene trees independently, estimate the species tree topology by summarizing the gene tree topologies, estimate the branch lengths using CASTLES-Pro (optionally followed by TCMM for high HGT), and date the tree using scalable methods such as TreePL [228] or MD-CAT [347]. With this pipeline, we can easily handle datasets with thousands of species and thousands of genes.

6.6 METHODS AND SOFTWARE COMMANDS

Here we bring the details of the methods and software commands. All experiments were performed on the University of Illinois campus cluster, with a memory limit of 128GB.

- **DISCO.** We used DISCO [90] version 1.3 to decompose multi-copy gene-family trees into single-copy gene trees. DISCO is available at <https://github.com/JSDoubleL/DISCO>. We used the following command:

```
python3 disco.py -i <multi-copy-gene-trees> -o  
<single-copy-gene-trees> -d _
```

- **CA-DISCO.** To run concatenation on multi-copy gene family sequences, we used the script `ca_disco.py` available at <https://github.com/JSDoubleL/DISCO> with the following command:

```
ca_disco.py -i <gene_tree_path> -a <alignment_list_path> -t  
<taxa_list_path> -o <output_path> -d _
```

where `<gene_tree_path>` is the path to the set of multi-copy gene family trees, `<align-ment_list_path>` is a file containing the list of individual sequence alignments for each gene family, and `<taxa_list_path>` is the set of all taxa. The output is the concatenated sequence alignment, that is then passed to RAxML (v8.2.12) [25], available at <https://github.com/stamatak/standard-RAxML>, to optimize branch lengths on a fixed tree `topology` with the option `-f e` using the following command.

```
raxmlHPC -PTHREADS -f e -t <species_tree_path> -m GTRGAMMA -s  
<alignment_path> -n RES -p 4321 -T 16
```

- **ERaBLE.** To run ERaBLE [322], we first calculated a matrix of pairwise `patristic distances` per gene using a custom script available at https://github.com/ytabatabae/CASTLES-Pro-paper/scripts/patristic_dist_matrix.py that uses `calculate_treecompare_get_length_diffs` from the package DendroPy [299]. When calculating the `patristic distance` matrix, we impute missing values with averages: i.e., if two `taxa` i and j do not appear in the same gene tree g together due to missing `taxa` in genes, as is the case in DISCO gene trees, the `patristic distance` of i and j in the matrix for gene tree g is replaced by the average `patristic distances` of these two `taxa` in the rest of the gene trees where they appear together. We used the following command to run this script

```
python3 patristic_dist_matrix.py -g <gene_tree_path> -o
<dist_mat.phylip> -m all
```

Then we ran ERaBLE (v1.0) available at <http://www.atgc-montpellier.fr/erable/> with the following command;

```
erable -i <dist_mat.phylip> -t <species_tree_path> -o <output_path>
```

- **FastME.** Similar to ERaBLE, to run FastME [297], we first computed a *single* distance matrix corresponding to average patristic distances between pairs of taxa using the following command

```
python3 patristic_dist_matrix.py -g <gene_tree_path> -o
<dist_mat.phylip> -m avg
```

and then we ran FastME version 2.1.6.2 with the following command

```
fastme-2.1.6.2-linux64 -i <dist_mat.phylip> -w BallS -u
<species_tree_path> -o <output_path>
```

- **CASTLES-Pro.** CASTLES-Pro is integrated inside the species tree estimation software ASTER that is available at <https://github.com/chaoszhang/ASTER>. To infer branch lengths on a fixed tree topology using ASTER (v1.19.3.5), we used the following commands for multi-copy and single-copy gene trees respectively

```
bin/ASTRAL-Pro2 -i <gene-tree-path> -C -c <species-tree-topology>
-o <output-path> --root <outgroup-name>
--genelength <gene-seq-length>
```

```
bin/astral4 -i <gene-tree-path> -C -c <species-tree-topology>
-o <output-path> --root <outgroup-name>
--genelength <gene-seq-length>
```

where `--root` specifies the `outgroup` name (if known) and `--genelength` specifies the average gene sequence length (default: 1000bp).

- **CASTLES.** CASTLES [74] is available at <https://github.com/ytabatabaei/CASTLES>. To run it, we first annotated the species tree topology using ASTER with the following command

```
astral -C -i <gene-tree-path> -c <species-tree-topology> -o
<output_path> --root <outgroup-name> > <annotated.tre>
```

where the annotated tree is printed to the file <annotated.tre>. We then ran the CASTLES script (v1.0) using the following command

```
python3 castles.py -t <annotated.tre> -g <gene_tree_path> -o  
<output_path>
```

- **TCMM.** TCMM [75] is available at <https://github.com/shayesteh99/TCMM>. To run the per-gene version, we used the command:

```
python3 multiple_tree_matching.py -i <species-tree-topology> -r  
<gene_tree_path> -l <lambd> -o <output_path>
```

To run the consensus version of TCMM, we used the command:

```
python3 weighted_tree_matching.py -i <species-tree-topology> -r  
<gene_tree_path> -l <lambd> -o <output_path>
```

6.7 ADDITIONAL FIGURES AND TABLES

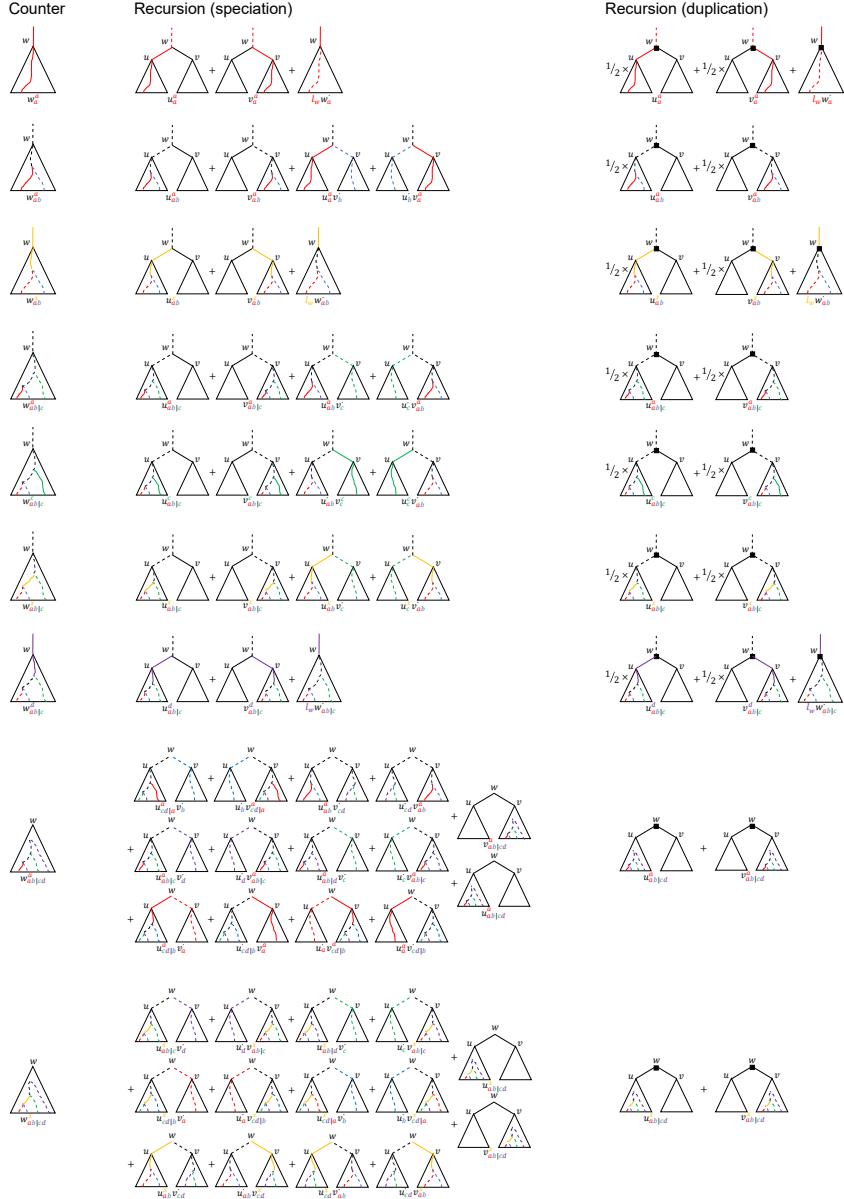


Figure 6.9: Counters for computing the weighted sums for **internal** and **terminal** branch lengths of gene tree quartets aligning with the species quadripartition $ab|cd$. The weighted mean **internal** branch lengths related to $ab|cd$ are $\sum_{w \in R} w_{ab|cd}^x / \sum_{w \in R} w_{ab|cd}$ for the matching case and $\sum_{w \in R} w_{ac|bd}^x + w_{ad|bc}^x / \sum_{w \in R} w_{ac|bd} + w_{ad|bc}$ for the non-matching cases (similarly computed by permuting a, b, c , and d). To compute the weighted mean for **terminal** branch lengths for the matching case, we use $\sum_{w \in R} w_{ab|cd}^a / \sum_{w \in R} w_{ab|cd}$, $\sum_{w \in R} w_{ab|cd}^b / \sum_{w \in R} w_{ab|cd}$, $\sum_{w \in R} w_{ab|cd}^c / \sum_{w \in R} w_{ab|cd}$, and $\sum_{w \in R} w_{ab|cd}^d / \sum_{w \in R} w_{ab|cd}$, respectively; to compute the weighted mean for **terminal** branch lengths for the non-matching cases, we use $\sum_{w \in R} w_{ac|bd}^a + w_{ad|bc}^a / \sum_{w \in R} w_{ac|bd} + w_{ad|bc}$, $\sum_{w \in R} w_{ac|bd}^b + w_{ad|bc}^b / \sum_{w \in R} w_{ac|bd} + w_{ad|bc}$, $\sum_{w \in R} w_{ac|bd}^c + w_{ad|bc}^c / \sum_{w \in R} w_{ac|bd} + w_{ad|bc}$, and $\sum_{w \in R} w_{ac|bd}^d + w_{ad|bc}^d / \sum_{w \in R} w_{ac|bd} + w_{ad|bc}$, respectively.

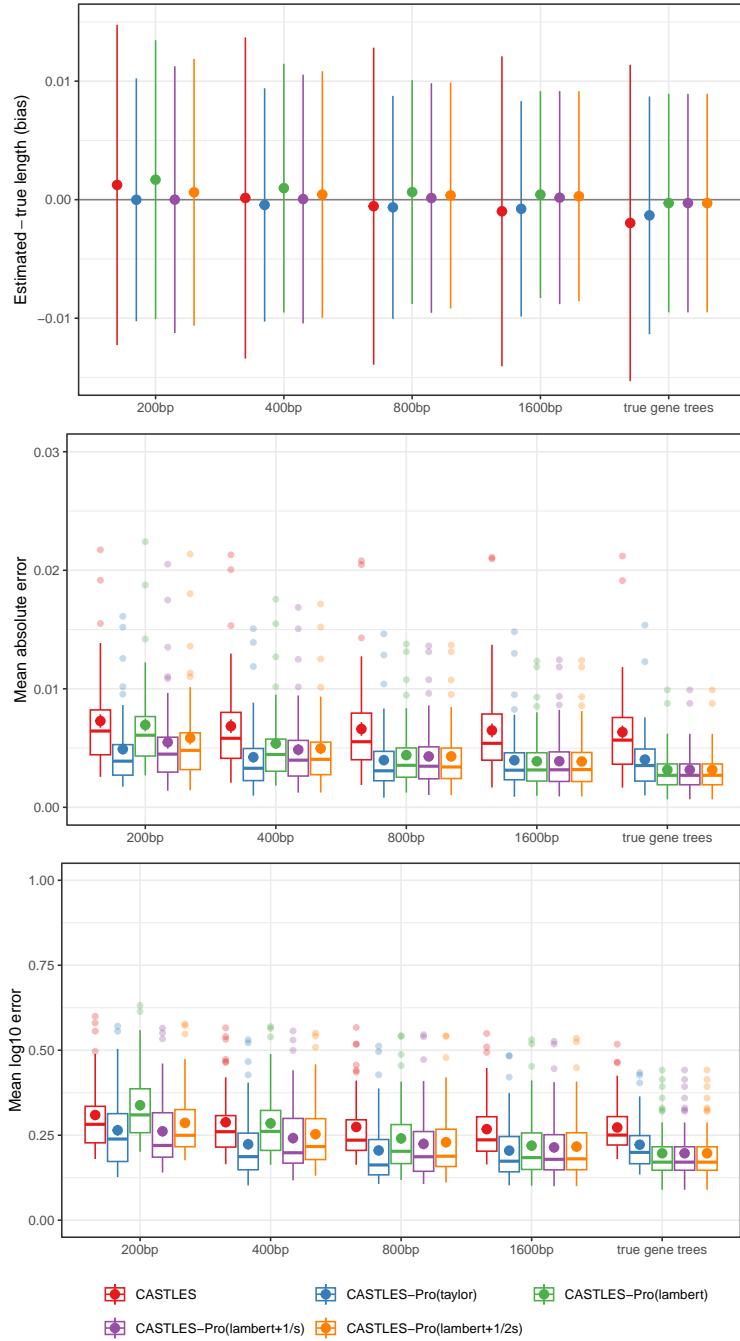


Figure 6.10: Bias, mean absolute error and mean log error for variants of CASTLES-Pro and CASTLES on 100-taxon simulated ILS datasets. The four variants of CASTLES-Pro either use the taylor approximation for calculating the length of the internal branch, or lambert function with two different pseudo-counts based on sequence lengths. The average ILS level on this dataset is 47% AD and the GTEE level varies between 0% for true gene trees to 55% for gene trees estimated from 200bp alignments. The number of genes is 1000 and the number of replicates is 50. The method shown in purple (CASTLES-Pro(lambert+1/s)) is the variant that we selected. The difference between CASTLES and CASTLES-Pro(taylor) is the way they compute mutation rates.

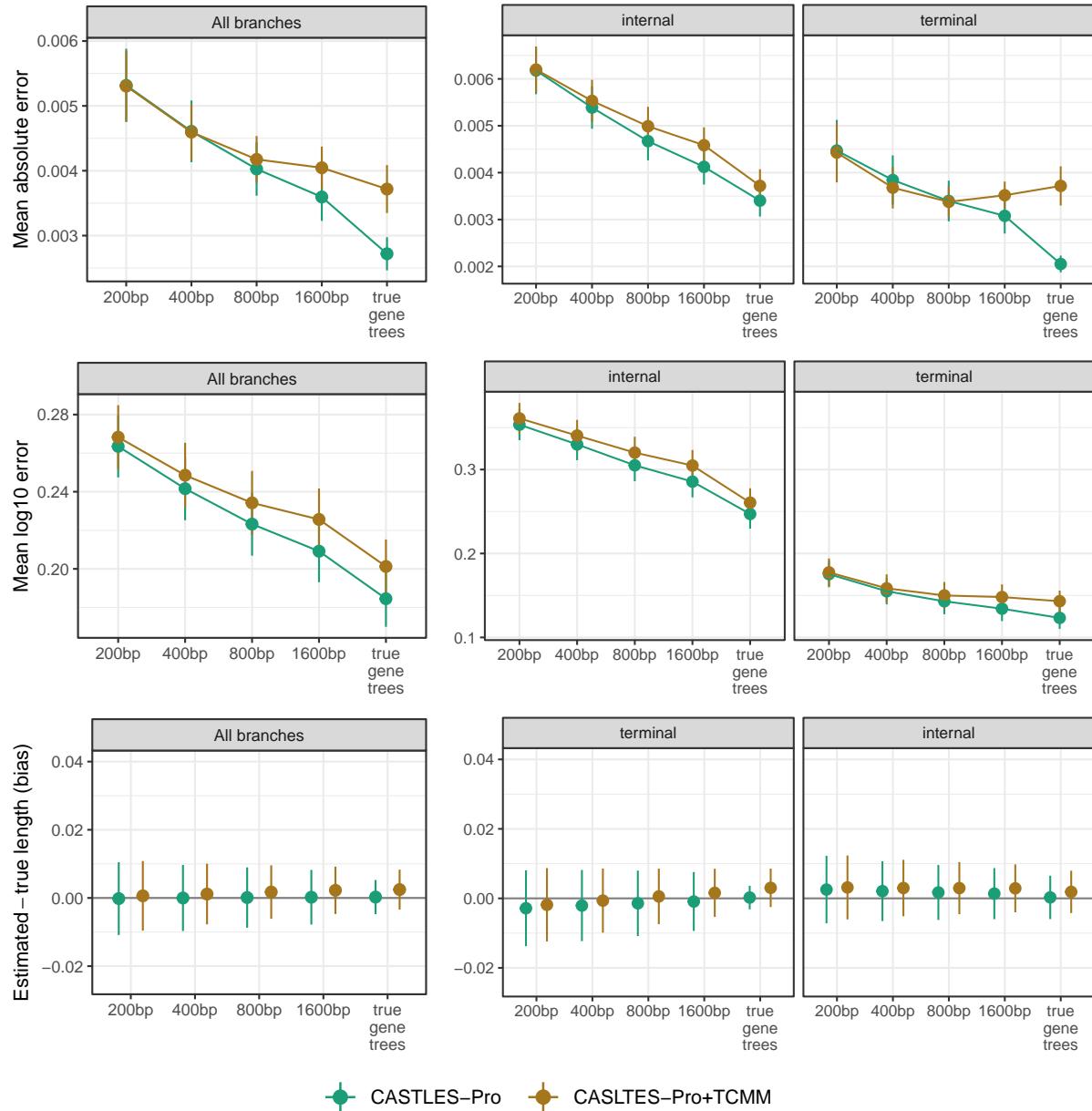


Figure 6.11: Bias, mean absolute error and mean log error for CASTLES-Pro and CASTLES-Pro+TCMM on 100-taxon simulated ILS datasets. The average ILS level on this dataset is 47% AD and the GTEE level varies between 0% for true gene trees to 55% for gene trees estimated from 200bp alignments. The number of genes is 1000 and the number of replicates is 50.

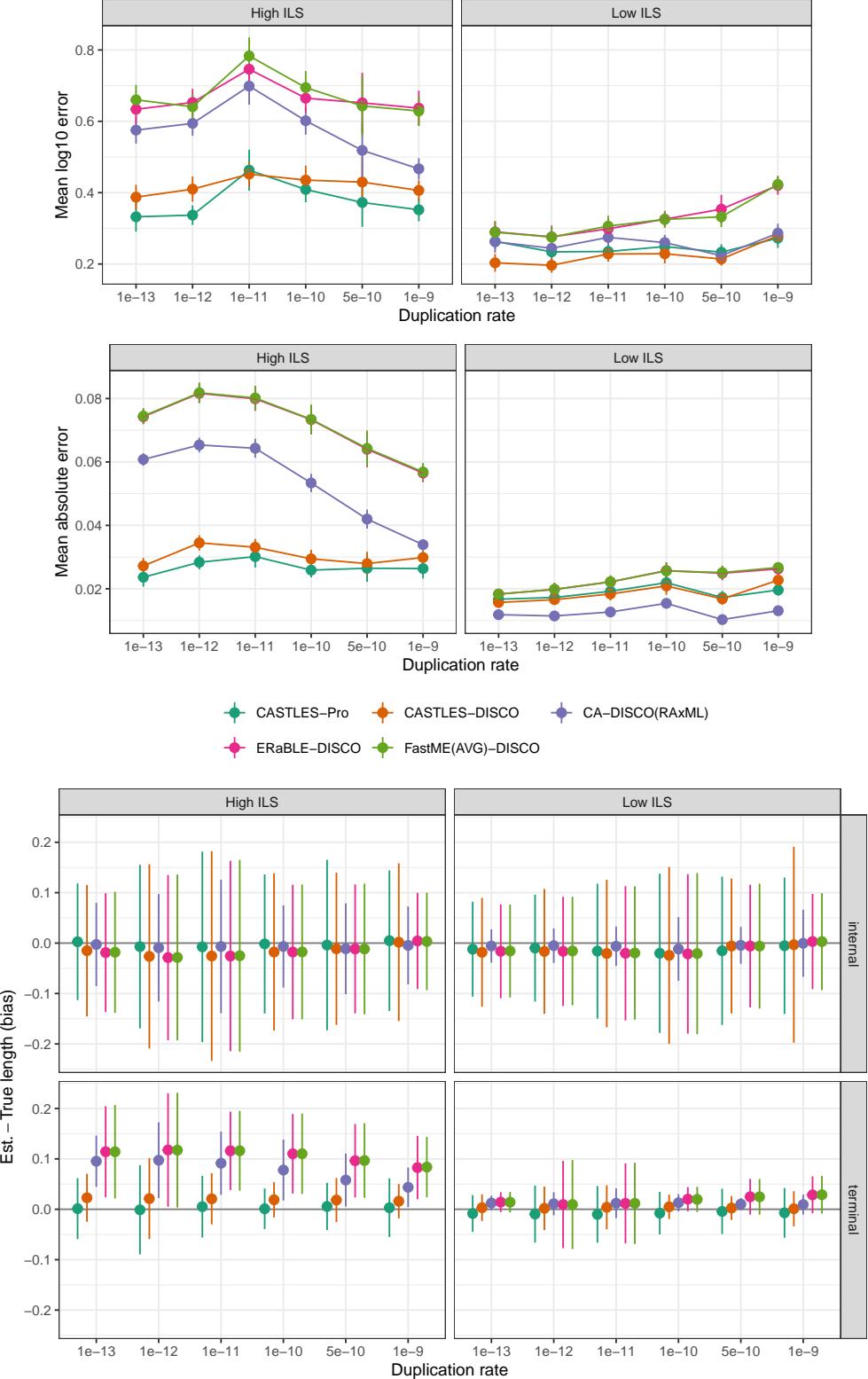


Figure 6.12: Mean log₁₀ error, mean absolute error and bias on the GDL datasets for varying duplication rates. The gene trees are estimated from 100bp alignments with average GTEE levels that varies between 33.7% to 42.2% for the low ILS condition and 38.3% to 42.6% for the high ILS condition. The number of taxa is 20, the number of genes is 1000 and the number of replicates is 10.

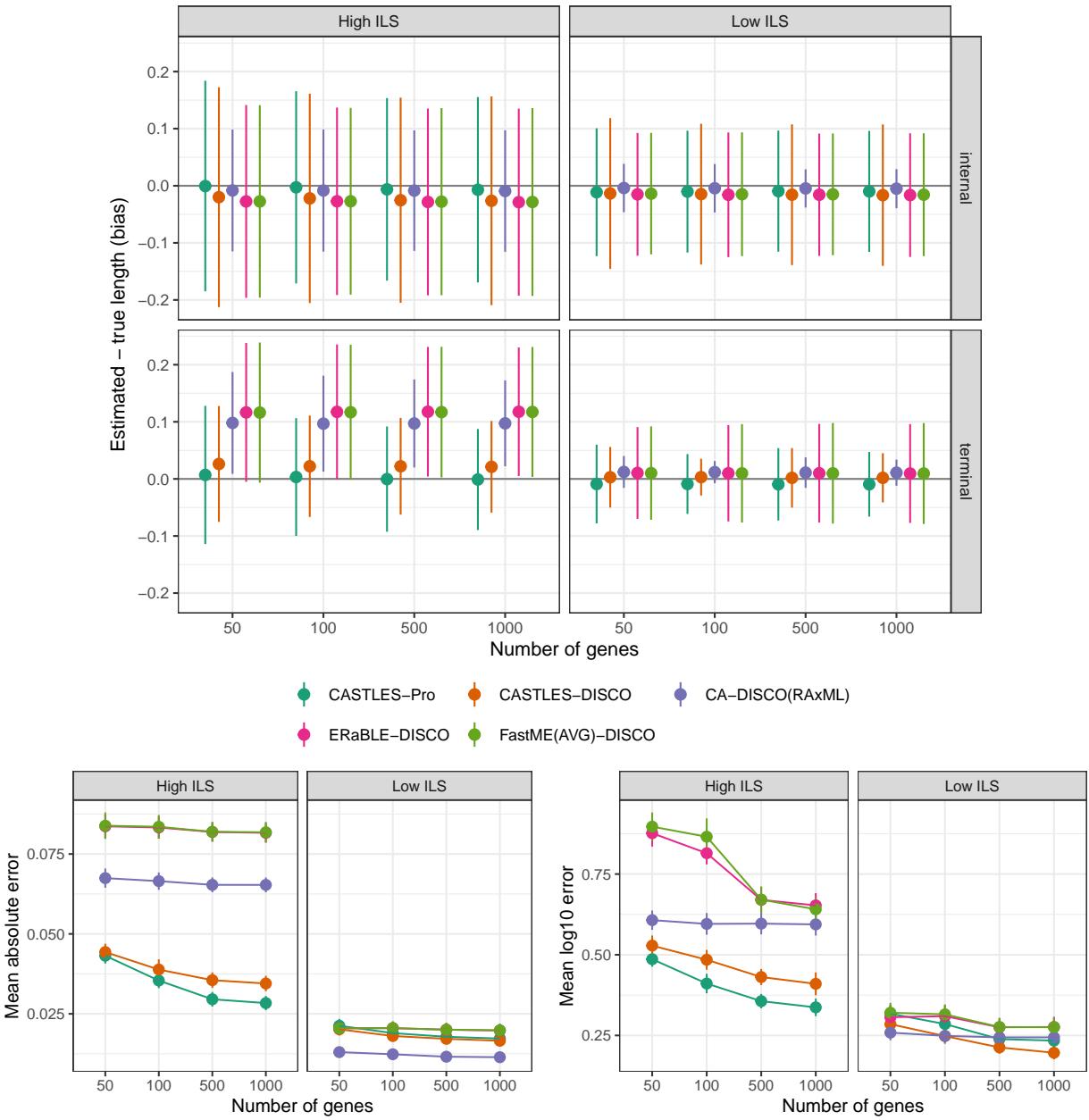


Figure 6.13: Bias, mean absolute error and mean log₁₀ error for simulated GDL datasets for varying number of genes. The duplication rate is 10^{-12} with an equal loss rate. Gene trees are estimated from 100bp alignments and the average GTEE rates for 1000 genes for the low ILS and high ILS conditions are 41.5% and 42.6% respectively. The number of taxa is 20 and the number of replicates is 10.

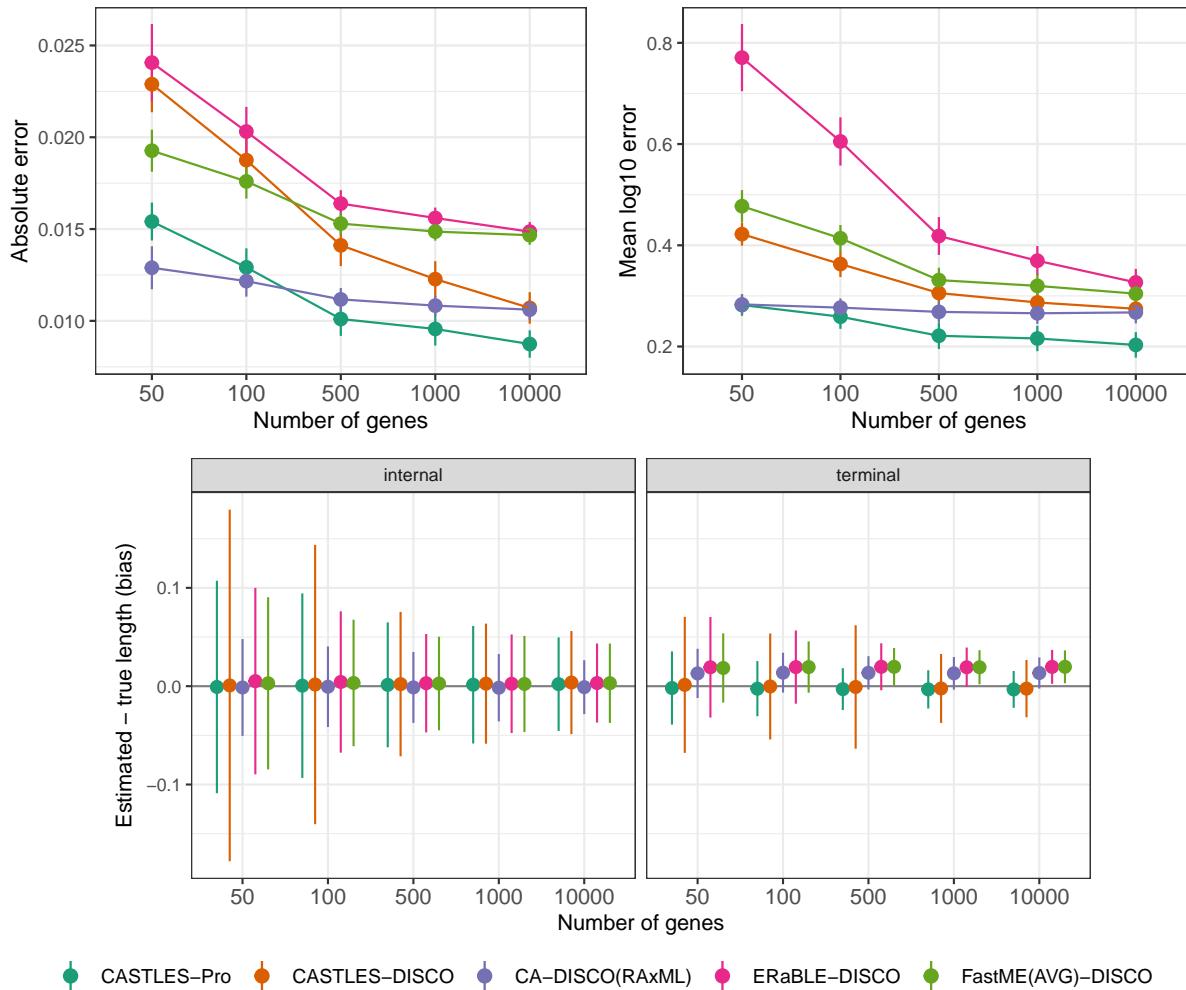


Figure 6.14: Mean log error, mean absolute error and bias on the 100-taxon [GDL](#) datasets for different number of genes. The [duplication rate](#) is 5×10^{-10} with equal [loss rate](#) and the level of [ILS](#) is low (20.3% [AD](#)). Gene trees are estimated from 100bp alignments with an average [GTEE](#) level of 41.1% for 10,000 genes. The number of replicates is 10.

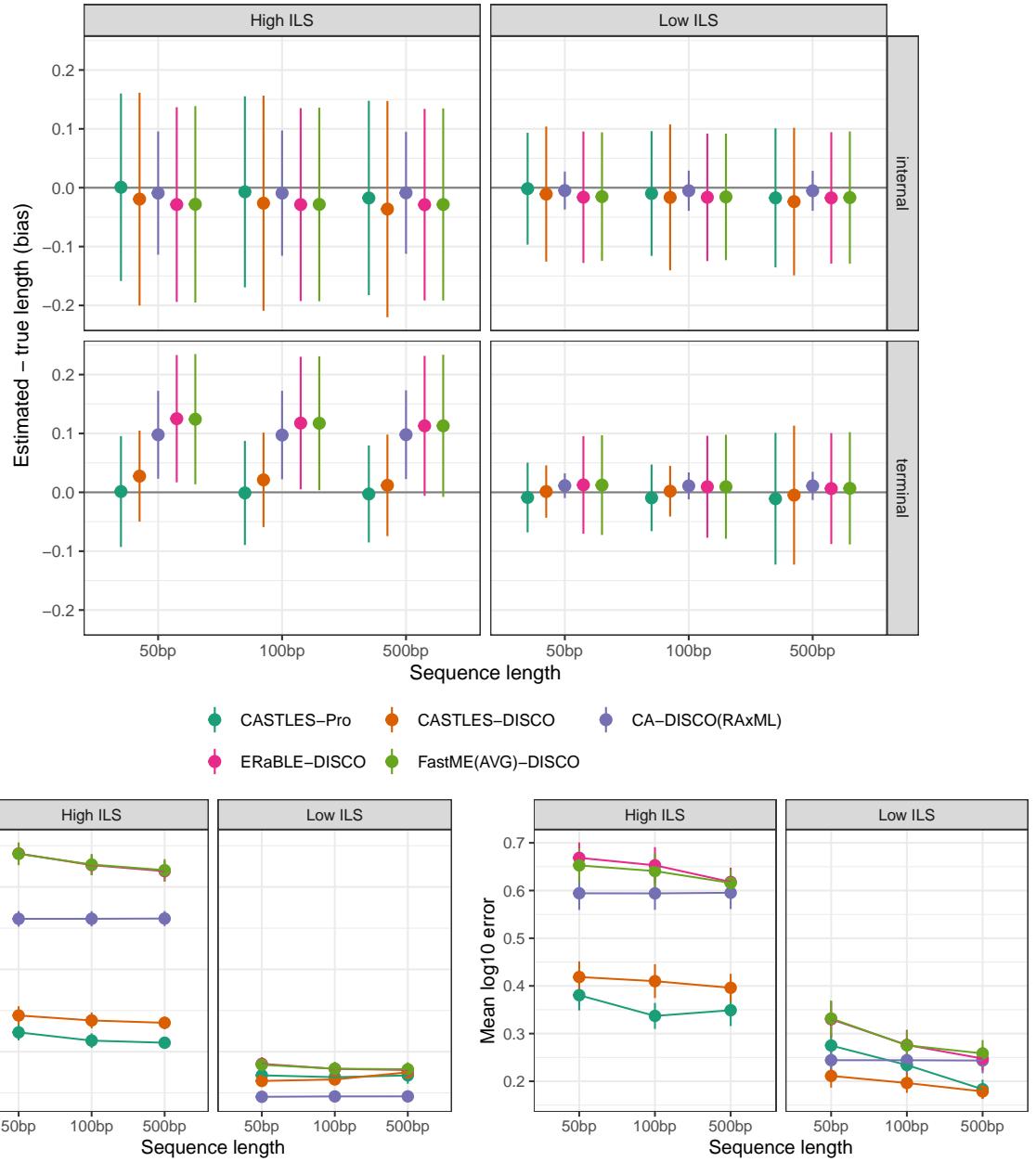
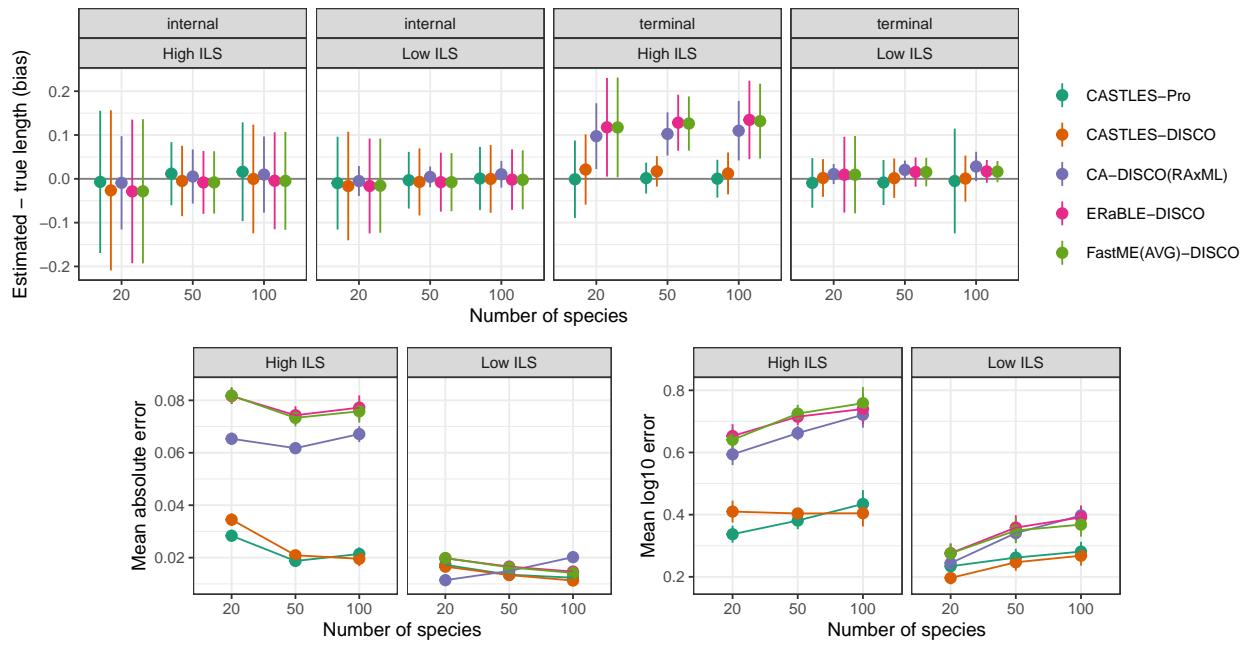


Figure 6.15: Bias, mean absolute error and mean log error for simulated GDL datasets for varying sequence lengths. The duplication rate is 10^{-12} with an equal loss rate. The average GTEE rates for the 50bp, 100bp and 500bp alignments for the low ILS condition are 52.5%, 41.5% and 18.4% respectively and for the high ILS condition are 55.7%, 42.6% and 19.2%. The number of taxa is 20, the number of genes is 1000 and the number of replicates is 10.

A) duplication rate: 10^{-12}



B) duplication rate: 5×10^{-10}

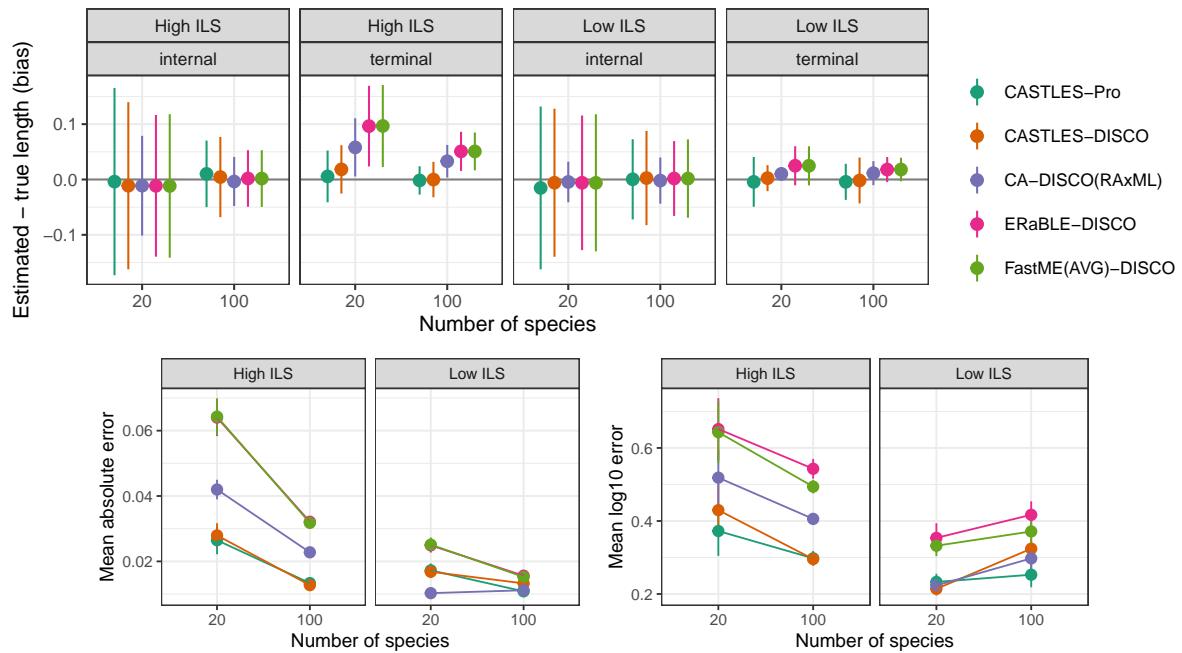


Figure 6.16: Bias, mean absolute error, and mean log error for simulated GDL datasets for varying numbers of species and level of ILS. The duplication rate is 10^{-12} (A) or 5×10^{-10} (B) with an equal loss rate. The gene trees are estimated from 100bp alignments with an average GTEE level that varies between 34.2% to 48.2% for different conditions. The number of genes is 1000, and the number of replicates is 10.

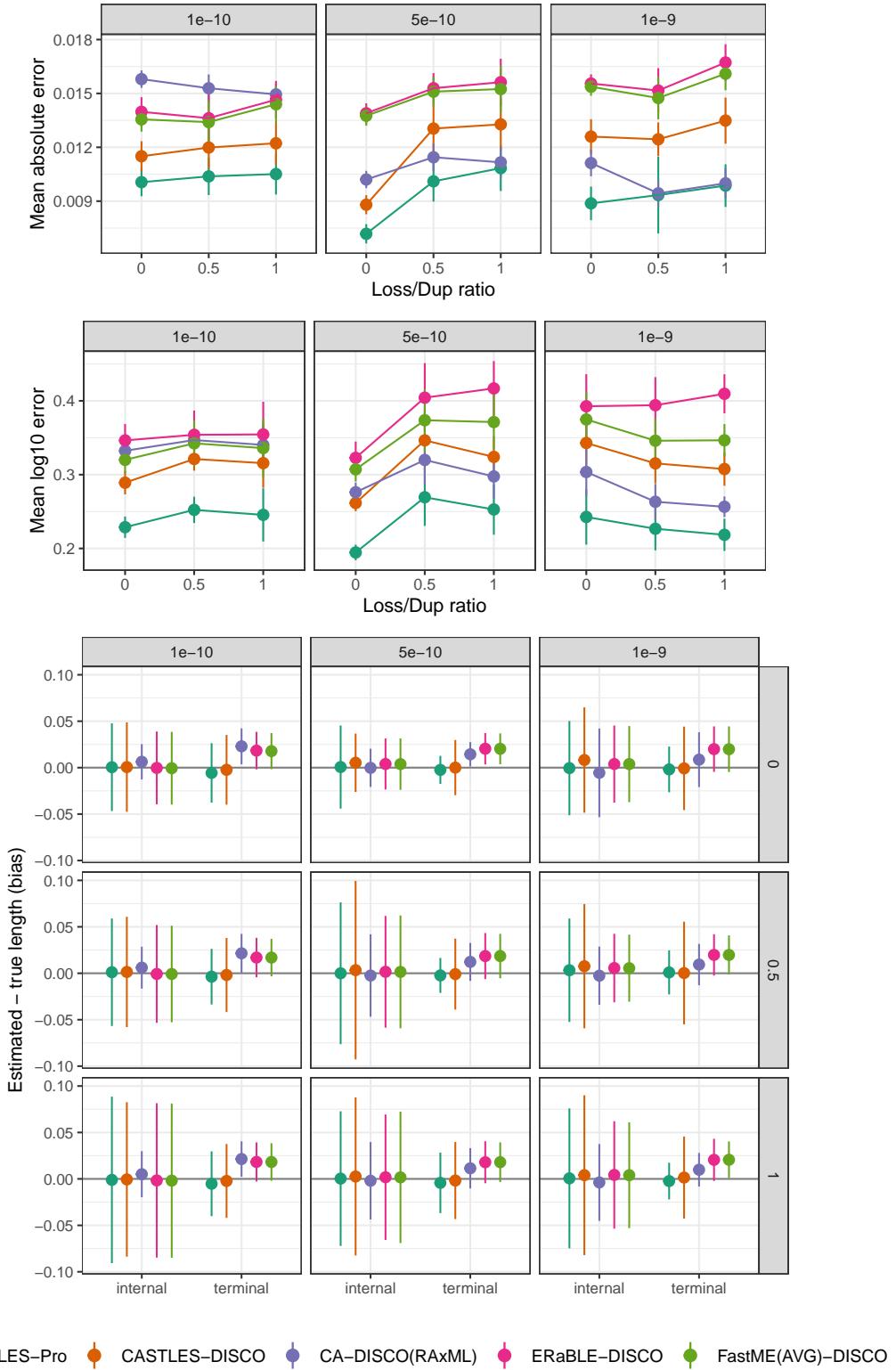


Figure 6.17: Mean log error, mean absolute error and bias on the 100-taxon [GDL](#) datasets for varying duplication/loss ratios. The gene trees are estimated from 100bp alignments with average [GTEE](#) levels that varies between 39.7% to 47.2%. The level of [ILS](#) varies between 19.1% to 26.6%. The number of [taxa](#) is 100, the number of genes is 1000 and the number of replicates is 10.

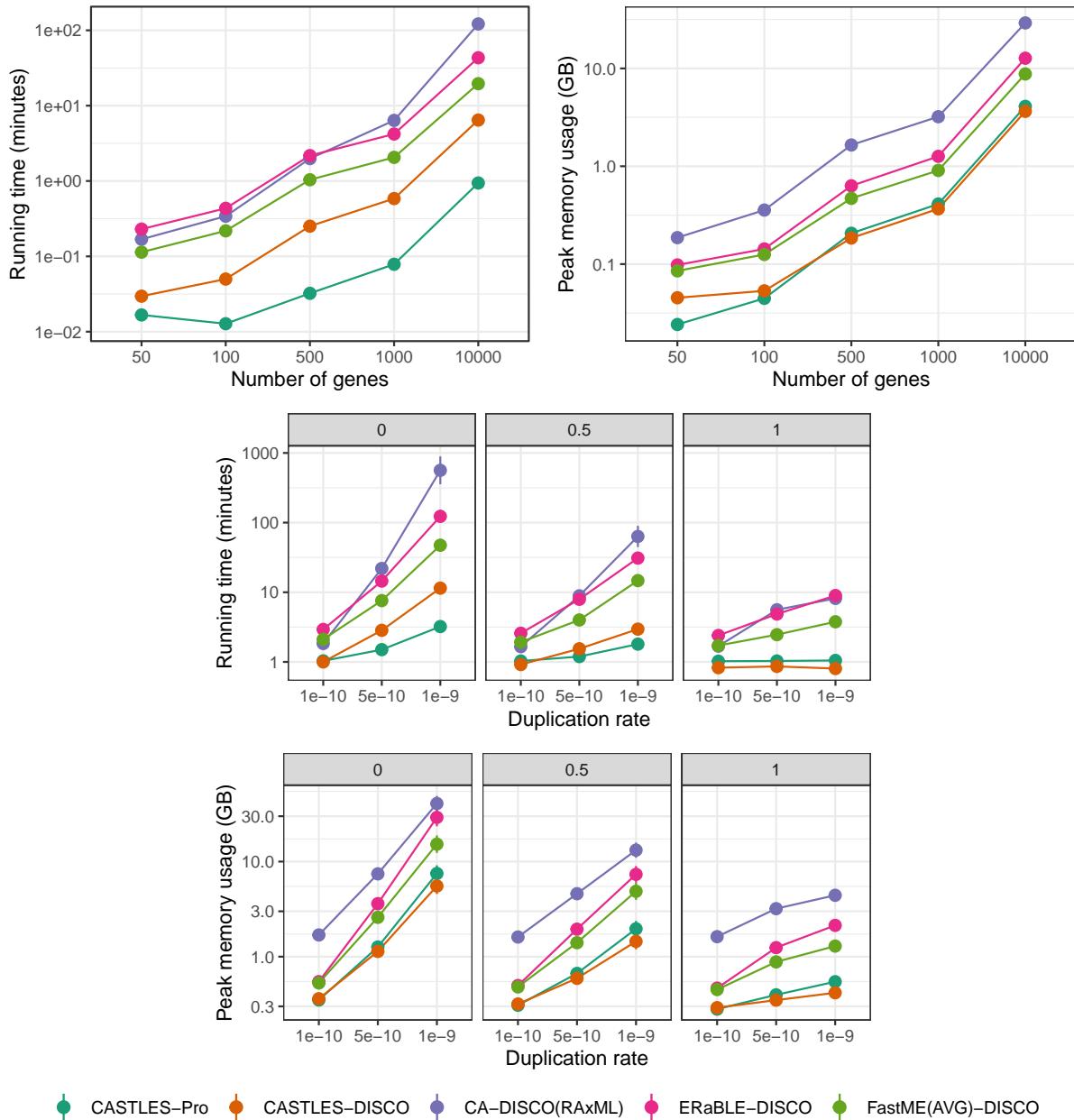


Figure 6.18: Runtime and peak memory usage of branch length estimation methods on 100-taxon GDL datasets for different number of genes and duplication rates. Gene trees are estimated from 100bp sequence alignments. The y-axes are shown in log scale. (top) The duplication rate is 5×10^{-10} with equal loss rate and the number of genes varies between 50 to 10,000. (bottom) The duplication rate varies between 10^{-10} to 10^{-9} and the number of genes is 1000. The panels show L/D ratio. The number of replicates is 10. The runtime does not include gene tree estimation or species tree topology estimation time, as all methods draw branch lengths on a fixed species tree topology.

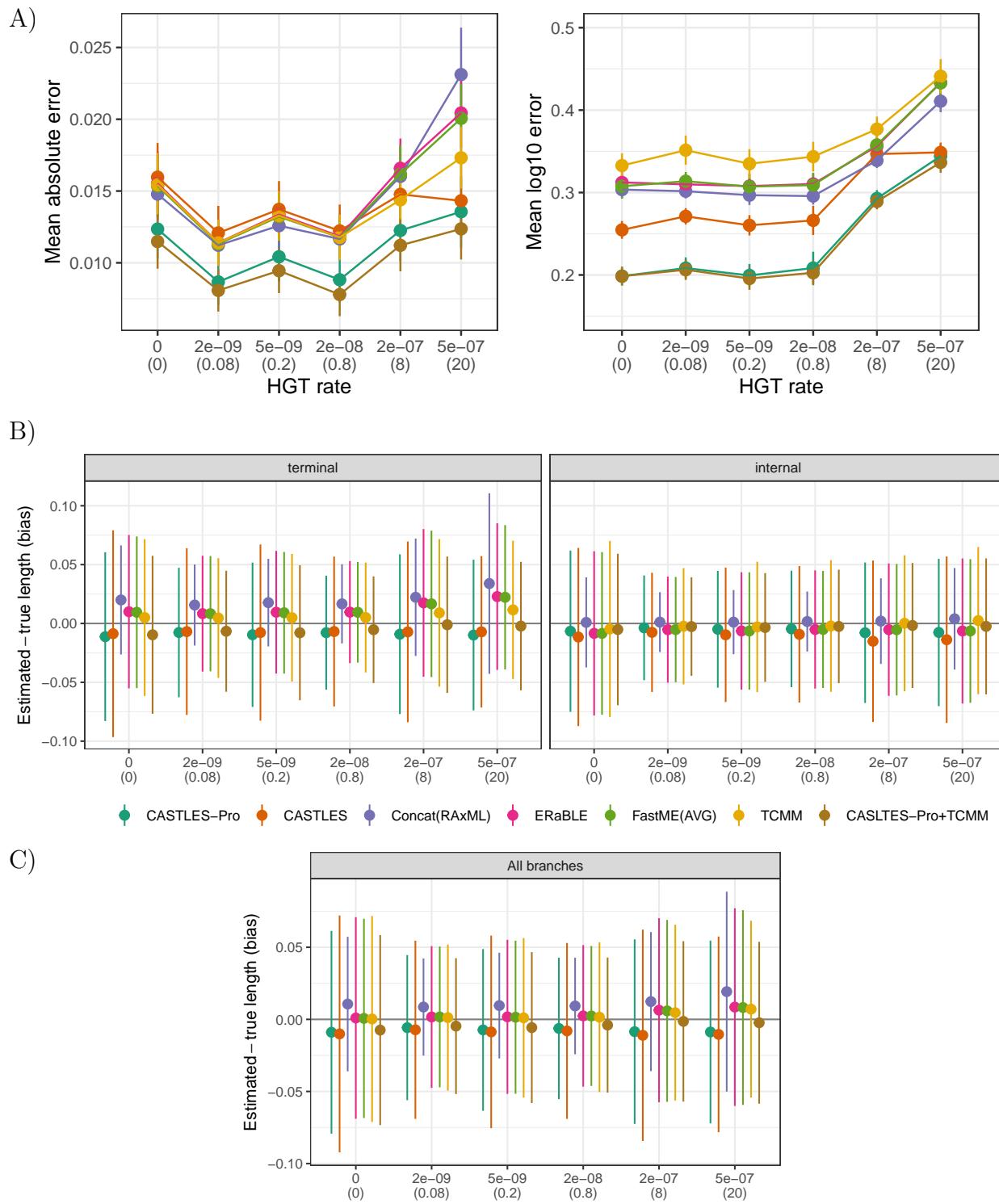


Figure 6.19: Mean absolute error, mean log error, and bias of branch length estimation methods on 50-taxon simulated HGT datasets. The x-axis indicates the rate of HGT and the expected number of HGT events per gene (in parentheses). The number of replicates is 50, and the number of genes is 1000.

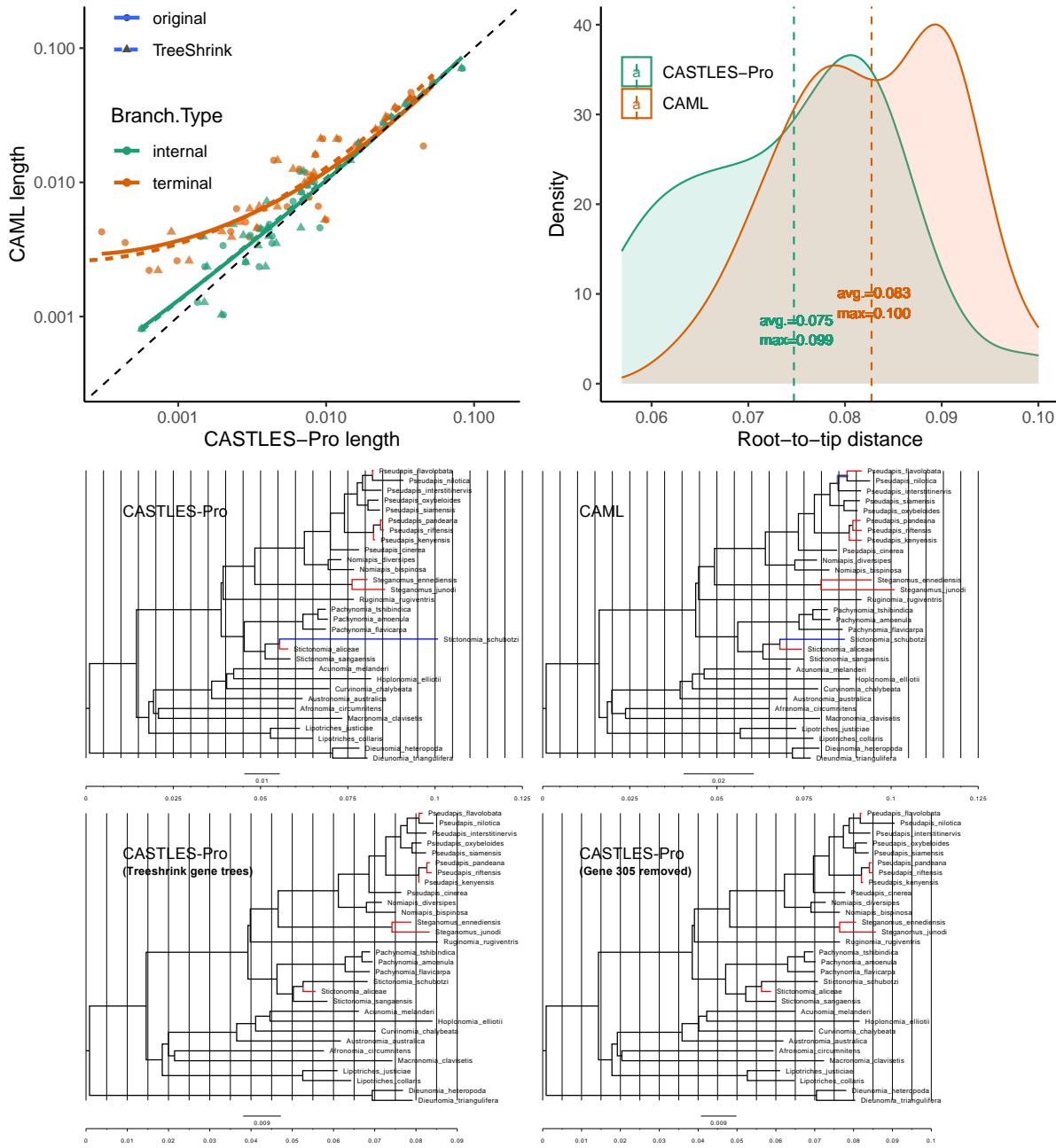


Figure 6.20: Comparison between the branch lengths of CASTLES-Pro (with or without TreeShrink) and CA-ML on the bees dataset of [102] after removing the *outgroup taxa Lasioglossum albipes* and *Dufourea novaeanglia*. The number of species (after removing the outgroups) is 30 and the number of gene trees is 853. We used the ASTRAL topology from the original study, and used concatenation and CASTLES-Pro to draw branch lengths on this topology. We set the average gene sequence length in CASTLES-Pro as 650bp, as the concatenated alignment had 867 loci with 576,041bp length in total, with an average loci length of 664.4. The branch lengths that are at least 2x longer or shorter in CASTLES-Pro tree compared to the concatenation tree are highlighted in blue and red respectively. The CASTLES-Pro tree estimates an unusually long branch length for taxon *Stictonomy schubotzi* due to an outlier gene, that is fixed when using TreeShrink gene trees or gene trees with the outlier gene 305 removed.

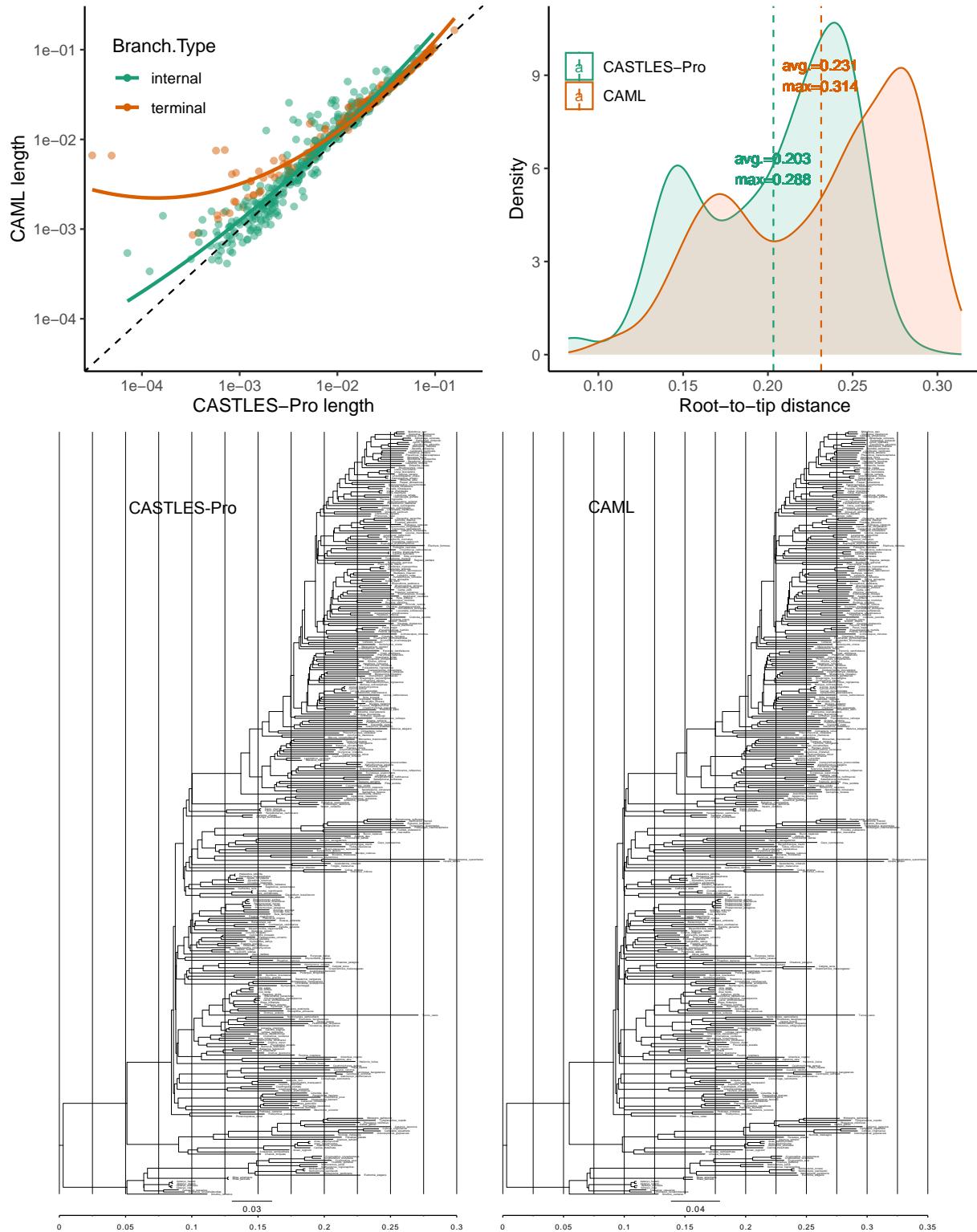


Figure 6.21: Comparison between branch lengths of CASTLES-Pro and CA-ML on the birds datasets of [328]. The number of species is 363 and the number of gene trees is 63,430. We used the ASTRAL topology from the original study with concatenation branch lengths, and estimated branch lengths with CASTLES-Pro on the same topology.

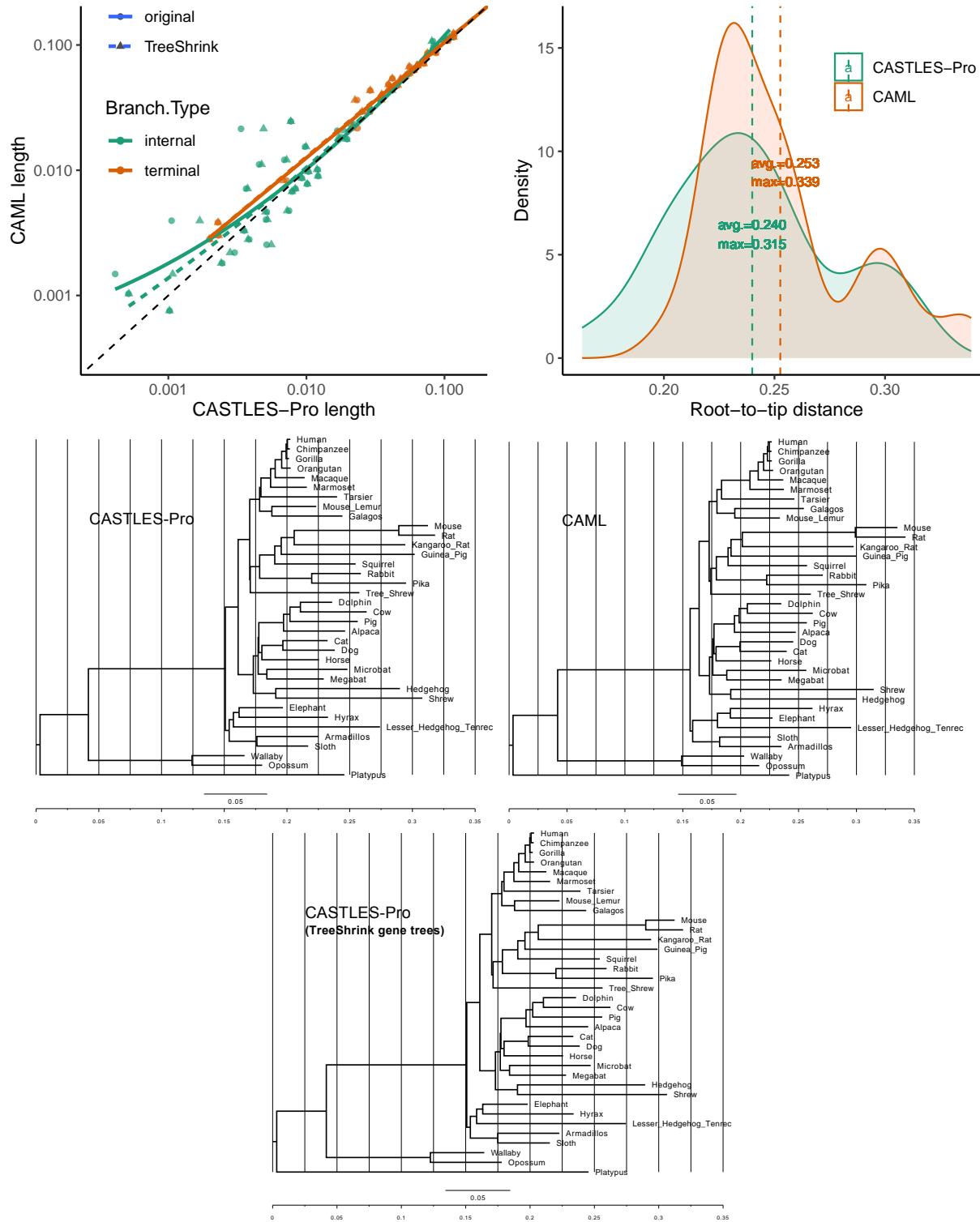


Figure 6.22: Comparison between branch lengths of CASTLES-Pro (with or without TreeShrink) and CA-ML on the mammals datasets of [250]. The number of species is 37 and the number of gene trees is 424. The average gene sequence length is 3099bp. We draw branch lengths using CASTLES-Pro and concatenation on an ASTRAL species tree topology. The branch lengths are drawn after removing the **outgroup taxa** *Chicken*.

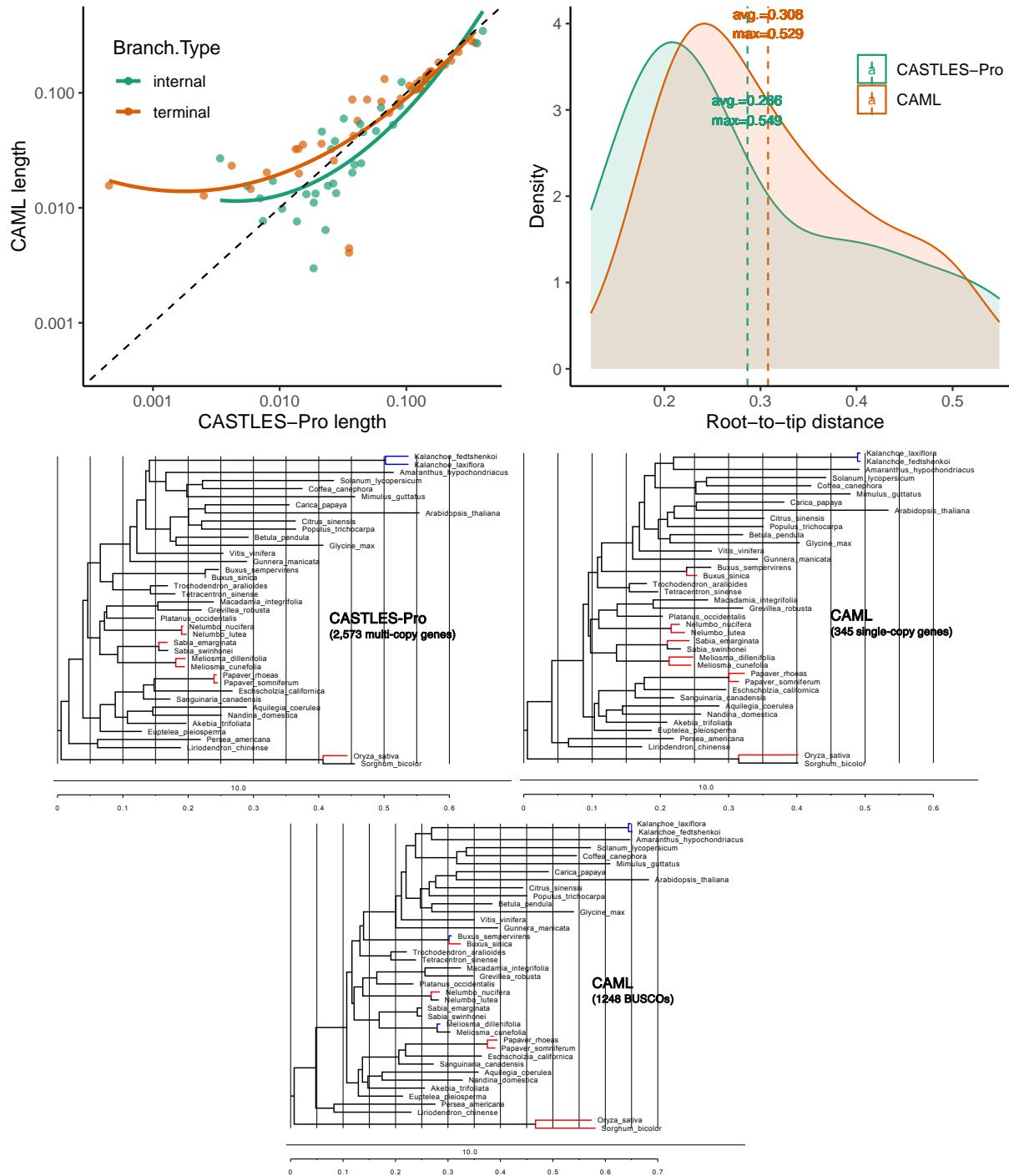


Figure 6.23: Comparison between branch lengths of CASTLES-Pro and CA-ML based on Angioseprm353 genes and BUSCO genes on the 40-taxon Eudicots datasets of [319] (after removing the *outgroup Amborella trichopoda*). The two CA-ML trees are estimated from the concatenation of 345 filtered Angioseprm353 single-copy genes or 1248 BUSCO single-copy genes, and CASTLES-Pro uses the 2,753 multi-copy gene family trees. The coalescent tree is estimated using ASTRAL-Pro2 and is different in 3 branches with the two concatenation trees (which are identical). The branch lengths that are at least 2x longer or shorter in CASTLES-Pro tree compared to the concatenation trees are highlighted in blue and red respectively.

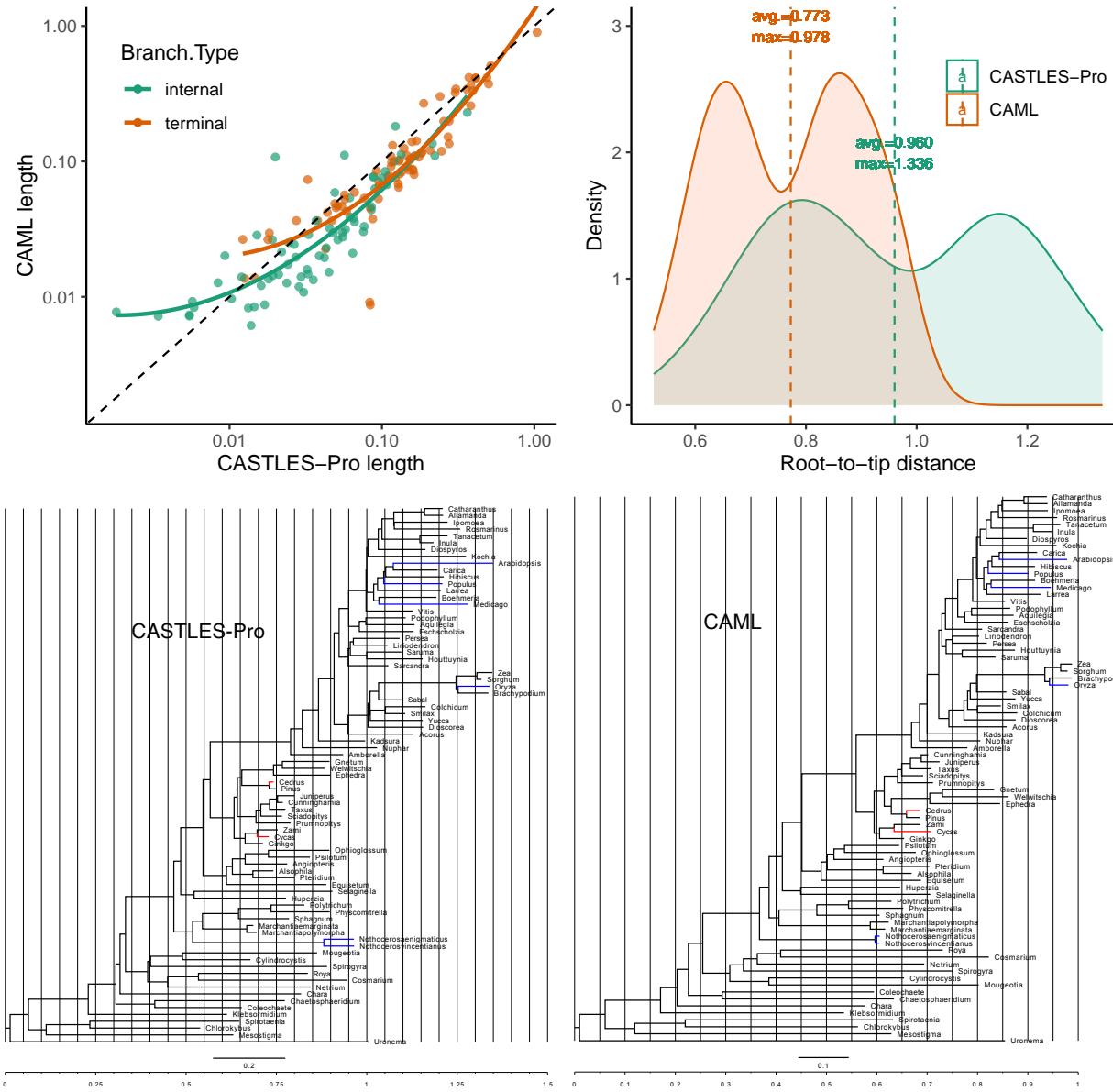


Figure 6.24: Comparison between the branch lengths of CASTLES-Pro and CA-ML on the plants dataset of [109]. The CA-ML tree is from the original study and was estimated from the concatenated alignment of 424 single-copy genes. In addition, we estimated a tree from the set of 9,610 multi-copy gene family trees from that study using ASTRAL-Pro2 and used CASTLES-Pro to draw branch lengths on this tree. The taxa in the two sets of genes are not entirely identical (see [109] for more detail). The RF distance between the CA-ML and ASTRAL-Pro trees on 79 shared taxa is 9.2%. The correlation is reported only for the shared branches between two trees. The branch lengths that are at least 2x longer or shorter in CASTLES-Pro tree compared to the concatenation tree are highlighted in blue and red respectively.

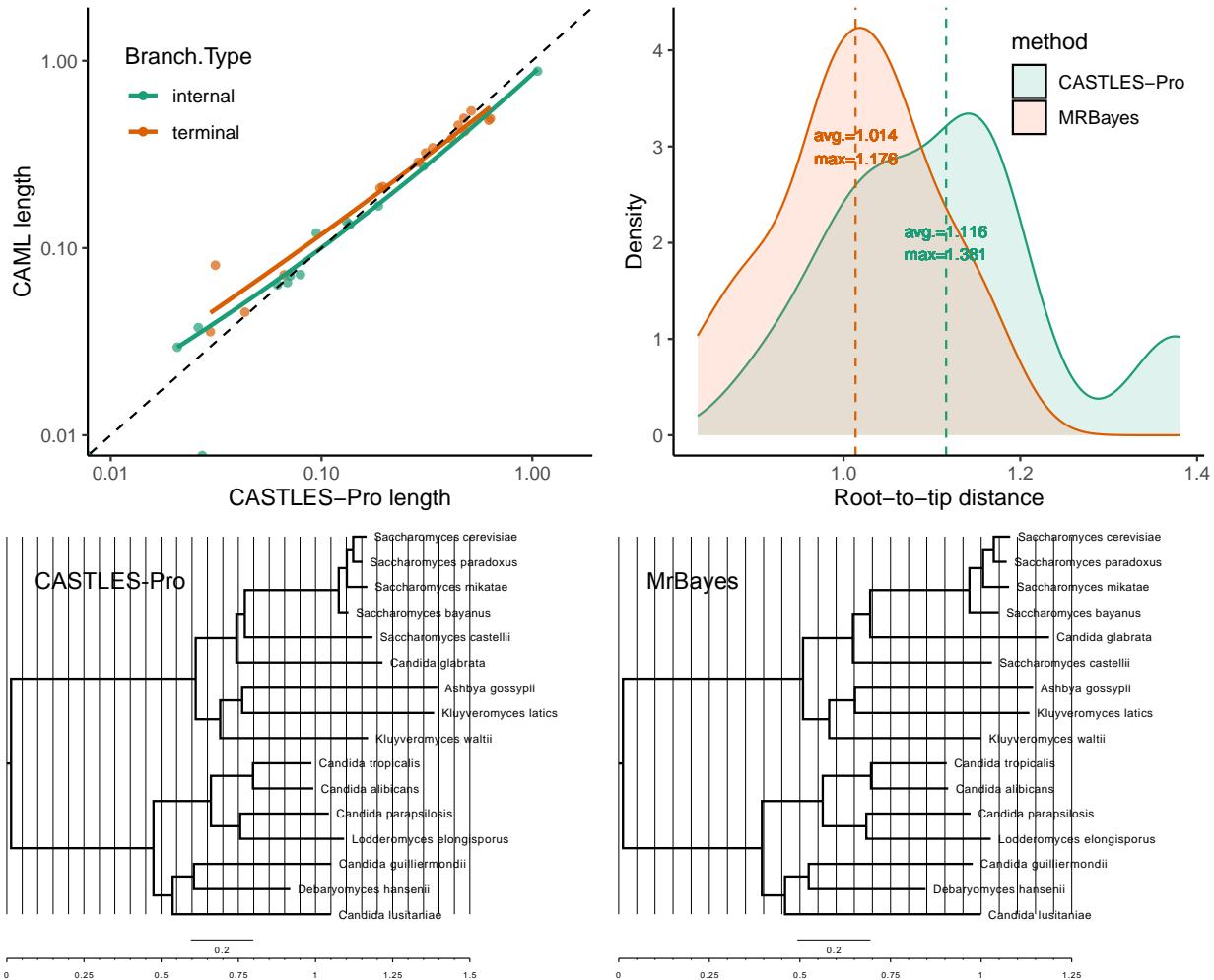


Figure 6.25: Comparison between branch lengths of CASTLES-Pro and concatenation with MrBayes on the fungal datasets of [329]. The number of species is 16. The original study had used MrBayes [333] on a concatenated alignment created by sampling 30,000 sites from 706 individual gene family orthologous peptide sequences. We used ASTRAL-Pro2 to estimate a species tree using all 7,180 gene family trees, and used CASTLES-Pro to estimate branch lengths on that tree. The two trees are different in one branch, with an RF distance of 7.6%.

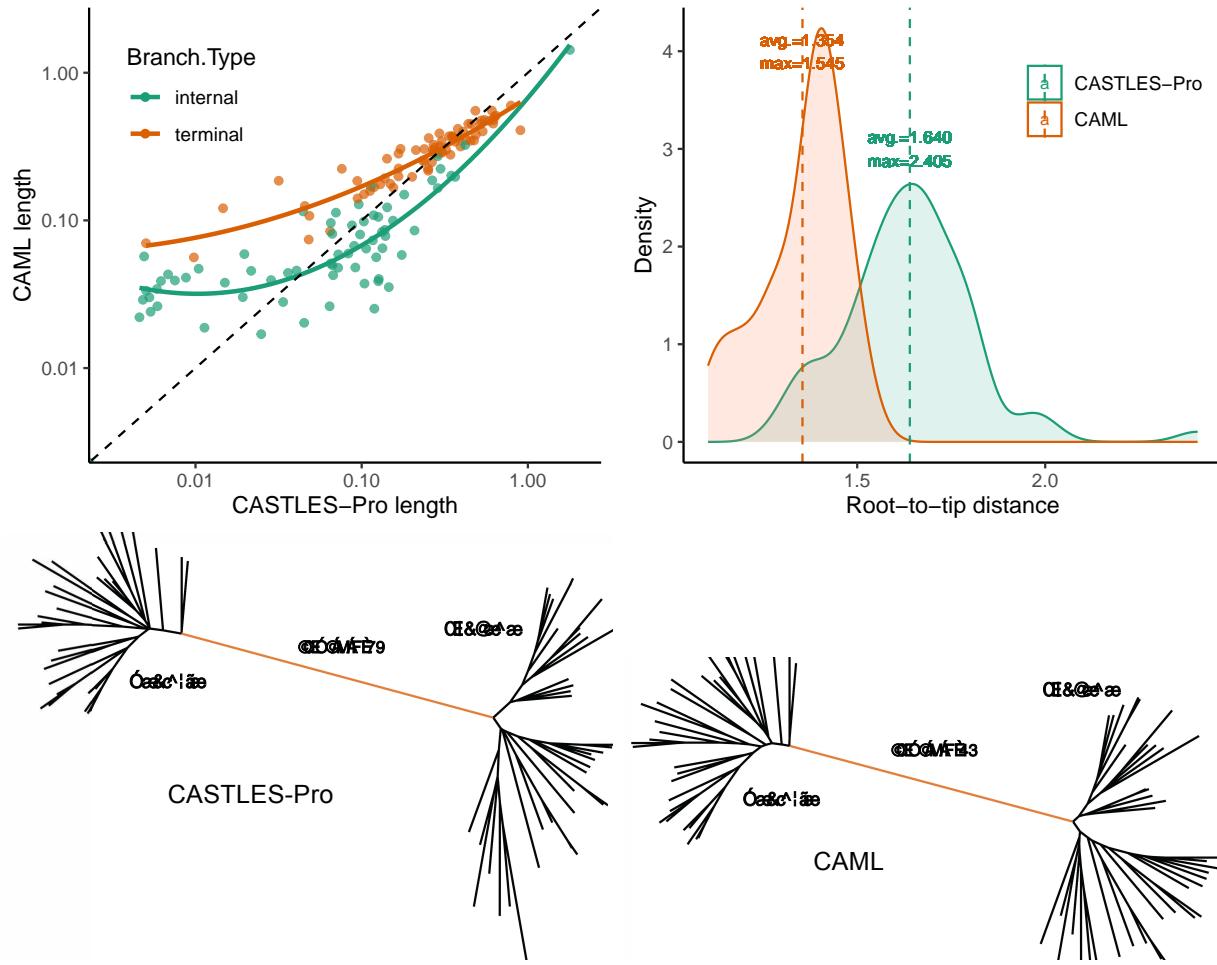


Figure 6.26: Comparison between branch lengths of CASTLES-Pro and CA-ML on the bacterial dataset with core genes from [330]. The number of species is 72 and the number of genes is 49. Both methods draw branch lengths on an ASTRAL tree topology. The branch colored in orange is the AB branch.

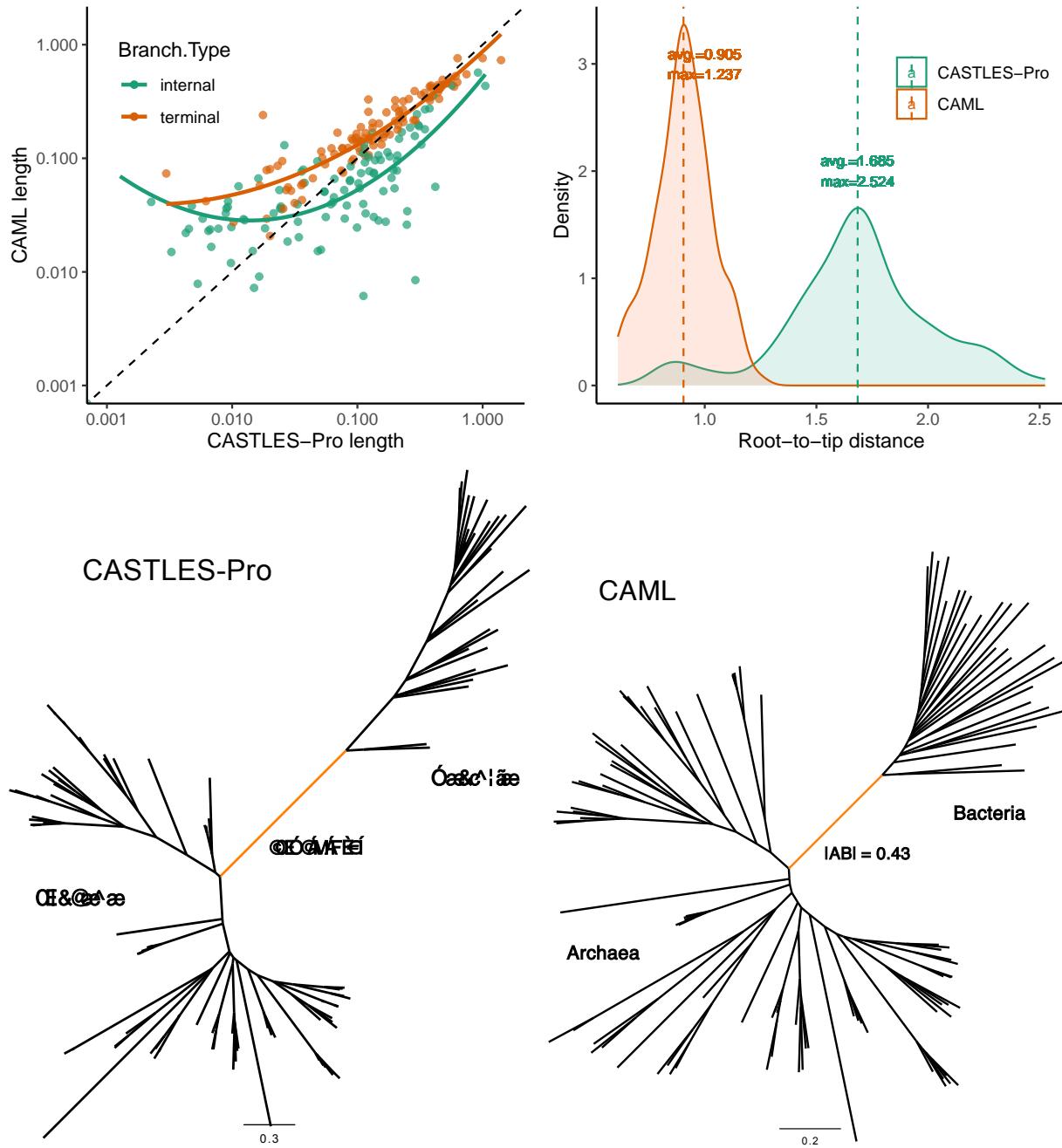


Figure 6.27: Comparison between branch lengths of CASTLES-Pro and CA-ML on the bacterial dataset with non-ribosomal genes from [331]. The number of species is 108 and the number of genes is 38. Both methods draw branch lengths on an ASTRAL tree topology. The total alignment length is 6,534bp. The branch colored in orange is the AB branch.

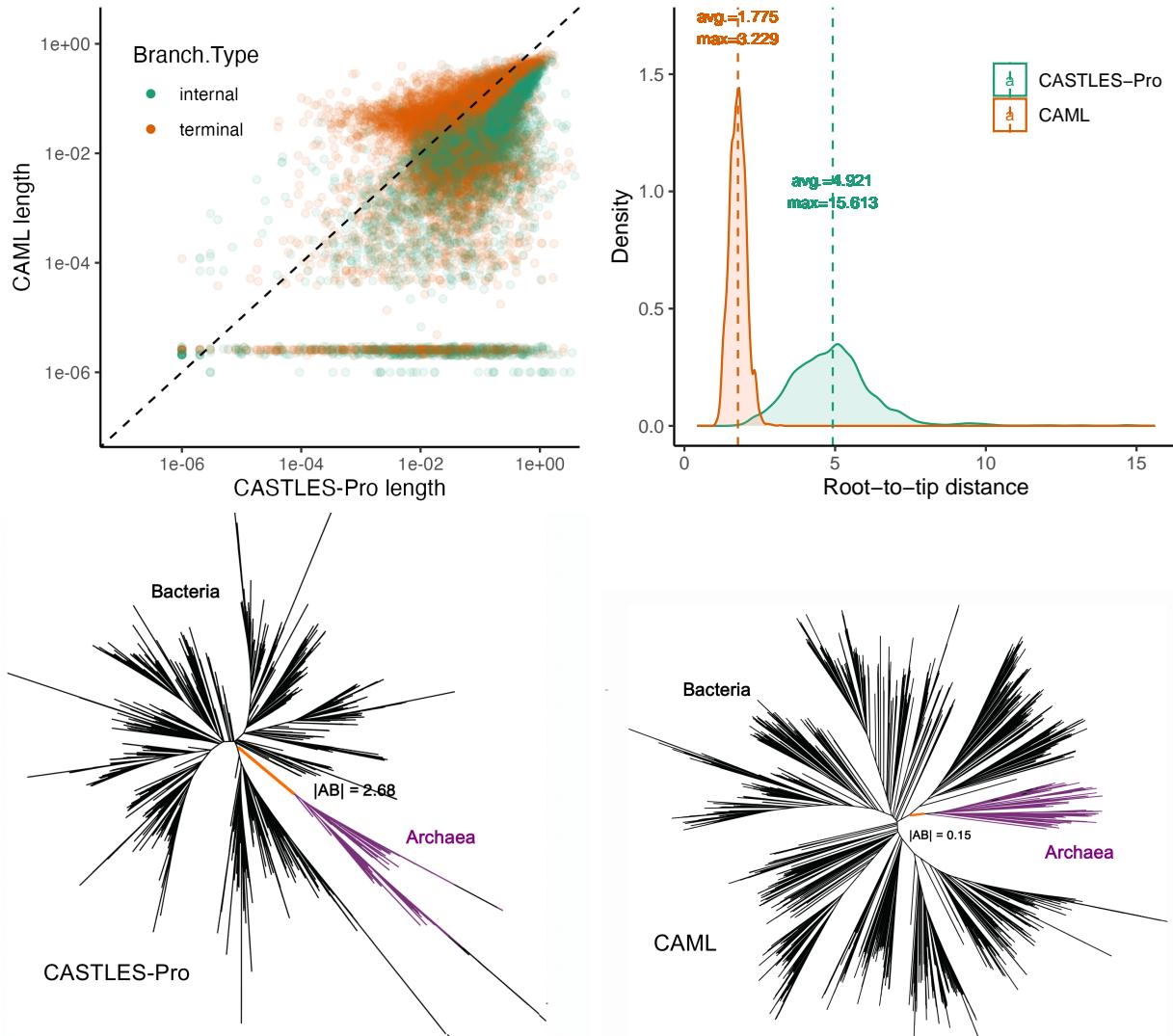


Figure 6.28: Comparison between branch lengths of CASTLES-Pro and CA-ML on the Web of Life (WoL) bacterial dataset from [60]. The number of species is 10,575 and the number of marker genes is 381. The original study had estimated an ASTRAL tree topology and furnished that with concatenation branch lengths, using a concatenated alignment of size 38kbp. This alignment was created by selecting 100 random sites from each gene sequence, and was 5X shorter than the full-length alignment that had 192k sites in total. We draw branch lengths on the same tree topology using CASTLES-Pro. The long tail of branches with $1e-6$ or $2e-6$ length in the concatenation tree correspond to no-event branches. The branch colored in orange is the AB branch.

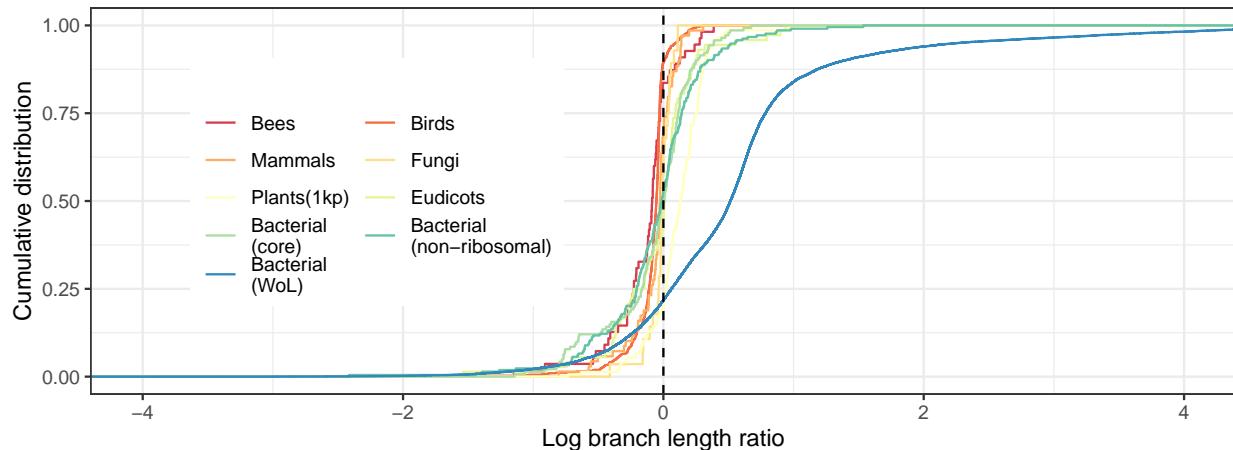


Figure 6.29: Cumulative distribution of the log ratio between the branch lengths produced by CASTLES-Pro and branch lengths produced by concatenation on nine biological datasets with different sources of gene tree [heterogeneity](#).

Table 6.4: Empirical statistics of the simulated [HGT](#) datasets. [AD](#) refers to average normalized [RF](#) distance between the model species tree and true gene trees, and [GTEE](#) refers to average normalized [RF](#) distance between true and estimated gene trees. The number of [taxa](#) is 51 and the number of genes is 1000.

| HGT rate | Expected number of HGT events per gene | AD | GTEE |
|--------------------|--------------------------------------------------------|--------------------|----------------------|
| 0 | 0 | 29.7% | 27.0% |
| 2×10^{-9} | 0.08 | 30.1% | 31.2% |
| 5×10^{-9} | 0.2 | 30.9% | 28.9% |
| 2×10^{-8} | 0.8 | 34.1% | 28.4% |
| 2×10^{-7} | 8 | 53.4% | 27.0% |
| 5×10^{-7} | 20 | 68.4% | 26.9% |

Table 6.5: Empirical statistics of the simulated [GDL](#) datasets. [AD](#) refers to average normalized [RF](#) distance between the [locus](#) tree and the true gene trees, and [GTEE](#) refers to average normalized [RF](#) distance between true and estimated gene trees. L/D refers to the ratio between loss and [duplication rates](#). The last column shows the average number of leaves in each gene family tree across the replicates. Default parameters: 1000 genes, 100bp sequence length, 10 replicates, 1 L/D ratio.

| Dup. rate | # taxa | AD | GTEE (500bp) | GTEE (100bp) | GTEE (50bp) | L/D ratio | # Leaves |
|---------------------|--------|--------------------|------------------------------|------------------------------|-----------------------------|-----------|----------|
| Low ILS | | | | | | | |
| 1×10^{-13} | 21 | 18.4% | — | 37.9% | — | 1 | 21.0 |
| 1×10^{-12} | 21 | 22.1% | 18.4% | 41.5% | 52.5% | 1 | 21.0 |
| 1×10^{-12} | 51 | 20.9% | — | 42.1% | — | 1 | 51.0 |
| 1×10^{-12} | 101 | 25.1% | — | 45.9% | — | 1 | 101.2 |
| 1×10^{-11} | 21 | 25.1% | — | 42.2% | — | 1 | 21.3 |
| 1×10^{-10} | 21 | 19.0% | — | 36.8% | — | 1 | 24.1 |
| 1×10^{-10} | 101 | 23.4% | — | 44.0% | — | 1 | 116.6 |
| 1×10^{-10} | 101 | 23.5% | — | 45.0% | — | 0.5 | 128.0 |
| 1×10^{-10} | 101 | 24.0% | — | 44.1% | — | 0 | 145.1 |
| 5×10^{-10} | 21 | 15.8% | — | 34.2% | — | 1 | 35.8 |
| 5×10^{-10} | 101 | 20.3% | 19.29% | 43.36% | 55.68% | 1 | 165.3 |
| 5×10^{-10} | 101 | 22.71% | — | 45.08% | — | 0.5 | 290.6 |
| 5×10^{-10} | 101 | 26.59% | — | 43.09% | — | 0 | 550.0 |
| 5×10^{-10} | 1001 | 23.49% | — | 44.4% | — | 1 | 1578.1 |
| 1×10^{-9} | 21 | 15.6% | — | 33.7% | — | 1 | 52.1 |
| 1×10^{-9} | 101 | 19.1% | — | 39.7% | — | 1 | 228.5 |
| 1×10^{-9} | 101 | 20.2% | — | 43.4% | — | 0.5 | 993.0 |
| 1×10^{-9} | 101 | 23.9% | — | 47.2% | — | 0 | 3727.8 |
| High ILS | | | | | | | |
| 1×10^{-13} | 21 | 69.3% | — | 42.2% | — | 1 | 21.0 |
| 1×10^{-12} | 21 | 67.4% | 19.2% | 42.6% | 55.7% | 1 | 21.0 |
| 1×10^{-12} | 51 | 75.7% | — | 45.8% | — | 1 | 51.2 |
| 1×10^{-12} | 101 | 78.4% | — | 48.2% | — | 1 | 101.2 |
| 1×10^{-11} | 21 | 67.0% | — | 41.0% | — | 1 | 21.3 |
| 1×10^{-10} | 21 | 64.5% | — | 39.0% | — | 1 | 24.2 |
| 5×10^{-10} | 21 | 54.5% | — | 39.5% | — | 1 | 36.2 |
| 5×10^{-10} | 101 | 50.0% | — | 43.9% | — | 1 | 170.1 |
| 1×10^{-9} | 21 | 44.2% | — | 38.3% | — | 1 | 47.9 |

Table 6.6: Empirical statistics of the biological datasets. ILS, GDL and HGT refer to the main source of gene tree discordance in each dataset.

| Study | Dataset type/species | # taxa | # single-copy genes | # multi-copy genes |
|------------|---------------------------------|--------|---------------------|--------------------|
| ILS | | | | |
| [328] | Neoavian birds | 363 | 63,430 | NA |
| [250] | Mammals | 37 | 424 | NA |
| [102] | Bees (subfamily Nomiinae) | 32 | 853 | NA |
| GDL | | | | |
| [109] | Plants (1kp) | 80 | 424 | 9,610 |
| [319] | Eudicots | 40 | 345 | 2,573 |
| [329] | Fungi | 16 | 706 | 7,180 |
| HGT | | | | |
| [330] | Bacterial (core genes) | 72 | 49 | NA |
| [331] | Bacterial (non-ribosomal genes) | 108 | 38 | NA |
| [60] | Bacterial (WoL) | 10,575 | 381 | NA |

Table 6.7: Runtime and peak memory usage of CASTLES-Pro on the biological datasets. Branch lengths are estimated on a fixed species tree topology. CASTLES-Pro uses multi-copy gene trees for the fungi, plants, and eudicots dataset and single-copy gene trees for the rest of the datasets.

| Dataset | # taxa | # genes | time (seconds) | peak memory (GB) |
|---------------------------------|--------|---------|----------------|------------------|
| Neoavian birds | 363 | 63,430 | 2034.64 | 54.83 |
| Bees (subfamily Nomiinae) | 32 | 853 | 1.85 | 0.06 |
| Mammals | 37 | 424 | 8.60 | 0.05 |
| Fungi | 16 | 7,180 | 1.60 | 0.20 |
| Plants (1kp) | 83 | 9,610 | 35.92 | 3.16 |
| Eudicots | 40 | 2,573 | 26.28 | 0.63 |
| Bacterial (core genes) | 72 | 49 | 1.72 | 0.01 |
| Bacterial (non-ribosomal genes) | 108 | 38 | 1.89 | 0.01 |
| Bacterial (WoL) | 10,575 | 381 | 3180.94 | 14.48 |

CHAPTER 7: COALESCENT-BASED BRANCH LENGTH ESTIMATION IMPROVES DATING OF SPECIES TREES

This chapter contains material that has appeared in the preprint “Y. Tabatabae, S. Claramunt, and S. Mirarab. (2025). coalescent-based branch length estimation improves dating of species trees, bioRxiv , <https://doi.org/10.1101/2025.02.25.640207>”[324]. The datasets and scripts used in this study are available at <https://github.com/ytabatabae/gls{coalescent}-based-dating>.

7.1 INTRODUCTION

Phylogenetic trees with branch lengths in units of time enable the study of evolutionary processes, such as species diversification, across geological times. The challenge [348] is to extrapolate from assumed dates for some nodes to the rest of the tree while modeling complex processes such as changes in substitution rates across lineages [140, 349]. A rich computational toolkit has been developed to deal with such complexities [350, 351, 352, 353]. Some of these methods take as input a tree with branch lengths measured in units of the expected number of substitutions per sequence site (SU), while others, typically Bayesian methods [352], co-estimate the tree topology and branch lengths in units of time (Figure 7.1A).

A challenge facing accurate dating is the impact of the heterogeneity of evolution across the genome. Beyond well-documented substitution rate differences across the genome [354, 355, 356], different regions can have different evolutionary histories [15]. In particular, each recombination-free region of the genome (a locus) has its own coalescence history, potentially including Incomplete lineage sorting (ILS) and topological differences across the genome. This stochastic process can be studied using the Multi-Species Coalescent (MSC) model [357]. Under this model that considers coalescence *times* and not just topologies, each locus has a *unique* coalescent history; although they can be arbitrarily close, no two loci have exactly the same history.

It has long been appreciated that there is a gap between gene divergence and species divergence times [217, 358] (Figure 7.1B), which we will refer to as gene/species (GS) gap in this chapter. Gene genealogies from any two species will coalesce some time *earlier* than the corresponding speciation event. Thus, gene divergences always predate species divergences, and this GS gap is in expectation proportional to the ancestral effective population size (Figure 7.1C); the expected length of the GS gap under the Wright-Fisher model for two lineages is a single coalescent unit (CU). Note that the GS gap does not impact all branches

equally due to several reasons, including varying population sizes. More significantly, the **terminal** branches in the **gene tree** are always longer than the **terminal** branches in the species tree. In contrast, **internal** branches in the **gene tree** may or may not be present in the species tree, and when present, they may be longer or shorter (Figure 7.1B). Thus, ignoring the **GS** gap will not just add random noise but rather will bias the dates: **terminal** branch lengths are expected to be systematically over-estimated.

Modeling the impact of heterogeneity due to **ILS** has been an active area of research, leading to many methods to infer species trees [307, 359]. These methods broadly fall into three categories: Bayesian methods that co-infer species trees and gene trees with parameters corresponding to divergence times and population sizes, two-step methods that first infer gene trees and then summarize them to obtain a species tree, and site-based methods that infer a species tree without inferring gene trees. While all these methods infer the species tree topology and are **statistically consistent** estimators, many have paid scarce attention to branch lengths. Concatenating loci as one input matrix, in contrast, is a **statistically inconsistent** estimator of topology [28]; it produces branch lengths but suffers from the **GS** gap bias mentioned earlier. We can expect that using **concatenation** for dating will introduce systematic biases, including over-estimation of **terminal** branch lengths.

This bias in dating **terminal** branches is not just a theoretical concern and has been observed empirically [70, 71, 360, 361, 362]. It can impact downstream analyses, such as the estimation of **lineage** diversification dynamics. In particular, the **GS** gap may contribute to the apparent pattern of early burst and slow-downs in diversification shown by many empirical phylogenies based on gene trees, as the long **terminal** branches result in lower estimates of speciation rates [363]. In addition, the overestimation of **terminal** branches may be behind the inability to detect extinction from molecular phylogenies, because extinction manifests itself as an excess of short **terminal** branches in reconstructed phylogenies [364].

Bayesian methods for inferring species trees under **MSC** [38, 223, 357] have always modeled branch lengths using explicit parameters and thus can produce correct species divergence times. In fact, inferring correct branch lengths has been one of their main advantages [38] over two-step approaches. Their main shortcoming is that they are far less scalable than the two-step approach, which has scaled to thousands of species [e.g., 60, 365, 366] and tens of thousands of loci [e.g., 328]. However, because **gene tree** summarization methods have not traditionally produced reliable branch lengths, users have needed to resort to **concatenation** for inferring branch lengths in substitution units [e.g., 250, 328] for dating. Using **concatenation** to get branch lengths makes the two-step approach prone to the negative effects of the **GS** gap. Finally, some likelihood-based methods [367, 368] allow site-based methods to infer divergence times and are more scalable than Bayesian methods; yet, these are not

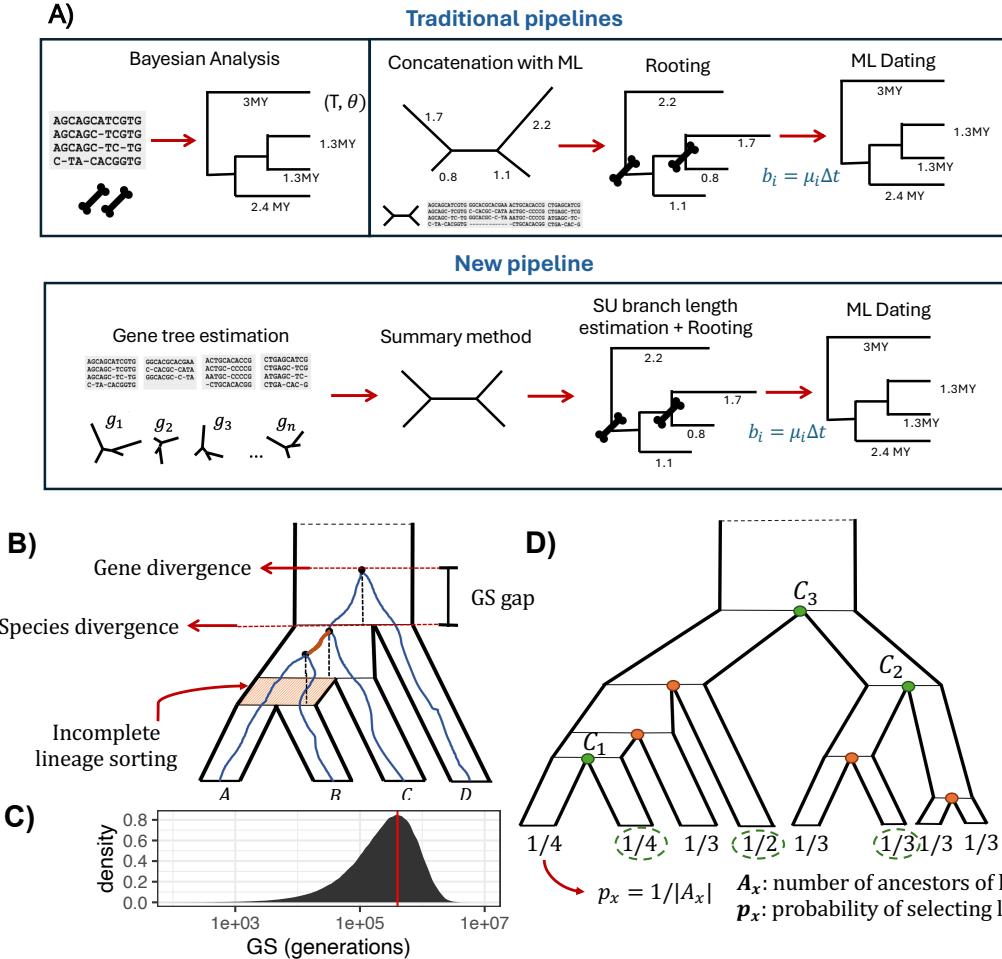


Figure 7.1: A) Traditional pipelines for estimating divergence times: Bayesian dating directly uses sequence data and calibration points and co-estimates the topology and branch lengths in time units; the two-step **concatenation** method estimates the branch lengths and/or topology using **concatenation** with maximum likelihood to produce a tree with branch lengths in units of number of substitutions per site, and then dates this tree by transforming these branch lengths. The scalable four-step pipeline introduced in this study uses discordance-aware methods for both topology and branch length estimation. B) Gene tree evolving inside a four-taxon species tree and the gap between gene divergence and species divergence (GS-gap). While **Incomplete lineage sorting** can increase the GS-gap, this gap exists even when there is no ILS. Due to the GS-gap, **terminal** branches in the gene trees are always longer than their corresponding branch in the species tree, but **internal** branches may be shorter or longer. C) The distribution of **GS** gap measured in units of generations is shown for **terminal** branches of 101 species, 50 replicates, with 1000 gene trees per replicate; $N_e = 4 \times 10^5$ (red line). D) Protocol for simulating calibration points on a species tree topology. Each leaf x is assigned a probability $p_x = 1/|A_x|$ where A_x is the number of nodes between x and the root (inclusive). Then, k **taxa** are selected randomly based on the probabilities assigned to them (in the figure, $k = 3$ and the three **taxa** with green circles are selected). Then for each selected **taxa** x , one of its A_x ancestors is selected with probability $1/A_x$ (the green **internal** nodes). With this process, k **internal** nodes are selected and the true divergence times of these nodes are used as calibration points.

as scalable as the existing two-step methods. Therefore, existing scalable approaches have not provided unbiased branch lengths and dates and approaches that do provide unbiased branch lengths are not scalable.

Recent advances hold the promise of enabling *scalable and unbiased* branch length estimation in both substitution and time units. In particular, our new method, CASTLES-Pro, can compute **SU** branch lengths by summarizing **gene tree** branches while accounting for the **GS** gap [305]. Simulation studies showed that it is far more accurate than **concatenation** (and similar methods) in the presence of sufficiently high **ILS** [75]. It reduced **terminal** branch lengths systematically compared to concatenation, as one would expect. At the same time, several scalable methods of dating exist, including those based on maximum likelihood (ML) [e.g., 80, 228, 369, 370]. Combining these advances, scalable branch length estimation and ML-based dating, promises a scalable and accurate approach for dating species trees that accounts for the **GS** gap.

In this chapter, we study a scalable four-step approach for producing dated phylogenomic trees relying on new methods for **SU** branch length estimation accounting for coalescence (Figure 7.1A). The accuracy and scalability of this pipeline have not been studied before. We show in extensive simulations that with enough **ILS**, this approach produces divergence times that are more accurate than using **concatenation** for branch lengths, as is commonly done. In addition, our proposed pipeline is far more scalable than pipelines based on concatenation, easily scaling to datasets with 10,000 species and 1000 genes when paired with fast ML-based dating methods. On real data, we show that the approach leads to substantially shorter **terminal** branches and eliminates artifactual inferences of sudden drops in speciation rates close to the present time.

7.2 MATERIAL AND METHODS

Our proposed four-step pipeline using **coalescent**-based branch lengths (**CoalBL**) is as follows:

- Step 1.** Infer gene trees for individual loci (e.g., using RAxML or IQ-TREE).
- Step 2.** Summarize the gene trees to get the species tree topology (e.g., using ASTRAL).
- Step 3.** Estimate the expected number of substitutions per site along the branches of the tree while accounting for the **GS** gap (e.g., using CASTLES-Pro).
- Step 4.** Date the tree using a scalable ML-based method (e.g., TreePL, LSD, MD-Cat, or wLogDate).

We used both simulated and biological datasets to compare the proposed four-step dating pipeline against the standard pipeline that uses [concatenation](#) for branch length estimation ([ConBL](#)). Note that the [ConBL](#) approach can infer the species tree topology using either [concatenation](#) or summary methods. Thus, it can share steps 1 and 2 with our proposed pipeline but uses [concatenation](#) for step 3; alternatively, it can replace steps 1–3 with a single ML analysis using concatenation. Both pipelines share step 4.

7.2.1 Study design: pipeline variations

We tested several variations of the proposed pipelines. We ensured [ConBL](#) and [CoalBL](#) are compared on the same underlying topology. For simulated datasets, we dated the true species tree topology for both pipelines, thus eliminating step 2. On biological datasets, we performed dating on topologies estimated using ASTRAL or concatenation. The species trees are rooted before dating, using either the true rooting (in simulations) or outgroups (for biological datasets). For both simulated and biological data, we used estimated gene trees available from prior studies, therefore eliminating the need to perform step 1.

The main difference between the two pipelines is the [SU](#) branch length calculation (step 3). For step 3 of [CoalBL](#), we used gene trees as input to CASTLES-Pro, which is a scalable method that explicitly models the [GS](#) bias. For [ConBL](#), we used RAxML [25] applied on a fixed tree topology to estimate [SU](#) branch lengths. We also tested the Bayesian molecular dating using MCMCTree [225] on a subset of our datasets. The input to MCMCTree is the tree topology (without branch lengths) and the sequence data; it bypasses step 3 and directly uses sequence data to date the topology. Unlike the ML-based methods that use fixed calibration points or min-max bounds, MCMCTree uses soft fossil calibrations with flexible probability distributions to describe the uncertainty in fossil ages. We used MCMCTree with approximate likelihood calculation [371, 372] to speed up the analyses (suggested for large inputs). All parameters of the MCMCTree analyses in simulations were taken from [328] study, with the exception of burnin, sample frequency, and the number of samples in the [MCMC](#) sampling that were set to 50,000, 100, and 50,000, respectively.

For dating (step 4), we examined four ML dating methods: LSD [373], which can be considered ML under a strict clock with Gaussian estimation noise (run in the QPD* mode and minimum branch length set to 0.001); wLogDate [374], which is ML under a LogNormal rate model; MD-Cat [80], which is ML under a categorical model of rates with k (default 50) rates; and TreePL [228], which uses a likelihood framework with penalties for rate divergences [375]. These methods take as input a tree topology with branch lengths in substitution units and (optionally) a set of calibration points. For dating methods that need sequence length

as input (LSD and TreePL), we specified the sequence length as the total number of [sites](#) across all genes.

In total, we compared nine dating pipelines (four ML-based dating methods with [CoalBL](#) and [ConBL](#), as well as [MCMCTree](#)). For both [ConBL](#) and [MCMCTree](#), we used [unpartitioned](#) analysis, where all gene sequences were concatenated and passed as a single partition to the method. Further details about each method and how we ran them are provided in Section [7.5](#).

7.2.2 Simulated datasets

We examined three simulated datasets with [gene tree](#) discordance due to [ILS](#) that were generated using the simulation software [SimPhy](#) [272]. These datasets have model conditions that vary in terms of the level of [ILS](#), deviation from the clock, number of species and genes, and the amount of [gene tree estimation error \(GTEE\)](#). To evaluate the accuracy of dating, we need to have ground-truth species tree topologies in the unit of time. [SimPhy](#) generates true species trees in the unit of the number of generations and true gene trees in substitution units. To convert the species trees to units of million years, we multiplied the generation unit lengths by 5 (assumed generation time) and divide them by 10^6 . We created model conditions with different numbers of calibration points, ranging from zero to 3, 5, and 10 calibration points for small datasets and up to 50 calibrations for large datasets. When the ML-based dating methods are used without any input calibration points, they produce a unit-length [ultrametric](#) tree ([MCMCTree](#) needs calibrations). While some dating methods can use calibrations in the form of min-max bounds or probability distributions, we specified the calibration points as fixed values everywhere.

To select k calibration points on the species tree, we first assigned a probability to each [terminal](#) node (leaf) inversely proportional to the number of nodes from the root to that leaf; we then randomly sampled k leaves based on these probabilities. For each selected leaf, we picked one node along the subtending [lineage](#) uniformly at random but avoided the ones already selected, if any (Figure [7.1D](#)). This procedure helps distribute the calibration nodes widely across the tree; if sampled uniformly at random, most calibration nodes would be placed near the [tips](#) and in species-rich clades. We then used the true divergence times of the sampled nodes as calibration points. In some of the experiments, we fixed one calibration point on the root of the tree and selected the $k - 1$ other calibration points using the approach described above. We refer to these experiments as “root-fixed”, as opposed to “root-unfixed”. For the root-unfixed experiments, we set a soft upper bound for the root age for [MCMCTree](#) as $10 + T$ where T is the true root age. Unless otherwise noted, results

are for the *root-fixed* experiments.

We measure the error of dating using mean absolute error, bias, RMSE, and mean absolute logarithmic error, averaged across all species tree branches compared to the true tree. Absolute error and bias emphasize longer branches more than short branches, and RMSE and log error emphasize shorter branches more. Since the higher levels of ILS correlate with shorter tree heights, we report branch length error normalized by the species tree height (including the `outgroup` edge, when present). In addition to branch length error, we report errors in node ages, using the same four metrics. We also report the time of the most recent common ancestor (tMRCA) for the *root-unfixed* experiments. Finally, we report *treeness* (i.e., sum of `internal` branch lengths divided by sum of all branch lengths) of the dated trees. While neither CASTLES-Pro nor ConBL produce zero or negative SU branch lengths, some of the dating methods (in particular, LSD) can produce zero-length branches in time units. For all methods, we replace negative and zero lengths with the small pseudo-count of 10^{-6} before calculating error metrics. We measure the level of ILS in terms of average normalized Robinson–Foulds (RF) [83] distance between the model species tree and true gene trees (referred to as average distance or AD) that can take values between 0% to 100%.

30-taxon dataset. We used a 30-taxon dataset from [241] with six model conditions, varying in terms of deviation from the clock and inclusion of an outgroup. The gene trees in this dataset were estimated using FastTree-2 [273] from alignments with gene sequence lengths that were drawn from a log-normal distribution (average: 495bp). The ILS level was heterogeneous across replicates, with a mean of 46%. The average GTEE level is 38% across all model conditions and replicates. This dataset has three levels of deviation from the clock, specified with parameter α of the gamma distribution for branch rate heterogeneity, set to 5 (low), 1.5 (medium), or 0.15 (high). The number of genes in this dataset is 500 and the number of replicates is 100. For this dataset, we created conditions with 0, 3, or 5 calibration points, both *root-fixed* and *root-unfixed*. The total number of model conditions for this dataset is 30 (2 `outgroup` settings \times 3 α values \times 5 calibration settings).

100-taxon dataset. We used the 101-taxon (100 ingroup and 1 outgroup) dataset from [51] (50 replicates) with model conditions that vary in terms of the level of gene tree estimation error (GTEE). The four model conditions have GTEE levels of 23%, 31%, 42%, and 55%, respectively, corresponding to gene trees estimated using FastTree-2 from sequence alignments of length 1600bp, 800bp, 400bp, and 200bp, respectively. The level of ILS varied between 30% to 58% across replicates (mean: 46%). We used 0, 3, or 10 calibration points both as *root-fixed* and *root-unfixed* (for 3 and 10). The default number of genes in this

dataset was 1000, which we also subsampled to get 500, 200, and 50 genes for the *root-fixed* experiments. In total, we created 56 different model conditions for this dataset.

Large dataset. To evaluate scalability, we generated a new dataset with large trees with up to 10,000 species using a modified version of the simulation software SimPhy created by [74] that creates species trees with **SU** and **CU** branch lengths. Species trees are generated using a birth-death process, with birth and death rates sampled from log-uniform distributions. To simulate heterogenous levels of **ILS**, we draw the **effective population size** from a uniform distribution between 20,000 and 2,000,000. We simulated conditions with 50, 100, 200, 500, 1K, 2K, 5K, and 10K ingroups and 1 outgroup. We simulated 500bp gene sequence alignments using INDELible [327] under the **GTR** + Γ model with model parameters estimated from three different real datasets (identical to parameters used in [376]) and then used FastTree-2 to estimate gene trees under the **GTR** model. The **ILS** level is highly heterogeneous across replicates (minimum: 0%, maximum: 92.35%), with an average of 47.26%, and the **GTEE** level is, on average, 46.47% across all model conditions. For a model condition with n taxa, we simulated $\sqrt{n}/2$ calibration points, resulting in 3, 5, 7, 11, 15, 22, 35, and 50 calibrations for the eight tree sizes, respectively. This dataset has 20 replicates for each model condition and 1000 genes in each replicate. Table ?? summarize further information about this dataset. We ran all methods given 64GB of memory across 16 cores. For this scalability experiment, we only used TreePL, which was one of the fastest and most accurate dating methods.

7.2.3 Biological datasets

We examine two avian biological datasets representing **rapid radiations** at two different timescales.

We studied a dataset of modern birds (Neornithes) by [328], including 363 bird species and 63,430 genes. The main species tree used in the original study was estimated using ASTRAL and furnished with **SU** branch lengths using RAxML run on the **concatenation** of all 63K genes. Dating analyses in the original study were performed using MCMCTree with approximate likelihood calculation and a sequential-subtree approach using 34 calibration points. The tree was divided into 11 subtrees (with 19–42 **taxa** in each), and an initial dating analysis was performed on a backbone tree with 56 **taxa** containing two representatives of each of the 11 subtrees. The subtrees were then attached to the backbone tree to get a time tree on all 363 taxa. The analyses were done using a subset of the 10,494 most clock-like loci (i.e., those with the lowest coefficient of variation in **root-to-tip** distances).

We furnished the main ASTRAL topology from the original study with CASTLES-Pro branch lengths (using all 63K loci) and used three ML-based dating methods, MD-Cat, TreePL and wLogDate to estimate divergence times. While MCMCTree can take [calibration densities](#) in the form of probability distributions as input, the ML-based methods either work with fixed calibration bounds (MD-Cat and wLogDate) or min-max bounds (TreePL). Due to this difference, we cannot directly compare the ML-based trees to the output of MCMCTree, and mainly focus on comparing the dated trees based on [ConBL](#) or [CoalBL](#) branch lengths. To generate fixed calibration points and minimum-maximum bounds for the ML-based dating methods, we used the 0%, 50%, and 95% quantiles of the probability distributions of each fossil calibration density used in the original MCMCTree analysis. We used the median values (50% quantiles) as calibrations for MD-Cat and wLogDate that use fixed calibration points, and both median and min-max values (0%-95% quantiles) for TreePL that use min-max bounds. We examine variations in the age of orders, families, and genera, the correlation between [terminal](#) and [internal](#) branches, and changes in substitution rates across time.

We also study a suboscines dataset of [365], including 1940 individuals from 1283 species (98.2% of all suboscines species, which present the largest bird radiation in the tropics) with 2389 orthologous loci. [365] used this dataset to study the diversification of suboscines through time, and found that diversification rates (including speciation and [extinction rates](#)) have been relatively constant across the history of the group (between 40 to 51 million years) except for a dramatic drop in the past 2 million years. They suggested that this drop could have resulted from incomplete sampling or unsorted ancestral polymorphism. They also attribute this drop to challenges in studying diversification histories in extant [taxa](#) (terminal branches). Here, we re-analyze this dataset to see if this drop can be an artifact of a potential overestimation bias in branch lengths estimated using [ConBL](#) and whether CASTLES-Pro can reduce or eliminate this problem.

The original study [365] had used both [concatenation](#) (with EXaML) and ASTRAL to estimate a species tree topology, but since the [concatenation](#) topology better matched previously known relationships, the [concatenation](#) tree was used for the final diversification analyses. The species trees were dated using TreePL, with four calibration points. We used CASTLES-Pro to estimate branch lengths on the ASTRAL and [concatenation](#) trees and used [ConBL](#) as provided by the original paper [365] for both trees. We then used TreePL with the same calibration points, smoothing parameters and cross-validation method as the original study to date these two species trees. We then repeated the diversification-through-time analyses performed in [365] using a Bayesian [MCMC](#) analysis on the different species tree topologies furnished with [ConBL](#) or [CoalBL](#) branch lengths (in total four trees). We study

the trend in the speciation rates and extinction rates by performing CoMet analysis using the TESS library [377] where the diversification rates are estimated using reversible-jump MCMC. We remove the outgroup before the CoMet analysis and use automatic empirical hyperprior search and set the maximum number of iterations to 1,000,000. We also examine differences between terminal and internal branch lengths as well as lineage-through-time plots.

7.3 RESULTS

7.3.1 Simulation studies

30-taxon dataset. On this dataset, using the CoalBL pipeline instead of ConBL decreases the mean absolute error of branch lengths in all conditions when the level of ILS is at least moderate (Figure 7.2 and Figure 7.7). As the level of ILS increases, the height-normalized error for all methods increases, but more slowly for CoalBL pipelines. For the lowest level of ILS (< 0.25 AD), ConBL pipelines have a small advantage, whereas, for remaining ILS levels, pipelines that use CoalBL are always advantageous, regardless of the dating method. As expected, more calibration points lead to lower errors for all methods; however, with high ILS, CoalBL with fewer calibrations is better than ConBL with more. Increased deviations from a strict molecular clock increase the error for all methods, and the positive impact of CoalBL is more substantial for conditions with low deviation from the clock (Figure 7.2a).

The reduction in error is mostly due to better terminal branch lengths (Figure 7.8). The internal branch lengths change less and improve only in the two conditions with the highest ILS and the condition with the lowest deviation from the clock. Using ConBL leads to an overestimation bias for terminal branches and a smaller underestimation bias for internal branches, and this bias increases as the level of ILS increases (Figure 7.2b). Because of these biases, treeness is underestimated in all pipelines based on ConBL (Figure 7.2d) and for MCMCTree (Figure 7.9), especially for high levels of ILS. The use of CoalBL reduces the underestimation bias of internal branches and eliminates the overestimation for terminal ones, often causing a relatively small underestimation bias for terminal branches (Figure 7.2b). The CoalBL pipeline leads to relatively accurate levels of treeness, with a small overestimation bias for high deviation from the clock or ILS (Figure 7.9). Deviation from the clock has very little to no impact on the treeness when using ConBL or MCMCTree; CoalBL pipelines change from no bias for conditions with low clock deviation to a small overestimation bias in conditions with high deviation.

Similar trends in bias are observed with or without an outgroup when the root is fixed in

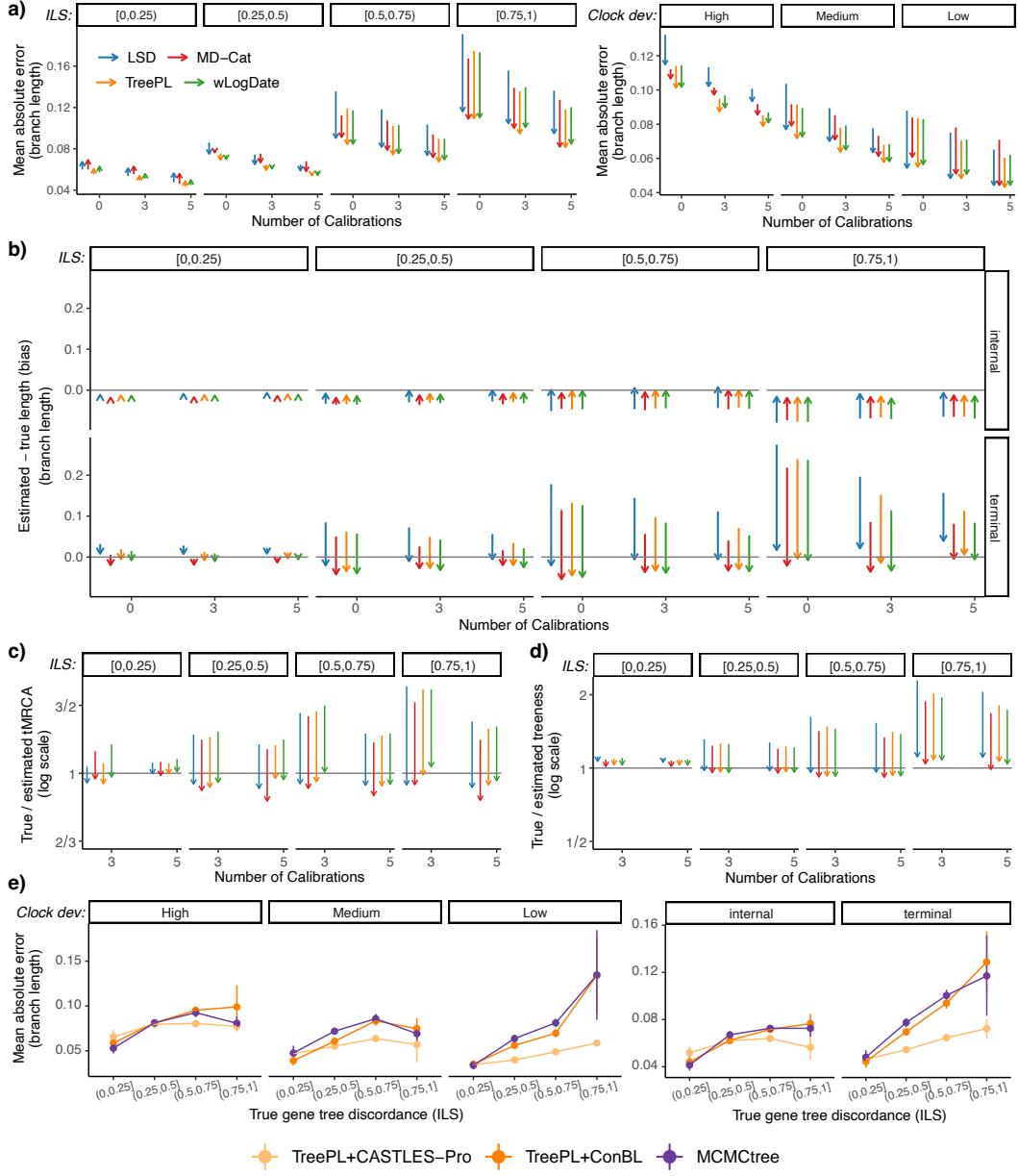


Figure 7.2: Results on 30-taxon simulated ILS datasets for conditions without outgroup. a) Change in height-normalized mean absolute error of branch lengths in the dating pipelines going from ConBL (arrow tail) to CoalBL (arrow head). Conditions vary in terms of the level of ILS (left: boxes), deviation from the clock (right: boxes), and number of calibrations (x -axis). b) Change in bias of branch lengths from ConBL to CoalBL. c) Change in true time of the most recent common ancestor (tMRCA) divided by estimated tMRCA from ConBL to CoalBL for the root-unfixed experiments in log2 scale. d) Change in true treeness (sum of internal branch lengths divided by sum of all branch lengths) divided by estimated treeness from ConBL to CoalBL for the root-fixed experiments in log2 scale. e) Comparison between mean absolute error of MCMCTree, TreePL+CASTLES-Pro and TreePL+ConBL for different levels of ILS, deviation from the clock, and branch type for experiments with five calibration points. Panels a, b, d, and e are for root-fixed experiments, while c is for root-unfixed (since tMRCA is fixed for root-fixed). The number of genes is 500 (100 replicates).

terms of [treeness](#) (Figure 7.9) and bias (Figure 7.10). However, not fixing the age of the root changes the patterns to some degree. In the *root-unfixed* conditions, the [ConBL](#) pipeline has a much larger underestimation bias for [internal](#) branches but lower levels of overestimation bias for [terminal](#) branches.

The inferred [tMRCA](#) (meaningful only for *root-unfixed* experiments) is substantially underestimated in all pipelines based on [ConBL](#) and slightly overestimated in most [CoalBL](#) pipelines (Figure 7.2c). MCMCTree also substantially underestimates [tMRCA](#) (Figure 7.11). The error in [tMRCA](#) becomes larger for all methods, especially those based on ConBL, as the level of [ILS](#) increases. Increasing the number of calibrations and decreasing deviation from the clock reduces the error of [tMRCA](#) for all methods, regardless of whether an [outgroup](#) is included (Figure 7.11).

The relative accuracy of dating methods is variable and depends on the number of calibrations and level of [ILS](#) (Figure 7.2a). Overall, the strict-clock LSD method is clearly the least accurate method, while TreePL is the most accurate, followed by MD-Cat. Moreover, patterns of error remain largely similar for the [RMSE](#) error and mean log error that emphasize shorter branches more than absolute error (Figure 7.7). For the log error, LSD has very high errors (perhaps due to the abundance of near-zero lengths), and pairing it with [ConBL](#) or [CoalBL](#) makes very little difference. The trends also remain similar if we examine node ages instead of branch lengths (Figure 7.12), focusing on absolute error and RMSE. There is, however, a somewhat larger underestimation bias for [CoalBL](#) pipelines when evaluating node age.

We compare MCMCTree with the most accurate ML-based dating method in our simulations, TreePL (Figures 7.2,7.13). The comparison between MCMCTree and TreePL+ConBL depends on the deviation from the clock, with TreePL+ConBL performing better in more conditions overall. TreePL+CoalBL is more accurate than the other two methods for all levels of deviation from the clock and [ILS](#), except for the lowest level of [ILS](#) (< 0.25 AD). These trends are consistent across different numbers of calibrations and when assessing node ages (Figure 7.13). The bias of MCMCTree and [ConBL](#) are very close in all conditions, with MCMCTree performing slightly better for [internal](#) branches and [ConBL](#) for [terminal](#) ones (Figure 7.13). Biases of both [internal](#) and [terminal](#) branches are substantially reduced when using TreePL+CoalBL.

S100 dataset. On the 100-taxon datasets, with 1000 genes, using [CoalBL](#) instead of [ConBL](#) reduces the mean absolute error of branch lengths and node ages across most sequence length levels and dating methods (Figure 7.3a). Increasing the number of calibration points reduces the error for all dating pipelines, and the advantage of [CoalBL](#) is larger when using

fewer calibrations. Given poorly estimated gene trees from 200bp genes and 10 calibration points, **CoalBL** and **ConBL** are tied. As the sequence length increases and **GTEE** decreases, dating using **CoalBL** improves far more than ConBL, such that with 1600bp, **CoalBL** has a clear advantage even with 10 calibration points. The trends for **RMSE** and log error are similar, though the magnitude of improvements differs in each case (Figure 7.14).

In terms of bias, using **CoalBL** instead of **ConBL** reduces the overall bias for all dating methods in all conditions when calibration points are used (Figure 7.3b), whether this bias is towards overestimation or underestimation. With the exception of MD-Cat, pipelines based on **ConBL** have a small underestimation bias for **internal** branch lengths and a larger overestimation bias for **terminal** branches. Both biases are substantially reduced with **CoalBL** and are either eliminated or transformed into a slight bias in the opposite direction. For MD-Cat, the trends for **internal** branches are similar to other methods, but **terminal** branches have a small underestimation bias that is slightly increased when using **CoalBL** in conditions with calibration points; as a result, it has an underestimation bias for node ages. When no calibration points are used, trends change for MD-Cat but remain similar for other methods. Similar patterns are observed when examining node ages (Figure 7.14).

Increasing the number of genes tends to improve the accuracy for all methods, but it also impacts the relative accuracy of methods (Figure 7.3c), with similar patterns across various metrics (Figure 7.15). Setting aside the least accurate method, LSD, given only 50 genes, **CoalBL** tends to be *worse* than ConBL. As the number of genes increases beyond 200, **CoalBL** performs better regardless of the sequence length or dating method. The impact of the number of genes is perhaps the clearest with 1600bp (i.e., accurate) gene trees, where **CoalBL** is worse than **ConBL** with 50 genes but becomes better at 200, and its advantage increases going from 200 to 1000 genes.

Both **tMRCA** and **treeness** have a substantial underestimation bias with **ConBL** and a smaller overestimation bias with **CoalBL** for all sequence lengths, number of calibrations, and dating methods (Figure 7.3de). For the most difficult replicates, the over/underestimation can reach 2 \times or more (Figure 7.16). The trends are consistent across different number of calibrations, but using more calibrations reduces the variation in **tMRCA** estimations of **CoalBL** pipelines, while having little to no impact on ConBL-based pipelines (Figure 7.16).

Scalability experiments. We evaluate the scalability of **ConBL** and **CoalBL** dating pipelines (both run with TreePL, which was the fastest ML-based dating method in our simulations; Figures 7.4, 7.18, 7.17). On the datasets with more than 1000 taxa, **ConBL** fails to finish with 64GB of memory on all replicates. In conditions with up to 1000 taxa, using **CoalBL** is far more scalable than using ConBL. With 1000 taxa, **CoalBL** finishes in about a

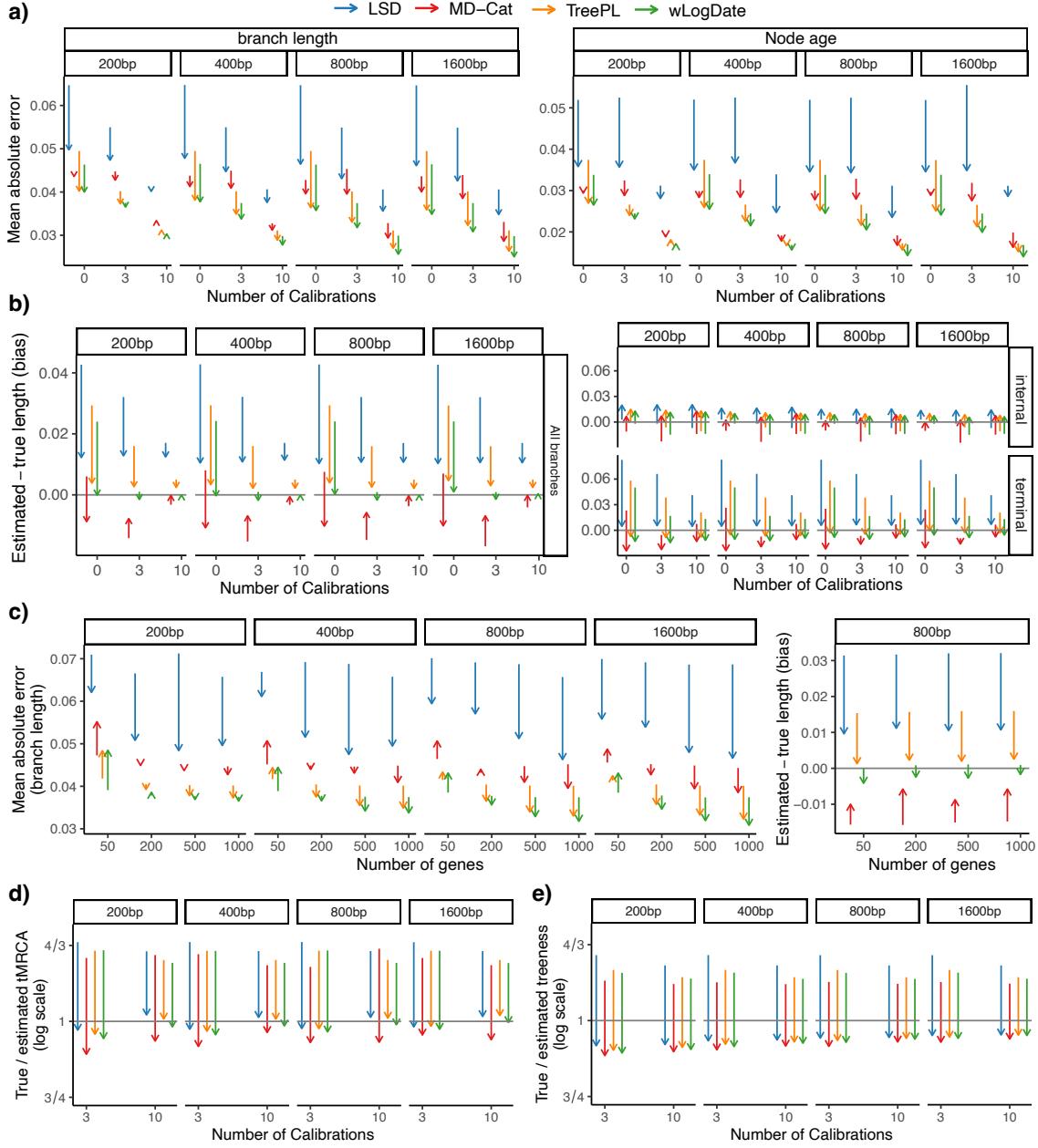


Figure 7.3: Results on 100-taxon ILS simulated datasets. a) Change in height-normalized mean absolute error of branch lengths and node ages in the dating pipelines when going from **ConBL** (arrow tail) to **CoalBL** (arrowhead) in the root-fixed experiments. Conditions vary in terms of the sequence length and number of calibrations. The number of genes is 1000. b) Change in bias of branch lengths for conditions with different sequence lengths in the *root-fixed* experiments. The number of genes is 1000. c) Change in mean absolute error and overall bias of branch lengths for conditions with different numbers of genes and sequence lengths in the *root-fixed* experiments. The number of calibrations is 3. d) Change in tMRCA ratio (true tMRCA divided by estimated tMRCA) for the *root-unfixed* experiments. e) Change in treeness ratio (true treeness divided by estimated treeness) for the *root-fixed* experiments. Conditions vary in terms of sequence length and number of calibrations.

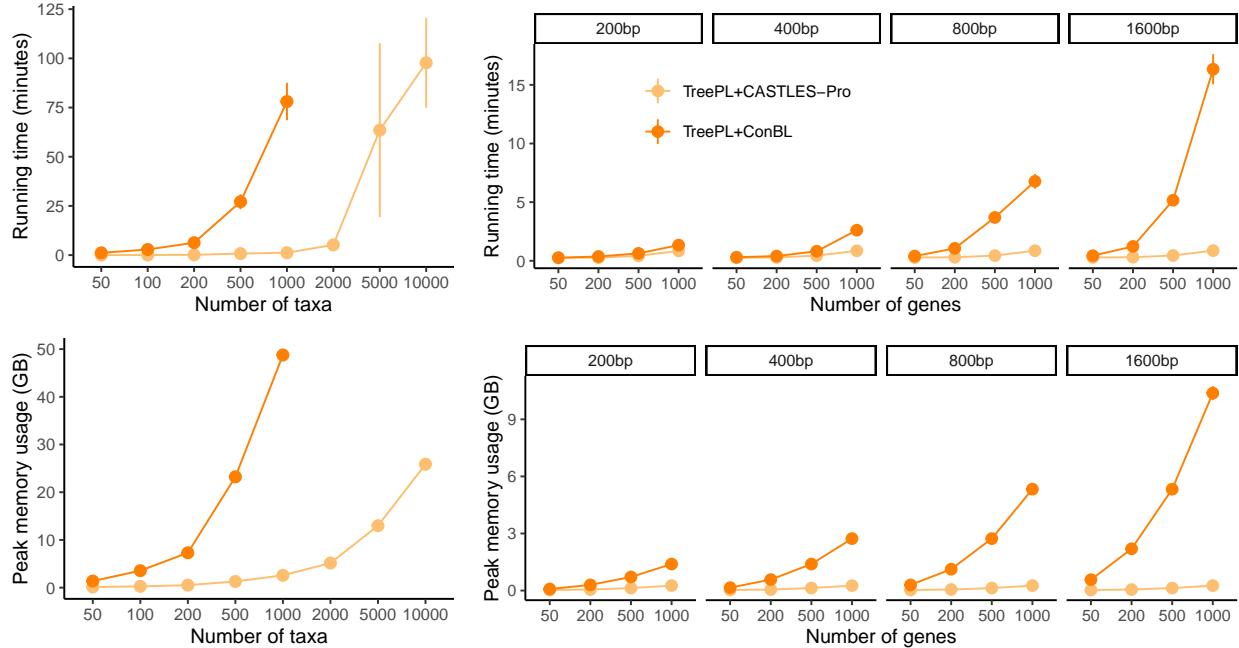


Figure 7.4: Scalability on the large (left) and S100 (right) simulated datasets, varying number of taxa, number of genes and sequence length. Results show average total runtime of branch length estimation and dating and peak memory usage for TreePL with [CoalBL](#) or [ConBL](#) in the *root-fixed* experiments. The reported runtime is the sum of the time spent for dating and branch length estimation, and does not include the time spent for gene tree estimation or species tree topology estimation; those are assumed to be available beforehand, and [ConBL](#) and [CoalBL](#) both estimate branch lengths for a fixed tree topology. [ConBL](#) fails on trees with more than 1000 [taxa](#) due to memory limit. The results are averaged over 20 replicates for the large dataset and 50 replicates for S100 dataset in each model condition.

minute on average, while [ConBL](#) takes more than an hour. In this model condition, [ConBL](#) uses up to 49 GB of memory, while [CoalBL](#) uses up to 3 GB on average. In addition, [CoalBL](#) finishes in all 20 replicates of the final model condition with 10,000 species, taking 1.6 hours on average and up to 26 GB of memory. In conditions with 5,000 and 10,000 species, the dating step (TreePL) takes 76.93% and 56.15% of the total runtime and CASTLES-Pro uses less than 45 and 15 minutes on average, respectively, whereas in other conditions, more than 75% of the total runtime is used by CASTLES-Pro (Figure 7.17).

On the S100 dataset, for all numbers of genes and all sequence lengths, [CoalBL](#) pipeline uses less time and memory than ConBL. As the number of genes or the sequence length increases, the gap between the runtime and memory usage of the two pipelines increases, so that for 1000 genes with 1600bp sequence length, [CoalBL](#) is more than an order of magnitude faster and more memory-efficient (260MB compared to 10 GB on average) than ConBL. In all cases when using ConBL, the runtime and memory usage is almost entirely dominated

by the branch length estimation step using RAxML in conditions with more than 50 genes (Figure 7.18b). In pipelines using CoalBL, however, TreePL takes 96%, 84%, 56% and 28% of the total runtime when using 50, 200 and 500 genes, respectively, dominating the CASTLES-Pro step.

7.3.2 Suboscines: corrected species diversification rates

Using CoalBL instead of ConBL for dating the subsocines phylogeny results in substantial reductions in the length of terminal branches and ages of shallow nodes for both ASTRAL and concatenation topologies (Figures 7.5, 7.19 and 7.20). Using either topology, the mean terminal branch length decreases by 5% or more for 23 out of 36 suboscines families when we use CoalBL instead of ConBL, with a marked difference in Philepittidae, Platyrinchidae and Strigopidae families with 2X decrease (Figure 7.21). Similar to results in simulations, the treeness of the dated trees increases when using CoalBL instead of ConBL (from 0.4 to 0.46 for the concatenation topology and from 0.35 to 0.40 for ASTRAL).

Similar to the results reported in [365], there is a significant drop in diversification rates, in particular speciation rates, in the last 2 million years when both topology and branch lengths are estimated using concatenation (Figure 7.5b). Using CoalBL instead of ConBL on the concatenation topology reduces the drop in speciation rates and almost eliminates the final drop in extinction rates. As a result, the net diversification rates are far more stable with CoalBL dating compared to ConBL: across the 30–40 million years of suboscine evolution. The trends when using the ASTRAL topology are similar, with the net diversification rates becoming much more stable up until the final 1 Mya (Figure 7.22). On this topology, CoalBL fully eliminates the final drop in both speciation and extinction rates and results in almost completely constant rates across the diversification history of suboscines.

Therefore, for all three topologies, the use of CoalBL for branch length estimation reduces the severe drop in diversification rates close to the present time. These drops were likely an artifact of the overestimation bias of concatenation for terminal branch lengths. The results for the ASTRAL topology show more stable speciation, extinction, and diversification rates when using CoalBL branch lengths compared to ConBL. However, it is worth noting that [365] preferred the concatenation topology due to discrepancies with known relationships observed in the ASTRAL topology. These discrepancies are likely due to very high levels of gene tree estimation error, given short UCEs and large numbers of taxa.

The 1683 species of the suboscines dataset are distributed across 36 families and 325 genera. Examining the age of the genera with more than 10 representative species (Figure 7.23a), the age estimated by CoalBL is shallower than the age estimated by concatenation

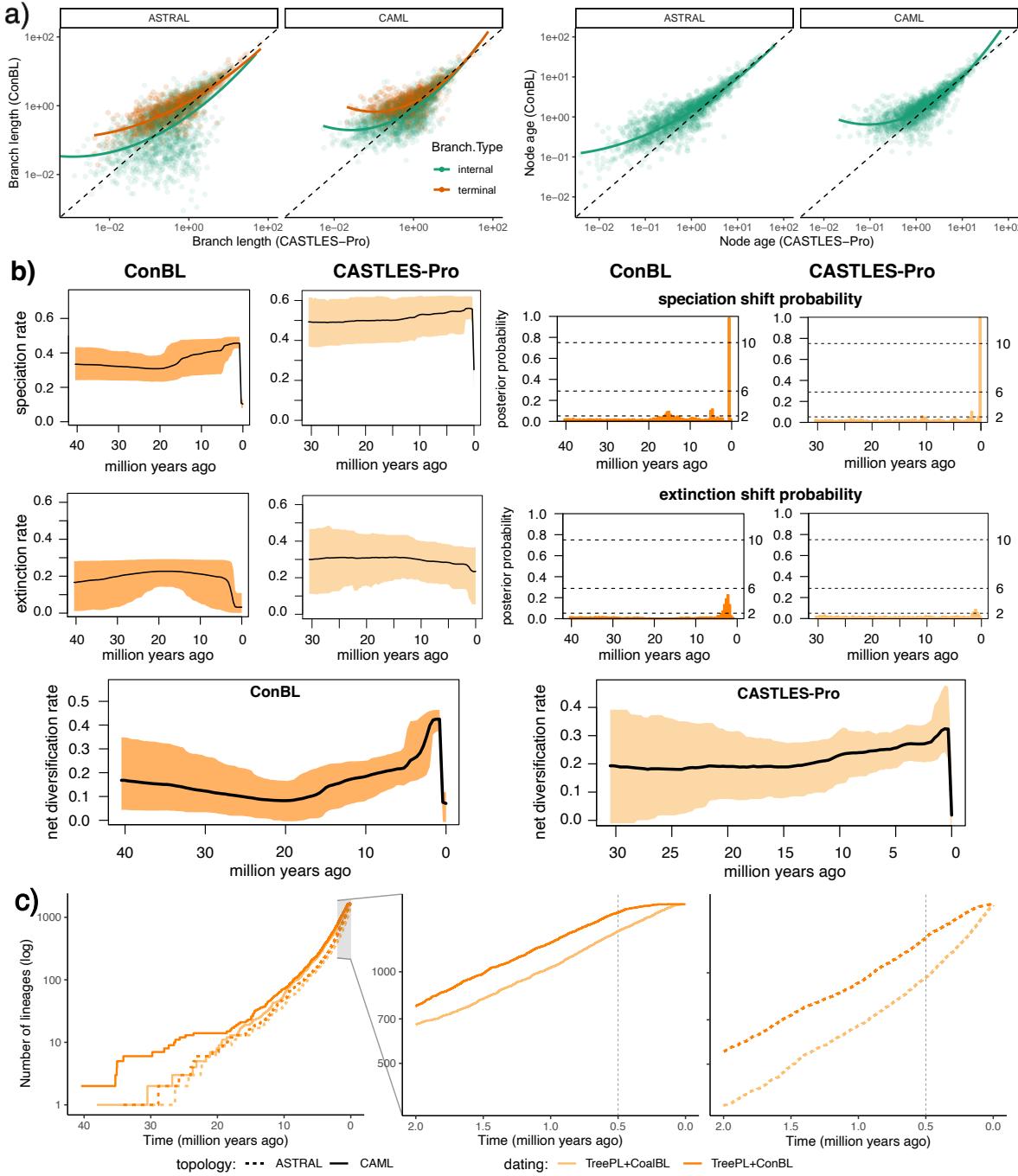


Figure 7.5: Results on the Suboscines dataset of [365]. Dating is done using TreePL with four calibration points as in the original study. a) Correlation between time-unit branch lengths and node ages of the trees dated based on branch lengths of **ConBL** and **CoalBL** on the concatenation and ASTRAL topologies. b) Diversification rates, including speciation and extinction rates and speciation and extinction shift times for the concatenation topology from the original study (after removing the **outgroup** clade) based on branch lengths estimated using **ConBL** or CoalBL. c) Lineage-through-time plots (log scale y -axis) for the dated trees with the concatenation topology. The left plot shows the complete diversification history of the group and the right plots focuses on the final two million years.

for 27 out of 39 genera. Notable reductions in clade ages occur for *Pitta* (from 15.45 to 10.10 Mya) and *Gralria* (from 14.93 to 11.08 Mya). Assessing the 19 families with at least two representative genera (Figure 7.23b), the dates estimated by TreePL with the CoalBL pipeline are younger than those based on ConBL in 16 out of the 19 families, with an average decrease of 18.36%.

lineage-through-time (LTT) plots also show notable differences between ConBL and CoalBL (Figure 7.5). Regardless of the topology, LTT plots for all dated trees agree with a birth-death diversification model (i.e., showing a steep upward shift in the slope of the LTT curves in semi-log scale near the present time [378]) for up until the past half a million years. However, the concatenation tree dated with ConBL shows a drastic slowdown in the LTT curve right around half a million years ago and nearly halts closer to the present; the ASTRAL tree dated using ConBL also shows a slowdown but less drastically (Figure 7.5c). The LTT plots of CoalBL trees do not show this slowdown with either topology and, in fact, show accelerated growth with the ASTRAL topology. This trend aligns with the significant drop in diversification rates close to the present time observed for trees dated based on ConBL but not CoalBL. The overestimation bias for terminal branch lengths (seen in simulations for ConBL but not CoalBL) can explain the unexpected drop in rate of growth in LTT plots.

7.3.3 Neoavian phylogeny

We used the three most accurate ML-based dating methods in simulations (TreePL, MD-Cat, and wLogDate) to date the 363-taxon avian phylogeny of [328] where the topology was estimated using ASTRAL. Using CoalBL instead of ConBL results in shorter terminal branches and shallower recent nodes for all three dating methods (Figures 7.6, 7.24). This trend agrees with the results in simulations in which ConBL produced longer terminal branches than CoalBL. For internal branches, in contrast, lengths were similar (with the exception of MD-Cat, where CoalBL results in shorter internal branches). Treeness was higher for CoalBL combined with MD-Cat dating (0.291 versus 0.277), just as in simulations, but TreePL (0.271 versus 0.269) and wLogDate (0.286 vs 0.285) did not result in significant differences in treeness.

Terminal branches produced by ConBL are, on average, longer than those produced by CoalBL in 9 out of 11 higher-order clades (Figure 7.6) and only slightly shorter in two clades (Mirandornithes and Opisthocomiformes for MD-Cat, and Columbimorphae and Opisthocomiformes for TreePL, with ratios greater than 0.9). The differences are most dramatic for the hard-to-place orders Strigiformes (owls) and Apterygiformes (kiwis), as well as Falconiformes (falcons) and associated families (Figure 7.26). Changes in terminal branch lengths

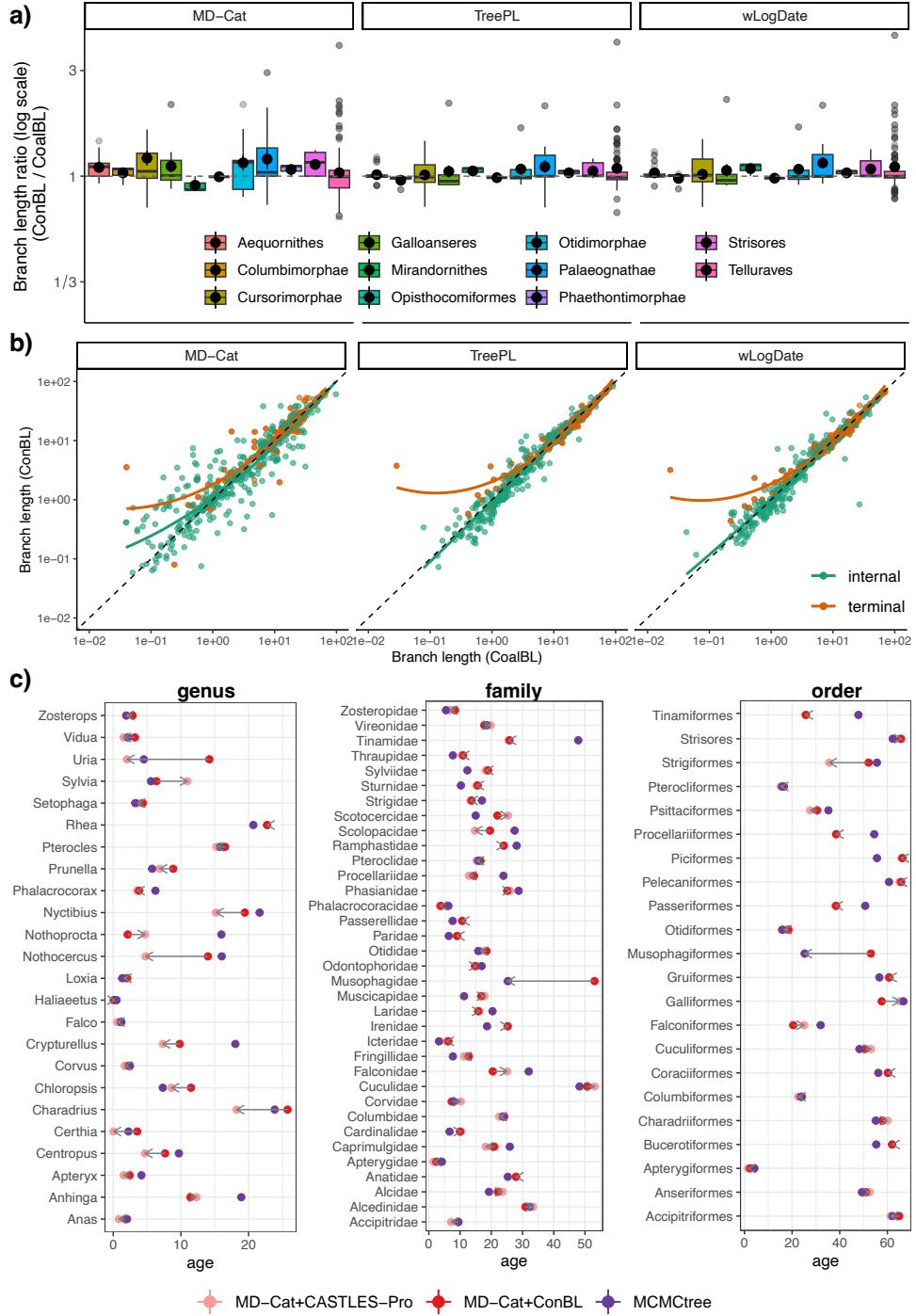


Figure 7.6: Dating the 363-taxon ASTRAL topology from the [328] bird study using TreePL, MD-Cat and wLogDate. a) terminal branch lengths of dated trees based on **ConBL** divided by terminal branch lengths of dated trees based on **CoalBL** in log2 scale across 11 higher order clades. b) Correlation between all branch lengths estimated by dating pipelines based on **ConBL** and **CoalBL**. c) Age of the genera that have at least two representative species and the age of families and orders with at least three representative species estimated by MD-Cat (see Figures 7.28 and 7.27 for TreePL and wLogDate) based on **CoalBL**, **ConBL**, as well as MCMCTree analysis from the original study.

also affect the ages of recent nodes. At least 19 out of 24 genera with two or more representative species were younger in the CoalBL results (regardless of the dating method used), and in some cases, the difference was substantial (Figures 7.27, 7.28 and 7.6c). For example, for the MD-Cat tree, *Uria* goes from 14 Mya using ConBL to 2 Mya using CoalBL.

Differences in the age of families and orders, in contrast, were marginal, with a couple of notable exceptions (Figure 7.26). The age of the order Musophagiformes was estimated as 25.4 Mya in the original study and 25.2 Mya in the MD-Cat+CASTLES-Pro tree, but was more than twice older in the MD-Cat+ConBL tree (53.1 Mya). The age of order Strigiformes was estimated as 35.6 Mya in the MD-Cat+CASTLES-Pro tree, which is substantially younger than its age in the MD-Cat+ConBL tree (52.1 Mya) and MCMCTree (55.6 Mya). The more stable dating of families and orders compared to genera using different dating pipelines can be explained by the observation from simulations that the overestimation bias of ConBL is mostly concentrated on terminal branches, thus affecting deeper nodes less than more recent ones (Figure 7.24).

A focal point of [328] was the age of Neoaves in relation to the K-Pg extinction event (estimated at 66.04 Mya). For this deep node, the choice of the dating method matters more than whether we use ConBL or CoalBL, and the results differ slightly from the original study. [328] put the age of the MRCA of Neoaves incredibly close to the K-Pg boundary, at 67.4 Mya, and only one additional speciation event occurred prior to K-Pg boundary (though several more nodes had 95% confidence intervals that overlapped with the boundary). In all of our six dating pipelines, which unlike the original analyses do not include confidence intervals for fossils (see Material and Methods), Neoaves was dated a bit earlier (between 72.5 and 68.8 Mya, depending on the pipeline, see Table 7.1). Moreover, between 16 and 18 speciation events occurred right before K-Pg boundary in rapid succession in all these trees. These results, however, need to be treated with caution and underscore the importance of considering fossil dating uncertainty for dating deeper nodes of the tree.

7.4 DISCUSSION AND CONCLUSIONS

We introduced a scalable four-step coalescence-aware pipeline for dating species trees that uses coalescent-based methods to estimate the topology and branch lengths of the species tree from gene trees (CoalBL). Simulation analyses showed that this pipeline is substantially more scalable than pipelines using concatenation for divergence time estimation (ConBL) and easily scales to datasets with thousands of species and genes in about an hour, enabling us to date ultra-large species trees. Furthermore, dated trees obtained from this pipeline better capture species divergence times (as opposed to genic divergences) compared to pipelines

using concatenation when the level of Incomplete lineage sorting is at least moderate. In particular, in the presence of substantial Incomplete lineage sorting, dating using concatenation has an overestimation bias for terminal branches, an issue that the coalescence-aware pipeline eliminates. Our results on two avian biological datasets corroborate results from simulations: concatenation produced longer terminal branches (especially for shorter ones), pushing genus-level nodes further back in time. The coalescence-aware pipeline reduced terminal lengths and substantially changed the age of many genera, and some families and orders (Figures 7.5,7.6,7.23).

An overestimation of terminal branches may compromise inferences about diversification dynamics. First, an overestimation of terminal branches may erase the signal of extinction in dated trees because extinction results in an apparent excess of recent cladogenetic events, producing an upturn in the log-scaled lineages-through-time plots [364]. In contrast, empirical dated trees derived from concatenation methods not only rarely show an upturn in the LTT plot but more frequently show a downturn near the present [e.g., 379, 380]. In fact, diversification slowdowns are detected so frequently in empirical phylogenies that they have inspired a variety of hypotheses for their explanation [381]. Slowdowns have been attributed to incomplete species sampling [382], non-random species sampling [383], DNA substitution model misspecification and thus shortening of internal branches [384], protracted speciation ([385], geographic speciation [386], ecological niche saturation [387], and gene tree-species tree discordance [363]. Our simulations suggest that deep coalescence of gene trees may be a major source of terminal branch overestimation, resulting in misleading inference of extinction-free slowdown in diversification dynamics. Our new methodology provides a solution to this pervasive bias and may improve studies of diversification dynamics.

Our simulation studies show that using a coalescent-aware dating is especially beneficial when few calibrations are available, where the overestimation bias of concatenation is largest. Therefore, the discordance-aware pipeline can be most useful to biologists in studies where the size of the species tree is large but few fossil calibrations (or sampling times) are available. While the four-step dating pipelines have substantially less bias than pipelines based on concatenation and reduce the overestimation bias for terminal branches, they are not completely unbiased. In particular, when gene tree estimation error or deviation from the clock is high, coalescent-based dating pipelines can have a small underestimation bias for terminal branches (Figure 7.2). In addition, the four-step pipeline is more impacted by gene tree estimation error; when using very few genes or gene trees with very high gene-tree estimation error, it can be less accurate than using concatenation (Figure 7.3).

For the third step of our pipeline, substitution branch length calculation from gene trees, we only considered CASTLES-Pro. This was motivated by results by [75, 305] that show that

it outperforms other methods in terms of branch length accuracy and scalability. However, as better methods are developed for this step, they can be used instead of CASTLES-Pro. The goal of the present chapter is not to argue that CASTLES-Pro is the best method of branch length calculation, but rather, that [coalescent](#)-aware methods should be used in dating.

Although our simulation study covered a broad range of model conditions, it has limitations. In particular, we only simulated calibration points as fixed values, and used the true time of the calibrations as input to the dating methods. Future studies could improve upon this by incorporating more advanced calibration simulation protocols, such as modeling calibrations as probability densities or min-max bounds, which are used by some Bayesian and ML-based dating methods, or by simulating calibration points with associated errors. Such estimates of uncertainty may have to be adjusted to account for the effects of [deep coalescence](#).

Although this study focused solely on datasets with [ILS](#), other sources of [gene tree](#) discordance can also affect dating and subsequent diversification analyses. For example, introgression can contribute to biases in divergence time estimates [388]. Furthermore, [75, 305] show that using CASTLES-Pro or CASTLES-Pro+TCMM instead of concatenation can substantially improve the accuracy of [SU](#) branch lengths in the presence of sources of [gene tree](#) discordance other than [ILS](#), such as gene duplication and loss and horizontal gene transfer. For some of these sources of discordance, specific dating methods have been developed. For example, MaxTiC [216] uses horizontal gene transfer events to estimate divergence times on a species tree. This method can be especially useful for bacterial or microbial organisms that have abundant levels of horizontal gene transfer [114], but for which the fossil record is scarce, complicating the use of typical fossil-based dating methods. However, its application is limited to datasets with sufficient levels of horizontal gene transfer. The scalable dating pipeline introduced here can be applied more broadly. Future work should explore dating species trees on datasets that include multiple sources of [gene tree](#) discordance.

7.5 METHODS AND SOFTWARE COMMANDS

This section includes the details of the experimental study and software commands. All scripts and data used in this study are avaialable at <https://github.com/ytabatabae/\gls{coalescent}-based-dating>. The experiments were run on the University of Illinois campus cluster given 64GB of memory.

Dating Methods

Least-square dating (LSD). LSD [373] is an ML-based dating method that assumes a strict molecular clock and models the uncertainty of the substitution-unit branch lengths using a Gaussian distribution. LSD solves a convex optimization problem, where the likelihood function is defined in the form of a weighted least-squares criterion that minimizes deviation from a strict clock. LSD was run using QPD* mode described in [373] with minimum branch length in the time-scaled tree (option `-u`) set to 0.001.

We used the LSD software (v2) available from <https://github.com/tothuhien/lsd2> with the following command when inferring a unit-ultrametric tree without calibration points:

```
lsd2 -i <rooted-species-tree> -a 0 -z 1 -s <seq-length> -u 0.001 -o  
<dated-species-tree>
```

and the following command when using fossil calibrations specified with `-d`

```
lsd2 -i <rooted-species-tree> -d <calib-file> -s <seq-length> -u  
0.001 -o <dated-species-tree>
```

where the option `-s` specifies the sum of the lengths of the sequence alignments for all loci, and option `-u` specifies the minimum branch length in the output time-scaled tree, which we set at 1e-3.

wLogDate. wLogDate [374] is an ML-based method for estimating divergence times that uses a similar optimization function as LSD, but it uses a log transformation before minimizing the variance of the branch rates. The advantage of wLogDate to LSD is that the log transformation results in a symmetrical penalty function, where increases or decreases in the rates have a similar effect on the optimization function, which is useful when the clock model deviates from the strict molecular clock. However, unlike LSD, wLogDate solves a non-convex optimization problem. wLogDate was run with default settings in our experiments.

We used the wLogDate software (v1.0.4) available at <https://github.com/uym2/wLogDate> with the following command, where the option `-t -b` specifies the fossil calibrations in backward time:

```
python3 launch_wLogDate.py -i <rooted-species-tree> -o <dated-tree>  
[-t <calib-file> -b]
```

MD-Cat. MD-Cat [80] is an ML-based dating method that uses a categorical model of rates to approximate different clock models and co-estimates rate categories and branch lengths in the unit of time using an Expectation-Maximization (EM) algorithm. In this model, the branch rates are drawn from a discrete distribution defined by k rate categories (default $k = 50$). The likelihood function in MD-Cat considers a parameter for each of the k rate categories and the tree branch lengths in time units, and the EM algorithm is used to maximize this likelihood function and estimate all branch length and rate parameters.

We used the MD-Cat software available at <https://github.com/uym2/MD-Cat> with the following command:

```
python3 md_cat.py -i <rooted-species-tree> -o <dated-species-tree>
-p 10 [-t <calib-file> -b] [-l <seq-length>]
```

where the option `-t -b` specifies the fossil calibrations in backward time and option `-l` specifies the length of the sequence alignment used to infer the tree. In simulations, MD-Cat was run with the number of starting points set as 10 (option `-p`) to speed up the runtime given the large number of replicates, but on biological data we used it with the default 100 starting points. On the neoavian dataset we set the sequence length parameter (`-l`) as 40,000. Other parameters were set as their default settings.

TreePL. TreePL [228] is a method for estimating divergence times on large phylogenies that uses the penalized likelihood framework of [389]. This optimization function allows for rate variation across different branches, with a smoothing parameter that determines the penalty for rate differences over the tree. The TreePL algorithm operates in two steps to optimize the likelihood: a greedy hill-climbing phase and a stochastic phase using simulated annealing.

We used the TreePL software (v1.0) available at <https://github.com/blackrim/treePL>. For the simulation study, we set the smoothing parameter to 100 and the number of `sites` as the sum of the lengths of the sequence alignments for all loci. For the Neornithes dataset, we set the smoothing parameter as 1000. For the suboscines biological dataset, we used the same setting as the original study. In all experiments, We used both `thorough` and `prime` flags to do a thorough analysis and determine the best optimization parameters.

MCMCTree. MCMCTree [225] is a Bayesian method for estimating species divergence times under different molecular clock models. Unlike the ML-based methods that use fixed calibration points or min-max bounds, MCMCTree uses soft fossil calibrations with flexible probability distributions for describing the uncertainty in fossil ages. We use MCMCTree

with approximate likelihood calculation [371, 372] that speeds up the calculation of likelihood function during the MCMC sampling for large alignments and trees. For the *root-unfixed* experiments in the simulation studies, where the age of the root is not one of the calibration points, we set the soft bound for root age for MCMCTree as $10 + T$ where T is the true root age.

We used the MCMCTree software from PAML (v4.10.7) package [390] available at <http://abacus.gene.ucl.ac.uk/software/paml.html> under the GTR + Γ model. Most of the parameters in our analysis were taken from the MCMCTree analysis in [328]. We set the calibration bounds as $(x-0.01, x+0.01)$ where x is the exact calibration point. We passed the concatenation of all gene sequences as a single partition to MCMCTree (specified with `ndata`). For nodes without calibrations, we used a birth-death process with $\lambda = \gamma = 1, \rho = 0.1$ to get an approximately uniform kernel. We used a relaxed clock model where rates are log-normally distributed across branches and a gamma-Dirichlet prior on rates. For the simulation study, during the MCMC sampling, samples were taken every 100 steps with a total number of iterations of 5,050,000 where the first 50,000 steps were treated as burn-in. The following shows all the parameters used for running MCMCTree.

```

seed = 1
ndata = 1
seqtype = 0      * 0: nucleotides; 1:codons; 2:AA
clock = 2        * 1: global clock; 2: independent rates; 3: correlated
                  rates
model = 7        * 0:JC69, 1:K80, 2:F81, 3:F84, 4:HKY85, 5:T92, 6:TN93,
                 7:REV, 8:UNREST, 9:REVu; 10:UNRESTu
alpha = 0.5      * alpha for gamma rates at sites
ncatG = 4        * No. categories in discrete gamma
cleandata = 0    * remove sites with ambiguity data (1:yes, 0:no)?
BDparas = 1 1 0.1 * birth, death, sampling
kappa_gamma = 6 2      * gamma prior for kappa
alpha_gamma = 1 1      * gamma prior for alpha
rgene_gamma = 2 2      * gammaDir prior for rate for genes
sigma2_gamma = 1 10 * gammaDir prior for sigma^2 (for clock=2 or 3)
finetune = 1: .1 .1 .1 .1 .1 * auto (0 or 1): times, musigma2,
                  rates, mixing, paras, FossilErr
print = 1        * 0: no mcmc sample; 1: everything except branch rates
                  2: everything
burnin = 50000
sampfreq = 100

```

```

nsample = 50000
usedata = 2      * 0: no data; 1:seq like; 2:normal approximation;
                 3:out.BV(in.BV)

```

Branch length estimation

- **CASTLES-Pro.** We used CASTLES-Pro as implemented in the species tree estimation software ASTER (v1.19.3.5) available at <https://github.com/chaoszhang/ASTER> to estimate SU branch lengths. To estimate lengths on a fixed species tree topology (specified with the option -C -c), we used the following command:

```

astral4 -i <gene-tree-path> -C -c <species-tree-topology> -o
<output-path> --root <outgroup-name> --genelength <gene-seq-len>

```

where --root specifies the `outgroup` name (if known) and --genelength specifies the average gene sequence length (default: 1000bp).

- **Concatenation.** We used RAxML (v8.2.12) that is available at <https://github.com/stamatak/standard-RAxML> to estimate substitution-unit branch lengths on a fixed species tree topology (with the option -f -e) using a concatenated sequence alignment. We used the following command:

```

raxmlHPC - PTHREADS -f e -t <species_tree_path> -m GTRGAMMA -s
<alignment_path> -n RES -p 4321 -T 16

```

7.5.1 Simulated Datasets.

We used a modified version of the simulation software SimPhy that generates species trees with SU branch lengths to generate the large ILS dataset. We used the following command, where the parameters are taken from Table 7.2.

```

./simphy -rs "$s" -rl f:"$g" -rg 1 -sb lu:0.0000001,0.000001 -sd
lu:0.0000001,sb -st ln:16,1 -sl f:"$sp" -so f:1 -si f:1 -sp
u:$min_pp,$max_pp -su ln:-17.27461,0.6931472 -hh f:1 -hs ln:1.5,1
-hl ln:1.551533,0.6931472 -hg ln:1.5,1 -cs 14907 -v 3 -o $sp
-o t 0 -op 1 -od 1 > log_$sp.txt

```

7.6 ADDITIONAL FIGURES AND TABLES

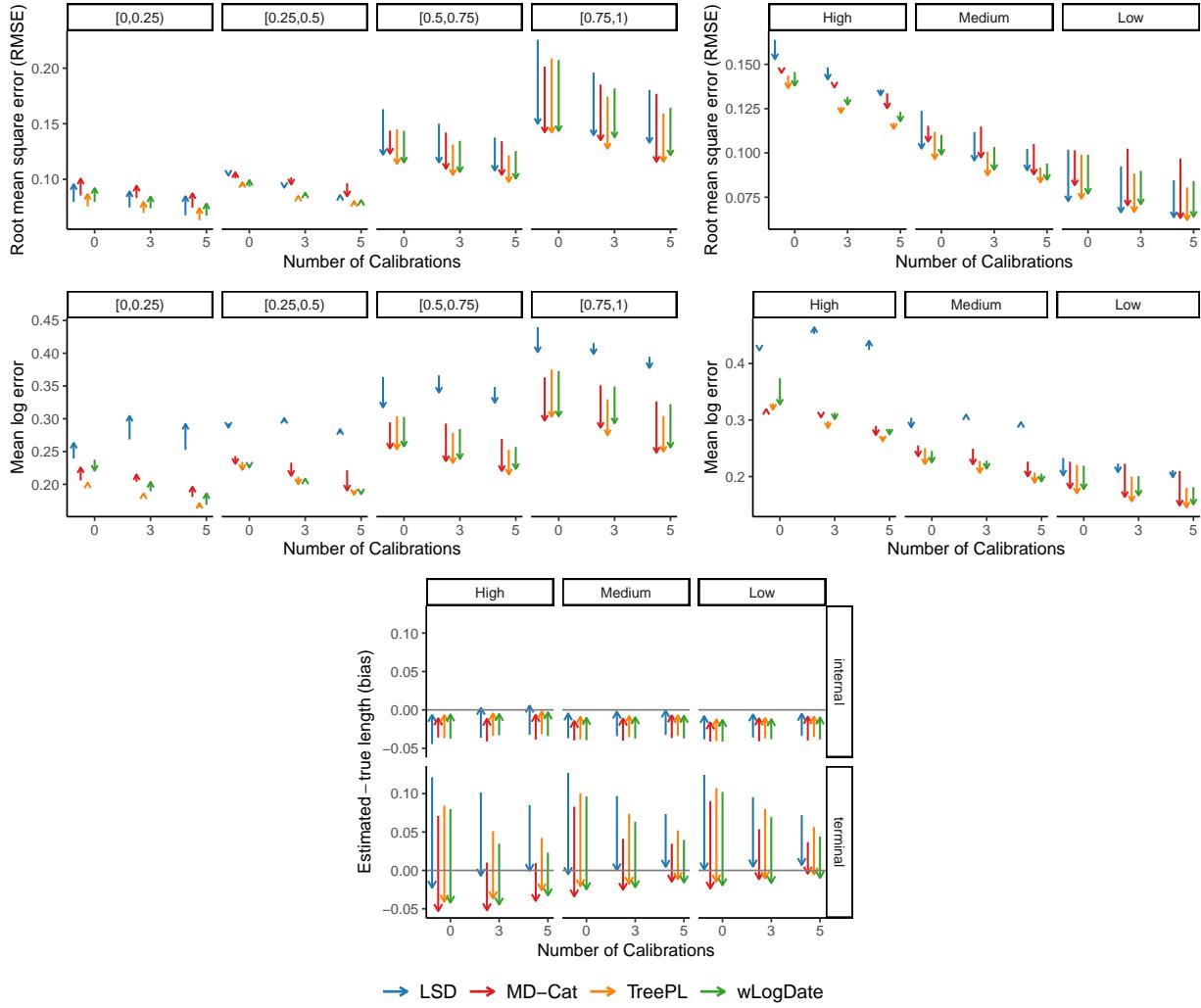


Figure 7.7: (**MVroot, Branch length**) Reduction (or increase) in height-normalized RMSE, mean log error and bias of branch lengths in time units using CASTLES-Pro instead of [ConBL](#) for branch length estimation in different ML-based dating pipelines. For each dating method, the direction of the arrow is from average error in the final dated tree when branch length estimation is done using [ConBL](#) to the average error when using CASTLES-Pro for branch length estimation. The results are shown on the 30-taxon simulated datasets for conditions without [outgroup](#) and in the *root-fixed* experiments. The number of genes is 500 and the number of replicates is 100. In the left panels of the top two rows, the columns show different levels of ILS and in the right panels they show different levels of deviation from the clock.

Table 7.1: Age of the neoaves as estimated by different dating pipelines (MCMCTree and ML-based dating using CASTLES-Pro and [ConBL](#) branch lengths) and the number of speciations before the K–Pg boundary on the 363-taxon dataset of [328]. Dating is done on an ASTRAL topology from the original study.

| | Age of neoaves | # speciations before K–Pg |
|---------------------------|----------------|---------------------------|
| MCMCTree (original study) | 67.38 | 2 |
| TreePL+CASTLES-Pro | 70.83 | 17 |
| TreePL+ConBL | 70.00 | 16 |
| MD-Cat+CASTLES-Pro | 69.03 | 16 |
| MD-Cat+ConBL | 68.83 | 18 |
| wLogDate+CASTLES-Pro | 72.48 | 17 |
| wLogDate+ConBL | 71.22 | 17 |

Table 7.2: Parameters used in SimPhy simulation of large dataset.

| Arg. | Description | Value |
|------|---------------------------------------------------------|---------------------------------|
| RS | Number of replicates | 20 |
| RL | Number of loci | 1000 |
| RG | Number of genes | 1 |
| ST | Maximum tree length | LogNormal(16,1) |
| SL | Number of taxa | 50-10,000 |
| SB | Speciation rate | LogNormal(1.0e-7,1.0e-6) |
| SD | Extinction rate | LogNormal(1.0e-7,SB) |
| SP | Global population size | Uniform(20000,2000000) |
| SU | Global substitution rate | LogNormal(-17.27461, 0.6931472) |
| HS | Species-specific branch rate heterogeneity modifiers | LogNormal (1.5,1) |
| HL | Gene-family-specific rate heterogeneity modifiers | LogNormal (1.551533,0.6931472) |
| HG | Gene by lineage specific rate heterogeneity modifiers | LogNormal (1.5,1) |
| HH | Gene-by-lineage-specific locus tree parameter | 1 |
| SO | Outgroup branch length relative to half the tree length | 1 (with outgroup) |
| CS | Random number generator seed | 14907 |

Table 7.3: Empirical statistics of the large simulated dataset. [AD](#) refers to average normalized [RF](#) distance between the model species tree and true gene trees, and [GTEE](#) refers to average normalized [RF](#) distance between true and estimated gene trees. The number of replicates in each model condition is 20 and the number of genes is 1000.

| Number of taxa | Number of calibrations | AD | GTEE |
|----------------|------------------------|--------|--------|
| 50 | 3 | 35.98% | 28.84% |
| 100 | 5 | 31.22% | 29.01% |
| 200 | 7 | 45.11% | 39.08% |
| 500 | 11 | 39.33% | 34.51% |
| 1000 | 15 | 39.73% | 36.89% |
| 2000 | 22 | 51.06% | 42.21% |
| 5000 | 35 | 35.53% | 36.44% |
| 10000 | 50 | 50.18% | 43.83% |

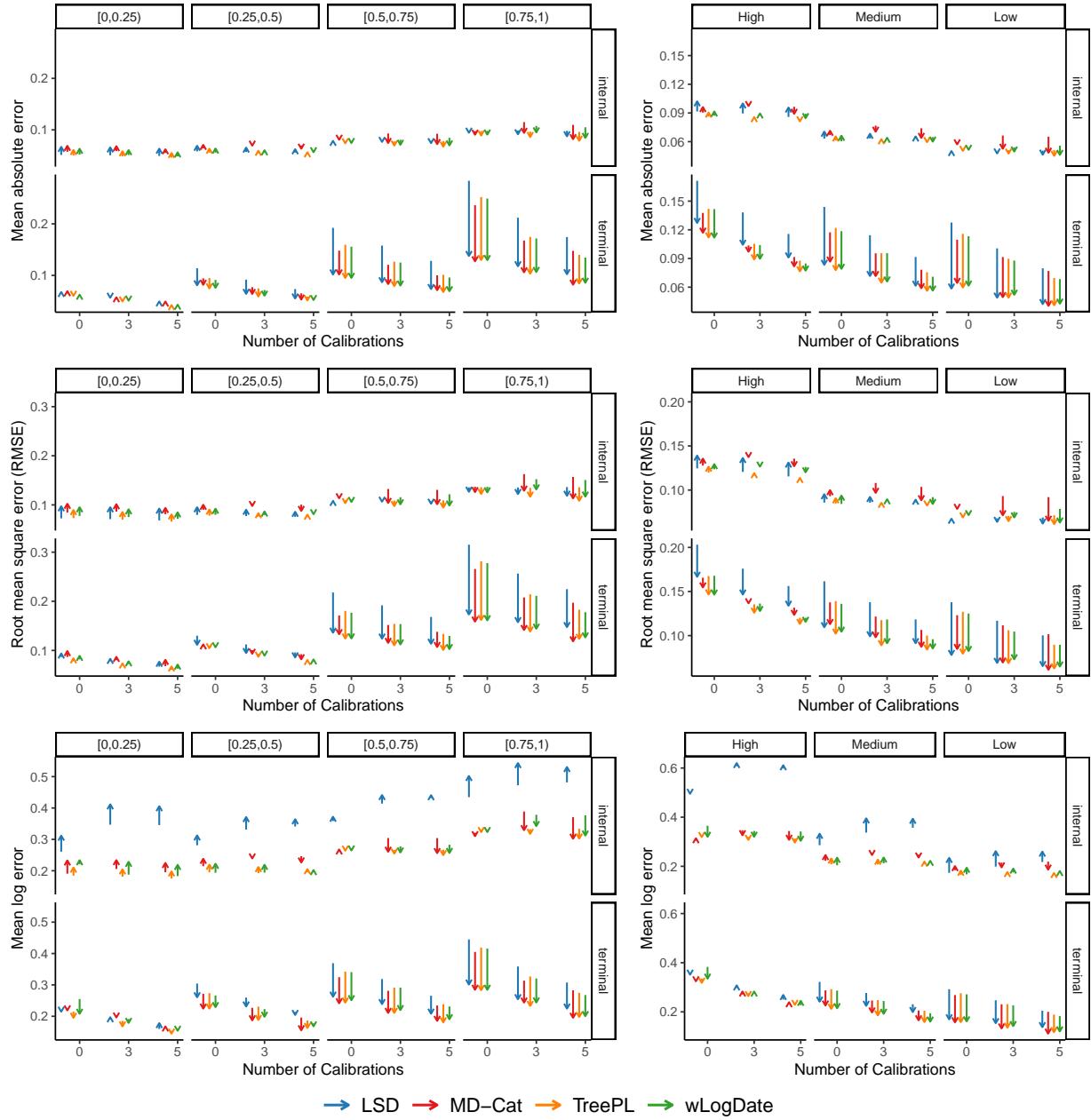


Figure 7.8: (**MVroot, Branch length**) Reduction (or increase) in height-normalized mean absolute error, RMSE and mean log error of **terminal** or **internal** branch lengths in time units using CASTLES-Pro instead of **ConBL** for branch length estimation in different ML-based dating pipelines. For each dating method, the direction of the arrow is from average error in the final dated tree when branch length estimation is done using **ConBL** to the average error when using CASTLES-Pro for branch length estimation. The results are shown on the 30-taxon simulated datasets for conditions without **outgroup** and in the *root-fixed* experiments. The number of genes is 500 and the number of replicates is 100. In the left panels, the columns show different levels of **ILS** and in the right panels they show different levels of deviation from the clock.

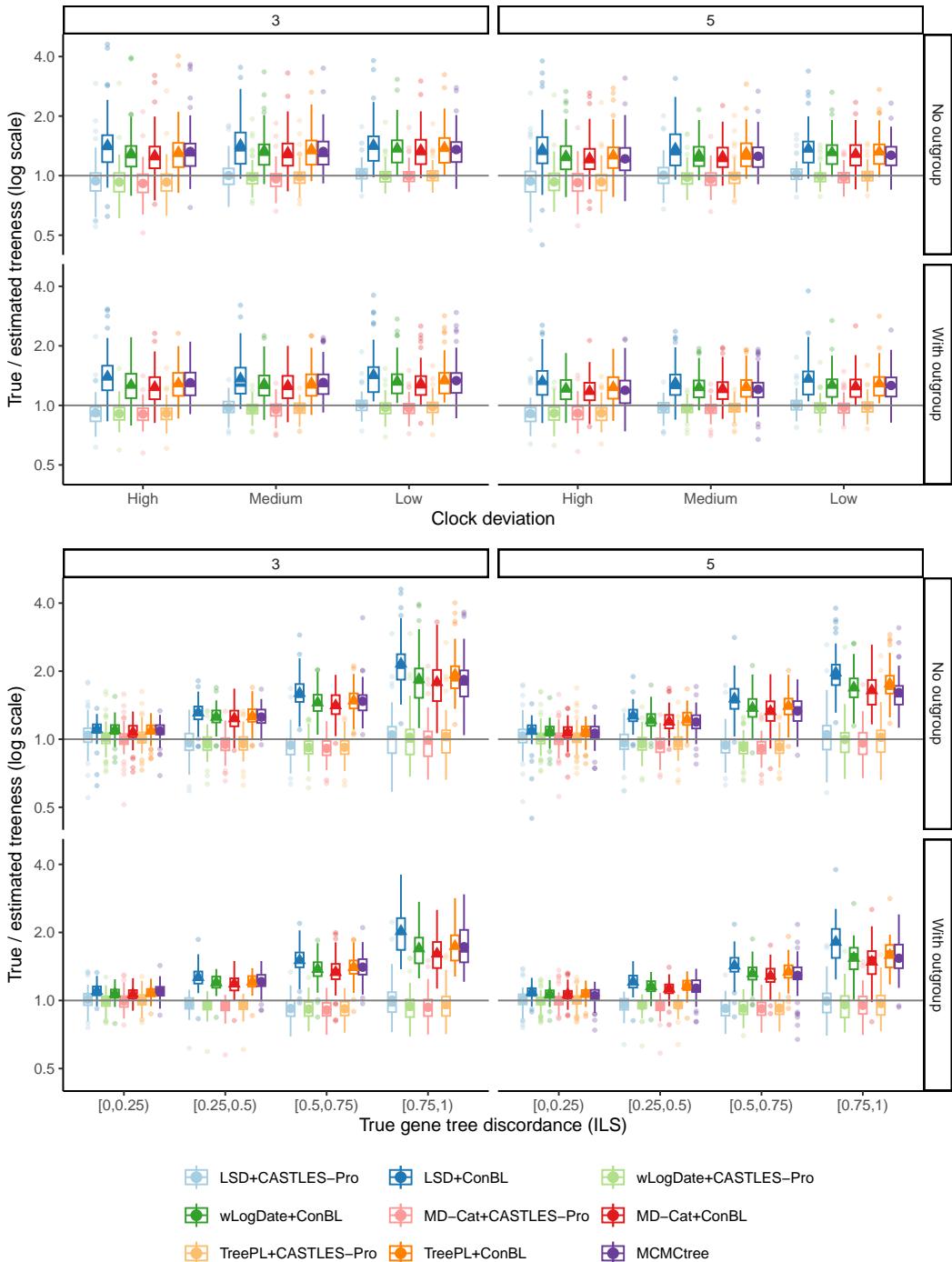


Figure 7.9: (**MVroot, Treeness**) True **treeness** (sum of **internal** branch lengths divided by sum of all branch lengths) divided by estimated **treeness** in log2 scale for trees estimated using different dating pipelines on the 30-taxon simulated datasets in the *root-fixed* experiments. The columns show the number of calibrations and the rows show conditions with or without outgroup. The number of genes is 500 and the results show mean and standard deviation in addition to boxplots across 100 replicates.

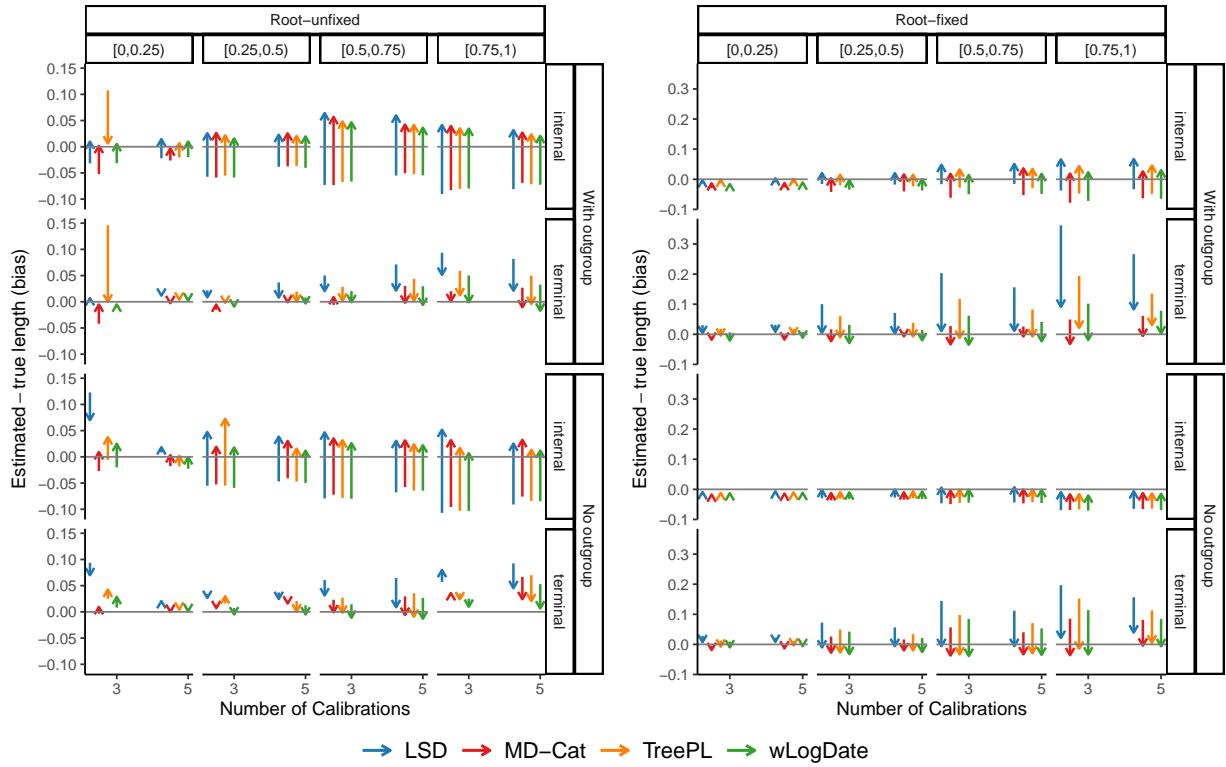


Figure 7.10: (**MVroot, root-fixed vs root-unfixed**) Reduction (or increase) in bias of branch lengths in time units using CASTLES-Pro instead of [ConBL](#) for branch length estimation in different ML-based dating pipelines. In conditions with outgroup, the error is calculated after excluding the outgroup. For each dating method, the direction of the arrow is from average bias in the final dated tree when branch length estimation is done using [ConBL](#) to the average bias when using CASTLES-Pro for branch length estimation. The results are shown on the 30-taxon simulated datasets for conditions with or without [outgroup](#) and in the *root-fixed* or *root-unfixed* experiments. The number of genes is 500 and the number of replicates is 100. The panels show different levels of ILS.

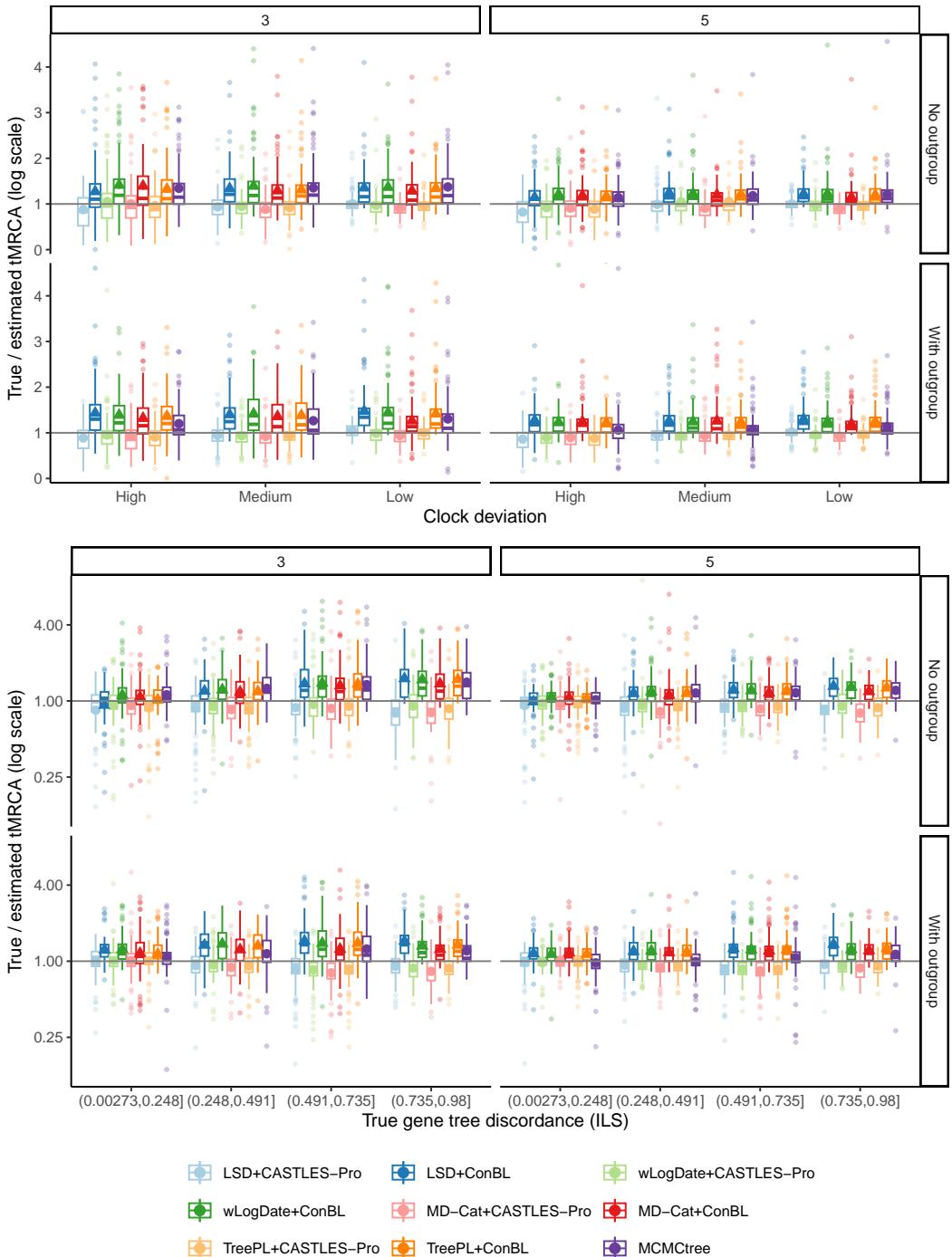


Figure 7.11: (**MVroot, tMRCA**) True time of the most recent common ancestor (tMRCA) of the ingroup divided by estimated tMRCA in log₂ scale for trees estimated using different dating pipelines on the 30-taxon simulated dataset in the *root-unfixed* experiments. The columns show the number of calibrations and the rows show conditions with or without outgroup. The number of genes is 500 and the results show mean and standard deviation in addition to boxplots across 100 replicates.

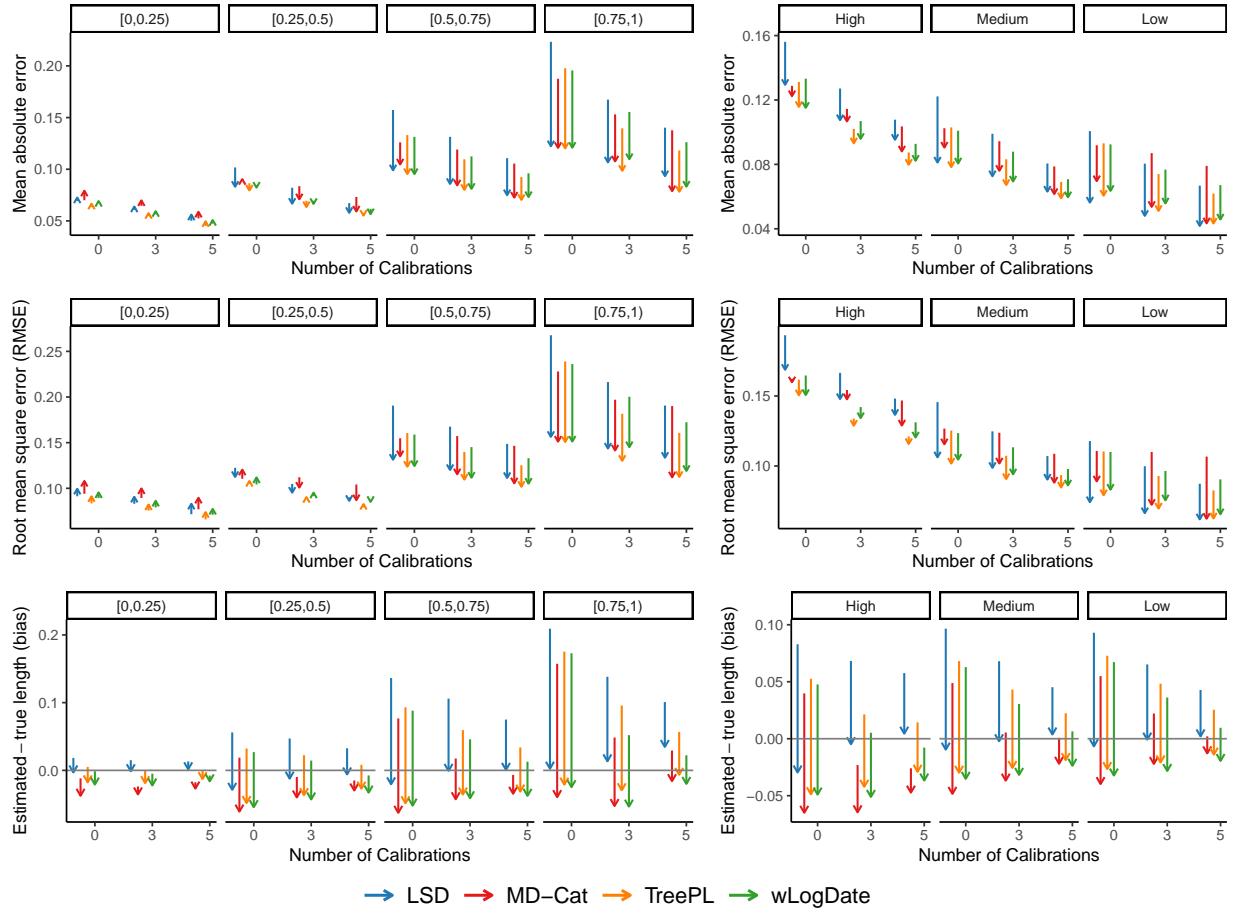


Figure 7.12: (**MVroot, Node age**) Reduction (or increase) in height-normalized mean absolute error, RMSE and bias of node ages using CASTLES-Pro instead of **ConBL** for branch length estimation in different ML-based dating pipelines. The direction of time is considered backward, so that all extant (terminal) **taxa** have age 0 and **internal** nodes have non-zero ages. For each dating method, the direction of the arrow is from average error in the final dated tree when branch length estimation is done using **ConBL** to the average error when using CASTLES-Pro for branch length estimation. The results are shown on the 30-taxon simulated datasets for conditions without **outgroup** and in the *root-fixed* experiments. The number of genes is 500 and the number of replicates is 100. In the left panels, the columns show different levels of **ILS** and in the right panels they show different levels of deviation from the clock.

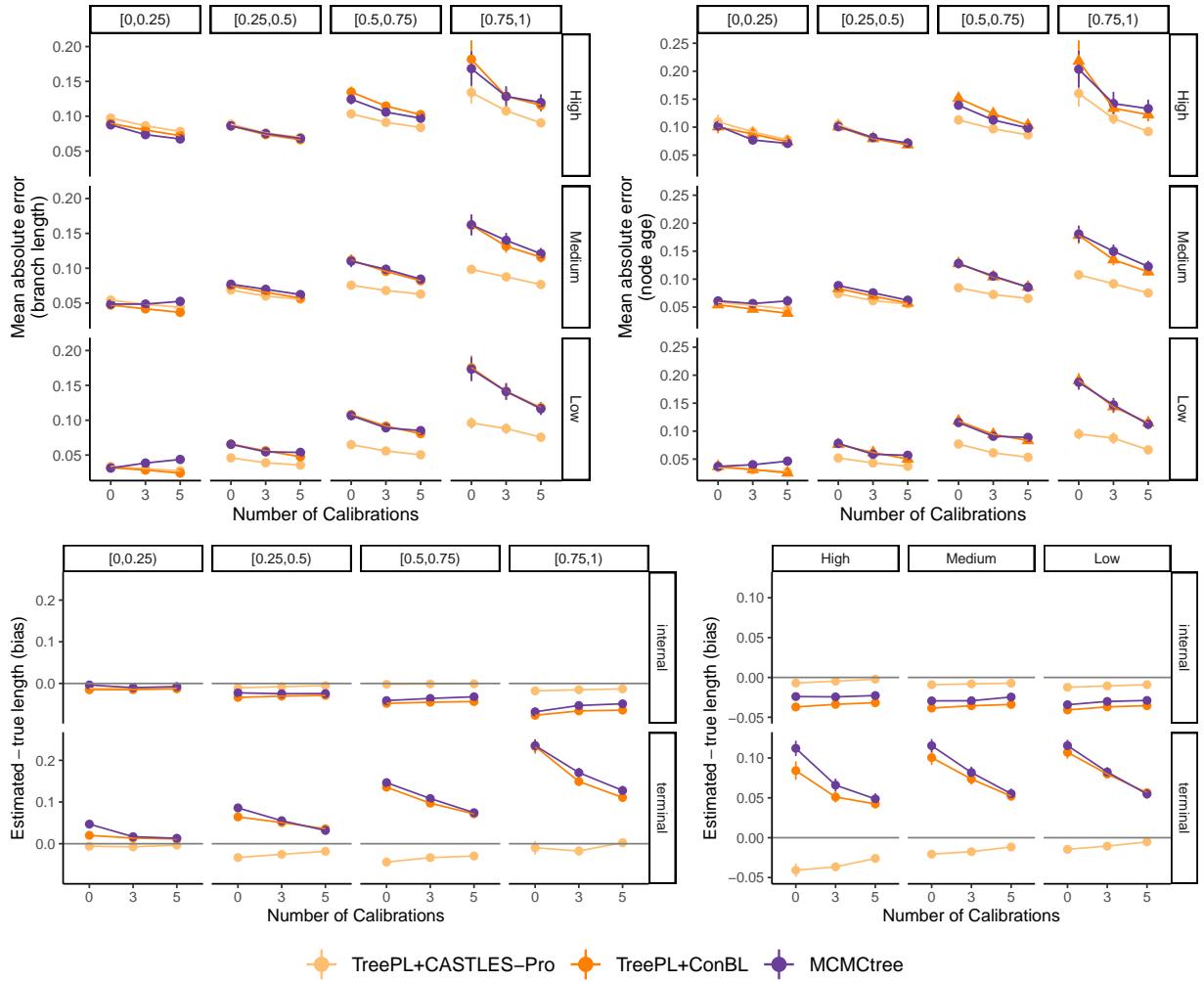


Figure 7.13: (**MVroot, MCMCTree**) Comparison between MCMCTree, TreePL+CASTLES-Pro and TreePL+ConBL in terms of height-normalized mean absolute error and bias of branch lengths and node ages in time unit on the 30-taxon simulated datasets for conditions without [outgroup](#) and in the *root-fixed* experiments. The number of genes is 500 and the results show mean and standard deviation across 100 replicates. For the bias plots, the left panel shows different levels of [ILS](#) and the right panel show different levels of deviation from the clock.

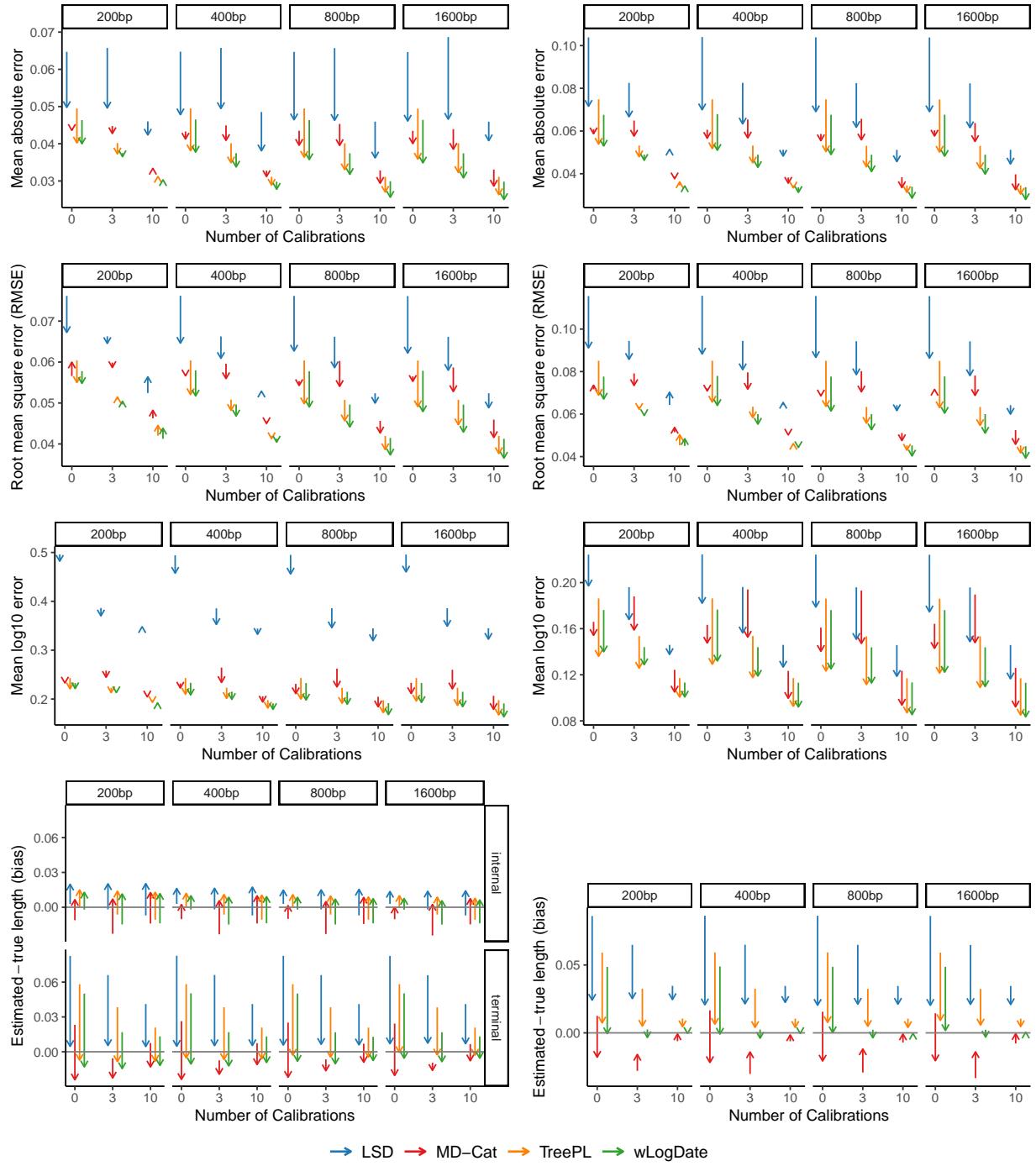


Figure 7.14: (**S100, Branch length and node age**) Reduction (or increase) in height-normalized mean absolute error, RMSE, mean log error and bias of branch lengths (left) and node ages (right) in time units using CASTLES-Pro instead of [ConBL](#) for branch length estimation in different ML-based dating pipelines on the 100-taxon simulated datasets in the *root-fixed* experiments. For node ages, the direction of time is considered backward, so that all extant (terminal) **taxa** have age 0 and **internal** nodes have non-zero ages. For each dating method, the direction of the arrow is from average error in the final dated tree when branch length estimation is done using [ConBL](#) to the average error when using CASTLES-Pro for branch length estimation. The number of genes is 1000 and the average level of **ILS** is 46% AD.

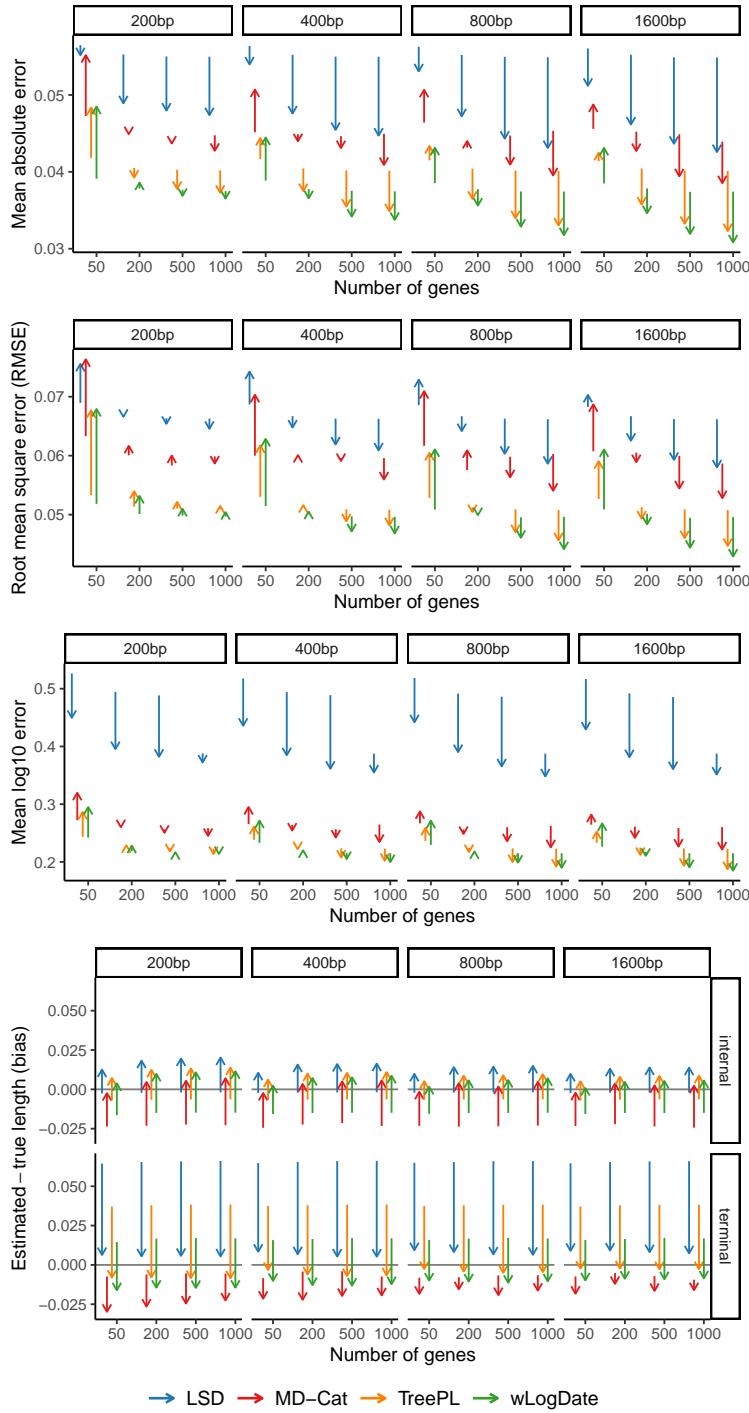


Figure 7.15: (**S100, Branch length**) Reduction in height-normalized mean absolute error, RMSE, mean log error and bias of branch lengths in time units estimated using different ML-based dating pipelines on the 100-taxon simulated datasets in the *root-fixed* experiments. The number of genes vary between 50 to 1000 (default: 1000) and the average level of ILS is 46% AD. For each dating method, the direction of the arrow is from average error in the final dated tree when branch length estimation is done using **ConBL** to the average error when using **CASTLES-Pro** for branch length estimation. The number of calibrations is 3.

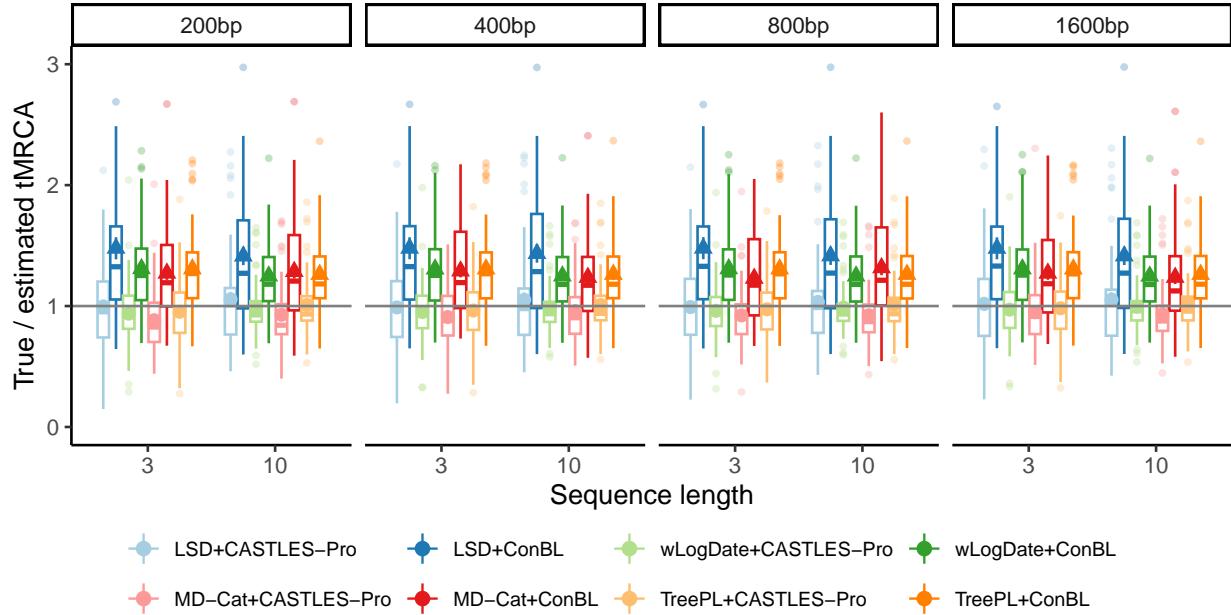


Figure 7.16: **(S100, tMRCA)** True time of the most recent common ancestor (tMRCA) divided by estimated tMRCA for trees estimated using different dating pipelines on the 100-taxon simulated datasets in the *root-unfixed* experiments. The columns show the sequence length. The number of genes is 1000 and the results show mean and standard deviation in addition to boxplots across 50 replicates.

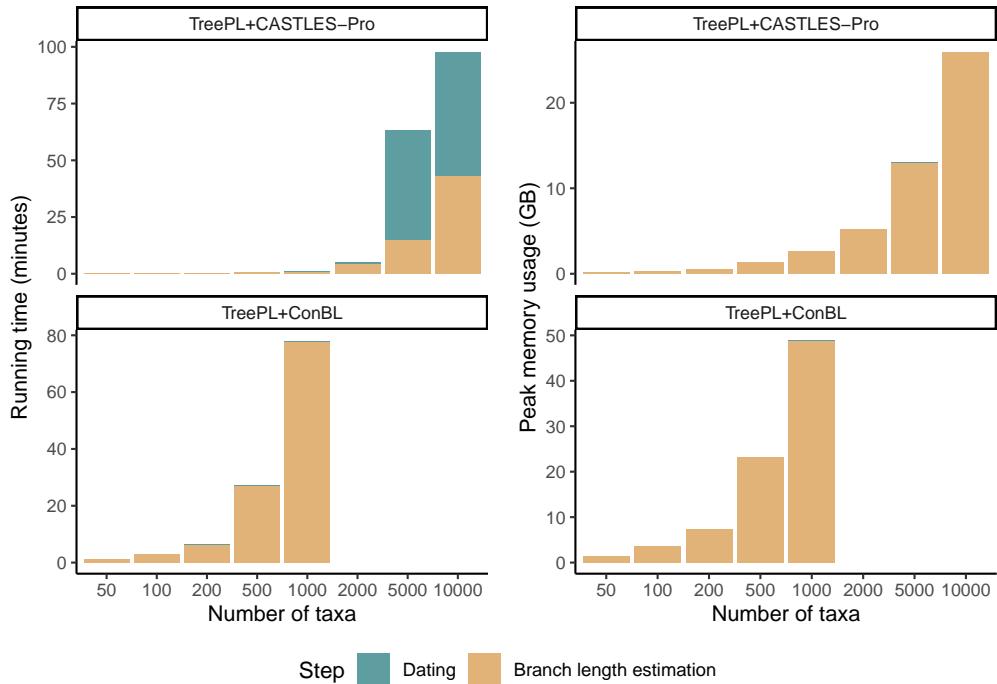


Figure 7.17: **(Large dataset, time and memory)** Runtime and memory separately reported for the branch length estimation and dating steps on the large dataset. The reported runtime does not include the time spent for gene tree estimation or species tree topology estimation, as these are assumed to be available beforehand. The results are averaged over 20 replicates in each model condition. ConBL fails on trees with more than 1000 taxa due to memory limit.

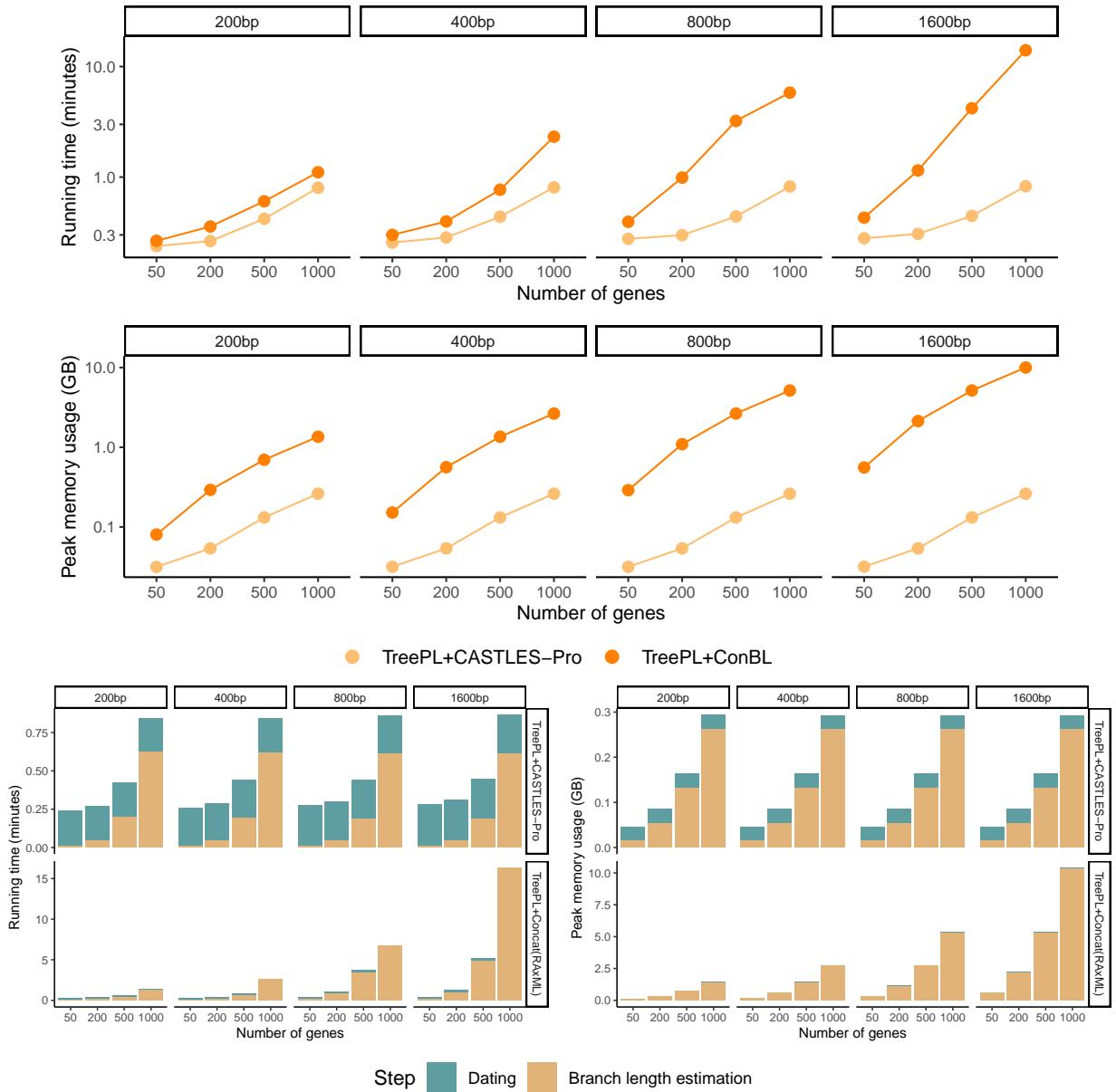


Figure 7.18: (**S100, time and memory**) (top) Average total running time of branch length estimation and dating in minutes and peak memory usage of the dating pipeline in gigabytes for TreePL with CASTLES-Pro or [ConBL](#) on the 100-taxon simulated datasets in the *root-fixed* experiments. The y-axes are shown in log-scale. (bottom) Runtime and memory separately reported for the branch length estimation and dating steps. The number of calibrations is 3. The reported runtime does not include the time spent for [gene tree](#) estimation or species tree topology estimation, as these are assumed to be available beforehand. The results are averaged over 50 replicates in each model condition.

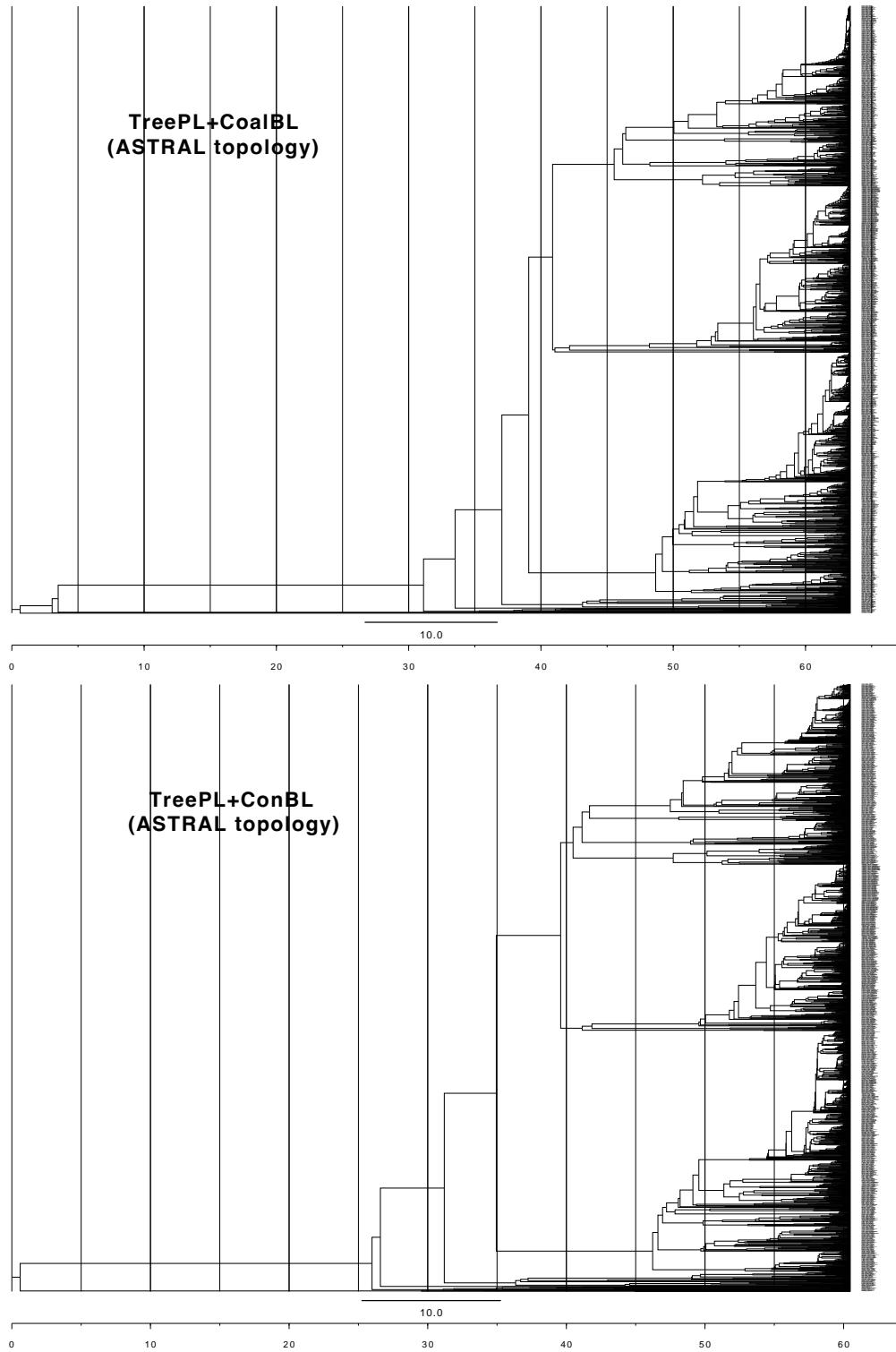


Figure 7.19: Suboscines time trees dated with TreePL using CASTLES-Pro or [ConBL SU](#) branch lengths drawn on the ASTRAL topology from the original study in [365]. The number of species is 1683, and the number of gene trees is 2389. Gene trees are resolved and are estimated from the minimally filtered T400F alignments. TreePL is run using four calibration points, as in the main study.

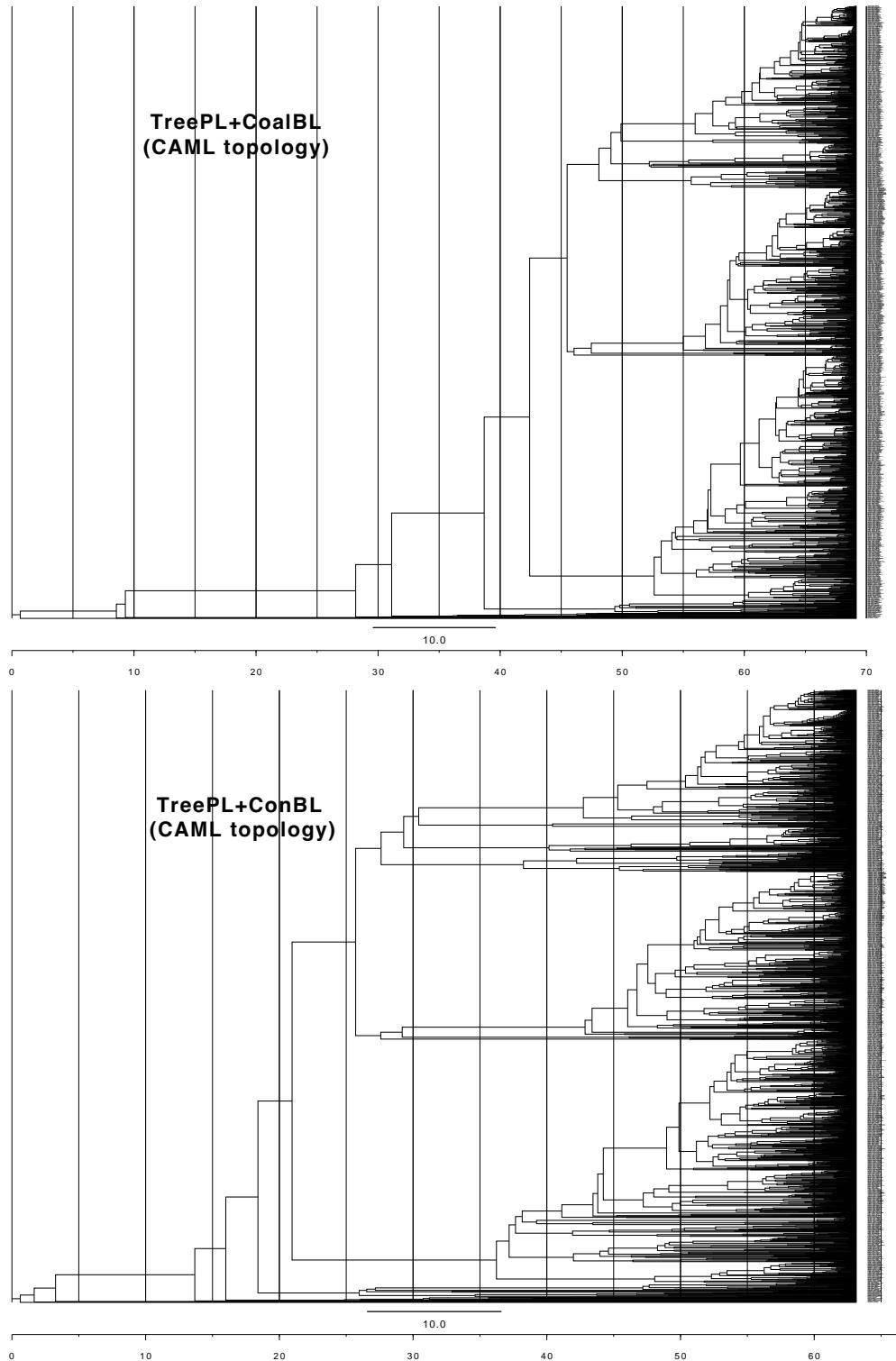


Figure 7.20: Suboscines time trees dated with TreePL using CASTLES-Pro or [ConBL SU](#) branch lengths, drawn on the concatenation topology from the original study in [365]. The number of species is 1684, and the number of gene trees is 2389. Gene trees are resolved and are estimated from the minimally filtered T400F alignments. TreePL is run using four calibration points, as in the main study.

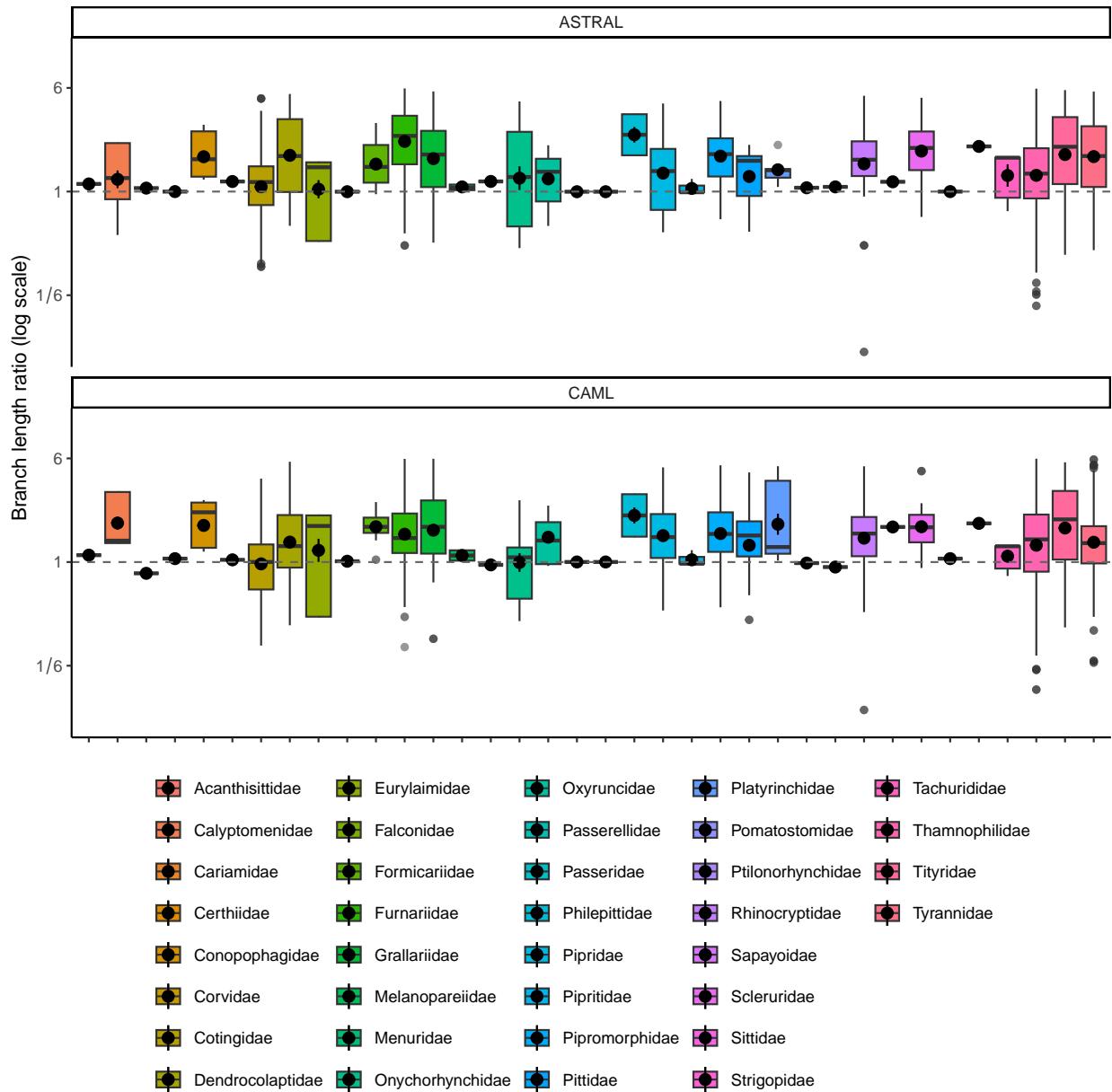


Figure 7.21: Dating the suboscines phylogeny [365] using TreePL. The panels show the two tree topologies from the main study, estimated using concatenation and ASTRAL. The y-axis shows terminal branch lengths of TreePL+ConBL divided by terminal branch lengths of TreePL+CASTLES-Pro in log2 scale, and the x-axis shows different suboscine families. The number of species is 1683 and the number of gene trees is 2389. Gene trees are resolved and are estimated from the minimally filtered T400F alignments. TreePL is run using four calibration points as in the original study. Dotted line indicates 1.

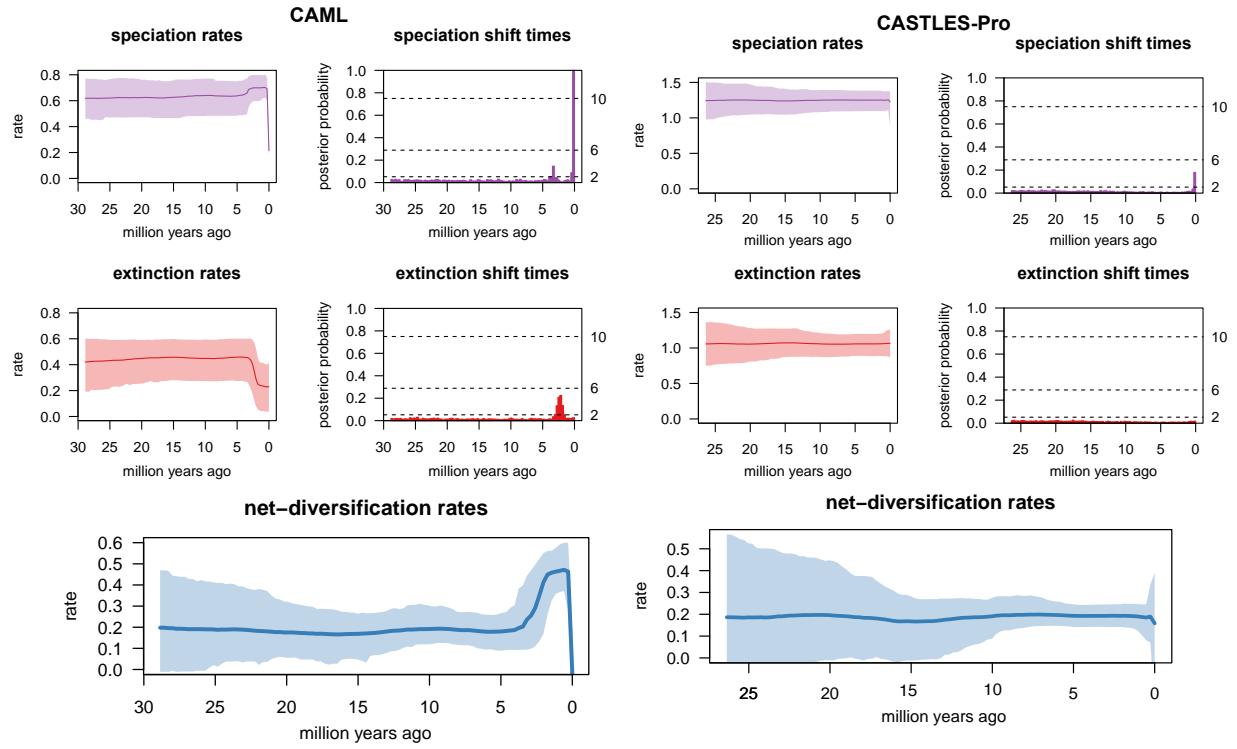


Figure 7.22: Diversification analysis on the suboscines dataset of [365]. Dating is done using TreePL with four calibration points, as in the original study. The plots show diversification rates, including speciation and extinction rates and speciation and extinction shift times for the ASTRAL topology from the original study (after removing the `outgroup` clade) based on branch lengths estimated using ConBL (labeled CAML) or CoalBL (labeled CASTLES-Pro).

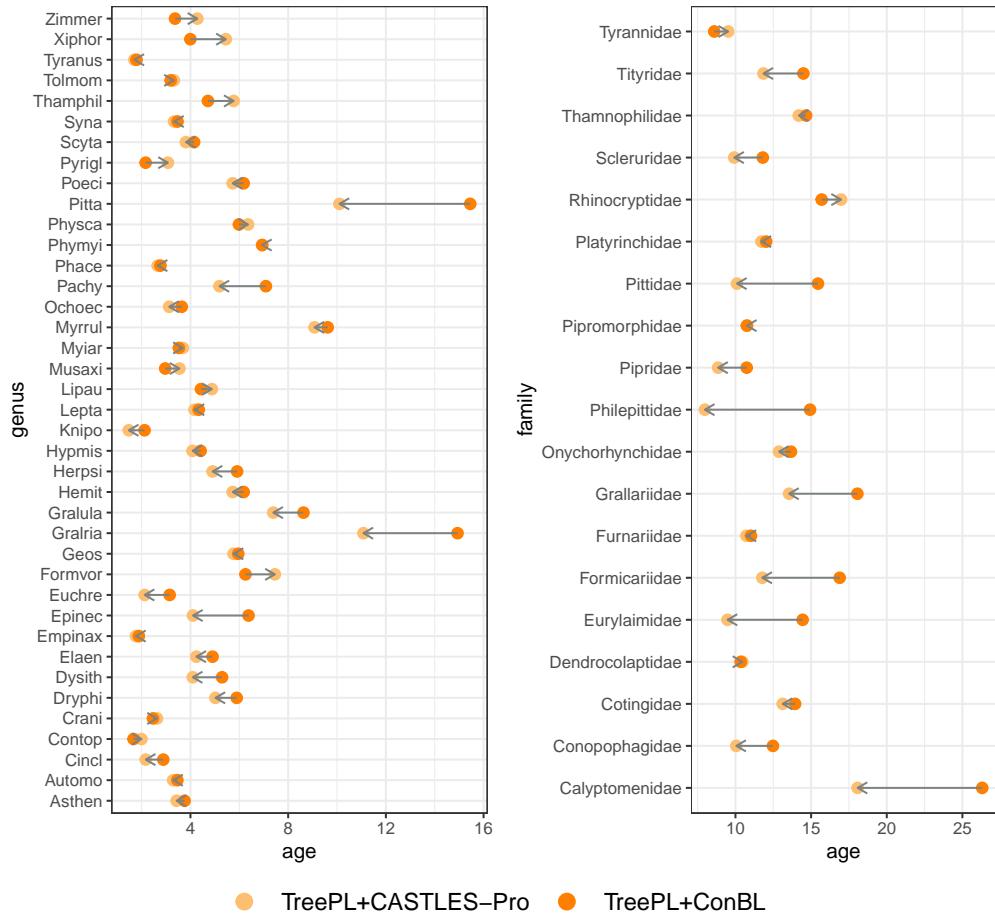


Figure 7.23: Dating the suboscines [phylogeny](#) [365] using TreePL on the concatenation topology from the original study. Age of the genera that have more than 10 representative species (left), and age of families with at least two representative genera (right) estimated by TreePL+CASTLES-Pro and TreePL+ConBL. The direction of the arrow is from ConBL-based dates to CASTLES-Pro dates.

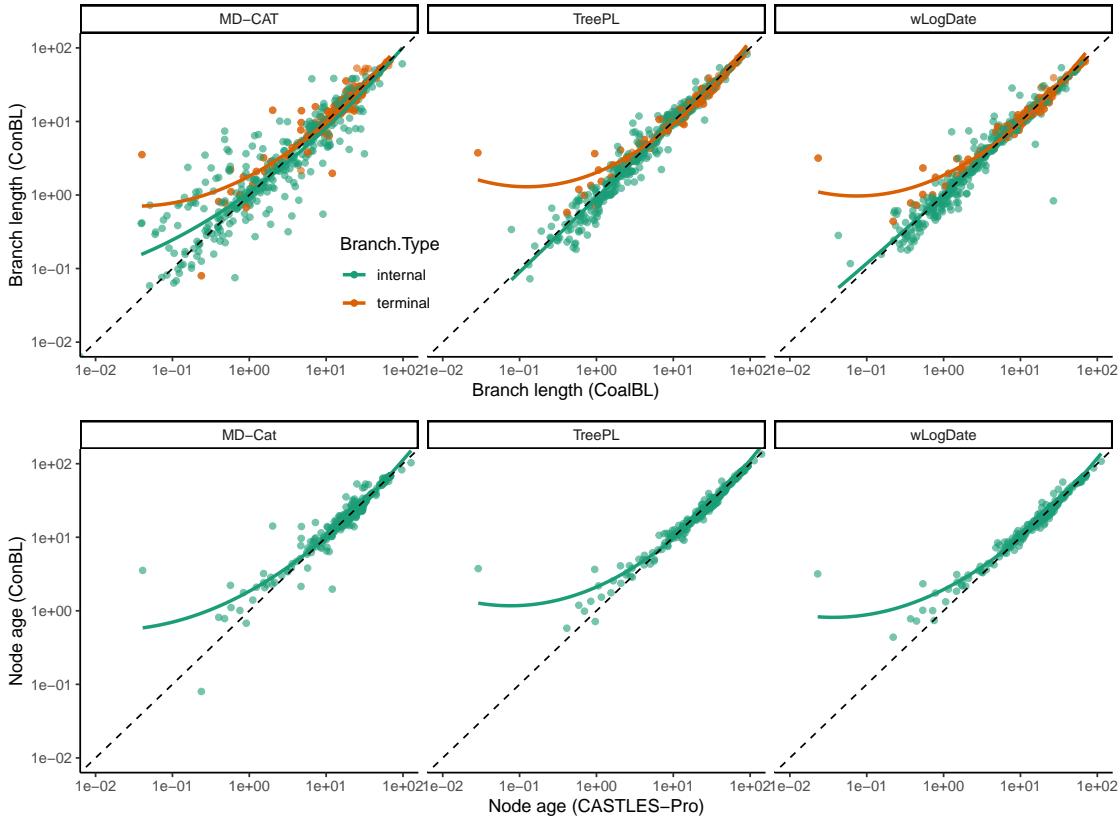


Figure 7.24: Correlation between time-unit branch lengths (top) and node ages (bottom) of different ML-based dating pipelines using CASTLES-Pro or `ConBL` on the 363-taxon neaoavian phylogeny from [328]. The tree topology is the main ASTRAL topology from the original study estimated from 63K gene trees. MD-Cat, TreePL, and wLogDate are run with fixed calibration points using median quantiles of [calibration densities](#) from the main study. The direction of time in the bottom plot is considered backward so that all extant (terminal) [taxa](#) have age 0 and [internal](#) nodes have non-zero ages, and so only the age of [internal](#) nodes are shown. Both axes are in log scale.

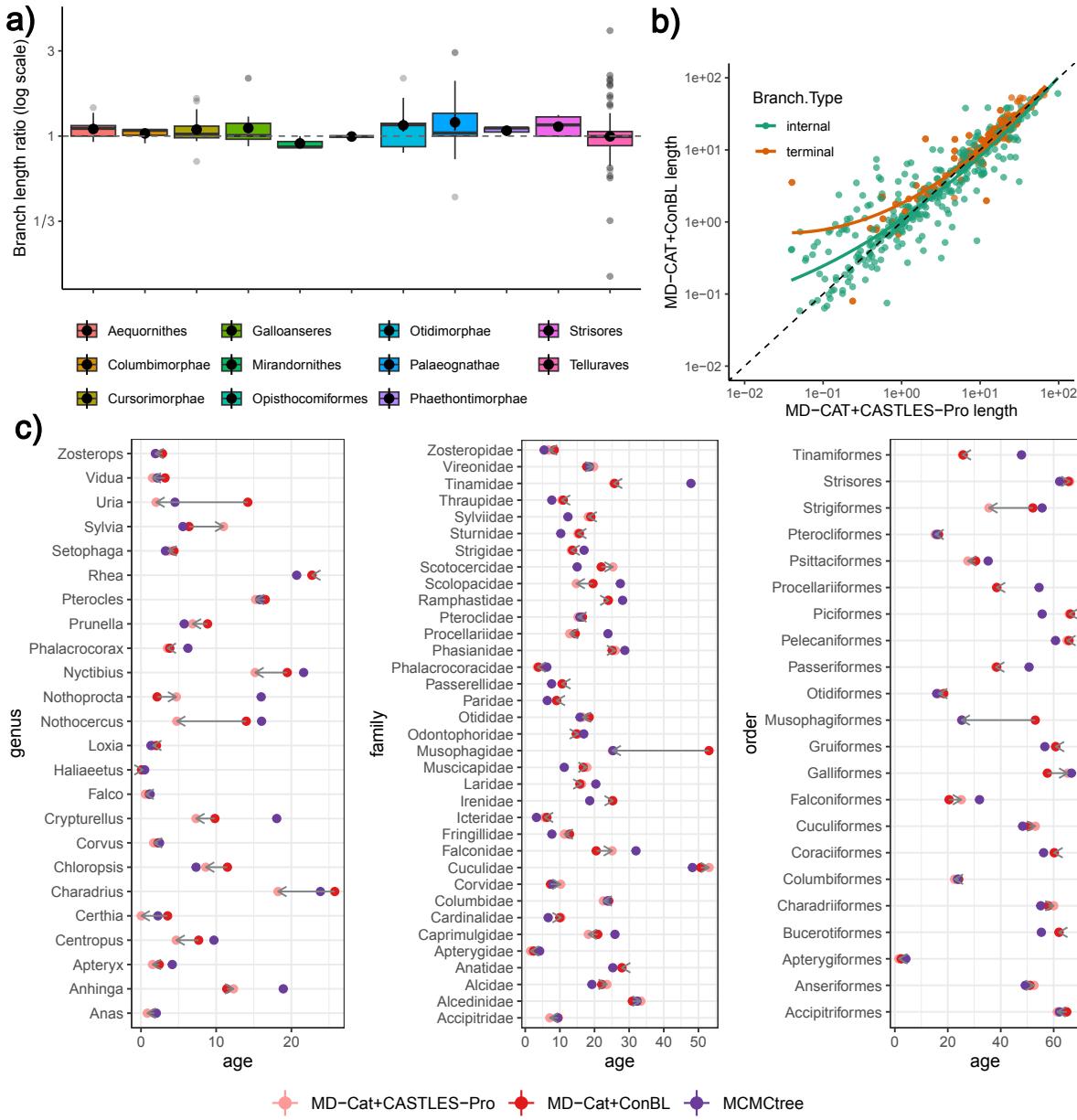


Figure 7.25: Results of dating the 363-taxon neoavian phylogeny [328] using MD-Cat on an ASTRAL topology from the original study. a) terminal branch lengths of MD-Cat+ConBL divided by terminal branch lengths of MD-Cat+CASTLES-Pro in log2 scale across 11 higher order clades. b) Correlation between all branch lengths estimated by MD-Cat+ConBL and MD-Cat+CASTLES-Pro. c) Age of the genera that have at least two representative species, and age of families and orders with at least three representative species estimated by MD-Cat+CASTLES-Pro, MD-Cat+ConBL, as well as MCMCTree analysis from the original study. The direction of the arrow is from ConBL dates to CASTLES-Pro dates.

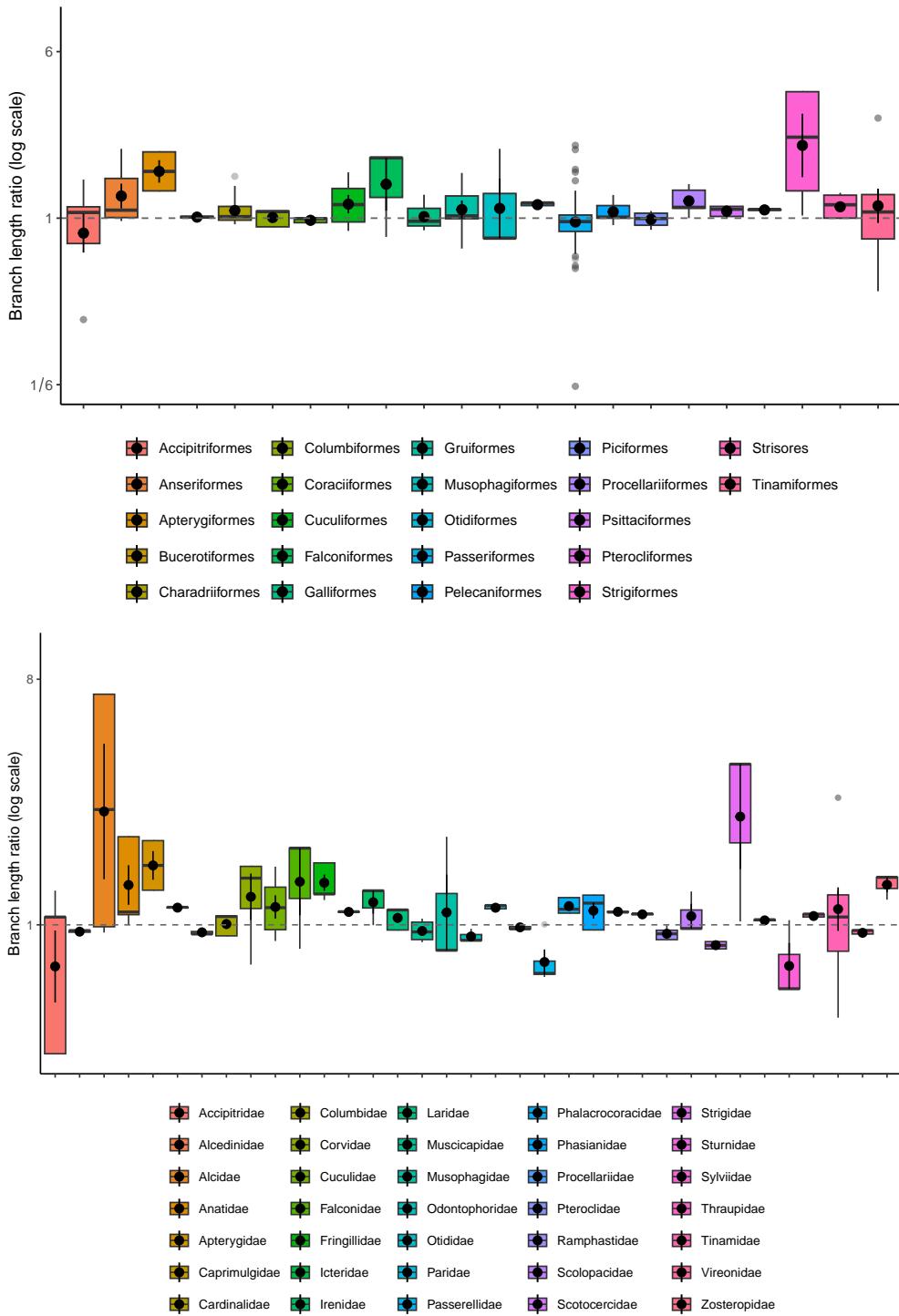


Figure 7.26: Ratio between [terminal](#) branch lengths of trees dated using MD-Cat based on [ConBL](#) and [CoalBL](#) for the 363-taxon neoavian [phylogeny](#) [328] on the ASTRAL topology from the original study. The results show [terminal](#) branch lengths of MD-Cat+ConBL divided by [terminal](#) branch lengths of MD-Cat+CASTLES-Pro across orders (top) and families (bottom) with more than two representative species in the log₂ scale. The panels show the mean (shown with a circle) and standard deviation in addition to boxplots.

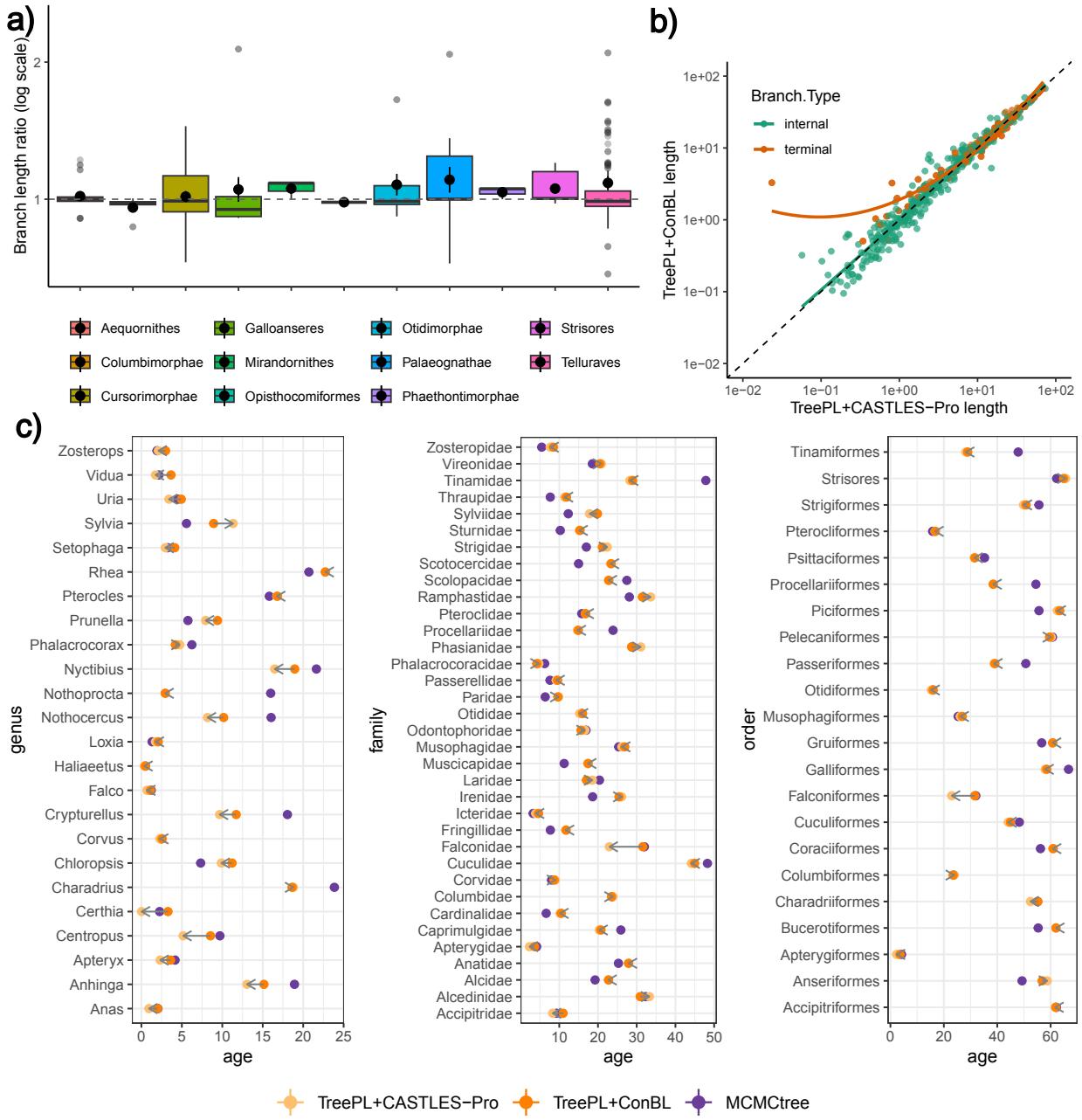


Figure 7.27: Results of dating the 363-taxon neoavian phylogeny [328] using TreePL on an ASTRAL topology from the original study. a) **terminal** branch lengths of TreePL+ConBL divided by **terminal** branch lengths of TreePL+CASTLES-Pro in log2 scale across 11 higher order clades. b) Correlation between all branch lengths estimated by TreePL+ConBL and TreePL+CASTLES-Pro. c) Age of the genera that have at least two representative species, and age of families and orders with at least three representative species estimated by TreePL+CASTLES-Pro, TreePL+ConBL as well as MCMCTree analysis from the original study (which used a subset of loci used in other analyses, and hence is not directly comparable). The direction of the arrow is from **ConBL** dates to **CASTLES-Pro** dates.

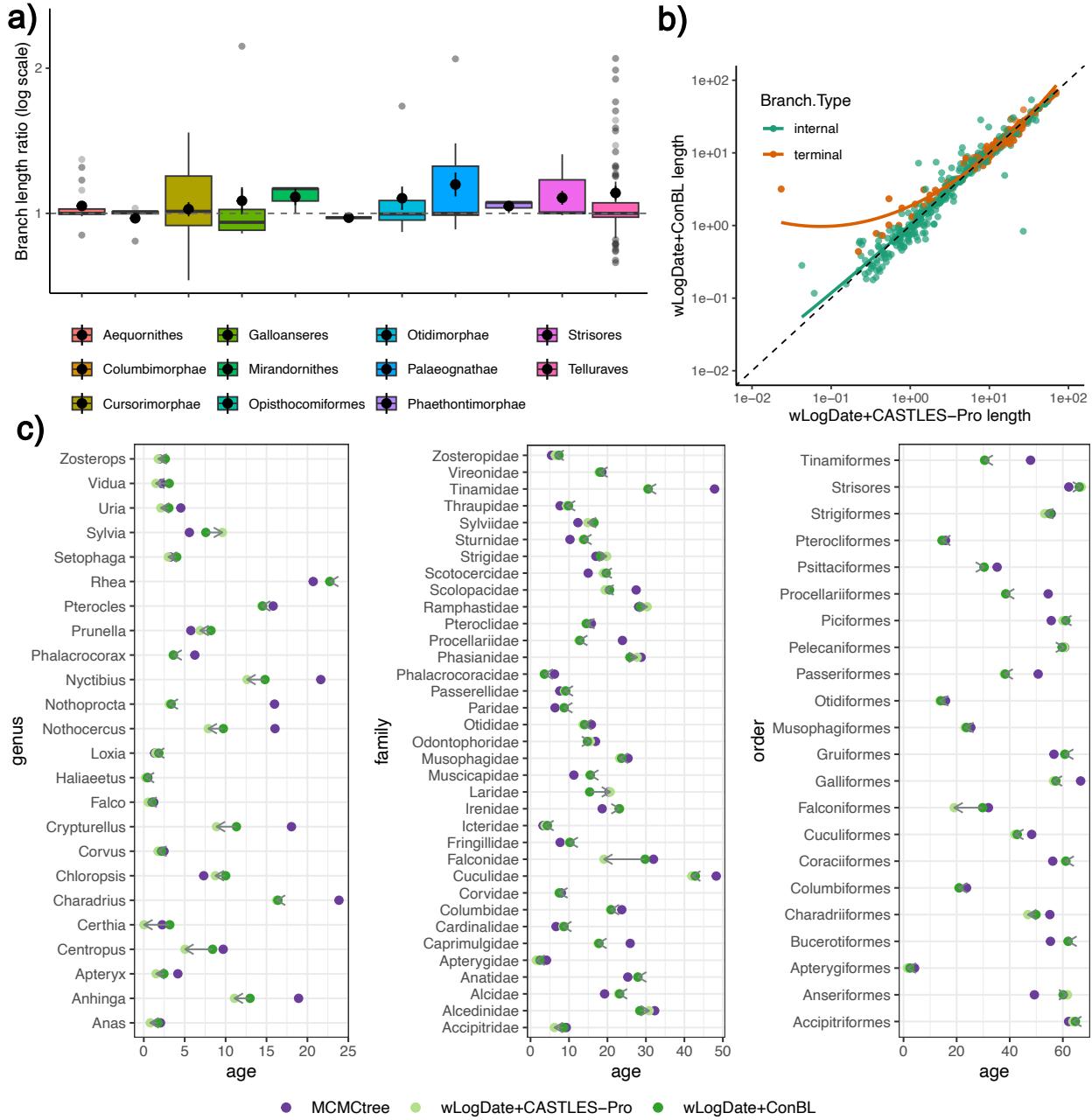


Figure 7.28: Results of dating the 363-taxon neoavian phylogeny [328] using wLogDate on an ASTRAL topology from the original study. a) terminal branch lengths of wLogDate+ConBL divided by terminal branch lengths of wLogDate+CASTLES-Pro in log2 scale across 11 higher order clades. b) Correlation between all branch lengths estimated by wLogDate+ConBL and wLogDate+CASTLES-Pro. c) Age of the genera that have at least two representative species, and age of families and orders with at least three representative species estimated by wLogDate+CASTLES-Pro, wLogDate+ConBL as well as MCMCTree analysis from the original study. The direction of the arrow is from ConBL dates to CASTLES-Pro dates.

CHAPTER 8: CONCLUSIONS

In this dissertation, we introduced four novel computational methods for rooting and branch length estimation—QR, QR-STAR, CASTLES, and CASTLES-Pro—as well as a new scalable pipeline for estimating divergence times on species trees that account for gene tree discordance and can be used alongside existing summary methods such as ASTRAL in large-scale phylogenomic analyses. All of these methods are summary methods, in that they combine information from a set of gene trees, and have strong theoretical foundations based on the [Multi-Species Coalescent \(MSC\)](#) model.

We first presented Quintet Rooting (QR), a summary method for rooting species trees that uses the signal from discordance due to [Incomplete lineage sorting \(ILS\)](#) in a collection of unrooted gene trees to find the position of the root in an unrooted species tree. QR is based on a theoretical result by Allman, Degnan, and Rhodes [81], which establishes that rooted five-taxon species trees are identifiable from the distribution of unrooted gene trees under the multi-species coalescent. We evaluated QR on simulated datasets with varying levels of gene tree error and on an avian biological dataset, showing that it generally outperforms alternative methods under moderate [ILS](#), except when gene tree estimation error is very high. We then showed that QR is not guaranteed to be statistically consistent under the [MSC](#), and presented QR-STAR, an extension that includes an additional step to infer the topology of the rooted form for each five-taxon species subtree. We proved the statistical consistency of QR-STAR and provided a sample complexity analysis. Simulations demonstrated that QR-STAR improves over QR, particularly under low [ILS](#), and achieves an accuracy close to that of the optimal rooting under moderate to high [ILS](#).

We next presented CASTLES, a summary method for estimating branch lengths of a species tree in substitution units given a set of unrooted gene trees with branch lengths. CASTLES is based on derivations of expected gene tree branch lengths under the coalescent theory, and therefore explicitly models incomplete lineage sorting. Extensive simulations and analysis of a mammalian biological dataset show that CASTLES outperforms existing branch length estimation methods under various model conditions. We then presented CASTLES-Pro, an extension of CASTLES that supports multi-copy gene family trees and incorporates improved analytical formulas, resulting in better performance even for single-copy data. We evaluated CASTLES-Pro on datasets with [ILS](#), [GDL](#), and [HGT](#), and our results show that it provides more accurate results compared to other methods for all three sources of discordance. We also applied CASTLES-Pro to a range of biological datasets, from deep evolutionary splits to recent speciations, and our results demonstrate its ability to reduce

biases introduced by concatenation. Finally, we presented a four-step scalable pipeline for dating species trees that accounts for gene tree discordance due to [ILS](#), and our results in simulations and on biological datasets show that this pipeline can provide more accurate estimation of divergence times and corrects biases in downstream diversification analyses.

Overall, this dissertation makes several novel contributions in computational approaches to phylogenomics, and provide tools that can be useful for different types of biological analysis. The methods presented in this dissertation are all scalable summary methods, and generally more accurate than their counterparts when they can get enough signal from the sources of gene tree discordance they are considering. The methods presented here are all implemented in open-source software and are available on Github for use by the biological research community. In particular, CASTLES and CASTLES-Pro are implemented in the ASTER package of tools [316] and can be used with ASTRAL and ASTRAL-Pro, popular tools for species tree estimation from single-copy or multi-copy gene trees, respectively. In addition, the contributions of this dissertation make novel theoretical advances in the field of computational phylogenomics, by introducing new theoretical frameworks for coalescent-based post-species tree analysis and proofs of statistical consistency and sample complexity for some of the methods introduced in this work.

Throughout this dissertation, we have highlighted directions for future work related to each topic; here, we summarize broad avenues for future research and method development.

Although this dissertation focuses on species *trees*, the methods developed here can be extended to species *networks* by incorporating additional sources of gene tree discordance, such as hybridization and recombination, which give rise to reticulate evolutionary histories. Future directions include developing theoretical frameworks for rooting, branch length estimation, and estimating divergence times under the [Network Multi-Species Coalescent \(NMSC\)](#) model [122], an extension of the [MSC](#) that accounts for reticulate events, as well as designing scalable algorithms and software for estimating these parameters on species networks. Because of the complexity introduced by reticulate evolutionary events, these frameworks can be substantially more challenging to develop than those explored in this dissertation [94].

All methods developed in this dissertation have reduced accuracy in the presence of gene tree estimation error and missing data. This effect is especially pronounced for our proposed rooting methods, which rely on signal from incomplete lineage sorting; in such cases, noise introduced by gene tree error can substantially distort this signal and reduce the rooting accuracy. Developing techniques to detect and mitigate gene tree estimation error, as well as other systematic biases that distort gene tree distributions, is an important direction for future research that could improve the robustness of these methods and pipelines.

REFERENCES

- [1] F. Delsuc, H. Brinkmann, and H. Philippe, “Phylogenomics and the reconstruction of the tree of life,” *Nature Reviews Genetics*, vol. 6, no. 5, pp. 361–375, 2005.
- [2] P. Kapli, Z. Yang, and M. J. Telford, “Phylogenetic tree building in the genomic age,” *Nature Reviews Genetics*, vol. 21, no. 7, pp. 428–444, 2020.
- [3] A. D. Young and J. P. Gillung, “Phylogenomics—principles, opportunities and pitfalls of big-data phylogenetics,” *Systematic Entomology*, vol. 45, no. 2, pp. 225–247, 2020.
- [4] R. C. Hardison, “Comparative genomics,” *PLoS biology*, vol. 1, no. 2, p. e58, 2003.
- [5] S. M. Philpott, W. J. Arendt, I. Armbrecht, P. Bichier, T. V. Diestch, C. Gordon et al., “Biodiversity loss in latin american coffee landscapes: review of the evidence on ants, birds, and trees,” *Conservation Biology*, vol. 22, no. 5, pp. 1093–1105, 2008.
- [6] S. V. Edwards, V. Robin, N. Ferrand, and C. Moritz, “The evolution of comparative phylogeography: putting the geography (and more) into comparative population genomics,” *Genome Biology and Evolution*, vol. 14, no. 1, p. evab176, 2022.
- [7] B. G. Hall and M. Barlow, “Phylogenetic analysis as a tool in molecular epidemiology of infectious diseases,” *Annals of epidemiology*, vol. 16, no. 3, pp. 157–169, 2006.
- [8] W. M. Fitch and E. Margoliash, “Construction of phylogenetic trees: a method based on mutation distances as estimated from cytochrome c sequences is of general applicability.” *Science*, vol. 155, no. 3760, pp. 279–284, 1967.
- [9] J. J. Wiens, “The role of morphological data in phylogeny reconstruction,” *Systematic biology*, vol. 53, no. 4, pp. 653–661, 2004.
- [10] A. Rokas, B. L. Williams, N. King, and S. B. Carroll, “Genome-scale approaches to resolving incongruence in molecular phylogenies,” *Nature*, vol. 425, no. 6960, pp. 798–804, 2003.
- [11] D. Posada, “Phylogenomics for systematic biology,” *Systematic biology*, vol. 65, no. 3, pp. 353–356, 2016.
- [12] C. Zhang, R. Nielsen, and S. Mirarab, “CASTER: Direct species tree inference from whole-genome alignments,” *Science*, p. eadk9688, Jan. 2025.
- [13] J. H. Degnan and N. A. Rosenberg, “Gene tree discordance, phylogenetic inference and the multispecies coalescent,” *Trends in ecology & evolution*, vol. 24, no. 6, pp. 332–340, 2009.
- [14] L. L. Knowles, “Estimating species trees: methods of phylogenetic analysis when there is incongruence across genes,” *Systematic biology*, vol. 58, no. 5, pp. 463–467, 2009.

- [15] W. P. Maddison, “Gene trees in species trees,” *Systematic biology*, vol. 46, no. 3, pp. 523–536, 1997.
- [16] N. Galtier and V. Daubin, “Dealing with incongruence in phylogenomic analyses,” *Philosophical Transactions of the Royal Society B: Biological Sciences*, vol. 363, no. 1512, pp. 4023–4029, 2008.
- [17] J. L. Steenwyk, Y. Li, X. Zhou, X.-X. Shen, and A. Rokas, “Incongruence in the phylogenomics era,” *Nature Reviews Genetics*, vol. 24, no. 12, pp. 834–850, 2023.
- [18] S. Mirarab, L. Nakhleh, and T. Warnow, “Multispecies coalescent: theory and applications in phylogenetics,” *Annual Review of Ecology, Evolution, and Systematics*, vol. 52, pp. 247–268, 2021.
- [19] D. H. Huson and D. Bryant, “Application of phylogenetic networks in evolutionary studies,” *Molecular biology and evolution*, vol. 23, no. 2, pp. 254–267, 2006.
- [20] R. L. Elworth, H. A. Ogilvie, J. Zhu, and L. Nakhleh, “Advances in computational methods for phylogenetic networks in the presence of hybridization,” *Bioinformatics and phylogenetics: seminal contributions of Bernard Moret*, pp. 317–360, 2019.
- [21] W. P. Maddison and L. L. Knowles, “Inferring phylogeny despite incomplete lineage sorting,” *Systematic biology*, vol. 55, no. 1, pp. 21–30, 2006.
- [22] C. Scornavacca and N. Galtier, “Incomplete lineage sorting in mammalian phylogenomics,” *Systematic biology*, vol. 66, no. 1, pp. 112–120, 2017.
- [23] I. Rivas-González, M. Rousselle, F. Li, L. Zhou, J. Y. Dutheil, K. Munch, Y. Shao, D. Wu, M. H. Schierup, and G. Zhang, “Pervasive incomplete lineage sorting illuminates speciation and selection in primates,” *Science*, vol. 380, no. 6648, p. eabn4409, 2023.
- [24] E. D. Jarvis, S. Mirarab, A. J. Aberer, B. Li, P. Houde, C. Li, S. Y. Ho, B. C. Faircloth, B. Nabholz, J. T. Howard et al., “Whole-genome analyses resolve early branches in the tree of life of modern birds,” *Science*, vol. 346, no. 6215, pp. 1320–1331, 2014.
- [25] A. Stamatakis, “RAxML version 8: a tool for phylogenetic analysis and post-analysis of large phylogenies,” *Bioinformatics*, vol. 30, no. 9, pp. 1312–1313, 2014.
- [26] L.-T. Nguyen, H. A. Schmidt, A. Von Haeseler, and B. Q. Minh, “IQ-TREE: a fast and effective stochastic algorithm for estimating maximum-likelihood phylogenies,” *Molecular biology and evolution*, vol. 32, no. 1, pp. 268–274, 2015.
- [27] N. Saitou and M. Nei, “The neighbor-joining method: a new method for reconstructing phylogenetic trees.” *Molecular biology and evolution*, vol. 4, no. 4, pp. 406–425, 1987.
- [28] S. Roch and M. Steel, “Likelihood-based tree reconstruction on a concatenation of aligned sequence data sets can be statistically inconsistent,” *Theoretical population biology*, vol. 100, pp. 56–62, 2015.

- [29] S. Roch, M. Nute, and T. Warnow, “Long-branch attraction in species tree estimation: inconsistency of partitioned likelihood and topology-based summary methods,” *Systematic Biology*, vol. 68, no. 2, pp. 281–297, 2019.
- [30] L. S. Kubatko and J. H. Degnan, “Inconsistency of phylogenetic estimates from concatenated data under coalescence,” *Systematic biology*, vol. 56, no. 1, pp. 17–24, 2007.
- [31] M. S. Bayzid and T. Warnow, “Naive binning improves phylogenomic analyses,” *Bioinformatics*, vol. 29, no. 18, pp. 2277–2284, 2013.
- [32] E. K. Molloy and T. Warnow, “To include or not to include: the impact of gene filtering on species tree estimation methods,” *Systematic biology*, vol. 67, no. 2, pp. 285–303, 2018.
- [33] S. Mirarab, M. S. Bayzid, and T. Warnow, “Evaluating summary methods for multilocus species tree estimation in the presence of incomplete lineage sorting,” *Systematic Biology*, vol. 65, no. 3, pp. 366–380, 2016.
- [34] J. Chou, A. Gupta, S. Yaduvanshi, R. Davidson, M. Nute, S. Mirarab, and T. Warnow, “A comparative study of SVDquartets and other coalescent-based species tree estimation methods,” *BMC genomics*, vol. 16, no. 10, pp. 1–11, 2015.
- [35] J. F. Kingman, “On the genealogy of large populations,” *Journal of applied probability*, vol. 19, no. A, pp. 27–43, 1982.
- [36] B. Rannala and Z. Yang, “Bayes estimation of species divergence times and ancestral population sizes using dna sequences from multiple loci,” *Genetics*, vol. 164, no. 4, pp. 1645–1656, 2003.
- [37] J. Heled and A. J. Drummond, “Bayesian inference of species trees from multilocus data,” *Molecular biology and evolution*, vol. 27, no. 3, pp. 570–580, 2009.
- [38] H. A. Ogilvie, R. R. Bouckaert, and A. J. Drummond, “StarBEAST2 brings faster species tree inference and accurate estimates of substitution rates,” *Molecular biology and evolution*, vol. 34, no. 8, pp. 2101–2114, 2017.
- [39] M. Rabiee and S. Mirarab, “QuCo: quartet-based co-estimation of species trees and gene trees,” *Bioinformatics*, vol. 38, no. Supplement_1, pp. i413–i421, 2022.
- [40] H. A. Ogilvie, J. Heled, D. Xie, and A. J. Drummond, “Computational performance and statistical accuracy of *BEAST and comparisons with other methods,” *Systematic biology*, vol. 65, no. 3, pp. 381–396, 2016.
- [41] T. Zimmermann, S. Mirarab, and T. Warnow, “BBCA: Improving the scalability of *BEAST using random binning,” *BMC genomics*, vol. 15, no. 6, pp. 1–9, 2014.
- [42] J. Chifman and L. Kubatko, “Quartet inference from SNP data under the coalescent model,” *Bioinformatics*, vol. 30, no. 23, pp. 3317–3324, 2014.

- [43] D. Bryant, R. Bouckaert, J. Felsenstein, N. A. Rosenberg, and A. RoyChoudhury, “Inferring species trees directly from biallelic genetic markers: bypassing gene trees in a full coalescent analysis,” *Molecular biology and evolution*, vol. 29, no. 8, pp. 1917–1932, 2012.
- [44] N. De Maio, C. Schlötterer, and C. Kosiol, “Linking great apes genome evolution across time scales using polymorphism-aware phylogenetic models,” *Molecular biology and evolution*, vol. 30, no. 10, pp. 2249–2262, 2013.
- [45] M. Wascher and L. Kubatko, “Consistency of svdquartets and maximum likelihood for coalescent-based species tree estimation,” *Systematic biology*, vol. 70, no. 1, pp. 33–48, 2021.
- [46] L. Liu, L. Yu, and S. V. Edwards, “A maximum pseudo-likelihood approach for estimating species trees under the coalescent model,” *BMC evolutionary biology*, vol. 10, pp. 1–18, 2010.
- [47] L. Liu and L. Yu, “Estimating species trees from unrooted gene trees,” *Systematic biology*, vol. 60, no. 5, pp. 661–667, 2011.
- [48] S. Mirarab, R. Reaz, M. S. Bayzid et al., “ASTRAL: genome-scale coalescent-based species tree estimation,” *Bioinform*, vol. 30, no. 17, pp. i541–i548, 2014.
- [49] P. Vachaspati and T. Warnow, “ASTRID: accurate species trees from internode distances,” *BMC Genomics*, vol. 16, no. 10, pp. 1–13, 2015.
- [50] S. Mirarab and T. Warnow, “ASTRAL-II: coalescent-based species tree estimation with many hundreds of taxa and thousands of genes,” *Bioinformatics*, vol. 31, no. 12, pp. i44–i52, 2015.
- [51] C. Zhang, M. Rabiee, E. Sayyari, and S. Mirarab, “ASTRAL-III: polynomial time species tree reconstruction from partially resolved gene trees,” *BMC bioinformatics*, vol. 19, no. 6, pp. 15–30, 2018.
- [52] J. Yin, C. Zhang, and S. Mirarab, “ASTRAL-MP: scaling astral to very large datasets using randomization and parallelization,” *Bioinformatics*, vol. 35, no. 20, pp. 3961–3969, 2019.
- [53] M. Rabiee, E. Sayyari, and S. Mirarab, “Multi-allele species reconstruction using ASTRAL,” *Molecular Phylogenetics and Evolution*, vol. 130, pp. 286–296, 2019.
- [54] C. Zhang, C. Scornavacca, E. K. Molloy, and S. Mirarab, “ASTRAL-Pro: quartet-based species-tree inference despite paralogy,” *Molecular biology and evolution*, vol. 37, no. 11, pp. 3292–3307, 2020.
- [55] P. Dibaeinia, S. Tabe-Bordbar, and T. Warnow, “Fastral: improving scalability of phylogenomic analysis,” *Bioinformatics*, vol. 37, no. 16, pp. 2317–2324, 2021.

- [56] C. Zhang and S. Mirarab, “ASTRAL-Pro 2: ultrafast species tree reconstruction from multi-copy gene family trees,” *Bioinformatics*, vol. 38, no. 21, pp. 4949–4950, 2022.
- [57] C. Zhang and S. Mirarab, “Weighting by gene tree uncertainty improves accuracy of quartet-based species trees,” *Molecular Biology and Evolution*, vol. 39, no. 12, p. msac215, 2022.
- [58] Z. Yan, M. L. Smith, P. Du, M. W. Hahn, and L. Nakhleh, “Species tree inference methods intended to deal with incomplete lineage sorting are robust to the presence of paralogs,” *Systematic Biology*, vol. 71, no. 2, pp. 367–381, 2022.
- [59] L. Chen, Q. Qiu, Y. Jiang, K. Wang, Z. Lin, Z. Li, F. Bibi, Y. Yang, J. Wang, W. Nie et al., “Large-scale ruminant genome sequencing provides insights into their evolution and distinct traits,” *Science*, vol. 364, no. 6446, p. eaav6202, 2019.
- [60] Q. Zhu, U. Mai, W. Pfeiffer, S. Janssen, F. Asnicar, J. G. Sanders, P. Belda-Ferre, G. A. Al-Ghalith, E. Kopylova, D. McDonald et al., “Phylogenomics of 10,575 genomes reveals evolutionary proximity between domains bacteria and archaea,” *Nature communications*, vol. 10, no. 1, p. 5477, 2019.
- [61] One Thousand Plant Transcriptomes Initiative, “One thousand plant transcriptomes and the phylogenomics of green plants,” *Nature*, vol. 574, no. 7780, pp. 679–685, 2019.
- [62] S. Feng, J. Stiller, Y. Deng, J. Armstrong, Q. Fang, A. H. Reeve, D. Xie, G. Chen, C. Guo, B. C. Faircloth et al., “Dense sampling of bird diversity increases power of comparative genomics,” *Nature*, vol. 587, no. 7833, pp. 252–257, 2020.
- [63] E. S. Allman, J. H. Degnan, and J. A. Rhodes, “Species tree inference by the STAR method and its generalizations,” *Journal of Computational Biology*, vol. 20, no. 1, pp. 50–61, 2013.
- [64] M. W. Hahn, T. De Bie, J. E. Stajich, C. Nguyen, and N. Cristianini, “Estimating the tempo and mode of gene family evolution from comparative genomic data,” *Genome research*, vol. 15, no. 8, pp. 1153–1160, 2005.
- [65] S. L. Kosakovsky Pond and S. D. Frost, “Not so different after all: a comparison of methods for detecting amino acid sites under selection,” *Molecular biology and evolution*, vol. 22, no. 5, pp. 1208–1222, 2005.
- [66] J. Felsenstein, “Phylogenies and the comparative method,” *The American Naturalist*, vol. 125, no. 1, pp. 1–15, 1985.
- [67] B. C. O’Meara, “Evolutionary inferences from phylogenies: a review of methods,” *Annual Review of Ecology, Evolution, and Systematics*, vol. 43, pp. 267–285, 2012.
- [68] E. M. Volz, K. Koelle, and T. Bedford, “Viral phylodynamics,” *PLoS computational biology*, vol. 9, no. 3, p. e1002947, 2013.

- [69] L. Shavit, D. Penny, M. D. Hendy, and B. R. Holland, “The problem of rooting rapid radiations,” *Molecular biology and evolution*, vol. 24, no. 11, pp. 2400–2411, 2007.
- [70] M. J. Phillips, “Branch-length estimation bias misleads molecular dating for a vertebrate mitochondrial phylogeny,” *Gene*, vol. 441, no. 1-2, pp. 132–140, 2009.
- [71] R. S. Schwartz and R. L. Mueller, “Branch length estimation and divergence dating: estimates of error in bayesian and maximum likelihood frameworks,” *BMC Evolutionary Biology*, vol. 10, no. 1, pp. 1–21, 2010.
- [72] Y. Zheng, R. Peng, M. Kuro-o, and X. Zeng, “Exploring patterns and extent of bias in estimating divergence time from mitochondrial dna sequence data in a particular lineage: a case study of salamanders (order caudata),” *Molecular Biology and Evolution*, vol. 28, no. 9, pp. 2521–2535, 2011.
- [73] M. van Tuinen and C. R. Torres, “Potential for bias and low precision in molecular divergence time estimation of the canopy of life: an example from aquatic bird families,” *Frontiers in genetics*, vol. 6, p. 203, 2015.
- [74] Y. Tabatabaei, C. Zhang, T. Warnow, and S. Mirarab, “Phylogenomic branch length estimation using quartets,” *Bioinformatics*, vol. 39, no. Supplement_1, pp. i185–i193, June 2023.
- [75] S. Arasti, P. Tabaghi, Y. Tabatabaei, and S. Mirarab, “Branch length transforms using optimal tree metric matching,” *bioRxiv*, 2024, <https://doi.org/10.1101/2023.11.13.566962>.
- [76] E. R. Moody, T. A. Mahendarajah, N. Dombrowski, J. W. Clark, C. Petitjean, P. Ofre, G. J. Szöllősi, A. Spang, and T. A. Williams, “An estimate of the deepest branches of the tree of life from ancient vertically evolving genes,” *Elife*, vol. 11, p. e66695, 2022.
- [77] S. Mirarab, “Species tree estimation using ASTRAL: practical considerations,” *arXiv preprint arXiv:1904.03826*, 2019.
- [78] U. Mai, E. Sayyari, and S. Mirarab, “Minimum variance rooting of phylogenetic trees and implications for species tree reconstruction,” *PLoS One*, vol. 12, no. 8, p. e0182238, 2017.
- [79] F. D. K. Tria, G. Landan, and T. Dagan, “Phylogenetic rooting using minimal ancestor deviation,” *Nature ecology & evolution*, vol. 1, no. 7, p. 0193, 2017.
- [80] U. Mai, E. Charvel, and S. Mirarab, “Expectation-maximization enables phylogenetic dating under a categorical rate model,” *Systematic Biology*, p. syae034, 2024.
- [81] E. S. Allman, J. H. Degnan, and J. A. Rhodes, “Identifying the rooted species tree from the distribution of unrooted gene trees under the coalescent,” *Journal of mathematical biology*, vol. 62, pp. 833–862, 2011.

- [82] T. Warnow, *Computational phylogenetics: an introduction to designing methods for phylogeny estimation.* Cambridge University Press, 2017.
- [83] D. F. Robinson and L. R. Foulds, “Comparison of phylogenetic trees,” *Mathematical biosciences*, vol. 53, no. 1-2, pp. 131–147, 1981.
- [84] M. S. Springer and J. Gatesy, “The gene tree delusion,” *Molecular Phylogenetics and Evolution*, vol. 94, pp. 1–33, 2016.
- [85] R. Nielsen, A. H. Vaughn, and Y. Deng, “Inference and applications of ancestral recombination graphs,” *Nature Reviews Genetics*, vol. 26, no. 1, pp. 47–58, 2025.
- [86] E. Rachman, V. Bafna, and S. Mirarab, “CONSULT: accurate contamination removal using locality-sensitive hashing,” *NAR Genomics and Bioinformatics*, vol. 3, no. 3, p. lqab071, 2021.
- [87] A. M. Altenhoff and C. Dessimoz, “Inferring orthology and paralogy,” *Evolutionary Genomics: Statistical and Computational Methods, Volume 1*, pp. 259–279, 2012.
- [88] E. K. Molloy and T. Warnow, “FastMulRFS: fast and accurate species tree estimation under generic gene duplication and loss models,” *Bioinformatics*, vol. 36, no. Supplement_1, pp. i57–i65, 2020.
- [89] B. Legried, E. K. Molloy, T. Warnow, and S. Roch, “Polynomial-time statistical estimation of species trees under gene duplication and loss,” *Journal of Computational Biology*, vol. 28, no. 5, pp. 452–468, 2021.
- [90] J. Willson, M. S. Roddur, B. Liu, P. Zaharias, and T. Warnow, “DISCO: species tree inference using multicity gene family tree decomposition,” *Systematic biology*, vol. 71, no. 3, pp. 610–629, 2022.
- [91] R. C. Edgar and S. Batzoglou, “Multiple sequence alignment,” *Current opinion in structural biology*, vol. 16, no. 3, pp. 368–373, 2006.
- [92] B. Q. Minh, H. A. Schmidt, O. Chernomor, D. Schrempf, M. D. Woodhams, A. Von Haeseler, and R. Lanfear, “IQ-TREE 2: new models and efficient methods for phylogenetic inference in the genomic era,” *Molecular biology and evolution*, vol. 37, no. 5, pp. 1530–1534, 2020.
- [93] L. Liu, L. Yu, D. K. Pearl, and S. V. Edwards, “Estimating species phylogenies using coalescence times among sequences,” *Systematic biology*, vol. 58, no. 5, pp. 468–477, 2009.
- [94] S. Bjornson, H. Verbruggen, N. S. Upham, and J. L. Steenwyk, “Reticulate evolution: Detection and utility in the phylogenomics era,” *Molecular Phylogenetics and Evolution*, p. 108197, 2024.
- [95] P. Pamilo and M. Nei, “Relationships between gene trees and species trees.” *Molecular biology and evolution*, vol. 5, no. 5, pp. 568–583, 1988.

- [96] S. Tavaré, “Some probabilistic and statistical problems in the analysis of DNA sequences,” *Lectures on Mathematics in the Life Sciences*, vol. 17, no. 2, pp. 57–86, 1986.
- [97] J. F. C. Kingman, “The coalescent,” *Stochastic processes and their applications*, vol. 13, no. 3, pp. 235–248, 1982.
- [98] R. R. Hudson, “Testing the constant-rate neutral allele model with protein sequence data,” *Evolution*, pp. 203–217, 1983.
- [99] J. B. Pease, D. C. Haak, M. W. Hahn, and L. C. Moyle, “Phylogenomics reveals three sources of adaptive variation during a rapid radiation,” *PLoS biology*, vol. 14, no. 2, p. e1002379, 2016.
- [100] J. E. McCormack, B. C. Faircloth, N. G. Crawford, P. A. Gowaty, R. T. Brumfield, and T. C. Glenn, “Ultraconserved elements are novel phylogenomic markers that resolve placental mammal phylogeny when combined with species-tree analysis,” *Genome research*, vol. 22, no. 4, pp. 746–754, 2012.
- [101] M. P. Blom, J. G. Bragg, S. Potter, and C. Moritz, “Accounting for uncertainty in gene tree estimation: summary-coalescent species tree inference in a challenging radiation of australian lizards,” *Systematic Biology*, vol. 66, no. 3, pp. 352–366, 2017.
- [102] S. Bossert, E. A. Murray, A. Pauly, K. Chernyshov, S. G. Brady, and B. N. Danforth, “Gene tree estimation error with ultraconserved elements: an empirical study on pseudapis bees,” *Systematic Biology*, vol. 70, no. 4, pp. 803–821, 2021.
- [103] J. H. Degnan and N. A. Rosenberg, “Discordance of species trees with their most likely gene trees,” *PLoS genetics*, vol. 2, no. 5, p. e68, 2006.
- [104] J. H. Degnan, “Anomalous unrooted gene trees,” *Systematic biology*, vol. 62, no. 4, pp. 574–590, 2013.
- [105] J. H. Degnan and N. A. Rosenberg, “Gene tree discordance, phylogenetic inference and the multispecies coalescent,” *Trends Ecol. Evol.*, vol. 24, no. 6, pp. 332–340, 2009.
- [106] J. H. Wakeley, *Coalescent Theory: An Introduction*. Greenwood Village, CO: Roberts Company Publishers, 2009.
- [107] N. A. Rosenberg, “The probability of topological concordance of gene trees and species trees,” *Theoretical population biology*, vol. 61, no. 2, pp. 225–247, 2002.
- [108] S. Tavaré, “Line-of-descent and genealogical processes, and their applications in population genetics models,” *Theoretical population biology*, vol. 26, no. 2, pp. 119–164, 1984.

- [109] N. J. Wickett, S. Mirarab, N. Nguyen, T. Warnow, E. Carpenter, N. Matasci, S. Ayyampalayam, M. S. Barker, J. G. Burleigh, M. A. Gitzendanner et al., “Phylogenomic analysis of the origin and early diversification of land plants,” *Proceedings of the National Academy of Sciences*, vol. 111, no. 45, pp. E4859–E4868, 2014.
- [110] S. M. Glasauer and S. C. Neuhauss, “Whole-genome duplication in teleost fishes and its evolutionary consequences,” *Molecular genetics and genomics*, vol. 289, pp. 1045–1060, 2014.
- [111] L. Arvestad, J. Lagergren, and B. Sennblad, “The gene evolution model and computing its associated probabilities,” *Journal of the ACM (JACM)*, vol. 56, no. 2, pp. 1–44, 2009.
- [112] M. D. Rasmussen and M. Kellis, “Unified modeling of gene duplication, loss, and coalescence using a locus tree,” *Genome research*, vol. 22, no. 4, pp. 755–765, 2012.
- [113] H. Ochman, J. G. Lawrence, and E. A. Groisman, “Lateral gene transfer and the nature of bacterial innovation,” *nature*, vol. 405, no. 6784, pp. 299–304, 2000.
- [114] E. V. Koonin, K. S. Makarova, and L. Aravind, “Horizontal gene transfer in prokaryotes: quantification and classification,” *Annual Reviews in Microbiology*, vol. 55, no. 1, pp. 709–742, 2001.
- [115] B. J. Arnold, I.-T. Huang, and W. P. Hanage, “Horizontal gene transfer and adaptive evolution in bacteria,” *Nature Reviews Microbiology*, vol. 20, no. 4, pp. 206–218, 2022.
- [116] A. Tofigh, “Using trees to capture reticulate evolution: lateral gene transfers and cancer progression,” Ph.D. dissertation, KTH, PhD Dissertation, 2009.
- [117] G. J. Szöllősi, B. Boussau, S. S. Abby, E. Tannier, and V. Daubin, “Phylogenetic modeling of lateral gene transfer reconstructs the pattern and relative timing of speciations,” *Proceedings of the national academy of sciences*, vol. 109, no. 43, pp. 17513–17518, 2012.
- [118] G. J. Szöllősi, E. Tannier, V. Daubin, and B. Boussau, “The inference of gene trees with species trees,” *Systematic biology*, vol. 64, no. 1, pp. e42–e62, 2015.
- [119] L. H. Rieseberg, S.-C. Kim, R. A. Randell, K. D. Whitney, B. L. Gross, C. Lexer, and K. Clay, “Hybridization and the colonization of novel habitats by annual sunflowers,” *Genetica*, vol. 129, pp. 149–165, 2007.
- [120] D. E. Neafsey, B. M. Barker, T. J. Sharpton, J. E. Stajich, D. J. Park, E. Whiston, C.-Y. Hung, C. McMahan, J. White, S. Sykes et al., “Population genomic sequencing of coccidioides fungi reveals recent hybridization and transposon control,” *Genome research*, vol. 20, no. 7, pp. 938–946, 2010.
- [121] E. H. Stukenbrock, “The role of hybridization in the evolution and emergence of new fungal plant pathogens,” *Phytopathology*, vol. 106, no. 2, pp. 104–112, 2016.

- [122] C. Meng and L. S. Kubatko, “Detecting hybrid speciation in the presence of incomplete lineage sorting using gene tree incongruence: a model,” *Theoretical population biology*, vol. 75, no. 1, pp. 35–45, 2009.
- [123] Y. Yu, J. H. Degnan, and L. Nakhleh, “The probability of a gene tree topology within a phylogenetic network with applications to hybridization detection,” *PLoS genetics*, vol. 8, no. 4, p. e1002660, 2012.
- [124] J. Zhu, Y. Yu, and L. Nakhleh, “In the light of deep coalescence: revisiting trees within networks,” *BMC bioinformatics*, vol. 17, pp. 271–282, 2016.
- [125] C. Solís-Lemus, M. Yang, and C. Ané, “Inconsistency of species tree methods under gene flow,” *Systematic biology*, vol. 65, no. 5, pp. 843–851, 2016.
- [126] J. H. Degnan, “Modeling hybridization under the network multispecies coalescent,” *Systematic biology*, vol. 67, no. 5, pp. 786–799, 2018.
- [127] H. Baños, “Identifying species network features from gene tree quartets under the coalescent model,” *Bulletin of mathematical biology*, vol. 81, no. 2, pp. 494–534, 2019.
- [128] C. Solís-Lemus and C. Ané, “Inferring phylogenetic networks with maximum pseudo-likelihood under incomplete lineage sorting,” *PLoS genetics*, vol. 12, no. 3, p. e1005896, 2016.
- [129] E. S. Allman, H. Baños, M. Garrote-Lopez, and J. A. Rhodes, “Identifiability of level-1 species networks from gene tree quartets,” *Bulletin of Mathematical Biology*, vol. 86, no. 9, p. 110, 2024.
- [130] T. Jukes and C. Cantor, “Evolution of protein molecules,” *Mammalian Protein Metabolism*, p. 21–132, 1969.
- [131] M. Kimura, “A simple method for estimating evolutionary rate of base substitutions through comparative studies of nucleotide sequences,” *Journal of Molecular Evolution*, vol. 16, no. 2, pp. 111–120, 1980.
- [132] M. Hasegawa, H. Kishino, and T.-a. Yano, “Dating of the human-ape splitting by a molecular clock of mitochondrial dna,” *Journal of molecular evolution*, vol. 22, pp. 160–174, 1985.
- [133] S. Tavaré, “Some Probabilistic and Statistical Problems in the Analysis of DNA Sequences,” *Lectures on Mathematics in the Life Sciences*, vol. 17, pp. 57–86, 1986.
- [134] M. Steel, “Recovering a tree from the leaf colourations it generates under a Markov model,” *Applied Mathematics Letters*, vol. 7, no. 2, pp. 19–23, Mar. 1994.
- [135] E. S. Allman and J. A. Rhodes, “Identifying evolutionary trees and substitution parameters for the general markov model with invariable sites,” *Mathematical Biosciences*, vol. 211, no. 1, pp. 18–33, 2008.

- [136] Z. Yang, “Maximum likelihood phylogenetic estimation from dna sequences with variable rates over sites: approximate methods,” *Journal of Molecular evolution*, vol. 39, pp. 306–314, 1994.
- [137] Z. Yang, “Among-site rate variation and its impact on phylogenetic analyses,” *Trends in ecology & evolution*, vol. 11, no. 9, pp. 367–372, 1996.
- [138] P. Lopez, D. Casane, and H. Philippe, “Heterotachy, an Important Process of Protein Evolution,” *Molecular Biology and Evolution*, vol. 19, no. 1, pp. 1–7, Jan. 2002.
- [139] C. Tuffley and M. Steel, “Links between maximum likelihood and maximum parsimony under a simple model of site substitution.” *Bulletin of mathematical biology*, vol. 59, no. 3, pp. 581–607, 1997.
- [140] S. Kumar, “Molecular clocks: four decades of evolution,” *Nature Reviews Genetics*, vol. 6, no. 8, pp. 654–662, Aug. 2005.
- [141] S. Aris-Brosou and Z. Yang, “Effects of models of rate evolution on estimation of divergence dates with special reference to the metazoan 18s ribosomal rna phylogeny,” *Systematic Biology*, vol. 51, no. 5, pp. 703–714, 2002.
- [142] J. Felsenstein, “Cases in which parsimony or compatibility methods will be positively misleading,” *Systematic zoology*, vol. 27, no. 4, pp. 401–410, 1978.
- [143] T. A. Heath, S. M. Hedtke, and D. M. Hillis, “Taxon sampling and the accuracy of phylogenetic analyses,” *Journal of Systematics and Evolution*, vol. 46, pp. 239–257, 2008.
- [144] M. N. Price, P. S. Dehal, and A. P. Arkin, “FastTree-2 – Approximately Maximum-Likelihood Trees for Large Alignments,” *PLoS ONE*, vol. 5, no. 3, p. e9490, Mar. 2010.
- [145] F. Ronquist, M. Teslenko, P. van der Mark, D. L. Ayres, A. Darling, S. Höhna, B. R. Larget, L. Liu, M. A. Suchard, and J. P. Huelsenbeck, “MrBayes 3.2: efficient Bayesian phylogenetic inference and model choice across a large model space,” *Systematic biology*, vol. 61, no. 3, pp. 539–542, 2012.
- [146] N. Saitou and M. Nei, “The neighbor-joining method: a new method for reconstructing phylogenetic trees.” *Molecular Biology and Evolution*, vol. 4, no. 4, pp. 406–425, 07 1987.
- [147] S. Roch, “A short proof that phylogenetic tree reconstruction by maximum likelihood is hard,” *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*, vol. 3, no. 1, pp. 92–94, 2006.
- [148] J. Felsenstein, “Inferring phylogenies,” in *Inferring phylogenies*. Sunderland, MA: Sinauer Associates, 2004, pp. 664–664.

- [149] A. Stamatakis, “RAxML-VI-HPC: maximum likelihood-based phylogenetic analyses with thousands of taxa and mixed models,” *Bioinformatics*, vol. 22, no. 21, pp. 2688–2690, 08 2006. [Online]. Available: <https://doi.org/10.1093/bioinformatics/btl446>
- [150] S. Guindon, J.-F. Dufayard, V. Lefort, M. Anisimova, W. Hordijk, and O. Gascuel, “New Algorithms and Methods to Estimate Maximum-Likelihood Phylogenies: Assessing the Performance of PhyML 3.0,” *Systematic Biology*, vol. 59, no. 3, pp. 307–321, May 2010.
- [151] P. Buneman, “A note on the metric properties of trees,” *Journal of Combinatorial Theory, Series B*, vol. 17, no. 1, pp. 48–50, 1974.
- [152] K. Atteson, “The Performance of Neighbor-Joining Methods of Phylogenetic Reconstruction,” *Algorithmica*, vol. 25, no. 2-3, pp. 251–278, 1999.
- [153] P. L. Erdős, M. A. Steel, L. Székely, and T. J. Warnow, “A few logs suffice to build (almost) all trees: Part ii,” *Theoretical Computer Science*, vol. 221, no. 1-2, pp. 77–118, 1999.
- [154] P. Buneman, “The recovery of trees from measures of dissimilarity,” *Mathematics in the archaeological and historical sciences*, 1971.
- [155] J. P. Huelsenbeck, F. Ronquist, R. Nielsen, and J. P. Bollback, “Bayesian inference of phylogeny and its impact on evolutionary biology,” *science*, vol. 294, no. 5550, pp. 2310–2314, 2001.
- [156] M. Holder and P. O. Lewis, “Phylogeny estimation: traditional and bayesian approaches,” *Nature reviews genetics*, vol. 4, no. 4, pp. 275–284, 2003.
- [157] J. Truszkowski, A. Perrigo, D. Broman, F. Ronquist, and A. Antonelli, “Online tree expansion could help solve the problem of scalability in bayesian phylogenetics,” *Systematic Biology*, vol. 72, no. 5, pp. 1199–1206, 2023.
- [158] S. Holmes, “Statistics for phylogenetic trees,” *Theoretical Population Biology*, vol. 63, no. 1, pp. 17–32, Feb. 2003.
- [159] D. L. Swofford, G. J. Olsen, P. J. Waddell, D. M. Hillis, C. Moritz, B. Mable et al., “Phylogenetic inference.” *Molecular systematics*, 2nd ed., pp. 407–514, 1996.
- [160] L. Kubatko and L. L. Knowles, Eds., *Species Tree Inference: A Guide to Methods and Applications*. Princeton University Press, 2023.
- [161] L. Liu and D. K. Pearl, “Species trees from gene trees: reconstructing bayesian posterior distributions of a species phylogeny using estimated gene tree distributions,” *Systematic biology*, vol. 56, no. 3, pp. 504–514, 2007.
- [162] B. R. Larget, S. K. Kotha, C. N. Dewey, and C. Ané, “BUCKy: gene tree/species tree reconciliation with bayesian concordance analysis,” *Bioinformatics*, vol. 26, no. 22, pp. 2910–2911, 2010.

- [163] G. Dasarathy, R. Nowak, and S. Roch, “Data requirement for phylogenetic inference from multiple loci: a new distance method,” *IEEE/ACM transactions on computational biology and bioinformatics*, vol. 12, no. 2, pp. 422–432, 2014.
- [164] L. S. Kubatko, B. C. Carstens, and L. L. Knowles, “STEM: species tree estimation using maximum likelihood for gene trees under coalescence,” *Bioinformatics*, vol. 25, no. 7, pp. 971–973, 2009.
- [165] E. Mossel and S. Roch, “Incomplete lineage sorting: consistent phylogeny estimation from multiple loci,” *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 7, no. 1, pp. 166–171, 2008.
- [166] L. Liu, L. Yu, and S. Edwards, “A maximum pseudo-likelihood approach for estimating species trees under the coalescent model,” *BMC. Evol. Biol.*, vol. 10, p. 302, 2010.
- [167] E. Mossel and S. Roch, “Incomplete lineage sorting: consistent phylogeny estimation from multiple loci.” *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*, vol. 7, no. 1, pp. 166–171, Jan. 2010.
- [168] M. P. Simmons, M. S. Springer, and J. Gatesy, “Gene-tree misrooting drives conflicts in phylogenomic coalescent analyses of palaeognath birds,” *Molecular phylogenetics and evolution*, vol. 167, p. 107344, 2022.
- [169] J. H. Degnan, “Anomalous unrooted gene trees,” *Systematic biology*, vol. 62, no. 4, pp. 574–590, 2013.
- [170] N. A. Rosenberg, “Discordance of species trees with their most likely gene trees: a unifying principle,” *Molecular Biology and Evolution*, vol. 30, no. 12, pp. 2709–2713, 2013.
- [171] M. Mahbub, Z. Wahab, R. Reaz, M. S. Rahman, and M. S. Bayzid, “wQFM: highly accurate genome-scale species tree estimation from weighted quartets,” *Bioinformatics*, vol. 37, no. 21, pp. 3734–3743, 2021.
- [172] C. Zhang, R. Nielsen, and S. Mirarab, “Aster: A package for large-scale phylogenomic reconstructions,” *Molecular Biology and Evolution*, vol. 42, no. 8, p. msaf172, 2025.
- [173] T. Jiang, P. Kearney, and M. Li, “A Polynomial Time Approximation Scheme for Inferring Evolutionary Trees from Quartet Topologies and Its Application,” *SIAM Journal on Computing*, vol. 30, no. 6, pp. 1942–1961, 2001.
- [174] M. Lafond and C. Scornavacca, “On the Weighted Quartet Consensus problem,” *Theoretical Computer Science*, vol. 769, pp. 1–17, May 2019.
- [175] S. Arasti and S. Mirarab, “Median quartet tree search algorithms using optimal subtree prune and regraft,” *Algorithms for Molecular Biology*, vol. 19, no. 1, p. 12, 2024.

- [176] K. Strimmer and a. von Haeseler, “Quartet puzzling - a quartet maximum-likelihood method for reconstructing tree topologies,” *Molecular biology and evolution*, vol. 13, pp. 964–969, 1996.
- [177] S. Snir and S. Rao, “Quartets MaxCut: A divide and conquer quartets algorithm,” *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*, vol. 7, no. 4, pp. 704–718, 2010.
- [178] D. Bryant and M. Steel, “Constructing Optimal Trees from Quartets,” *Journal of Algorithms*, vol. 38, no. 1, pp. 237–259, Jan. 2001.
- [179] S. Mirarab, R. Reaz, M. S. Bayzid, T. Zimmermann, M. S. Swenson, and T. Warnow, “ASTRAL: genome-scale coalescent-based species tree estimation,” *Bioinformatics*, vol. 30, no. 17, pp. i541–i548, Sep. 2014.
- [180] L. Liu and L. Yu, “Estimating Species Trees from Unrooted Gene Trees,” *Systematic Biology*, vol. 60, no. 5, Oct. 2011.
- [181] P. Vachaspati and T. Warnow, “ASTRID: Accurate Species TRees from Internode Distances,” *BMC Genomics*, vol. 16, no. Suppl 10, p. S3, 2015.
- [182] O. Gascuel, “BIONJ: an improved version of the NJ algorithm based on a simple model of sequence data.” *Molecular Biology and Evolution*, vol. 14, no. 7, pp. 685–695, 07 1997. [Online]. Available: <https://doi.org/10.1093/oxfordjournals.molbev.a025808>
- [183] B. Liu and T. Warnow, “Weighted astrid: fast and accurate species trees from weighted internode distances,” *Algorithms for Molecular Biology*, vol. 18, no. 1, p. 6, 2023.
- [184] D. Bryant, R. Bouckaert, J. Felsenstein, N. A. Rosenberg, and A. Roychoudhury, “Inferring species trees directly from biallelic genetic markers: Bypassing gene trees in a full coalescent analysis,” *Molecular Biology and Evolution*, vol. 29, no. 8, pp. 1917–1932, 2012.
- [185] J. Chifman and L. S. Kubatko, “Quartet Inference from SNP Data Under the Coalescent Model,” *Bioinformatics*, vol. 30, no. 23, pp. 3317–3324, Aug. 2014.
- [186] G. Dasarathy, R. Nowak, and S. Roch, “Data requirement for phylogenetic inference from multiple loci: a new distance method,” *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*, vol. 12, no. 2, pp. 422–432, 2015.
- [187] D. L. Swofford, “PAUP*. Phylogenetic Analysis Using Parsimony (*and other methods). Version 4.” 2003.
- [188] R. Reaz, M. S. Bayzid, and M. S. Rahman, “Accurate phylogenetic tree reconstruction from quartets: A heuristic approach,” *PloS one*, vol. 9, no. 8, p. e104008, 2014.
- [189] S. Mirarab, M. S. Bayzid, B. Boussau et al., “Statistical binning enables an accurate coalescent-based estimation of the avian tree,” *Science*, vol. 346, no. 6215, p. 1250463, 2014.

- [190] S. Patel, “Error in Phylogenetic Estimation for Bushes in the Tree of Life,” *Journal of Phylogenetics & Evolutionary Biology*, vol. 01, no. 02, p. 110, 2013.
- [191] J. S. Patané, J. Martins, and J. C. Setubal, “Phylogenomics,” *Comparative Genomics: Methods and Protocols*, pp. 103–187, 2018.
- [192] L. Liu, “BEST: Bayesian estimation of species trees under the coalescent model,” *Bioinformatics*, vol. 24, no. 21, pp. 2542–2543, Nov. 2008.
- [193] J. Heled and A. J. Drummond, “Bayesian inference of species trees from multilocus data,” *Molecular Biology and Evolution*, vol. 27, no. 3, pp. 570–580, Mar. 2010.
- [194] J. Douglas, C. L. Jiménez-Silva, and R. Bouckaert, “Starbeast3: adaptive parallelized bayesian inference under the multispecies coalescent,” *Systematic Biology*, vol. 71, no. 4, pp. 901–916, 2022.
- [195] J. E. McCormack, H. Huang, and L. L. Knowles, “Maximum likelihood estimates of species trees: how accuracy of phylogenetic inference depends upon the divergence history and sampling design,” *Systematic biology*, vol. 58, no. 5, pp. 501–508, 2009.
- [196] S. D. Leavitt, F. Grawe, T. Widholm, L. Muggia, B. Wray, and H. T. Lumbsch, “Resolving evolutionary relationships in lichen-forming fungi using diverse phylogenomic datasets and analytical approaches,” *Scientific reports*, vol. 6, no. 1, p. 22262, 2016.
- [197] A. Wehe, M. S. Bansal, J. G. Burleigh, and O. Eulenstein, “DupTree: a program for large-scale phylogenetic analyses using gene tree parsimony.” *Bioinformatics*, vol. 24, no. 13, pp. 1540–1541, 2008.
- [198] D. Durand, B. V. Halldórsson, and B. Vernot, “A hybrid micro-macroevolutionary approach to gene tree reconstruction,” *Journal of Computational Biology*, vol. 13, no. 2, pp. 320–335, 2006.
- [199] B. Morel, P. Schade, S. Lutteropp, T. A. Williams, G. J. Szöllősi, and A. Stamatakis, “Speciesrax: a tool for maximum likelihood species tree inference from gene family trees under duplication, transfer, and loss,” *Molecular biology and evolution*, vol. 39, no. 2, p. msab365, 2022.
- [200] A. Markin and O. Eulenstein, “Quartet-based inference is statistically consistent under the unified duplication-loss-coalescence model,” *Bioinformatics*, vol. 37, no. 22, pp. 4064–4074, 2021.
- [201] M. Hill, B. Legried, and S. Roch, “Species tree estimation under joint modeling of coalescence and duplication: sample complexity of quartet methods,” *The Annals of Applied Probability*, vol. 32, no. 6, pp. 4681–4705, 2022.
- [202] S. Roch and S. Snir, “Recovering the treelike trend of evolution despite extensive lateral genetic transfer: a probabilistic analysis,” *Journal of Computational Biology*, vol. 20, no. 2, pp. 93–112, 2013.

- [203] C. Daskalakis and S. Roch, “Species trees from gene trees despite a high rate of lateral genetic transfer: A tight bound,” in *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms*. SIAM, 2016, pp. 1621–1630.
- [204] R. Davidson, P. Vachaspati, S. Mirarab, and T. Warnow, “Phylogenomic species tree estimation in the presence of incomplete lineage sorting and horizontal gene transfer,” *BMC genomics*, vol. 16, no. 10, pp. 1–12, 2015.
- [205] D. Wen, Y. Yu, J. Zhu, and L. Nakhleh, “Inferring phylogenetic networks using phylogenonet,” *Systematic biology*, vol. 67, no. 4, pp. 735–740, 2018.
- [206] D. Wen and L. Nakhleh, “Coestimating reticulate phylogenies and gene trees from multilocus sequence data,” *Systematic biology*, vol. 67, no. 3, pp. 439–457, 2018.
- [207] J. Zhu, D. Wen, Y. Yu, H. M. Meudt, and L. Nakhleh, “Bayesian inference of phylogenetic networks from bi-allelic genetic markers,” *PLoS computational biology*, vol. 14, no. 1, p. e1005932, 2018.
- [208] J. Zhu and L. Nakhleh, “Inference of species phylogenies from bi-allelic markers using pseudo-likelihood,” *Bioinformatics*, vol. 34, no. 13, pp. i376–i385, 2018.
- [209] J. Zhu, X. Liu, H. A. Ogilvie, and L. K. Nakhleh, “A divide-and-conquer method for scalable phylogenetic network inference from multilocus data,” *Bioinformatics*, vol. 35, no. 14, pp. i370–i378, 2019.
- [210] C. Solís-Lemus, P. Bastide, and C. Ané, “Phylonetworks: a package for phylogenetic networks,” *Molecular biology and evolution*, vol. 34, no. 12, pp. 3292–3298, 2017.
- [211] E. S. Allman, H. Baños, and J. A. Rhodes, “Nanuq: a method for inferring species networks from gene trees under the coalescent model,” *Algorithms for Molecular Biology*, vol. 14, no. 1, p. 24, 2019.
- [212] C. Than, D. Ruths, and L. Nakhleh, “Phylogenonet: a software package for analyzing and reconstructing reticulate evolutionary relationships,” *BMC bioinformatics*, vol. 9, no. 1, p. 322, 2008.
- [213] T. Warnow, Y. Tabatabaei, and S. N. Evans, “Advances in estimating level-1 phylogenetic networks from unrooted snps,” *Journal of Computational Biology*, vol. 32, no. 1, pp. 3–27, 2025.
- [214] D. M. Emms and S. Kelly, “STRIDE: species tree root inference from gene duplication events,” *Molecular biology and evolution*, vol. 34, no. 12, pp. 3267–3278, 2017.
- [215] S. G. Brady, J. R. Litman, and B. N. Danforth, “Rooting phylogenies using gene duplications: an empirical example from the bees (apoidea),” *Molecular Phylogenetics and Evolution*, vol. 60, no. 3, pp. 295–304, 2011.

- [216] A. A. Davín, E. Tannier, T. A. Williams, B. Boussau, V. Daubin, and G. J. Szöllősi, “Gene transfers can date the tree of life,” *Nature ecology & evolution*, vol. 2, no. 5, pp. 904–909, 2018.
- [217] S. Edwards and P. Beerli, “Perspective: gene divergence, population divergence, and the variance in coalescence time in phylogeographic studies,” *Evolution*, vol. 54, no. 6, pp. 1839–1854, 2000.
- [218] Z. Yang, *Molecular evolution: a statistical approach*. Oxford University Press, 2014.
- [219] F. Rutschmann, “Molecular dating of phylogenetic trees: a brief review of current methods that estimate divergence times,” *Diversity and Distributions*, vol. 12, no. 1, pp. 35–48, 2006.
- [220] P. N. Hess and C. A. De Moraes Russo, “An empirical test of the midpoint rooting method,” *Biological Journal of the Linnean society*, vol. 92, no. 4, pp. 669–674, 2007.
- [221] F. Tria, G. Landan, and T. Dagan, “Phylogenetic rooting using minimal ancestor deviation,” *Nat. Ecol. Evol.*, vol. 1, 2017.
- [222] D. M. Emms and S. Kelly, “STRIDE: species tree root inference from gene duplication events,” *Mol. Biol. and Evol.*, vol. 34, pp. 3267 – 3278, 2017.
- [223] T. Flouri, X. Jiao, B. Rannala, and Z. Yang, “Species Tree Inference with BPP Using Genomic Sequences and the Multispecies Coalescent,” *Molecular Biology and Evolution*, vol. 35, no. 10, pp. 2585–2593, Oct. 2018.
- [224] A. J. Drummond and A. Rambaut, “BEAST: Bayesian evolutionary analysis by sampling trees.” *BMC evolutionary biology*, vol. 7, p. 214, 2007.
- [225] Z. Yang and B. Rannala, “Bayesian estimation of species divergence times under a molecular clock using multiple fossil calibrations with soft bounds,” *Molecular biology and evolution*, vol. 23, no. 1, pp. 212–226, 2006.
- [226] B. Rannala and Z. Yang, “Inferring Speciation Times under an Episodic Molecular Clock,” *Systematic Biology*, vol. 56, no. 3, pp. 453–466, 06 2007. [Online]. Available: <https://doi.org/10.1080/10635150701420643>
- [227] T.-H. To, M. Jung, S. Lycett, and O. Gascuel, “Fast Dating Using Least-Squares Criteria and Algorithms,” *Systematic Biology*, vol. 65, no. 1, pp. 82–97, Jan. 2016.
- [228] S. A. Smith and B. C. O’Meara, “treePL: divergence time estimation using penalized likelihood for large phylogenies,” *Bioinformatics*, vol. 28, no. 20, pp. 2689–2690, 2012.
- [229] H. Morlon, “Phylogenetic approaches for studying diversification,” *Ecology letters*, vol. 17, no. 4, pp. 508–525, 2014.

- [230] Y. Tabatabaei, K. Sarker, and T. Warnow, “Quintet Rooting: rooting species trees under the multi-species coalescent model,” *Bioinformatics*, vol. 38, no. Supplement_1, pp. i109–i117, 2022.
- [231] A. Pascual-García, M. Arenas, and U. Bastolla, “The molecular clock in the evolution of protein structures,” *Systematic Biology*, vol. 68, no. 6, pp. 987–1002, 2019.
- [232] T. Lepage, D. Bryant, H. Philippe, and N. Lartillot, “A general comparison of relaxed molecular clock models,” *Molecular biology and evolution*, vol. 24, no. 12, pp. 2669–2680, 2007.
- [233] J. O. Wertheim, M. J. Sanderson, M. Worobey, and A. Bjork, “Relaxed molecular clocks, the bias-variance trade-off, and the quality of phylogenetic inference,” *Systematic biology*, vol. 59, no. 1, pp. 1–8, 2010.
- [234] E. W. Wilberg, “What’s in an outgroup? the impact of outgroup choice on the phylogenetic position of thalattosuchia (crocodylomorpha) and the origin of crocodyliformes,” *Systematic Biology*, vol. 64, no. 4, pp. 621–637, 2015.
- [235] W. P. Maddison, M. J. Donoghue, and D. R. Maddison, “Outgroup analysis and parsimony,” *Systematic biology*, vol. 33, no. 1, pp. 83–103, 1984.
- [236] T. Kinene, J. Wainaina, S. Maina, and L. Boykin, “Rooting trees, methods for,” *Encyclopedia of Evolutionary Biology*, p. 489–493, 2016.
- [237] P. L. Erdős, M. A. Steel, L. A. Székely, and T. J. Warnow, “A few logs suffice to build (almost) all trees (i),” *Random Structures & Algorithms*, vol. 14, no. 2, pp. 153–184, 1999.
- [238] B. Holland, D. Penny, and M. Hendy, “Outgroup misplacement and phylogenetic inaccuracy under a molecular clock—a simulation study,” *Systematic biology*, vol. 52, no. 2, pp. 229–238, 2003.
- [239] W. Wheeler, “Nucleic sequence phylogeny and random outgroups,” *Cladistics*, vol. 6, pp. 363 – 367, 1990.
- [240] R. Tarrio, F. Rodriguez-Trelles, and F. Ayala, “Tree rooting with outgroups when they differ in their nucleotide composition from the ingroup: the *Drosophila saltans* and *willistoni* groups, a case study,” *Molecular phylogenetics and evolution*, vol. 16, no. 3, pp. 344–349, 2000.
- [241] U. Mai, E. Sayyari, and S. Mirarab, “Minimum variance rooting of phylogenetic trees and implications for species tree reconstruction,” *PLoS ONE*, vol. 12, 2017.
- [242] J. P. Huelsenbeck, J. P. Bollback, and A. M. Levine, “Inferring the root of a phylogenetic tree,” *Systematic biology*, vol. 51, no. 1, pp. 32–43, 2002.
- [243] B. Bettsworth and A. Stamatakis, “Root Digger: a root placement program for phylogenetic trees,” *BMC Bioinformatics*, vol. 22, no. 1, p. 225, 2021.

- [244] Y. Tian and L. Kubatko, “Rooting phylogenetic trees under the coalescent model using site pattern probabilities,” *BMC evolutionary biology*, vol. 17, pp. 1–11, 2017.
- [245] N. A. Rosenberg, “Counting coalescent histories,” *Journal of Computational Biology*, vol. 14, no. 3, pp. 360–377, 2007.
- [246] S. Mirarab, M. S. Bayzid, B. Boussau, and T. Warnow, “Statistical binning enables an accurate coalescent-based estimation of the avian tree,” *Science*, vol. 346, no. 6215, p. 1250463, 2014.
- [247] E. D. Jarvis, S. Mirarab, A. J. Aberer et al., “Whole-genome analyses resolve early branches in the tree of life of modern birds,” *Science*, vol. 346, no. 6215, pp. 1320–1331, 2014.
- [248] C. Zhang, M. Rabiee, E. Sayyari, and S. Mirarab, “ASTRAL-III: Polynomial time species tree reconstruction from partially resolved gene trees,” *BMC Bioinformatics*, vol. 19, p. 153, 05 2018.
- [249] S. Mirarab, M. S. Bayzid, B. Boussau, and T. Warnow, “Datasets for: Statistical binning enables an accurate coalescent-based estimation of the avian tree; IDEALS,” 2022, <http://dx.doi.org/10.13012/C5MW2F2P>, last accessed March 14, 2022.
- [250] S. Song, L. Liu, S. V. Edwards et al., “Resolving conflict in eutherian mammal phylogeny using phylogenomics and the multispecies coalescent model,” *Proc Natl Acad Sci U S A*, vol. 109, no. 37, pp. 14 942–14 947, 2012.
- [251] M. Binet, O. Gascuel, C. Scornavacca, E. J. P. Douzery, and F. Pardi, “Fast and accurate branch lengths estimation for phylogenomic trees,” *BMC Bioinformatics*, vol. 17, no. 23, 2016.
- [252] J. Sukumaran and M. T. Holder, “DendroPy: a Python library for phylogenetic computing,” *Bioinformatics*, vol. 26, no. 12, pp. 1569–1571, 2010.
- [253] J. Huerta-Cepas, F. Serra, and P. Bork, “Ete 3: reconstruction, analysis, and visualization of phylogenomic data,” *Molecular biology and evolution*, vol. 33, no. 6, pp. 1635–1638, 2016.
- [254] E. D. Jarvis, S. Mirarab, A. J. Aberer, B. Li, P. Houde, C. Li, S. Y. Ho, B. C. Faircloth, B. Nabholz, J. T. Howard et al., “Phylogenomic analyses data of the avian phylogenomics project,” *GigaScience*, vol. 4, no. 1, pp. s13 742–014, 2015.
- [255] Y. Tabatabaei, S. Roch, and T. Warnow, “QR-STAR: A polynomial-time statistically consistent method for rooting species trees under the coalescent,” *Journal of Computational Biology*, vol. 30, no. 11, pp. 1146–1181, 2023.
- [256] Y. Tabatabaei, S. Roch, and T. Warnow, “Statistically consistent rooting of species trees under the multispecies coalescent model,” in *International Conference on Research in Computational Molecular Biology*. Springer, 2023, pp. 41–57.

- [257] S.-R. Jun, M. R. Leuze, I. Nookaew, E. C. Uberbacher, M. Land, Q. Zhang, V. Wan-chai, J. Chai, M. Nielsen, T. Trolle et al., “Ebolavirus comparative genomics,” *FEMS microbiology reviews*, vol. 39, no. 5, pp. 764–778, 2015.
- [258] C. Skarp-de Haan, A. Culebro, T. Schott et al., “Comparative genomics of unintrogressed *Campylobacter coli* clades 2 and 3,” *BMC Genomics*, vol. 15, no. 1, pp. 1–14, 2014.
- [259] S. S. Renner, G. W. Grimm, G. M. Schneeweiss, T. F. Stuessy, and R. E. Ricklefs, “Rooting and dating maples (*acer*) with an uncorrelated-rates molecular clock: implications for north american/asian disjunctions,” *Systematic biology*, vol. 57, no. 5, pp. 795–808, 2008.
- [260] M. P. Simmons and J. Gatesy, “Coalescence vs. concatenation: sophisticated analyses vs. first principles applied to rooting the angiosperms,” *Molecular phylogenetics and evolution*, vol. 91, pp. 98–122, 2015.
- [261] S. W. Graham, R. G. Olmstead, and S. C. Barrett, “Rooting phylogenetic trees with distant outgroups: a case study from the commelinoid monocots,” *Molecular biology and evolution*, vol. 19, no. 10, pp. 1769–1781, 2002.
- [262] C. Li, K. A. Matthes-Rosana, M. Garcia, and G. J. Naylor, “Phylogenetics of chondrichthyes and the problem of rooting phylogenies with distant outgroups,” *Molecular phylogenetics and evolution*, vol. 63, no. 2, pp. 365–373, 2012.
- [263] A. J. Drummond, S. Y. W. Ho, M. J. Phillips et al., “Relaxed phylogenetics and dating with confidence,” *PLoS biology*, vol. 4, no. 5, p. e88, 2006.
- [264] S. Naser-Khdour, B. Quang Minh, and R. Lanfear, “Assessing confidence in root placement on phylogenies: an empirical study using nonreversible models for mammals,” *Systematic Biology*, vol. 71, no. 4, pp. 959–972, 2022.
- [265] T. Wade, L. T. Rangel, S. Kundu et al., “Assessing the accuracy of phylogenetic rooting methods on prokaryotic gene families,” *PLoS One*, vol. 15, no. 5, p. e0232950, 2020.
- [266] A. R. Alanzi and J. H. Degnan, “Inferring rooted species trees from unrooted gene trees using approximate bayesian computation,” *Molecular phylogenetics and evolution*, vol. 116, pp. 13–24, 2017.
- [267] M. Mitzenmacher and E. Upfal, *Probability and computing: Randomization and probabilistic techniques in algorithms and data analysis*. Cambridge University Press, 2017.
- [268] P. L. Erdős, M. A. Steel, L. A. Székely, and T. J. Warnow, “A few logs suffice to build (almost) all trees (i),” *Random Structures & Algorithms*, vol. 14, no. 2, pp. 153–184, 1999.

- [269] P. L. Erdős, M. A. Steel, L. Székely, and T. J. Warnow, “A few logs suffice to build (almost) all trees: Part ii,” *Theoretical Computer Science*, vol. 221, no. 1-2, pp. 77–118, 1999.
- [270] T. Warnow, B. M. Moret, and K. S. John, “Absolute convergence: true trees from short sequences,” in *Symposium on Discrete Algorithms: Proceedings of the twelfth annual ACM-SIAM symposium on Discrete algorithms*, vol. 7, no. 09, 2001, pp. 186–195.
- [271] S. Roch, “Hands-on introduction to sequence-length requirements in phylogenetics,” in *Bioinformatics and Phylogenetics: Seminal Contributions of Bernard Moret*, T. Warnow, Ed. Springer, 2019, pp. 47–86.
- [272] D. Mallo, L. De Oliveira Martins, and D. Posada, “SimPhy : Phylogenomic Simulation of Gene, Locus, and Species Trees,” *Systematic Biology*, vol. 65, no. 2, pp. 334–344, Mar. 2016.
- [273] M. N. Price, P. S. Dehal, and A. P. Arkin, “FastTree 2—approximately maximum-likelihood trees for large alignments,” *PLoS One*, vol. 5, no. 3, p. e9490, 2010.
- [274] J. Willson, Y. Tabatabaei, B. Liu, and T. Warnow, “Disco+ qr: rooting species trees in the presence of gdl and ils,” *Bioinformatics Advances*, vol. 3, no. 1, p. vbad015, 2023.
- [275] S. Shekhar, S. Roch, and S. Mirarab, “Species tree estimation using ASTRAL: how many genes are enough?” *IEEE/ACM transactions on computational biology and bioinformatics*, vol. 15, no. 5, pp. 1738–1747, 2017.
- [276] Y.-b. Chan, Q. Li, and C. Scornavacca, “The large-sample asymptotic behaviour of quartet-based summary methods for species tree inference,” *Journal of Mathematical Biology*, vol. 85, no. 3, p. 22, 2022.
- [277] M. W. Hahn, T. De Bie, J. E. Stajich, C. Nguyen, and N. Cristianini, “Estimating the tempo and mode of gene family evolution from comparative genomic data,” *Genome Research*, vol. 15, no. 8, pp. 1153–1160, Aug. 2005.
- [278] J. Felsenstein, “Phylogenies and the Comparative Method,” *Am. Nat*, vol. 125, no. 125, pp. 3–147, 1985.
- [279] B. C. O’Meara, “Evolutionary Inferences from Phylogenies: A Review of Methods,” *Annual Review of Ecology, Evolution, and Systematics*, vol. 43, no. 1, pp. 267–285, Dec. 2012.
- [280] E. M. Volz, K. Koelle, and T. Bedford, “Viral Phylodynamics,” *PLoS Computational Biology*, vol. 9, no. 3, 2013.
- [281] B. Rannala, “The art and science of species delimitation,” *Current Zoology*, vol. 61, no. 5, pp. 846–853, Oct. 2015.

- [282] D. P. Faith, “Quantifying Biodiversity: a Phylogenetic Perspective,” *Conservation Biology*, vol. 16, no. 1, pp. 248–252, Feb. 2002.
- [283] C. Lozupone and R. Knight, “UniFrac : a New Phylogenetic Method for Comparing Microbial Communities,” *Applied and environmental microbiology*, vol. 71, no. 12, pp. 8228–8235, 2005.
- [284] S. L. Kosakovsky Pond and S. D. W. Frost, “Not So Different After All: A Comparison of Methods for Detecting Amino Acid Sites Under Selection,” *Molecular Biology and Evolution*, vol. 22, no. 5, pp. 1208–1222, May 2005.
- [285] R. Lanfear, J. J. Welch, and L. Bromham, “Watching the clock: Studying variation in rates of molecular evolution between species,” *Trends in Ecology & Evolution*, vol. 25, no. 9, pp. 495–503, Sep. 2010.
- [286] A. Rokas, B. L. Williams, N. King, and S. B. Carroll, “Genome-scale approaches to resolving incongruence in molecular phylogenies.” *Nature*, vol. 425, no. 6960, pp. 798–804, Oct. 2003.
- [287] P. Pamilo and M. Nei, “Relationships between gene trees and species trees,” *Molecular biology and evolution*, vol. 5, no. 5, pp. 568–583, 1988.
- [288] L. Liu, L. Yu, and S. V. Edwards, “A maximum pseudo-likelihood approach for estimating species trees under the coalescent model,” *BMC Evolutionary Biology*, vol. 10, no. 1, p. 302, 2010.
- [289] S. Song, L. Liu, S. V. Edwards, and S. Wu, “Resolving conflict in eutherian mammal phylogeny using phylogenomics and the multispecies coalescent model.” *Proceedings of the National Academy of Sciences*, vol. 109, no. 37, pp. 14942–7, Sep. 2012.
- [290] E. R. Moody, T. A. Mahendarajah, N. Dombrowski, J. W. Clark, C. Petitjean, P. Offre, G. J. Szöllősi, A. Spang, and T. A. Williams, “An estimate of the deepest branches of the tree of life from ancient vertically evolving genes,” *eLife*, vol. 11, p. e66695, Feb. 2022.
- [291] Q. Zhu, U. Mai, W. Pfeiffer, S. Janssen, F. Asnicar, J. G. Sanders, P. Belda-Ferre, G. A. Al-Ghalith, E. Kopylova, D. McDonald, T. Kosciolek, J. B. Yin, S. Huang, N. Salam, J.-y. Jiao, Z. Wu, Z. Z. Xu, K. Cantrell, Y. Yang, E. Sayyari, M. Rabiee, J. T. Morton, S. Podell, D. Knights, W.-j. Li, C. Huttenhower, N. Segata, L. Smarr, S. Mirarab, and R. Knight, “Phylogenomics of 10,575 genomes reveals evolutionary proximity between domains Bacteria and Archaea,” *Nature Communications*, vol. 10, no. 1, p. 5477, Dec. 2019.
- [292] M. Binet, O. Gascuel, C. Scornavacca, E. J. P. Douzery, and F. Pardi, “Fast and accurate branch lengths estimation for phylogenomic trees,” *BMC Bioinformatics*, vol. 17, no. 1, p. 23.

- [293] L. Bromham and D. Penny, “The modern molecular clock,” *Nature Reviews Genetics*, vol. 4, no. 3, pp. 216–224, Mar. 2003.
- [294] H. A. Ogilvie, R. R. Bouckaert, and A. J. Drummond, “StarBEAST2 Brings Faster Species Tree Inference and Accurate Estimates of Substitution Rates,” *Molecular Biology and Evolution*, vol. 34, no. 8, pp. 2101–2114, Aug. 2017.
- [295] E. Sayyari and S. Mirarab, “Fast Coalescent-Based Computation of Local Branch Support from Quartet Frequencies,” *Molecular Biology and Evolution*, vol. 33, no. 7, pp. 1654–1668, July 2016.
- [296] C. Zhang and S. Mirarab, “Weighting by Gene Tree Uncertainty Improves Accuracy of Quartet-based Species Trees,” *Molecular Biology and Evolution*, vol. 39, no. 12, p. msac215, Oct. 2022.
- [297] V. Lefort, R. Desper, and O. Gascuel, “FastME 2.0: A comprehensive, accurate, and fast distance-based phylogeny inference program,” *Molecular Biology and Evolution*, vol. 32, no. 10, 2015.
- [298] A. Stamatakis, “RAxML version 8: A tool for phylogenetic analysis and post-analysis of large phylogenies,” *Bioinformatics*, vol. 30, no. 9, pp. 1312–1313, 2014.
- [299] J. Sukumaran and M. T. Holder, “DendroPy: a Python library for phylogenetic computing.” *Bioinformatics*, vol. 26, no. 12, pp. 1569–1571, 2010.
- [300] A. Rambaut, “FigTree (v1.4.4),” 2023, <http://tree.bio.ed.ac.uk/software/figtree/>, Date last accessed: January 19, 2023.
- [301] C. Li, K. A. Matthes-Rosana, M. Garcia, and G. J. P. Naylor, “Phylogenetics of Chondrichthyes and the problem of rooting phylogenies with distant outgroups,” *Molecular Phylogenetics and Evolution*, vol. 63, no. 2, pp. 365–373, May 2012.
- [302] N. Moshiri and S. Mirarab, “A Two-State Model of Tree Evolution and Its Applications to Alu Retrotransposition,” *Systematic Biology*, vol. 67, no. 3, pp. 475–489, May 2018.
- [303] U. Mai and S. Mirarab, “Completing gene trees without species trees in sub-quadratic time,” *Bioinformatics*, vol. 38, no. 6, pp. 1532–1541, Mar. 2022.
- [304] V. Lefort, R. Desper, and O. Gascuel, “FastME 2.0: a comprehensive, accurate, and fast distance-based phylogeny inference program,” *Molecular biology and evolution*, vol. 32, no. 10, pp. 2798–2800, 2015.
- [305] Y. Tabatabaei, C. Zhang, S. Arasti, and S. Mirarab, “Species tree branch length estimation despite incomplete lineage sorting, duplication, and loss,” *bioRxiv*, 2025, <https://doi.org/10.1101/2025.02.20.639320>.
- [306] X. Jiang, S. V. Edwards, and L. Liu, “The multispecies coalescent model outperforms concatenation across diverse phylogenomic data sets,” *Systematic biology*, vol. 69, no. 4, pp. 795–812, 2020.

- [307] S. Mirarab, L. Nakhleh, and T. Warnow, “Multispecies Coalescent: Theory and Applications in Phylogenetics,” *Annual Review of Ecology, Evolution, and Systematics*, vol. 52, no. 1, pp. 247–268, Nov. 2021.
- [308] L. Liu, L. Yu, D. K. Pearl, and S. V. Edwards, “Estimating species phylogenies using coalescence times among sequences,” *Systematic Biology*, vol. 58, no. 5, pp. 468–477, Oct. 2009.
- [309] C. Solís-Lemus and C. Ané, “Inferring Phylogenetic Networks with Maximum Pseudolikelihood under Incomplete Lineage Sorting,” *PLOS Genetics*, vol. 12, no. 3, p. e1005896, Mar. 2016.
- [310] Y. Wang and L. Nakhleh, “Towards an accurate and efficient heuristic for species/gene tree co-estimation,” *Bioinformatics*, vol. 34, no. 17, pp. i697–i705, Sep. 2018.
- [311] R. Chaudhary, J. G. Burleigh, and D. Fernández-Baca, “Inferring species trees from incongruent multi-copy gene trees using the Robinson-Foulds distance.” *Algorithms for Molecular Biology*, vol. 8, p. 28, 2013.
- [312] C. Zhang, C. Scornavacca, E. K. Molloy, and S. Mirarab, “ASTRAL-Pro: Quartet-Based Species-Tree Inference despite Paralogy,” *Molecular Biology and Evolution*, vol. 37, no. 11, pp. 3292–3307, Nov. 2020.
- [313] B. Legried, E. K. Molloy, T. Warnow, and S. Roch, “Polynomial-Time Statistical Estimation of Species Trees Under Gene Duplication and Loss,” *Journal of Computational Biology*, vol. 28, no. 5, pp. 452–468, May 2021.
- [314] E. K. Molloy and T. Warnow, “FastMulRFS: fast and accurate species tree estimation under generic gene duplication and loss models,” *Bioinformatics*, vol. 36, no. Supplement_1, pp. i57–i65, July 2020.
- [315] M. L. Smith and M. W. Hahn, “New Approaches for Inferring Phylogenies in the Presence of Paralogs,” *Trends in Genetics*, vol. 37, no. 2, pp. 174–187, Feb. 2021.
- [316] C. Zhang, R. Nielsen, and S. Mirarab, “ASTER: A package for large-scale phylogenomic reconstructions,” *Molecular Biology and Evolution*, 2025.
- [317] C. Li, D. Wickell, L.-Y. Kuo, X. Chen, B. Nie, X. Liao, D. Peng, J. Ji, J. Jenkins, M. Williams et al., “Extraordinary preservation of gene collinearity over three hundred million years revealed in homosporous lycophytes,” *Proceedings of the National Academy of Sciences*, vol. 121, no. 4, p. e2312607121, 2024.
- [318] Y.-M. Ding, X.-X. Pang, Y. Cao, W.-P. Zhang, S. S. Renner, D.-Y. Zhang, and W.-N. Bai, “Genome structure-based juglandaceae phylogenies contradict alignment-based phylogenies and substitution rates vary with dna repair genes,” *Nature communications*, vol. 14, no. 1, p. 617, 2023.

- [319] A. S. Chanderbali, L. Jin, Q. Xu, Y. Zhang, J. Zhang, S. Jian, E. Carroll, D. Sankoff, V. A. Albert, D. G. Howarth et al., “Buxus and tetracentron genomes help resolve eudicot genome history,” *Nature communications*, vol. 13, no. 1, p. 643, 2022.
- [320] X. Guo, D. Fang, S. K. Sahu, S. Yang, X. Guang, R. Folk, S. A. Smith, A. S. Chanderbali, S. Chen, M. Liu et al., “Chloranthus genome provides insights into the early diversification of angiosperms,” *Nature communications*, vol. 12, no. 1, p. 6930, 2021.
- [321] E. Sayyari and S. Mirarab, “Fast coalescent-based computation of local branch support from quartet frequencies,” *Molecular biology and evolution*, vol. 33, no. 7, pp. 1654–1668, 2016.
- [322] M. Binet, O. Gascuel, C. Scornavacca, E. J. P. Douzery, and F. Pardi, “Fast and accurate branch lengths estimation for phylogenomic trees,” *BMC bioinformatics*, vol. 17, pp. 1–18, 2016.
- [323] S. Edwards and P. Beerli, “Perspective: gene divergence, population divergence, and the variance in coalescence time in phylogeographic studies,” *Evolution*, vol. 54, no. 6, pp. 1839–1854, Dec. 2000.
- [324] Y. Tabatabaei, S. Claramunt, and S. Mirarab, “Coalescent-based branch length estimation improves dating of species trees,” *bioRxiv*, 2025, <https://doi.org/10.1101/2025.02.25.640207>.
- [325] R. Davidson, P. Vachaspati, S. Mirarab, and T. Warnow, “Phylogenomic species tree estimation in the presence of incomplete lineage sorting and horizontal gene transfer,” *BMC Genomics*, vol. 16, no. Suppl 10, p. S1, 2015.
- [326] J. Willson, Y. Tabatabaei, B. Liu, and T. Warnow, “DISCO+QR: rooting species trees in the presence of GDL and ILS,” *Bioinformatics Advances*, vol. 3, no. 1, p. vbad015, 2023.
- [327] W. Fletcher and Z. Yang, “INDELible: A Flexible Simulator of Biological Sequence Evolution,” *Molecular Biology and Evolution*, vol. 26, no. 8, pp. 1879–1888, Aug. 2009.
- [328] J. Stiller, S. Feng, A.-A. Chowdhury, I. Rivas-González, D. A. Duchêne, Q. Fang, Y. Deng, A. Kozlov, A. Stamatakis, S. Claramunt et al., “Complexity of avian evolution revealed by family-level genomes,” *Nature*, pp. 1–3, 2024.
- [329] G. Butler, M. D. Rasmussen, M. F. Lin, M. A. Santos, S. Sakthikumar, C. A. Munro, E. Rheinbay, M. Grabherr, A. Forche, J. L. Reedy et al., “Evolution of pathogenicity and sexual reproduction in eight candida genomes,” *Nature*, vol. 459, no. 7247, pp. 657–662, 2009.
- [330] T. A. Williams, C. J. Cox, P. G. Foster, G. J. Szöllősi, and T. M. Embley, “Phylogenomics provides robust support for a two-domains tree of life,” *Nature ecology & evolution*, vol. 4, no. 1, pp. 138–147, 2020.

- [331] C. Petitjean, P. Deschamps, P. López-García, and D. Moreira, “Rooting the domain archaea by phylogenomic analysis supports the foundation of the new kingdom proteoarchaeota,” *Genome biology and evolution*, vol. 7, no. 1, pp. 191–204, 2015.
- [332] C. Zhang and S. Mirarab, “ASTRAL-Pro 2: ultrafast species tree reconstruction from multi-copy gene family trees,” *Bioinformatics*, vol. 38, no. 21, pp. 4949–4950, Sep. 2022.
- [333] J. P. Huelsenbeck and F. Ronquist, “MRBAYES: Bayesian inference of phylogenetic trees.” *Bioinformatics*, vol. 17, no. 8, 2001.
- [334] M. G. Johnson, L. Pokorny, S. Dodsworth, L. R. Botigué, R. S. Cowan, A. Devault, W. L. Eiserhardt, N. Epitawalage, F. Forest, J. T. Kim et al., “A universal probe set for targeted sequencing of 353 nuclear genes from any flowering plant designed using k-medoids clustering,” *Systematic biology*, vol. 68, no. 4, pp. 594–606, 2019.
- [335] F. A. Simão, R. M. Waterhouse, P. Ioannidis, E. V. Kriventseva, and E. M. Zdobnov, “BUSCO: assessing genome assembly and annotation completeness with single-copy orthologs,” *Bioinformatics*, vol. 31, no. 19, pp. 3210–3212, 2015.
- [336] J. P. Gogarten, H. Kibak, P. Dittrich, L. Taiz, E. J. Bowman, B. J. Bowman, M. F. Manolson, R. J. Poole, T. Date, T. Oshima et al., “Evolution of the vacuolar h+-atpase: implications for the origin of eukaryotes.” *Proceedings of the National Academy of Sciences*, vol. 86, no. 17, pp. 6661–6665, 1989.
- [337] N. Iwabe, K.-i. Kuma, M. Hasegawa, S. Osawa, and T. Miyata, “Evolutionary relationship of archaebacteria, eubacteria, and eukaryotes inferred from phylogenetic trees of duplicated genes.” *Proceedings of the National Academy of Sciences*, vol. 86, no. 23, pp. 9355–9359, 1989.
- [338] C. J. Cox, P. G. Foster, R. P. Hirt, S. R. Harris, and T. M. Embley, “The archaebacterial origin of eukaryotes,” *Proceedings of the National Academy of Sciences*, vol. 105, no. 51, pp. 20 356–20 361, 2008.
- [339] P. Houde, E. L. Braun, and L. Zhou, “Deep-Time Demographic Inference Suggests Ecological Release as Driver of Neoavian Adaptive Radiation,” *Diversity*, vol. 12, no. 4, p. 164, Apr. 2020.
- [340] K. Nadachowska-Brzyska, C. Li, L. Smeds, G. Zhang, and H. Ellegren, “Temporal Dynamics of Avian Populations during Pleistocene Revealed by Whole-Genome Sequences,” *Current Biology*, vol. 25, no. 10, pp. 1375–1380, May 2015.
- [341] U. Mai and S. Mirarab, “TreeShrink: fast and accurate detection of outlier long branches in collections of phylogenetic trees,” *BMC Genomics*, vol. 19, no. S5, p. 272, May 2018.
- [342] A. Wallberg, S. Glémén, and M. T. Webster, “Extreme Recombination Frequencies Shape Genome Variation and Evolution in the Honeybee, *Apis mellifera*,” *PLOS Genetics*, vol. 11, no. 4, p. e1005189, Apr. 2015.

- [343] J. D. Lozier, J. P. Strange, and S. D. Heraghty, “Whole genome demographic models indicate divergent effective population size histories shape contemporary genetic diversity gradients in a montane bumble bee,” *Ecology and Evolution*, vol. 13, no. 2, p. e9778, Feb. 2023.
- [344] S. P. Pfeifer, “Spontaneous Mutation Rates,” in *The Molecular Evolutionary Clock*, S. Y. W. Ho, Ed. Cham: Springer International Publishing, 2020, pp. 35–44.
- [345] R. G. Beiko, T. J. Harlow, and M. A. Ragan, “Highways of gene sharing in prokaryotes,” *Proceedings of the National Academy of Sciences*, vol. 102, no. 40, pp. 14 332–14 337, 2005.
- [346] P. Puigbò, Y. I. Wolf, and E. V. Koonin, “Search for a ‘tree of life’ in the thicket of the phylogenetic forest,” *Journal of biology*, vol. 8, pp. 1–17, 2009.
- [347] U. Mai, E. Charvel, and S. Mirarab, “Expectation-Maximization enables Phylogenetic Dating under a Categorical Rate Model,” *Systematic Biology*, vol. 73, no. 5, pp. 823–838, July 2024.
- [348] F. Forest, “Calibrating the tree of life: fossils, molecules and evolutionary timescales,” *Annals of Botany*, vol. 104, no. 5, pp. 789–794, 2009.
- [349] C. H. Langley and W. M. Fitch, “An examination of the constancy of the rate of molecular evolution,” *Journal of Molecular Evolution*, vol. 3, pp. 161–177, 1974.
- [350] S. Y. Ho and S. Duchêne, “Molecular-clock methods for estimating evolutionary rates and timescales,” *Molecular ecology*, vol. 23, no. 24, pp. 5947–5965, 2014.
- [351] S. Kumar and S. B. Hedges, “Advances in Time Estimation Methods for Molecular Data,” *Molecular Biology and Evolution*, vol. 33, no. 4, pp. 863–869, 02 2016.
- [352] M. dos Reis, P. C. Donoghue, and Z. Yang, “Bayesian molecular clock dating of species divergences in the genomics era,” *Nature Reviews Genetics*, vol. 17, no. 2, pp. 71–80, 2016.
- [353] A. Rambaut and L. Bromham, “Estimating divergence dates from molecular sequences.” *Molecular biology and evolution*, vol. 15, no. 4, pp. 442–448, 1998.
- [354] K. H. Wolfe, P. M. Sharp, and W.-H. Li, “Mutation rates differ among regions of the mammalian genome,” *Nature*, vol. 337, no. 6204, pp. 283–285, Jan. 1989.
- [355] J. L. Thorne and H. Kishino, “Divergence time and evolutionary rate estimation with multilocus data,” *Systematic Biology*, vol. 51, no. 5, pp. 689–702, Sep. 2002.
- [356] M. D. Rasmussen and M. Kellis, “Accurate gene-tree reconstruction by learning gene- and species-specific substitution rates across multiple complete genomes,” *Genome Research*, vol. 17, no. 12, pp. 1932–1942, Dec. 2007.

- [357] B. Rannala and Z. Yang, “Bayes estimation of species divergence times and ancestral population sizes using DNA sequences from multiple loci.” *Genetics*, vol. 164, no. 4, pp. 1645–1656, 2003.
- [358] B. S. Arbogast, S. V. Edwards, J. Wakeley, P. Beerli, and J. B. Slowinski, “Estimating Divergence Times from Molecular Data on Phylogenetic and Population Genetic Timescales,” *Annual Review of Ecology and Systematics*, vol. 33, no. 1, pp. 707–740, Nov. 2002.
- [359] B. Rannala, S. V. S. V. Edwards, A. Leaché, and Z. Yang, “The Multi-species Coalescent Model and Species Tree Inference,” in *Phylogenetics in the Genomic Era*, C. Scornavacca, F. Delsuc, and N. Galtier, Eds. No commercial publisher | Authors open access book, 2020, pp. 3.3:1–3.3:21.
- [360] S. Y. Ho, M. J. Phillips, A. Cooper, and A. J. Drummond, “Time dependency of molecular rate estimates and systematic overestimation of recent divergence times,” *Molecular biology and evolution*, vol. 22, no. 7, pp. 1561–1568, 2005.
- [361] F. K. Mendes and M. W. Hahn, “Gene tree discordance causes apparent substitution rate variation,” *Systematic biology*, vol. 65, no. 4, pp. 711–721, 2016.
- [362] D. T. Ksepka, J. L. Ware, and K. S. Lamm, “Flying rocks and flying clocks: disparity in fossil and molecular dates for birds,” *Proceedings of the Royal Society B: Biological Sciences*, vol. 281, no. 1788, p. 20140677, 2014.
- [363] F. T. Burbrink and R. A. Pyron, “The Impact of Gene-Tree/Species-Tree Discordance on Diversification-Rate Estimation: Diversification Rates from Gene Trees,” *Evolution*, vol. 65, no. 7, pp. 1851–1861, July 2011.
- [364] S. Nee, E. C. Holmes, R. M. May, and P. H. Harvey, “Extinction rates can be estimated from molecular phylogenies,” *Philosophical Transactions of the Royal Society of London. Series B: Biological Sciences*, vol. 344, no. 1307, pp. 77–82, Apr. 1994.
- [365] M. G. Harvey, G. A. Bravo, S. Claramunt, A. M. Cuervo, G. E. Derryberry, J. Battilana, G. F. Seeholzer, J. S. McKay, B. C. O’Meara, B. C. Faircloth et al., “The evolution of a tropical biodiversity hotspot,” *Science*, vol. 370, no. 6522, pp. 1343–1348, 2020.
- [366] A. R. Zuntini, T. Carruthers, O. Maurin, P. C. Bailey, K. Leempoel, G. E. Brewer, N. Epitawalage, E. Françoso, B. Gallego-Paramo, C. McGinnie et al., “Phylogenomics and the rise of the angiosperms,” *Nature*, pp. 1–8, 2024.
- [367] D. Schrempf, B. Q. Minh, N. De Maio, A. von Haeseler, and C. Kosiol, “Reversible polymorphism-aware phylogenetic models and their application to tree inference,” *Journal of Theoretical Biology*, vol. 407, pp. 362–370, Oct. 2016.
- [368] J. Peng, D. L. Swofford, and L. Kubatko, “Estimation of speciation times under the multispecies coalescent,” *Bioinformatics*, vol. 38, no. 23, pp. 5182–5190, 2022.

- [369] E. Volz and S. Frost, “Scalable relaxed clock phylogenetic dating,” *Virus evolution*, vol. 3, no. 2, p. vex025, 2017.
- [370] P. Sagulenko, V. Puller, and R. A. Neher, “TreeTime: Maximum-likelihood phylodynamic analysis,” *Virus Evolution*, vol. 4, no. 1, 01 2018, vex042.
- [371] M. dos Reis and Z. Yang, “Approximate likelihood calculation on a phylogeny for bayesian estimation of divergence times,” *Molecular biology and evolution*, vol. 28, no. 7, pp. 2161–2172, 2011.
- [372] J. L. Thorne, H. Kishino, and I. S. Painter, “Estimating the rate of evolution of the rate of molecular evolution.” *Molecular biology and evolution*, vol. 15, no. 12, pp. 1647–1657, 1998.
- [373] T.-H. To, M. Jung, S. Lycett, and O. Gascuel, “Fast dating using least-squares criteria and algorithms,” *Systematic biology*, vol. 65, no. 1, pp. 82–97, 2016.
- [374] U. Mai and S. Mirarab, “Log transformation improves dating of phylogenies,” *Molecular Biology and Evolution*, vol. 38, no. 3, pp. 1151–1167, 2021.
- [375] M. J. Sanderson, “Estimating absolute rates of molecular evolution and divergence times: a penalized likelihood approach,” *Molecular biology and evolution*, vol. 19, no. 1, pp. 101–109, 2002.
- [376] S. Mirarab and T. Warnow, “ASTRAL-II: Coalescent-based species tree estimation with many hundreds of taxa and thousands of genes,” *Bioinformatics*, vol. 31, no. 12, pp. i44–i52, June 2015.
- [377] S. Höhna, M. R. May, and B. R. Moore, “TESS: an R package for efficiently simulating phylogenetic trees and performing Bayesian inference of lineage diversification rates,” *Bioinformatics*, vol. 32, no. 5, pp. 789–791, 2016.
- [378] A. J. Helmstetter, S. Glemin, J. Käfer, R. Zenil-Ferguson, H. Sauquet, H. de Boer, L.-P. M. Dagallier, N. Mazet, E. L. Reboud, T. L. Couvreur et al., “Pulled diversification rates, lineages-through-time plots, and modern macroevolutionary modeling,” *Systematic Biology*, vol. 71, no. 3, pp. 758–773, 2022.
- [379] E. P. Derryberry, S. Claramunt, G. Derryberry, R. T. Chesser, J. Cracraft, A. Aleixo, J. Pérez-Emán, J. V. Remsen Jr., and R. T. Brumfield, “Lineage Diversification and Morphological Evolution in a Large-Scale Continental Radiation: The Neotropical Ovenbirds and Woodcreepers (aves: Furnariidae): Diversification of a Continental Radiation,” *Evolution*, vol. 65, no. 10, pp. 2973–2986, Oct. 2011.
- [380] C. D. Cadena, A. M. Cuervo, L. N. Céspedes, G. A. Bravo, N. Krabbe, T. S. Schulenberg, G. E. Derryberry, L. F. Silveira, E. P. Derryberry, R. T. Brumfield, and J. Fjeldså, “Systematics, biogeography, and diversification of Scytalopus tapaculos (Rhinocryptidae), an enigmatic radiation of Neotropical montane birds,” *The Auk*, vol. 137, no. 2, p. ukz077, Apr. 2020.

- [381] D. Moen and H. Morlon, “Why does diversification slow down?” *Trends in Ecology & Evolution*, vol. 29, no. 4, pp. 190–197, Apr. 2014.
- [382] T. Stadler, “On incomplete sampling under birth–death models and connections to the sampling-based coalescent,” *Journal of theoretical biology*, vol. 261, no. 1, pp. 58–66, 2009.
- [383] N. Cusimano and S. S. Renner, “Slowdowns in Diversification Rates from Real Phylogenies May Not be Real,” *Systematic Biology*, vol. 59, no. 4, pp. 458–464, July 2010.
- [384] L. J. Revell, L. J. Harmon, and R. E. Glor, “Under-parameterized Model of Sequence Evolution Leads to Bias in the Estimation of Diversification Rates from Molecular Phylogenies,” *Systematic Biology*, vol. 54, no. 6, pp. 973–983, Dec. 2005.
- [385] R. S. Etienne and J. Rosindell, “Prolonging the Past Counteracts the Pull of the Present: Protracted Speciation Can Explain Observed Slowdowns in Diversification,” *Systematic Biology*, vol. 61, no. 2, p. 204, Mar. 2012.
- [386] A. L. Pigot, A. B. Phillimore, I. P. F. Owens, and C. D. L. Orme, “The Shape and Temporal Dynamics of Phylogenetic Trees Arising from Geographic Speciation,” *Systematic Biology*, vol. 59, no. 6, pp. 660–673, Dec. 2010.
- [387] J. T. Weir, “Divergent timing and patterns of species accumulation in lowland and highland neotropical birds,” *Evolution*, vol. 60, no. 4, pp. 842–855, 2006.
- [388] X.-X. Pang and D.-Y. Zhang, “Impact of ghost introgression on coalescent-based species tree inference and estimation of divergence time,” *Systematic Biology*, vol. 72, no. 1, pp. 35–49, 2023.
- [389] M. J. Sanderson and H. B. Shaffer, “Troubleshooting molecular phylogenetic analyses,” *Annual review of ecology and Systematics*, vol. 33, no. 1, pp. 49–72, 2002.
- [390] Z. Yang, “PAML 4: phylogenetic analysis by maximum likelihood,” *Molecular biology and evolution*, vol. 24, no. 8, pp. 1586–1591, 2007.

APPENDIX A: LIST OF ABBREVIATIONS

AB archaea bacteria. 199, 201, 203, 211–215, 236–238

AD average distance. 13, 45, 48, 103–106, 110, 111, 114, 120, 121, 155, 156, 179, 186, 199, 200, 220, 221, 224, 239, 240, 248, 251, 253, 269

ADR Allman, Degnan, and Rhodes. 34–38, 75–78, 81, 82, 90, 91, 111

ARGs ancestral recombination graphs. 10

bp base pair. 121, 123, 124, 199, 200, 217, 240

BUSCO Benchmarking Universal Single-Copy Orthologue. 203, 233

CA-ML Concatenation analysis using maximum likelihood. 24, 25, 212, 213, 230–234, 236–238

CoalBL coalescent-based branch lengths. 245–247, 250–261, 284, 288

ConBL concatenation-based branch lengths. 246–248, 250–261, 268–270, 272, 274, 276, 277, 279–282, 284, 286–290

CU coalescent units. 13, 16, 30, 31, 125, 127–130, 142, 164, 165, 188, 189, 197, 198, 210, 215, 249

DLC Duplication-Loss-Coalescence. 14

DP dynamic programming. 26, 127, 128

FN false negative. 9, 102

FP false positive. 9

GDL Gene duplication and loss. 1, 3, 14, 28, 33, 57, 73, 74, 113, 165, 187, 188, 198, 200, 201, 206, 208, 210, 212, 214, 222–228, 240, 241, 291

GM General Markov. 18, 19

GS gene/species. 242–246

GTEE gene tree estimation error. 21, 42, 45, 48–50, 52–56, 103–107, 112, 120, 121, 153, 156, 159, 163, 176, 193, 195, 196, 200, 204–206, 220–227, 239, 240, 247–249, 254, 269

GTR Generalized Time Reversible. 12, 18, 19, 31, 101, 200, 249, 266

HGT Horizontal gene transfer. 1, 4, 14, 15, 28, 57, 187, 188, 198–201, 203, 209, 211–215, 229, 239, 241, 291

i.i.d. independently and identically distributed. 12

ILS Incomplete lineage sorting. 1–3, 13, 16, 25, 28, 29, 31, 33, 35, 44, 45, 48, 57, 73, 74, 83, 103, 104, 106–113, 119–121, 125, 126, 153, 155–158, 160–163, 165, 176, 179, 180, 187, 188, 196, 198–202, 204–209, 211, 212, 214, 215, 220–225, 227, 241–245, 247–249, 251–253, 255, 262, 263, 267, 268, 270, 274–277, 291, 292

JC69 Jukes-Cantor. 18, 22, 35

K-Pg Cretaceous-Paleogene. 261

LBA Long branch attraction. 20

LCA least common ancestor. 190, 192

LGT lateral gene transfer. 14

LTT lineage-through-time. 32, 251, 259, 262

MCMC Markov Chain Monte Carlo. 2, 23, 27, 187, 246, 250, 266

ML Maximum likelihood. 20–22, 24, 45, 126

MP Maximum parsimony. 20

MQSS Maximum Quartet Support Supertree. 26

MRCA most recent common ancestor. 6, 14, 261

MSA multiple sequence alignment. 11, 23

MSC Multi-Species Coalescent. 2–4, 12–14, 16, 24–27, 29, 30, 33–37, 44, 56, 57, 74–76, 83, 91, 92, 95, 98–101, 107, 113, 125–127, 129, 130, 155, 164, 165, 188, 189, 192, 196, 200, 242, 243, 291, 292

Mya million years ago. 257, 259, 261

nCD normalized clade distance. 102, 103, 105–109, 114, 115

NCM No Common Mechanism. 19

NJ Neighbor Joining. 21, 22, 26

NMSC Network Multi-Species Coalescent. 16, 28, 29, 292

PSMC Pairwise Sequentially Markovian Coalescent. 210

RF Robinson-Foulds. 8, 13, 21, 30, 42, 45, 103, 104, 106, 111, 115, 122–124, 155, 179, 186, 199, 200, 202, 203, 234, 235, 239, 240, 269

RMSE root-mean square error. 153, 157, 158, 168, 174, 176, 178, 248, 253, 254, 270, 274

RTT root-to-tip. 10, 209–212, 249

SU substitution units. 126, 127, 129, 130, 144, 151, 155, 164, 165, 168, 188–191, 197, 198, 200, 210, 214, 245, 246, 248, 249, 263, 267, 281, 282

SVD singular value decomposition. 27

TENT total evidence nucleotide tree. 44, 55, 56, 59

tMRCA time of the most recent common ancestor. 9, 248, 252–255, 273, 278

UCEs ultraconserved elements. 20, 44, 60, 257

WoL Web of Life. 199, 202, 203, 212

APPENDIX B: LIST OF TERMS

- additive . 22
- agrees . 7
- alleles . 10
- ancestor . 6
- anomalous . 13
- anomaly zone . 13
- bifurcating . 6
- binary . 6, 36, 40–43, 76, 81, 83, 91, 118
- bipartition encoding . 7
- bipartitions . 7, 45, 90, 102, 103
- blob . 16
- calibration densities . 31, 250, 286
- clades . 7, 42, 43, 102, 103, 259
- coalescent . 10, 164, 165, 242, 245, 261–263
- coalescent genes . 10
- compatible . 7
- concatenation . 11, 243–246, 249, 250
- contraction . 6
- correlated . 20
- deep coalescence . 13, 188, 197, 262, 263

descendant . 6
directed . 5
disagrees . 7
dissimilarity . 22
duplication event . 14, 190, 192, 195
duplication rate . 14, 15, 198–200, 206–208, 222–226, 228, 240
effective population size . 12, 242, 249
extinction rate . 32, 250, 251, 257, 258, 284
Felsenstein Zone . 20
Four-Point Condition . 22
fully resolved . 6, 83, 153, 193
gene tree . 10, 11, 33, 35–39, 44, 45, 54, 57, 103, 106, 111, 113, 120, 129–134, 136, 138, 141, 144–147, 151, 153, 155, 156, 158–160, 162, 164, 165, 180, 243, 245, 247, 256, 257, 262, 263, 279, 280
Hamming distance . 22
Hasse diagram . 7, 77
heterogeneity . 11, 125–127, 153, 155, 156, 164, 187–189, 211, 239
heterotachy . 19, 155
homogeneous . 17, 18
hybridization . 15
hybridization parameter . 15
identifiable . 24, 34, 36, 74, 75, 165
in-degree . 6

induced . 7, 40, 41, 45, 59, 60, 190

inheritance probability . 15

internal . 5, 37, 45, 77, 79, 83, 85, 99–102, 125, 128–133, 140, 142, 144–146, 148–154, 158, 161, 162, 164, 172, 173, 188, 191–193, 195, 204, 207–212, 214, 219, 220, 243, 244, 248, 250–254, 259, 262, 270, 271, 274, 276, 286

internode distance . 6

introgression . 15

level . 16

lineage . 10, 127, 128, 130, 203, 242, 243, 247

locus . 10, 33, 199, 200, 214, 240, 242

loss rate . 14, 15, 199, 200, 208, 223–226, 228

metric . 5, 6

molecular clock . 19, 35, 44, 74, 126, 153, 156

MUL-tree . 5, 14

multi-labeled . 5

multifurcating . 6

nearly additive . 22

net diversification rate . 32, 257

non-parametric bootstrapping . 23

non-reversible . 17, 35

nucleotides . 10

orthologs . 14, 202

out-degree . 6

outgroup . 29, 33, 34, 45, 60, 74, 114, 160–165, 169, 199, 200, 213, 217, 230, 232, 233, 248, 251, 253, 258, 267, 268, 270, 272, 274, 275, 284

paralogs . 14, 192

partitioned . 24, 202

patristic distance . 6, 164, 167, 168, 188, 201, 216, 217

phylogenetic network . 15

phylogenetic tree . 5, 60, 101

phylogeny . 5, 34, 257, 259, 283, 285–290

polytomy . 6

positively misleading . 24, 125

quartet . 8, 35, 75, 127, 187

quartet concordance factors . 16

rapid radiation . 13, 44, 202, 249

rate heterogeneity across sites . 19

recombination . 10, 15

refinement . 6

relative rate multiplier . 19

relaxed molecular clock . 20

restriction . 6

reticulation edges . 15

Reticulation nodes . 15

root-to-tip distance . 9

rooted . 5, 33–44, 55–57, 59, 60, 62–67, 69–78, 80–82, 99, 100, 102–106, 108–111, 113–118, 127, 151, 152

safety radius . 22

sample complexity . 24, 75, 99–102, 104, 113

semidirected . 16

singly-labeled . 5

site pattern . 18, 35, 74

sites . 10, 74, 156, 202, 203, 206, 212, 214, 235, 238, 247, 265

speciation . 10

speciation rate . 32

species tree . 10, 39, 40, 42, 44–46, 48, 73–76, 78, 81–84, 86, 88, 91, 92, 94–96, 98, 99, 125–129, 133, 196, 197, 199–203, 208, 212–215

star tree . 6

stationary . 17, 18

statistically consistent . 24, 35, 56, 73, 75, 95, 97, 98, 112, 113, 125, 130, 165, 243

statistically inconsistent . 24, 73, 243

strict molecular clock . 20, 33, 35, 74, 126, 127, 155, 165, 251, 264

supertree . 7

suppressing . 7, 60

taxa . 5, 34, 45, 59, 60, 78, 79, 82, 83, 91, 92, 97, 98, 100, 129, 149, 152, 156, 192, 197, 201–203, 213, 214, 216, 217, 222, 223, 225, 227, 230, 232, 234, 239–241, 244, 249, 250, 256, 269, 274, 276, 279, 286

terminal . 5, 129, 130, 134, 136, 138, 140, 141, 143, 144, 147, 149–153, 157, 158, 162, 164, 172, 173, 189, 192, 193, 197, 198, 204, 206–211, 214, 219, 243–245, 247, 250, 251, 253, 254, 257, 259–262, 270, 283, 287–290

time-calibrated . 31

time-reversible . 18

tips . 5, 247

topology . 5, 34, 36, 37, 42, 43, 46, 54, 59, 68, 69, 73–78, 80, 90, 92, 96–99, 101, 104, 108, 111, 113, 126, 127, 129, 130, 133, 134, 136, 138, 140, 141, 146–148, 153, 162, 166, 168, 169, 185, 187–193, 201–204, 208, 213, 215–217, 228, 230, 231, 238

transfer rate . 15

tree edges . 15

Tree nodes . 15

treeness . 9, 251–255, 257, 271

triangle inequality . 22

triplet . 8, 97, 98

ultrametric . 10, 247

uncorrelated . 20

unpartitioned . 25, 247

unresolved . 6, 103

unrooted . 5, 16, 34–46, 57, 59, 62, 66, 67, 73–79, 81–84, 90–95, 97–99, 101, 102, 108, 111–114, 129–134, 136, 138, 140, 144–148, 152, 168, 169, 175

weighted . 5